Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Selective Deferral for Budgeted LLM Answer Selection: Failure-Trace Signals under Matched-Budget Evaluation

Soroush Vahidi

sv96@njit.edu

New Jersey Institute of Technology

Research Article

Keywords: Large language models, answer selection, selective deferral, budgeted inference, test-time compute

Posted Date: May 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9783817/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Selective Deferral for Budgeted LLM Answer Selection: Failure-Trace Signals under Matched-Budget Evaluation

Soroush Vahidi[1*]

[1*]Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA.

Corresponding author(s). E-mail(s): sv96@njit.edu;

Abstract

We study final-answer selection from an already generated set of candidate large language model reasoning outputs under a fixed inference-budget contract. In this post-generation setting, candidate-producing methods have already spent their matched per-method budgets, and the remaining question is whether a gold-free selector can choose among their outputs without issuing additional model calls or accessing ground-truth labels at runtime.

We frame the problem as selective deferral, or answer-source routing, across candidate producers. We instantiate this framework with the Failure-Trace Allocator, a deterministic policy that routes between a frontier answer and external answer sources using runtime-observable failure-trace metadata and cross-candidate agreement. Its Low-Depth Guard defers away from the frontier when logged metadata indicate a structurally weak single-branch outcome, while its External-Consensus Fallback defers when three independent external baselines unanimously disagree with the frontier answer.

We evaluate on the GSM8K benchmark using Cohere command-r-plus-08-2024 under a matched per-method budget cap of six logical inference calls per example. On a disjoint 720-example aggregate spanning three evaluation seeds, the Failure-Trace Allocator achieves 80.69% accuracy versus 77.64%, 77.08%, and 75.14% for L1, S1, and TALE. Against a pooled four-answer ensemble, it remains ahead by point estimate but is not statistically separated. The evidence is therefore bounded: in the tested Cohere/GSM8K/$\boldsymbol{B = 6}$ setting, deterministic trace-guided routing improves over individual baselines and is competitive with pooled aggregation, while broader generalization requires independent replication.

Keywords: Large language models, answer selection, selective deferral, budgeted inference, test-time compute

1 Introduction

Scaling inference-time computation offers a practical path to improved accuracy without additional model training [1]. By generating multiple candidate answers under a fixed budget and aggregating or selecting among them, a system can achieve accuracy gains comparable to those from model scaling, at inference time [2]. In settings where the inference budget is strictly matched per method—every compared candidate-producing method receives the same number of model calls per example—the quality of the final answer depends entirely on the selection rule applied to the already-generated pool. Majority vote treats all candidates equally; but generation runs produce metadata that is richer than raw candidate answers, and this metadata may carry signal about which candidates are more reliable.

This paper studies that downstream decision problem explicitly: once several candidate producers have already run under a matched-budget contract, can a final selector use runtime-visible signals to decide when to keep one answer source and when to defer to another? We use the term *selective deferral* to emphasize that the selector may retain the default frontier answer on most examples while routing to an alternative source on a narrow subset identified from runtime metadata and cross-source agreement.

We study this question in a concrete, reproducible setting. The task is grade-school arithmetic reasoning on the GSM8K benchmark [3], the provider is Cohere command-r-plus-08-2024, and the budget is $B = 6$ inference calls per example. A frontier generator produces candidate answers via program-aided execution [4]; three independent external methods (L1, S1, and TALE) provide reference answers under the same budget. Given these already-generated outputs, the goal is to select a final answer without additional inference calls and without using gold labels.

Failure-Trace Allocator (FTA) is the specific instantiated policy studied here. FTA is a deterministic two-gate selective-deferral policy that relies on failure-trace metadata logged during normal frontier execution plus external answer agreement. The *Low-Depth Guard* (LDG) fires when the frontier's `override_reason` field signals a structurally weak single-branch output, substituting an external majority vote. The *External-Consensus Fallback* (ECO) fires when all three external baselines unanimously prefer a different answer than the frontier's selection. Both gates are strictly gold-free: no correctness labels, ground-truth answers, or example identifiers are accessed at decision time. The two gates operate in strict priority order; at most one fires per example.

On a disjoint Aggregate-720 combining three independent evaluation seeds (seeds 41, 61, 71; $300 + 120 + 300$ examples with zero example-id or question-hash overlap), Failure-Trace Allocator (FTA) achieves 80.69% accuracy versus 77.64% (L1), 77.08% (S1), and 75.14% (TALE). Source-stratified 95% bootstrap confidence intervals are strictly positive for all four pairwise comparisons, including versus the source-local best external (CI lower bound +1.11 pp). We also report a stronger pooled-answer ensemble baseline over frontier, L1, S1, and TALE; FTA remains ahead by point estimate on Final-300 and Aggregate-720, but that comparison is not statistically separated. Accordingly, the paper's claim is not broad superiority over ensembling; it is that gold-free failure-trace metadata can identify selective recovery cases under a

matched per-method budget contract. This contract is an evaluation design: generating frontier+L1+S1+TALE from scratch is not equivalent to a single six-call run. All claims are explicitly scoped to the tested provider and benchmark; we make no universal generalization claims.

The paper's contributions are therefore organized around the general decision problem rather than around a single implementation label:

- We define a post-generation answer-selection problem under matched per-producer budgets, where candidate generation and final selection are treated as separate stages.
- We study a class of deterministic, gold-free selective-deferral policies that use runtime-observable metadata and cross-source agreement rather than learned post-hoc supervision.
- We instantiate that policy class with FTA, a two-gate routing policy using failure-trace and agreement predicates that are interpretable and runtime-legal.
- We present a matched-budget evaluation protocol with disjoint aggregate evidence, making explicit that the budget contract applies to candidate producers and not to constructing the full multi-source pool from scratch.
- We report bounded empirical evidence: in the tested Cohere/GSM8K/$B = 6$ setting, FTA improves over the individual external baselines while remaining competitive with, but not statistically separated from, a strong pooled-answer baseline.

We also report five additional evaluated variants to clarify the boundary of the supported claim and to separate exploratory development from held-out evaluation evidence. Their implementation labels are retained in Section 5.1 and Table 10 for reproducibility, but they are not part of the final two-gate policy analyzed in the main results.

The remainder of the paper is organized as follows. Section 2 reviews related work on test-time compute scaling, verifier-guided reranking, and cost-aware routing. Section 3.3 specifies FTA precisely, including exact trigger conditions and a policy summary table. Sections 4 and 5 describe the evaluation protocol and report results with full confidence intervals. Section 5.1 summarizes ablation variants and additional evaluated outcomes. Sections 6–7 discuss implications, limitations, and directions for future work.

2 Related Work

Ensembles and answer aggregation. Classical ensemble methods show that simple aggregation rules can be strong baselines when component errors are at least partly decorrelated [5–9]. That perspective matters directly here: a pooled four-answer ensemble over frontier, L1, S1, and TALE is a meaningful comparator rather than a straw baseline. The paper therefore does not claim broad superiority over ensembling; instead, it asks whether interpretable trace-guided routing can be competitive with simple pooled aggregation under the same already-generated answer pool.

Self-consistency and reasoning-time aggregation for LLMs. Chain-of-thought prompting and zero-shot chain-of-thought showed that explicitly eliciting reasoning steps can improve mathematical and symbolic problem solving [10, 11]. Self-consistency then aggregates over multiple sampled reasoning paths by selecting the most frequent answer [2], with later work extending this idea to broader candidate pools [12]. Tree-structured and graph-structured reasoning methods broaden inference-time exploration further by organizing candidate traces into explicit search procedures [13, 14], while program-aided reasoning couples generation to executable intermediate structure [4]. Failure-Trace Allocator is adjacent to this line of work but different in scope: it does not sample new reasoning paths at selection time and it does not choose by answer frequency alone. Instead, it selects among already generated answers using runtime-visible trace metadata plus external agreement signals.

Verifier-based reranking and iterative correction. Another line of work uses learned verifiers, reward models, or self-correction mechanisms to rerank or revise candidate solutions. For GSM8K, Cobbe et al. [3] showed that an outcome verifier can materially improve answer selection when trained with correctness supervision. Process reward models refine this idea by scoring intermediate reasoning steps rather than only final answers [15]. Related work on self-verification and iterative self-correction asks the model to critique or refine its own reasoning before committing to a final answer [16, 17]. The method studied here differs in two ways: it uses deterministic gold-free routing predicates rather than a learned verifier, and it does not invoke additional corrective generation after the candidate pool is available.

Selective prediction, deferral, and calibration. Selective prediction studies when a system should abstain, defer, or route away from its default prediction on subsets where reliability is lower [18, 19]. Calibration analyses are relevant because such routing decisions depend on whether confidence-like signals carry meaningful reliability information [20]. Failure-Trace Allocator is closer to this selective-routing perspective than to pure majority voting: the frontier answer remains the default on most examples, while LDG and ECO intervene only on a narrow runtime-identifiable subset. This framing also clarifies the paper's proposition-level claim: if a gold-free runtime signal isolates examples where an alternative selector is more reliable than the frontier default, then conditional deferral on that subset can improve overall accuracy.

Budgeted inference and test-time compute. Recent work on test-time compute asks how additional inference effort should be allocated to improve reasoning accuracy [1, 21]. Budget-forcing approaches such as s1 regulate the amount of reasoning performed during generation [22], while broader cost-aware systems such as FrugalGPT and language-model cascades study routing under resource constraints [23, 24]. Adaptive computation inside a model or computation graph provides a related perspective on conditional compute allocation [25]. The present paper addresses a narrower problem: after candidate-producing methods have already run under the same per-method budget cap, can a deterministic zero-extra-call selector use logged trace metadata to improve the final answer choice?

3 Problem Setup and Method

3.1 Problem Setup

We consider post-generation answer selection under a fixed per-method budget parameter (budget = 6) on Cohere command-r-plus-08-2024 runs over `openai/gsm8k` examples. GSM8K, a grade-school math benchmark, was introduced by Cobbe et al. [3] and provides 1,319 test problems with numerical final answers that are amenable to exact-match evaluation. For each example, candidate-producing methods produce final-answer candidates under the same provider, model, and budget cap, together with method-specific runtime metadata. A downstream selector then chooses a final answer from those already-generated candidates. The objective is to maximize exact-match accuracy without using gold labels at runtime.

This framing separates *candidate generation* from *answer selection*. Candidate producers are the methods that spend the matched budget to generate answers. The selector is a post-generation policy that observes candidate answers and runtime-visible metadata and may either keep the default frontier answer or defer to another answer source. No new reasoning branches are sampled during this selection step.

The frontier method is a direct-reserve semantic frontier generator. At a high level, it allocates the fixed budget across a small set of direct-reserve attempts and frontier exploration branches, canonicalizes their final answers into answer groups, and selects a default frontier answer from the resulting candidate pool. During this execution it records runtime trace metadata: answer-group counts, support for the selected frontier group, whether the direct-reserve answer agrees with the frontier answer, and an `override_reason` field summarizing why the frontier selector committed to its final answer. The selector proposed in this paper treats those metadata as diagnostics, not as correctness labels.

Two `override_reason` values are central. The value `single_weak_frontier_branch` is emitted when the selected frontier answer is supported by a narrow or weak frontier trace, which suggests limited exploration of alternative answer groups. The value `direct_frontier_agree` is emitted when the direct-reserve and frontier answers agree internally. Both fields are produced before any comparison to the gold answer; they are therefore runtime-visible and gold-free.

The selective-deferral view of the problem is therefore: given a candidate pool and associated runtime-observable features, identify a subset of examples on which retaining the frontier answer is less reliable than deferring to another answer source. In this manuscript, the alternative sources are the three external baselines L1, S1, and TALE, and the final selector is deterministic.

Evaluation is reported on:

- final300 (seed 71),
- aggregate720 disjoint primary set (seeds 41, 61, 71),
- optional sensitivity set (+100; labeled sensitivity-only).

3.2 Terminology

The following terms are used consistently throughout the paper. Where helpful for readability, we use short manuscript names and keep the corresponding implementation identifiers in Table 16 for reproducibility.

Candidate pool
: The set of final answers produced by the available methods (frontier and external baselines) for a given example. Candidate-pool construction is separate from the downstream selection policy.

Frontier method
: The primary, computationally intensive answer selected by the direct-reserve semantic frontier policy (`direct_reserve_semantic_frontier_v2`), which serves as the default output before any policy gate fires.

External baseline
: Any of the three independently-sourced single-model methods used as reference signals: L1, S1, or TALE (defined below). External baselines provide gold-free consensus signals without access to gold labels.

L1
: An external method that selects the answer with the highest support count across sampled model outputs.

S1
: An external method that applies sequential budget-forcing to elicit a final answer within the allocated budget.

TALE
: An external method that uses prompt-level budget cues to guide answer selection.

Logical API call
: A single provider inference request counted against a candidate-producing method's budget.

Budget
: The integer upper bound $B = 6$ on logical API calls per example for each candidate-producing method, applied identically to all compared methods (matched-budget condition).

Matched budget
: All compared candidate-producing methods operate under the same per-method budget cap B, ensuring that accuracy differences are not confounded by unequal per-method computational limits. If a deployment generates frontier, L1, S1, and TALE outputs from scratch, its total wall-clock and API cost depends on which producers are executed.

Postrun replay
: Offline re-evaluation of a previously generated run artifact using a policy rule without issuing new provider API calls. Postrun replay is used to compute policy-specific accuracy from existing logs.

Gold-free runtime feature
: Any input to a policy decision rule that is derived solely from model outputs, run metadata, or structural properties of the candidate pool—never from gold labels, exact-match signals, or correctness indicators.

Held-out evaluation evidence
: Evidence collected on evaluation sets that were not used during rule development, meeting the disjointness, seed-independence, and aggregate-scale criteria documented in Section 4.

Provider-specific evidence
: Results obtained from a particular provider–model pairing (here, Cohere command-r-plus-08-2024). Claims are explicitly scoped to this setting and do not generalize beyond it without additional replication.

3.3 Method

We study a broader class of **post-generation selective-deferral selectors** that operate after candidate generation is complete (Figure 1). Such selectors do not generate new candidates and do not issue additional inference calls: they choose a final answer from already-produced outputs under the same per-method budget cap. The scientific object is therefore a routing problem over answer sources, not a new candidate-generation method.

Failure-Trace Allocator (FTA) is the deterministic instantiation of that broader framework studied in this paper. Its two gates—the *weak-frontier trigger* and the *direct-agreement trigger*—fire in strict priority order; at most one gate acts per example. All runtime decisions are gold-free: no `gold_answer`, `exact_match`, or correctness label is accessed.

Throughout the paper we use short manuscript names for readability. The combined Failure-Trace Allocator (FTA) policy is the *failure-trace-guided allocator*. Its two internal conditions are the *weak-frontier trigger* and the *direct-agreement trigger*. The three external baselines are abbreviated *L1*, *S1*, and *TALE*; the best-performing external on a given evaluation split is the *best external*. Implementation-name mapping is retained in Table 16 for reproducibility.

At each example, the selector consumes three classes of runtime-legal inputs: the frontier answer candidate, the three evaluated external-baseline answers, and failure-trace metadata emitted during normal frontier execution. The output is a single final answer plus an explicit policy outcome (`fix2`, `fix4`, or frontier default), so every decision can be audited after the run using only logged runtime fields. This interface is intentionally narrow: the instantiated policy does not depend on learned scores or post-hoc labels.

Conceptually, the allocator uses three abstract predicates. WeakFrontier(x) is true when the frontier trace indicates low support or a single-branch selection. ExternalMajority(x) returns an external answer when at least two of L1, S1, and TALE agree, with a deterministic fallback only when all three differ. ExternalConsensusDisagree(x) is true when L1, S1, and TALE unanimously agree with one another and disagree with the frontier answer. The raw log strings in Table 1 and Table 6 are implementation realizations of these predicates, not scientific assumptions tied to a private script.

The weak-frontier trigger is motivated by a structural reliability pattern. In practice, branches and candidate groups are formed from the frontier's logged branch traces: each produced branch yields a parsed answer and is assigned to an answer

group by normalized predicted text. Support counts are the integer counts of branches per answer-group (`answer_group_support_counts`) observed in the trace; higher support indicates multiple independent branches converging on the same answer. The field `override_reason` records the runtime decision pathway (for example, `single_weak_frontier_branch` or `direct_frontier_agree`) and is emitted by the frontier when a guard or post-processing event signals limited search breadth or internal agreement. Weak-frontier traces are emitted when expansion depth or family diversity falls below configured thresholds; direct-frontier agreement is emitted when the frontier's reserve selection matches the incumbent. These metadata fields are runtime-visible and used only for gold-free routing predicates; they are not derived from gold labels or post-hoc scoring. When the frontier trace indicates a single weak branch (for example, low-depth termination in the logged metadata), the resulting candidate is treated as evidence of under-exploration. In those cases, routing to an external majority answer provides an independent cross-policy signal without spending extra compute.

The direct-agreement trigger addresses a complementary pattern of selective deferral. If the frontier and its reserve agree internally, but all three external methods unanimously select a different answer, this unanimous disagreement is treated as high-confidence evidence that the frontier selection may be wrong under the matched-budget protocol. Partial external agreement (for example, only one or two externals disagreeing) is intentionally not used in the main method, because it is more ambiguous in this setting and did not provide validation-grade evidence.

Frontier metadata interface (runtime, gold-free). The allocator consumes only fields produced during standard frontier execution:

- **Candidate answer grouping:** parsed branch answers are canonicalized into answer groups (normalization/canonicalization details are implementation-level and documented in reproducibility artifacts).
- **Support counts:** each answer group has an integer branch-support count; these counts define whether support is concentrated or dispersed.
- **Logged pathway metadata:** `override_reason` records the frontier selector pathway, including `single_weak_frontier_branch` and `direct_frontier_agree`.
- **Agreement flags:** direct-frontier agreement and external answer agreement are computed from already generated answers.

All of these are runtime-visible model-output artifacts; none require ground-truth answers.

Conditional-deferral proposition. Let S be a runtime-identifiable subset of examples. Let $e_F(S)$ be frontier error rate on S and $e_M(S)$ be external-majority error rate on S. If $e_F(S) > e_M(S)$, then replacing frontier with external majority on S improves expected accuracy by

$$P(S)\,\big(e_F(S) - e_M(S)\big).$$

Low-Depth Guard (LDG) and External-Consensus Fallback (ECO) are empirical, gold-free routing predicates intended to identify such subsets without using correctness labels at runtime.

The control flow is a strict precedence chain (Figure 1, Table 6). The weak-frontier trigger is checked first; when it fires, the policy routes to an external majority answer and stops. This priority reflects the interpretation of the weak-frontier trace as a generation-quality warning: when exploration itself appears narrow, the policy does not also ask whether the frontier was internally self-consistent. If it does not fire, the policy checks the direct-agreement trigger; when that condition fires, the policy routes to the external unanimity answer. If neither trigger fires, the frontier answer is kept. The default therefore preserves the original frontier method on all examples where neither high-confidence failure trace is present. Conflict handling is deterministic: at most one gate can act, and external majority ties are resolved by the fixed priority order documented in Table 6. That fallback is used for reproducibility rather than claimed as optimal; the sensitivity noted in Table 14 shows that alternative tie orders change Aggregate-720 replay by less than one percentage point.

All trigger features are runtime-legal and gold-free by construction. They are computed only from model outputs and frontier trace metadata that are already visible at inference time, never from `gold_answer`, `exact_match`, example identifiers, or artifact paths. Table 15 gives the complete legality inventory, including allowed inputs and explicitly excluded fields. This legality boundary is central to the method definition, not only a reporting constraint.

Relative to L1, S1, TALE, and the frontier baseline, Failure-Trace Allocator (FTA) plays a different role in the stack. L1, S1, TALE, and frontier each provide candidate answers under the same per-method matched budget; the selector then decides whether to keep the frontier answer or defer to another answer source using trace and agreement signals. This separation between candidate generation and final allocation is what allows the policy to improve selection quality without additional model calls. Observed gains come from switching the final answer on a small trace-identified subset of examples rather than from increasing inference budget. Table 16 provides the manuscript-to-implementation name mapping.

3.4 Low-Depth Guard (LDG)

The Low-Depth Guard detects frontier answers arising from structurally weak search branches and defers to an external majority vote. The trigger relies exclusively on fields logged during the frontier run: the override reason string and the candidate-pool group count.

3.5 External-Consensus Fallback (ECO)

The External-Consensus Fallback identifies examples where the direct-agreement trigger fires—meaning the frontier agreed with its own internal reserve selection—yet all three external baselines unanimously chose a different answer. When this unanimous disagreement holds, the policy defers to the external consensus answer.

Table 1 Policy summary for the Failure-Trace Allocator. All inputs are gold-free runtime fields. Named constants and name mapping are documented in the appendix reproducibility tables.

Step	Operation
Inputs per example	
	`frontier_answer`, `result_metadata` (from frontier run logs)
	`l1_answer`, `s1_answer`, `tale_answer` (external baseline outputs)
	`override_reason`, `frontier_support` $\in \mathbb{Z}_{\geq 0}$, `candidate_pool_answer_group_count` $\in \mathbb{Z}_{>0}$
Step 1 —Low-Depth Guard (LDG)	
1a	**if** `override_reason` contains `"single_weak_frontier_branch"` **or** (`frontier_support` $= 0$ **and** `candidate_pool_answer_group_count` ≥ 2) **then:**
1b	Compute `ext` $\leftarrow$ majority answer among {`l1`, `s1`, `tale`} (at least 2 agree); if no majority, use priority order TALE $\succ$ S1 $\succ$ L1
1c	**if** `ext` exists **and** `ext` $\neq$ `frontier_answer`: **return** `ext`, `policy` $\leftarrow$ `fix2` *(guard fires; stop)*
Step 2 —External-Consensus Fallback (ECO)	
2a	**if** `override_reason` $=$ `"direct_frontier_agree"` **and** `l1_answer` $=$ `s1_answer` $=$ `tale_answer` **and** `l1_answer` $\neq$ `frontier_answer` **then:**
2b	**return** `l1_answer`, `policy` $\leftarrow$ `fix4` *(consensus fallback fires; stop)*
Step 3 —Default	
3	**return** `frontier_answer`, `policy` $\leftarrow$ `original`
Output	
	Selected answer and `policy` label; no gold labels accessed at any step.

3.6 Combined Policy Summary

Table 1 gives the complete pseudocode using exact runtime field names. Parameter settings and implementation-name mapping are documented in the appendix reproducibility tables.

The exact field-level trigger conditions are also tabulated in Table 6 for reproducibility.

4 Experimental Setup

4.1 Data, model, and protocol

All main comparisons are conducted on **Cohere command-r-plus-08-2024** with a fixed per-method budget cap of $B = 6$ logical API calls per example. The dataset is `openai/gsm8k`. Evaluation conditions are matched across all compared candidate-producing methods: every method operates under the same budget cap, provider, and model, so accuracy differences are not confounded by unequal per-method compute.

Matched-budget definition. A comparison is *matched-budget* if and only if every compared candidate-producing method is subject to the same maximum logical API

call count B per example, with the same provider and model. No method in the main comparison receives extra calls or a different model. Failure-Trace Allocator (FTA) is then applied as a post-generation selection policy over the logged candidate answers and frontier metadata. This contract supports clean method-to-method comparison, but it does not imply total compute parity with a single baseline run when multiple candidate-producing methods are executed together.

Evaluation role in the paper. The matched-budget protocol is part of the contribution because it isolates answer-source routing from candidate generation. The paper is not evaluating a new end-to-end solver that spends more compute; rather, it evaluates whether a post-generation selector can improve final-answer choice when each candidate producer is held to the same budget contract.

4.2 Cost and resource usage

Table 8 reports per-method resource usage measured from logged Aggregate720 runs on Cohere command-r-plus-08-2024. Reported metrics include average logical API calls, average input tokens, average output tokens, and average latency in seconds. Total tokens (input + output) can be derived by summing the respective columns. Budget parameter $B = 6$ is shown for completeness.

Failure-Trace Allocator (FTA) adds negligible selection overhead once candidate answers and frontier metadata are available, but the main deployment cost is candidate-pool construction. If a deployment must generate frontier, L1, S1, and TALE outputs from scratch, total wall-clock and API cost depends on which candidate producers are executed and may exceed a single method's $B = 6$ run. In other words, FTA is a zero-extra-call selector after generation, not a claim of one-run compute equivalence to all baselines combined. In the logged Aggregate720 runs, the frontier method has higher average latency than L1, S1, or TALE. We report cost in logical calls, tokens, and latency; monetary cost depends on provider pricing and is not used as the primary comparison metric. Dollar-normalized cost analysis, latency percentile distributions, and multi-provider token cost benchmarks are identified as future work (Section 6.1).

4.3 Parameter grounding

Table 14 (Appendix) documents the key parameters and design choices used in Failure-Trace Allocator (FTA), including their values, roles, and the robustness evidence available. We report parameter grounding and robustness checks rather than a full continuous hyperparameter sweep; a full threshold sweep is left for future work.

4.4 Primary splits and disjointness

The primary aggregate (Aggregate720) is formed from three disjoint sources: seed41 main300, seed61 independent120, seed71 final300. Cross-source overlap checks confirm zero shared example IDs and zero shared question hashes, ensuring that the aggregate evidence is drawn from non-overlapping subpopulations.

4.5 Statistics

We report paired bootstrap confidence intervals [26] ($n = 5{,}000$ resamples) on final300 and source-stratified aggregate confidence intervals on Aggregate720. The bootstrap resample count of 5,000 is standard for stable CI estimation at the sample sizes used. All confidence intervals are two-sided at the 95% level. When shown, the *bootstrap positive-delta fraction* is the fraction of bootstrap resamples in which the Failure-Trace Allocator (FTA) delta is positive; it is a descriptive bootstrap summary and should not be interpreted as a classical frequentist p-value.

Offline vs. live reproduction. The submission package is designed to support offline reproduction of the reported metrics, paper tables, and bootstrap summaries from processed artifacts without new provider API calls. Full end-to-end regeneration of raw model outputs requires commercial provider credentials and live API access, and may differ because of provider-side nondeterminism, model/version changes, rate limits, or transient service failures.

5 Results and Analysis

Main results are listed in Table 2 and Figure 2. Failure-Trace Allocator (FTA) is the main method; additional variants are summarized in Section 5.1 and Table 10.

Final-300. On the unbiased 300-example evaluation (seed 71, budget 6), Failure-Trace Allocator (FTA) reaches **86.67%** (260/300), versus L1 83.00% (249/300), S1 82.00% (246/300), and TALE 78.33% (235/300). Paired bootstrap 95% confidence intervals (5 000 resamples) for Failure-Trace Allocator (FTA) minus each baseline on this split:

- vs. L1: +3.67 pp, 95% CI [+0.33, +7.00], bootstrap positive-delta fraction 0.983
- vs. S1: +4.67 pp, 95% CI [+1.00, +8.33], bootstrap positive-delta fraction 0.993
- vs. TALE: +8.33 pp, 95% CI [+5.00, +12.00], bootstrap positive-delta fraction 1.000

All three CI lower bounds are strictly positive on this split.

Aggregate-720 (primary). The primary claim rests on the disjoint Aggregate-720 evaluation combining three independent sources (seeds 41, 61, 71; comprising 300, 120, and 300 examples respectively, with no example-id or question-hash overlap). Failure-Trace Allocator (FTA) reaches **80.69%** (581/720), versus L1 77.64% (559/720), S1 77.08% (555/720), and TALE 75.14% (541/720). Source-stratified bootstrap 95% confidence intervals (5 000 resamples):

- vs. L1: +3.06 pp, 95% CI [+0.83, +5.42], bootstrap positive-delta fraction 0.995
- vs. S1: +3.61 pp, 95% CI [+1.39, +5.69], bootstrap positive-delta fraction 0.999
- vs. TALE: +5.56 pp, 95% CI [+3.61, +7.50], bootstrap positive-delta fraction 1.000
- vs. source-local best external: +3.06 pp, 95% CI [+1.11, +5.00], bootstrap positive-delta fraction 0.999

All aggregate source-stratified CI lower bounds are strictly above zero. The margin vs. best external exceeds 1 pp even at the lower CI bound.

Pooled ensemble baseline. We also evaluate a simple pooled answer ensemble over the four already-generated answers: frontier, L1, S1, and TALE (Table 4). The primary pooled baseline uses strict majority vote over the four answers and falls back to the frontier answer for 2–2 ties or all-different cases; this tie-breaker is deterministic and gold-free. On Final-300, this Pooled-4 baseline reaches 84.33% (253/300), while Failure-Trace Allocator (FTA) reaches 86.67% (260/300), a +2.33 pp difference with paired bootstrap 95% CI [−0.67, +5.67]. On Aggregate-720, Pooled-4 reaches 79.86% (575/720), while Failure-Trace Allocator (FTA) reaches 80.69% (581/720), a +0.83 pp difference with source-stratified 95% CI [−1.11, +2.78]. An external-only L1/S1/TALE majority baseline is even closer: 85.33% on Final-300 and 80.28% on Aggregate-720. Thus, Failure-Trace Allocator (FTA) remains ahead by point estimate, but the pooled-ensemble comparison is not statistically separated and is treated as a robustness check rather than a retained superiority claim. The external fallback order used by Low-Depth Guard (LDG) is deterministic and fixed for reproducibility; it is not presented as an optimized learned ranker. A replay sensitivity check over alternative external tie orders changes Aggregate-720 Low-Depth Guard (LDG)-only accuracy from 80.00% to 80.42%, with frontier fallback on all-different external ties reaching 80.83%; these variants are reported only as sensitivity because they were not the main method.

Source-by-source breakdown. Table 3 gives the per-source accuracy for all evaluated methods. Estimates are consistent across sources: Failure-Trace Allocator (FTA) leads by point estimate in every individual source split, with seed-41 showing the smallest advantage (83.33% vs. 80.33% for all three baselines) and seed-71 showing the largest (86.67% vs. 83.00% for L1). The seed-61 source is substantially harder in absolute terms: frontier, L1, S1, and TALE all fall to 57.50%, 57.50%, 56.67%, and 54.17%, respectively. The provider/model/budget configuration is unchanged, and the mean question length is only modestly higher than the other sources (48.5 words vs. 46.1 for seed 41 and 44.6 for seed 71). The available evidence is therefore more consistent with source-sample difficulty than with a method-specific run configuration change. Failure-Trace Allocator (FTA) still leads the best external on seed 61 by +1.67 pp, and the source-stratified aggregate confidence interval accounts for this source variation.

Win/loss/tie decomposition. Appendix Table 12 gives the per-example win/loss/tie counts versus each baseline on Final-300. On the Final-300 evaluation, Failure-Trace Allocator (FTA) wins on 19 examples vs. L1 (losses: 8, ties: 273), 23 examples vs. S1 (losses: 9, ties: 268), and 27 examples vs. TALE (losses: 2, ties: 271). The extremely low loss counts vs. TALE confirm that the policy's recoveries substantially outweigh its regressions on this evaluation.

Gate-action decomposition. Table 7 decomposes the main method into gate actions. Across Aggregate-720, Low-Depth Guard (LDG) fires on 122 examples, External-Consensus Fallback (ECO) fires on 5 examples after Low-Depth Guard (LDG) does not fire, and no gate fires on 593 examples. The External-Consensus Fallback (ECO) actions are rare but positive in this artifact: before ECO, the frontier answer is incorrect on all five; after ECO, all five are correct. The same pattern holds on Final-300, where ECO fires on three examples and contributes three wins,

Table 2 Main validated results (final300 and aggregate720). Deltas are percentage points vs baselines on aggregate720.

Policy	Final300 Acc.	Final300 c/n	Agg720 Acc.	Agg720 c/n	ΔvsL1	ΔvsS1	ΔvsTALE	Main policy?
frontier original	76.67	230/300	75.28	542/720	−2.36	−1.81	+0.14	no
Low-Depth Guard (LDG)	85.67	257/300	80.00	576/720	+2.36	+2.92	+4.86	no
Failure-Trace Allocator (FTA)	**86.67**	**260/300**	**80.69**	**581/720**	**+3.06**	**+3.61**	**+5.56**	**yes**
TALE-default router	78.67	236/300	75.28	542/720	−2.36	−1.81	+0.14	no
Pooled-4 ensemble	84.33	253/300	79.86	575/720	+2.22	+2.78	+4.72	no
External-3 ensemble	85.33	256/300	80.28	578/720	+2.64	+3.19	+5.14	no
L1	83.00	249/300	77.64	559/720	0.00	+0.56	+2.50	no
S1	82.00	246/300	77.08	555/720	−0.56	0.00	+1.94	no
TALE	78.33	235/300	75.14	541/720	−2.50	−1.94	0.00	no
best external	83.00	249/300 (L1)	77.64	559/720 (L1)	0.00	+0.56	+2.50	no

Table 3 Source-by-source accuracy (%) and aggregate context. Sensitivity source (seed 31) is a labeled auxiliary set, not part of the headline Aggregate-720 evidence. Bold denotes the best value within each source row (ties allowed).

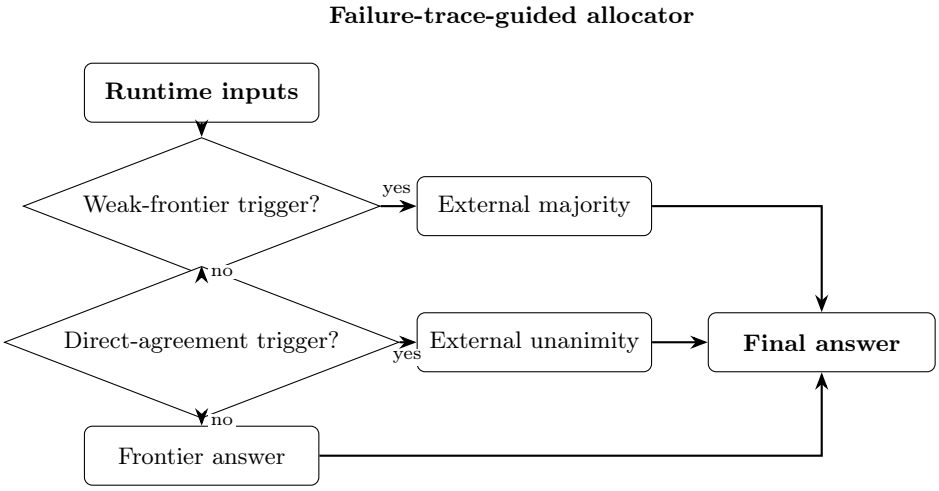
Source	Seed	N	Frontier	Low-Depth Guard (LDG)	Failure-Trace Allocator (FTA)	L1	S1	TALE	Δ(FTA vs best ext.)
main_300_seed41	41	300	81.00	82.67	**83.33**	80.33	80.33	80.33	+3.00
independent_120_seed61	61	120	57.50	**59.17**	**59.17**	57.50	56.67	54.17	+1.67
final_300_seed71	71	300	76.67	85.67	**86.67**	83.00	82.00	78.33	+3.67
sensitivity_100_seed31	31	100	73.00	80.00	**82.00**	76.00	77.00	**82.00**	0.00
Aggregate720 (primary)	41+61+71	720	75.28	80.00	**80.69**	77.64	77.08	75.14	**+3.06**

Table 4 Pooled answer-ensemble baselines computed from the same per-example frontier, L1, S1, and TALE outputs. Pooled-4 uses a strict majority over four answers and falls back to the frontier answer on ties or all-different cases. External-3 uses majority vote over L1/S1/TALE with the fixed, deterministic TALE≻S1≻L1 tie fallback used by the main method; tie sensitivity is summarized in Table 14. Deltas and win/loss/tie counts compare Failure-Trace Allocator (FTA) to the pooled baseline.

Split	Failure-Trace Allocator (FTA)	Pooled-4	External-3	Δ vs Pooled-4	W/L/T vs Pooled-4	95% CI
Final-300	260/300 (86.67%)	253/300 (84.33%)	256/300 (85.33%)	+2.33 pp	16/9/275	[−0.67, +5.67]
Aggregate-720	581/720 (80.69%)	575/720 (79.86%)	578/720 (80.28%)	+0.83 pp	28/22/670	[−1.11, +2.78]

zero losses, and zero ties relative to the frontier answer and to Low-Depth Guard (LDG)-only replay. Accordingly, ECO should be interpreted as a low-frequency high-precision safeguard in this artifact, not as a standalone high-coverage component with independent statistical reliability. Most of the aggregate recovery mass comes from LDG.

Sensitivity set (supporting diagnostic only). Table 3 includes a row for a 100-example set (seed 31, labeled "sensitivity-only") that was used as a development diagnostic during early calibration of precursor rules and the low-depth gate. Because this set was involved in parameter selection for earlier FIX variants, it is excluded from the primary Aggregate-720 evidence and from the bootstrap confidence intervals reported above. On this set, Failure-Trace Allocator (FTA) achieves 82.00%, tied with TALE and ahead of L1 (76.00%) and S1 (77.00%). When the sensitivity set is added to the aggregate ($N = 820$), the positive point-estimate advantage of Failure-Trace Allocator (FTA) over all three external baselines is preserved, and the overall picture is unchanged. All headline accuracy claims in this paper are based on Final-300 and Aggregate-720 only.



Journal-facing trigger names map to implementation identifiers in Table 16.

Fig. 1 Overview of the Failure-Trace Allocator (FTA) selective-deferral policy. The policy first checks the weak-frontier trigger; if it fires, the external majority answer is selected. Otherwise, the policy checks the direct-agreement trigger; if all external baselines unanimously disagree with the frontier answer, the unanimous external answer is selected. If neither trigger fires, the frontier answer is retained. Journal-facing trigger names map to implementation identifiers in Table A6.

5.1 Ablation Study and Additional Variants

We evaluate ablation and extension variants to test whether the final two-gate allocator can be improved by additional routing, extra-action, clustering, parsing, or pool-miss logic. Table 5 summarizes the evaluated variants; implementation labels are retained in the table for reproducibility. All variants were evaluated via postrun replay on existing candidate artifacts; no additional inference calls were issued for ablation purposes. None of the additional variants improved on Failure-Trace Allocator (FTA) with sufficient held-out evidence, and they are reported to delimit the supported claim.

Retained gate components. Low-Depth Guard (LDG) alone achieves 80.00% on Aggregate-720 (+2.36 pp over the frontier baseline), confirming that the Low-Depth Guard carries most of the combined gain. External-Consensus Fallback (ECO) alone achieves 75.00% on final-300—below Low-Depth Guard (LDG)—but contributes an additional +0.69 pp when combined with Low-Depth Guard (LDG), reflecting its role in recovering cases where Low-Depth Guard (LDG) does not fire. A score-absence calibration variant was evaluated but applied to zero examples in the primary splits, contributing no accuracy change; it is excluded from the combined policy.

Additional variants not included in the final policy. A TALE-default router produced zero switches on seeds 41 and 61, equalling TALE performance on both splits, and only a marginal net positive switch on seed 71 (+1 example over TALE, final-300 78.67% vs. 78.33%). This outcome is −8.00 pp below Failure-Trace Allocator (FTA) on the same split. Note that the Aggregate-720 accuracy for this router

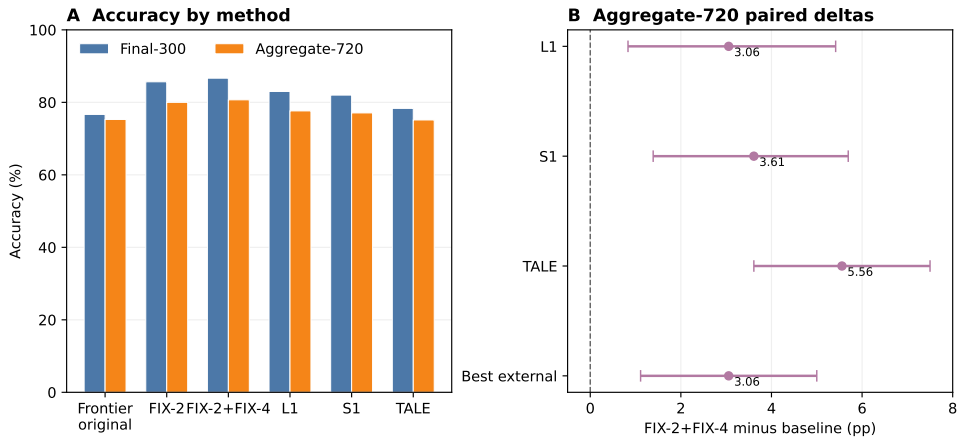


Fig. 2 Main accuracy summary for the main method. Panel A compares Failure-Trace Allocator (FTA) with the frontier baseline and external baselines on Final-300 and Aggregate-720. Panel B shows Aggregate-720 paired deltas for Failure-Trace Allocator (FTA) versus L1, S1, TALE, and the best external baseline; the zero line marks no improvement. The primary main evidence is the disjoint Final-300 and Aggregate-720 evaluation, and the figure is descriptive rather than a new claim.

(75.28%) numerically coincides with that of the frontier baseline; this reflects the zero-switch behavior on seeds 41 and 61 (the router reverts to TALE, and TALE plus the single seed-71 switch happen to sum to the same total as the frontier), not an equivalence by design. The TALE-default design assumed a reliable agreement region that did not materialise consistently across evaluation seeds; see Section 6 for interpretation. An extra-action variant showed a nonnegative pilot signal in the initial 40-case overlapping evaluation, but an independent 80-case follow-up yielded net -1 for extra frontier actions and net -4 for extra TALE retries relative to Failure-Trace Allocator (FTA). Early-stage signal that fails to replicate on disjoint data is not sufficient for inclusion in the final policy. A cluster-level selector achieved at most +0.67 pp in the best rule variant on final-300, below the threshold for inclusion in the main method. A robust parser variant produced zero accuracy-changing switches on both final-300 and Aggregate-720, indicating that parse-failure residuals are rare in this evaluation setting. A pool-miss diagnostic found 23 of 24 all-methods-wrong residuals to be generation-limited, with no selector-only remedy available; see the failure analysis in Section 5.2.

5.2 Failure Analysis

Table 9 and Figure 3 classify the 40 residual errors in the Final-300 evaluation by root cause. Three categories emerge. *All-methods-wrong* cases (24/40, 60.0%) are examples where every evaluated method—the frontier generator and all three external baselines—produced an incorrect answer. Because no correct answer appears in the candidate pool, no selection policy can recover these cases; they represent a hard ceiling for the current approach. *Parser/canonicalization failures* (12/40, 30.0%) are cases where the correct numerical answer was present in the pool but not extracted or

Table 5 Ablation variants and additional evaluated outcomes. Bold denotes the strongest result in the final-policy comparison block.

Policy	Target failure mode	Accuracy / delta summary	Main policy?
frontier original	baseline reference	final300=76.67%; agg720=75.28%	no
FIX-1 support-aware	tie / present-not-selected	77.00% (delta vs frontier +4.00 pp)	no
LDG (FIX-2) low-depth guard	single-weak-frontier-branch / low depth	80.00%	no (superseded by FTA)
FIX-3 calibration	score-absent non-weak-frontier edge cases	73.00% (applied 0 cases)	no
ECO (FIX-4) consensus gate	direct-frontier-agree + unanimous external disagreement	75.00% alone; 82.00% with LDG (FIX-2)	part of FTA
FTA (FIX-2+FIX-4)	low-depth + external-consensus	final300=86.67%; agg720=80.69%	**yes**
FIX-1+2+3+4	combined heuristic stack	81.00%	no
FIX-5 TALE-default router	agreement-region switching	overnight300=80.33%; final300=78.67%	no
FIX-6 extra-action variant	residual errors needing extra action	net vs FTA: frontier −1, TALE −4	no
FIX-7 cluster selector	cluster-level reranking / parser guards	best-rule local delta vs FTA: +0.67 pp	no
FIX-8 robust parser	parser/canonicalization errors	final300 delta +0.00 pp; agg720 +0.00 pp	no
FIX-9 pool-miss detector	generation-limited residuals	23/24 all-methods wrong generation-likely; 0 rule opportunities	no

matched, due to format mismatches between the model's output format and the evaluation matcher. *Present-not-selected* cases (4/40, 10.0%) are cases where the correct answer was available in the pool but the Failure-Trace Allocator (FTA) rule chain did not surface it.

Within the all-methods-wrong category, the dominant subtype is *multi-step composition traps* (10/24), problems where the error is embedded in intermediate reasoning rather than in the final answer selection step. Near-miss numeric errors (5/24), different-wrong-answer divergence with no correct cluster (4/24), and false consensus (2/24)—where all methods agree on the same incorrect answer—account for most of the remainder. False-consensus cases are the most resistant to any selection-based intervention: unanimous agreement provides no gating signal when the consensus itself is wrong.

These findings directly inform the scope of Failure-Trace Allocator (FTA) and the directions for future work. The all-methods-wrong class (60% of failures) is generation-limited: addressing it may require either stronger base generators or training-time interventions that improve coverage of hard compositional patterns, not better post-hoc selection policies. The parser/canonicalization class (30% of failures) is potentially tractable with improved answer normalization; however, the robust-parser variant reported in Table 10 produced zero accuracy-changing switches in this setting, suggesting the problem is subtler than simple string-matching improvements.

Table 6 Exact policy rule definitions for reproducibility. Abstract predicates are implemented by gold-free runtime fields only.

Component	Predicate / exact condition	Action	Gold-free rationale
Low-Depth Guard (LDG): WeakFrontier	`override_reason` contains `single_weak_frontier_branch` **or** (`frontier_support`= 0 **and** `pool_group_count`≥ 2)	Fallback to external majority vote; if no majority use priority TALE≻S1≻L1; else keep frontier	Uses only runtime metadata and baseline answers; no gold/exact labels
External-Consensus Fallback (ECO): ExternalConsensusDisagree	`override_reason`=`direct_frontier_agree` **and** `l1`=`s1`=`tale` **and** unanimous external ≠ frontier	Replace frontier with the unanimous external answer	Uses only runtime override reason and method answers
Combined Failure-Trace Allocator (FTA)	If Low-Depth Guard (LDG) fires: apply Low-Depth Guard (LDG) and stop; else if External-Consensus Fallback (ECO) fires: apply External-Consensus Fallback (ECO); else keep frontier	Single-fire precedence chain	No correctness labels in trigger chain

Table 7 Gate-action decomposition for the final Failure-Trace Allocator (FTA) policy. External-Consensus Fallback (ECO) fires only after Low-Depth Guard (LDG) does not fire. The final columns report correctness before and after ECO on the examples where ECO fires, relative to the original frontier answer.

Split	N	Low-Depth Guard (LDG) fires	ECO fires	No gate	ECO before	ECO after	ECO W/L/T
seed41	300	38	2	260	0/2	2/2	2/0/0
seed61	120	21	0	99	0/0	0/0	0/0/0
seed71 (Final-300)	300	63	3	234	0/3	3/3	3/0/0
Aggregate-720	720	122	5	593	0/5	5/5	5/0/0

Table 8 Aggregate720 per-method cost and resource summaries from logged runs. Provider: Cohere command-r-plus-08-2024; each candidate-producing method is evaluated with budget cap $B = 6$ (matched-budget condition). Avg total tokens = avg input + avg output. Monetary cost is not reported; it depends on provider pricing (see Section 4.2).

Method	Avg logical calls	Avg input tok.	Avg output tok.	Avg total tok.	Avg latency (s)	Budget
Frontier	2.60	1153.97	229.94	1383.91	13.30	6
L1	1.21	515.87	114.14	630.01	4.11	6
S1	2.62	1110.20	160.50	1270.70	5.15	6
TALE	1.44	612.38	117.90	730.28	4.63	6

The present-not-selected class (10% of failures) represents the only residual opportunity that a selection policy could in principle address without improved candidate generation.

6 Discussion and Limitations

Interpreting the trace-guided routing signals. The two gates in Failure-Trace Allocator (FTA) instantiate a selective-deferral framework in which the frontier answer is retained by default and overridden only on runtime-identifiable subsets without

Table 9 Failure taxonomy for final300 residual errors.

Failure group	Count	Share	Likely fix class
final300 total failures	40	1.000	mixed
all methods wrong	24	0.600	new candidate generation likely
parser / canonicalization issue	12	0.300	parser / canonicalization
present not selected	4	0.100	selector
Subtypes of all-methods-wrong			
multi-step composition trap	10	0.250	selector-only unlikely
near-miss numeric error	5	0.125	selector-only unlikely
different wrong answers, no correct cluster	4	0.100	mixed (rare selector possible)
same wrong answer, false consensus	2	0.050	selector-only unlikely
shared unit/rate trap	2	0.050	selector-only unlikely
quantity selection trap	1	0.025	selector-only unlikely

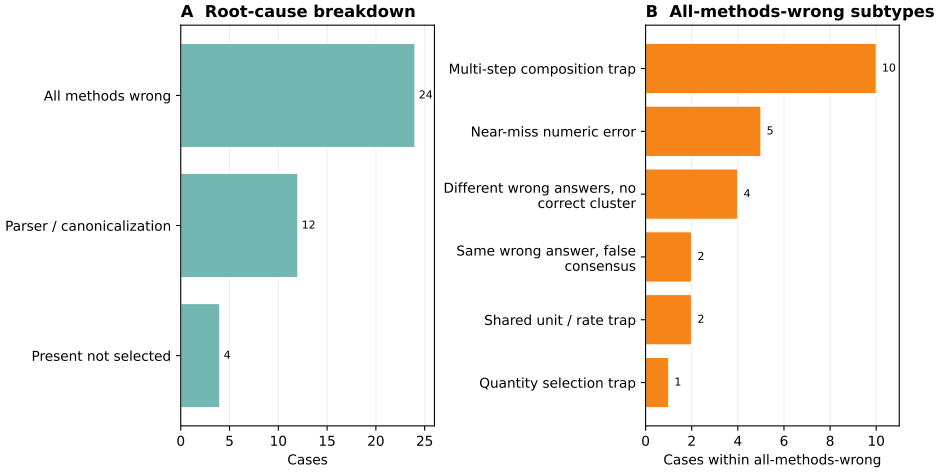


Fig. 3 Descriptive failure analysis for the 40 residual Final-300 errors. Panel A shows the top-level root-cause split; Panel B expands the all-methods-wrong cases into supported subtypes. This is a failure taxonomy, not a new benchmark result, and it is used to frame the residual headroom discussed in the text.

any access to gold labels. The Low-Depth Guard (LDG) addresses a structural weakness: when the frontier search terminates with a *single weak branch*—signalled by the `override_reason` field—the resulting answer is drawn from a narrow slice of the candidate space. In such cases, the external majority vote is empirically more reliable. The External-Consensus Fallback (ECO) targets a different pattern: when the direct-agreement trigger fires—signalling that the frontier selected its own answer confidently—yet all three independent external baselines unanimously prefer a different answer, that unanimous disagreement suggests that the frontier answer may be less reliable in this evaluation. Both signals are available as logged metadata from normal frontier execution; no additional inference or labelling is required.

Relation to pooled aggregation and selective routing. The pooled four-answer ensemble is an important comparator because classical ensemble results suggest that simple aggregation can already be powerful when component errors are not perfectly aligned. In this manuscript, the pooled baseline is deliberately treated as a strong reference point rather than as a weak baseline to be dismissed. Failure-Trace Allocator is competitive with that pooled baseline and remains ahead by point estimate on both Final-300 and Aggregate-720, but the paired/bootstrap confidence intervals include zero. Accordingly, the claim is not that trace-guided routing is broadly superior to ensembling. The narrower claim is that interpretable, gold-free answer-source routing can help in this matched-budget answer-selection protocol and remain competitive with a simple pooled ensemble built from the same already-generated answers.

Additional variants and why they were not adopted. A TALE-default router produced zero switches on seeds 41 and 61, equalling TALE performance exactly on both splits (80.33% and 54.17%, respectively). On seed 71 (Final-300), that router made one net positive switch over TALE (78.67% vs. 78.33%), a difference of a single example that is negligible compared to the +8.33 pp advantage of Failure-Trace Allocator (FTA) over TALE on the same split. The TALE-default design assumed a reliable agreement region that did not materialise consistently across evaluation seeds, making the router effectively inactive in most conditions.

An extra-action variant proposed collecting extra frontier or TALE-retry actions for predicted failure cases. The first 40-case overlapping pilot showed nonnegative action-value signal, but an independent 80-case follow-up (disjoint from the pilot) yielded net -1 for extra frontier actions and net -4 for extra TALE retries relative to Failure-Trace Allocator (FTA), with meaningful regressions on the corresponding held-out subsets (8.9% for frontier, 11.1% for TALE retry). The failure to replicate on disjoint data is the key indicator: the pilot signal appears to have been artefact-specific rather than generalisable.

A cluster-level selector showed only limited positive movement in offline replay, with a best-rule local gain of +0.67 pp on final-300, but it did not provide aggregate evidence or a robust policy improvement and was therefore not adopted for the final policy. A robust parser variant produced zero accuracy-changing switches on final-300 and aggregate-720. Taken together, these outcomes indicate that the targeted residual patterns are either rare or too weakly actionable in the current evaluation setting.

How aggregate-720 changes the evidence posture. The most substantive evidence shift in this work is the transition from a single-run 300-example evaluation (seed 41, where the CI lower bound vs. L1 still crossed zero at $[-0.48, +5.71]$) to the disjoint Aggregate-720 evidence (seeds 41, 61, 71). By combining three independently sampled unbiased evaluation splits, the stratified bootstrap CI narrows sufficiently that the lower bound vs. every external baseline is strictly positive. The source-stratified design controls for seed-specific variance, and the zero-overlap check on example-id and question-hash confirms independence. The result is that the point-estimate advantage of Failure-Trace Allocator (FTA) is consistent across all three seeds, which is the primary reason the aggregate CIs no longer include zero. This strengthens the case for the selective-deferral framing in the tested setting, but it does not transform the paper into a broad claim of universal routing superiority.

Evidence scope and generalization. The main findings are scoped to the tested evaluation setting. What this evidence does *not* establish is generalization beyond that scope: whether the same failure-trace signals are informative on other providers, benchmarks, or budget levels is an open empirical question. Existing repository artifacts include earlier MATH-500, AIME, OpenAI/Cohere, and budget-sweep outputs, but these do not contain the same four-method per-example records and frontier trace metadata needed to replay Failure-Trace Allocator (FTA) under the current protocol. They are therefore unsuitable as a clean second benchmark/provider/budget result for this selective-deferral study without new matched-budget generation. The most efficient next replication would be a second GSM8K provider run or a same-provider second benchmark with the identical frontier, L1, S1, and TALE logging contract. Section 6.1 enumerates the scope boundaries explicitly.

6.1 Limitations

We first summarize what is established by the current evidence, then state the main boundaries.

Established in this submission. The selective-deferral policy studied here is runtime-legal and gold-free by construction (Table 15); evaluation is disjoint by design across the Aggregate-720 sources; and FTA leads each individual external baseline (L1/S1/TALE) by point estimate with strictly positive source-stratified CI lower bounds on Aggregate-720. The pooled four-answer comparison is also reported and shows a closer gap whose CI includes zero.

Provider-specific scope. All main evidence is obtained on Cohere command-r-plus-08-2024. The method is not claimed to generalize to other providers, model families, or API versions without independent replication.

Benchmark scope. Results are scoped to GSM8K (`openai/gsm8k`) under the matched-budget answer-selection protocol described in Section 4. We make no universal accuracy claim beyond this dataset and task family.

Budget scope and compute accounting. The matched-budget claim is a per-method evaluation contract, not a claim that generating the full frontier/L1/S1/TALE candidate pool costs the same as one method run. We report logical calls, tokens, and latency; monetary cost depends on provider pricing and remains outside the main comparison metric.

Moderate margin and pooled-baseline caveat. The primary gain of FTA over the best external baseline on Aggregate-720 is +3.06 percentage points (95% CI $[+1.11, +5.00]$), while the pooled-four-answer comparison is +0.83 pp with CI including zero. These are meaningful but moderate margins that require broader replication.

Tie-breaker sensitivity. The TALE$\succ$S1$\succ$L1 fallback used when all three external answers differ is deterministic and gold-free, but it is not claimed to be optimal. Alternative tie orders move Aggregate-720 replay by less than one percentage point in the available artifacts.

Residual pool-miss failures. A residual failure class exists in which all methods—frontier and external baselines—produce incorrect answers for the same example. The proposed policy cannot recover from this class, as no correct answer is present in the candidate pool.

Additional evaluated variants. Additional cluster-selection, parser, and pool-miss variants were explored during development but did not meet the threshold for inclusion in the final policy. Their implementation labels are retained in Table 10 for reproducibility.

7 Conclusion

We presented Failure-Trace Allocator (FTA) as a deterministic selective-deferral policy for matched-budget LLM answer selection. On a disjoint 720-example aggregate evaluation (three independent seeds, zero overlap), Failure-Trace Allocator (FTA) achieves 80.69% (581/720) versus L1 77.64%, S1 77.08%, and TALE 75.14%. Source-stratified 95% bootstrap confidence intervals are strictly positive for all four baseline comparisons, with the lower bound vs. best external at +1.11 pp. On the Final-300 evaluation (seed 71, budget 6), Failure-Trace Allocator (FTA) reaches 86.67% (260/300), compared to 83.00% for L1, 82.00% for S1, and 78.33% for TALE. Against the pooled four-answer ensemble, Failure-Trace Allocator remains ahead by point estimate but is not statistically separated, so the relevant conclusion is competitiveness with strong aggregation baselines rather than broad superiority over ensembling.

The broader contribution is the framing of post-generation answer selection as a matched-budget selective-deferral problem over answer sources. FTA is one deterministic instantiation of that framework, consisting of two rule-based gates applied in strict priority order. The Low-Depth Guard (LDG) detects structurally weak frontier search branches using the logged `override_reason` field and substitutes the external majority vote. The External-Consensus Fallback (ECO) identifies cases where all three external baselines unanimously disagree with the frontier's selection, overriding to the consensus answer. Both gates are gold-free: no correctness labels, example identifiers, or artifact paths are accessed at runtime. The rules, trigger conditions, and exact field names are fully specified in Table 6 and Table 1, with reproducibility details documented in the appendix and submission package.

The evidence remains bounded to one tested provider/benchmark/budget setting, so the paper should be read as an answer-selection and selective-routing study rather than as a claim of a generally superior end-to-end solver. Further accuracy gains in this setting may require advances in candidate generation rather than selection policy. Failure analysis of the 40 residual errors on Final-300 shows that 24 (60%) are *all-methods-wrong* cases in which no evaluated method produced the correct answer; the dominant subtype is multi-step composition traps (10/24) where the error is embedded in intermediate reasoning rather than in the final answer selection step. Addressing this residual category may require either stronger base generators or training-time interventions that improve coverage of hard compositional patterns.

Table 10 Additional evaluated variants and their limitations.

Prototype	Result	Implication
TALE-default router (FIX-5)	Failed as the preferred policy on larger unbiased validation; no reliable transfer to the main-policy setting.	Keep Failure-Trace Allocator (FTA, implementation: FIX-2+FIX-4) as the main method; do not route by TALE-default in the manuscript's deployment-facing claim.
Extra-action variant (FIX-6)	Independent follow-up run showed negative net vs Failure-Trace Allocator (FTA, implementation: FIX-2+FIX-4) (extra_frontier −1, extra_tale −4).	No accuracy-changing extra-action policy was included in the final method.
Cluster selector (FIX-7)	Weak final300 gain (+0.67 pp) without sufficient held-out support.	Prototype only; insufficient for inclusion in the main method.
Robust parser/canonicalizer (FIX-8)	0 recoveries and 0 aggregate gain (final300 and aggregate720 deltas both 0).	No parser-driven policy change justified.
Pool-miss detector audit (FIX-9)	23/24 cases generation-likely; selector-only opportunity limited.	Residual error frontier is mostly candidate-generation limited under current logs.

Acknowledgements

The author thanks Professor Ioannis Koutis for his support, guidance, and helpful discussions during this work. The author is also deeply grateful to his mother for her continuous support and encouragement. The author acknowledges Cohere for providing research API credits that supported the model-inference experiments in this study.

References

[1] Snell, C., Lee, J., Xu, K., Kumar, A.: Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In: The Thirteenth International Conference on Learning Representations (2025). Oral. https://openreview.net/forum?id=4FWAwZtd2n

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

[3] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)

[4] Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Carbonell, J., Neubig, G.: PAL: Program-aided language models. In: International Conference on Machine Learning, pp. 10764–10799 (2023). PMLR

[5] Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)

[6] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1), 119–139 (1997)

[7] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3), 226–239 (1998)

[8] Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systems. Lecture Notes in Computer Science, pp. 1–15. Springer, Berlin, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1 . https://doi.org/10.1007/3-540-45014-9_1

[9] Zhou, Z.-H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Boca Raton, FL (2012)

[10] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837 (2022)

[11] Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In: Advances in Neural Information Processing Systems, vol. 35, pp. 22199–22213 (2022)

[12] Chen, X., Aksitov, R., Alon, U., Ren, J., Xiao, K., Yin, P., Prakash, S., Sutton, C., Wang, X., Zhou, D.: Universal Self-Consistency for Large Language Model Generation (2023). https://arxiv.org/abs/2311.17311

[13] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. In: Advances in Neural Information Processing Systems, vol. 36 (2023)

[14] Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., Hoefler, T.: Graph of thoughts: Solving elaborate problems with large language models. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, vol. 38, pp. 17682–17690 (2024). https://doi.org/10.1609/aaai.v38i16.29720

[15] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's verify step by step. In: International Conference on Learning Representations (2024)

[16] Weng, Y., Zhu, M., Xia, F., Li, B., He, S., Liu, S., Sun, B., Liu, K., Zhao, J.: Large Language Models are Better Reasoners with Self-Verification (2023).

https://arxiv.org/abs/2212.09561

[17] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-Refine: Iterative Refinement with Self-Feedback (2023). https://arxiv.org/abs/2303.17651

[18] Geifman, Y., El-Yaniv, R.: Selective Classification for Deep Neural Networks (2017). https://arxiv.org/abs/1705.08500

[19] Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. In: International Conference on Machine Learning (2019)

[20] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On Calibration of Modern Neural Networks (2017). https://arxiv.org/abs/1706.04599

[21] Wu, Y., Sun, Z., Li, S., Welleck, S., Yang, Y.: Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. arXiv preprint arXiv:2408.00724 (2024)

[22] Muennighoff, N., Yang, Z., Shi, W., Li, X.L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., Hashimoto, T.: s1: Simple test-time scaling. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 20275–20321. Association for Computational Linguistics, Suzhou, China (2025). https://doi.org/10.18653/v1/2025.emnlp-main.1025 . https://aclanthology.org/2025.emnlp-main.1025/

[23] Chen, L., Zaharia, M., Zou, J.: FrugalGPT: How to use large language models while reducing cost and improving performance. Transactions on Machine Learning Research (2024). ISSN 2835-8856

[24] Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Gontijo Lopes, R., Wu, Y., Michalewski, H., Saurous, R.A., Sohl-dickstein, J., Murphy, K., Sutton, C.: Language model cascades. arXiv preprint arXiv:2207.10342 (2022)

[25] Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016)

[26] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC, New York (1993)

Table 11 Reproducibility checklist.

Item	Status	Value	Artifact/reference
Dataset	complete	openai/gsm8k primary subsets	manifests + aggregate validation CSV
Provider / model	complete	Cohere / command-r-plus-08-2024	validation_manifest.json
Seeds	complete	41, 61, 71 (primary); 31 sensitivity	aggregate validation CSV
Budget	complete	$B = 6$ matched across all methods	manifests + per-example rows
Output roots	complete	major roots enumerated	supplementary package + LATEST_RESULTS_AND_CLAIMS.md
Disjointness proofs	complete	source overlaps = 0	final_postrun_metrics.json checks
Dry-run logs	complete	dry-run plans present for all validation runs	dry_run_call_plan.log
Postrun outputs	complete	final postrun metrics and CI tables present	final_fix24_all_external_postrun root
Bootstrap method	complete	5000 resamples; pooled + source-stratified summaries	bootstrap_ci_summary.csv
Offline reproduction	complete	reported metrics/tables reproducible from processed artifacts without new API calls	reviewer package + no-API table scripts
Reviewer package contents	complete	quickstart, environment specs, manifests, per-example outputs, summaries, bootstrap files	submission package inventory
Excluded/redacted material	complete	no API keys, secrets, `.env` files, private credentials, or unrelated local files	package policy
Live regeneration boundary	partial	full raw-output reruns require commercial credentials and may vary at replay time	provider access + live service behavior
Policy code files	complete	key selector / parser files listed	experiments/
Tests	partial	policy/parser tests present; not all scripts formalized as tests	`tests/`
Ablation variants	complete	Additional development variants are documented with negative or insufficient evidence	Table 10 + supplementary documentation
Rerun requirements	partial	Provider API access required for live reruns	validation scripts/manifests

A Reproducibility and Supplementary Material

This appendix collects reproducibility information, additional result decompositions, parameter grounding, runtime feature legality, and terminology-to-identifier mappings.

Reproducibility package contents. The submission package is designed to include an anonymous reviewer-access archive for offline reproduction of the reported summaries. The package contents include a README or reviewer quickstart, environment or requirements specifications, configs and run manifests, processed per-example outputs, method summaries, bootstrap summaries, and scripts that regenerate the reported paper tables and metrics without new API calls. The package excludes API keys, secrets, `.env` files, private credentials, and unrelated local files.

Table 12 Win/loss/tie decomposition for Failure-Trace Allocator (FTA) versus each external baseline on Final-300 (seed 71, budget 6, $n = 300$). A win is a case where FTA is correct and the baseline is not; a loss is the reverse; a tie is when both policies agree on correctness. The low loss counts confirm that FTA recoveries substantially outweigh regressions, particularly vs. TALE.

Comparison	Wins	Losses	Ties
FTA vs. L1	19	8	273
FTA vs. S1	23	9	268
FTA vs. TALE	27	2	271

Table 13 Examples of safe and unsafe claim formulations for this manuscript.

Type	Claim statement
Safe	Failure-Trace Allocator (FTA) outperforms L1/S1/TALE and best-external on aggregate720 (matched budget), with positive source-stratified bootstrap CI lower bounds for all baseline comparisons.
Safe	Failure-Trace Allocator (FTA) is ahead of pooled-answer ensemble baselines by point estimate on Final-300 and Aggregate-720, but the confidence intervals include zero.
Safe	Failure-trace-guided gold-free rules recover low-depth and external-consensus failure modes.
Safe	Additional development variants summarized in Table 10 do not provide sufficient evidence for inclusion in the final policy.
Safe	The reported metrics and paper tables are reproducible offline from processed artifacts in the submission package; exact live reruns require commercial API access and may differ.
Unsafe	Universal superiority over all reasoning methods.
Unsafe	Statistically separated superiority over pooled-answer ensemble baselines.
Unsafe	Universal gains across all math datasets/providers.
Unsafe	Reliable gains from the cluster-selector, parser, or pool-miss variants beyond the reported final policy.
Unsafe	Dollar/latency savings without direct measurement.
Unsafe	Exact live regeneration is guaranteed from the submission package alone.

Additional Result Decompositions

The following tables give the per-example win/loss/tie decomposition (Table 12) and claim-scope examples for manuscript wording (Table 13).

Parameter Grounding and Sensitivity

Table 14 documents the key parameters and design choices used in Failure-Trace Allocator (FTA), including their values, roles, and available robustness evidence.

Table 14 Parameter grounding for Failure-Trace Allocator (FTA) and the evaluation protocol. We report parameter grounding and available robustness checks rather than a full continuous hyperparameter sweep; a full threshold sweep is left for future work. All values are fixed and applied identically across all evaluated splits.

Parameter / Design choice *(value used)*	Role, justification	Robustness evidence available	Remaining limitation
Budget ($B = 6$)	Hard cap on logical API calls per example, ensuring matched-budget comparison. Fixed by provider run protocol; all methods use the same B.	Sensitivity source (+100 examples) included as separate labeled set; B held constant.	No multi-B sweep reported.
Low-depth guard: `single_weak_frontier_branch` *(string match; implementation constant)*	Identifies frontier answers from structurally weak search paths. Derived from failure-trace analysis of frontier run logs (`support_aware_selector.py`).	Trigger frequency and recovery/regression counts available in postrun artifacts.	No threshold variation sweep in manuscript.
Low-depth guard: `frontier_support`$= 0 \wedge$ `pool_group_count`≥ 2 *(integer thresholds: 0 and 2)*	Detects zero-support frontier outputs when the pool contains ≥ 2 distinct answer groups. Conservative: requires complete absence of support plus pool diversity.	Condition derived from offline evaluation review; trigger counts in supplementary artifacts.	No continuous threshold sweep.
External-consensus condition (ECO): `l1`=`s1`=`tale` $\wedge$ all $\neq$ frontier *(unanimity gate)*	Identifies unanimous external agreement diverging from frontier. Unanimity is a strong, conservative signal; partial agreement is not acted on.	Trigger frequency available from postrun replay artifacts.	No partial-consensus variant evaluated.
Bootstrap resamples *($n = 5{,}000$)*	Controls Monte Carlo variance in paired bootstrap CI estimation. Standard practice for stable 95% CIs at $n \leq 720$.	Single-seed results show consistent sign agreement.	No resample-count sensitivity sweep reported.
Evaluation seeds / splits *(seeds 41, 61, 71)*	Define disjoint aggregate and source-stratified analysis. Seeds span independent run batches; disjointness verified by example ID and question hash.	Source-stratified CIs show consistent positive sign for all baselines.	Single task/provider; no cross-dataset seed check.
Disjointness check *(overlap = 0)*	Prevents data leakage between primary sources. Verified programmatically on all three source sets.	Confirmed in postrun evaluation artifacts; overlap count = 0.	Within GSM8K only; no cross-dataset overlap check.
Provider / model *(Cohere command-r-plus-08-2024)*	Inference backend for all runs. Chosen for API availability and run stability during the experiment period.	All compared methods use the same provider/model.	Provider-specific; generalization not established.
External tie fallback *(TALE$\succ$S1$\succ$L1)*	Deterministic gold-free fallback when L1, S1, and TALE all disagree. Used as the fixed convention in the reported main policy rather than optimized on Aggregate720.	Alternative LDG-only replays on Aggregate720 give 80.14% for S1-first and 80.42% for L1-first; frontier fallback on all-different external ties gives 80.83%.	Tie order is not separately validated as an optimal policy.

Table 15 Runtime feature legality for Failure-Trace Allocator (FTA). All gating inputs are logged as side-effects of normal frontier execution and are available at inference time without any access to gold answers, correctness labels, example IDs, or artifact paths.

Feature	Source	Gold-free?	Why it is allowed
`override_reason`	Frontier run metadata	Yes	Logged search-state signal; e.g., `"single_weak_frontier_branch"` or `"direct_frontier_agree"`; not a correctness label.
`frontier_support`	Frontier run metadata	Yes	Integer support count from the frontier search; derived from pool structure, not correctness.
`candidate_pool_answer_group_count`	Frontier run metadata	Yes	Number of distinct answer groups in the pool; reflects diversity, not correctness.
`frontier_answer`	Frontier generator output	Yes	Frontier candidate selected before gating; proposed output only.
`l1_answer`	External L1 baseline output	Yes	L1 inference output, not the ground-truth label.
`s1_answer`	External S1 baseline output	Yes	S1 budget-forcing output only.
`tale_answer`	External TALE baseline output	Yes	TALE prompt-budgeting output only.
Excluded at runtime	N/A	N/A	`gold_answer`, `exact_match`, `example_id`, and artifact paths are never used in routing.

Runtime Feature Legality

Table 15 lists all features used by Failure-Trace Allocator (FTA) at inference time and confirms their gold-free status.

Table 16 Terminology and identifier mapping. Short names used throughout the manuscript and their implementation identifiers, roles, runtime availability, and gold-free status. Raw identifiers remain visible in Table 1, Table 6, and Table 15.

Journal-facing name	Implementation identifier	Role	Used at runtime?	Gold-free?
failure-trace-guided allocator	`FIX-2+FIX-4`	Main two-gate policy	Yes	Yes
weak-frontier trigger	`single_weak_frontier_branch`	Low-Depth Guard condition; `override_reason` value	Yes	Yes
direct-agreement trigger	`direct_frontier_agree`	External-Consensus Fallback condition; `override_reason` value	Yes	Yes
L1	`external_l1_max`	External baseline; L1-max answer selector	Yes	Yes
S1	`external_s1_budget_forcing`	External baseline; S1 budget-forcing method	Yes	Yes
TALE	`external_tale_prompt_budgeting`	External baseline; TALE prompt-budgeting method	Yes	Yes
best external	best external comparator	Highest-accuracy external baseline on a given split	No	Yes

Terminology and Identifier Mapping

Table 16 maps manuscript short names to their implementation identifiers, roles, runtime availability, and gold-free status.

Data Availability

The GSM8K benchmark questions used for evaluation are available from the original source [3]. This submission is accompanied by a package designed for anonymous reviewer access and offline reproduction of the reported metrics, tables, and figures. The processed artifacts in that package include per-example records, experiment manifests, method summaries, bootstrap summaries, and paper-table generation outputs; these materials are sufficient to recompute the reported summary metrics without live model calls. A public archival release of the processed artifacts will be created upon publication, preferably in a versioned repository with a persistent identifier.

Code Availability

This submission is accompanied by a package designed to include the analysis code, table-generation scripts, and reproduction instructions needed to recompute the reported paper metrics from the processed artifacts offline. Full end-to-end regeneration of raw model outputs requires commercial provider credentials and live API access to Cohere command-r-plus-08-2024, and may vary because of provider-side nondeterminism, model/version changes, rate limits, or transient service failures. No API keys, secrets, `.env` files, private credentials, or unrelated local files are included in the reviewer package or any future public release. A public archival release of the code

and processed artifacts will be created upon publication, preferably in a versioned repository with a persistent identifier.

Funding

This work was supported by Cohere research API credits. No other specific grant funding was received from any funding agency in the public, commercial, or not-for-profit sectors.

Competing Interests

The authors declare no competing interests.

Use of AI Tools

The author used ChatGPT, Cursor, Claude, GitHub Copilot, Cohere models, and Perplexity AI during parts of the coding, literature-search, manuscript-drafting, editing, and artifact-preparation workflow. These tools were used for assistance with code navigation, implementation support, language editing, reference discovery, and drafting support. All scientific claims, experimental design choices, code changes, analyses, reported results, citations, and final manuscript text were reviewed and approved by the author, who takes full responsibility for the content of the paper.

Author Contributions

Soroush Vahidi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing — original draft, Writing — review and editing, Visualization.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- mljfinalsupplementaryreproducibilitycurrent.zip