PAY FOR THE DIFFERENCE: EXACT FEEDBACK REDUCTION FOR LANGUAGE MODEL COMPARISON

Anonymous authors
Paper under double-blind review

ABSTRACT

Model iteration requires comparing checkpoints, but not every response score affects the comparisons of interest. Caching avoids rescoring identical responses, but still requires scores whose contributions cancel in the requested comparison. The challenge is to identify necessary scores before acquisition, accounting for the different requirements of means and covariances. Coupled generation increases response sharing while preserving each model's output distribution; Contrast-Aware Feedback Acquisition (CAFA) then aggregates report coefficients to skip unnecessary scores and exactly recover the fully scored sample statistics. For a fixed sample bank and an otherwise unrestricted scorer, we characterize the necessary and sufficient queries for exact recovery of comparison means, either alone or jointly with ordinary sample covariances. On controlled updates, CAFA eliminates 55.8% and 90.1% of the scoring inputs remaining after caching for Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct, respectively. Across three online group relative policy optimization (GRPO) trajectories, Independent Star + CAFA reaches an observed root mean square Monte Carlo standard error (RMS MCSE) of at most 0.04 at 46.1–51.3% lower recorded generation-plus-feedback cost than Independent autoregressive (AR) + Cache-only on the evaluated budget grid. All 504 report cells reproduce the fully scored statistics. We provide a modular implementation for existing response banks and external scorers, making exact feedback reduction a reusable step in language-model comparison.

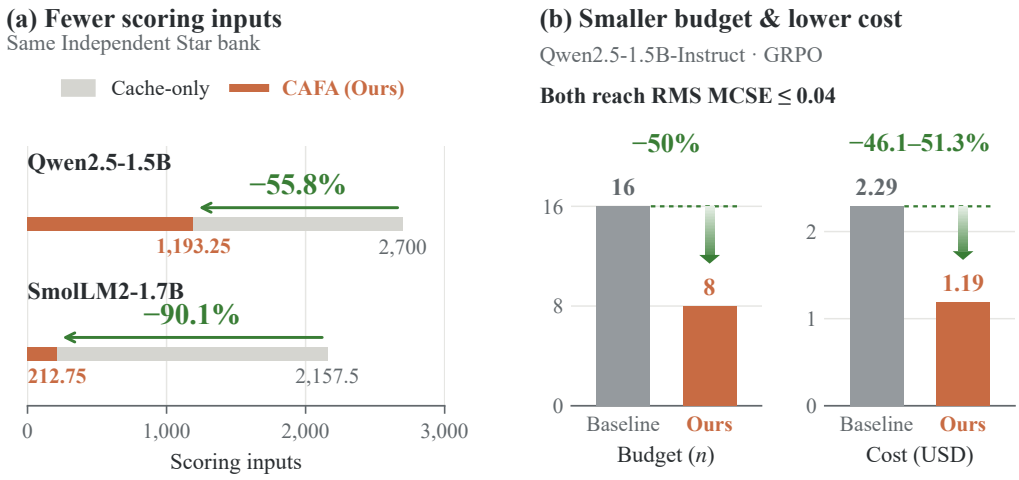


Figure 1: **Fewer inputs, smaller budget, lower cost.** Left: Cache-only versus CAFA (Ours) on identical Independent Star response banks and draw budgets; CAFA preserves the fully scored report statistics across all four repetitions (Appendix Table 5). Right: complete workflows for online GRPO with Qwen2.5-1.5B-Instruct. Independent AR + Cache-only (Baseline, $n = 16$) and Independent Star + CAFA (Ours, $n = 8$) both reach RMS MCSE ≤ 0.04 across all six matched seed–repetition cells. Costs include generation and feedback per 128-prompt comparison. The panels use separate experiments (Sections 5.2 and 5.3).

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.

1 INTRODUCTION

Model development is a sequence of comparisons. Instruction tuning, preference optimization and reinforcement learning from feedback (Ouyang et al., 2022; Rafailov et al., 2023; Schulman et al., 2017; Shao et al., 2024) produce checkpoints that must be evaluated against their predecessors to determine whether an update improves performance, quantify any improvement and select a checkpoint. When evaluation relies on a paid model-based scorer or a human rater, the question is which scores are needed for the comparison.

A report specifies the outputs and statistics to compute. Suppose a base model and three updates produce the same response to a prompt. Scoring the four response occurrences requires four score acquisitions; caching reduces this to one. The three gains over the base require zero score acquisitions because the common score cancels in every difference. Adding the absolute base score to the report requires that score. The required scores depend on the report and can be identified before scoring.

Independent generation can produce few exact matches between complete responses. Coupled generation increases sharing while preserving each checkpoint's output distribution (Benz et al., 2026; Wen, 2026). Contrast-Aware Feedback Acquisition (CAFA) then aggregates coefficients over identical scoring inputs to recover the requested means and covariances with only the necessary scores. Throughout, *Cache-only* denotes scoring after cache deduplication; *CAFA (Ours)* denotes the proposed scoring strategy, regardless of sampler. Independent autoregressive (AR), shared-token AR, Independent Star and total-variation (TV) tree denote sampling methods. Figure 1 contrasts same-bank CAFA savings with complete-workflow improvements from coupled generation and CAFA.

CAFA exactly recovers fully scored sample means and, when requested, ordinary covariances of per-draw contrasts. With unrestricted scores, each reporting mode's support is necessary. A response can cancel from the mean while affecting covariance, so the requested statistics are part of the report. We evaluate the method on fixed controlled updates to isolate reductions in score acquisitions on identical samples, and on online group relative policy optimization (GRPO) trajectories to assess the cost and statistical precision of checkpoint comparisons (Figure 1). We make the following contributions:

(i) We introduce CAFA, which aggregates coefficients over exact scoring identities for any joint sampler and supports both mean-only and mean-with-covariance reports, including general linear reports with absolute anchors.

(ii) We establish exact recovery and query necessity for both modes, show that invertible coordinate changes preserve the required support and that adding outputs takes the union of supports, and characterize when immediate CAFA attains the batch support.

(iii) We evaluate the method on controlled updates and online GRPO trajectories. CAFA removes feedback beyond caching, while coupled generation with CAFA lowers cost at a common root mean square Monte Carlo standard error (RMS MCSE); all 504 report cells reproduce the fully scored statistics.

(iv) We provide an installable implementation with interfaces for existing response banks, external scorers and Independent Star generation from local checkpoints (Appendix I).

2 PROBLEM SETUP

Policies, prompts and scores. Fix a base language-model policy p_0 and updated policies $p_1, \dots, p_J$. Each policy is defined by its decoding settings and stopping rule, with any length cap treated as part of that rule. Distinct prompts x_i, $i = 1, \dots, P$, carry positive weights w_i summing to one ($w_i = 1/P$ in all our analyses). A scorer r maps a prompt and a response to a real number. The scorer is fixed or versioned, with its identity defined by the rubric, judge deployment and settings. For a stochastic judge, we store and reuse the first score for each input in a versioned cache. Write $\mu_j(x_i) = \mathbb{E}_{Y \sim p_j(\cdot|x_i)}[r(x_i, Y)]$.

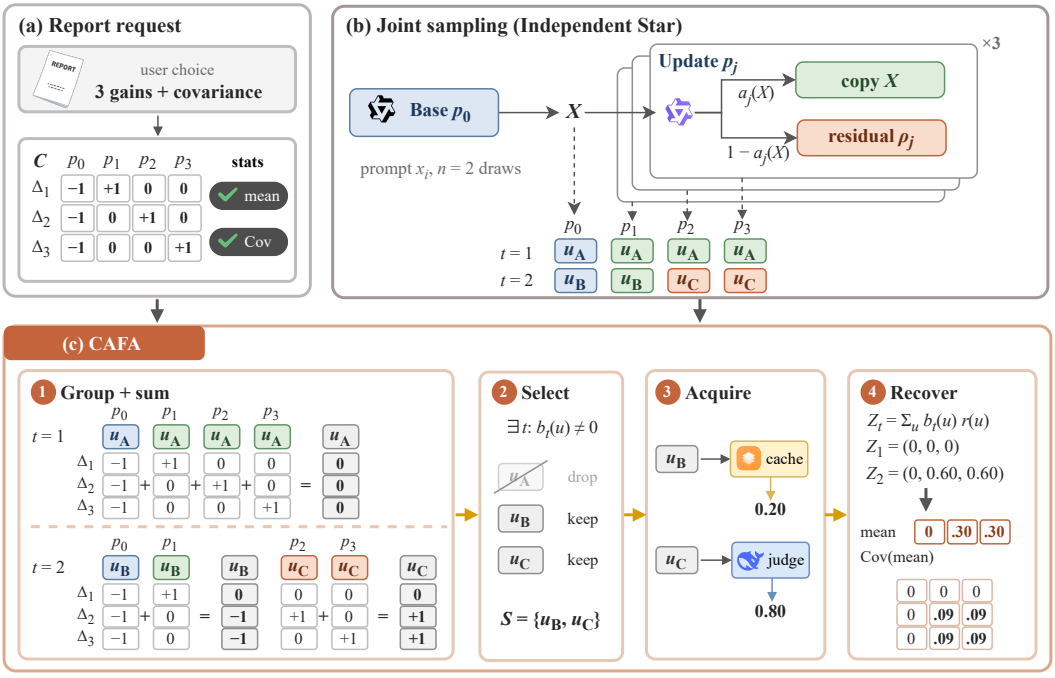


Figure 2: **Report-driven CAFA on a jointly sampled bank.** (a) The report requests three gains over the base and their covariance. (b) Independent Star draws $X \sim p_0$; conditional on X, each update independently copies X or samples $\rho_j \propto (p_j - p_0)_+$, preserving its marginal p_j. Generation costs include all residual proposals, including rejected ones. (c) CAFA groups identical scoring inputs, sums their C columns, selects the required support, acquires uncached scores, and recovers the report. A scoring identity fixes the prompt and response under the same scorer. Both original draws retain weight $1/2$, including the zero-contribution draw. For the illustrative bank and scores, the displayed matrix is the covariance of the reported mean. The support is $\{u_B, u_C\}$, with u_B cached; adding the base absolute mean would require u_A. Deduplication, cancellation and caching reduce eight occurrences to three identities, two required scores and one new query, respectively.

The requested report. A comparison is a linear report

$$\theta = \sum_i w_i\, C\, \mu(x_i), \qquad C \in \mathbb{R}^{d\times(J+1)}. \tag{1}$$

Each row of C defines one requested output; column j contains the coefficients applied to the mean score of model j. The main case is a contrast, $C\mathbf{1} = 0$; for instance $C = [-\mathbf{1}, I_J]$ requests the J gains over the base, and a single row $e_j^\top - e_0^\top$ requests one of them. Appending $e_0^\top$ as an additional row adds the absolute mean of the base model. The resulting anchored report is general linear, with a nonzero row sum. The report also specifies the requested statistics: either the mean alone or the mean together with the ordinary sample covariance of the per-draw contrast vectors. This covariance is used to compute Monte Carlo standard errors (MCSE).

Joint samples and two identities. At prompt x_i an evaluator draws n_i joint samples $Y_{i,t} = (Y_{i,t,0}, \ldots, Y_{i,t,J})$ whose coordinate j has law $p_j(\cdot \mid x_i)$; the coordinates may be dependent. Write $\boldsymbol{r}(Y_{i,t}) = (r(x_i, Y_{i,t,0}), \ldots, r(x_i, Y_{i,t,J}))^\top$. The fully scored estimator is $\widehat{\theta} = \sum_i w_i\, n_i^{-1} \sum_t C\, \boldsymbol{r}(Y_{i,t})$. The scoring identity $\kappa(x_i, y)$ is determined by prompt x_i and the exact decoded text of response y. Responses with the same scoring identity share a score. Probability identity instead uses the original token sequence under the stated policy. Sequences with identical decoded text can share a score while remaining distinct probability events; neither identity uses text normalization, truncation or semantic matching.

What is counted. We distinguish three quantities related to evaluation cost. The required support contains the distinct scoring identities needed for a report, determined from the bank before observing

scores. New paid calls count uncached required scores acquired in an execution. Recorded cost sums generation and feedback costs at the stated tariffs. The goal is to obtain the scores on the required support and recover the same sample statistics as full scoring of the bank.

3 CONTRAST-AWARE FEEDBACK ACQUISITION (CAFA)

Given a report (C, means alone or with covariance) and a joint response bank (Figure 2a,b), CAFA groups identical scoring inputs, sums their coefficients, selects the required support, acquires its uncached scores and recovers the report with the original draw weights (Figure 2c).

3.1 COUPLED GENERATION

CAFA accepts any joint sampler with the specified marginals, including independent sampling. We illustrate coupled generation with ***Independent Star***. For each joint draw, sample a root $X \sim p_0$. For each update j, draw an independent uniform U_j and copy the root, $Y_j = X$, if $U_j \leq \min\{1, p_j(X)/p_0(X)\}$; otherwise draw Y_j from the positive residual $\rho_j \propto (p_j - p_0)_+$. The copied mass $\min(p_0, p_j)$ and residual mass $(p_j - p_0)_+$ sum to p_j, giving the required marginal. Since $\Pr(Y_j \neq X) = \mathrm{TV}(p_0, p_j)$, (X, Y_j) is a maximal coupling of p_0 and p_j at the token-sequence level; simultaneous agreement across all updates is a separate property. Sequence probabilities are evaluated on complete original token sequences under the sampling temperature and stopping rule. Every residual proposal is included in generation cost; a joint draw that exhausts its residual-proposal budget without acceptance is recorded as incomplete. Appendix B.3 specifies acceptance probabilities, random streams and the baseline samplers.

3.2 CANCEL BEFORE SCORING

For each prompt x_i, draw t and scoring identity u, sum the report columns of the policies whose responses have identity u:

$$b_{i,t,u} = \sum_{j:\, \kappa(x_i, Y_{i,t,j}) = u} C_{:j}, \qquad A_{i,u} = \sum_{t=1}^{n_i} b_{i,t,u}. \tag{2}$$

Here $b_{i,t,u}$ is the net coefficient vector for identity u within one draw, and $A_{i,u}$ is its sum over all draws for prompt x_i. Only columns with a non-zero entry in C are active, so a single-gain report excludes responses from the other updates. Then

$$\bar{Z}_i = \frac{1}{n_i} \sum_u A_{i,u}\, r(u), \qquad \widehat{\theta} = \sum_i w_i \bar{Z}_i, \tag{3}$$

which is the fully scored estimator with terms grouped by scoring identity. Coefficients are aggregated exactly before division by n_i, and a coefficient vector is treated as zero only when all d components are zero. The identities that remain are

$$S_{\mathrm{mean}} = \{(i,u) : A_{i,u} \neq 0\} \;\subseteq\; S_{\mathrm{draw}} = \{(i,u) : \exists t,\ b_{i,t,u} \neq 0\} \;\subseteq\; S_{\mathrm{cache}},$$

where S_{cache} collects every identity observed on an active column. We write $Q_{\mathrm{cache}} = |S_{\mathrm{cache}}|$, $Q_{\mathrm{mean}} = |S_{\mathrm{mean}}|$ and $Q_{\mathrm{mean+cov}} = |S_{\mathrm{draw}}|$ for the corresponding numbers of required scoring inputs. The operation removes score queries, never draws: a draw in which all policies agree still counts towards n_i.

3.3 RECOVER THE REQUESTED REPORT

Recovering the mean requires the scores on S_{mean}. Reporting uncertainty may require additional scores: with $Z_{i,t} = \sum_u b_{i,t,u} r(u)$, the ordinary sample covariance $\widehat{\Sigma}_i = (n_i - 1)^{-1} \sum_t (Z_{i,t} - \bar{Z}_i)(Z_{i,t} - \bar{Z}_i)^\top$ and the Monte Carlo covariance of the weighted mean, $\widehat{V} = \sum_i w_i^2 \widehat{\Sigma}_i / n_i$, are recovered from the per-draw vectors determined by the scores on S_{draw}. For the fixed-prompt MCSE, joint draws and generation streams across distinct prompts are independent under the sampling protocol; the algebraic recovery identities themselves do not require independence. For example, two mirror draws for the same prompt return (A, B) and (B, A), with the base listed first. Their

differences, $r(B) - r(A)$ and $r(A) - r(B)$, cancel, so the sample mean is zero without either score. The ordinary sample variance is $2[r(B) - r(A)]^2$, however, and still requires both scores for unrestricted unknown $r(A)$ and $r(B)$ (Appendix Figure 6).

Appendix B.5 gives complete pseudocode and the windowed variant. Multi-output reports are planned jointly; changed reports or budgets are re-planned with cached scores. Missing-score handling and separate mean/covariance support counts are specified in Appendix B.5.

3.4 Implementation

Our library supports existing banks, local Independent Star generation, versioned caches and external scorers. It plans support before reading scores; the scorer sees only the prompt and response. Appendix I documents export–import, returned statistics and query logs, with runnable examples and tests.

4 Exact recovery and required feedback

The guarantees condition on a fixed bank's responses, scoring identities and report; the scorer is an otherwise unrestricted oracle.

4.1 Exact recovery and required mean feedback

Proposition 1 (Pointwise recovery and necessary mean queries)**.** *For every bank and every scorer, the estimate* (3) *computed from S_{mean} equals the fully scored estimate $\widehat{\theta}$. Conversely, suppose each unknown score can take any value in a nondegenerate interval, independently of the others. Then any procedure that returns $\widehat{\theta}$ exactly for every such scorer must query every identity in S_{mean} whose score it does not already hold.*

Sufficiency follows by collecting terms with the same scoring identity. For necessity, suppose an initially unknown score at $(i, u) \in S_{\mathrm{mean}}$ remains unqueried; two scorers that agree on every queried input and differ at u produce the same transcript but reports differing by $w_i n_i^{-1} A_{i,u}\,(r(u) - r'(u)) \neq 0$ (Appendix A.1).

4.2 Joint mean and covariance reporting

Proposition 2 (Required support for the mean with ordinary covariance)**.** *Condition on the bank, report and identities, and let $n_i \geq 2$. Suppose each unknown score can vary independently over a nondegenerate interval. The scores on S_{draw} suffice for exact recovery of $\bar{Z}_i$ and $\widehat{\Sigma}_i$; every score not already held in this support must be queried to guarantee recovery for every such scorer. The same holds for $\widehat{\theta}$ with the stratified covariance $\widehat{V}$ over distinct prompt identities with positive weights.*

Sufficiency: the scores on S_{draw} determine every $Z_{i,t}$. Necessity: if an initially unknown, unqueried u has $A_{i,u} \neq 0$, the mean changes with $r(u)$. Otherwise some output k has $\sum_t b_{i,t,u,k}^2 > 0$ while $\sum_t b_{i,t,u,k} = 0$, so $\widehat{\Sigma}_{i,kk}$ is a quadratic in $r(u)$ with strictly positive leading coefficient $\sum_t b_{i,t,u,k}^2/(n_i - 1)$ and cannot be constant on an interval; with positive weights the same holds for $\widehat{V}_{kk}$ (Appendix A.2). Hence $|S_{\mathrm{draw}} \setminus S_{\mathrm{mean}}|$ counts the additional scoring identities required for joint mean-and-covariance reporting. The result does not characterize the minimum number of queries needed to recover covariance alone, which may require fewer scores when an unknown contribution is constant across draws.

4.3 Operational consequences

Write $S(C)$ for the support of report C on a fixed bank.

Corollary 1 (Changing coordinates versus changing the report)**.** *If H is invertible, $S(HC) = S(C)$. Stacking reports takes the union, $S\big([C; D]\big) = S(C) \cup S(D)$. In particular, reporting all gains over the base together with its absolute mean requires every observed identity: $S = S_{\mathrm{cache}}$.*

An occurrence vector counts the appearances of a scoring identity under each policy. Invertible maps preserve zero vectors, and a stacked vector vanishes only if both blocks do; $[-\mathbf{1}, I_J]$ stacked with $e_0^\top$ is invertible, so no nonzero occurrence vector is mapped to zero. Both claims hold for S_{mean} and separately within each draw for S_{draw}. Changing the outputs requires recomputing their joint support and missing-score cost, which need not equal d times single-output cost because supports can overlap.

Under Independent Star with base-relative rows, a token sequence z receives a negative coefficient for update j only when it appears as a rejected root ($p_j(z) < p_0(z)$), and a positive coefficient only when it appears as a residual draw ($p_j(z) > p_0(z)$). Its nonzero per-draw coefficients cannot cancel across draws. With identities defined by original token sequences, $S_{\mathrm{mean}} = S_{\mathrm{draw}}$: immediate CAFA attains the batch required-input count, and the same scores recover the covariance (Proposition 3 in Appendix A.4). For decoded-text identities, this conclusion requires $p_j - p_0$ to have a consistent sign within each decoded group; the appendix gives a counterexample when this condition fails. Other samplers can require the covariance increment, as the experiments show.

5 Experiments

5.1 Setup

To assess CAFA's practical benefits and the factors that determine them, we ask three questions: (RQ1) How much feedback does CAFA remove beyond caching? (RQ2) How does coupled generation with CAFA affect cost and precision? (RQ3) How do report choice and checkpoint position affect savings? We compare controlled updates and online GRPO (Table 2, Appendix B). Controlled updates use three fixed endpoints per model, one per training seed, for Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct. Online GRPO uses three Qwen2.5-1.5B-Instruct trajectories, comparing the base with steps 10, 50 and 250. Offline checkpoints saved at steps 2, 8 and 32 are evaluated in Section 5.4. Evaluation prompts come from No Robots, WildChat and ELI5 (Rajani et al., 2023; Zhao et al., 2024; Fan et al., 2019). Policies sample at temperature 1 with a 768-token cap; sequence probabilities are evaluated on the original token IDs. Judges return scalar scores from 1 to 10, reused from version-pinned caches: DeepSeek Flash for online GRPO evaluation, and DeepSeek Flash and Qwen Flash separately for controlled updates. Judge settings and sensitivity are in Appendices B.4 and F.1.

The controlled strict holdout has 32 prompts and 64 joint draws per prompt; Figure 3 uses two repetitions per sampler, with all four Independent Star repetitions in Appendix C. GRPO uses 128 held-out prompts, split equally for selection and testing. Independent AR, shared-token AR, Independent Star and TV tree are evaluated at 4, 8 and 16 draws per prompt, across three seeds and two generation repetitions: 72 banks and 504 cells over seven reports. Precision is fixed-prompt MCSE from $\widehat{V}$, summarized across gains by RMS MCSE. Costs in USD price generation from uninterrupted runtime measurements and feedback from measured token usage at frozen tariffs (Appendix G). Fixed-bank comparisons hold the response bank, score values and full recorded generation cost fixed to compare Cache-only with CAFA; workflow comparisons also change the sampler, relative to Independent AR + Cache-only.

5.2 Controlled updates: isolating feedback reduction

CAFA removes substantial feedback beyond caching while preserving the sampled reports exactly. On the first two fixed Independent Star banks, it eliminates 55.8% of the scoring inputs required after deduplication for Qwen and 90.6% for SmolLM2 (Figure 3a). The same rule removes 63.5% and 91.1% on TV tree banks, but 4.4% and 39.7% on shared-token AR banks, respectively, and essentially none on independent AR banks. Across samplers, savings depend on the pattern of identical responses in each bank and the requested report. Mean and mean-plus-covariance supports coincide in all four Independent Star repetitions. For Qwen with independent AR, the mean-only support averages 8132.5 scoring inputs, compared with 8133 for the mean-with-covariance report (Table 5, Appendix C).

Avoiding score acquisitions also lowers total recorded cost. With the complete joint-star generation cost held fixed, Figure 3b compares Cache-only and CAFA total costs, yielding 7.5–23.2% lower

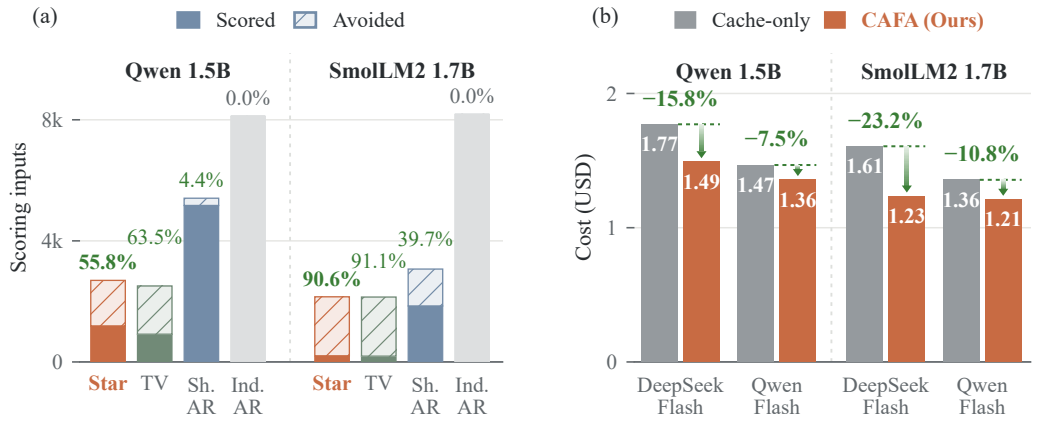


Figure 3: **CAFA reduces scoring inputs and same-bank cost.** Controlled strict holdout: 32 prompts, three updates and 64 joint draws per prompt; means over the first two repetitions for every sampler. (a) Full bars count deduplicated inputs; solid portions are required for means and covariance, and hatched portions are avoided. (b) Independent Star generation-plus-feedback cost in USD, comparing Cache-only and CAFA on the same banks with the full generation charge fixed. Sh. AR, Ind. AR and TV denote shared-token AR, independent AR and TV tree. All four Independent Star repetitions appear in Appendix Table 5.

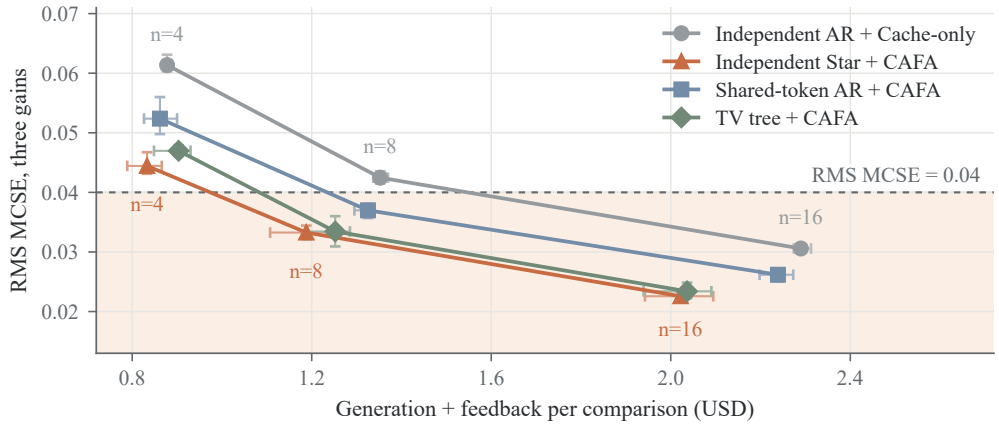


Figure 4: **Online GRPO cost–precision tradeoffs.** Points average total cost and cell-level RMS MCSE over three training seeds × two generation repetitions; whiskers span the six-cell minima and maxima. Budgets $n = 4, 8, 16$ increase from left to right along each curve. Cost includes generation and feedback per 128-prompt, three-gain comparison with covariance, including the TV-tree pilot. The dashed line marks RMS MCSE $= 0.04$. All 72 banks appear individually in Appendix Figure 10.

generation-plus-feedback cost across the two models and two judges on the first two repetitions. Savings increase with the feedback share.

5.3 Online GRPO: practical model comparison

Independent test references, using 64 autoregressive draws per policy and test prompt, estimate final step-250 gains of 0.317–0.350 judge-score points over the base across the three training seeds (Appendix Table 12). **The coupled workflow reduces recorded cost while improving precision.** Across all 18 equal-budget comparisons, Independent Star + CAFA has lower recorded cost and lower RMS MCSE than Independent AR + Cache-only (Figure 4). At 16 draws per prompt (Table 1), it uses 31.1–36.9% fewer scoring inputs, costs 9.4–14.6% less and has 0.716–0.772 times the baseline RMS MCSE. TV tree and shared-token AR lower RMS MCSE in all 18 comparisons but exceed baseline cost in five and two four-draw cells, respectively; TV-tree cost includes its pilot stage. At 16 draws

Table 1: **Online GRPO workflows at 16 draws per prompt**, reporting three gains and their covariance. The "Scored" column reports input counts: Q_{cache} for Cache-only and $Q_{\text{mean+cov}} = |S_{\text{draw}}|$ for CAFA. Counts and generation-plus-feedback costs per 128-prompt comparison are means over three training seeds $\times$ two generation repetitions. Reduction and RMS MCSE-ratio ranges are the six-cell minima and maxima relative to Independent AR + Cache-only in each cell.

		Recorded workload ↓		Reduction ↑		Precision ↓
Sampler	Scoring strategy	Scored	Cost (USD)	Inputs (%)	Cost (%)	MCSE ratio
Independent AR	Cache-only	8,170	2.29	–	–	1
Shared-token AR	CAFA (Ours)	7,853	2.24	2.2–6.5	1.2–3.3	0.826–0.875
Independent Star	CAFA (Ours)	5,424	**2.02**	31.1–36.9	9.4–14.6	0.716–0.772
TV tree	CAFA (Ours)	**4,950**	2.04	36.7–42.5	8.5–14.7	0.733–0.823

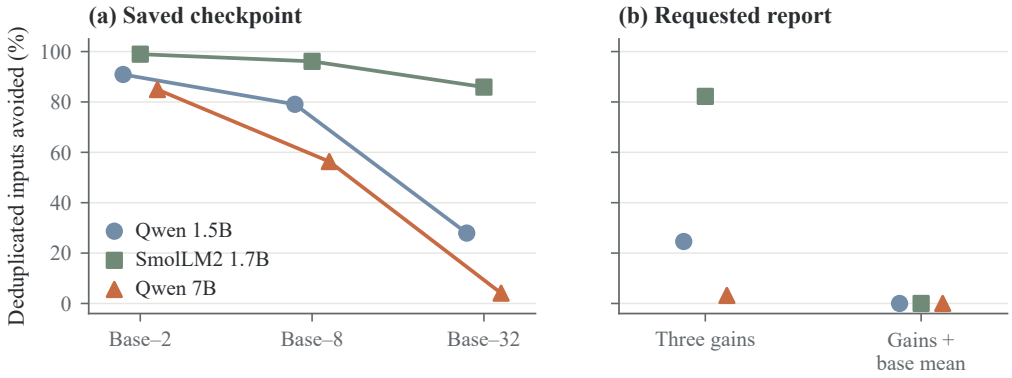


Figure 5: **Feedback savings vary with checkpoints and requested outputs.** Independent Star + CAFA on offline checkpoints: 32 prompts, 64 draws per prompt and a 768-token cap. The vertical axis is the fraction of deduplicated inputs avoided. Left: individual base-relative gains at steps 2, 8 and 32. Right: all three gains, with and without the base mean. Points average three trajectories per small model and one for Qwen 7B.

per prompt, Independent Star's reduction in feedback cost exceeds its increase in recorded generation cost in all six cells; the per-cell breakdown and CAFA bookkeeping timings are in Appendix Table 10 and Appendix G.4.

The precision advantage lowers the budget at a common empirical threshold. On the evaluated grid, RMS MCSE ≤ 0.04 requires eight draws per prompt for Independent Star + CAFA versus sixteen for Independent AR + Cache-only, reducing recorded generation-plus-feedback cost by 46.1–51.3% across all six seed–repetition cells (Figure 1). Further analyses of known-truth error and interval coverage appear in Appendices F.2–F.3.

CAFA adds a report-dependent cost reduction. On fixed Independent Star banks with full four-policy generation cost retained, CAFA lowers recorded cost by 20.9–27.0% for the step-10 gain over the base and 1.54–2.07% for the joint three-gain report (Appendix E.1, Table 13).

CAFA preserves all 504 sampled reports exactly: their means and Monte Carlo covariances match full scoring; an independent recomputation confirms one cached-score access per required identity (Appendix D.5). Mean-only and mean-plus-covariance supports coincide in 485 cells, while 19 autoregressive-sampler cells need additional scores for covariance (Proposition 2). All 72 fixed checkpoint selections match full scoring of the same banks; 71 of 72 also match the independent 64-draw test-reference best checkpoint; the remaining selection has estimated regret 0.151 (MCSE 0.020; Appendix D.4).

5.4 What determines the feedback requirement?

The requested output. Appendix E.1, Table 13 compares seven reports on six fixed Independent Star banks. **Joint reports require the union of constituent supports (Corollary 1).** Scores can cancel in one output but be needed for another. Gains plus the base mean require full Cache-only support; selection uses gains alone because subtracting a common base mean preserves the maximizer.

Which checkpoints. Figure 5 compares feedback reduction across offline checkpoints (768-token cap). On the single Qwen2.5-7B trajectory, reductions in scoring inputs are 84.9%, 56.3% and 4.1% at steps 2, 8 and 32, respectively, and 3.2% for the joint report. **Earlier-checkpoint reports save more feedback in this trajectory.** Steps label checkpoints rather than measured distribution distances; output-cap and temperature effects are in Appendices E.2–E.3.

Prices and execution. For fixed feedback reduction, total cost savings increase with the feedback cost share. At one quarter of the frozen feedback tariff, shared-token AR is the cheapest workflow at 16 draws per prompt in all six GRPO cells (Appendix G.3). Appendix H evaluates shared acceptance, residual grouping and budgeted scoring; residual grouping reduces joint support by 2.1–3.9% while raising recorded cost by 1.6–5.1%.

6 Related work

Coupling and shared randomness. Couplings construct joint laws with prescribed marginals (Levin & Peres, 2017; Thorisson, 2000); common random numbers exploit dependence to sharpen simulated comparisons (Glasserman & Yao, 1992; Glasserman, 2003), and related ideas underlie unbiased MCMC (Jacob et al., 2020), Gumbel-based sampling (Maddison et al., 2014; Kool et al., 2019) and coupled rejection samplers (Corenflos & Särkkä, 2022). The Independent Star is the classical maximal coupling applied to each base–update pair. CAFA determines which scores their response banks require for a report.

Coupled evaluation and feedback sharing. Benz et al. (2026) use shared randomness to reach the same language-model benchmark conclusions with provably fewer samples, although pairwise-comparison rankings can change. Wen (2026) couples policies on a tree, sharing reward queries for matched components. Beyond caching identical responses, CAFA identifies their contribution to the report before scoring. All-matched draws need no scores for zero-sum gains; single-gain reports exclude other updates; covariance can require scores that cancel in the mean. The rule accepts all these samplers, including independent sampling, where score reductions are typically small.

Evaluating with fewer labels. Sawade et al. (2012) compare predictive models directly under a labeling budget. Active testing and active surrogate estimation select informative labels (Kossen et al., 2021; 2022), while subset selection and anchor points reduce the evaluation set (Maia Polo et al., 2024; Vivek et al., 2024). These methods balance labeling cost against estimation error. CAFA removes only scores irrelevant to the reported statistic and composes with budgeted selection (Appendix H.4). We assume a scalar, versioned scorer; pairwise judges (Zheng et al., 2023; Chiang et al., 2024) define a different estimand whose value can depend on the coupling itself (Appendix A.5).

7 Conclusion

CAFA plans required scores before acquisition. Under an unrestricted-score oracle, we characterize exact support for means alone or jointly with ordinary sample covariances. Controlled-update experiments isolate acquisition reductions beyond caching; online GRPO experiments with coupling lower recorded cost at a common empirical precision threshold. All evaluated reports and fixed selections match full scoring of the same banks. Gains plus the base mean require full Cache-only support; focused comparisons can retain savings. The modular implementation integrates CAFA into existing comparison workflows. Feedback savings depend on response overlap and the report; net cost savings also depend on generation overhead and scorer prices. Our evidence covers the evaluated model families, checkpoint regimes and scalar judges.

Reproducibility statement

The assumptions and complete proofs of Propositions 1–2 and Corollary 1 are in Appendix A. Models, prompts, the three checkpoint regimes, decoding and stopping settings, random-stream derivation, samplers, scoring identities and judge settings are in Appendix B; the online GRPO protocol and bank inventory are in Table 3. Every cell behind the main-text results remains available: Appendices C–D visualize or tabulate them, and the complete per-cell records for the parts that are summarized there are in the canonical result tables of the supplementary material, which Appendix I.4 maps number by number. Cost conventions are in Appendix G. The supplementary material contains the canonical result tables, the offline scripts for all empirical figures and generated tables, editable method diagrams, complete chart CSV files, the audit of every main-text number, and an anonymous code archive with the acquisition method, its tests and its local-generation adapters (Appendix I). Regenerating the reported support, precision and cost values requires no GPU and no judge call; regenerating the banks themselves requires the recorded model checkpoints.

AI use statement

Generative AI tools assisted with experiment and analysis code, software implementation and debugging, figure and table generation from frozen results, numerical verification, and language polishing. Responses and scores were produced by the model and judge calls specified in the experimental protocol (Section 5.1); the supplementary scripts compute the reported values from those frozen records. The authors take responsibility for the reported results and released artifacts.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, et al. SmolLM2: When Smol Goes Big — Data-Centric Training of a Fully Open Small Language Model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=3JiCl2A14H.

Nina Corvelo Benz, Stratis Tsirtsis, Eleni Straitouri, Ivi Chatzi, Ander Artola Velasco, Suhas Thejaswi, et al. Evaluation of Large Language Models via Coupled Token Generation. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 2215–2223. PMLR, 2026. URL https://proceedings.mlr.press/v300/benz26a.html.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, et al. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chiang24b.html.

C. J. Clopper and E. S. Pearson. The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404. URL https://doi.org/10.1093/biomet/26.4.404.

Adrien Corenflos and Simo Särkkä. The Coupled Rejection Sampler. *arXiv preprint arXiv:2201.09585*, 2022. URL https://arxiv.org/abs/2201.09585.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1383–1392. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128/.

Yann Dubois, Percy Liang, and Tatsunori B. Hashimoto. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer New York, 2003. doi: 10.1007/978-0-387-21617-1. URL https://doi.org/10.1007/978-0-387-21617-1.

Paul Glasserman and David D. Yao. Some Guidelines and Guarantees for Common Random Numbers. *Management Science*, 38(6):884–908, 1992. doi: 10.1287/mnsc.38.6.884. URL https://doi.org/10.1287/mnsc.38.6.884.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-aos1991. URL https://doi.org/10.1214/20-aos1991.

John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/mcse.2007.55. URL https://doi.org/10.1109/mcse.2007.55.

Pierre E. Jacob, John O'Leary, and Yves F. Atchadé. Unbiased Markov Chain Monte Carlo Methods with Couplings. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(3):543–600, 2020. doi: 10.1111/rssb.12336. URL https://doi.org/10.1111/rssb.12336.

Seungone Kim, Jay Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, et al. Prometheus: Inducing Fine-Grained Evaluation Capability in Language Models. In *The Twelfth International Conference on Learning Representations*, volume 2024, pp. 29927–29962, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/803485352e61e3ebf41221e4776c9fd4-Abstract-Conference.html.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, et al. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3499–3508. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active Testing: Sample-Efficient Model Evaluation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5753–5763. PMLR, 2021. URL https://proceedings.mlr.press/v139/kossen21a.html.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Thomas Rainforth. Active Surrogate Estimators: An Active Learning Approach to Label-Efficient Model Evaluation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24557–24570. Curran Associates, Inc., 2022. doi:

10.52202/068431-1783. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9b9cfd5428153ccfbd4ba34b7e007305-Paper-Conference.pdf.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, et al. RewardBench: Evaluating Reward Models for Language Modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, second edition, 2017. URL https://darkwing.uoregon.edu/~dlevin/MARKOV/. With contributions by Elizabeth L. Wilmer.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, et al. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification, Outstanding Certification.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* Sampling. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/937debc749f041eb5700df7211ac795c-Paper.pdf.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Anuran Makur and Japneet Singh. Doeblin Coefficients and Related Measures. *IEEE Transactions on Information Theory*, 70(7):4667–4692, 2024. doi: 10.1109/tit.2024.3367856. URL https://doi.org/10.1109/tit.2024.3367856.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein Bounds and Sample Variance Penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009. URL https://www.cs.mcgill.ca/~colt2009/papers/012.pdf.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Qwen Team. Qwen2.5 Technical Report. arXiv:2412.15115, 2024. URL https://arxiv.org/abs/2412.15115v2. Version v2.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model.

In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No Robots. Dataset card, 2023. URL https://huggingface.co/datasets/HuggingFaceH4/no_robots. Human-written instruction and demonstration dataset; accessed September 12, 2026.

Aadirupa Saha, Aniket Wagde, and Branislav Kveton. LLM-as-a-Judge on a Budget. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4375–4383. PMLR, 2026. URL https://proceedings.mlr.press/v300/saha26a.html.

Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active Comparison of Prediction Models. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/92fb0c6d1758261f10d052e6e2c1123c-Paper.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347v2. Version v2.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300v3. Version v3.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=uyTL5Bvosj. Featured Certification.

Hermann Thorisson. *Coupling, Stationarity, and Regeneration*. Springer New York, 2000. doi: 10.1007/978-1-4612-1236-2. URL https://hermann.hi.is/iid/csr/.

Pat Verga, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, et al. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. arXiv:2404.18796, 2024. URL https://arxiv.org/abs/2404.18796v2. Version v2.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor Points: Benchmarking Models with Much Fewer Examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-long.95. URL https://aclanthology.org/2024.eacl-long.95/.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-Preference Bias in LLM-as-a-Judge. arXiv:2410.21819, 2024. URL https://arxiv.org/abs/2410.21819v2. Version v2.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024. doi: 10.1093/jrsssb/qkad009. URL https://doi.org/10.1093/jrsssb/qkad009.

Yuxiao Wen. Fast A/B/n Testing: Exact Multi-Policy Comparison via Tree-Coupled Feedback Sharing. *arXiv preprint arXiv:2608.12831*, 2026. URL https://arxiv.org/abs/2608.12831v2. Version v2, revised August 14, 2026; preprint.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, et al. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *International Conference on Learning Representations*, volume 2024, pp. 34590–34605, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9421261e06f1a63a352b068f1ac90609-Paper-Conference.pdf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned Large Language Models are Scalable Judges. In *International Conference on Learning Representations*, volume 2025, pp. 51257–51296, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7f8f73134e253845a8f82983219a8452-Paper-Conference.pdf.

APPENDIX OVERVIEW

This appendix contains the complete proofs, experimental protocols, results for each configuration, statistical diagnostics, cost analyses and software documentation. The table below lists where to find each topic. The naming convention of Section 1 applies throughout: Independent Star, Independent AR, Shared-token AR and TV tree are samplers; CAFA (Ours) is the post-sampling score-acquisition module and Cache-only is its scoring-strategy control. All sampler configurations using CAFA use our module. Fixed-bank CAFA comparisons and cross-sampler workflow comparisons have different interpretations.

Question	Where
Why is the acquired support sufficient, and why can none of it be dropped?	A
Which models, prompts, checkpoints, samplers and scoring settings were used?	B
What happened in every controlled-update bank, sampler and budget?	C
What happened in every online-GRPO bank, target and budget?	D
How do the requested output, checkpoint position, length cap and temperature matter?	E
How sensitive are results to the judge, and what do the intervals mean?	F
How were generation and feedback costs measured and priced?	G
What do shared acceptance, residual grouping and budgeted scoring add?	H
How is the software installed, integrated and tested?	I

Notation and source-record aliases. Prompts are x_i, with prompt index i and policies p_0, p_j. Single-prompt derivations suppress i: $x = x_i$, $n = n_i$, $Y_{tj} = Y_{i,t,j}$ and $B_t[y] = b_{i,t,u}$; their coefficient map $A[y]$ is $A_{i,u}$. The reused blocks write y for a scoring identity (the main text writes u), ϕ or "decoded text" for the identity map κ, a_y or $\bar{b}_y$ for the per-prompt coefficient $A_{i,u}/n_i$. Throughout, S denotes a support set and Q its cardinality: $Q_{\text{cache}} = |S_{\text{cache}}|$, $Q_{\text{mean}} = |S_{\text{mean}}|$ and $Q_{\text{mean+cov}} = |S_{\text{draw}}|$. Feedback fees use the same subscripts, F_{cache}, F_{mean} and $F_{\text{mean+cov}}$; total workflow cost is $T = G + F$. "Cancellation" denotes the coefficient-cancellation operation within CAFA; "star" means the Independent Star unless "shared" is stated; "Token" or "shared tokens" is shared-token (Gumbel) autoregressive sampling; "Tree" is the total-variation spanning tree. Three checkpoint regimes are never pooled (Table 2). These aliases affect display only: the original CSV keys, file paths and program arguments keep their recorded names, and Appendix I.4 holds the mapping.

A EXACT RECOVERY AND REQUIRED FEEDBACK

A.1 POINTWISE RECOVERY AND MEAN-QUERY NECESSITY

Proof of Proposition 1. Fix prompt x_i. By definition of $b_{i,t,u}$, $\sum_t C\, \boldsymbol{r}(Y_{i,t}) = \sum_t \sum_j C_{:j}\, r(\kappa(x_i, Y_{i,t,j})) = \sum_u \big(\sum_t b_{i,t,u}\big) r(u) = \sum_u A_{i,u} r(u)$, and terms with $A_{i,u} = 0$ vanish for every scorer. Dividing by n_i and weighting by w_i gives the fully scored estimator. For necessity, let an algorithm (possibly adaptive) return the report exactly for every admissible scorer while never querying some $(i, u) \in S_{\text{mean}}$ whose score it does not hold. Take two scorers equal on every queried input and differing at u by $\delta \neq 0$; both are admissible because unknown coordinates vary independently over a nondegenerate interval. The algorithm sees the same transcript under both, but the reports differ by $w_i n_i^{-1} A_{i,u} \delta \neq 0$, a contradiction. Scores already held need not be purchased, which is why the necessary support and the new paid calls are separate quantities. The argument assumes no restriction linking the scores of different inputs; under such a restriction fewer queries could suffice.

A.2 MEAN AND ORDINARY COVARIANCE

This subsection gives the proof of Proposition 2 with its oracle assumptions, the single-prompt argument and the stratified extension used for the main-experiment MCSEs. The argument requires distinct prompt identities. If the same full prompt appears in two nominal strata, those identities must first be merged. The software validates this condition when loading a bank (Appendix I.2).

A.2.1 REPORTING SUPPORT: PROOF, ASSUMPTIONS, AND CHECKS

Oracle and identity assumptions. The bank, target coefficients, stopping policy and scoring identity are fixed before score acquisition. Each unqueried scoring coordinate may independently take any value in a nondegenerate interval; no known functional relationships link distinct coordinates. The lower bound also applies after some coordinates have been observed, because the argument holds other scores fixed. An adaptive exact-query algorithm that omits a required coordinate has the same transcript under two admissible completions with different outputs. The ten-point score cache used in the experiment is fixed. The same argument applies to an unrestricted score alphabet with at least three distinct values, because a nonconstant quadratic cannot take the same value at three distinct points.

Single prompt. Set $\bar{b}_y = n^{-1}\sum_t b_{t,y}$. If $\bar{b}_y \neq 0$, $\bar{Z}$ has a nonzero linear dependence on $r(y)$. If $\bar{b}_y = 0$ but some $b_{t,y} \neq 0$, choose a coordinate k with a nonzero coefficient. Holding other scores fixed, write $Z_{t,k} = b_{t,y,k}s + v_{t,k}$. Then

$$\widehat{\Sigma}_{kk}(s) = \frac{1}{n-1}\sum_t \big((b_{t,y,k} - \bar{b}_{y,k})s + v_{t,k} - \bar{v}_k\big)^2.$$

Its squared-term coefficient is positive. Hence at least one of the required outputs depends on every coordinate in S_{draw}. Outside that support every draw coefficient is zero, so no required output depends on the score. This proves the exact support characterization.

The stratified MCSE used in the paper. Let prompt x_i have $n_i \geq 2$ draws, positive fixed weight w_i, sample mean $\bar{Z}_i$ and ordinary covariance $\widehat{\Sigma}_i$. The reported mean and Monte Carlo covariance estimate are

$$\widehat{\theta} = \sum_i w_i \bar{Z}_i, \qquad \widehat{V} = \sum_i \frac{w_i^2}{n_i}\widehat{\Sigma}_i.$$

Here $w_i = 1/P$ and $n_i = n$. The scoring key contains the exact prompt, so a score identity belongs to a single distinct prompt group. Its nonzero average coefficient changes $\widehat{\theta}$; if that coefficient is zero, a nonzero draw coefficient gives a strictly positive squared-term coefficient in a diagonal of $\widehat{V}$. The positive factor w_i^2/n_i cannot remove this dependence. Thus the union of the prompt-wise draw supports is necessary and sufficient. If identical prompt keys are reused across nominal strata, their identities and coefficients must instead be collected together; the distinct-prompt argument must not be applied without this check. These algebraic conclusions do not require draws to be independent, although interpreting $\widehat{V}$ as a Monte Carlo covariance estimator does require the sampling assumptions stated elsewhere.

A.3 REPORTING MORE OUTPUTS OR CHANGING COORDINATES

Proof of Corollary 1. For a fixed identity write h_u for its occurrence vector over policies, so that its aggregated coefficient is Ch_u (and per draw $Ch_{t,u}$). If H is invertible, $HCh_u = 0$ if and only if $Ch_u = 0$, so $S(HC) = S(C)$ for S_{mean} and, draw by draw, for S_{draw}; outputs transform as $H\widehat{\theta}$ and covariances as $H\widehat{V}H^\top$. A stacked coefficient $[C; D]h_u$ vanishes exactly when both Ch_u and Dh_u vanish, giving the union. For $C = [-\mathbf{1}, I_J]$ stacked with $e_0^\top$ the square matrix is invertible, and every observed identity has $h_u \neq 0$; hence all observed identities are required and $S = S_{\mathrm{cache}}$. The zero-sum gains alone suffice for checkpoint selection because subtracting the common base mean does not change the maximizer; Appendix D.4 reports the selection audit for the fixed online-study selectors. Changing the target, for example from base-relative to adjacent-checkpoint gains, is not a change of coordinates of the same report unless the map is invertible on the requested outputs; its support is recomputed (Appendix E.1).

A.4 IMMEDIATE AND WINDOWED CAFA

This subsection gives the full statement and proof of sign purity (Proposition 3), its decoded-text condition, and the property that windowed reporting with a cumulative coefficient map buys a superset of the batch support and never refunds a score. The measured window and support audits are in Appendix C.3.

A.4.1 IMMEDIATE ACQUISITION AND THE REQUESTED STATISTIC

Using the coefficients from Section 4.2, define

$$S_{\mathrm{draw}} = \{y : \exists t,\ b_{t,y} \neq 0\}, \qquad S_{\mathrm{mean}} = \{y : \textstyle\sum_t b_{t,y} \neq 0\}.$$

Cancelling within a draw and caching acquires S_{draw}; final-batch aggregation acquires $S_{\mathrm{mean}} \subseteq S_{\mathrm{draw}}$. Their difference is the precise additional opportunity offered by waiting. The simple pairwise implementation is therefore an exact implementation of the mean rule whenever these sets agree.

Proposition 3 (Sign purity at the sampled support)**.** *For a base-relative contrast under the splitting construction, fix a sampled sequence z. Across draws its nonzero coefficients all have the same sign. Thus at sequence-level identity, $S_{\mathrm{mean}} = S_{\mathrm{draw}}$. This conclusion also holds for decoded-text groups if, for each contrast row, $p_j(z) - p_0(z)$ does not change sign among sequences in any one decoded group.*

A positive term at z requires a residual endpoint, where $p_j(z) > p_0(z)$; a negative term requires a rejected root, where $p_j(z) < p_0(z)$. These supports are disjoint. Appendix C.3.2 gives the decoded-text counterexample and the support audits. This result explains when immediate reporting already attains the final-batch CAFA query count.

Once S_{draw} is acquired, every draw vector, its sample covariance and any fixed resampling statistic are available. The strict inclusion can occur in the mirror-draw example of Figure 6; the requested statistics therefore remain an explicit input to CAFA.

A.4.2 WINDOWED ACQUISITION

Let A_w be the running coefficient map after window w and $S_{\mathrm{acq}}^{\mathrm{mean}} = \bigcup_w \{y : A_w[y] \neq 0\}$ the inputs accessed during the mean stage. Since $A_{n/W} = A$ is the full-batch map, $S_{\mathrm{mean}} \subseteq S_{\mathrm{acq}}^{\mathrm{mean}} \subseteq S_{\mathrm{draw}}$, with the first inclusion an equality at $W = n$. Windowing returns the fully scored mean at every window boundary. Earlier accesses remain in $S_{\mathrm{acq}}^{\mathrm{mean}}$ when later draws cancel their coefficients, so its size depends on the bank and W (Appendix C.3). For the final joint mean-and-covariance report, Algorithm 2 completes the accesses to $S_{\mathrm{acq}} = S_{\mathrm{acq}}^{\mathrm{mean}} \cup S_{\mathrm{draw}} = S_{\mathrm{draw}}$ for every W. Cache hits and newly paid queries are recorded separately. Interval procedures are specified in Appendix F.3.

A.5 FINITE VERIFICATION AND INTERFACE EXAMPLES

This subsection supplies the boundary examples (all-copy draws, mirror draws, a repeated pair, decoded-text merging and the absolute anchor), the finite enumeration checks of the source study, and a pairwise-preference counterexample showing why a pairwise judge changes the estimand. The enumeration counts are given below. These checks were run in the source study, and finite enumeration supplements rather than replaces the proofs of Appendices A.1–A.4. The released software has its own independent test suite, including the twelve acceptance fixtures and a dense oracle that does not share code with the coefficient path (Appendix I.3).

Boundary examples. All-copy draws have zero coefficients for zero-sum targets and require no feedback. The mirror draws of Figure 6 are the named boundary case in which mean-only acquisition is insufficient. Repeating (A, B) twice has zero sample variance for all scores, but its mean still requires both scores: the proposition must request mean and covariance jointly. Decoded-text merging can create the first type of cancellation from distinct sampled sequences; the score key may merge them while their probability identities remain distinct. Appending an absolute anchor requires the full cached support even on all-copy banks.

CPU checks and empirical audit. The accompanying test enumerates all 729 three-draw, two-policy banks over three response identities, adds 700 vector-target cases, five named boundary cases and 200 stratified coefficient cases, and checks dependence by varying one score over three admissible values. It also checks invertible row changes and appended outputs. These finite checks supplement the proof. The separate frozen-bank recomputation of the main-study report cells is an audit of real banks and is reported in Appendix D.5.

Why an unrestricted pairwise judge changes the target. For a concrete boundary example, take three responses A, B, C and an antisymmetric preference function with $q(A, B) = q(B, C) = q(C, A) = 1$. It cannot equal $r(y') - r(y)$ on every pair: summing these three scalar differences would give zero rather than three. If both policies are uniform on the three responses, a coupling that always selects $(A, B), (B, C), (C, A)$ has expected preference 1, while the reversed coupling has expected preference -1, with identical policy marginals. Scalar-score expected differences are marginal quantities; arbitrary pairwise preferences need an explicitly specified joint sampling law. A new preference API therefore cannot be inserted as a same-estimand baseline without additional assumptions.

B MODELS, DATA AND IMPLEMENTATION PROTOCOL

B.1 PROMPT AND CHECKPOINT PROVENANCE

Three checkpoint regimes appear in this paper. They answer different questions and are never pooled; in particular, the three controlled endpoints are three training seeds, not three points of one trajectory, and generation repetitions are never counted as training seeds.

Table 2: The three checkpoint regimes. Sources are the frozen job records; the online protocol is Table 3.

Regime	Checkpoints	Training and use
Controlled fixed endpoints	Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct; three endpoints `blend75_202609084/085/086`, one per training seed	8 optimizer steps at learning rate 10^{-7} on fixed responses (objective in Appendix B.2); saved offline and never retrained. Used in Section 5.2 and Appendix C.
Offline distance scan	Steps 2, 8 and 32 of one trajectory per seed: three trajectories per small model, one for Qwen2.5-7B-Instruct	32 steps at 10^{-7} repeating the eight fixed batches four times, FP32, not online GRPO; output caps 256 and 768. Used in Section 5.4 and Appendix E.2.
Online GRPO	Qwen2.5-1.5B-Instruct base and steps 10, 50 and 250 of three 250-step trajectories	LoRA GRPO, learning rate 10^{-5}, 16 prompts $\times$ 8 fresh completions per step. Used in Section 5.3 and Appendix D.

The controlled studies use an initial bank of 16 fixed prompts, a comparison bank of 32 prompts drawn from historical response pools, and a strict holdout of 32 prompts that excludes all 614 recorded historical prompt texts (Appendix C.1). All prompts are open-ended generation tasks from No Robots, WildChat and ELI5 (Rajani et al., 2023; Zhao et al., 2024; Fan et al., 2019). The strict holdout has two sampling strata: three No Robots and thirteen WildChat prompts, and sixteen ELI5 prompts. The online study evaluates 128 prompts excluded from training and from historical pools (12 No Robots, 52 WildChat and 64 ELI5; Table 3), split into 64 selection and 64 test prompts; the test split has 32 ELI5, 26 WildChat and six No Robots prompts. Strata name data sources; they are not replications. This subsection closes with how sampling budgets, residual ceilings and incomplete prefixes were handled in the controlled campaign.

B.1.1 EVIDENCE BOUNDARIES

Sampling budgets. Completed distance evaluations retain every frozen prompt and checkpoint. Residual pools resume from their saved proposal prefix when necessary, with a final ceiling of 8,192 proposals; this changes the computation allowance while retaining the candidate stream, model

checkpoints and stopping policy. Saved-prefix hashes are audited in Appendix G.5.1. No exhausted prefix is treated as a zero-gain observation or removed from the completed results.

The initial four-repeat Qwen study reuses a fixed prompt set and fixed short offline updates. The separately reported strict holdout provides new-prompt validation; generation repetitions do not constitute additional training-seed replication. Finite-support expected scores are exact for the constructed proxy distributions and fixed cached scores only. The repeated baselines on the comparison bank and both 7B generation conditions have passed reconstruction checks; their query-count evidence is reported in Appendices C and E. Judge-conditioned precision and total-cost results are included only after their scoring coverage and accounting checks pass.

B.2 Update procedures

The controlled and distance checkpoints come from a fixed-response, length-normalized, advantage-weighted log-probability objective; its definition and schedule are given here. It is not online GRPO. The online GRPO protocol, the evaluation grid and the final 93-bank inventory follow.

B.2.1 Fixed update trajectories

The distance study uses a frozen offline, group-relative log-probability objective. Each prompt group contains eight saved responses with fixed 40-rubric score vectors s_i. For the frozen weight vector w, let $R_i = w^\top s_i$ and $A_i = (R_i - \overline{R}_g)/(\mathrm{sd}_g(R) + 10^{-4})$, with sample standard deviation within the eight-response group. The minibatch objective is

$$\mathcal{L}(\phi) = -\frac{1}{m}\sum_{i=1}^{m} \frac{A_i}{|y_i|} \log p_\phi(y_i \mid x_i). \tag{4}$$

FP32 Adam updates use learning rate 10^{-7} and gradient-norm clipping at 1. The same eight frozen minibatches are repeated four times, saving parameters after steps 2, 8 and 32. Qwen 1.5B and Smol 1.7B use seeds 202609084, 202609085 and 202609086; Qwen 7B uses the first seed. The 7B training candidates are retokenized with its own tokenizer, retaining decoded text and stopping status. These are fixed offline trajectories, not online GRPO rollouts. The evaluated-policy guarantees apply to their saved distributions. The released numeric configuration includes the full reward vector, minibatch sizes, source hashes and all seven trajectory protocol hashes.

B.2.2 Final GRPO protocol and result inventory

The final bundle distinguishes the controlled fixed-response update experiments from the three fresh-response GRPO trajectories. The former use fixed offline endpoints reached by a fixed number of low-learning-rate update steps; the latter use online sampled completions and group-normalized rewards.

The main matrix has 72 comparison banks and three independent reference banks. Residual-group validation adds 18 banks, for 93 in total. The versioned cache contains 338,653 successfully scored unique inputs across these banks; the count is an experimental union, not a deployment query count. Full scoring permits direct auditing of CAFA. Query savings in the paper compare the inputs each fixed task would purchase, while validation procurement, failed API responses and generation restarts remain in separate raw records. Historical studies retain their original model/judge identities and are not pooled into this cache count.

The released analysis contains 504 main-study target reports and 126 residual-group reports, with all mean and covariance identities checked. Twelve earlier isolated timing replays reproduce the original budget-4 independent-AR/shared-token response identities. Seven further uninterrupted replays cover all identified main-grid cells with failure/recovery histories. These 19 timing replays add no statistical samples; the six new Tree budget-4 banks do. The final grid, including its new cells, is independently reconstructed under the original 5,000-resample stratified test protocol. Three complete training backups, all frozen banks, token probabilities, namespaces, pilot decisions and attempt ledgers remain preserved. The paper-source bundle supplies the generated tables and figures; raw-model and scoring assets remain separate because they are substantially larger.

Table 3: Frozen online GRPO configuration. Policy exports merge the saved adapters for FP32 evaluation.

Item	Value
Policy	Qwen2.5-1.5B-Instruct, FP32
Training implementation	TRL 0.24.0 GRPOTrainer; token importance ratios; group reward scaling; one iteration per rollout
Seeds and steps	202609121, 202609122, 202609123; 250 optimizer steps each
Optimizer	AdamW; learning rate 10^{-5}; 10 warmup steps
GRPO coefficients	KL coefficient 0.04; clipping parameter 0.2
Batch	16 prompts $\times$ 8 fresh answers; microbatch 2, gradient accumulation 64
LoRA	Rank 32, alpha 64, dropout 0; q/k/v/o and gate/up/down projections
Training reward	Local Qwen2.5-7B-Instruct-AWQ rubric judge; original 40-criterion natural weights
Evaluation judge	Separate DeepSeek Flash cache with a frozen returned model identity and fingerprint
Sampling	Temperature 1, top-p 1, no top-k truncation; output cap 768; EOS IDs 151645 and 151643
Compared policies	Base and steps 10, 50, 250
Evaluation prompts	128: No Robots 12, WildChat 52, ELI5 64; 64 selection and 64 test
Residual safeguards	8,192 proposals per prompt/update/repetition; probability replay tolerance 0.01

B.3 Joint samplers and probability computation

Independent Star on token sequences. Every policy is sampled with temperature 1, top-p 1, top-k 0, repetition penalty 1 and the stated length cap; the first end-of-sequence token ends a response, otherwise it is cut at the cap and kept. For a root sequence X drawn from p_0, the log-probability of the complete token sequence (including the stopping event) is replayed under each p_j with the same transform, and X is copied into policy j when $U_j \le \exp(\min\{0, \log p_j(X) - \log p_0(X)\})$. Otherwise residual proposals $Z \sim p_j$ are accepted with probability $\max\{0, 1 - p_0(Z)/p_j(Z)\}$ in their original order; proposals drawn in batches are all charged, and a cell that reaches the proposal ceiling is resumed from its saved prefix or reported as incomplete, never filled with another sample. Densities are recorded and replayed on the original token IDs, within the replay tolerance in Table 3. Random streams are named and reproducible. Giving all models the same random seed and sampling them separately does not reproduce the Independent Star law.

Baselines. Independent autoregressive sampling draws each policy with its own random stream. Shared-token (Gumbel) sampling uses the same named random stream to sample tokens for every policy. Two policies emit the same token whenever the shared noise yields the same argmax. The total-variation tree first estimates pairwise total-variation distances from a score-free pilot (16 selection prompts, eight draws per policy, log-probabilities only), builds the minimum spanning tree rooted at the base, and couples each child with its parent by the same copy-or-residual construction as the star, with the parent in place of the base; the pilot is charged to every tree task. These are established constructions used as inputs to CAFA. Shared acceptance and residual grouping are extensions (Appendix H).

Configuration a reimplementation must match. Assumptions, decoding, stopping, identity hashing, random-stream derivation, proposal caps and scoring calls of the controlled campaign are extracted from the frozen job records.

B.3.1 Assumptions and reproduction configuration

Assumptions the results actually need. **(A1) Fixed policies.** $p_0, \dots, p_J$ do not change during evaluation and do not depend on acquired feedback. This is what allows a query to be deferred until the whole batch is aggregated; an adaptive policy may need feedback to choose its next action. **(A2) A report fixed in advance.** C, the prompt set and the budget n are all fixed before acquisition, because each of them changes the query set. The zero-query result for an all-copy draw additionally needs

a zero-sum contrast, $C\mathbf{1} = 0$; anchored and other general linear reports use the same aggregation with their own support. **(A3) A scoring function that is deterministic given its version.** $r_x(y)$ depends on the version, the prompt and the response and on nothing else. A stochastic judge is reached through a versioned cache, so every claim here is a claim about that cached function — not about repeated judge noise, and not about human preference. **(A4) Bounded scores, otherwise unrestricted.** Boundedness gives the estimator a well-defined mean; the absence of further structure linking scores across inputs is what makes the remaining queries necessary in Proposition 1. If scores were known to be, say, a linear function of features, fewer queries could suffice. **(A5) Two identities, kept apart.** Scoring identity is exact equality of decoded text; probability identity is the sampled token sequence. **(A6) Densities available on realized tokens.** Shared acceptance needs $p_j(y)/p_0(y)$ on the realized tokens; we replay it and accept the run only within a stated tolerance. The cancellation rule itself needs none of (A6).

Configuration. Table 4 is extracted from the frozen protocol, density and judge records of the completed jobs. A reimplementation that matches the window, the aggregation scope, the cache key and the two identities will reproduce the query counts exactly, because the counts are set functions of the realized texts; matching the generated texts themselves additionally requires the random-stream rows and the same kernels.

B.4 Scoring, identities and frozen analyses

A scoring identity is the SHA-256 of the exact prompt and decoded response (special tokens removed); the cache key adds the scorer version (rubric, judge model or deployment, temperature 0, thinking disabled, output schema). DeepSeek Flash returns a system fingerprint, checked on every call; Qwen Flash can only be pinned by model name. Successful scores, terminal refusals and pending calls are stored separately. Accessed identities (including cache hits) and newly paid calls are counted separately.

B.5 Acquisition algorithms

Both algorithms use the per-draw and per-prompt coefficients $b_{i,t,u}$ and $A_{i,u}$ of Equation (2) and the supports $S_{\mathrm{mean}} \subseteq S_{\mathrm{draw}}$ of Section 3.2; they differ in when scores are bought and which statistics are requested. Algorithm 1 is the specification of the released software: a fixed bank and an explicit report mode, so a mean-only report buys S_{mean} and a mean-and-covariance report buys S_{draw}. Algorithm 2 describes the implementation used in the source study to produce all reported bank results. It processes draws as they arrive, can report after every window of W draws, always acquires the covariance support before reporting uncertainty, and accumulates its coefficient map rather than resetting it, so a score bought early is never refunded when later draws cancel it (Appendix A.4).

The windowed variant is presented below, together with notes from the source study on provenance, online reporting and the distinction between caching and contrast evaluation.

Acquisition and provenance. Coefficients are accumulated separately for each prompt, contrast and budget. With $W = n$ the mean stage accesses exactly S_{mean}; uncertainty can enlarge that set. Smaller windows also support earlier reports and never refund an acquired score, because the coefficient map is cumulative: an early nonzero coefficient may cancel later. $|S_{\mathrm{acq}}|$ counts accessed inputs, including cache hits; newly paid calls are recorded separately. Scoring identities, the cache key and the replay of probabilities on the original sampled token IDs are defined in Appendices B.3 and B.4, and Appendix B.3.1 specifies the configuration.

Online reporting and uncertainty. The retained maps B_t determine which feedback each draw requires, and their union support suffices to reconstruct all per-draw contrast vectors and hence their sample variance and covariance. The algorithm explicitly acquires that additional support before passing the draw vectors to the uncertainty routine; U records the chosen estimand, sample covariance and interval procedure, and a terminally unavailable score instead produces the bounded record of Appendix F.4.1 under the missing-feedback rules of Appendix F.4. Proposition 3 (Appendix A.4) states the sequence-level condition under which $W = 1$ incurs no extra feedback and the decoded-text condition it requires; the measured window and support audits are in Appendix C.3. Other samplers

Table 4: Small-model star configuration for the comparison and strict-holdout banks, extracted from frozen job records.

Item	Value
Base policies p_0	`Qwen/Qwen2.5-1.5B-Instruct` at snapshot `989aa7980e4c`; `HuggingFaceTB/SmolLM2-1.7B-Instruct` at snapshot `31b70e2e869a`
Fixed updates $p_1..p_3$	`blend75_202609084, blend75_202609085, blend75_202609086`, learning rate 1e-07, 8 optimizer steps, saved offline and never retrained during evaluation
Contrast	$C = [-\mathbf{1}, I_3]$, fixed before acquisition
Prompt sets	Comparison bank, 32 prompts (`30815c9a2a46`); strict holdout 32 prompts (`6e2d4578862c`) after excluding 614 recorded historical prompt texts
Sampling budget	64 root draws per prompt, generated as 8 batches of 8; analysed at $n \in \{8, 16, 32, 64\}$
Decoding	temperature 1.0, top-p 1.0, top-k 0, typical-p 1.0, repetition penalty 1.0, no-repeat n-gram 0, 1 beam, renormalized logits, cap 768 new tokens
Precision	FP32, `sdpa` attention
Stopping rule	first end-of-sequence token ends the response, otherwise the response is cut at 768 tokens and recorded as `length`; capped responses stay in the analysis
End-of-sequence ids	Qwen `[151645, 151643]` / pad `151643`; SmolLM2 `[2]` / pad `2`
Scoring identity	decoded text with special tokens removed; hashed as `sha256(json({QUESTION, RESPONSE}))`
Probability identity	the sampled token ids; every density is replayed on them and accepted within 0.01 nats
Random streams	seed `int(sha256(namespace)[:8],16)`, uniforms `(int(sha256(namespace)[:13],16)+0.5)/2**52`
Acceptance namespaces	independent `PREFIX:copy:independent:<checkpoint>:<pid>:<rid>`; shared `PREFIX:copy:shared:<pid>:<rid>` — the update index is absent, so one uniform serves every update
Residual stream	`PREFIX:residual:<checkpoint>:<pid>:<used>`; proposal $Z \sim p_j$ accepted with probability $(1 - p_0(Z)/p_j(Z))_+$
Residual proposal cap	128 batches of 8 = 1024 proposals per prompt and update; one accepted pool serves both star arms in a job
Scoring versions	deepseek_flash = `deepseek-flash`; qwen_flash = `qwen3.8-flash`; qwen_max = `qwen3.8-max-0902`
Scoring call	temperature 0, at most 256 output tokens, thinking disabled, json_object with one integer key overall_score in 1..10
Cache key	the eight fields `system`, `requested_model`, `returned_model`, `system_fingerprint`, `temperature`, `thinking`, `question`, `response`; the last two are the hashed input, and the system prompt hashes to `538965bce0b3`
Aggregation	`coefficients[uid(base)] -= e_c ; coefficients[uid(endpoint)] += e_c`; query set {`uid : coefficient vector is not the zero vector`}
Identity assertion	`assert np.max(abs(total-naive))<1e-10`, evaluated for every method, budget and cached judge
Strong baselines	`independent_ar`, `gumbel_ar`, `TV_MST`; the tree is chosen from a pilot of 8 samples per policy per prompt with no score access
Incomplete cells	primary comparisons retain every frozen prompt and propagate unavailable scores through $[1, 10]$; historical common-coverage audits state their intersection explicitly

Algorithm 1 CAFA for a fixed bank.

Require: bank $\{Y_{i,t,j}\}$; report C with statistics $s \in \{\text{mean},\ \text{mean+cov}\}$; scorer r at version v with available-score cache $\mathcal{K}_v$; optional score range $[r_{\min}, r_{\max}]$

for each prompt x_i and draw $t = 1, \ldots, n_i$ **do**
  $b_{i,t,\cdot} \leftarrow 0$
  for each active column j **do** ▷ no score is read here
    $u \leftarrow \kappa(x_i, Y_{i,t,j})$; $b_{i,t,u} \leftarrow b_{i,t,u} + C_{:j}$
  end for
end for
$A_{i,u} \leftarrow \sum_t b_{i,t,u}$; $S_{\text{mean}} \leftarrow \{A_{i,u} \neq 0\}$; $S_{\text{draw}} \leftarrow \{\exists t : b_{i,t,u} \neq 0\}$
$S \leftarrow S_{\text{mean}}$ if $s = $ mean, else S_{draw}
for $u \in S \setminus \text{dom}(\mathcal{K}_v)$ **do** ▷ acquire missing feedback
  $\mathcal{K}_v[u] \leftarrow r(u)$, or record the failure; an unavailable score is never set to 0
end for
$M \leftarrow S \setminus \text{dom}(\mathcal{K}_v)$
if $M \neq \varnothing$ **then**
  if a score range is declared **then**
    return per-output mean bounds (Appendix F.4), no covariance; $|S|$ and the new paid calls
  else
    return an incomplete report with missing identities M; $|S|$ and the new paid calls
  end if
end if
$\bar{Z}_i \leftarrow n_i^{-1} \sum_{u: A_{i,u} \neq 0} A_{i,u} \mathcal{K}_v[u]$; $\widehat{\theta} \leftarrow \sum_i w_i \bar{Z}_i$
if $s = $ mean+cov **then**
  $Z_{i,t} \leftarrow \sum_{u: b_{i,t,u} \neq 0} b_{i,t,u} \mathcal{K}_v[u]$ for all t, including all-zero draws
  $\widehat{\Sigma}_i \leftarrow$ sample covariance of $\{Z_{i,t}\}_t$; $\widehat{V} \leftarrow \sum_i w_i^2 \widehat{\Sigma}_i / n_i$
end if
return $\widehat{\theta}$, and $(\widehat{\Sigma}_i, \widehat{V})$ if requested; $|S|$ and the new paid calls

can need extra accesses, and acquisition equality does not turn a finite-sample variance estimate into a coverage guarantee.

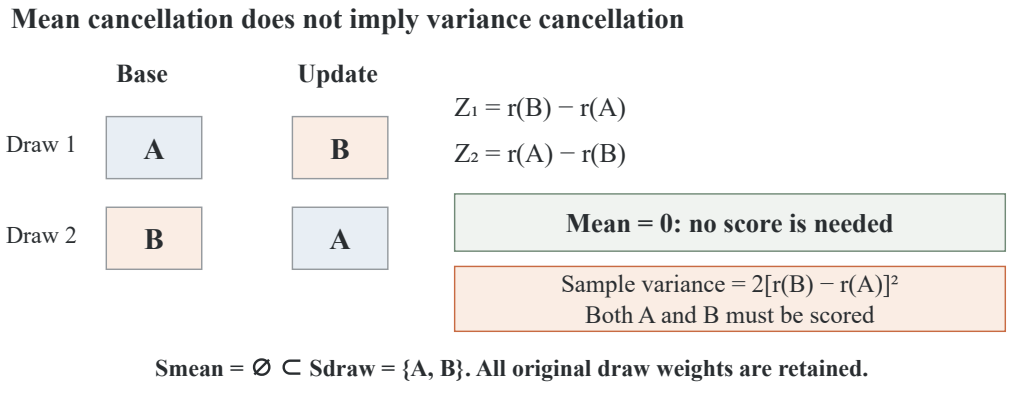


Figure 6: **The requested statistic determines which scores survive.** The two mirror draws cancel in the mean, but their ordinary sample variance still depends on both scoring identities. Both draws and their original weights are retained.

C CONTROLLED-UPDATE EVIDENCE

C.1 EXPERIMENT SETS AND COMPLETED COVERAGE

The controlled experiments use the fixed-endpoint models of Table 2 and the sampling and scoring protocol of Appendix B. Three prompt sets serve different purposes: an initial bank of 16 fixed

Algorithm 2 Contrast-Aware Feedback Acquisition (CAFA), for one prompt $x = x_i$.

Require: contrast matrix $C \in \mathbb{R}^{d\times(J+1)}$ with $C\mathbf{1} = 0$; budget $n \geq 2$; window W dividing n; joint sampler $\mathcal{S}$ with marginals $p_0, \ldots, p_J$; scoring interface R_v at version v with cache $\mathcal{K}$ and key κ

Ensure: gain estimate $\widehat{\theta}(x)$, the query set S_{mean}, the inputs accessed S_{acq}, and an uncertainty record U

$A \leftarrow$ empty map, scoring input $\mapsto$ coefficient in $\mathbb{R}^d$; $\quad S_{\mathrm{acq}} \leftarrow \emptyset$
for $t = 1, \ldots, n$ **do**
  $(Y_{t0}, \ldots, Y_{tJ}) \leftarrow \mathcal{S}(x)$ ▷ keep token IDs; densities use these IDs
  $B_t \leftarrow$ empty coefficient map
  for $j = 0, \ldots, J$ **do**
    $y \leftarrow \mathrm{text}(Y_{tj})$ ▷ decode, drop special tokens
    $B_t[y] \leftarrow B_t[y] + C_{:j}; \quad A[y] \leftarrow A[y] + C_{:j}$ ▷ aggregate before querying
  end for
  if $t \bmod W = 0$ **then** ▷ $W = n$: one batch; $W = 1$: every draw
    $S \leftarrow \{y : A[y] \neq 0\} \setminus S_{\mathrm{acq}}$
    $r[y] \leftarrow \mathcal{K}[\kappa(v, x, y)]$, else $R_v(x, y)$ stored there, for $y \in S$
    $S_{\mathrm{acq}} \leftarrow S_{\mathrm{acq}} \cup S$
  end if
end for
$S_{\mathrm{mean}} \leftarrow \{y : A[y] \neq 0\}; \quad \widehat{\theta}(x) \leftarrow n^{-1} \sum_{y\in S_{\mathrm{mean}}} A[y]\, r[y]$
$S_{\mathrm{draw}} \leftarrow \bigcup_{t=1}^{n} \{y : B_t[y] \neq 0\}; \quad S_{\mathrm{acq}}^{\mathrm{mean}} \leftarrow S_{\mathrm{acq}}$
$S_+ \leftarrow S_{\mathrm{draw}} \setminus S_{\mathrm{acq}}^{\mathrm{mean}}$
$r[y] \leftarrow \mathcal{K}[\kappa(v, x, y)]$, else $R_v(x, y)$ stored there, for $y \in S_+$
$S_{\mathrm{acq}} \leftarrow S_{\mathrm{acq}} \cup S_+$ ▷ count uncertainty accesses before reporting
$U \leftarrow \textsc{Uncertainty}(\{\sum_{y:B_t[y]\neq 0} B_t[y] r[y]\}_{t=1}^{n})$
return $\widehat{\theta}(x)$, S_{mean}, S_{acq}, U

prompts (the first shared-acceptance replication, Appendix H.2); a comparison bank of 32 prompts drawn from historical response pools, with four generation repetitions for every sampler; and a strict holdout of 32 prompts excluding all 614 recorded historical prompt texts. On the strict holdout each star arm has four generation repetitions and each other sampler two, and the four eight-prompt shards of a baseline repetition are scheduling units that together form one repetition, not independent replicates. Each repetition uses 64 joint draws per prompt. Figure 3 uses repetitions 1 and 2 for every sampler, whereas Figure 1 and Table 5 retain all four Independent Star repetitions, with repetitions 3 and 4 listed in Table 6. Every realized pair and proposal is reconstructed from the saved tokens, probabilities and random namespaces; Appendix B.3.1 lists the configuration a reimplementation must match. Nonthinking API judges return integer scores from one to ten, so the query counts of Appendix C.2 do not depend on the returned values, and judge-conditioned precision results stay separated by returned model identity. Missing feedback is handled as specified in Appendix F.4; every bank retains all prompts and none is removed from the primary analysis.

C.2 Complete CAFA results

The comparison bank (Tables 7 and 8) and the strict holdout (Table 5) are reported separately; the former is not a holdout. In Table 7, the two Independent Star columns use identical independent-star samples under Cache-only and under CAFA, whereas the third column keeps CAFA and changes the sampler to shared acceptance: the CAFA saving and the shared-acceptance increment have different denominators and must not be added. Table 5(A) retains the complete strict-holdout results for all five sampler arms, with four generation repetitions for each star arm and two for every other sampler; main-text Figure 3 instead uses the first two repetitions for every sampler, so its Star averages differ slightly from this full-set table. Part (A) keeps Q_{mean} and $Q_{\mathrm{mean+cov}}$ in separate columns: for Qwen independent AR the mean-only support 8132.50 is the mean over two repetitions of 8132 and 8133, whereas the mean-plus-covariance support is 8133 in both repetitions, which is the value in Table 9 and in the archived full-repetition precision table. Part (B) is in USD; the remaining tables of this appendix keep the original CNY of the frozen records.

Table 5: **Controlled strict holdout** (32 prompts, three fixed updates, 64 joint draws per prompt, R generation repetitions; counts averaged over repetitions), the complete repetition set; Figure 3 restricts every sampler to repetitions 1 and 2. (A) Scoring inputs for the three gains on the same sampled banks, for all five sampler arms: Q_{cache} deduplicates, Q_{mean} and $Q_{\text{mean+cov}}$ are the supports for the mean and for the mean with ordinary covariance; "Removed" is $1 - Q_{\text{mean}}/Q_{\text{cache}}$, the mean-only convention. Independent Star and Shared-acceptance Star are two different samplers scored under the same CAFA rule, each with its own sampled bank; their rows are not two readings of one bank. (B) Independent Star, cost per repetition in USD at the frozen token tariffs (Appendix G.1), with the full joint-star generation G charged to both sides, so the saving is feedback avoided on the same responses.

A. Scoring support (same responses on both sides of each row)

Model	Sampler	R	Q_{cache}	Q_{mean}	$Q_{\text{mean+cov}}$	Removed (%)
Qwen 1.5B	Independent Star	4	2700.00	1193.25	1193.25	55.8
	Shared-acceptance Star	4	2708.75	1058.25	1058.25	60.9
	TV tree	2	2509.50	917.00	917.00	63.5
	Shared-token AR	2	5409.00	5169.50	5169.50	4.4
	Independent AR	2	8133.00	8132.50	8133.00	0.0
SmolLM2 1.7B	Independent Star	4	2157.50	212.75	212.75	90.1
	Shared-acceptance Star	4	2166.25	190.50	190.50	91.2
	TV tree	2	2144.00	191.50	191.50	91.1
	Shared-token AR	2	3068.50	1850.00	1850.00	39.7
	Independent AR	2	8192.00	8192.00	8192.00	0.0

B. Cost per repetition, Independent Star (USD; same generation G in both totals)

Model	Judge	R	G	$G+F_{\text{cache}}$	$G+F_{\text{mean+cov}}$	Saved (%)
Qwen 1.5B	DeepSeek Flash	4	1.285	1.785	1.503	15.8
	Qwen Flash	4	1.285	1.481	1.371	7.5
SmolLM2 1.7B	DeepSeek Flash	4	1.327	1.741	1.369	21.4
	Qwen Flash	4	1.327	1.490	1.344	9.8

Table 6: **Additional Star repetitions retained outside the main comparison.** Strict holdout, 64 joint draws per prompt. These are repetitions 3 and 4, excluded from Figure 3 but included in the four-repetition averages of Table 5 and Figure 1. Required inputs recover both the mean and covariance; the mean-only counts coincide.

Model	Repeat	Cached	Required	Removed (%)
Qwen 1.5B	3	2,696	1,194	55.7
Qwen 1.5B	4	2,715	1,198	55.9
SmolLM2 1.7B	3	2,158	214	90.1
SmolLM2 1.7B	4	2,168	231	89.3

C.2.1 How much feedback is avoidable

Table 7 separates two effects on the comparison bank: eliminating zero coefficients in an already sampled batch, and changing acceptance dependence to create additional cancellations. These are averages over four repetitions, with 64 root draws per prompt and three fixed updates. Full aggregation is verified to reproduce the naive cached-score contrast in every checked cell, at every budget and for every cached judge. CAFA removes 52.5% and 88.9% of the initially unique inputs; shared acceptance then removes 7.4% and 19.1% of what is left, a different denominator that must not be added to the first. On the strict 32-prompt holdout, whose counts are the Independent Star and shared-acceptance rows of Table 5(A), CAFA removes 55.8% and 90.1% of the deduplicated inputs, with per-repetition ranges of 55.5–56.2% and 89.3–91.1% that the averaged table does not show. Those prompts exclude 614 recorded historical prompt texts. The before–after comparison uses identical independent-star responses.

Table 7: Required scoring inputs per judge on the comparison bank, averaged over four repetitions. The two Independent Star columns use exactly the same independent-acceptance star samples under Cache-only and under CAFA; the third column keeps CAFA and changes the sampler to shared acceptance, so its increment has a different denominator. Counts are per judge; monetary costs are evaluated separately.

	Independent Star		Shared acceptance
Model	Cache-only	CAFA	CAFA
Qwen 1.5B	2786.25	1322.50	1224.75
Smol 1.7B	2170.50	240.00	194.25

C.2.2 Does the saving survive a different sampler?

The same CAFA rule applies to every joint sampler. Table 8 reports four independent generation repetitions for each strong baseline, using the same 32 prompts in the comparison bank and 64 root draws per prompt. Independent autoregressive sampling exposes essentially no removable vector-valued queries. Shared token randomness creates agreement and therefore some cancellation, while the score-free total-variation tree exposes a substantially larger reduction. On the tree samples, CAFA removes 59.0% of the unique inputs for Qwen and 90.9% for SmolLM2. The CAFA benefit therefore extends to a strong competing coupling construction; it is not specific to the star sampler. The tree requires a mean of 1059.50 inputs on Qwen, with a run range of 1012–1097, and 195.50 on

Table 8: Unique scoring inputs on the comparison bank at $n = 64$, averaged over four independent generation repetitions per model and sampler. Cache-only and CAFA use identical sampled responses within each row. All samplers receive the same cancellation and cache rules. These counts do not require judge scores.

	Qwen 1.5B		Smol 1.7B	
Sampler	Cache-only	CAFA	Cache-only	CAFA
Independent AR	8154.75	8154.75	8192.00	8192.00
Shared tokens	5678.50	5512.50	3059.50	1845.25
TV tree	2586.00	1059.50	2146.25	195.50

SmolLM2 with a run range of 179–204; the shared-acceptance means it is compared with are the last column of Table 7. The cost–precision comparison of Appendix C.4 evaluates these samplers jointly on acquisition demand and Monte Carlo standard error. Baseline and star runs share prompts but use independently generated Monte Carlo streams. The strict-holdout baseline experiment repeats this comparison on the held-out bank with separately frozen sampling namespaces; its judge-conditioned comparison is evaluated separately.

C.2.3 Strict holdout: the same CAFA scoring strategy for every sampler

The strict holdout compares samplers on the same 32 prompts under the repetition and shard convention of Appendix C.1. All samplers receive exact-text caching and the same CAFA module, so the counts differ only through the responses each sampler produces. Table 5(A) reports Q_{cache}, Q_{mean} and $Q_{\text{mean+cov}}$ for every sampler arm; Figure 8 pairs the support needed for sample variances with the precision it buys.

C.3 Where the savings arise

Three implementations are compared on the same responses: text caching alone, skipping zero coefficients within each draw and caching the rest, and aggregating across the batch. The theory behind the equality for the star is Proposition 3; the measured evidence comes from five audits with different scopes and different units of analysis:

Audit	Unit	Scope and result
Mechanism snapshot	67 sampler–job cells	Historical 32-job campaign; within-draw plus cache equals batch aggregation in every cell
Window sweep	469 per-job budget rows	Same campaign, windows 1 to 64; every window attains the batch count
Completed support audit	56 joint-target cells	Strict-holdout and distance banks, all recorded samplers and trajectories; equal in 55
Online GRPO preservation	504 report cells	Equal in 485, every exception in a non-star sampler (Appendix D.5)
Access audit	336 base-relative star cells	14 scored jobs, no vector-level gap

These audits share data and count different units; their totals are not independent replications and must not be added into one verification sample size.

C.3.1 When is the saving available?

We separate text deduplication, cancellation within each root draw followed by caching, and cancellation across the entire batch. The last two have exactly equal required-input counts in the mechanism snapshot, and every audited window attains the same final acquisition count (scopes and counts in Appendix C.3). In the audited star cells these savings are therefore available online: reporting after every draw requires no additional scores. The expanded audit checks this equality again for each new sampler and budget; generic non-star outputs can require additional inputs for variance. Appendix C.3.2 gives the decomposition, the sequence-level sign-purity result and its decoded-text boundary. Exact sample variance is available at the same acquired support in the audited star cells; confidence-interval coverage remains a separate question.

C.3.2 Online CAFA and uncertainty access

Section 3.2 and Appendix A.4 define the per-draw coefficient $b_{t,y} = \sum_{j:\phi(Y_{tj})=y} C_{:j}$, where ϕ decodes the sampled sequence, and the two supports $S_{\mathrm{mean}} \subseteq S_{\mathrm{draw}}$. Cancelling within each draw and caching accesses S_{draw}; whole-batch cancellation accesses S_{mean}. Their equality is a directly checkable condition under which immediate reporting attains the batch CAFA query count, and Proposition 3 gives the sequence-level case together with its decoded-group condition. The examples and audits below establish how it applies to the recorded scoring identities.

Why decoding matters. Sequence-level splitting does not automatically give maximal coupling of decoded texts. A concrete boundary uses four atoms with $p_0 = (.4, .1, .4, .1)$ and $p_1 = (.1, .4, .1, .4)$, decoded as (X, X, Y, Z). Two noncopy draws, atom pairs $(1, 4)$ and $(3, 2)$, have positive joint probability 0.0225. Their decoded contrasts are (X, Z) and (Y, X): the score of X cancels from the mean but remains in the draw contributions. With $r(Y) = 0$, $r(Z) = 2$, setting $r(X)$ to zero or one preserves the mean at one and changes the sample variance from two to zero. Therefore our token-space proposition is not an unconditional decoded-cache theorem. The actual text-support equality is audited directly.

Measured mechanism. The snapshot and window scopes are listed in Appendix C.3; the agreement holds in every cell, including the available tree and autoregressive baselines. What that list does not show is why cross-draw aggregation adds nothing here: across the audited records 115 scoring inputs occur in more than one root draw, and none of them eliminates an additional vector-valued query by cancelling across draws. The saving therefore does not require delayed feedback on these samples. Figure 7 separates the mechanisms.

Which uncertainty quantities are available? Given S_{draw}, the vectors $D_t = \sum_y b_{t,y} r_x(y)$ are known exactly. Their sample variance, covariance and resampling statistics can then be computed without additional score acquisitions. This is exact recovery of sample statistics, not exact knowledge of population variance or guaranteed confidence-interval coverage. For a general sampler, the mean may have a smaller support; Appendix A.2 and Figure 6 give the boundary case.

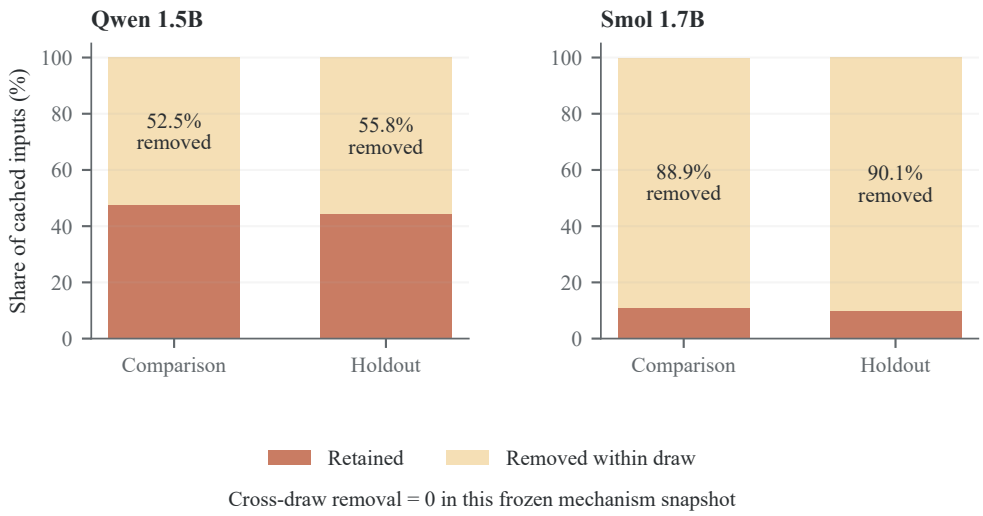


Figure 7: Decomposition on fixed generated samples. Cancelling within a draw and then caching attains the complete observed reduction. Cross-draw aggregation adds no measured vector-query saving in this snapshot.

The access audit listed in Appendix C.3 covers 43,781 input incidences and finds no vector-level gap. For individual contrast rows of other samplers, gaps do occur: at $n = 64$, the tree has 2 hidden groups among 2,556, and independent autoregressive sampling has 7 among 36,829. A row-level gap is not a joint-target gap. These row-level gaps need not produce extra purchases when another row of the jointly reported vector already acquires the same input. This explains their coexistence with the vector-level zero gap above.

C.4 Precision, gains and cost

Two MCSE summaries, kept apart. Precision on the strict holdout is summarized in two ways that use different repetition sets and aggregation orders; the resulting summaries are not interchangeable. This appendix shows the one that puts every sampler on a common repetition count: Table 9 restricts each sampler to its first two generation repetitions and averages, over those repetitions, the root-mean-square MCSE across the three gains, reporting 1190.50 mean inputs for the Qwen independent star. The other summary uses the complete repetition set for each sampler: four repetitions for the star arms and two for every other sampler. For each update, MCSE is averaged across repetitions before the root mean square is computed across updates. Under this convention, the same arm has 1193.25 variance-support inputs. Complete values, the aggregation definition and score status for each row are retained in the supplementary numeric archive. Both summaries were reproduced from the unrounded records for each repetition. Neither corrects the other, and their rows should not be combined in a single table. Table 5 continues to carry the full-repetition scoring requirement, and Figure 8 uses the first two repetitions with the single-repetition RMS convention.

Cost under a common generation charge. Table 5(B) charges the complete joint-star generation task to both sides of each comparison, so its percentage isolates feedback reduction on identical generated responses; it is not a comparison of two independently deployed workflows, and it is the four-repetition result, whereas main-text Figure 3(b) uses the first two repetitions. The same four-repetition amounts in their recorded CNY, per repetition and judge, are retained in the supplementary numeric archive. Cross-sampler workflow costs with their own overhead conventions are in Table 9, break-even prices in Appendix G.3, the general pricing definitions in Appendix G.1, and the recorded task intervals behind them in Appendix G.5.1.

At the fixed root budget, the tree requires fewer scoring inputs on Qwen, while the tree and shared star are nearly tied on SmolLM2. Figure 8 places these counts beside their observed precision; a count advantage alone does not establish a precision-matched winner.

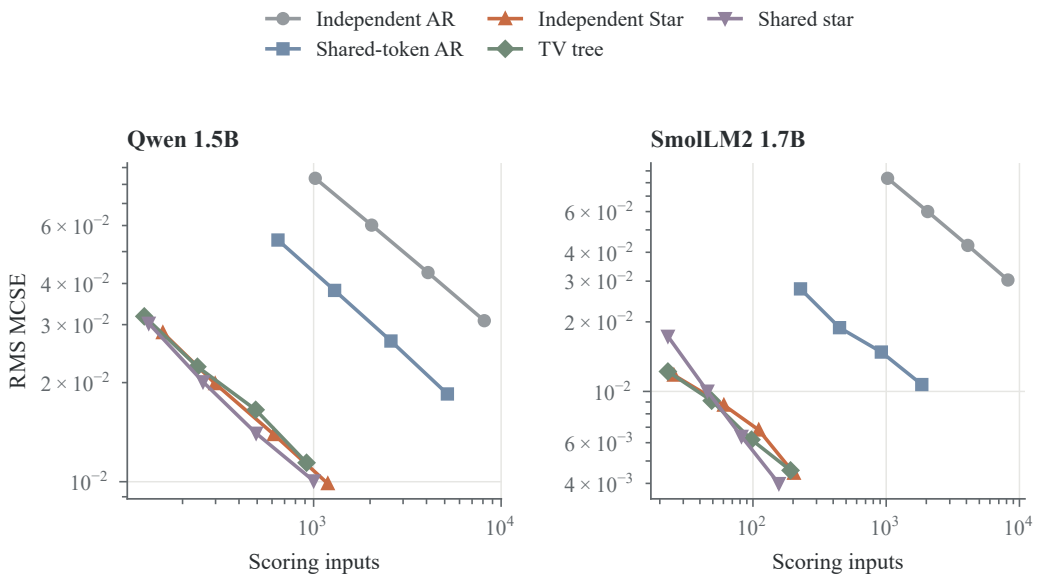


Figure 8: **Feedback demand and Monte Carlo precision across samplers.** Strict holdout, DeepSeek Flash, fixed budgets $n \in \{8, 16, 32, 64\}$, first two generation repetitions for every sampler. The horizontal axis includes any additional variance-support inputs. The vertical axis averages the single-repetition RMS MCSE across the three gains. Both axes are logarithmic; known-truth error is evaluated in the finite-support experiment.

Snapshot and incomplete feedback. The primary score snapshot is `966b4222f3b1`; the supplementary cost and application analysis is identified by `77b616340eb3`. Valid scores, terminal provider refusals and pending calls are tracked separately. An unavailable score contributes its full interval $[1, 10]$ through its signed coefficient; no prompt is dropped and no missing value is replaced by zero. Unless a figure specifies fixed-prompt Monte Carlo intervals, gain intervals resample prompts within the two frozen sampling strata. Fixed-prompt Monte Carlo standard errors condition on these prompts and the versioned score cache. Complete campaign usage and the distinction between task expenditure and per-evaluation costs are reported in Appendix G.5.1.

The gains recovered by those scores. Appendix B.2.1 specifies the frozen offline objective and training schedule that produce the compared checkpoints. Figure 9 reports the corresponding quality changes at the 768-token cap. These gains are measured on the ten-point scale of the fixed, versioned judge. The small-model entries average three saved update trajectories; the 7B entries follow one trajectory. Each uses 32 prompts and 64 root draws per prompt per trajectory. Thus the figure shows both the magnitude of the update and the Monte Carlo resolution at which it is compared. Per-trajectory and prompt-resampling results accompany the machine-readable tables.

Complete cost for the same generated comparison. We reconstruct complete task intervals from the start and exit events recorded by the persistent controller. These intervals include model loading, response generation, cross-policy probability computation, residual proposals, saving and in-process waiting. Reserved GPU time is priced at the experimental rental rate of 3 CNY per card-hour. The independent and shared star arms were generated together using a shared residual pool; we charge that entire recorded generation cost to both sides of each before–after comparison. Consequently, no allocation of shared generation overhead is needed to establish the CAFA saving.

To price the scores that cancellation had made unnecessary, we query the frozen unscored responses with the same two version-checked judges and retain their actual input and output token counts. The comparison uses a common cold-cache tariff for both arms: 2/8 CNY per million input/output tokens for DeepSeek Flash and 0.8/2.7 for Qwen Flash. This separates the algorithmic difference from changing platform discounts. The generation bank, stopping rule and realized gain are identical before and after CAFA.

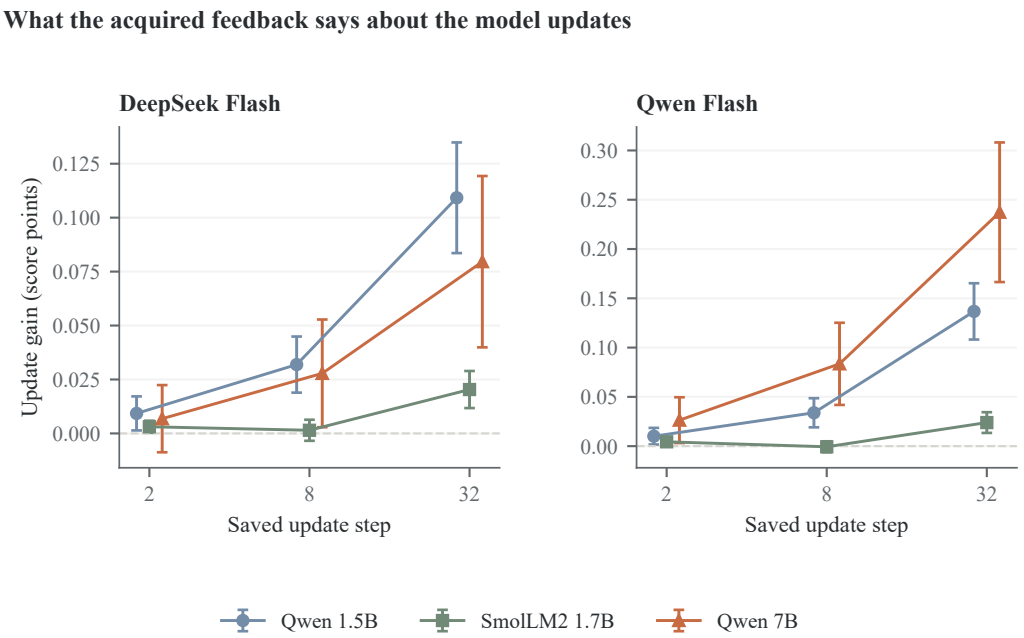


Figure 9: **Quality improvement and comparison resolution in the offline checkpoint scan.** These are the measured quality changes at checkpoints 2, 8 and 32 of the offline distance regime. Two fixed Flash judges score the same generated responses. Error bars are the fixed-prompt Monte Carlo intervals (mean ± 1.96 fixed-prompt MCSE). The small-model configurations have three trajectories and the 7B extension has one.

The resulting total-cost reduction is 7.5–21.4%, while every realized gain is unchanged. The DeepSeek column uses its published peak tariff; applying its half-price nonpeak tariff to both scoring policies gives 9.2–12.1% total savings for that judge. Coefficient processing is measured separately on the local CPU. Appendix G.5.1 separates this per-evaluation comparison from historical campaign spending and from the observed phase durations of the independently generated strong baselines.

An integrated comparison across recorded workflows. Table 9 joins CAFA scoring requirements, precision and expenditure on the same strict holdout at $n = 64$. All methods use their first two generation repetitions; the four-repeat star results remain in Table 5. The MCSE column is the mean, across these two repetitions, of the RMS standard error across the three gains. The score column reports the cost of acquiring the union of the mean and variance supports, computed from observed input/output token usage at the same historical normalized tariff.

Historical precision diagnostics. An earlier snapshot of the same campaign is superseded for every current conclusion by Tables 5 and 9. Two of its diagnostics motivated checks in the current design: comparisons at that stage were underpowered, and some prompt cells had zero empirical variance. Both are properties of that historical snapshot, not rates for the completed experiments; the finite-support and rare-event checks that address them are in Appendices F.2 and F.3.

C.4.1 Historical precision diagnostics and their support

In that snapshot the tree coverage was one Qwen repetition and two SmolLM2 repetitions against four star repetitions, on separate generation streams that share prompts rather than matched root draws; its per-sampler counts and stage ordering are superseded and are retained only in the supplementary archive.

We retain two diagnostics from that historical snapshot. Across 184 pairwise standard-error comparisons, no reported interval excluded zero and 88 rows were flagged as underpowered, which does not demonstrate equivalent precision. In some SmolLM2 configurations, 30–49% of prompt cells had zero empirical variance because all sampled pairs agreed. This rate applies to the historical snapshot

Table 9: Strict holdout, first two repetitions, DeepSeek Flash. "Extra" counts variance inputs beyond mean support. Costs are CNY per repetition. Star generation charges the complete joint-star task; the other methods charge their recorded inclusive phase plus all common startup/finalization overhead. These are reconstructed workflow costs with the stated overhead conventions.

Model	Sampler	Mean inputs	Extra	MCSE	Gen.	Score	Total
Qwen 1.5B	Ind. AR	8132.5	0.5	0.0308	6.85	10.11	16.97
Qwen 1.5B	Shared tokens	5169.5	0.0	0.0185	6.91	6.86	13.76
Qwen 1.5B	TV tree	917.0	0.0	0.0114	7.93	1.08	9.01
Qwen 1.5B	Ind. star	1190.5	0.0	0.0099	8.58	1.48	10.06
Qwen 1.5B	Shared star	1000.5	0.0	0.0101	8.58	1.22	9.80
Smol 1.7B	Ind. AR	8192.0	0.0	0.0303	5.74	10.57	16.32
Smol 1.7B	Shared tokens	1850.0	0.0	0.0107	5.77	2.58	8.35
Smol 1.7B	TV tree	191.5	0.0	0.0046	7.33	0.25	7.58
Smol 1.7B	Ind. star	203.0	0.0	0.0044	8.07	0.27	8.33
Smol 1.7B	Shared star	156.5	0.0	0.0040	8.07	0.22	8.28

rather than the completed experiments. Zero observed variance does not imply zero population variance. A precision threshold selected from such estimates can therefore look artificially favourable. The completed evidence uses all planned repetitions: strict-holdout strong baselines were absent here, and their completed generations enter Appendix C.2 and main-text Figure 3.

D ONLINE GRPO EVIDENCE

The online study (Table 2, protocol in Table 3) evaluates base and steps 10, 50 and 250 of three GRPO trajectories on 128 held-out prompts. Its 72 comparison banks are four samplers, three budgets, three training seeds and two generation repetitions; seven report targets give 504 report cells. They are fixed workflows, not 504 or 72 independent training runs, and generation repetitions are not training seeds.

D.1 ONLINE GRPO TRAJECTORIES AND CHECKPOINT SELECTION

This appendix uses the three independent online LoRA GRPO trajectories of Table 2, whose objective, optimizer, LoRA, batch, judge and sampling settings are in Table 3 (Shao et al., 2024). They are not the controlled fixed-response updates of Appendix C, and the two regimes are never pooled.

Six TV-tree budget-4 banks complete the original grid using the same exports, prompts, sampling protocol and frozen judge; they are genuinely new generation, but no additional training or reference generation was performed for them. Every TV-tree task pays for its own score-free pilot on 16 stratified selection prompts with eight answers per policy and prompt, charged to that task. Original token IDs determine probabilities; exact prompt and decoded-response identity determines scoring reuse.

D.2 COMPLETE BANK, TARGET AND BUDGET RESULTS

Figures 11–16 show the same-bank CAFA increment for all four samplers, all three budgets and six of the seven reports, covering all 432 cells. The seventh report, gains plus the base mean, yields zero savings in all 72 banks; this result is stated in the text, with per-bank records in the supplementary tables. Table 10 reports every $n = 16$ workflow cell, including RMS MCSE, accepted runtime and columns absent from the three-gain rows of Figure 16. The six tree banks at 4 draws per prompt that complete the grid are included in the 72 banks (Appendix D.2.6). Figure 10 is the six-panel version of main-text Figure 4, drawn from the same cells. The interpretation block retains the tariff sensitivity and the precision-threshold readout; both are descriptive readings of the completed fixed-budget grid, not stopping rules.

D.2.1 INCREMENT ATTRIBUTABLE TO CAFA

We first keep each generated bank and scoring function fixed. The three implementations are text caching alone, within-draw zero-coefficient skipping followed by caching, and whole-batch aggregation followed by the additional queries needed for ordinary covariance. Text caching requires Q_{cache} inputs, and within-draw skipping requires $Q_{\mathrm{mean+cov}}$. Batch aggregation first requires Q_{mean} inputs for the mean and then $Q_{\mathrm{mean+cov}} - Q_{\mathrm{mean}}$ additional inputs for covariance. Thus within-draw skipping and batch aggregation acquire the same support S_{draw}, of size $Q_{\mathrm{mean+cov}}$, for the joint mean-and-covariance report (Proposition 2).

Figure 16 prices the CAFA layer independently of coupling. Each target uses only its active policy columns for the caching baseline. The generation charge G is the same full recorded four-policy, 128-prompt task in both conditions, including for a single-target row. Thus the absolute saving is identified on the frozen bank, while the percentage uses an explicit joint-generation allocation, not a fresh pair-only runtime. F prices observed input/output token usage at the frozen judge tariff. CPU execution time is reported separately in Table 17 (Appendix G.4).

All 504 target cells reproduce the dense, fully scored means and stratified covariance with zero numerical discrepancy; all 72 fixed-rule choices are preserved. Mean and draw support agree in 485 cells. The other 19 require extra feedback for covariance, rather than a free reuse of the mean-only result. For example, independent AR, seed 1/repeat 2 at $n = 16$, needs 4065 inputs for the step-10 mean and 4071 for the joint mean/covariance report.

D.2.2 WORKFLOW COMPARISONS AND CHECKPOINT SELECTION

All 72 comparison banks reproduce the fully scored means and fixed checkpoint-selection decisions after CAFA. Variance uses the per-draw support whenever it exceeds the mean support. Figure 10 reports each training seed and generation repetition separately, with complete recorded generation costs and measured feedback token usage at the frozen tariff. Twelve earlier isolated replays replace contended independent-AR/shared-token budget-4 collection times; seven additional uninterrupted replays replace cells with failure/recovery histories. Replays match the original token and text identities and add no statistical samples. All GPU times are normalized at 3 CNY per card-hour; this is a common accounting tariff, not the new server invoice. Historical charges are preserved separately (Appendix D.2.6).

D.2.3 WHAT THE GRPO RESULTS IMPLY FOR CAFA

Useful training progress and lower comparison cost coexist. Training progress and comparison cost are separate measurements, and the independent test references of Appendix D.4 keep them apart: each reference uses independent autoregressive samples, so improvement is not agreement induced by a sampler. For all three trajectories, the final checkpoint has a higher judge score than the base (Tables 11 and 12). The $n = 16$ input, cost and MCSE ratios behind this are reported once, in main-text Table 1, with the per-cell breakdown in Table 10. These comparisons hold the prompt bank and draw budget fixed; they do not select a stopping time using the observed MCSE.

Separating sampler-induced agreement from CAFA savings. Two different quantities are easy to conflate. On a fixed star bank, CAFA removes a further 2.9–3.8% of the inputs already deduplicated within that workflow at $n = 16$; the larger cross-workflow saving of Appendix D.2.2 also includes the agreement that the coupling itself creates, and the two must not be added. The additional reduction on a fixed bank depends on the requested report: Table 13 and the complete grid of Figures 11–16 give it per target, and it is far larger for a nearby contrast than for one that includes a distant update. Within-draw and batch supports agree in 72 of 72 joint cells. The coefficient interface identifies when the simpler within-draw implementation already attains the batch support.

Fewer labels do not identify the cheapest sampler. At $n = 16$, TV-tree reduces scoring demand by 36.7–42.5% relative to independent AR plus caching, but its pilot and generation overhead also enter the comparison. Figure 10 and Table 1 therefore compare total expenditure with precision, rather than selecting a winner from label counts. The timings they use are not all same-machine benchmarks: some cells were remeasured in uninterrupted replays that reproduce the original tokens and decoded scoring identities and add no statistical observations, and the accepted historical timings

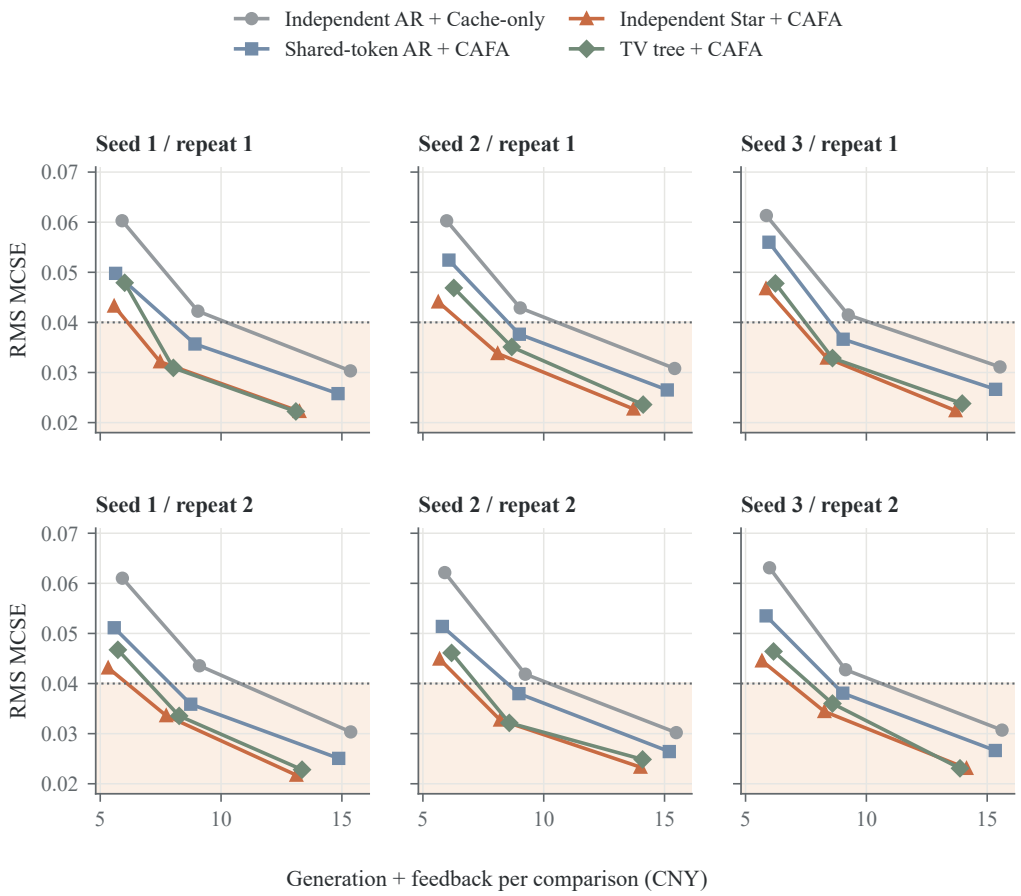


Figure 10: Per-seed and per-repetition cost–precision curves underlying main-text Figure 4. Panels separate three training seeds and two generation repetitions on the 128-prompt GRPO bank. Along each curve, budgets increase from 4 to 8 to 16 draws per prompt from left to right. RMS MCSE summarizes the three base-relative gains; the shaded region marks RMS MCSE ≤ 0.04. Independent AR uses Cache-only; the other samplers use CAFA (Ours). Costs include generation and feedback for means and covariance, with the Tree pilot and accepted uninterrupted runtimes. Costs are in CNY; the main-text figure uses USD.

are retained for the rest. Appendix G.5.1 records which cells were replaced, why, and what the historical-minus-replay difference contains.

The independent test gains and their intervals are reported once, in Appendix D.4 (Tables 11 and 12), under the current 5,000-resample source-stratified procedure. An exploratory 2,000-resample unstratified analysis of the 64 test-prompt means gave 95% percentile intervals $[0.220, 0.417]$, $[0.256, 0.455]$ and $[0.239, 0.409]$ for the step-250 gains of seeds 1–3. That is a different resampling procedure on a different unit, not an earlier printing of Table 12, and the two sets of intervals are never shown as one series. Both separate generation noise on a fixed bank from variation across the sampled test prompts, and the training seeds remain separate throughout. The step-10 prompt-bootstrap intervals span zero for all three seeds under either procedure.

Holding the measured generation costs and response lengths fixed, scaling the feedback tariff changes which $n = 16$ workflow is cheapest; Appendix G.3 reports the complete result at one quarter, one and twice the frozen tariff. These are observed expenditure comparisons at a fixed draw budget, and the complete cost–MCSE curves retain that second dimension.

A precision-constrained readout of the same fixed grid $\{4, 8, 16\}$ gives a larger operational difference, reported in Section 5.3: at RMS MCSE at most 0.04, independent star with eight draws is the cheapest eligible workflow in all six cells while independent AR needs 16 draws. The 0.04 threshold is a post

hoc readout of the completed fixed-budget grid. At four draws per prompt, the TV tree has RMS MCSE 0.0461–0.0479, so none of its six cells meets the 0.04 threshold.

Checkpoint selection needs differences, not an absolute anchor. Adding the base mean to all three gains restores the full cached scoring support in every GRPO cell, because those outputs jointly identify all four absolute policy means. Checkpoint selection has a different requirement: subtracting the common base mean preserves the maximizer, so the zero-sum gains suffice, which is the algebra of Appendix A.3. The selection results themselves, the single disagreeing selector and its estimated regret are reported in Appendix D.4; that agreement counts fixed workflows, not independent training trials.

Practical choice. Two decisions are separable. Given a bank and a report, use CAFA: it preserves the estimate and never increases the required score set. Choosing the generation workflow is a different decision, made from the measured cost–precision curve at the feedback tariff that applies, because how much CAFA saves depends on the report and how much the workflow costs depends on the sampler. Coverage evidence is reported separately in Appendices F.2 and F.3.

D.2.4 COMPLETED-GRID FROZEN-BANK AUDIT

All cells below are recomputed from the 75 frozen main/reference banks. The source manifest hashes every bank, its completion marker, the versioned score export, protocol, questions, cost ledger and accepted isolated timing replays. The original sources are checked again after the audit. Each CSV/JSON row retains the bank hash, complete target, seed, generation repeat, score identity and cost convention. No response generation, judge request or new training run is performed.

Reading the CAFA tables. AR, Token, Star and Tree denote independent autoregressive, shared-token, independent-star and TV-tree generation. $Q_{\mathrm{cache}}, Q_{\mathrm{mean}}, Q_{\mathrm{mean+cov}}$ count the inputs required by cache-only scoring, mean-only reporting and joint mean-and-covariance reporting. Immediate skipping and batch aggregation with covariance both acquire $Q_{\mathrm{mean+cov}}$ inputs; the additional covariance fee is $F_{\mathrm{mean+cov}} - F_{\mathrm{mean}}$. The heatmaps and accompanying CSV files retain all budgets and all seven targets. $F_{\mathrm{cache}}, F_{\mathrm{mean}}, F_{\mathrm{mean+cov}}$ are scoring CNY; $T_{\mathrm{cache}} = G + F_{\mathrm{cache}}$ and $T_{\mathrm{mean+cov}} = G + F_{\mathrm{mean+cov}}$ charge the full 128-prompt four-policy task. They do not identify fresh subset or single-target deployment costs. The machine-readable cells additionally expose the original attempt charge and whether an accepted isolated replay replaced it. One report target is omitted from this series because every one of its cells is exactly zero. Across all 72 banks, requesting all three gains together with the base mean restores the full cached support; the complete per-bank records are retained in the supplementary results. In this case, CAFA provides no scoring reduction because the report requires the full cached support (Corollary 1).

D.2.5 COMPLETE $n = 16$ WORKFLOW CELLS

Table 10 lists the 24 sampler–seed–repeat cells at 16 draws per prompt, with their generation and feedback costs and RMS MCSE.

D.2.6 GRID COMPLETION AND UNINTERRUPTED TIMING

The completed comparison grid is four methods $\times$ three budgets ($n = 4, 8, 16$) $\times$ three training seeds $\times$ two generation repetitions, for 72 banks. The six Tree $n = 4$ cells are inside that grid, not additional to it: they use the original exports and protocol, the same score-free pilot rule and the frozen evaluation judge, and their pilot, tree edges, copy decisions, residual acceptance and probability replay were independently reconstructed. No new training, reference generation or replay scoring was required for them. Their per-cell support counts, RMS MCSE and costs are part of the complete grid, retained in the supplementary canonical acquisition table with every other cell.

Which cells use replayed rather than original timings, the replay hardware and scope, the identity checks, and why the historical-minus-replay difference is not a failure-only estimate are all recorded once, in Appendix G.5. When reading the cost curves of this appendix, the reported generation costs use uninterrupted measurements for the affected cells.

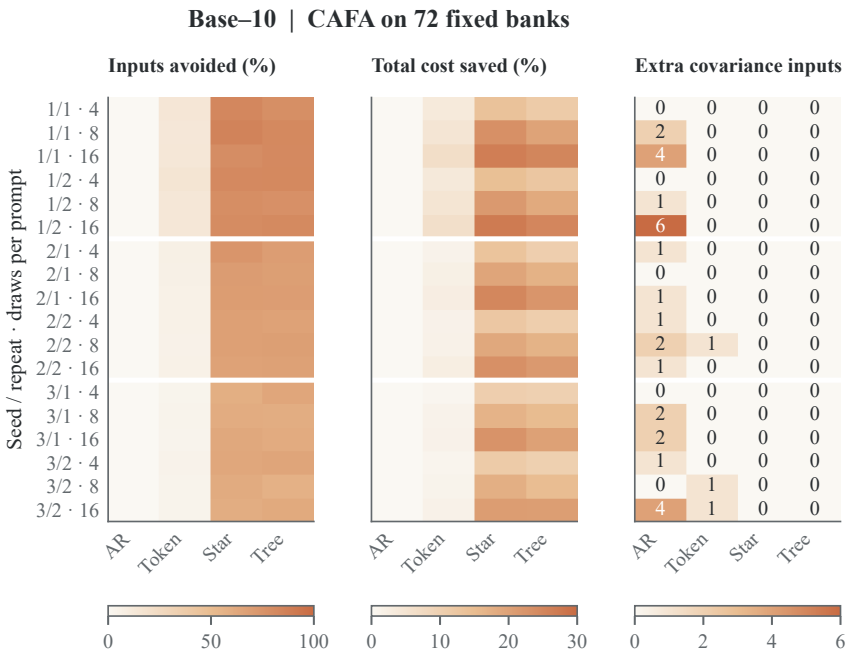


Figure 11: **Complete acquisition grid: Base–10.** Every sampler, budget, seed and generation repeat is shown (72 cells). Left: $100(1-Q_{\text{mean+cov}}/Q_{\text{cache}})$; middle: same-bank total-cost saving $100(1-T_{\text{mean+cov}}/T_{\text{cache}})$; right: $Q_{\text{mean+cov}}-Q_{\text{mean}}$, the additional identities needed for covariance. Monetary ratios use the recorded CNY totals. The complete counts, fees and unrounded values accompany the figure in the CSV bundle. Color scales are shared across report targets for the two percentage panels.

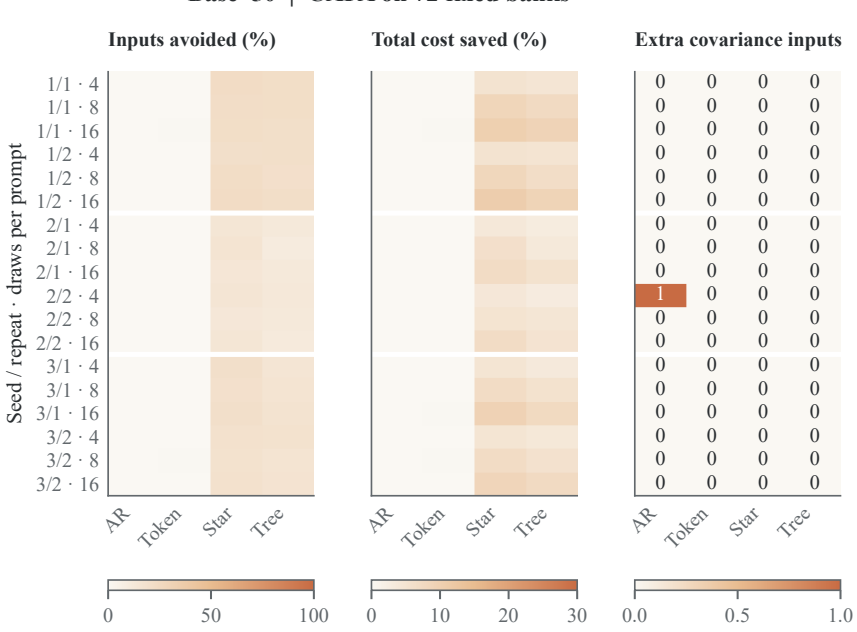


Figure 12: **Complete acquisition grid: Base–50.** Continuation of Figure 11 for this report: all 72 cells (every sampler, budget, seed and generation repeat), the same three panels ($100(1-Q_{\text{mean+cov}}/Q_{\text{cache}})$, the same-bank total-cost saving $100(1-T_{\text{mean+cov}}/T_{\text{cache}})$, and the covariance increment $Q_{\text{mean+cov}}-Q_{\text{mean}}$), the same shared colour scales, and the same CNY basis. The complete counts, fees and unrounded values accompany the figure in the CSV bundle.

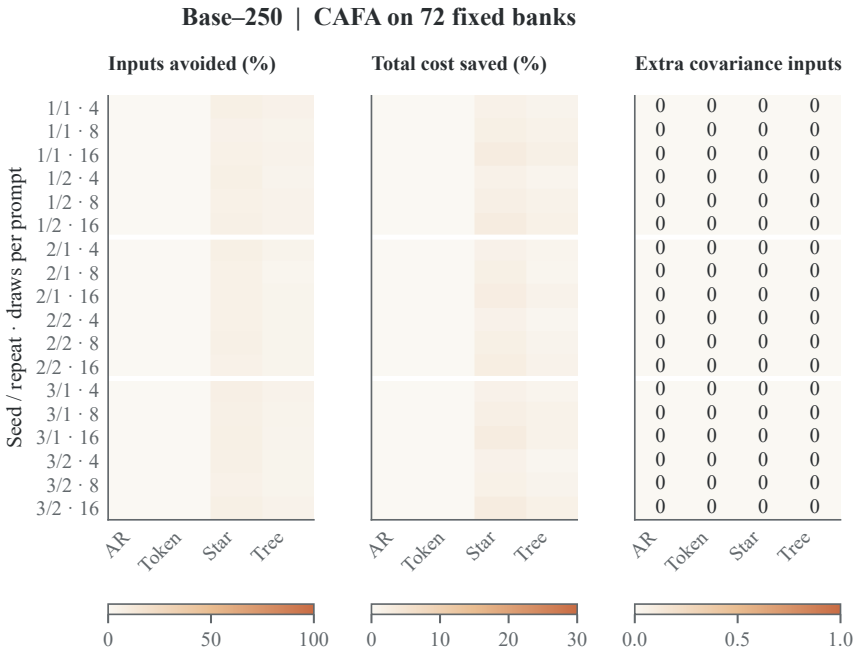


Figure 13: **Complete acquisition grid: Base–250.** Continuation of Figure 11 for this report: all 72 cells (every sampler, budget, seed and generation repeat), the same three panels ($100(1 - Q_{\text{mean+cov}}/Q_{\text{cache}})$, the same-bank total-cost saving $100(1-T_{\text{mean+cov}}/T_{\text{cache}})$, and the covariance increment $Q_{\text{mean+cov}} - Q_{\text{mean}}$), the same shared colour scales, and the same CNY basis. The complete counts, fees and unrounded values accompany the figure in the CSV bundle.

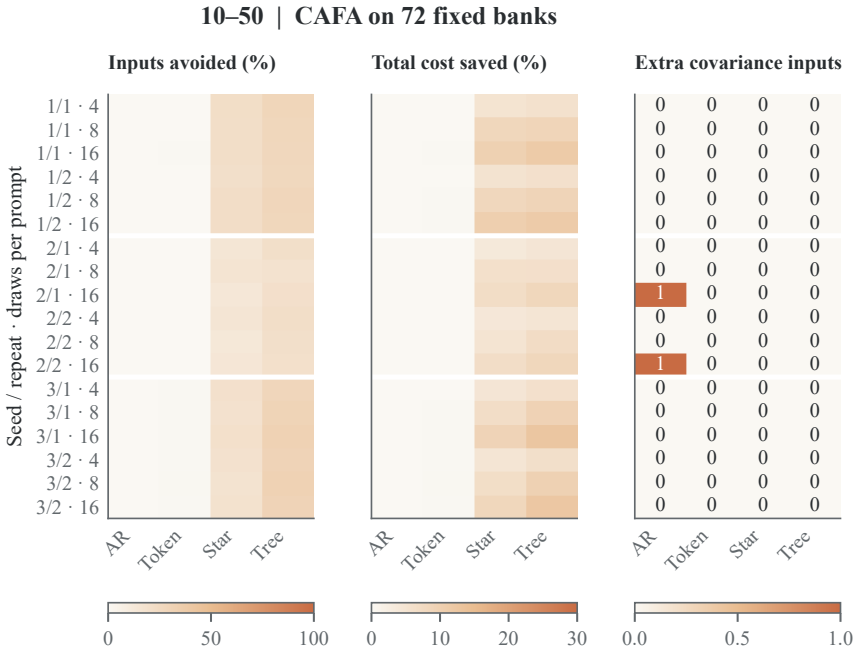


Figure 14: **Complete acquisition grid: 10–50.** Continuation of Figure 11 for this report: all 72 cells (every sampler, budget, seed and generation repeat), the same three panels ($100(1 - Q_{\text{mean+cov}}/Q_{\text{cache}})$, the same-bank total-cost saving $100(1 - T_{\text{mean+cov}}/T_{\text{cache}})$, and the covariance increment $Q_{\text{mean+cov}} - Q_{\text{mean}}$), the same shared colour scales, and the same CNY basis. The complete counts, fees and unrounded values accompany the figure in the CSV bundle.

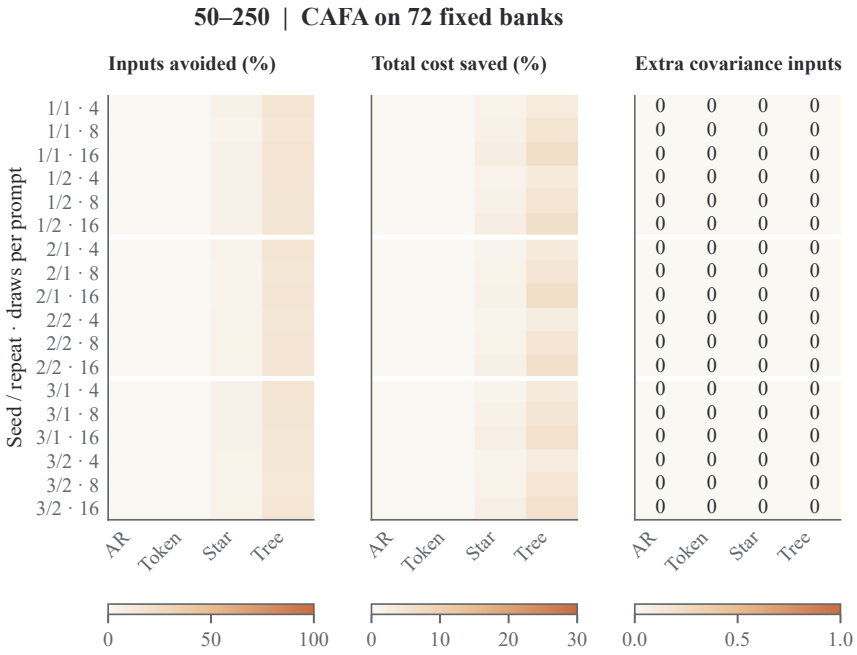


Figure 15: **Complete acquisition grid: 50–250.** Continuation of Figure 11 for this report: all 72 cells (every sampler, budget, seed and generation repeat), the same three panels ($100(1 - Q_{\text{mean+cov}}/Q_{\text{cache}})$, the same-bank total-cost saving $100(1 - T_{\text{mean+cov}}/T_{\text{cache}})$, and the covariance increment $Q_{\text{mean+cov}} - Q_{\text{mean}}$), the same shared colour scales, and the same CNY basis. The complete counts, fees and unrounded values accompany the figure in the CSV bundle.

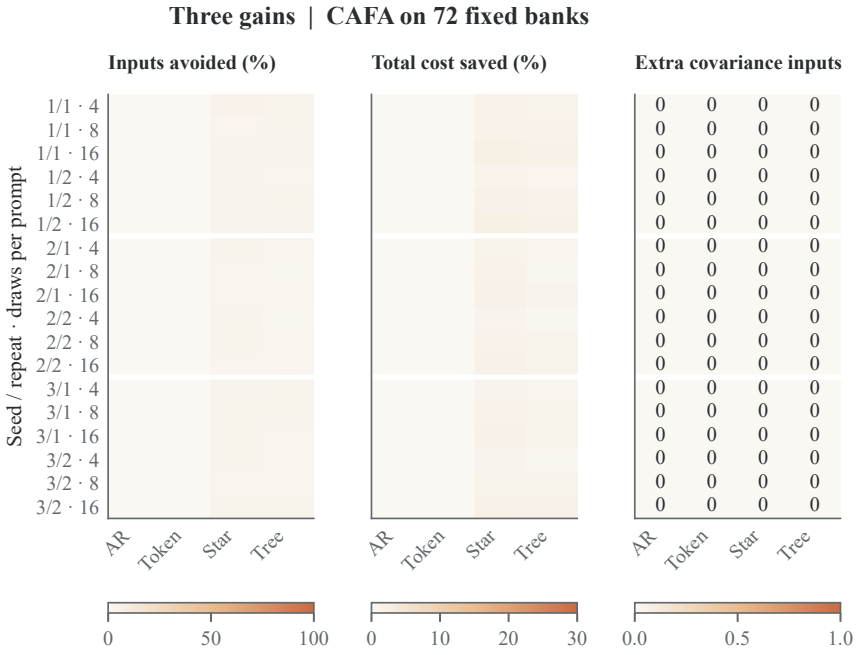


Figure 16: **Complete acquisition grid: three gains.** Continuation of Figure 11 for this report: all 72 cells (every sampler, budget, seed and generation repeat), the same three panels ($100(1 - Q_{\text{mean+cov}}/Q_{\text{cache}})$, the same-bank total-cost saving $100(1 - T_{\text{mean+cov}}/T_{\text{cache}})$, and the covariance increment $Q_{\text{mean+cov}} - Q_{\text{mean}}$), the same shared colour scales, and the same CNY basis. The complete counts, fees and unrounded values accompany the figure in the CSV bundle.

Table 10: All $n = 16$ GRPO comparisons, retaining each seed and generation repeat. Q_{cache}, Q_{mean} and $Q_{\text{mean+cov}}$ count the inputs required by text caching, exact means, and means with covariance, respectively. G is accepted full-process generation expenditure, using uninterrupted replays for failure-affected cells; $G + F_{\text{mean+cov}}$ adds measured-token feedback cost for variance support. Monetary units are CNY at the frozen tariff.

Seed/repeat	Sampler	Q_{cache}	Q_{mean}	$Q_{\text{mean+cov}}$	RMS MCSE	G	$G + F_{\text{mean+cov}}$
1/1	Independent AR	8165	8165	8165	0.0303	4.13	15.35
1/1	Shared token	7638	7634	7634	0.0258	4.21	14.84
1/1	Independent star	5394	5197	5197	0.0222	6.04	13.24
1/1	TV-tree	4872	4695	4695	0.0222	6.56	13.10
1/2	Independent AR	8166	8166	8166	0.0303	4.15	15.37
1/2	Shared token	7655	7654	7654	0.0251	4.21	14.87
1/2	Independent star	5354	5150	5150	0.0217	5.92	13.12
1/2	TV-tree	4901	4723	4723	0.0228	6.71	13.35
2/1	Independent AR	8172	8172	8172	0.0308	4.16	15.42
2/1	Shared token	7930	7929	7929	0.0265	4.11	15.11
2/1	Independent star	5751	5586	5586	0.0227	5.85	13.71
2/1	TV-tree	5267	5142	5142	0.0236	6.90	14.12
2/2	Independent AR	8175	8175	8175	0.0302	4.18	15.49
2/2	Shared token	7921	7921	7921	0.0264	4.18	15.20
2/2	Independent star	5801	5633	5633	0.0233	6.10	14.00
2/2	TV-tree	5310	5173	5173	0.0249	6.84	14.09
3/1	Independent AR	8172	8172	8172	0.0311	4.19	15.53
3/1	Shared token	7992	7992	7992	0.0266	4.20	15.35
3/1	Independent star	5638	5441	5441	0.0224	6.02	13.69
3/1	TV-tree	5144	5002	5002	0.0238	6.93	13.97
3/2	Independent AR	8171	8171	8171	0.0307	4.28	15.61
3/2	Shared token	7987	7986	7986	0.0266	4.19	15.33
3/2	Independent star	5731	5539	5539	0.0231	6.37	14.14
3/2	TV-tree	5135	4965	4965	0.0231	6.88	13.87

D.3 Nearby and far-end monitoring

All 144 test-monitoring cells (72 banks $\times$ the base–10 and base–250 targets) are retained, positive, negative and inconclusive alike. One resampling protocol applies to all of them: the original 64 test prompts with their source counts (32 ELI5, 26 WildChat, six No Robots) held fixed, 5,000 within-source prompt resamples with seed 20260914, the same prompt resamples reused across banks, and independently generated samplers never treated as paired root draws. Training seeds and generation repetitions stay separate and are never pooled into an interval. For each fixed-budget report, an interval above zero is labelled "+" (evidence of improvement), one below zero "−" (evidence of decline) and one crossing zero "?" (inconclusive). These labels are exploratory: they condition on the collected generations, they are not simultaneous across the budget grid, they are not calibrated real-model coverage, and neither counting them nor choosing a favourable budget estimates a training success rate. An earlier 2,000-resample unstratified analysis of the test-prompt means is a separate, historical procedure reported in the interpretation block of Appendix D.2.

D.3.1 Nearby monitoring: savings and resolution together

This reanalysis fixes the base–step-10 contrast and follows the monitoring protocol of Appendix D.3. The six star cells are shown with the independent reference at 64 draws per policy in the base–10, $n = 16$ panel of Figure 19; that reference is itself estimated from a finite sample and is not a known truth.

Four of the six star intervals lie above zero and two cross zero; all three independent-reference intervals cross zero. These six Star cells represent two generation repetitions for each of three training

seeds; the independent references are also finite-sample estimates. CAFA gives the same estimates and intervals as full scoring; its contribution is the feedback avoided in reaching that identical report.

Test-set monitoring at every recorded budget. The following figures retain both the nearby and the far-end contrast at every recorded budget, under the protocol and interval labels of Appendix D.3. The acquisition heatmaps show the corresponding 128-prompt reductions; exact costs are retained in their CSV files. The separate CSV also reports test-only support and score fees, reference uncertainty, and paired-prompt bootstrap differences from the reference; it does not infer half of the generation charge from the test split.

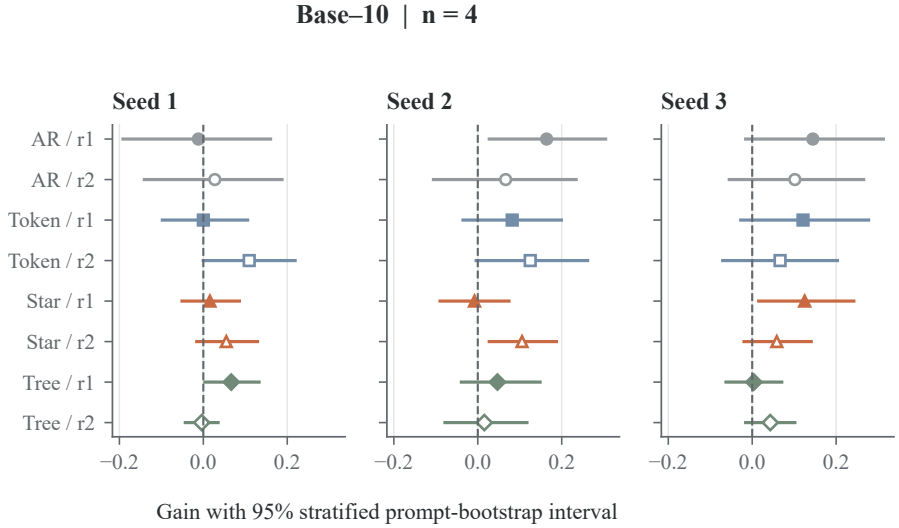


Figure 17: **Monitoring across all recorded budgets.** Each panel set contains all 24 sampler–seed–repeat cells for the stated target and budget. Points are gains on the 64 test prompts; intervals are the original 95% source-stratified prompt-bootstrap intervals (5,000 resamples). Open markers indicate generation repeat 2. The zero line separates positive and negative gains; crossing it is inconclusive. Fixed-prompt MCSE and all support/fee fields remain separately available in the CSV. These exploratory intervals are not simultaneous or anytime-valid guarantees.

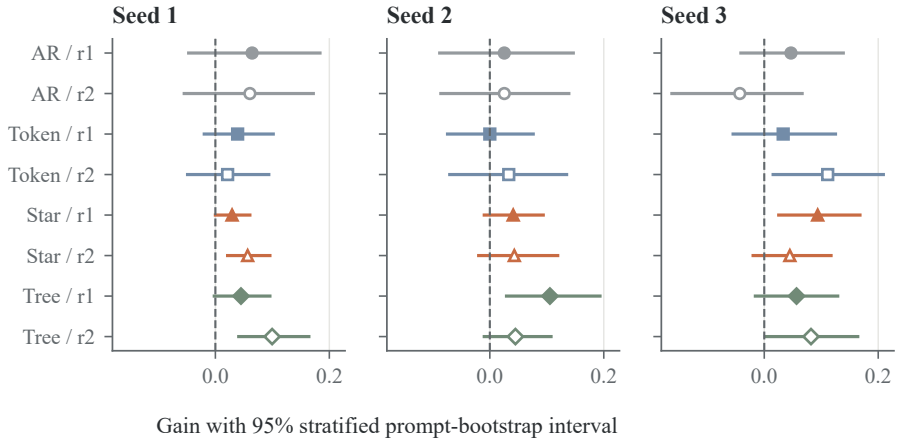


Figure 18: **Monitoring: base–10,** $n = 8$**.** Continuation of Figure 17, with the same definitions and preserved per-cell values.

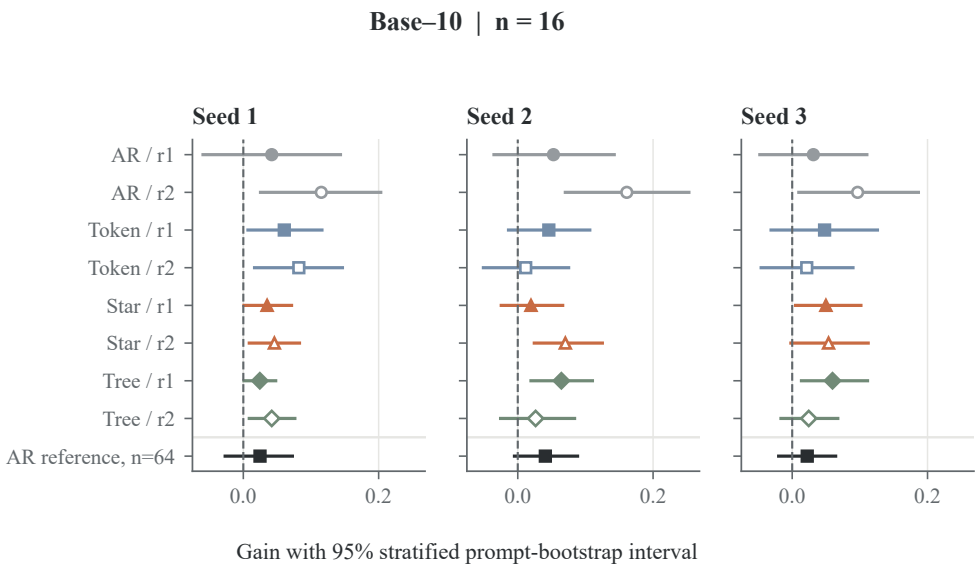


Figure 19: **Monitoring: base–10, $n = 16$, with the independent reference.** Continuation of Figure 17, with the same definitions and preserved per-cell values. Below the horizontal rule, each panel includes the independent autoregressive reference for that seed, using 64 draws per policy on the same 64 test prompts with the same frozen judge. The reference is estimated from a finite sample, is separately uncertain, and is not a known truth; its own interval is in Table 12.

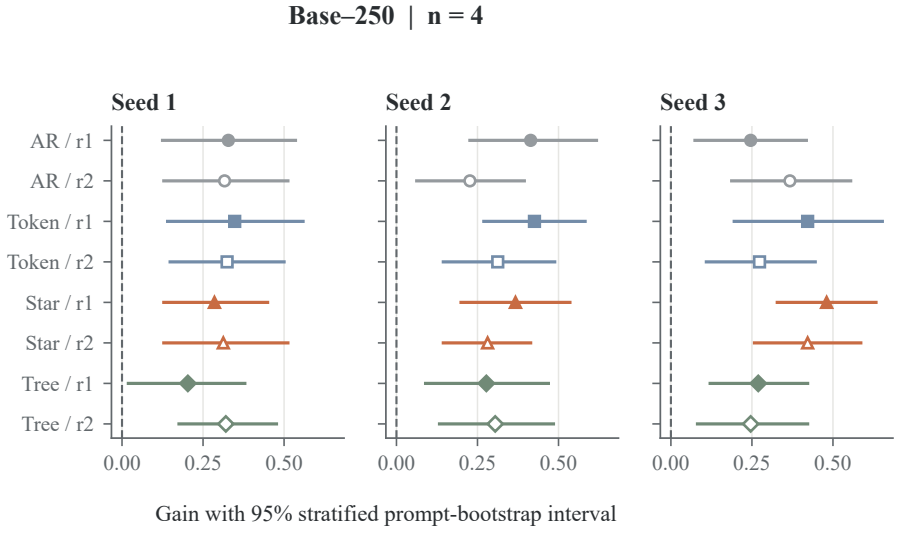


Figure 20: **Monitoring: base–250, $n = 4$.** First of the three far-end panels; continuation of Figure 17, with the same definitions and preserved per-cell values.

D.4 Independent references and checkpoint selection

Checkpoints are selected using 64 selection prompts and evaluated on 64 separate test prompts. Three independent-AR reference banks, one per training seed with 64 draws per test prompt and policy (49,152 answers in total), estimate test quality (Tables 11 and 12). Of the 72 fixed selectors, 71 agree with the estimated reference maximizer. The exception occurs for independent AR with seed 2, eight draws and generation repetition 2. It selects step 50, with estimated regret 0.151 and fixed-pair MCSE 0.020. CAFA preserves every selection made from the original bank; it does not eliminate

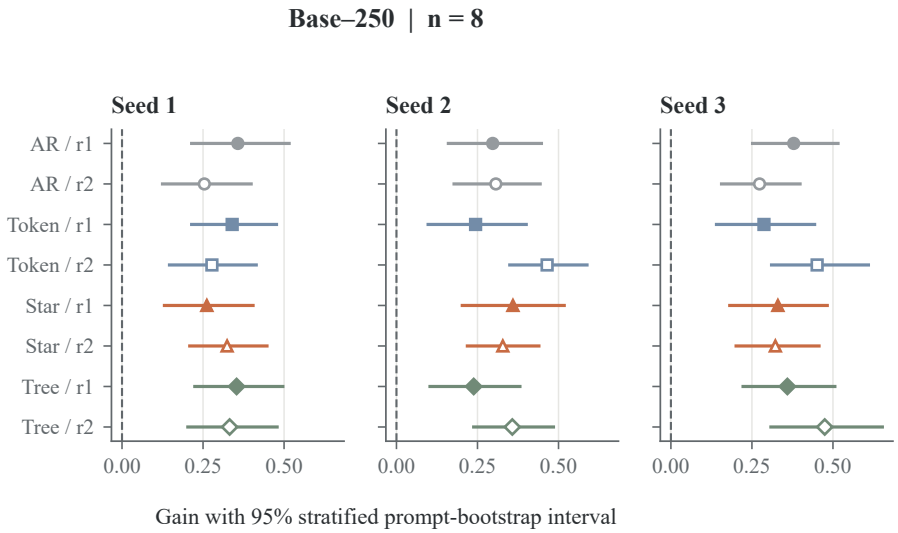


Figure 21: **Monitoring: base–250,** $n = 8$**.** Continuation of Figure 17, with the same definitions and preserved per-cell values.

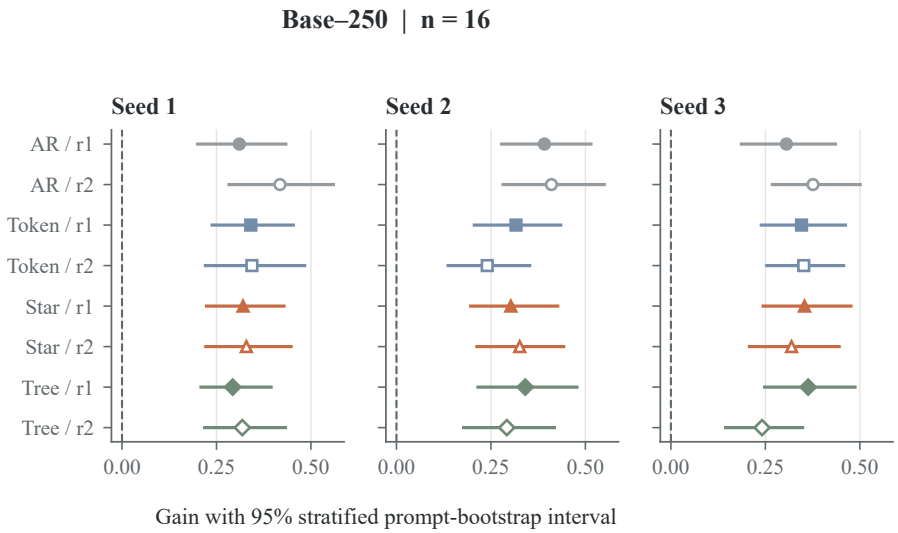


Figure 22: **Monitoring: base–250,** $n = 16$**.** Continuation of Figure 17, with the same definitions and preserved per-cell values.

sampling error relative to an independently estimated reference. The reference maximizer is not a known population optimum, and its uncertainty remains part of the interpretation. Deployment accounting charges the complete 128-prompt workflow rather than inferring half its generation cost from the selection subset. Pairwise reference MCSE conditions on the estimated maximizer.

Table 11: Independent test-reference quality (mean $\pm$ Monte Carlo standard error), by training seed. Training seeds and generation repetitions are not pooled. The base samples use independent reference namespaces across seeds.

Training seed	Base	Step 10	Step 50	Step 250
1	3.034 ± 0.016	3.058 ± 0.015	3.200 ± 0.014	3.350 ± 0.014
2	3.021 ± 0.016	3.061 ± 0.015	3.219 ± 0.014	3.371 ± 0.014
3	3.046 ± 0.015	3.068 ± 0.015	3.193 ± 0.014	3.367 ± 0.014

Complete per-seed and per-repetition counts are retained in Table 10.

Table 12: Independent references under the same 5,000-resample stratified test protocol; 64 draws per policy and test prompt. This reanalysis is separately labeled from earlier unstratified exploratory intervals.

Seed	Target	Gain	MCSE	95% interval
1	Base–10	0.0247	0.0216	[-0.0269, 0.0725]
1	Base–250	0.3167	0.0210	[0.2285, 0.4160]
2	Base–10	0.0408	0.0216	[-0.0049, 0.0884]
2	Base–250	0.3501	0.0210	[0.2554, 0.4573]
3	Base–10	0.0222	0.0214	[-0.0200, 0.0642]
3	Base–250	0.3206	0.0207	[0.2427, 0.4102]

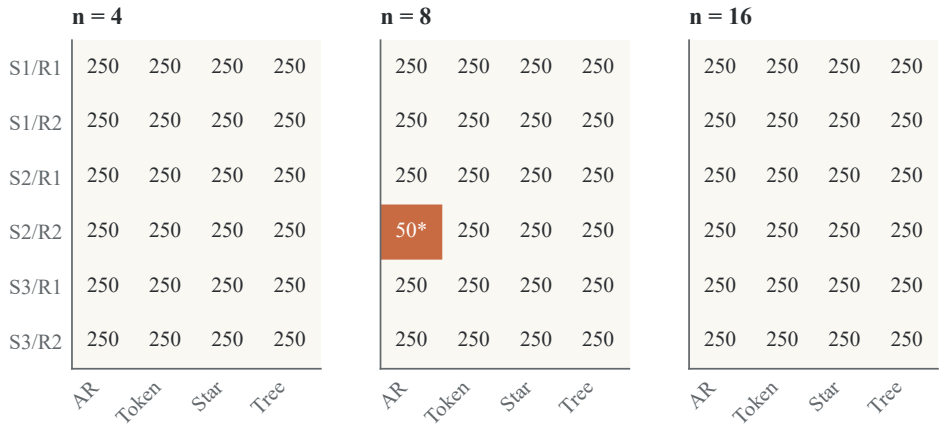
D.5 Report preservation

Every one of the 504 report cells reproduces the fully scored mean and Monte Carlo covariance from the acquired support, and all 72 fixed checkpoint selections are unchanged. Mean and draw supports coincide in 485 cells; the other 19 (13 independent-AR and three shared-token cells for base–10, two independent-AR cells for 10–50, one independent-AR cell for base–50) need extra scores for the covariance. For example, independent AR with seed 1, repetition 2 and $n = 16$ requires 4065 scores for the base–10 mean and 4071 to recover both the mean and ordinary covariance of that contrast. The three-gain report requires 8166 scores in either reporting mode.

Independent recomputation. For this paper the 504 cells were recomputed a second time by a separate implementation of Algorithm 1 written without reference to the analysis code. For each of the 72 frozen banks, the implementation verified the SHA-256 hash of the bank file against the recorded hash, planned every target without reading scores and acquired cached scores through a counting scorer. It then compared the recovered statistics with the published results and an independent dense computation. All 504 cells matched exactly in cached, mean and mean-plus-covariance support counts, and scorer-call counts equalled the required counts in every cell. Mean differences were exactly 0, and covariance differences were at most 4.3×10^{-19}. The full covariance matrices and fixed selections follow.

Selection and covariance are different operations. The figure below retains all 72 fixed selection decisions and their estimated independent-reference regret; a selected reference maximizer has zero empirical regret by construction, the one nonzero regret retains its original fixed-pair MCSE of 0.020, and no pooled coverage or selection-error guarantee is inferred from these cells. The joint-gain Monte Carlo covariance of every bank is a separate matter: CAFA preserves it exactly within a bank, a different sampler can change both it and the selected checkpoint, and the complete per-bank matrices at every budget are retained in the supplementary canonical tables.

71 / 72 agree with the estimated reference maximizer



* Regret 0.151; fixed-pair MCSE 0.020. Full vs. CAFA selection: 72 / 72 identical.

Figure 23: **Checkpoint selection on every frozen bank.** The matrix gives the saved checkpoint step selected in each of the 72 banks and highlights the sole disagreement with the estimated independent reference, whose estimated regret is 0.151 with fixed-pair MCSE 0.020: 71 of 72 selectors agree with that reference, and full scoring and CAFA make identical selections in all 72. Regret is measured against an estimated maximizer, not a known optimum.

E Report targets and generation configurations

E.1 Single, joint, adjacent and anchored reports

The same frozen bank can support different reports, but the support must be recomputed for each report. Required counts cannot be inferred by dividing a joint count by the number of outputs or by summing counts across reports. Table 13 gives all seven online-GRPO outputs over the six independent-star $n = 16$ banks; their per-cell counts and six-cell ranges are in the supplementary canonical tables, and every GRPO target is visualized cell by cell in Appendix D.2. Adjacent-checkpoint contrasts reuse the original joint samples; a separately generated adjacent-pair coupling would be a different experiment. Adding the absolute mean of the base model to the gains changes the report and restores the complete cached support (Corollary 1); it is not a change of coordinates. Averaged over all four independent-star repetitions on the strict holdout, the required input count increases from 1193.25 to 2700.00 for Qwen and from 212.75 to 2157.50 for SmolLM2.

E.1.1 Simple skipping, full aggregation, and additional outputs

The CAFA interface exposes three stages: cache equal scoring inputs; skip zero differences within each draw and cache the remaining inputs; aggregate coefficients across the complete batch. Whether the last two coincide on the completed strict-holdout and distance banks is measured in Appendix C.3, and Proposition 3 supplies the structural explanation for the star construction while that audit checks decoded identities.

All-gains plus an anchor identifies every policy mean, so the return to the cached support noted above follows from the requested target. Source-specific gain reporting continues to use zero-sum contrasts within each source, so requesting a per-source breakdown does not itself reintroduce an anchor.

Table 14 compares base-relative and adjacent-checkpoint targets using the same generated bank. An adjacent contrast can drop inputs needed only by the base or another endpoint. The result measures the value of requesting a different output from existing samples; a newly generated adjacent-pair coupling would be a separate experiment. The machine-readable results include both output caps, all available trajectories, separate mean and variance supports, and both cached judges.

Table 13: **Same banks, different reports.** Online GRPO, Independent Star, 16 draws per prompt; counts are means over three seeds × two generation repetitions, and percentages are cellwise ranges. Q_{cache} counts deduplicated inputs; Q_{mean} and $Q_{\text{mean+cov}}$ count scores needed for means and means with covariance, respectively, and coincide on these banks. "Removed" is $1 - Q_{\text{mean+cov}}/Q_{\text{cache}}$. Cost savings compare CAFA with Cache-only while retaining the full four-policy generation charge.

Report	Q_{cache}	Q_{mean}=$Q_{\text{mean+cov}}$	Removed (%)	Cost saved (%)
Single gains				
Base–10	2,413.3	747.0	59.7–79.6	20.86–26.96
Base–50	3,431.8	2,777.5	14.5–23.3	6.96–10.68
Base–250	3,844.0	3,601.7	5.6–6.8	2.49–2.92
Adjacent gains				
10–50	3,449.8	2,809.0	14.2–22.6	6.79–10.35
50–250	3,906.7	3,717.7	4.3–5.3	1.87–2.40
Joint reports				
Three gains	5,611.5	5,424.3	2.9–3.8	1.54–2.07
Gains + base mean	5,611.5	5,611.5	**0.0**	**0.00**

Table 14: Reusing the frozen distance bank for different requested outputs and output caps. Qwen 1.5B is seed 202609084; Qwen 7B is its single trajectory. All rows use 32 prompts, 64 root draws per prompt and independent-acceptance star sampling, and every row recomputes its own support from the same cached responses for its cap; the individual rows request one contrast each, so their counts must not be added to reach a joint count. Gains use DeepSeek Flash. "–" means no single scalar gain is defined for that report (vector and absolute-anchor rows); "n/r" means the source record for those four 256-token rows reports support only, and no gain is imputed for them. Adjacent differences use the original joint samples, including their covariance.

Model	Cap	Requested output	Cached	Required	Removed (%)	Gain
Qwen 1.5B	768	Base to 2	2223	358	83.9	-0.002
Qwen 1.5B	768	Base to 8	2477	865	65.1	0.025
Qwen 1.5B	768	Base to 32	3363	2636	21.6	0.121
Qwen 1.5B	768	2 to 8	2556	1023	60.0	0.027
Qwen 1.5B	768	8 to 32	3389	2684	20.8	0.096
Qwen 1.5B	768	All gains	3975	3274	17.6	–
Qwen 1.5B	768	Gains + base mean	3975	3975	0.0	–
Qwen 7B	768	Base to 2	2215	334	84.9	0.007
Qwen 7B	768	Base to 8	2620	1144	56.3	0.028
Qwen 7B	768	Base to 32	3935	3774	4.1	0.080
Qwen 7B	768	2 to 8	2701	1306	51.6	0.021
Qwen 7B	768	8 to 32	3939	3782	4.0	0.052
Qwen 7B	768	All gains	4674	4525	3.2	–
Qwen 7B	768	Gains + base mean	4674	4674	0.0	–
Qwen 7B	256	Base to 2	2173	250	88.5	n/r
Qwen 7B	256	Base to 8	2475	854	65.5	n/r
Qwen 7B	256	Base to 32	3777	3458	8.4	n/r
Qwen 7B	256	All gains	4329	4027	7.0	–

E.2 Checkpoints, model configurations and stopping caps

These results use the offline distance regime of Table 2: three saved trajectories per small model, one for Qwen2.5-7B, checkpoints after 2, 8 and 32 steps, and two output caps. Step counts are checkpoint labels, not measured distances, and the 1.5B and 7B configurations are not a controlled model-size comparison. The 7B bank contains an independent-star and a shared-acceptance arm on one trajectory; the figures quoted for 7B in the main text are the independent-star arm. A length cap defines a stopped-response policy: at a 256-token cap, 99.0% of the 7B independent-star endpoint

slots (copies included) stop at the cap, versus 9.7% at 768 tokens, so 768 is the primary configuration and 256 a supplementary one.

E.2.1 Checkpoint and output-cap sensitivity

The prompt sets and their exclusions are in Appendix B.1, the samplers and strong baselines in Appendix B.3, and the cost conventions in Appendix G. We examine how scoring reductions vary with the checkpoint, output cap and model: a distance study over checkpoints at 2, 8 and 32 update steps under 256- and 768-token output limits, and a 7B extension at a larger generation cost. Generation and probability-replay verification are complete for both 7B output limits and for all eight strong-baseline jobs on the comparison bank, and unavailable provider scores propagate through the reported bounds.

Model and output configurations change accessible savings. Table 14 evaluates Qwen2.5-7B with checkpoints after 2, 8 and 32 update steps at both output caps, alongside the 1.5B rows. For the joint target under independent star sampling, CAFA removes 302 of 4329 unique inputs at a 256-token cap and 149 of 4674 at a 768-token cap. Thus large savings for nearby updates do not imply equally large savings for a collection that also includes a more distant update. The contrast target itself, alongside policy overlap and response length, determines the accessible saving. Each output cap defines a stopped-response policy; Figure 25 reports how often endpoints reach that cap, Table 14 carries both caps, and Figure 9 gives the quality change these supports are bought to measure. For the joint target, shared acceptance requires 4057 and 4455 inputs at the two caps, versus 4027 and 4525 for independent acceptance. The sign of this realized increment changes across the two conditions, whereas coefficient cancellation preserves the estimate in either case. This distinguishes a general CAFA rule from a coupling choice whose practical advantage depends on the generated support. The 7B experiment evaluates one saved update trajectory at both response caps, connecting the removable support to quality changes on that trajectory. Training steps index the saved checkpoints rather than measured distribution distances, and the 1.5B and 7B runs use different training configurations.

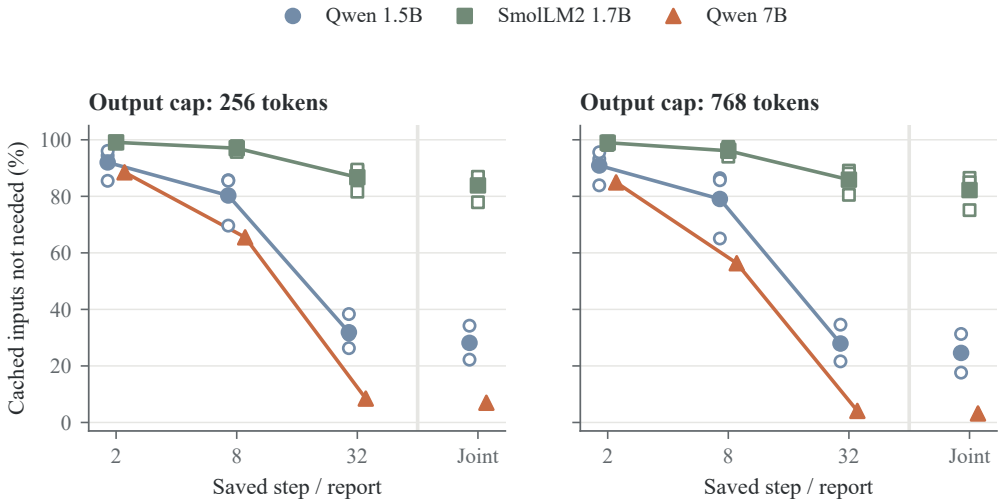


Figure 24: **The requested comparison determines the saving.** Independent Star + CAFA at 64 roots per prompt and 32 prompts. Steps 2, 8 and 32 are checkpoint labels, not measured distribution distances, and request individual base–update gains; "Joint" requests all three gains from the same draws. Hollow points retain individual trajectories and filled points show their means; the 7B extension has one trajectory. Output limits define different generation distributions. Individual query counts must not be added to infer the joint cost.

E.3 Temperature and deterministic generation

Lower temperature is a different generation policy, not a cheaper way to evaluate the same one: greedy decoding has no Monte Carlo repetition, and temperature 0.3 changes both the agreement between

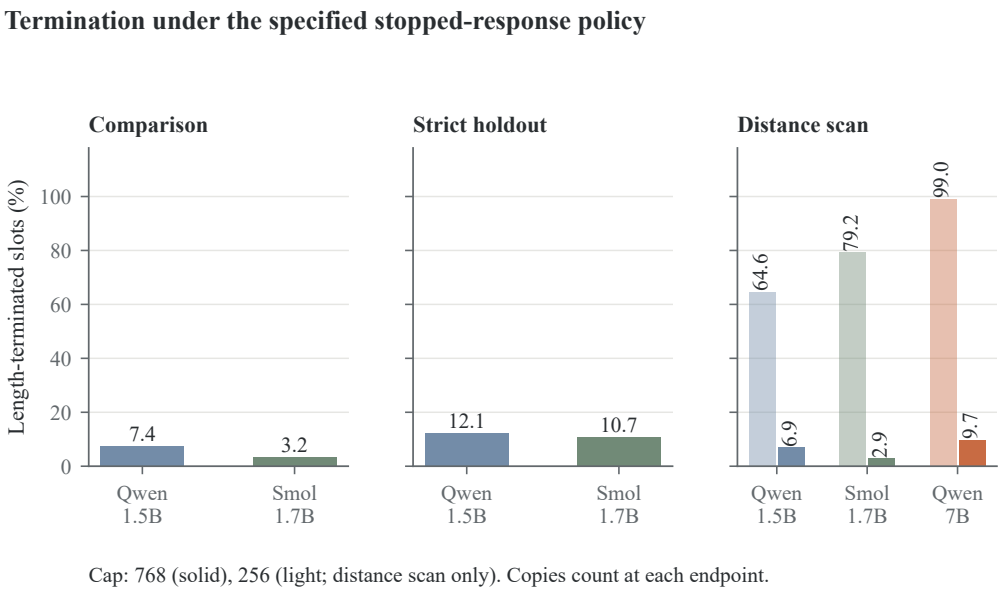


Figure 25: **Termination at the frozen output caps.** The denominator is endpoint policy–root slots, including copied answers at every represented policy. Solid bars use cap 768; light bars use cap 256 in the distance scan. Comparison and holdout banks have four star repetitions; the distance scan has three trajectories per small model and one for Qwen 7B. These are stopped-response policies, not uncapped completions. The original exact displayed cells and source counts are included in the CSV bundle.

policies and the estimand. The study freezes the same 32 strict-holdout prompts and the base plus three Qwen 1.5B updates. At $T = 0.3$, independent AR, shared-token randomness and independent-acceptance star each use 16 draws per prompt and two independent generation repetitions; the $T = 1$ reference reuses the first 16 draws of the first two original repetitions; greedy decoding uses one response per policy and prompt, 128 in total. Every condition keeps the original 768-token cap, EOS, chat template and FP32 execution, and temperature is applied identically in generation and in cross-policy density replay. Costs below are standalone generation runs; the break-even column is a threshold feedback price, not a measured one, and no judge was called for it.

E.3.1 Does ordinary caching suffice at lower temperature?

Even with deterministic decoding, cache sharing and contrast cancellation are distinct operations: the greedy bank has 80 cached inputs and needs 77 scores for the requested gains. Greedy policies have a unique joint response vector, so changing the coupling cannot improve their overlap; contrast cancellation can still remove scores. Lower-temperature comparisons therefore measure both the sharing achieved by ordinary generation and the extra queries removed by the target. Each temperature defines its own response policy; the table measures CAFA requirements for that policy without needing a judge score.

Table 15: New $T = 0.3$ standalone sampler runs, CNY per repetition at 3 CNY per card-hour. Each arm runs in a fresh process with its own residual streams, including loading, generation, probability replay and saving. Thresholds compare with independent AR plus caching; actual judge calls are zero.

Sampler	Generation	ΔG	ΔL	CNY/1k labels
Shared tokens	0.916	0.011	817.0	0.013
Ind. AR	0.905	0.000	0.0	–
Ind. star	1.395	0.490	1535.0	0.319

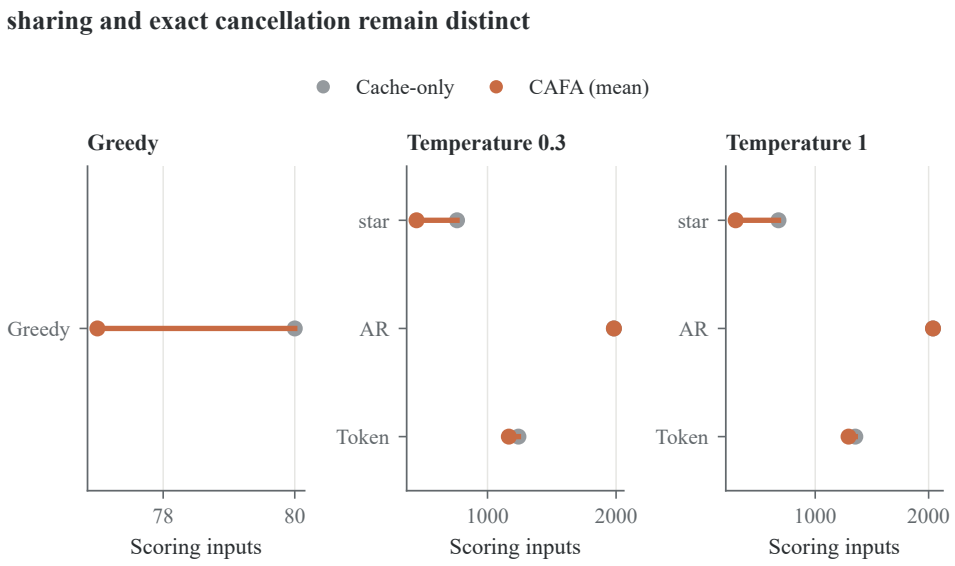


Figure 26: **Cache sharing and cancellation at different temperatures.** Gray points show cached inputs; orange points show mean support. Randomized policies use 16 draws per prompt and two repetitions; greedy uses one deterministic draw and has no Monte Carlo replication. Each temperature defines a different response policy. Axes are separate so that the deterministic case remains readable. At $T = 1$, independent AR needs 2038.5 inputs for the mean versus 2039 for the mean with covariance; both supports and the complete original columns remain in CSV.

The supplementary result package reports both repetitions, each individual gain, response lengths, truncation, proposal counts and per-prompt support. The protocol and technical canary were frozen before the new outputs. Neither query savings nor judge values were used to select prompts, checkpoints or a stopping time.

F Scoring and Statistical Inference

F.1 Versioned Judge Sensitivity

This paper reports four judge evaluations with different coverage, which are analyzed separately. (i) Every main-text GRPO result uses one version-pinned DeepSeek Flash judge, whose returned fingerprint was checked on every call; there is no multi-judge re-scoring of the online study. (ii) The controlled studies score the same responses with DeepSeek Flash and Qwen Flash in separate caches, covering all 32 strict-holdout prompts. (iii) New Qwen Max calls are allocated to a fixed stratified subset of four prompts per bank, which is an audit, not full-bank coverage. (iv) A historical version-matched three-judge comparison is retained under its original identities, on 31–32 matched prompts per cell. The current paired comparison and the historical broad-coverage one differ in coverage and are reported separately. A separate agreement check compares Flash scores with previously collected Max scores: of 500 previously Max-rated inputs, 499 returned valid Flash scores, with Spearman correlation 0.937 and mean absolute difference 0.339 on the ten-point scale. That is an operational agreement screen. None of these checks validates agreement with human preference, and a pinned model name cannot rule out a silent backend change where no fingerprint is returned.

F.1.1 Sensitivity to the Scoring Function

The current strict-holdout comparison is Figure 27. The diagnostics below instead refer to the historical version-matched analysis of scope (iv) above, whose incomplete cells are excluded from

both judges with no score imputation; reconstructing its per-sample differences reproduces all 28,704 available versioned per-prompt means exactly.

For each fixed checkpoint, we compute the variance of the paired judge difference directly. Across 1,152 checkpoint–sampler–budget–repetition–judge-pair cells, the median ratio of this standard error to the value obtained under independent judge estimates is 0.451. These cells share data and are not independent replications. At $n = 64$, median within-prompt correlations on non-copy samples range from 0.858 to 0.895 across judge pairs. Identical base and update responses contribute zero under every judge; excluding them from this correlation diagnostic avoids crediting the judges for algebraically forced agreement.

Figure 28 shows the paired gains. Of 288 plotted cells, 30 have at least one gain of magnitude at least 0.05 score points; none of these has opposite signs. Across all four budgets, the corresponding count is 131 of 1,152. The 0.05 band is a descriptive threshold for summarizing sign agreement. Most gains are small, and this check does not establish that near-tied updates are correctly ordered.

Uncertainty depends on the target population. On the independent-acceptance star arm at $n = 64$, 13 of 36 fixed-prompt repetition-bootstrap intervals exclude zero, compared with 5 of 36 source-stratified prompt-bootstrap intervals. A crossed resampling of prompts and generation repetitions gives 0 of 36 (Appendix F.1.2). We report all three as sensitivity analyses.

Versioned feedback and strong-judge audit. New full evaluations run with thinking disabled and separate caches per judge. Version identity is the returned model name together with the system fingerprint where one is returned; historical aliases retain their original identities and are never renamed retrospectively, which is what keeps the historical analysis of Appendix F.1.1 separable from the current one.

For shared-star sampling at $n = 64$ on the strict holdout, the largest absolute Flash–Flash difference across the three fixed update gains is 0.0089 score points on Qwen and 0.0052 on SmolLM2. The largest Flash–Max difference on the four fixed audit prompts is 0.0498. These numerical gaps are small on the ten-point scale, but some prompt-bootstrap intervals exclude zero and the comparison does not establish judge equivalence. Judge scores affect estimated gains and precision; the acquired-input counts depend only on the frozen responses and contrast coefficients.

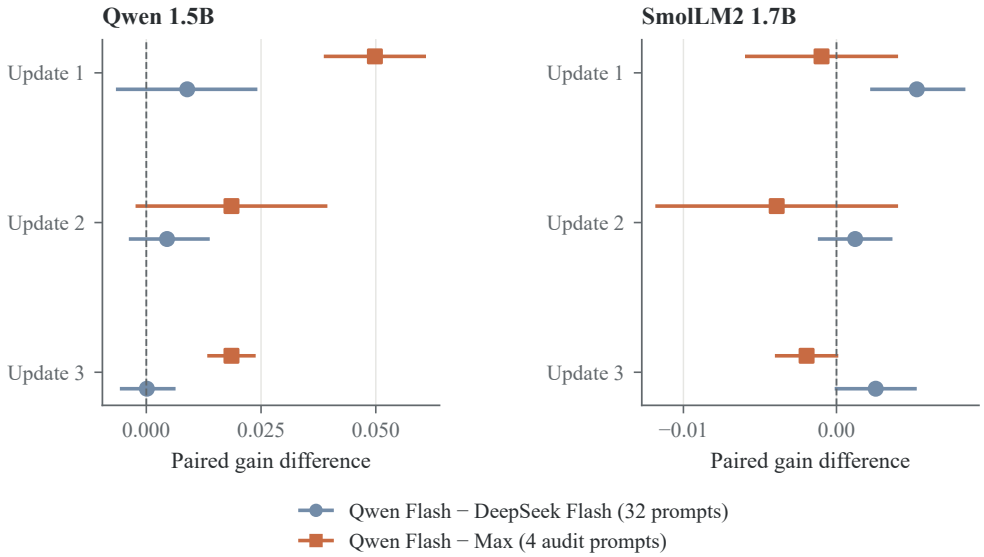


Figure 27: **Judge comparisons preserve input pairing.** Shared-star strict holdout, $n = 64$, four generation repetitions. Flash–Flash comparisons use all 32 prompts; Flash–Max uses the four fixed audit prompts. Points show differences between the gain estimates from the two judges on identical inputs. Bars are 95% prompt-bootstrap intervals within the frozen sampling strata, conditional on collected generations. The small subset audited with Max does not provide full-bank coverage. An interval containing zero does not establish equivalence.

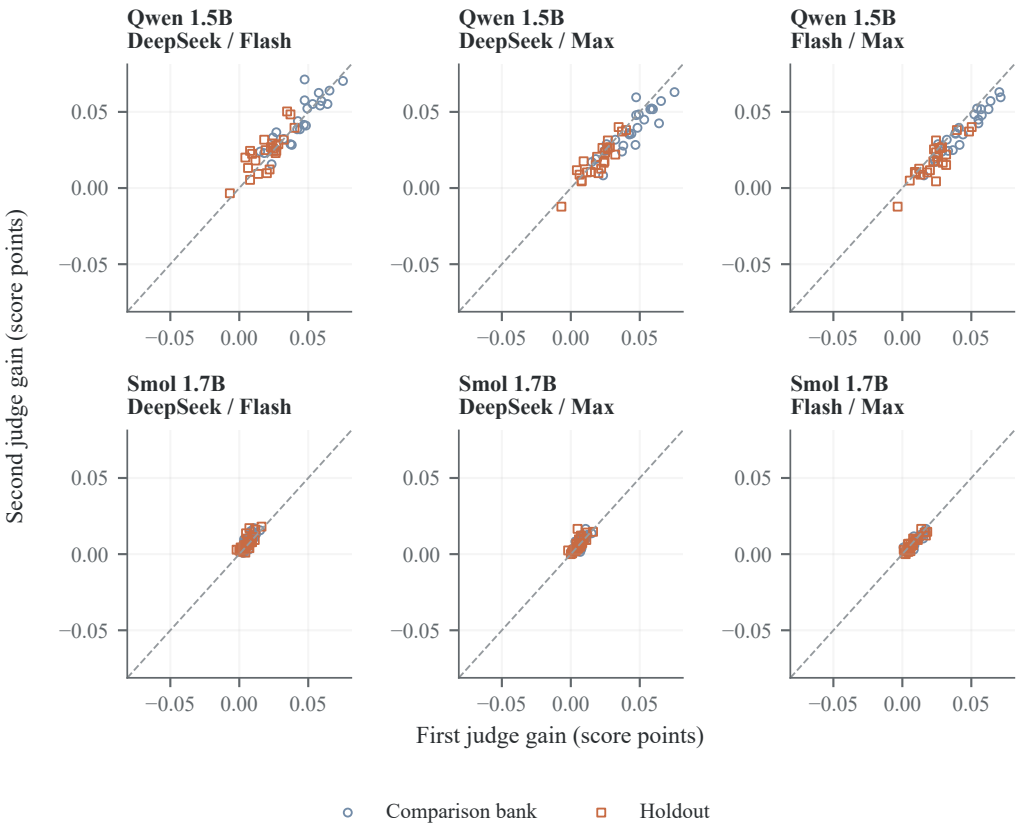


Figure 28: Paired update gains under three scoring functions at $n = 64$. Each marker denotes a fixed checkpoint, sampler, generation repetition, and judge pair, averaged over their common complete prompts. The dashed line is equality. Columns separate judge pairs; rows separate model families. Blue and orange distinguish the two prompt banks. The two methods and four budgets reuse generated data.

F.1.2 Judge comparison units and coverage

All judge comparisons condition on the saved policy endpoints. With $D_{it}^{(a)}$ and $D_{it}^{(b)}$ denoting same-sample base–update score differences, the within-prompt variance estimate is $s^2(D^{(b)} - D^{(a)})/n$, including covariance. Prompt averages use equal weights over the common complete cells. Version identity includes the returned model and, when available, the system fingerprint; absent Qwen fingerprints leave model-name pinning weaker than full backend pinning.

The independent-acceptance star analysis at $n = 64$ contains 36 population-comparison cells: two prompt sets, two model families, three fixed checkpoints, and three judge pairs. Each uses four generation repetitions and 31–32 common prompts. Ten thousand bootstrap draws use the same selected indices for both judges. The fixed-prompt scheme resamples repetition rows. The prompt scheme averages repetitions first and resamples prompt columns within source strata. The combined scheme independently resamples the row and stratified column indices; it is a crossed row–column bootstrap, not a nested bootstrap over new training runs. These are descriptive uncertainty checks for different conditioning choices.

The strong-judge cache covers substantially more than the new four-prompt audit allocation, but is not literally complete: six stage–job–budget–judge-pair combinations lose one checkpoint–sampler–prompt cell (191 of 192 retained). The missing cell is removed symmetrically from both judges. Source snapshots and extraction manifests retain those exclusions. Comparisons against tree sampling use matched prompts, but independently generated streams; assigning the same repetition number to two runs does not create common-random-number pairing.

F.2 KNOWN-TRUTH FINITE-PROXY ERROR

To measure error against known means, this study defines finite-support proxy policies over 32 stored responses for each of 16 prompts, with two historical fixed scoring functions. The base is uniform on the response indices, endpoint probabilities are proportional to the stored likelihood ratios, and every cached score is known, so the expected scores are exact. Five samplers—independent categorical, shared categorical Gumbel, independent star, shared acceptance and a total-variation spanning tree—receive the same aggregation and deduplication over 2,000 Monte Carlo repetitions at 8 to 128 draws per prompt, alongside a full-pool exact evaluation that removes Monte Carlo error entirely and makes finite-cache saturation visible. The proxies are not the unrestricted language models; exactness at the full pool is exactness for a finite constructed distribution, not for any finite budget on a real model.

F.2.1 DOES THE REPORTED ANSWER CHANGE?

Shared acceptance has overlapping empirical MSE ranges with independent star coupling and uses fewer required labels at the largest sampled budget (Figure 29); the tree shows a different cost–error tradeoff. Ranking error can stay large when true means are nearly tied even where MSE is small, so we report gain estimation first and selection regret as secondary.

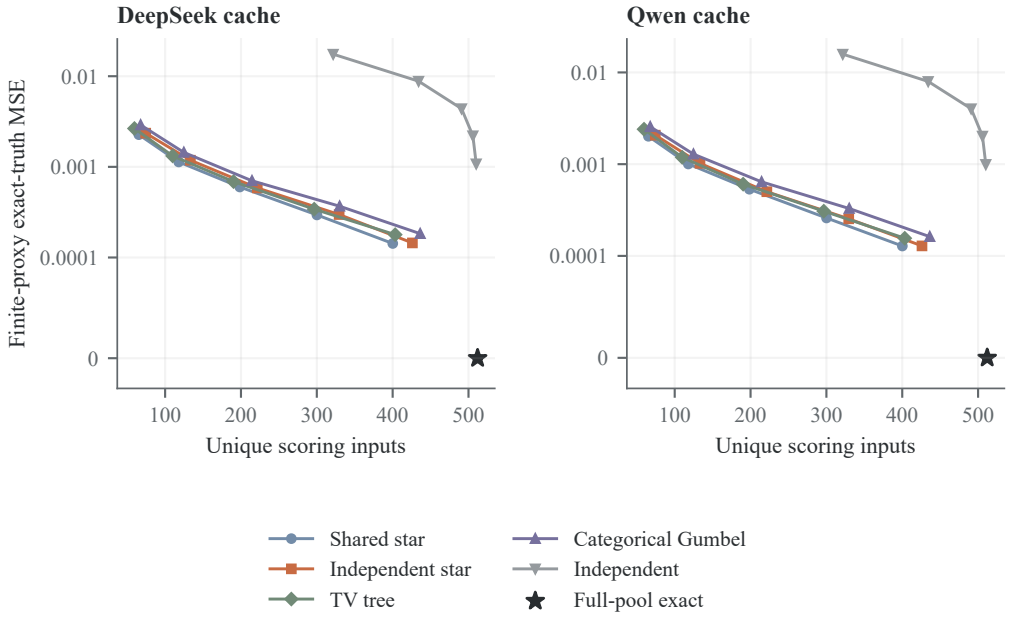


Figure 29: Cost–error diagnostic for finite-support proxy policies on 16 prompts with 32 stored responses each, using two historical fixed scoring functions. Each sampled budget uses 2,000 Monte Carlo repetitions, with 8–128 draws per prompt. The vertical scale is symmetric logarithmic, linear near zero. The exact full-pool point includes all unique support inputs and has zero estimation error. Categorical Gumbel is a finite-policy baseline, not an autoregressive token implementation.

F.3 RARE EVENTS AND INTERVAL DIAGNOSTICS

We conduct synthetic coverage checks with known means for three distinct estimands: a single prompt, a fixed 32-prompt bank and a prompt population with a fixed source mixture. Five fixed conditions are used—a symmetric null (C1), a small nonzero gain (C2), a one-sided gain (C3), correlated updates with unchanged individual marginals (C4), and heterogeneous source means with unequal sampling allocation (C5)—each with 2,000 independent simulation repetitions at the budgets $n = 8, 16, 32, 64$. The all-copy Clopper–Pearson envelope applies to the specified conservative procedure rather than providing a universal sampling bound. These are not calibrated coverage claims for the language models.

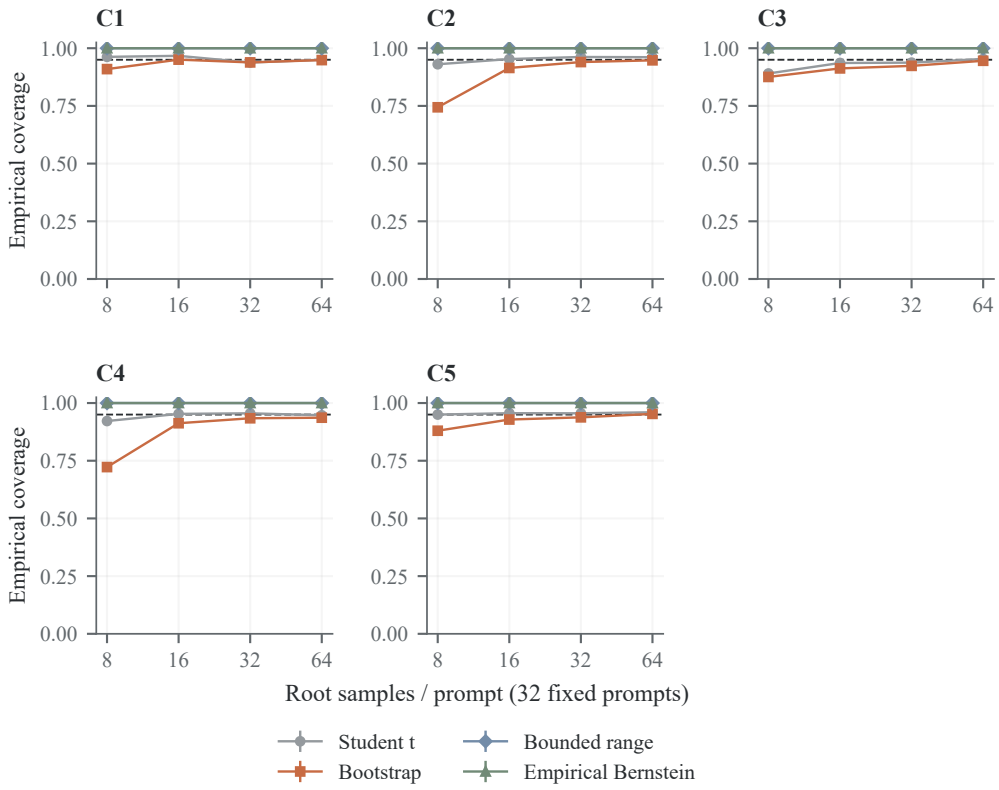


Figure 30: Fixed-prompt marginal coverage in all five finite-support diagnostic conditions. The dashed line is 0.95; error bars show 1.96 simulation standard errors. The range and empirical Bernstein intervals retain a nonzero range term when all observed differences are zero. This figure compares fixed-budget intervals, not anytime-valid confidence sequences. C1–C5 are defined in the text.

F.3.1 Rare-event interval diagnostics

For the fixed-prompt diagnostics, let $D_{i,t}$ be one update's score difference at prompt x_i and draw t, and set $U_t = P^{-1} \sum_{i=1}^{P} D_{i,t}$ with $P = 32$. Student intervals use the standard error $\sqrt{s_U^2/n}$, where s_U^2 is the sample variance of the n draw averages; the bootstrap resamples these U_t values. The bounded-range and empirical Bernstein procedures also use the n draw averages. With independent prompt streams, s_U^2/n and the stratified estimator in Section 3.3 estimate the same variance of the fixed-prompt mean; their finite-sample values can differ.

The implementation also records interval width, all-zero sample frequency, missed-tail direction, and simultaneous coverage across the three updates. Three unadjusted marginal 95% intervals do not promise joint 95% coverage. The ordinary bootstrap and Student intervals can collapse to $[0, 0]$ on an all-zero sample, whereas bounded-range and empirical Bernstein procedures retain uncertainty (Maurer & Pontil, 2009). This is why an empirical Monte Carlo standard error of zero is reported as a sample statistic, not as proof of a zero population variance. The full diagnostic data include the single-prompt and population settings and preserve their distinct estimands. Results from confidence sequences are retained separately with their stronger time-uniform validity requirement (Howard et al., 2021; Waudby-Smith & Ramdas, 2024).

What an all-copy cell can certify. For one fixed prompt and update, let d be the probability that the two decoded responses differ. Since the score difference is in $[-9, 9]$, $|\theta(x)| \leq 9d$. With no disagreements in n independent draws, the upper endpoint of a two-sided 95% Clopper–Pearson interval is $d_U = 1 - 0.025^{1/n}$ (Clopper & Pearson, 1934). Thus this envelope is $|\theta(x)| \leq 9(1 - 0.025^{1/n})$, approximately 0.504 at $n = 64$; reaching 0.05 with this particular envelope requires

663 all-copy draws. These are properties of a conservative confidence procedure, not a universal sample-complexity lower bound or a requirement for the bank-level estimate. An all-zero sample makes an ordinary bootstrap or Student interval collapse to $[0, 0]$: it misses a nonzero target but contains a target equal to zero. The conditional coverage results from these simulations therefore do not imply zero coverage for every underlying policy or establish coverage for real models.

F.4 Missing feedback and resampling units

An unavailable score is never recorded as 0. In the primary analysis, each unavailable required score contributes its admissible range through its signed, aggregated coefficient. Each unknown identity is therefore bounded once, rather than once per occurrence. A covariance point estimate is withheld whenever a required draw score is missing. Three uncertainty units are distinguished throughout: fixed-prompt Monte Carlo error (MCSE, conditional on the prompts and cache), prompt-resampling bootstrap intervals (within sampling strata), and variation across training seeds, which is described by ranges and never pooled into a confidence interval. Historical intersection analyses that drop prompts with incomplete feedback are labelled as such and do not replace the complete primary analysis.

F.4.1 Unified evidence and inference contract

Scope of this contract. The evaluated model families, prompt sources and sampling strata are specified in Appendix B.1, and the frozen-snapshot procedure that produces the analysis inputs in Appendix I.4. These are open-ended generation tasks; no exact-answer rule is used as the primary judge. The models are Qwen2.5 (Qwen Team, 2024) and SmolLM2 (Ben Allal et al., 2025); generation uses PyTorch and Transformers (Paszke et al., 2019; Wolf et al., 2020), and the statistical pipeline and vector figures use NumPy and Matplotlib (Harris et al., 2020; Hunter, 2007).

Unavailable scores. Write a requested linear statistic as $\sum_u b_u r(u)$ and let M be the unavailable scoring identities. With $r(u) \in [1, 10]$, the exact identified interval conditional on acquired scores is

$$\left[\sum_{u\notin M} b_u r(u) + \sum_{u\in M} \min\{b_u, 10b_u\}, \quad \sum_{u\notin M} b_u r(u) + \sum_{u\in M} \max\{b_u, 10b_u\}\right].$$

Coefficients of a shared missing identity are combined before computing this interval. A platform refusal is a terminal observation about accessibility, not a quality score. The missing input is retained in the support count. Pending calls keep a snapshot interim; terminal refusals allow completion with explicitly bounded results.

Mean and variance support. The two supports are defined in Section 3.2 and Appendix A.2; the pipeline reports $|S_{\text{draw}} \setminus S_{\text{mean}}|$ explicitly. If any required draw score is unavailable, it withholds a point variance rather than using an uncharged or imputed value. For a contrast in $[-9, 9]$ and n draws, $s^2/n \le 81/(n-1)$ supplies a conservative upper bound. An observed zero variance is separately flagged and never treated as a population-variance certificate.

Two uncertainty targets. For a fixed bank of P prompts and R independent generation repetitions, each single-update gain averages PRn bounded differences. The estimated fixed-prompt Monte Carlo variance is the sum of prompt–repetition variance-of-mean estimates divided by $(PR)^2$. Assuming independent root draws under the frozen sampler and a fixed score cache, a two-sided Hoeffding interval expands the identified mean interval $[L, U]$ to

$$[L - \varepsilon,\ U + \varepsilon], \qquad \varepsilon = 18\sqrt{\frac{\log(2/\alpha)}{2PRn}}.$$

This remains nondegenerate when every observed pair copies. It is a fixed-budget bound, not an optional-stopping guarantee. This is the standard bounded-difference tail bound; the terms need not be identically distributed, but they must satisfy the independence stated above. For a variance-sensitive alternative that retains a nonzero range term, see Maurer & Pontil (2009).

For generalization over prompts, the pipeline instead resamples prompts within each frozen stratum, keeping the same prompt resamples across methods and judges. It averages collected generation repetitions within each sampled prompt. These percentile intervals describe prompt heterogeneity

conditional on collected generations; they do not incorporate a new distribution of training trajectories. Independently generated samplers are not paired by root or repetition index. The descriptive three-trajectory distance scan and single-trajectory 7B extension are kept separate from fixed-policy generation replication.

For an independently sampled prompt population, a prompt-level mean and its across-prompt variance estimate only need the per-prompt means. Within each independently sampled stratum, total variance gives

$$\mathrm{Var}(\bar{\theta}) = P^{-1}\{\mathrm{Var}_x[\theta(x)] + \mathbb{E}_x \mathrm{Var}[\widehat{\theta}(x) \mid x]\}.$$

Fixed source allocations require stratum weights and within-stratum variance estimates; the pooled iid formula is not applied to the fixed mixture without adjustment. Such intervals concern the prompt population, whereas resampling independent generation repetitions concerns Monte Carlo variation on the fixed prompt set. Repetition-level intervals require the acquisition set for every repetition and incur no additional scoring cost only when those scores have already been acquired.

Paired judges and repeated looks. Judge differences are computed from the same per-draw contrasts before taking a variance, so their observed covariance is retained rather than assuming independent judge estimates; the resampling units and coverage scopes are in Appendices F.1.2 and F.1. Fixed-budget confidence intervals are not interpreted as confidence sequences (Howard et al., 2021; Waudby-Smith & Ramdas, 2024), and failure to detect a difference is not an equivalence claim.

Coverage rather than imputation. The primary versioned analysis retains the complete frozen prompt set, and a required score that is still unavailable after bounded retries contributes its admissible range $[1, 10]$ to the contrast and uncertainty bounds rather than being dropped or imputed. Historical common-coverage analyses instead state their matched intersection explicitly, and some therefore contain 31 prompts; they are a separate analysis object with its own coverage record and never replace the primary one. Supplementary cost-audit calls are recorded in a separate ledger and do not overwrite the primary score cache.

F.5 Extended related work

The main text keeps the most direct comparisons. The complete discussion covers the relation to tree-coupled feedback sharing, coupled rejection sampling, label-efficient evaluation, pairwise judges and uncertainty under repeated reporting. In the paragraph on Wen (2026), "Section 6.1" and "Appendix E.1" refer to sections of that paper.

Coupling and controlled randomness. Section 6 places CAFA against coupling and common-random-number methods: they control the randomness of a comparison, whereas CAFA conditions on the resulting samples and identifies which score queries can change the requested statistic. The shared-acceptance extension in Appendix H.2 applies the classical Doeblin intersection bound (Makur & Singh, 2024) to a component-query objective.

Coupled evaluation and feedback sharing. Benz et al. (2026) couple autoregressive token generation for language-model evaluation; shared token randomness is a direct baseline. Wen (2026) develops tree-coupled feedback sharing with preserved policy trajectories. Their original query rule purchases one reward per matched component, with cost $T + \sum_{t,e} D_{e,t}$. Section 6.1 includes nonadaptive language-model policies. Appendix E.1 extends their variance bound to fixed zero-sum contrasts by expressing each contrast as a sum of pairwise differences. Our additional operation removes a score query when its realized coefficient vector is zero. For an all-matched round this changes one shared query to zero; on the comparison-bank tree samples it removes 59.0% and 90.9% of the already deduplicated inputs. Thus the improvement applies on top of their sampling construction.

The interfaces explain the distinction. TCAB preserves reward-bearing trajectories, including histories that may determine later adaptive actions. Here the policies are frozen and feedback is a versioned scoring function: the requested contrast specifies which scores to acquire, decoded text specifies which scores to share, and the requested uncertainty statistic determines whether within-draw support

must also be purchased. Proposition 3 characterizes when immediate CAFA already reaches the final-batch count. These conclusions for CAFA are distinct from the fixed-contrast variance extension in TCAB.

Coupled rejection sampling and residual design. Corenflos & Särkkä (2022) construct coupled rejection samplers from dominating proposals and study their coupling probability and execution time. Their shared acceptance and independent completion mechanisms supply prior foundations for our residual kernel. Our decision variable is a partition of residual requests, selected from an independent score-free pilot for the generation-plus-acquisition cost of a specified contrast. Proposition 6 records the pair-law preservation needed by this evaluation interface. We evaluate pilot-selected residual partitions by their required scoring support and full recorded cost.

Evaluation with fewer labels or examples. Sawade et al. (2012) directly compare prediction-model risks under a labeling budget, allocating labels according to model disagreement and the power of the comparison test. This shares our motivation to estimate differences directly. Their interface purchases a supervised outcome for a test instance; our interface purchases a score for a realized generated response and first removes terms that are irrelevant for every score function. Active testing (Kossen et al., 2021) and active surrogate estimators (Kossen et al., 2022) further select informative labels while controlling estimation error. Evaluation-set reduction (Maia Polo et al., 2024; Vivek et al., 2024), uncertainty-driven allocation of repeated judge queries (Saha et al., 2026) and broad task suites (Srivastava et al., 2023; Liang et al., 2023) are further budget dimensions that compose with exact support pruning; our paid-pilot experiment measures that composition using identical pilots and surrogates in its controls.

What an automatic score measures. An unrestricted pairwise preference judge evaluates a function $q_x(y, y')$, which need not be representable as $r_x(y') - r_x(y)$ for any scalar scoring function. Its expected preference can depend on the chosen coupling, even with fixed marginals; Benz et al. (2026) explicitly show that pairwise rankings can differ under coupled and independent generation. A pairwise API therefore changes the estimand unless an additional representation assumption is supplied. Treating its input as an unordered pair also requires a symmetry and orientation convention, which a position-sensitive judge need not satisfy. The experiments evaluate scalar-score differences.

Model-based judging is studied with pairwise arenas (Zheng et al., 2023; Chiang et al., 2024), structured criteria (Liu et al., 2023), trained evaluators (Kim et al., 2024a;b; Zhu et al., 2025) and preference-model benchmarks (Lambert et al., 2025). Two boundaries carry over to this paper. A pairwise target is not the scalar-score target, and it needs a position and orientation convention that a position-sensitive judge need not satisfy. Our pointwise guarantee applies to the selected scoring function and does not establish agreement with human judgment. Known length bias (Dubois et al., 2024) and self-preference (Wataoka et al., 2024) motivate the paired cross-judge checks. Diverse smaller judges can offer an attractive alternative to a single large judge (Verga et al., 2024); we retain a fixed strong-judge audit alongside full Flash evaluation.

Uncertainty and repeated reporting. Appendices F.3 and F.4 apply the statistical background this paper relies on: resampling of the relevant experimental unit (Efron, 1979), evaluation design for NLP and few-run reinforcement-learning comparisons (Dror et al., 2018; Agarwal et al., 2021), bounded-variable concentration and exact binomial limits for the rare mismatches that yield zero empirical variance (Maurer & Pontil, 2009; Clopper & Pearson, 1934), and confidence sequences for optional stopping (Howard et al., 2021; Waudby-Smith & Ramdas, 2024). Our budget comparisons use fixed sample counts, and reporting an estimate online is a CAFA reporting property that does not by itself provide time-uniform coverage.

G Computation and monetary accounting

G.1 Cost definitions and measured scopes

Four quantities are kept apart. (i) The necessary support (Q_{cache}, Q_{mean}, $Q_{\mathrm{mean+cov}}$) is a count of scoring identities, fixed by the bank and report before any score is seen. (ii) New paid calls are what one execution buys; with a warm cache they can be zero although the necessary support is not.

(iii) Recorded workflow cost adds generation G to feedback F: GPU time is priced at a common reference tariff of 3 CNY (USD 0.4443) per card-hour rather than a supplier invoice, and feedback at the frozen judge tariff applied to measured token usage (DeepSeek Flash, 2 and 8 CNY per million input and output tokens; Appendix G.2). (iv) Research expenses, including full-cache audit scoring, failed calls, reference generation, training and restarts, are recorded separately from workflow costs. Same-bank comparisons hold G fixed on both sides and change only F; cross-sampler comparisons change both and are labelled as workflow comparisons. Unknown costs are left empty, never set to zero.

G.1.1 Separating acquisition and execution cost

For a specified comparator, write $\Delta L = L_{\text{comparator}} - L_{\text{method}} > 0$ and $\Delta G = G_{\text{method}} - G_{\text{comparator}}$. With comparable average feedback price c, net saving is $c\Delta L - \Delta G$; for unequal response lengths, actual token-based charges replace $c\Delta L$. Cancelling a fixed batch adds only aggregation work and avoids score calls. Introducing an expensive coupling has a different computational baseline. A shared residual pool used to compare two arms experimentally does not by itself identify two separate deployment bills. Job-level timers and isolated accelerator benchmarks must therefore be distinguished from matched end-to-end cost measurements.

G.2 Currency

All costs were recorded in Chinese yuan (CNY). The main text, Table 5 and the redrawn figures display them in US dollars using one fixed rate: the CFETS central parity published by the State Administration of Foreign Exchange for 18 September 2026, 675.21 CNY per 100 USD, that is 6.7521 CNY per USD, retrieved on 19 September 2026. Every displayed amount is $\text{cost}_{\text{USD}} = \text{cost}_{\text{CNY}}/6.7521$, computed from the unrounded CNY record and rounded only for display. This changes the display unit, not the prices: generation keeps the reference tariff and feedback the frozen token tariffs of the experiments. Percentages, counts, standard errors and rankings are unchanged, because a ratio of two amounts does not depend on the unit. The frozen unit prices convert as follows.

Frozen unit price	CNY	USD
Generation, per GPU-hour (reference tariff)	3	0.4443
DeepSeek Flash, per million input / output tokens	2 / 8	0.2962 / 1.1848
Qwen Flash, per million input / output tokens	0.8 / 2.7	0.1185 / 0.3999

With the single exception of Table 5(B), which is in USD, the tables and figures in Appendices C–H keep their original CNY amounts so that they match the frozen records line by line; each states its currency in its caption or column headings. The regenerated floats and a file listing every converted value (source record, CNY value, unrounded USD value, display string and where it is shown) are in the supplementary archive (Appendix I.4).

G.3 Tariff sensitivity and break-even analysis

Holding the measured generation costs and response lengths fixed, scaling the feedback tariff changes which $n = 16$ GRPO workflow is cheapest: at one quarter of the frozen tariff shared tokens are cheapest in all six seed–repetition cells; at the frozen tariff the star is cheapest in four and the tree in two; at twice the tariff the tree is cheapest in all six. These are observed expenditure comparisons at a fixed draw budget and not a precision ranking, so the cost–precision curves of Appendix D.2 retain that second dimension. The separate precision-constrained readout is in Section 5.3. The controlled break-even prices follow; a constant per-label threshold and a token-priced threshold over the full covariance support are different quantities. For an observed-workflow comparison with independent AR plus caching, the condition is

$$G_m - G_{\text{AR}} < \lambda\big(S_{\text{AR,cache}} - S_{m,variance}\big),$$

where S uses the observed token length of each input and λ scales the fixed scoring tariff. Table 16 supplies this threshold together with the constant-price-per-label threshold $\max(0, \Delta G/\Delta L)$. Both are calculated from unrounded records. The complete result package contains both judges and tariff factors $0, 0.25, 0.5, 1, 2, 5, 10$.

Table 16: Break-even feedback prices relative to the recorded independent-AR-plus-cache workflow. ΔG is additional generation CNY and ΔL is the number of mean labels avoided. The tariff factor instead prices the complete variance support using the actual token usage of each response under DeepSeek Flash. The different generation-overhead conventions from Table 9 remain explicit.

Model	Sampler	ΔG	ΔL	CNY/1k labels	Tariff factor
Qwen 1.5B	Shared tokens	0.051	2963.5	0.017	0.016
Qwen 1.5B	TV tree	1.072	7216.0	0.149	0.119
Qwen 1.5B	Ind. star	1.727	6942.5	0.249	0.200
Qwen 1.5B	Shared star	1.727	7132.5	0.242	0.194
Smol 1.7B	Shared tokens	0.024	6342.0	0.004	0.003
Smol 1.7B	TV tree	1.586	8000.5	0.198	0.154
Smol 1.7B	Ind. star	2.320	7989.0	0.290	0.225
Smol 1.7B	Shared star	2.320	8035.5	0.289	0.224

At the fixed DeepSeek tariff, the tree workflow for Qwen costs 9.01 CNY, compared with 16.97 CNY for independent AR plus caching; the corresponding RMS MCSE values are 0.0114 and 0.0308. The independent-star workflow costs 10.06 CNY under the full-joint-task charge. Scoring price can change the preferred recorded workflow: for SmolLM2 under Qwen Flash, shared tokens cost 6.79 CNY while the tree costs 7.43 CNY, with RMS MCSE 0.0119 and 0.0054 respectively. These joint measurements expose the price–precision tradeoff directly.

The same-generation CAFA comparison and the reconstructed cross-sampler comparison answer complementary deployment questions. Their generation terms cannot be cancelled interchangeably. For smaller n in the controlled experiments, we report measured support–MCSE curves but leave generation costs unspecified because complete timings were not collected. Full-budget task intervals do not determine execution times at lower budgets.

G.4 Bookkeeping and residual computation

On a local CPU, score-free support aggregation for the six predecoded star $n = 16$ GRPO banks took 0.081–0.150 seconds, measured as medians of five runs after a warm-up and excluding loading, hashing, scoring and generation. These timings cover only support aggregation and do not measure end-to-end latency. The signed-overhead sensitivity asks how much unrecorded extra cost a method could carry before losing its recorded advantage; zero and negative margins are kept and give no headroom. Partial batch timers cover different scopes (tree timers include only root generation) and do not rank methods. In the idealized untruncated construction without batch overshoot, each child consumes one residual proposal per root draw in expectation when $d = \mathrm{TV}(p_0, p_j) > 0$. Conditional on needing a residual, the expected count is $1/d$, so small positive residual mass produces rare but long requests. Pool-exhaustion probabilities follow the binomial bound in Appendix G.4.1.

Cost identification and unrecorded overhead. The same-bank monetary increment holds recorded generation fixed. A cross-sampler comparison additionally depends on execution history. Let $M = T_{\mathrm{AR,cache}} - T_{m,report}$ be the observed cost margin. If unrecorded method overhead minus unrecorded baseline overhead is δ, the observed ranking survives precisely when $\delta < M$. This signed sensitivity is not an estimate of missing execution time; the seven cells with failure/recovery histories now use uninterrupted full-process replays (Appendix D.2.6). Their historical charges remain separate; the difference between historical and replay charges is not an identified failure-only cost. Hardware and runtime differences remain possible sources of overhead. For the six $n = 16$ star cells, the margin over independent AR plus caching is 1.471–2.250 CNY. An additional net charge equal to the respective margin removes the recorded cost advantage. The complete signed margins for all workflows are retained in the audit. The signed margin of every one of the 72 frozen workflow cells against independent AR plus caching is retained in the supplementary canonical tables, in CNY: a net additional method-minus-baseline charge of that amount yields a tie. Seven margins are negative and eighteen are zero, so those cells offer no positive headroom at all. This is an accounting sensitivity on costs recorded elsewhere.

Local CPU aggregation timing. Python 3.11.14, NumPy 2.4.6, Windows, a single local process without pinned affinity. Six star banks at 16 draws each use predecoded scoring identities; one warm-up and five measurements per operation. These CPU seconds are not added to historical GPU bills without an identified CPU tariff.

Table 17: Local bookkeeping seconds: five-repeat medians. Cache-only constructs the distinct-identity set; support aggregation additionally constructs exact integer coefficient maps and both supports. Excludes reading, hashing, scores and generation.

Seed/repeat	Cache	Support	Difference
1/1	0.000442	0.119690	0.119248
1/2	0.000658	0.135111	0.134453
2/1	0.001007	0.143030	0.142022
2/2	0.000641	0.149643	0.149002
3/1	0.000396	0.080597	0.080200
3/2	0.000529	0.113345	0.112816

G.4.1 Residual moments and pool limits

Let $d = \mathrm{TV}(p_0, p_j) > 0$, and let N be the number of residual proposals consumed for one child in one root draw, without truncation or batch overshoot. A residual is needed with probability d. Conditional on that event, N is geometric with success probability d, so

$$\mathbb{E}N = 1, \qquad \mathbb{E}N^2 = \frac{2-d}{d}, \qquad \mathrm{Var}(N) = \frac{2(1-d)}{d}.$$

At $d = 0$, $N = 0$ almost surely. Small positive d therefore produces rare but long residual requests despite a constant unconditional mean. Both star arms have the same expected proposal count. This does not identify their incremental wall time when batches, devices, and a shared experimental pool differ.

Conditional on requiring m residuals, with iid proposals independent of this requirement, a cap of K proposals fails with probability

$$F_{K,m}(d) = \Pr\{\mathrm{Binomial}(K, d) < m\} = \sum_{k=0}^{m-1} \binom{K}{k} d^k (1-d)^{K-k}. \tag{5}$$

The geometric tail $(1-d)^K$ applies only to $m = 1$. As conditional arithmetic, $d = 0.003187$ and $K = 1024$ give failure probabilities approximately 0.038 for one residual and 0.769 for five. These are illustrative probabilities at an assumed fixed d, not measured failure rates of a heterogeneous job queue.

Heterogeneity has no universal direction for a general pool. For $1 \le m \le K$ and $0 < d < 1$,

$$F'_{K,m}(d) = -K \binom{K-1}{m-1} d^{m-1} (1-d)^{K-m}.$$

For $m > 1$, the sign of $F''_{K,m}(d)$ is the sign of $(K-1)d - (m-1)$. The function is concave below this threshold and convex above it. For $m = 1$ it is convex, and Jensen's inequality applies to a mean over the same population weights. A proposal-weighted empirical acceptance rate is not automatically that mean. Pool sizing requires the per-condition acceptance behavior and residual demand, rather than a queue-wide rate alone.

G.4.2 Computational cost audit

The original harness recorded batch timers whose scope is narrower than the accounting used here. Those timers measure generation only, not density evaluation, model loading or bookkeeping. Independent autoregressive and shared-token runs time their generation batches, whereas tree runs time root generation and omit the residual-proposal durations, so a smaller recorded sum for the

tree is not evidence of cheaper execution. A residual pool shared between two experimental arms likewise does not yield a separate measured bill for each arm, and reusing a frozen probability pilot is a different deployment scenario from paying for a new one. The 24 method–job sums themselves are retained in the supplementary audit records rather than reprinted here.

The original harness also stored flat-rate API estimates. Appendix G.5.1 replaces those estimates with usage-priced accounting and reconstructs the complete controller task intervals, which include the work the component timers did not cover. The main-text before–after comparison charges the same complete generation interval to both scoring policies.

G.5 Collection history, accepted timings and resource archive

This subsection is the one place that records which recorded timings were replaced and why. Twelve earlier isolated replays replaced contended budget-4 collection times, and seven uninterrupted full-process replays replaced cells with failure or recovery histories. All 19 reproduce the original token and decoded-text identities and add no statistical samples; the six new Tree budget-4 banks, by contrast, are real additional generation (Appendix D.2.6). The replays were run as fresh single-process child jobs, one per GPU, on a two-RTX-5090 server (32 GB per GPU; Python 3.12.3, PyTorch 2.8.0+cu128, Transformers 4.56.2), timing model loading, the Tree pilot where applicable, generation, density calculations and saving, but not independent reconstruction, archive transfer or local analysis; every archive was downloaded and SHA-256 checked. The remaining accepted historical timings are retained, so this is a targeted repair and not an all-cells same-machine benchmark. Historical charges, including the campaign GPU allocation, stay in the ledgers below and are never added to accepted workflow timings. The difference between historical and replay charges includes CPU, I/O and runtime differences, and several replay costs are slightly larger. The inventory of banks and of the versioned score cache is stated once, in Appendix B.2.2; that count is an experimental audit union, not a deployment query count or a post-saving bill.

G.5.1 Historical cost reconstruction

Controller events are matched by job and GPU UUID, with an explicit check that leases do not overlap; all recovered starts have exits, and resumed segments are costed under both their original and their continuation records. Reserved task time within the recorded five-card campaign came to 223.04 card-hours, or 669.12 CNY at 3 CNY per reserved card-hour, and continuous rental outside assigned tasks is a separate account-level expense. This total covers the entire research campaign, including validation, resumed stages, training, distance-scan generation, both star families and the baselines. It is not assigned to any single deployment comparison. Its per-category breakdown is retained in the supplementary audit records.

The API audit reconstructs usage-priced costs from returned token counts, cache-hit counts and model identity. DeepSeek Flash is priced using its documented peak/nonpeak schedule in Beijing time; Qwen uses its model-specific input, output and cache tariffs. We checked the DeepSeek tariff page and Qwen Flash tariff page on September 12, 2026. These reconstructed amounts represent research expenses rather than workflow costs and differ from both the earlier local flat-rate reservations and the provider account invoice. The cost breakdown by judge is retained in the supplementary audit records, including 258 calls with no returned usage data. These calls remain unresolved billing entries and are not assigned zero cost. Historical imported scores are outside that live-call accounting. Price sources, timestamps, raw responses and calculation code are retained in the result package.

Strong-baseline phase windows. The strict-holdout baseline process executes independent autoregressive, shared-token and TV-tree sampling sequentially. Original server save timestamps delimit their inclusive phase windows. Common loading/tree setup and finalization are listed separately. Each complete repetition comprises four disjoint eight-prompt shards. These are observed phase costs within that workflow, including density computations in each phase; they are not independently benchmarked standalone launches.

Saved-prefix and numerical audit. Continuation preserves the saved candidate prefix, the frozen policies, the output cap and the random namespace, so the completed pools use a ceiling of 8,192 proposals exactly where the original 1,024-proposal prefix needed continuing; all 896 copied checkpoint

Table 18: Inclusive strict-holdout baseline phase costs in CNY, reconstructed from original server timestamps. Common setup and finalization are separate.

Model	Repetition	Independent AR	Shared tokens	TV tree	Common
Qwen 1.5B	1	6.774	6.813	8.031	0.076
Qwen 1.5B	2	6.785	6.849	7.673	0.074
Smol 1.7B	1	5.681	5.662	7.857	0.079
Smol 1.7B	2	5.652	5.719	6.648	0.077

files match their recorded SHA-256 hashes, and the detailed logs stay in the archived ledgers. No budget-exhausted cell is removed from the completed analysis, and the original numerical acceptance records and source hashes are retained with the reproduced generation audit.

Table 19: All seven targeted timing replays (training seed 1). Historical charges and accepted uninterrupted charges remain separate; differences are not interpreted as isolated failure losses.

Sampler	n	Repeat	Historical CNY	Replay hours	Replay CNY
Independent star	4	1	3.603	1.259	3.777
Independent star	4	2	3.395	1.172	3.515
Independent star	8	1	3.925	1.286	3.859
TV-tree	8	1	4.877	1.564	4.693
TV-tree	8	2	4.916	1.632	4.895
TV-tree	16	1	6.574	2.187	6.560
TV-tree	16	2	6.776	2.237	6.712

H Extensions of the CAFA framework

H.1 Extension overview

Three extensions change what is sampled or how many scores are bought, rather than only which scores are skipped; none is part of the exact core of the main text. Shared acceptance (Appendix H.2) uses one copy uniform for all updates to create more all-copy draws. It preserves every base–update law, but its unique-input increment on top of CAFA is small and changes sign on some banks. Residual grouping (Appendix H.3) lets updates share residual proposals. It preserves every base–update law and reduces the required support, but its recorded cost including the pilot rises, so it buys fewer scores for more money. Budgeted acquisition (Appendix H.4) buys fewer scores than the exact support with a generalized-difference estimator; restricting the budget to the non-zero support lowers the conditional MSE substantially, while active allocation shows no consistent advantage over uniform allocation.

H.2 Shared acceptance

Shared acceptance uses a common uniform random variable for copy decisions across updates to increase all-copy draws without changing any base–update law (Proposition 4). It minimizes a restricted component-query objective (Proposition 5), although it changes the joint law across updates. On the initial fixed-prompt bank, the four-run unique-input total falls from 2728 to 2303. The conditional component ratio is 0.8925, with a 95% prompt-cluster interval of $[0.8846, 0.8994]$; this ratio has a different denominator from the unique-input reduction. On the comparison bank, shared acceptance removes a further 7.4% of the inputs remaining after cancellation for Qwen and 19.1% for SmolLM2. The increment changes sign on the 7B banks and in some strict-holdout repetitions, indicating a small and inconsistent additional benefit over CAFA.

H.2.1 COUPLING AND BUDGET EXTENSIONS

The following constructions use established sampling principles to explore how dependence changes CAFA support.

Shared acceptance for a common base The Independent Star of Section 3.1 and Appendix B.3 draws $X \sim p_0$ and copies it into policy j with probability $a_j(X) = \min\{1, p_j(X)/p_0(X)\}$, the classical maximal coupling (Levin & Peres, 2017); otherwise it samples the normalized residual law

$$\rho_j(y) = \frac{(p_j(y) - p_0(y))_+}{\mathrm{TV}(p_j, p_0)}, \tag{6}$$

which is never needed when the total variation is zero, and which is implemented by proposing $Z \sim p_j$ and accepting with probability $(1 - p_0(Z)/p_j(Z))_+$ at a cost that must be counted. The single change made here is the copy uniform: the Independent Star tests $U_j < a_j(X)$ with separate $U_j \sim \mathrm{Uniform}(0, 1)$, whereas we use one common U, independent of X and of the residual streams. Conditional on X, the probability of all copies changes from $\prod_j a_j(X)$ to $\min_j a_j(X)$. Each child still copies with its original probability. Residual streams remain independent across children within an arm; reproducible sharing between experimental arms is recorded separately.

For the primary base-relative target, an all-copy draw requires no score. Other draws use coefficient aggregation in Eq. (3). Sharing acceptance therefore targets an event with direct feedback consequences, without using quality scores to select favorable responses.

Proposition 4 (Preserved base–update laws)**.** *Under the independent and shared acceptance constructions, every pair (X, Y_j) has the same joint distribution. Consequently, for independent repetitions, each individual base–update contrast estimator has the same finite-sample distribution.*

For any measurable response sets A and B, $\Pr(X \in A,\, Y_j \in B)$ is the integral over $x' \in A$ of $p_0(x')[a_j(x')\mathbf{1}\{x' \in B\} + (1 - a_j(x'))\rho_j(B)]$. That expression involves only X, the conditional copy probability and the residual kernel for the fixed j, never the dependence of the copy uniform across children, so it is identical under independent and shared acceptance. In particular Y_j has marginal law p_j: the copied mass is $\min(p_0(y), p_j(y))$ and the residual mass is $(p_j(y) - p_0(y))_+$. The argument requires the residual streams to remain independent of the root and of the copy uniforms; independent generating namespaces across repetitions then preserve the complete sample law of each fixed-j average. The equality does not extend to the full vector $(Y_0, \ldots, Y_J)$: cross-contrast covariances and checkpoint-selection error can change. Reusing random streams across experimental arms changes the dependence between arms but preserves the marginal sampling law of each arm.

Proposition 5 (Restricted component-query objective)**.** *Fix $X = x'$ and the copy probabilities $a_1, \ldots, a_J$. Let B_j indicate copying and define $N_{\mathrm{comp}} = (1 + \sum_j (1 - B_j))\mathbf{1}\{\text{not all } B_j = 1\}$. Among joint copy indicators with these Bernoulli marginals,*

$$\mathbb{E}[N_{\mathrm{comp}} \mid X] \geq 1 + \sum_j (1 - a_j) - \min_j a_j, \tag{7}$$

with equality for shared acceptance. Its reduction relative to independent acceptance is $\min_j a_j - \prod_j a_j$.

An all-copy round has zero mismatches, so $N_{\mathrm{comp}} = 1 + \sum_j (1 - B_j) - \mathbf{1}\{B_1 = \cdots = B_J = 1\}$; taking conditional expectations gives $\mathbb{E}[N_{\mathrm{comp}} \mid X] = 1 + \sum_j (1 - a_j) - \Pr(B_1 = \cdots = B_J = 1 \mid X)$, and an intersection probability cannot exceed its smallest marginal, a bound that a common uniform attains. The multi-way overlap principle is classical (Makur & Singh, 2024); the specific objective here removes all-copy rounds because their contrast is identically zero. Two boundaries follow from the same identity. For a single update $\min_j a_j = \prod_j a_j$, so shared acceptance has no incremental component advantage at all; and when every $a_j = 1$ both arms require zero feedback. Extra text collisions and batch-level coefficient cancellation can lower realized cost further without changing this component identity.

Residual cost and break-even feedback price. The residual moments, the pool-exhaustion analysis and the break-even price $\Delta G/\Delta L$ are in Appendices G.4.1 and G.3; small overlap distance does

not imply cheap execution. One consequence is specific to this extension: costs common to both star constructions cancel in their incremental comparison, but they do not cancel when the full star sampler is compared against independent generation, so a within-star increment and a star-versus-AR saving must never be offset against each other.

A bridge from component counts to text-level acquisition. Let $K = \sum_t N_{\mathrm{comp},t}$ for the base-relative star construction. After removing all-copy draws, its component occurrences cover S_{draw}. Write $D_{\mathrm{reuse}} = K - |S_{\mathrm{draw}}| \geq 0$ for reductions from text grouping and reuse, and $D_{\mathrm{batch}} = |S_{\mathrm{draw}}| - |S_{\mathrm{mean}}| \geq 0$. Then, pathwise,

$$|S_{\mathrm{mean}}| = K - D_{\mathrm{reuse}} - D_{\mathrm{batch}}. \tag{8}$$

Shared acceptance minimizes the conditional expectation of the component term in Proposition 5. The two correction terms also depend on the sampler. Consequently its realized unique-input increment can change sign, consistently with the 7B results. Cancellation itself always weakly reduces acquisition on a fixed generated bank.

H.2.2 Additional coupling and computation results

Simultaneous agreement and decoded text The classical multi-way agreement bound is

$$\Pr(Y_0 = \cdots = Y_J) \leq \sum_y \min_{0 \leq j \leq J} p_j(y).$$

Shared acceptance attains it: integrating $p_0(y) \min_{j \geq 1} a_j(y)$ gives the right side. On a residual branch, the base and that child cannot coincide on the same token sequence. This is a specialization of simultaneous maximal coupling (Makur & Singh, 2024), not a guarantee about the number of scored texts after batch aggregation.

For example, let four token sequences decode to A, A, B, B, and take $p_0 = (1/2, 0, 1/2, 0)$ and $p_1 = (0, 1/2, 0, 1/2)$. All couplings have the maximum token-sequence agreement of zero. Pairing the first sequence with the second and the third with the fourth gives text agreement one; reversing these pairings gives zero. Independent residual sampling gives $1/2$. Thus token maximality alone does not determine text-level feedback cost.

H.2.3 Does changing the acceptance dependence add more?

Shared acceptance is an increment over an already coupled baseline, with cancellation available in both arms. The expected component-cost ratio, averaged over repetitions within prompt and then over prompts, is 0.8925 with a 95% prompt-cluster bootstrap interval of [0.8846,0.8994]; the four-run required-input totals are 2728 against 2303, a 15.6% reduction (per-repetition counts in Table 20). Proposition 5 fixes the sign of the analytical component reduction, not its magnitude, and the inequality implies no particular threshold. The interval pairs the same prompt resamples in numerator and denominator and is conditional on the fixed updates and collected repetitions; it is a component-cost statement, distinct from realized label savings and their uncertainty.

The increment is also smaller and less stable than cancellation itself. On SmolLM2 strict holdout at $n = 16$, shared acceptance costs more in two repetitions, ties in one, and costs less in one: the realized increment changes sign within a single bank family. Its comparison-bank means, and the tree means they sit against, are the last columns of Tables 7 and 8. Equation (8) separates the component guarantee from these realized text-level increments. Running multiple arms for verification incurs their combined acquisition costs, subject to caching.

H.3 Residual grouping

Residual grouping shares proposals across updates to increase agreement among their non-copied responses. A mixture kernel with a fixed active block and an independent fallback preserves each base–update law (Proposition 6); a score-free pilot selects the partition. We evaluate 18 banks from three seeds, three arms and two repetitions, using 32 selection prompts and $n = 8$ (Appendix H.3.2). All three arms use CAFA and differ in their residual dependence. The control uses shared copy acceptance with independent residuals, unlike the Independent Star used in the main study. Pilot-selected grouping consistently requires fewer scores than independent residuals, but costs more after charging its pilot and historical attempts. Pooling increases costs further.

Table 20: Required scoring inputs per judge on the fixed Qwen prompt set. Counts use per-run text deduplication after removing identical-pair contributions. Their sum does not deduplicate across repetitions and does not represent the total API expenditure of the experiment.

Generation repetition	Independent acceptance	Shared acceptance
1	669	566
2	650	547
3	713	627
4	696	563
Total	2728	2303

H.3.1 Using acquisition cost to choose residual dependence

The coefficient interface can also guide the sampler. For a fixed comparison target, two couplings with the same base–update laws can expose different scoring supports because their updates agree with one another at different rates. We examine this degree of freedom by choosing which residual requests share proposals. This extends the CAFA framework to a concrete design decision without making its exactness depend on that decision.

Retain the shared root $X \sim p_0$ and copy uniform from the star construction. Define $f_j(y) = (p_j(y) - p_0(y))_+$, $\delta_j = \int f_j(y)\,dy$, and $\rho_j = f_j/\delta_j$ when $\delta_j > 0$. For a partition $\mathcal{P}$ of the updates, intersect each block with the updates requesting a residual in this draw. A singleton uses independent residual sampling. For an active block B of size at least two, draw an independent sequence of proposals and shared uniforms according to

$$Z \sim m_B = \frac{1}{|B|}\sum_{k\in B} p_k, \qquad U \sim \mathrm{Unif}(0,1), \qquad j \text{ accepts if } U < \frac{f_j(Z)}{\sum_{k\in B} p_k(Z)}. \tag{9}$$

Each member keeps its first accepted proposal. The mixture retains the original active block throughout this request, including after some members finish. After $K = 32$ proposals, unfinished members use independent draws from ρ_j. The kernel requires policy densities but not the unknown residual normalizing constants. Shared proposal acceptance and independent completion are established rejection-sampling tools (Corenflos & Särkkä, 2022); the design variable here is the partition selected for contrast acquisition cost.

Proposition 6 (Partition choice preserves the pair laws)**.** *Fix a partition chosen from pilot randomness independent of the formal draws. Under Eq.* (9) *and the fixed-K fallback, each residual output has law ρ_j, conditional on any active block containing j. Consequently every (X, Y_j) has the same law as shared-copy star sampling with independent residuals. For independent draws, every individual base-relative gain retains its distribution and variance. Cross-gain covariance and multi-checkpoint selection error can change.*

For one proposal, the joint density of $Z = y$ and acceptance by j is $f_j(y)/|B|$, hence its conditional accepted law is ρ_j. A fixed number of failures followed by independent ρ_j completion has the same output law. Appendix H.3.3 proves the statement, including the conditioning on the active set and the implementation cap.

A score-free partition decision. With three updates there are five partitions, including independent residuals and one pooled block. On an independent pilot we choose

$$\widehat{\mathcal{P}} = \arg\min_{\mathcal{P}} \left\{ \widehat{G}(\mathcal{P}) + \lambda \widehat{Q}_{C,\mathrm{mean}}(\mathcal{P}) \right\}, \tag{10}$$

where $\widehat{G}$ estimates generation and density expenditure, $\widehat{Q}_{C,\mathrm{mean}}$ estimates the post-cancellation mean support, and λ is a frozen mean scoring-input price estimated from historical token usage. The implemented surrogate uses a finite, self-normalized approximation to the policy distributions; it is a decision rule, not a population cost bound. Formal cost reporting uses actual token usage and the support for both means and variances. The pilot is charged separately whenever the selected workflow is deployed. Section H.3.2 tests the resulting partition against both fixed alternatives and relates the outcome to the broader sampler comparison.

H.3.2 RESIDUAL GROUPING: WHAT THE TARGET-DRIVEN DESIGN CHANGES

We evaluate the partition rule in Section H.3.1 on 32 prespecified selection-set prompts, with eight draws, three GRPO seeds and two generation repetitions per seed. Independent residuals, one pooled group and the pilot-selected partition give 18 complete banks. Roots and copy uniforms are paired across the three arms. This experiment isolates residual dependence. The independent-residual control uses shared copy acceptance, whereas the independent star in the main study uses independent copy acceptance. The independent pilot uses 16 prespecified selection prompts per seed, and all three pilots select $\{\{1\}, \{2, 3\}\}$. This experiment is separate from the 128-prompt main study and its independent selection/test application.

A consistent but small residual-grouping increment under CAFA. All three arms here apply the same CAFA rule; the comparison varies only the residual dependence upstream of it. Pilot-selected grouping reduces joint mean/variance support by 2.1–3.9% relative to independent residuals across all six seed/repetition cells. Pooling reduces it by 2.8–5.1%. The single base-relative supports are unchanged in every corresponding cell: the extra saving comes from sharing across updates. For the step-50 to step-250 adjacent contrast, selected grouping reduces scoring support by 5.9–11.9%. This secondary target was recomputed from the fixed banks. The RMS MCSE ratio for the joint gains is 0.937–1.034 for selected grouping, with no uniform precision improvement. Table 21 summarizes the cells; Appendix H.3.4 retains target-specific results and uncertainty.

Table 21: Residual-group validation, descriptive means of six seed/repetition cells. Each bank has 32 prompts and eight draws. $Q_{\mathrm{mean+cov}}$ is the number of inputs required for joint mean-and-covariance reporting, G includes all historical generation attempts, P is the pilot charged per deployment, and $T = G + P + F_{\mathrm{mean+cov}}$ includes feedback at the frozen tariff. All costs are CNY; recovery overhead is not separately identifiable.

Residual grouping	$Q_{\mathrm{mean+cov}}$	G	P	T	RMS MCSE
Independent	714.83	5.532	0.000	6.493	0.06535
Pooled	686.83	6.773	0.000	7.701	0.06522
Pilot selected	694.50	5.076	0.707	6.724	0.06504

Efficient residual proposals do not by themselves pay for the workflow. Three quantities move in different directions here and are measured over different scopes. Selected grouping generates 55.7% fewer proposal tokens on average than independent residuals, and its recorded generation-batch time is 13.9% lower, but those timers exclude parts of loading, probability replay and I/O, so neither figure is an end-to-end speed-up and neither can be combined with the total cost below. Pooling instead increases recorded generation time and sends 3.34% of shared requests to independent fallback on average, versus no shared-request fallbacks for the selected partition. Thus both which members share and how their proposals execute affect cost.

Charging the complete recorded attempts and one pilot per deployment, selected grouping costs 1.6–5.1% more than independent residuals; pooling costs 9.7–31.5% more (Figure 31). Interrupted attempts contain both retained useful work and recovery overhead. We preserve these charges and do not interpret their difference as a measured uninterrupted-deployment difference. A conditional pilot-amortization calculation in Appendix H.3.3 suggests modest savings if the same partition serves multiple comparable banks. It is an extrapolation under a stated stability condition.

Relation to the strong baselines. Recomputing the four main samplers on the same 32 prompts and $n = 8$ gives a further 24 comparison cells. TV-tree requires fewer scoring inputs than selected grouping in all six seed/repetition cells; selected grouping requires 1.9–10.5% more. These baselines use different random streams. Their 32-prompt subsets have no separate full deployment timers, so we compare support and MCSE without dividing the 128-prompt cost by four.

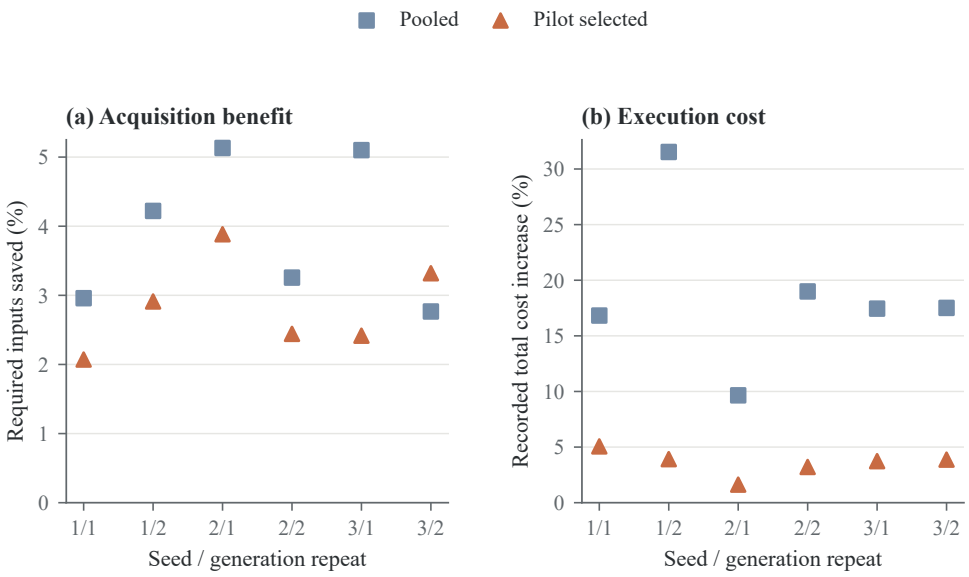


Figure 31: Residual grouping separates acquisition and execution outcomes. Each point is one training seed/generation repetition, with independent residuals as the matched control. Left: reduction in joint scoring support. Right: increase in recorded generation-plus-feedback expenditure, charging the pilot for selected grouping and retaining historical recovery overhead.

H.3.3 Residual grouping: laws, implementation, and finite checks

Correctness under a fixed proposal cap Condition on the independent pilot, a root and its copy decisions, and a nonempty active block B. For $j \in B$, a single proposal satisfies

$$m_B(y)\frac{f_j(y)}{\sum_{k\in B} p_k(y)} = \frac{f_j(y)}{|B|}, \qquad a_j = \Pr(j \text{ accepts}) = \frac{\delta_j}{|B|}.$$

Zero proposal-density points have probability zero and require no acceptance ratio. Successive proposals and uniforms are independent, so the first proposal accepted by j has law ρ_j. With K shared proposals followed by an independent residual draw, the output probability of any measurable set A is

$$\sum_{t=1}^{K}(1-a_j)^{t-1}a_j\rho_j(A) + (1-a_j)^K\rho_j(A) = \rho_j(A).$$

Other members may finish earlier, but neither the proposal law of the fixed block nor the acceptance rule for j changes. Shared uniforms affect dependence between members, not this marginal calculation. A singleton instead proposes from p_j and accepts with probability $1 - \min(1, p_0/p_j)$, which gives the same residual law. If $\delta_j = 0$, that update copies almost surely and requests no residual.

The conditional residual law is the same for every possible active block, root, and copy event. Integrating these conditions gives copied mass $\min(p_0(y), p_j(y))$ and noncopy joint density

$$(p_0(x) - p_j(x))_+\rho_j(y).$$

Thus both the policy marginal and the entire base–update pair law are preserved. Independent namespaces across root draws preserve the distribution of each single-gain sample mean. The covariance between two gains depends on their residual coupling, so neither adjacent-contrast precision nor a nonlinear selector is fixed by this result. The same statements hold after applying any deterministic decoding and fixed scoring function to the token sequences.

The implementation generates shared proposals in batches of eight and independent residual proposals in batches of sixteen. Accepted proposals are processed in their frozen order; unused proposals still incur generation cost. Each prompt, update and generation repetition has an 8,192-proposal ceiling. Exhaustion returns an incomplete artifact rather than a substituted answer. All 18 reported banks completed with all 32 prompts. The fixed-K fallback proof concerns the sampler.

Pilot and deterministic implementation For each training seed, the pilot uses 16 prespecified stratified selection prompts and eight independent responses per policy, with no score queries. At each prompt it evaluates all four policy densities on the 32 sampled answers, forms self-normalized mixture-importance weights and merges duplicate token sequences. Decoded-text collisions are also merged when computing CAFA support. Each of the five partitions is evaluated on 512 simulated banks of eight draws, using observed per-proposal generation, cross-density and weight-switch times. The surrogate includes batch overshoot and residual fallback costs. The frozen historical price estimate is approximately 0.001282 CNY per scoring input; formal feedback charges use measured input and output token counts at the fixed 2/8 CNY per million-token tariff.

The candidate partitions are

$$\{1\}|\{2\}|\{3\}, \quad \{1,2\}|\{3\}, \quad \{1,3\}|\{2\}, \quad \{1\}|\{2,3\}, \quad \{1,2,3\}.$$

All three pilots choose $\{1\}|\{2,3\}$ before formal validation. Pilot and validation prompts may overlap within the selection set, but their generated responses and random namespaces are independent. The validation matrix was fixed before its outcomes were available. No validation scores, sorting outcomes, or observed savings alter a partition.

For completeness, a finite selection rule has a simple conditional stability property. If $|\widehat{L}(\mathcal{P}) - L(\mathcal{P})| \le \epsilon$ simultaneously for all candidates, then $L(\widehat{\mathcal{P}}) \le \min_{\mathcal{P}} L(\mathcal{P}) + 2\epsilon$. This follows by adding the two approximation errors around the empirical minimizer. The pilot used here does not measure ϵ.

Root and copy streams are common across the three formal arms, while residual streams are arm-specific. Each policy export, pilot decision and protocol is identified by a hash. FP32 sequence log probabilities use the original token IDs, temperature one and the original EOS rules; decoding only supplies the feedback identity. Independent reconstruction recomputes copy decisions, mixture indices, shared acceptance, fallback paths and the final response bank. The maximum observed probability-replay discrepancy is 0.001302 against a tolerance of 0.01.

Exact finite-distribution checks and an attainable improvement The finite-law audit enumerates the joint distribution for all partitions with $K \in \{0, 1, 3, 32\}$. It covers identical policies, disjoint supports, residual masses of order 10^{-8}, unequal residual masses and singleton blocks. Marginals and base–update pair distributions are checked independently. Separate cases merge distinct token states into one decoded scoring input, and verify that a zero mean support can coexist with a nonempty variance support.

An explicit three-state example shows why residual dependence can affect CAFA requirements even with all pair laws fixed. Let

$$p_0 = (0.5, 0.25, 0.25), \qquad p_1 = p_2 = (0.3, 0.35, 0.35).$$

The independent-residual construction has expected joint query count 0.5 per root draw. Pooling with the fixed-32 fallback gives

$$0.4 + 0.1(0.9)^{32} = 0.4034336838\ldots.$$

With unlimited shared proposals the expectation is 0.4. Linear programs over all couplings, and over couplings constrained to preserve the base–update pair laws, both give lower bound 0.4 in this example. Merging the two residual states into one decoded identity also yields 0.4 with finite fallback. These exact finite results establish available CAFA improvement.

What the cost records identify Formal generation charges sum all recorded attempt intervals. Interrupted attempts retain useful completed batches as well as unseparated loading, recovery and repeated work. Removing every failed interval would remove valid generation; retaining only the last successful attempt would omit the saved prefix. The ledger therefore identifies the full observed execution cost, but does not identify a fresh uninterrupted deployment cost. The saved batch timers isolate generation and cross-density operations as a lower-bound diagnostic; they exclude parts of replay, loading and I/O.

Pilot reuse is a separate deployment assumption. If one frozen partition remains applicable to k comparable banks under the same policies, a descriptive amortization model is

$$T_{\text{grouped}}(k) = P + k(G_{\text{grouped}} + F_{\text{grouped}}), \qquad T_{\text{independent}}(k) = k(G_{\text{independent}} + F_{\text{independent}}).$$

Using the recorded cell means gives about 1.9% lower expenditure at $k = 2$ and 4.6% at $k = 4$, versus 3.5% higher expenditure for one bank. This assumes unchanged marginal throughput and prompt distribution; reusing the pilot for a new checkpoint collection requires the same conditions to hold. The measured 32-prompt evidence supports a bounded mechanism improvement, not a lowest-total-cost deployment claim. The supplementary tables retain the per-bank decomposition behind those means. For each of the 18 banks they list the recorded generation charge including all attempts, the deployed pilot and the measured-token feedback charge, each kept separate, together with support, MCSE and proposal counts. Historical recovery overhead is not separately identifiable within the generation charge.

H.3.4 Residual-group acquisition, precision, and matched baselines

All figures and companion numerical files retain the three training seeds and two generation repetitions separately. The 32 formal prompts belong to the selection partition. All 126 mean/variance support audits are exact against the fully scored frozen bank. Full per-cell covariance matrices and all seven target reports are retained in the machine-readable result bundle.

Prompt variability in the CAFA increment. For each training seed, we average the two paired generation repetitions within prompt and bootstrap the 32 prompt clusters 10,000 times. The following exploratory percentile intervals condition on the frozen pilot and model; training seeds are not pooled. They quantify a small support improvement.

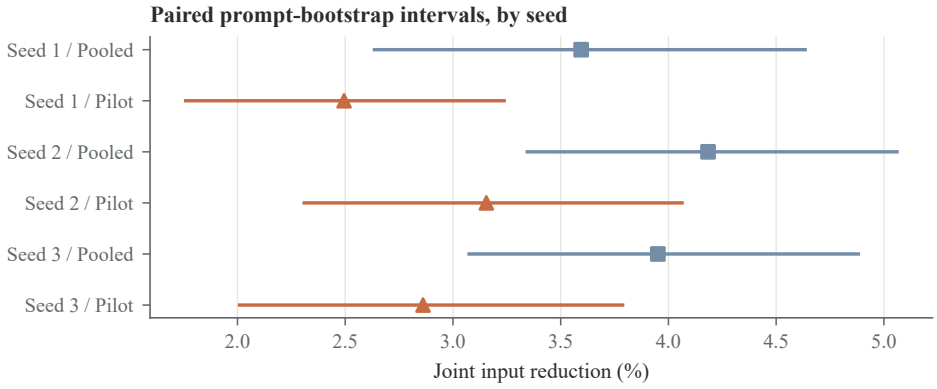


Figure 32: **Joint input reduction by training seed.** Points and 95% intervals use the original paired prompt bootstrap (10,000 resamples), averaging the two paired generation repetitions within prompt. Seeds are not pooled. These intervals describe CAFA increments on the frozen selection prompts.

Single and adjacent targets. The scoring counts for each single base-relative target agree across all three arms in every matched cell. Across the six cells their means are 108.33, 369.50 and 473.50 for steps 10, 50 and 250, respectively. The adjacent targets below can benefit from cross-update sharing even though the individual base-relative pair laws are unchanged. Adding the base absolute mean restores the full cached support within each arm; different arms can still have different cached supports.

Realized gains across the three arms. The base-relative gains themselves, for every seed, generation repeat, arm and target with their fixed-prompt MCSE, are retained in the supplementary tables. They make one point: a different coupling can change the gain a bank realizes, while CAFA recovers each bank's own statistics exactly. Changing the residual dependence is therefore a change of experiment, not a change of estimator.

H.4 Budgeted acquisition

When the budget is insufficient to score the full required support, we use a generalized-difference estimator with a paid pilot and a fitted score surrogate. Conditional on the bank and pilot, the estimator

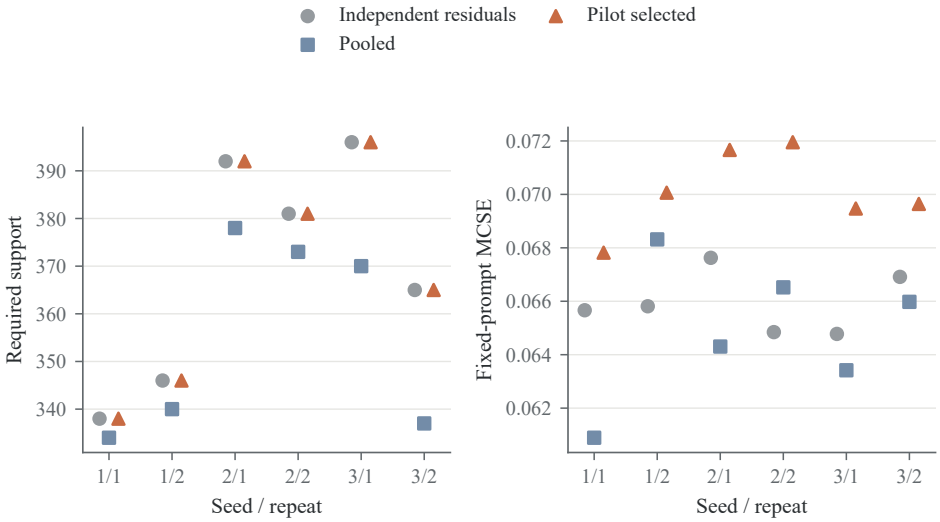


Figure 33: **Adjacent step 10–50: support and precision.** All three residual-grouping arms are recomputed on the same 18 banks. Panels show the support needed for mean and covariance, and the fixed-prompt MCSE, for the step-10 to step-50 adjacent contrast. Side-by-side markers compare the three arms within each seed/repeat cell; cells are not connected. Exact values are retained in CSV.

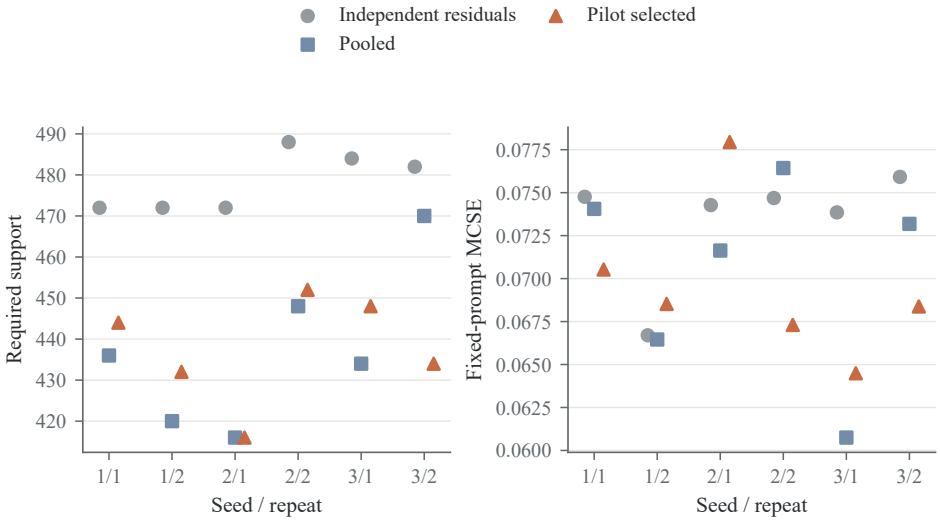


Figure 34: **Adjacent step 50–250: support and precision.** Continuation of Figure 33, with the same definitions and preserved per-cell values.

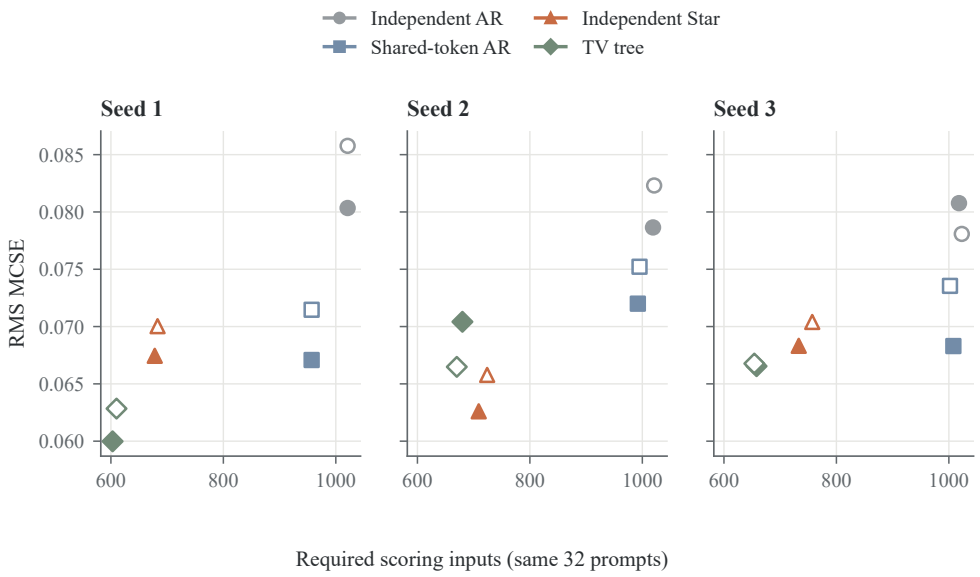


Figure 35: **Existing samplers on the same 32-prompt subset.** Each point is an existing $n = 8$ seed/repetition cell; filled and open markers denote the two repeats. Required support includes mean and covariance. These baselines have independent random namespaces from residual grouping. Standalone subset generation cost is unavailable and is not inferred from the support axis.

is unbiased for the fully scored bank contrast and has an exact query-induced MSE. This approximate extension is optional. We compare five acquisition policies on the strict-holdout Independent Star banks over a common budget grid (Appendix H.4.3). Restricting acquisition to the non-zero support reduces error, while active allocation offers no consistent advantage over uniform allocation at the same budget. These errors are conditional on the frozen bank and do not measure language-model population error. Exact recovery requires acquiring the full necessary support.

H.4.1 Acquisition under a smaller scoring budget

Exact acquisition supplies the endpoint of a budget–accuracy curve: once all nonzero groups are scored, the realized comparison is known. Smaller budgets can use the same coefficient map. This connects contrast cancellation to active testing and surrogate-based evaluation (Kossen et al., 2021; 2022).

Condition on a generated bank and a paid pilot P. Let a_i be the normalized contrast coefficient of unique input i, and let $\widetilde{r}_i$ be a prediction fitted using only the pilot. Include each remaining nonzero input independently with probability $\pi_i > 0$. With $I_i \sim \text{Bernoulli}(\pi_i)$, return

$$\widehat{\theta}_B = \sum_i a_i \widetilde{r}_i + \sum_{i \in P} a_i (r_i - \widetilde{r}_i) + \sum_{i \notin P : a_i \neq 0} \frac{I_i}{\pi_i} a_i (r_i - \widetilde{r}_i). \tag{11}$$

This is a generalized-difference estimator with classical inverse-probability correction. Its useful interface with cancellation is exact: zero groups receive no probability budget, and the full-support limit recovers the original comparison. Conditional on the bank and pilot,

$$\mathbb{E}[\widehat{\theta}_B] = \theta_{\text{bank}}, \tag{12}$$

$$\text{Cov}(\widehat{\theta}_B) = \sum_{i \notin P : a_i \neq 0} (\pi_i^{-1} - 1) a_i a_i^\top (r_i - \widetilde{r}_i)^2. \tag{13}$$

Taking expectations over an unbiased generation scheme preserves its target. The covariance in Eq. (13) is the additional uncertainty from purchasing only part of the frozen bank; generation uncertainty is reported separately.

A target, rather than a choice of coordinates. Corollary 1 shows that an invertible reparameterization leaves the required support unchanged. Reducing that support requires changing the requested comparisons: one nearby update instead of a vector that also includes a distant one can change the support substantially.

H.4.2 Conditional correctness and acquisition design

Fix the generated bank, pilot observations, fitted surrogate and inclusion probabilities. In Eq. (11), $\mathbb{E}[I_i/\pi_i] = 1$ recovers each remaining residual exactly in expectation. Independent inclusion makes cross-input covariance zero, and $\mathrm{Var}(I_i/\pi_i) = \pi_i^{-1} - 1$, proving Eq. (13). At $\pi_i = 1$ for all required inputs, no query uncertainty remains. This reasoning applies to any pilot-measurable surrogate and any strictly positive probabilities on nonzero coefficients, including a poor predictor.

Let K be the required-support size. For $K \geq 2$, each pilot samples $\min\{K - 1, 32, \max\{8, \lfloor 0.05K \rfloor\}\}$ nonzero inputs uniformly without replacement, using twenty fixed seeds. All experimental supports exceed eight inputs. For $K = 0$ the bank contrast is zero; for $K = 1$ acquiring the sole required score gives the exact bank contrast. Its scores are charged to all policies. The surrogate uses an intercept, log response length, log question length, and newline, digit and punctuation fractions. Features are centered and scaled using the pilot, clipped to $[-8, 8]$, and fitted by ridge regression with penalty 10 (intercept penalty 10^{-6}); predictions are clipped to $[1, 10]$. A second ridge fit predicts the squared leave-one-out pilot residual, clipped to $[1, 81]$. Active weights are $\|a_i\|_2$ times the square root of this predicted value. Inclusion probabilities are $\min(1, \lambda w_i)$, with λ chosen for the stated expected budget. Uniform policies set $w_i = 1$. The nonsurrogate policies use the fixed midpoint 5.5 in the generalized-difference estimator.

The second-stage expected budget is 10, 25, 50, 75 or 100% of the remaining required support. The cached-uniform comparator allocates that same expected count over its larger remaining cache. All support-aware comparators use exactly the same cancellation rule. The full-cache comparator buys every distinct input. Reported conditional MSE integrates the Bernoulli query randomness analytically; it is averaged over pilot splits and original frozen generation repetitions. Pilot splits are not additional model-generation or training seeds. Exact moment checks against Monte Carlo simulation accompany the implementation.

H.4.3 What can a smaller scoring budget recover?

We hold every generated response fixed and compare five score-acquisition policies using Eq. (11): uniform acquisition over the cached bank with a midpoint or fitted surrogate; uniform acquisition over nonzero support with those same two predictions; and active acquisition over nonzero support with the fitted surrogate. All policies pay for the identical pilot, including its inputs in the displayed budget. Every surrogate policy uses the same features and fitted model, so the paired comparisons isolate pruning and allocation. The active policy uses only pilot labels and known coefficients to set its probabilities.

The target here is the fully scored finite-bank contrast. The exact query-induced MSE is given by Eq. (13) and is distinct from the unknown population error of the language model. Twenty pilot splits per frozen repetition probe allocation stability; they are splits of the paid pilot, not additional training or generation repetitions, and the four original generation repetitions remain the sampling replication unit. Every method is evaluated on the same fixed budget grid, including the exact full-cache endpoint. The budget is an expected count of unique queries under independent inclusion, rather than a fixed query count per run. The resulting error arises from randomized acquisition on the frozen bank, not from model generation.

At a second-stage expected budget equal to half the remaining required support, exact support pruning reduces finite-bank MSE by 72.2–95.0% relative to uniform acquisition over the cache, with the same paid pilot and fitted score surrogate in both policies. The active-to-uniform-surrogate MSE ratios range from 0.96 to 1.07; the figure reports both policies across the complete budget grid. These are conditional errors for the frozen response bank, integrated exactly over query randomization and averaged over four generation repetitions and twenty pilot splits per repetition. Across the complete budget grid, the active-to-uniform-surrogate MSE ratio is 0.94–1.12.

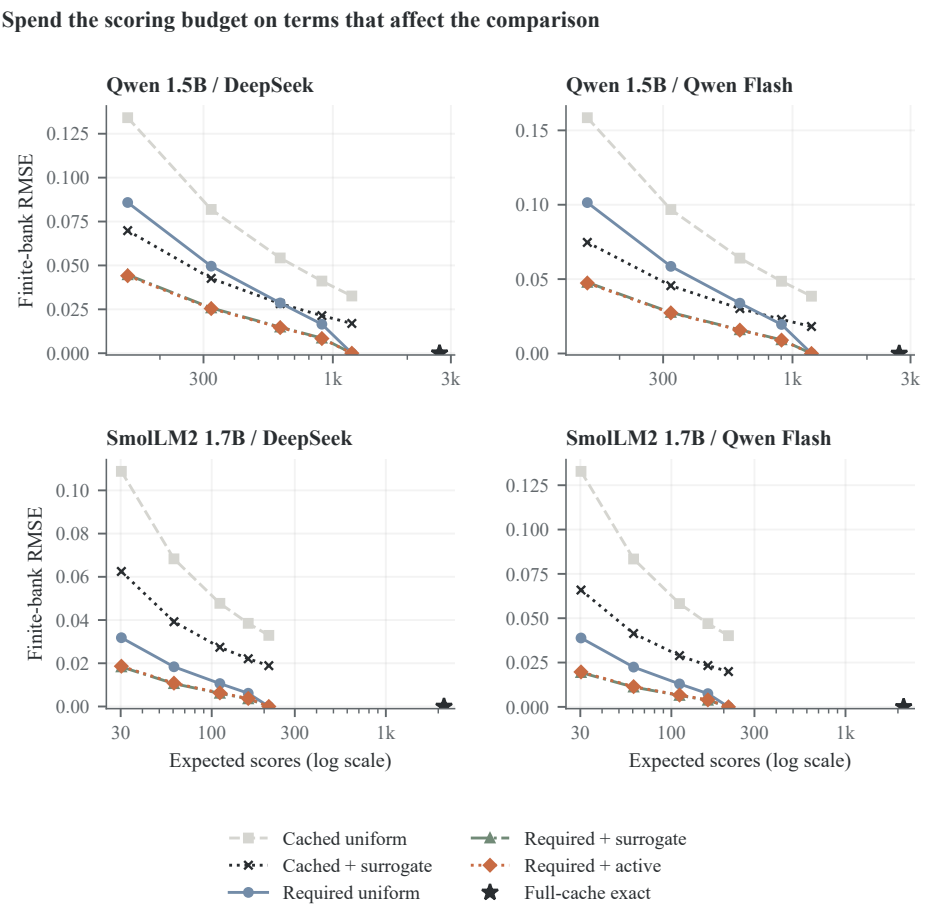


Figure 36: **Budget the scores that affect the comparison.** Strict holdout; the same independent-star bank, 32 prompts, three updates and 64 root draws per prompt. Lines show the square root of mean conditional MSE over four generated banks and twenty paid-pilot splits per bank. Budgets are expected unique-query counts under independent inclusion. At full nonzero support, the three support-aware policies recover the bank contrast exactly; the star marks full-cache exact evaluation.

I Software and Reproducibility

I.1 Installation and Minimal Workflows

The anonymous code archive contains an installable Python package (`contrast-eval` 0.3.0, Python 3.10 or newer) implementing Algorithm 1, a command-line interface, JSON schemas for banks, reports and score records, its test suite and its validation records. The exact core uses only the Python standard library and needs no GPU, no model weights and no network access. The package includes two CPU examples: a synthetic four-model bank evaluated with a single command (`contrast-eval demo`), and offline scoring of the mirror draws in Figure 6, for which the mean requires no scores and the covariance requires both. A third example runs local generation, scoring and recovery end to end. The workflow validates the bank and plans the required support without reading scores or calling a scorer. It then acquires missing scores through a user-supplied scorer or an export–import workflow and returns the recovered report as JSON or CSV. A dry run returns the plan and makes no call. The following example illustrates the workflow using a user-supplied scorer callback, `my_score(prompt, response)`.

```
bank   = load_bank("manifest.json", "draws.jsonl")
report = ReportSpec.gains(bank.model_ids, reference="base",
                          statistics="mean_covariance")
plan   = Evaluator().plan(bank, report)   # no score read
cache  = SQLiteScoreCache("s.db", scorer_version="rubric-v1")
result = Evaluator().compare(bank, report, my_score, cache)
```

I.2 Integrating models, banks and scorers

Banks. A bank is one set of policies, one generation repetition and one sampling configuration, with a fixed model order. Each record is a complete joint draw: one prompt, one draw index, and one response per model; optional token IDs carry the probability identity. Incomplete joint draws, duplicated draws, prompts without draws and declared draw counts that disagree with the file are rejected rather than repaired, because each would change n_i and the estimand. A prompt text that appears under two nominal prompt identities is rejected until the nominal groups are merged, so it is never silently treated as two independent coordinates (Appendix A.2). A bank also declares whether its draws and its prompts are independent. The Monte Carlo covariance of the mean and its standard error are reported only when both are declared; the mean and the per-prompt ordinary sample covariance are recovered in either case.

Reports. The mathematics allows any real report matrix C (Section 2). The implementation represents C exactly using integers or rational strings such as `"1/3"`, with columns ordered by the models in the bank. It supports contrast reports with zero-sum rows and general linear reports, including anchored reports. This is an interface choice that makes the zero test of a coefficient vector reliable: floating-point coefficients are refused rather than compared with a tolerance, because treating a small non-zero coefficient as zero would drop a score the report needs.

Scorers and caches. A scorer receives a prompt and a response and nothing else. Cached scores can be reused only with the same scorer version, defined by the rubric, judge deployment and settings. A changed fingerprint or a conflicting score for the same identity raises an error; conflicting values are not averaged. Refusals, timeouts and parse errors are stored as failures without a value. After a timeout or an unexplained scorer error, the identity remains claimed and marked as pending. A rerun does not issue another paid request until the provider outcome has been reconciled. A required score that remains unavailable either blocks the exact report or, if a score range is declared, yields per-output bounds with no covariance point estimate.

Generation. Imported banks from any sampler are supported. For local causal language models the archive implements two generators, the Independent Star and independent autoregressive sampling. Both keep the original token IDs, and both evaluate log-probabilities of given token sequences under each policy with the same temperature, top-k/top-p, repetition penalty, end-of-sequence rule and length cap used for sampling. They draw from named random streams; giving all models the same seed is not a substitute. The packaged Independent Star implements the copy and residual acceptance rules of the sampler used in the source study. When a residual reaches its proposal ceiling, the draw is returned as an incomplete prefix that can be resumed, never replaced by an independent sample. The port does not reproduce the historical random streams from the source study bit for bit and therefore generates new banks rather than replicas of the reported banks.

The adapters were checked on one GPU with the online-GRPO base model and its step-10 and step-250 LoRA checkpoints, using two prompts and two joint draws per prompt. Repeated generation reproduced the same tokens. Teacher-forced log-probabilities of the original sequences agreed with those computed by the evaluator used in the source study within an absolute tolerance of 0.01, including when temperature, top-k/top-p and repetition penalty were changed. An interrupted residual draw resumed to the same result as an uninterrupted run, and statistics recovered from the generated banks equalled full scoring. This is a functional check. The TV tree, shared-token AR and residual grouping have no packaged generator; banks they produced can be imported. The reported experimental results do not depend on these adapters, because every reported bank was generated by the recorded implementation used in the source study (Appendix B.3).

I.3 CORRECTNESS CHECKS

Six checks of different objects are kept separate.

Check	Coverage	Result	Evidence
Independent recomputation of the published cells	All 504 online-GRPO report cells from the 72 frozen banks and the frozen cache, by a separate implementation of Algorithm 1	Bank hashes, support counts, scorer-call counts, means and covariances all match	Appendix D.5
Released-package regression	The same 504 cells, plus the 126 residual-grouping reports (18 banks $\times$ 7 targets) and their 18 joint reports, evaluated through the plan–acquire–recover path implemented in the package	Bank hash, cached, mean and mean-plus-covariance counts and one score access per required identity match in every cell; means and covariances agree within 10^{-12} with both the published cell and full scoring, largest observed error 4.3×10^{-19}	Archive validation records
Package test suite	81 unit and integration tests	All pass with the optional JSON-schema validator installed; without it three schema-shape tests are *skipped*, not passed	Archive validation records
Acceptance fixtures	Twelve synthetic arithmetic cases: all-copy banks, a unique update, an inactive column, mirror draws, a repeated pair, an absolute anchor, a zero-contribution draw, exact rational cancellation, a tiny non-zero coefficient, an all-zero report, a single draw and a three-draw case	Each checked by an independent arithmetic routine against its expected support, mean and ordinary covariance	Archive validation records
Main-text numbers	The 32 audited claims and every prose number	26 reproduce exactly as stated, four under a scope the text now states explicitly, and two simulation-design statements were matched to frozen records without re-running the simulations; none contradicted	Supplementary numeric audit
Source-study checks	The finite enumerations of Appendix A.5 and the dense audits of the original analysis	Reported as run in the source study; *not* re-run here	Appendix A.5

The first two rows describe checks of the same cells using separate implementations: one verifies the reported numerical results, and the other validates the released package. Both checks were rerun on a CPU using the archived source for this paper. The validation records in the archive separately list passed, failed and skipped cases and document the environment for each run.

I.4 Artifact map and regeneration

A frozen snapshot, not a moving join. The analysis inputs are a snapshot, not a live query. The refresh procedure first copies the result database using the SQLite backup API within a read transaction. It records the ledger hash, all frozen generation and protocol hashes, accepted generation audits, the judge model and fingerprint, and the sampling grid. It refuses a changed input hash, incomplete generation, duplicate draw keys, missing root indices, or an incomplete four-shard repetition, rather than repairing any of them. Successful new responses are checked against their returned identities, and an imported response must match the same version in the historical ledger, which is what keeps historical judge versions separate without renaming them retrospectively.

Artifact map. The supplementary archive maps every reported number to a file. Canonical result tables preserve the frozen outputs unchanged. Each record has a verified natural key within its table and a source-record index that specifies that key. The key comprises model family, bank, output cap, sampler, budget, repetition, training seed, target and judge, with acquisition method and budget fraction added for allocation analyses. Each row also retains provenance fields for the experiment, checkpoint regime, bank hash where recorded, prompt set, score version, statistic and cost convention. These provenance fields are not part of the key. Sampler names are mapped explicitly across source files, and tests guard against merging records that differ in family, cap, bank, fraction or repetition. A claims table lists each main-text number with its recomputed value, scope and source path; a numeric-audit report records every scope correction. One command regenerates all empirical figures in the main text and appendix, the main workflow table, the report-target table in Appendix E.1 and Table 5 from the canonical tables and frozen auxiliary inputs, writes every plotted and tabulated value with its source records to a manifest, and logs every CNY-to-USD conversion (Appendix G.2). The method diagrams have native editable draw.io sources and separately verified exports; the plotting command copies these exports without editing them. The bundle retains complete CSV files for all replaced tables, a table-to-figure map and input hashes. No GPU computation or judge call is needed to regenerate any reported support, precision or cost value.