Towards Reliable LLM Evaluation: Correcting the Winner's Curse in Adaptive Benchmarking

Jiefu Zhang[*]
Purdue University
zhan4018@purdue.edu

Yang Xu[*]
Purdue University
xu1720@purdue.edu

Haixiang Sun
Purdue University
sun1321@purdue.edu

Zihan Zhou
Johns Hopkins University
zzhou150@jh.edu

Tianyu Cao
Purdue University
cao357@purdue.edu

Vaneet Aggarwal
Purdue University
vaneet@purdue.edu

Abstract

Adaptive prompt and program search makes LLM evaluation selection-sensitive. Once benchmark items are reused inside tuning, the observed winner's score need not estimate the fresh-data performance of the full tune-then-deploy procedure. We study conditional post-search inference under explicit tuning budgets. We propose SIREN, a selection-aware repeated-split reporting protocol that freezes the post-search shortlist, separates splitwise selection from held-out evaluation, and uses an item-level Gaussian multiplier bootstrap for uncertainty quantification. In a fixed-shortlist regime with smooth stabilized selection, the estimator admits a first-order item-level representation, and the bootstrap yields valid simultaneous inference on a finite budget grid. This supports confidence intervals for procedure-performance curves and pre-specified equal-budget and cross-budget comparisons. Controlled simulations and MMLU-Pro tuning experiments show that winner-based reporting can be optimistic and can change deployment conclusions, while SIREN remains close to the finite-sample reporting target. Codes are available at https://github.com/jznmsl/siren.

1 Introduction

Large language models (LLMs) have become a central tool for reasoning, coding, question answering, and related language-based tasks [26, 3, 16]. As their capabilities have improved, the question of how to evaluate them has become correspondingly more important. More concretely, for a given task and a collection of models, we would like an evaluation scheme whose reported scores remain informative about how those systems actually perform on that task. Even the simplest setting, where each model is evaluated using one fixed configuration chosen in advance, can be fragile and prone to bias. Recent work shows that changing this template can change both absolute performance and relative rankings across models [25, 28]. A growing line of LLM systems goes beyond this fixed system setting. Instead of fixing one prompt or one program configuration in advance, these methods use benchmark examples to search for a stronger way of using the same base model. Concretely, they may generate several candidate instructions or program variants, evaluate them on development examples, and then refine or select among them under an iterative optimization loop or an explicit budget constraint [35, 19, 29, 27, 4]. Once evaluation is used in this way, the concrete benchmark example, which is a task instance together with the scoring criteria, no longer serves only as a measuring device. It also becomes part of the search process that determines which configuration is ultimately reported. Throughout the paper, a system j is a fixed base model together with fixed pipeline components,

[*]Equal contribution.

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

and an artifact $a \in \mathcal{A}_j$ is a deployable configuration. We use tuning for the full procedure that searches over $\mathcal{A}_j$ and returns an artifact, and prompt/program search for the generation, evaluation, refinement, and selection of candidates within tuning. We call this setting adaptive benchmarking because benchmark examples influence which artifact is returned and reported. Thus, tuning searches over artifacts for each fixed system, while $j \in \{1, \ldots, N\}$ indexes the systems being compared.

This reuse of benchmark examples creates an immediate ambiguity about what the final reported number is meant to represent. One possibility is to report the artifact that happened to score highest on the observed development examples, where an artifact refers to a concrete configuration of the system, such as an instruction, a set of demonstrations, or a larger language model program configuration. The alternative is to study the full tune-then-deploy procedure: start from a base system, run a tuner with budget B to search over artifacts, and deploy the returned artifact on future task instances. After searching ends, the chosen artifact is the one actually used at test time. Evaluation must therefore distinguish two questions. A retrospective question asks which candidate happened to look best on the finite sample used during searching. An operational question asks something different: if we rerun the same tuning rule with budget B and then use its returned artifact on fresh data, what performance should we expect on average. These questions coincide when no search is performed, because then the reported system is fixed before any benchmark examples are observed. They diverge once benchmark examples are reused inside the tuning loop, because the final reported artifact is itself data dependent. We refer to this gap between the ex-post winning artifact and the fresh-data performance of the budget-B tune-then-deploy procedure as the winner's curse in adaptive benchmarking. This distinction leads us to treat the full tune-then-deploy procedure as the object of evaluation. The corresponding target is a budget-indexed procedure-performance curve: for each budget B, the expected fresh-data performance of the artifact returned by the budget-B tuner. The target is therefore not the artifact that happened to win on one observed sample, but the fresh-data performance of the rule that produced it.

Recent work makes this problem concrete, but does not yet fully solve it. Adaptive prompt and program optimization methods study how to search for stronger prompts or program configurations under iterative optimization loops or limited evaluation budgets [35, 19, 29, 27, 4]. A separate line studies prompt variation and statistically principled evaluation [25, 28, 22, 40]. These papers are closely related, but they target different statistical objects: robustness to prompt choice, efficient evaluation over a pre-specified prompt pool, stochastic evaluation under meaning-preserving perturbations, or prompt choice under controlled risk guarantees. We ask a different question. Specifically, after a budgeted tuner has completed its search and frozen a finite shortlist, what can be reported about fresh-data performance under the specified post-search reporting rule?

Motivated by the above questions, we propose SIREN: **S**election-aware **I**nference for **R**epeated-split **E**valuatio**N**. SIREN is a post-search reporting protocol for a tune-then-deploy procedure. It begins after the upstream tuning stage has ended, treats the retained candidate artifacts as a frozen shortlist, and uses a separate evaluation pool for reporting. The formal guarantees are therefore conditional on the completed search and the fixed shortlist, rather than on resampling an unrestricted search process. SIREN repeatedly partitions the evaluation pool into a scoring part and a held-out part. In each split, the scoring part determines stabilized weights over retained artifacts, while the held-out part evaluates the selected output on fresh items. Aggregating across splits preserves the selection step while separating selection evidence from final performance evaluation. This is the reporting layer analyzed in Section 3.

We validate SIREN both theoretically and empirically. In the fixed-shortlist, smooth stabilized-selection regime, we prove that the selected repeated-split estimator has a first-order item-level expansion and that a Gaussian multiplier bootstrap yields confidence intervals, simultaneous bands over tuning budgets, and pre-specified pairwise comparisons, without rerunning the tuner inside every resample. Controlled studies verify calibration and expose same-data winner optimism, while MMLU-Pro experiments across random search, DSPy, and eleven open-weight models show that selection-aware reporting can change tuned-versus-default conclusions and the interpretation of budget-aware prompt evaluation.

Key contributions.

- We formulate adaptive LLM benchmarking as inference for budget-indexed tune-then-deploy procedures, and distinguish this broader objective from the conditional post-search target studied by SIREN.

- We propose SIREN, a selection-aware repeated-split evaluation protocol based on a frozen shortlist, splitwise scoring and held-out evaluation, and stabilized selection.
- We prove that the selected repeated-split estimator admits a per-item asymptotic representation and that a Gaussian multiplier bootstrap yields valid simultaneous inference on a finite grid of tuning budgets, including equal-budget and cross-budget comparisons fixed in advance.
- We validate SIREN in controlled studies and real LLM tuning experiments, showing that selection-aware procedure-level evaluation can change conclusions about system choice and the reliability of budget-aware reporting after adaptive tuning.

2 Problem formulation

We first define the target of interest. The setup covers prompt search, demonstration selection, and language-model program optimization. Let P denote the distribution of a task instance together with the reference information for scoring. A system $j \in \{1, \dots, N\}$ is a base model plus fixed pipeline components, with $\mathcal{A}_j$ denoting its artifact space; an artifact $a \in \mathcal{A}_j$ is a deployable configuration such as instructions, demonstration sets, decoding choices, or larger language-model programs.

For a generic item $W \sim P$, artifact $a \in \mathcal{A}_j$, and execution randomness ξ, let $Z_j(W, a, \xi) \in [0, 1]$ denote the per-item score, and define

$$\mu_j(w, a) = \mathbb{E}_\xi \big[Z_j(W, a, \xi) \mid W = w \big] \tag{1}$$

as the item-conditional mean score. Any deterministic evaluation is the special case with no ξ. For system j and tuning budget B, let $D = (\widetilde{W}_1, \dots, \widetilde{W}_n)$ be the development sample of task instances used by the tuner, and let U_j collect the tuner's internal randomness. We model the budget-B tuner as a map from the development sample to a deployable artifact:

$$\widehat{a}_j(B; D) = T_{j,B}(D, U_j) \in \mathcal{A}_j, \tag{2}$$

where B records the cost of the tuner. A fixed baseline is the special case in which the returned artifact is constant in D. We further define the procedure-level target as

$$\theta_j(B) = \mathbb{E}\big[\mu_j(W^{\mathrm{new}}, \widehat{a}_j(B; D))\big], \tag{3}$$

where $W^{\mathrm{new}} \sim P$ is independent of the development sample D; the expectation is over D, U_j, and W^{new}. Thus $\theta_j(B)$ is the fresh-data performance of the rule "use D to run the budget-B tuner, then deploy its returned artifact," not the score of one ex post winner on the sample used to choose it. Our goal is to estimate these procedure-level performances on a finite, pre-specified budget grid $\mathcal{B} = \{b_1, \dots, b_L\}$, where $B \in \mathcal{B}$ denotes a generic tuning budget. Section 3 analyzes a conditional post-search target with the completed tuning stage and shortlist fixed. Its soft deployment rule need not have the same target as the tuner-returned artifact in (3); Appendix D.4 examines their relationship empirically.

3 SIREN

SIREN, short for Selection-aware Inference for Repeated-split EvaluatioN, is our approach for reliably evaluating a system together with its tuner. The task instances used by SIREN are separate from those used by the tuner. During tuning, a series of searches are performed to produce a finite set of candidate artifacts. SIREN then takes this frozen candidate set and an independent finite collection of benchmark items. For example, the benchmark items pool may consist of question-answer pairs with reference answers, summarization inputs with reference summaries or judging criteria, or coding problems with unit tests. This pool is the finite benchmark used to approximate the fresh deployment tasks represented by W^{new} in (3). SIREN repeatedly splits this benchmark pool into a score subset and a held-out subset. In each split, the score subset is used only to set selection weights over the frozen candidates, while the held-out subset is used to evaluate the selected or weighted output after that choice has been made. SIREN then aggregates the resulting scores across splits and uses a Gaussian multiplier bootstrap to form confidence intervals.

3.1 Protocol

We first describe one fixed system j and budget B, temporarily suppressing these indices. The frozen shortlist is $a_1, \dots, a_K$, with $K = K_{j,B}$, and the evaluation pool is $W_1, \dots, W_M$. In split r, D_r^{score}

scores artifacts for selection, while E_r evaluates them after the artifact weights have been formed; the score vectors are $\widehat{S}_r$ and $\widehat{T}_r$. Restoring the indices, $\mathcal{A}_j(B) = \{a_{j,B,1}, \ldots, a_{j,B,K_{j,B}}\}$. SIREN only scores and reweights these retained artifacts; it does not generate new ones.

For any task instance w in the support of P and any weight vector $q \in \Delta^{K_{j,B}}$, define the weighted mean score of the shortlist as $\mu_j(w, q; B) = \sum_{k=1}^{K_{j,B}} q_k \, \mu_j(w, a_{j,B,k})$. SIREN evaluates the retained shortlist on the separate evaluation pool $W_1, \ldots, W_M \overset{\text{i.i.d.}}{\sim} P$ via a repeated-split procedure. Concretely, denote R as the fixed number of repeated splits, for each $r \le R$, let D_r^{score} and E_r be disjoint subsets of $\{1, \ldots, M\}$, where the scoring subset D_r^{score} is used only to score the retained artifacts for selection and the held-out subset E_r is used only for held-out evaluation after the splitting has been made in each round. We denote the sigma-field generated by the repeated-split procedure as $\mathcal{G}_M = \sigma\big(\{D_r^{\text{score}}, E_r\}_{r=1}^R\big)$. After the subsets are partitioned, for artifact $a_{j,B,k}$, we compute its selection scores $\widehat{S}_{r,j,k}$ and held-out scores $\widehat{T}_{r,j,k}$ on split r as

$$\widehat{S}_{r,j,k}(B) = \frac{1}{|D_r^{\text{score}}|} \sum_{i \in D_r^{\text{score}}} Z_j\big(W_i, a_{j,B,k}, \xi_{irjk}^S\big), \tag{4}$$

$$\widehat{T}_{r,j,k}(B) = \frac{1}{|E_r|} \sum_{i \in E_r} Z_j\big(W_i, a_{j,B,k}, \xi_{irjk}^T\big), \tag{5}$$

and write $\widehat{S}_{r,j}(B)$ and $\widehat{T}_{r,j}(B)$ as the corresponding stacked vectors in $\mathbb{R}^{K_{j,B}}$. The variables ξ_{irjk}^S and ξ_{irjk}^T capture execution randomness.

SIREN next uses a selector to convert the selection scores into weights for each split, and combines those weights with the held-out scores and form a weighted average. This aims to reduce dependence on any single partition while preserving the separation between selection and evaluation. We denote a selector as a map $g_{j,B} : \mathbb{R}^{K_{j,B}} \to \Delta^{K_{j,B}}$ that converts the above selection scores into deployment weights. In the main experiments, the selector is ordinary softmax, $[g_{j,B}(s)]_k = \exp(s_k/\tau)/\sum_{\ell=1}^{K_{j,B}} \exp(s_\ell/\tau)$, with $\tau = 1$. Smaller τ concentrates more weight on the highest-scoring artifact, while larger τ spreads the weights. The theory permits any smooth selector satisfying the stated conditions.

The split-level selected output is denoted as $\widehat{q}_{r,j}(B) = g_{j,B}\big(\widehat{S}_{r,j}(B)\big)$, and the corresponding weighted-average of the held-out score is the form of $\widehat{Y}_{r,j}(B) = \widehat{q}_{r,j}(B)^\top \widehat{T}_{r,j}(B)$. This is the expected score of a randomized single-artifact rule that deploys artifact k with probability $\widehat{q}_{r,j,k}(B)$; each realized deployment uses one artifact. For computing the final estimator, given nonnegative weights $\omega_1, \ldots, \omega_R$ with $\sum_{r=1}^R \omega_r = 1$, the SIREN point estimator is

$$\widetilde{\theta}_{j,R}(B) = \sum_{r=1}^{R} \omega_r \, \widehat{Y}_{r,j}(B), \tag{6}$$

Here $\widehat{q}_{r,j}(B)$ weights artifacts within split r, whereas ω_r weights the split-level estimates. We use $\omega_r = 1/R$ in the experiments; weights proportional to $|E_r|$ are also allowed. The next subsection describes the Gaussian multiplier bootstrap used to form confidence intervals and simultaneous bands.

3.2 Bootstrap

Once the estimator in (6) is obtained, SIREN asks how variable this estimator would be if the evaluation pool were redrawn while the completed tuning stage and repeated-split design were held fixed. To formalize this target, we define the average score that the frozen shortlist, repeated-split protocol, and selector would produce under the realized split design:

$$\theta_{j,M}^{\text{RS}|\mathcal{G}}(B) = \mathbb{E}\big[\widetilde{\theta}_{j,R}(B) \mid \mathcal{G}_M\big]. \tag{7}$$

The bootstrap therefore aims to characterize the uncertainty gap $\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\text{RS}|\mathcal{G}}(B)$. This gap is not a simple sample-mean error because each benchmark item can affect the estimator in two ways: through held-out evaluation scores and through scoring-subset scores that change the splitwise selection weights. Fix j, B and suppress these indices. Let $\widehat{\Gamma}_{ir}^T$ denote the centered artifact-score vector for item i in split r, scaled by $M/|E_r|$ when $i \in E_r$ and zero otherwise; define $\widehat{\Gamma}_{ir}^S$ analogously

on D_r^{score}. The computable item-level contribution is

$$\widehat{\psi}_i = \sum_{r=1}^{R} \omega_r \left[\widehat{q}_r^{\top} \widehat{\Gamma}_{ir}^{T} + \widehat{T}_r^{\top} Dg(\widehat{S}_r) \widehat{\Gamma}_{ir}^{S} \right]. \tag{8}$$

The first term is the direct held-out evaluation effect, and the second is the scoring-subset effect through the selector derivative. The complete indexed definitions are given in Appendix B. Intuitively, $\widehat{\psi}_{i,j,B,M}$ estimates the summand in the linear approximation $\sqrt{M}\{\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)\} \approx \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \psi_{i,j,B,M}$. For each (j, B), define the empirical average contribution $\bar{\widehat{\psi}}_{j,B,M} = \frac{1}{M} \sum_{i=1}^{M} \widehat{\psi}_{i,j,B,M}$, SIREN then draws independent Gaussian multipliers $\zeta_1, \ldots, \zeta_M \overset{\text{i.i.d.}}{\sim} N(0,1)$ and forms the joint bootstrap process $\widehat{\mathbf{G}}_M^* = \left\{ \widehat{G}_{j,B}^* \right\}_{j \le N,\ B \in \mathcal{B}}$ with $\widehat{G}_{j,B}^* = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \zeta_i \big(\widehat{\psi}_{i,j,B,M} - \bar{\widehat{\psi}}_{j,B,M} \big)$. This randomly reweights benchmark items rather than splits or candidate artifacts. Using the same multiplier ζ_i for item W_i across all systems and budgets preserves the dependence created by evaluating the same benchmark pool across the whole grid.

The conditional distribution of $\widehat{\mathbf{G}}_M^*$ is used as an approximation to the sampling distribution of $\left\{ \sqrt{M} \big(\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B) \big) \right\}_{j \le N,\ B \in \mathcal{B}}$. Pointwise intervals use the corresponding coordinatewise bootstrap quantiles, while simultaneous bands use the quantile of $\max_{j \le N,\ B \in \mathcal{B}} |\widehat{G}_{j,B}^*|$. The same bootstrap process also supports fixed equal-budget comparisons, cross-budget gains, and other pre-specified linear contrasts on the grid. The theory below justifies this item-level resampling scheme for the selected repeated-split estimator, including both held-out evaluation noise and the first-order effect of selection.

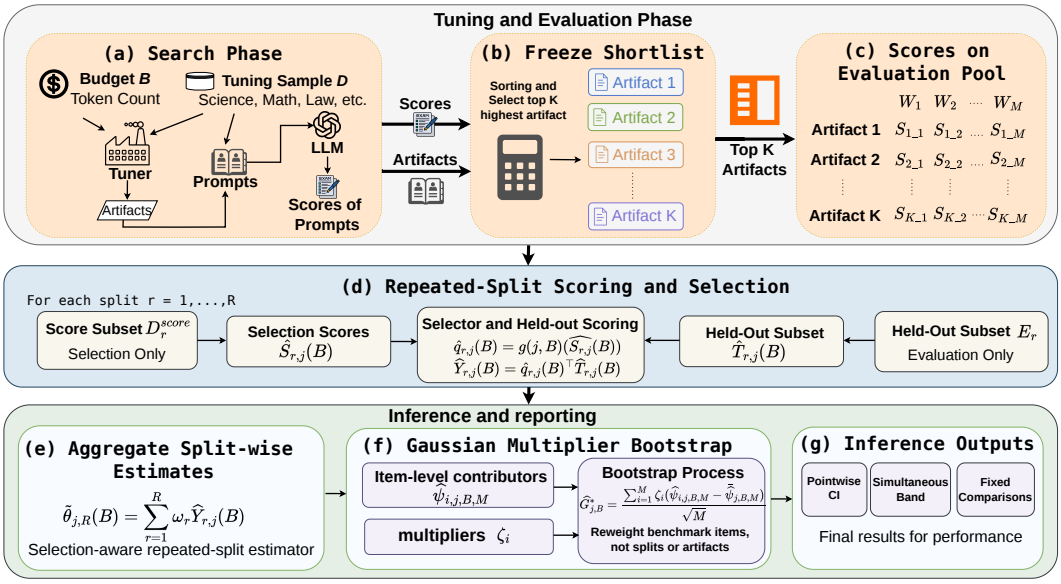

Figure 1: Overview of SIREN. Panels (a)-(b) show tuning with budget B and freezing the shortlist for one system j; panel (c) shows the per-item scores of its retained artifacts on evaluation items $W_1, \ldots, W_M$; panels (d)-(e) show scoring, held-out evaluation, and aggregation for splits $r = 1, \ldots, R$; panels (f)-(g) stack the item-level contributions across all N systems and budgets for bootstrap inference.

3.3 Theoretical guarantees

We now justify the two statistical outputs of SIREN: the repeated-split estimate and the bootstrap confidence interval. The formal assumptions, full details of SIREN's computation steps, and proofs are given in Appendix B. All statistical statements are conditioned on the completed tuning stage, so the retained shortlists and selectors are treated as fixed independent objects in SIREN's pipeline.

In the following theorem-ready regime, we consider the number of systems, budgets, repeated splits, and retained artifacts are fixed; split sizes are proportional to M; and the selector $g_{j,B}$ is a stabilized smooth map with uniformly bounded derivatives. Under these settings, the softmax-type selectors are covered directly, while a discontinuous hard argmax requires smoothing or an additional margin condition. We first define the joint scaled error over the finite system-budget grid as

$$\mathbf{Z}_M = \left\{ \sqrt{M}\big(\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)\big) \right\}_{j \le N,\ B \in \mathcal{B}}.$$

Theorem 1 (Selection-aware CLT)**.** *Conditioned on the repeated-split design, the joint error vector $\mathbf{Z}_M$ admits a first-order item-level linearization and converges to a mean-zero Gaussian law over the budget grid $\mathcal{B}$. The item-level linearization has one summand per benchmark task instance and includes both sources of uncertainty: the direct held-out evaluation effect and the first-order effect of using scoring subsets to choose splitwise weights.*

For each item W_i, the proof characterizes all of its impact across splits, systems, budgets, and retained artifacts into one item-level vector. These vectors are independent across items, although their coordinates are dependent within an item because the same task can appear in multiple split-level comparisons. A finite-dimensional CLT then applies to the full score table. The nonstandard step is the selection layer: the splitwise weights are chosen from the scoring subset rather than fixed in advance. Smoothness of $g_{j,B}$ lets us linearize how scoring-set perturbations change these weights, producing a derivative-weighted selection term in addition to ordinary held-out evaluation noise. This is the uncertainty ignored by winner-based reports that treat the selected artifact as fixed.

Theorem 1 explains what SIREN is estimating and motivates the repeated-split construction. The target is not the score of a single observed prompt, program, or demonstration set chosen after search, it is instead the performance of a frozen artifact shortlist plus a specified reporting rule. The theorem shows that this selected reporting rule still has a tractable large-sample distribution once the effect of selection is included. The following theorem further explains the validity of the bootstrapping approach.

Theorem 2 (Multiplier bootstrap validity)**.** *Suppose the computable item-level quantities used by the bootstrap consistently approximate the corresponding first-order item effects in the linearization, in the sense that their average squared discrepancy over benchmark items vanishes as M grows. Then the conditional law of the multiplier process $\widehat{\mathbf{G}}_M^*$ consistently approximates the joint law of $\mathbf{Z}_M$. Consequently, the bootstrap quantiles used in Section 3 yield asymptotically valid pointwise confidence intervals and simultaneous bands for $\theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)$ over the budget grid $\mathcal{B}$.*

Theorem 2 gives the computational payoff. SIREN resamples benchmark-item contributions rather than rerunning the tuner or regenerating candidates. Because each item contribution already encodes both evaluation noise and selection effects, one multiplier per item preserves dependence across systems, budgets, and splits. Thus the same bootstrap supports confidence intervals, simultaneous bands, and fixed grid contrasts. Conceptually, the condition in Theorem 2 requires the computable item-level contributions to recover the same two first-order effects from Theorem 1: the direct held-out score effect and the derivative-weighted selection effect. The proof compares the feasible multiplier process with the infeasible linear expansion; conditional on the observed contributions, the bootstrap is Gaussian with empirical covariance, and plug-in consistency makes this covariance match the limiting covariance of $\mathbf{Z}_M$.

4 Experiments

Adaptive LLM benchmarking is vulnerable to a winner's curse: when benchmark items are reused during prompt or program search, the reported winner can overstate the fresh-data performance of the full procedure. Our experiments evaluate whether this bias appears after real tuning and whether SIREN effectively estimates its conditional post-search target. We organize these studies around the following three Research Questions (**RQ**s):

(RQ1) Does SIREN estimate the procedure-level target without winner-induced optimism?

(RQ2) Does selection-aware reporting change tuned-versus-default deployment decisions?

(RQ3) Is budget-aware prompt evaluation over a pool sufficient after adaptive tuning, or is selection-aware inference still needed?

We evaluate SIREN on MMLU-Pro [33] *Math* and *Law* across two tuner families, random search and DSPy, and eleven open-weight instruction-tuned LLMs. For each subject, the tuner uses a small training pool to generate candidate artifacts under the shared token-budget grid $\mathcal{B} = \{500\text{K}, 1.5\text{M}, 3\text{M}, 6.5\text{M}\}$, and the reporting layer uses a disjoint evaluation pool to estimate procedure-level performance. All methods are compared on the same retained shortlist and evaluation pool, differing only in the final reporting rule. Unless otherwise stated, SIREN uses R=10 repeated splits, scoring fraction ρ_{score}=0.5, softmax selector temperature τ=1, uniform split weights ω_r, and 2,000 Gaussian multiplier bootstrap draws.

We compare SIREN with four internal baselines:

- **M1/M2** (*naive max / naive max + Wald*) evaluate every retained artifact on the full evaluation pool and use the same selected-winner point estimate, $\widehat{\theta}^{\text{M1}} = \max_k \bar{Z}_{\cdot,k}$. M2 differs only by adding a conventional Wald interval computed after selection. We combine M1 and M2 in point-estimate comparisons and distinguish them only when evaluating confidence intervals;
- **M3** (*single-split holdout*) partitions the evaluation set into a selection fold and a held-out scoring fold, picks the empirical winner on the selection fold, and reports its mean accuracy on the held-out fold;
- **M4** (*R-split argmax with Student-t CI*) repeats the selection-and-evaluation procedure R times with independent splits, takes the average of the per-split held-out means as the point estimate, and forms a Student-t confidence interval from the variance across splits.

Appendix C.1 provides controlled validation of the inference mechanism. Study A checks multiplier-bootstrap calibration and the predicted $M^{-1/2}$ width scaling. Study B shows why smooth selection matters near ties: hard-argmax coverage falls to 89.8%, whereas softmax coverage remains between 93.7% and 96.0% across the quality-gap sweep. Study C shows that same-data best-of reporting becomes more optimistic as the number of searched artifacts grows. We also compare against **PromptEval** [28], an IRT-based method for efficient evaluation over a prespecified prompt pool. It observes a fraction f of the $K \times M$ prompt–item score matrix, fits a Rasch model, and imputes the remaining cells. For the Math/Law common-reference comparisons, $\theta^\star$ is computed from $100,000$ random permutations and complementary splits of the fixed observed score matrix. This finite-matrix reference is distinct from the fresh-item target in (7). In the original comparisons, "bias" denotes the signed difference from this common soft-rule reference. For M1–M4, it can include both estimation error and differences between reporting rules. Appendix D.2 separately compares each method with its own target. We note that all models are served via vLLM on a single NVIDIA RTX 4090 GPU (24 GB VRAM) with sequential model loading, and the interval calibration is evaluated separately.

4.1 RQ1: Procedure-level target and winner-induced bias

We first answer **RQ1** by comparing each reported tuned estimate $\widehat{\theta}_A$ with the Monte Carlo reference $\theta^\star_A$. Each cell is a (subject, tuner, model, budget) configuration on the grid described above, with the model ranging over the 11 open-weight models that appear in all four subject vs tuner combinations. We report three summaries. The column "bias" is $\widehat{\theta}_A - \theta^\star_A$, in percentage points; positive values indicate scores above the common reporting reference. The column "dir" counts the number of cells in which the point-estimate ordering agrees with the Monte Carlo ordering $\text{sign}(\widehat{\theta}_A - \theta^\star_B) = \text{sign}(\theta^\star_A - \theta^\star_B)$. The column "MAE" averages $|\widehat{\theta}_A - \theta^\star_A|$ across cells, in percentage points. Exact ties with the default count as incorrect directions. Tables 1 and 2 bold the best direction count, signed bias closest to zero, and smallest MAE, including ties at the displayed precision. Per-model tables bold estimates closest to the reference by the same precision rule.

Tables 1 and 2 show a consistent positive bias for winner-based reporting. M1 is optimistic in all four subject and tuner summaries. M2 has the same point estimate as M1, and therefore inherits the same signed bias; the Wald interval changes only the uncertainty report, not the selected-winner estimate. M3 and M4 reduce direct same-data reuse but retain hard selection, so their differences from the soft-rule reference are not pure estimation bias. SIREN remains close to the Monte Carlo reference across budgets and tuners, with mean signed errors between -0.20 and $+0.08$ percentage points in the four aggregate rows. For **RQ1**, these tables compare proximity to the common reference. Appendix D.2 evaluates accuracy against fresh-data targets for each reporting rule.

Table 1: Directional accuracy and common-reference errors on MMLU-Pro Math.

Tuner	B	M1/M2 dir	M1/M2 bias	M1/M2 MAE	M3 dir	M3 bias	M3 MAE	M4 dir	M4 bias	M4 MAE	SIREN dir	SIREN bias	SIREN MAE
Random search	500K	6/11	+5.19	5.19	8/11	+3.83	3.90	7/11	+4.48	4.48	**11/11**	**−0.09**	**0.13**
	1.5M	5/11	+2.50	2.50	9/11	+1.19	1.59	6/11	+1.84	1.84	**10/11**	**+0.01**	**0.13**
	3M	4/11	+2.49	2.49	9/11	+0.65	1.17	8/11	+1.45	1.45	**10/11**	**+0.01**	**0.14**
	6.5M	6/11	+2.09	2.09	9/11	+0.26	1.16	8/11	+1.21	1.35	**11/11**	**+0.02**	**0.17**
All	*(44 cells)*	21/44	+3.06	3.06	35/44	+1.48	1.95	29/44	+2.25	2.28	**42/44**	**−0.01**	**0.14**
DSPy	500K	9/11	+1.37	1.37	6/11	−0.76	1.80	9/11	+0.55	0.94	**10/11**	**−0.09**	**0.21**
	1.5M	8/11	+1.52	1.52	8/11	−0.54	1.16	10/11	+0.78	1.13	**11/11**	**−0.03**	**0.24**
	3M	8/11	+1.49	1.49	9/11	**+0.00**	1.00	10/11	+0.70	1.01	**11/11**	−0.05	**0.26**
	6.5M	8/11	+1.33	1.33	8/11	−0.17	0.90	10/11	+0.63	0.92	**11/11**	**−0.03**	**0.24**
All	*(44 cells)*	33/44	+1.43	1.43	31/44	−0.37	1.21	39/44	+0.67	1.00	**43/44**	**−0.05**	**0.24**

Table 2: Directional accuracy and common-reference errors on MMLU-Pro Law.

Tuner	B	M1/M2 dir	M1/M2 bias	M1/M2 MAE	M3 dir	M3 bias	M3 MAE	M4 dir	M4 bias	M4 MAE	SIREN dir	SIREN bias	SIREN MAE
Random search	500K	4/11	+1.50	1.50	3/11	+2.13	2.13	8/11	+1.00	1.06	**11/11**	**+0.05**	**0.31**
	1.5M	2/11	+2.24	2.24	5/11	+1.59	2.49	5/11	+1.37	1.68	**10/11**	**+0.11**	**0.25**
	3M	4/11	+2.10	2.10	6/11	+1.47	2.11	6/11	+1.18	1.48	**11/11**	**+0.07**	**0.22**
	6.5M	5/11	+1.77	1.77	6/11	+1.74	1.92	8/11	+1.13	1.21	**10/11**	**+0.10**	**0.22**
All	*(44 cells)*	15/44	+1.90	1.90	20/44	+1.73	2.16	27/44	+1.17	1.36	**42/44**	**+0.08**	**0.25**
DSPy	500K	**10/11**	+0.77	0.77	8/11	**−0.01**	1.25	8/11	−0.15	0.61	9/11	−0.21	**0.34**
	1.5M	10/11	+1.10	1.10	9/11	+0.72	1.15	**11/11**	+0.25	0.63	10/11	**−0.17**	**0.33**
	3M	10/11	+1.13	1.13	9/11	+0.77	1.56	**11/11**	+0.22	0.44	10/11	**−0.20**	**0.34**
	6.5M	10/11	+1.04	1.04	9/11	+0.25	1.37	10/11	**−0.01**	0.61	**11/11**	−0.21	**0.34**
All	*(44 cells)*	**40/44**	+1.01	1.01	35/44	+0.43	1.33	**40/44**	**+0.08**	0.57	**40/44**	−0.20	**0.34**

4.2 Comparison against reporting baselines M1–M4

We next answer **RQ2** by examining whether the reporting differences in Tables 1-2 change the tuned-versus-default conclusion. For each model and budget, we compare the sign of the reported gap $\mathrm{sign}(\widehat{\theta}_A - \theta^\star_B)$ with the ground-truth direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$. Table 3 gives one representative comparison on MMLU-Pro Math with the DSPy tuner for eleven models shown as columns. The default reference $\theta^\star_B$ is fixed across budgets and is shown at the top row. For each budget, $\theta^\star_A$ is the Monte Carlo tuned-procedure reference, and each method row reports a score $\widehat{\theta}_A$ compared with this common reference. The ✓/✗marker records whether $\mathrm{sign}(\widehat{\theta}_A - \theta^\star_B)$ matches $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$.

Table 3: Unified per-model comparison across budgets on MMLU-Pro Math with the **DSPy** tuner.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.373	0.263	0.306	0.222	0.196	0.249	0.257	0.249	0.255	0.169	0.130
$500K$	$\theta^\star_A$	0.379	0.267	0.339	0.230	0.161	0.253	0.258	0.283	0.254	0.165	0.140
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	M1/M2	0.388 ✓	0.285 ✓	0.349 ✓	0.240 ✓	0.196 ✗	0.265 ✓	0.265 ✓	0.322 ✓	**0.257** ✗	0.168 ✓	0.146 ✓
	M3	0.369 ✗	0.281 ✓	0.345 ✓	0.216 ✗	0.177 ✓	0.232 ✗	0.217 ✗	0.304 ✓	0.232 ✓	0.152 ✓	0.121 ✗
	M4	0.382 ✓	0.274 ✓	0.343 ✓	**0.230** ✓	0.196 ✗	0.262 ✓	0.253 ✗	0.306 ✓	0.245 ✓	0.160 ✓	0.138 ✓
	SIREN	**0.380** ✓	**0.264** ✓	**0.342** ✓	**0.230** ✓	**0.162** ✓	**0.254** ✓	**0.256** ✗	**0.278** ✓	0.249 ✓	**0.164** ✓	**0.139** ✓
$1.5M$	$\theta^\star_A$	0.383	0.278	0.344	0.251	0.193	0.254	0.262	0.290	0.255	0.166	0.142
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	M1/M2	0.398 ✓	0.305 ✓	0.350 ✓	0.282 ✓	0.214 ✗	0.262 ✓	0.267 ✓	0.325 ✓	**0.257** ✗	0.174 ✗	0.149 ✓
	M3	**0.382** ✓	0.267 ✓	0.343 ✓	0.257 ✓	0.198 ✗	0.222 ✗	0.240 ✗	0.312 ✓	0.238 ✓	0.168 ✓	0.133 ✓
	M4	0.396 ✓	0.295 ✓	**0.344** ✓	0.274 ✓	0.211 ✗	0.262 ✓	0.258 ✓	0.316 ✓	0.248 ✓	0.163 ✓	0.137 ✓
	SIREN	0.387 ✓	**0.275** ✓	0.348 ✓	**0.250** ✓	**0.193** ✓	**0.257** ✓	**0.260** ✓	**0.284** ✓	**0.253** ✓	**0.165** ✓	**0.143** ✓
$3M$	$\theta^\star_A$	0.384	0.279	0.342	0.251	0.193	0.250	0.264	0.294	0.254	0.164	0.142
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	M1/M2	0.398 ✓	0.305 ✓	0.349 ✓	0.285 ✓	0.211 ✗	0.263 ✓	0.269 ✓	0.321 ✓	0.257 ✗	0.176 ✗	0.149 ✓
	M3	**0.382** ✓	0.302 ✓	**0.342** ✓	0.262 ✓	**0.193** ✓	0.246 ✗	0.240 ✗	0.310 ✓	0.241 ✓	0.169 ✓	0.133 ✓
	M4	0.396 ✓	0.293 ✓	**0.342** ✓	0.275 ✓	0.205 ✗	0.266 ✓	0.259 ✓	0.307 ✓	0.246 ✓	0.168 ✓	0.139 ✓
	SIREN	0.389 ✓	**0.275** ✓	0.347 ✓	**0.249** ✓	0.194 ✓	**0.252** ✓	**0.263** ✓	**0.287** ✓	**0.252** ✓	**0.163** ✓	**0.143** ✓
$6.5M$	$\theta^\star_A$	0.384	0.274	0.343	0.252	0.193	0.252	0.263	0.302	0.255	0.169	0.143
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	M1/M2	0.398 ✓	0.293 ✓	0.349 ✓	0.284 ✓	0.208 ✗	0.263 ✓	0.269 ✓	0.330 ✓	**0.257** ✗	0.176 ✗	0.149 ✓
	M3	**0.382** ✓	0.288 ✓	**0.342** ✓	0.262 ✓	0.196 ✗	0.246 ✗	0.240 ✗	0.315 ✓	0.238 ✓	**0.169** ✓	0.133 ✓
	M4	0.396 ✓	0.283 ✓	0.345 ✓	0.273 ✓	0.202 ✗	0.266 ✓	0.259 ✓	0.320 ✓	0.248 ✓	0.167 ✓	0.139 ✓
	SIREN	0.389 ✓	**0.270** ✓	0.347 ✓	**0.251** ✓	**0.194** ✓	**0.254** ✓	**0.261** ✓	**0.296** ✓	**0.253** ✓	0.168 ✓	**0.143** ✓

The table shows that winner-based reporting can change the deployment verdict, especially in small-gap cases. Because M1 and M2 use the same selected-winner point estimate, they inherit the same optimistic shift and can turn near ties or weak regressions into apparent gains. M3 and M4 reduce direct reuse of the evaluation pool, but their hard splitwise selections remain sensitive to the realized split. SIREN remains close to the common soft-rule reference, including in near-tie cases. The reporting differences in Tables 1–2 can therefore change which system would be deployed. The matched-target analysis distinguishes estimation error from the choice of reporting rule. We refer to the remaining subject vs tuner cells in Tables 9, 10, and 11 in the appendix. These results answer **RQ2** affirmatively. Selection-aware reporting can change tuned-versus-default conclusions because the same-data winner and the tune-then-deploy procedure are different statistical objects. SIREN evaluates tuned gains under its specified soft-selection rule using held-out splitwise scores.

4.3 Comparison against PromptEval

To answer **RQ3**, we compared PromptEval [28], the closest external baseline to our setting since it also studies budget-aware evaluation over a prompt pool. We sweep PromptEval's cell-observation fraction $f \in \{0.05, 0.10, 0.20, 0.40, 0.60, 0.80, 1.00\}$, where f is the fraction of the $K \times M$ prompt–item score matrix observed before fitting the Rasch IRT model and imputing the remaining cells. Small f corresponds to the sparse budget-constrained regime. When $f{=}1$, PromptEval gives the same selected-winner point estimate as M1 in our comparison. PromptEval addresses efficient evaluation over a fixed prompt pool. SIREN instead begins after an adaptive tuner has frozen a shortlist and accounts for splitwise selection in both the point estimate and its uncertainty.

Table 4: Unified PromptEval [28] vs SIREN comparison on MMLU-Pro Math with the **DSPy** tuner.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.373	0.263	0.306	0.222	0.196	0.249	0.257	0.249	0.255	0.169	0.130
$500K$	$\theta^\star_A$	0.379	0.267	0.339	0.230	0.161	0.253	0.258	0.283	0.254	0.165	0.140
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	PE $f{=}0.05$	0.254 ✗	0.193 ✗	0.267 ✗	0.136 ✗	0.109 ✓	0.180 ✗	0.157 ✗	0.181 ✗	0.164 ✓	0.092 ✓	0.076 ✗
	PE $f{=}0.10$	0.305 ✗	0.204 ✗	0.267 ✗	0.180 ✗	0.115 ✓	0.199 ✗	0.194 ✗	0.227 ✗	0.226 ✓	0.125 ✓	0.117 ✗
	PE $f{=}0.20$	0.401 ✓	**0.267** ✓	0.349 ✓	0.245 ✓	0.164 ✓	0.265 ✓	0.274 ✓	0.305 ✓	0.286 ✗	0.175 ✗	0.162 ✓
	PE $f{=}0.40$	0.399 ✓	0.281 ✓	0.346 ✓	0.252 ✓	0.164 ✓	0.266 ✓	0.270 ✓	0.322 ✓	0.268 ✗	0.183 ✗	0.149 ✓
	PE $f{=}0.60$	0.399 ✓	0.285 ✓	0.346 ✓	0.240 ✓	0.182 ✓	0.265 ✓	0.269 ✓	0.327 ✓	0.267 ✗	0.174 ✗	0.153 ✓
	PE $f{=}0.80$	0.391 ✓	0.286 ✓	0.352 ✓	0.237 ✓	0.192 ✓	0.263 ✓	0.268 ✓	0.326 ✓	0.265 ✗	0.171 ✗	0.149 ✓
	PE $f{=}1.00$ (M1)	0.388 ✓	0.285 ✓	0.349 ✓	0.240 ✓	0.196 ✗	0.265 ✓	0.265 ✓	0.322 ✓	**0.257** ✗	0.168 ✓	0.146 ✓
	SIREN	**0.380** ✓	0.264 ✓	**0.342** ✓	**0.230** ✓	**0.162** ✓	**0.254** ✓	**0.256** ✗	**0.278** ✓	0.249 ✓	**0.164** ✓	**0.139** ✓
$1.5M$	$\theta^\star_A$	0.383	0.278	0.344	0.251	0.193	0.254	0.262	0.290	0.255	0.166	0.142
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	PE $f{=}0.05$	0.271 ✗	0.175 ✗	0.212 ✗	0.174 ✗	0.123 ✓	0.173 ✗	0.168 ✗	0.227 ✗	0.151 ✓	0.102 ✓	0.079 ✗
	PE $f{=}0.10$	0.304 ✗	0.226 ✗	0.265 ✗	0.218 ✗	0.172 ✓	0.205 ✗	0.207 ✗	0.250 ✓	0.209 ✓	0.133 ✓	0.112 ✗
	PE $f{=}0.20$	0.393 ✓	0.299 ✓	0.353 ✓	0.273 ✓	0.224 ✗	0.268 ✓	0.279 ✓	0.321 ✓	0.264 ✗	0.181 ✗	0.151 ✓
	PE $f{=}0.40$	0.403 ✓	0.302 ✓	**0.345** ✓	0.291 ✓	0.229 ✗	0.269 ✓	0.276 ✓	0.318 ✓	0.264 ✗	0.180 ✗	0.151 ✓
	PE $f{=}0.60$	0.403 ✓	0.307 ✓	0.346 ✓	0.278 ✓	0.218 ✗	0.265 ✓	0.273 ✓	0.320 ✓	0.259 ✗	0.171 ✗	0.154 ✓
	PE $f{=}0.80$	0.403 ✓	0.308 ✓	0.349 ✓	0.283 ✓	0.210 ✗	0.262 ✓	0.267 ✓	0.324 ✓	**0.257** ✗	0.170 ✗	0.153 ✓
	PE $f{=}1.00$ (M1)	0.398 ✓	0.305 ✓	0.350 ✓	0.282 ✓	0.214 ✗	0.262 ✓	0.267 ✓	0.325 ✓	**0.257** ✗	0.174 ✗	0.149 ✓
	SIREN	**0.387** ✓	**0.275** ✓	0.348 ✓	**0.250** ✓	**0.193** ✓	**0.257** ✓	**0.260** ✓	**0.284** ✓	**0.253** ✓	**0.165** ✓	**0.143** ✓
$3M$	$\theta^\star_A$	0.384	0.279	0.342	0.251	0.193	0.250	0.264	0.294	0.254	0.164	0.142
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	PE $f{=}0.05$	0.298 ✗	0.181 ✗	0.221 ✗	0.182 ✗	0.129 ✓	0.190 ✗	0.167 ✗	0.229 ✗	0.151 ✓	0.107 ✓	0.067 ✗
	PE $f{=}0.10$	0.326 ✗	0.215 ✗	0.272 ✗	0.230 ✓	0.153 ✓	0.193 ✗	0.197 ✗	0.261 ✓	0.213 ✓	0.139 ✓	0.113 ✗
	PE $f{=}0.20$	0.412 ✓	**0.281** ✓	0.365 ✓	0.273 ✓	0.228 ✗	0.263 ✓	0.274 ✓	0.335 ✓	0.272 ✗	0.177 ✗	0.151 ✓
	PE $f{=}0.40$	0.397 ✓	0.310 ✓	0.353 ✓	0.280 ✓	0.217 ✗	0.268 ✓	0.273 ✓	0.322 ✓	0.279 ✗	0.171 ✗	0.154 ✓
	PE $f{=}0.60$	0.410 ✓	0.306 ✓	0.348 ✓	0.284 ✓	0.203 ✗	0.262 ✓	0.273 ✓	0.316 ✓	0.260 ✗	0.172 ✗	0.149 ✓
	PE $f{=}0.80$	0.402 ✓	0.309 ✓	**0.346** ✓	0.290 ✓	0.212 ✗	0.261 ✓	0.272 ✓	0.325 ✓	0.257 ✗	0.176 ✗	0.153 ✓
	PE $f{=}1.00$ (M1)	0.398 ✓	0.305 ✓	0.349 ✓	0.285 ✓	0.211 ✗	0.263 ✓	0.269 ✓	0.321 ✓	0.257 ✗	0.176 ✗	0.149 ✓
	SIREN	**0.389** ✓	0.275 ✓	0.347 ✓	**0.249** ✓	**0.194** ✓	**0.252** ✓	**0.263** ✓	**0.287** ✓	**0.252** ✓	**0.163** ✓	**0.143** ✓
$6.5M$	$\theta^\star_A$	0.384	0.274	0.343	0.252	0.193	0.252	0.263	0.302	0.255	0.169	0.143
	True dir.	A>B	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A<B	A<B	A>B
	PE $f{=}0.05$	0.302 ✗	0.169 ✗	0.215 ✗	0.179 ✗	0.115 ✓	0.163 ✗	0.192 ✗	0.231 ✗	0.153 ✓	0.110 ✓	0.087 ✗
	PE $f{=}0.10$	0.309 ✗	0.216 ✗	0.269 ✗	0.227 ✓	0.151 ✓	0.196 ✗	0.207 ✗	0.260 ✓	0.204 ✓	0.140 ✓	0.126 ✗
	PE $f{=}0.20$	0.397 ✓	0.283 ✓	0.353 ✓	0.276 ✓	0.224 ✗	0.274 ✓	0.269 ✓	0.330 ✓	0.264 ✗	0.184 ✗	0.164 ✓
	PE $f{=}0.40$	0.411 ✓	0.296 ✓	0.347 ✓	0.295 ✓	0.210 ✗	0.269 ✓	0.272 ✓	0.338 ✓	0.268 ✗	0.187 ✗	0.150 ✓
	PE $f{=}0.60$	0.403 ✓	0.294 ✓	**0.345** ✓	0.287 ✓	0.211 ✗	0.264 ✓	0.273 ✓	0.328 ✓	0.260 ✗	0.184 ✗	0.148 ✓
	PE $f{=}0.80$	0.401 ✓	0.291 ✓	0.347 ✓	0.290 ✓	0.212 ✗	0.262 ✓	0.270 ✓	0.332 ✓	0.258 ✗	0.180 ✗	0.150 ✓
	PE $f{=}1.00$ (M1)	0.398 ✓	0.293 ✓	0.349 ✓	0.284 ✓	0.208 ✗	0.263 ✓	0.269 ✓	0.330 ✓	**0.257** ✗	0.176 ✗	0.149 ✓
	SIREN	**0.389** ✓	**0.270** ✓	0.347 ✓	**0.251** ✓	**0.194** ✓	**0.254** ✓	**0.261** ✓	**0.296** ✓	**0.253** ✓	**0.168** ✓	**0.143** ✓

The comparison in Table 4 shows that PromptEval faces a bias trade-off that is not resolved by choosing a different cell-budget fraction. The sign of PromptEval's error depends on the observation fraction: sparse PromptEval is too pessimistic, while dense PromptEval approaches the optimistic same-data winner. SIREN does not interpolate between these two regimes. It evaluates the post-search reporting rule on held-out split evaluations and uses item-level resampling to account for

the first-order effect of selection. Across the four budgets in Table 5, SIREN's signed error stays between -0.09 and -0.03 percentage points, while every PromptEval fraction has substantially larger absolute bias. Full PromptEval tables for the other subject-tuner cells appear in Appendix C.

Table 5: PromptEval on **MMLU-Pro Math** for DSPy.

	500K		1.5M		3M		6.5M	
f	dir	bias	dir	bias	dir	bias	dir	bias
0.05	3/11	−8.37	3/11	−8.75	3/11	−8.15	3/11	−8.29
0.10	3/11	−5.20	4/11	−4.69	5/11	−4.59	5/11	−4.76
0.20	9/11	+1.49	8/11	+1.70	8/11	+1.93	8/11	+1.71
0.40	9/11	+1.56	8/11	+1.91	8/11	+1.86	8/11	+1.95
0.60	9/11	+1.62	8/11	+1.60	8/11	+1.48	8/11	+1.52
0.80	9/11	+1.55	8/11	+1.53	8/11	+1.68	8/11	+1.49
1.00	9/11	+1.37	8/11	+1.52	8/11	+1.49	8/11	+1.33
SIREN	**10/11**	**−0.09**	**11/11**	**−0.03**	**11/11**	**−0.05**	**11/11**	**−0.03**

Therefore, budget-aware prompt scoring is not the same as selection-aware inference. PromptEval is designed for efficient evaluation over a fixed prompt pool. In our post-tuning setting, however, the shortlist has already been produced by an adaptive tuner, and the target is the fresh-data performance of a specified reporting rule rather than the imputed performance distribution over prompts. In the results above, sparse PromptEval can have high directional accuracy because shrinkage moves estimates to the same side of $\theta^\star_B$ as the truth, rather than because it estimates $\theta^\star_A$ accurately. Dense PromptEval approaches the full-matrix selected winner, whose difference from the soft-rule reference can also reflect the deployment rule. SIREN targets the reporting quantity directly by evaluating the selected repeated-split procedure and accounting for the first-order effect of selection.

4.4 Additional empirical checks

The fresh-data analysis of 176 Math/Law cells gives M1/M2 and SIREN mean absolute errors of 0.22 and 0.24 pp against their own targets, with $165/176$ correct directions for both (Appendix D.2). This separates estimator accuracy from reporting-rule choice.

On Law, SIREN differs from matched outer-test evaluation of its soft rule by $+0.66$ pp (Appendix D.3). Fresh tuner reruns separate reporting error, tuner variation, and soft-versus-hard deployment differences, without establishing target equivalence (Appendix D.4). Appendix D.1 separates interval centering from the selection contribution to variance.

Under stochastic decoding, bootstrap coverage is 95.05% for both single-decode and fractional scores (Appendix D.5). Philosophy and Economics extend the domain comparisons; HumanEval+ Mini adds execution-scored programs (Appendices D.6 and D.7). SIREN remains close to its specified reporting reference.

5 Conclusion

Adaptive prompt and program search changes what an LLM benchmark score estimates. Once benchmark items are reused for tuning, the reported winner is a data-dependent artifact, not a fixed system evaluated on fresh data. SIREN addresses this tune-then-deploy setting by freezing the post-search shortlist, separating splitwise selection from held-out evaluation, and bootstrapping item-level contributions without rerunning the tuner. In the fixed-shortlist smooth-selector regime, this supports valid simultaneous inference over the budget grid. Experiments show that same-data winner reports can be optimistic and can change tuned-versus-default conclusions. The additional studies distinguish reporting-rule differences from estimation error and extend the evidence to stochastic decoding and execution-scored code generation. Fully open-ended adaptive search remains future work.

References

[1] Isaiah Andrews, Toru Kitagawa, and Adam McCloskey. Inference on winners. *The Quarterly Journal of Economics*, 139(1):305–358, 2024.

[2] Anastasios N. Angelopoulos, Jacob Eisenstein, Jonathan Berant, Alekh Agarwal, and Adam Fisch. Cost-optimal active AI model evaluation. *arXiv preprint arXiv:2506.07949*, 2025.

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[4] Rin Ashizawa, Yoichi Hirose, Nozomu Yoshinari, Kento Uchida, and Shinichi Shirakawa. Bandit-based prompt design strategy selection improves prompt optimizers. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20799–20817. Association for Computational Linguistics, 2025.

[5] Richard Berk, Lawrence Brown, Andreas Buja, Kai Zhang, and Linda Zhao. Valid post-selection inference. *The Annals of Statistics*, 41(2):802–837, 2013.

[6] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1006–1014, 2015.

[7] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[8] Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. *The Annals of Statistics*, 41(6):2786–2819, 2013.

[9] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 117–126, 2015.

[10] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[11] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

[12] Bradley Efron and Robert Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

[13] William Fithian, Dennis Sun, and Jonathan Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

[14] Riccardo Fogliato, Pratik Patil, Mathew Monfort, and Pietro Perona. A framework for efficient model evaluation through stratification, sampling, and estimation. *arXiv preprint arXiv:2406.07320*, 2024.

[15] Kristina Gligorić, Tijana Zrnic, Cinoo Lee, Emmanuel Candès, and Dan Jurafsky. Can unconfident LLM annotations be used for confident conclusions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3514–3533. Association for Computational Linguistics, 2025.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, 2024.

[18] Yinghao He, Mengyu Xu, Haixiang Sun, Donghan Li, Yibo Wang, Lixu Wang, Kezhen Chen, Chi Li, Chunwei Liu, Bharat Bhargava, et al. Suppressing pressure, amplifying evidence: Self-guided attention steering to mitigate sycophancy and stubbornness. *arXiv preprint arXiv:2610.04329*, 2026.

[19] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

[20] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*, 2023.

[21] Jason D. Lee, Dennis L. Sun, Yuekai Sun, and Jonathan E. Taylor. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016.

[22] Gili Lior, Eliya Habba, Shahar Levy, Avi Caciularu, and Gabriel Stanovsky. ReliableEval: A recipe for stochastic LLM evaluation via method of moments. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11146–11153. Association for Computational Linguistics, 2025.

[23] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

[24] 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*, pages 34303–34326. PMLR, 2024.

[25] Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

[26] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[27] Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366. Association for Computational Linguistics, 2024.

[28] Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. Efficient multi-prompt evaluation of LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pages 22483–22512, 2024.

[29] Chengshuai Shi, Kun Yang, Zihan Chen, Jundong Li, Jing Yang, and Cong Shen. Efficient prompt optimization through the lens of best arm identification. In *Advances in Neural Information Processing Systems*, 2024.

[30] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, 2020.

[31] Xiaoying Tian and Jonathan Taylor. Selective inference with a randomized response. *The Annals of Statistics*, 46(2):679–710, 2018.

[32] 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)*, pages 1576–1601. Association for Computational Linguistics, 2024.

[33] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290, 2024.

[34] Skyler Wu, Yash Nair, and Emmanuel J. Candès. Efficient evaluation of LLM performance with statistical guarantees. *arXiv preprint arXiv:2601.20251*, 2026.

[35] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024.

[36] Guanhua Zhang, Florian E. Dorner, and Moritz Hardt. How benchmark prediction from fewer data misses the mark. *arXiv preprint arXiv:2506.07673*, 2025.

[37] Xu-Xiang Zhong, Chao Yi, and Han-Jia Ye. Efficient evaluation of large language models via collaborative filtering. *arXiv preprint arXiv:2504.08781*, 2025.

[38] Jin Peng Zhou, Christian K. Belardi, Ruihan Wu, Travis Zhang, Carla P. Gomes, Wen Sun, and Kilian Q. Weinberger. On speeding up language model evaluation. In *International Conference on Learning Representations*, 2025.

[39] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*, 2023.

[40] Thomas P. Zollo, Todd Morrill, Zhun Deng, Jake C. Snell, Toniann Pitassi, and Richard Zemel. Prompt risk control: A rigorous framework for responsible deployment of large language models. In *International Conference on Learning Representations*, 2024.

[41] Tijana Zrnic and Emmanuel Candes. Active statistical inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62993–63010. PMLR, 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The assumptions for the formal theoretical results are explicitly stated in both the main text and appendix. The practical limitations are discussed in the conclusion when mentioning future work.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper include the full set of assumptions and a complete proof?

   Answer: [Yes]

   Justification: The assumptions are informally stated in the main text along with the informal version of the theorem statements, and are rigorously characterized in the appendix.

4. **Experimental Result Reproducibility**

   Question: Does the paper provide sufficient information for reproducing the main experimental results?

   Answer: [Yes]

   Justification: The paper's experiment section covers sufficient details for reproducibility.

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

   Question: Does the paper include the code, data, and instructions needed to reproduce the main experimental results?

   Answer: [Yes]

   Justification: Yes, we have uploaded it into the supplementary materials. It can also be accessed in `https://anonymous.4open.science/r/siren-supplementary-4B81/README.md`.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and evaluation details?

   Answer: [Yes]

   Justification: The experimental setup in Section 4 specifies: data splits (100 train / 1,251 or 1,101 eval items), model identifiers and serving parameters (vLLM, temperature 0, single RTX 4090), tuner configurations (candidate generation, budget grid $\mathcal{B}=\{500\text{K}, 1.5\text{M}, 3\text{M}, 6.5\text{M}\}$), retained shortlist sizes ($K=10$ for random search, $K=6$ for DSPy), SIREN hyperparameters ($R=10$, $\rho_{\text{score}}=0.5$, $\tau=1$, uniform ω_r, 2,000 bootstrap draws), and PromptEval settings (Rasch IRT, cell-observation fractions $f \in \{0.05, \ldots, 1.00\}$). Full model identifiers are listed in the appendix.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars or other measures of statistical significance?

   Answer: [Yes]

   Justification: SIREN produces 95% confidence intervals by design (multiplier bootstrap), reported in all main tables. Simulation studies report Monte Carlo standard errors (SE ≈ 0.5 pp at $N_{\text{sim}}=2{,}000$) and show error bands in all figures (Studies A–C). Ground truth is approximated by $10{,}000$ Monte Carlo redraws with reported MC error. The PromptEval comparison reports bootstrap CIs where applicable.

8. **Experiments Compute Resources**

   Question: Does the paper provide sufficient information on compute resources?

   Answer: [Yes]

   Justification: All 11 models are served via vLLM on a single NVIDIA RTX 4090 GPU (24 GB VRAM) with sequential model loading.

9. **Code of Ethics**

   Question: Have you read the NeurIPS Code of Ethics and ensured that your research conforms to it?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics. Our work is a statistical methodology paper; it does not involve human subjects, does not collect personal data, and does not develop models with dual-use concerns. All benchmark data used (MMLU-Pro) is publicly available.

10. **Broader Impacts**

    Question: Does the paper discuss potential broader societal impacts?

    Answer: [N/A]

    Justification: This paper is purely on statistical LLM evaluations and does not have broader societal impact.

11. **Safeguards**

    Question: Does the paper describe safeguards for responsible release of data/models with high risk for misuse?

    Answer: [N/A]

    Justification: This paper proposes a statistical evaluation protocol. It does not release pretrained language models, generate synthetic content, or produce assets with dual-use risk. No safeguards beyond standard academic release are required.

12. **Licenses**

    Question: Does the paper cite the creators of used assets and respect their licenses?

    Answer: [Yes]

    Justification: MMLU-Pro [33] is cited and used under its public release terms. All 11 open-weight models are cited by their original publications or model cards and are used under their respective open-source licenses (Apache 2.0, Llama Community License, etc.). DSPy [19] and vLLM [20] are cited. PromptEval [28] is cited and our reimplementation follows the published algorithm.

13. **New Assets**

    Question: Does the paper release new assets with proper documentation?

    Answer: [N/A]

    Justification: We do not release new datasets or pretrained models.

14. **Crowdsourcing and Human Subjects**

    Question: Does the paper involve crowdsourcing or research with human subjects?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing, human annotation, or human subjects research. All experiments use automated LLM evaluation on publicly available benchmark data.

15. **IRB Approvals**

    Question: Does the paper require IRB approval?

    Answer: [N/A]

    Justification: No human subjects research is conducted. IRB approval is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if they are an important component of the core methods?

Answer: [Yes]

Justification: LLMs are the objects being evaluated, not tools used for method development. The 11 open-weight LLMs used as evaluation targets are fully specified in Section 4 and the appendix. DSPy uses the target LLM itself to generate candidate instructions as part of the tuning pipeline (Section 4, "System-A: DSPy MIPROv2"); this is a standard use of DSPy and is documented. The authors also use LLMs for polishing the draft.

A Related works

A first line of related work develops methods that use data to search over prompts, demonstrations, and larger language-model programs. Early automatic prompting methods such as AutoPrompt optimize discrete trigger tokens for masked language models, while Automatic Prompt Engineer treats instructions as candidate programs generated and selected by an LLM [30, 39]. More recent prompt optimizers make the search loop itself increasingly adaptive: OPRO uses an LLM as a derivative-free optimizer that proposes new instructions from previous scores, and EvoPrompt connects LLM prompt generation with evolutionary search [35, 17]. At the program level, DSPy abstracts LM pipelines into parameterized modules and compiles them against a task metric, while MIPRO extends this idea by jointly optimizing instructions and few-shot demonstrations for multi-stage LM programs [19, 27]. Other work makes the evaluation budget explicit: TRIPLE formulates prompt selection as fixed-budget best-arm identification, and OPTS uses bandit-based strategy selection inside prompt optimization [29, 4]. These methods are natural upstream tuners for our setting, including the random-search and DSPy tuners used in our experiments. Our contribution is complementary: rather than proposing another optimizer, we study conditional post-search reporting after a budgeted tuner has frozen a shortlist.

A second line of work asks how to make model evaluation more stable, cheaper, or more statistically meaningful. Benchmarks such as MMLU-Pro increase task difficulty and robustness by emphasizing reasoning-focused questions and larger answer sets, but even fixed-benchmark reports can be fragile because prompt templates affect both absolute scores and model rankings [33, 25]. PromptEval addresses this prompt sensitivity by estimating a performance distribution over a prespecified prompt pool with an IRT-style model under limited cell budgets, making it the closest external baseline in our experiments [28]. ReliableEval instead treats evaluation as stochastic over meaning-preserving prompt perturbations and uses method-of-moments reasoning to determine how many perturbations are needed, while Prompt Risk Control selects prompts using upper bounds on deployment-relevant risk measures [22, 40]. A parallel body of efficient-evaluation work reduces benchmark cost through curated subsets, anchor points, matrix factorization, collaborative filtering, stratified or model-assisted sampling, and active querying with valid confidence intervals [24, 32, 38, 37, 14, 41, 18, 15, 2, 36, 34]. Our internal baselines M1–M4 instantiate common reporting choices in this space, same-data winner reporting, a Wald interval around the winner, a single holdout split, and repeated split-level argmax with a Student-t interval, and illustrate that budget awareness or data splitting alone is not the same as selection-aware inference. Our paper targets a different estimand from these works: the post-search, procedure-level performance of a frozen shortlist plus a specified reporting rule, together with uncertainty that accounts for both held-out evaluation noise and the first-order effect of selection.

A third line of work studies how repeated data use and data-dependent selection invalidate naive uncertainty estimates. The reusable holdout and related adaptive data-analysis theory show that repeatedly querying the same validation data can destroy nominal guarantees unless the reporting protocol controls information leakage; leaderboard-oriented methods such as the Ladder pursue similar goals for machine-learning competitions [10, 9, 6]. Post-selection inference studies valid inference after a model, coefficient, or hypothesis has been chosen from the data, using tools such as simultaneous inference, selective error control, exact conditioning for the lasso, and randomized selection [5, 13, 21, 31]. The winner's curse literature makes the bias mechanism explicit: the empirically best option is likely to have benefited from positive noise, so conventional estimates and confidence intervals for selected winners can be optimistic [11, 1]. Finally, bootstrap and resampling methods provide the computational machinery for quantifying uncertainty in complex estimators, including classical bootstrap corrections and Gaussian multiplier bootstraps for maxima over many coordinates [12, 8]. Our work brings these ideas into adaptive LLM benchmarking by defining the selected tune-then-deploy procedure as the inferential target and by deriving an item-level repeated-split linearization whose multiplier bootstrap simultaneously covers systems, budgets, and fixed contrasts without rerunning the tuner.

B Proofs for Section 3

B.1 Formal setup and assumptions

This appendix proves the main-text guarantees in four steps. Lemma 1 and Proposition 1 establish the score-table representation and conditional CLT. Lemmas 2–4 linearize the selector and aggregate

the split-level expansions, leading to Theorem 3, the formal version of Theorem 1. Theorem 4 proves Theorem 2, and Proposition 2 gives a sufficient condition for consistency of the computable contributions.

Let $\mathcal{F}^{\mathrm{tune}}$ denote the information generated by the completed tuning stage, including the retained shortlists and selectors. All arguments are conditional on $\mathcal{F}^{\mathrm{tune}}$; consequently, the retained shortlists and selectors are treated as fixed reporting-layer objects, and this conditioning is suppressed below. Asymptotics are taken as $M \to \infty$ with the number of systems, budgets, repeated splits, and retained artifacts fixed.

For each split, system, and budget, define the population score vectors

$$s_{r,j,M}(B) = \mathbb{E}\big[\widehat{S}_{r,j}(B) \mid \mathcal{G}_M\big], \qquad t_{r,j,M}(B) = \mathbb{E}\big[\widehat{T}_{r,j}(B) \mid \mathcal{G}_M\big], \qquad q_{r,j,M}(B) = g_{j,B}\big(s_{r,j,M}(B)\big). \tag{9}$$

For a differentiable selector $g_{j,B}$, let $Dg_{j,B}(x)$ denote its Jacobian at x. We also define the centered repeated-split score table

$$\mathbb{U}_M = \Big\{\widehat{S}_{r,j}(B) - s_{r,j,M}(B),\ \widehat{T}_{r,j}(B) - t_{r,j,M}(B)\Big\}_{r\le R,\ j\le N,\ B\in\mathcal{B}}. \tag{10}$$

Finally, define the joint scaled error vector over the finite system-budget grid as

$$\mathbf{Z}_M = \Big\{\sqrt{M}\big(\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)\big)\Big\}_{j\le N,\ B\in\mathcal{B}}. \tag{11}$$

Assumption 1 (Fixed-shortlist smooth-selector regime)**.** *Fix the number of systems N, budget grid $\mathcal{B}$, and number of repeated splits R, and assume $\sup_{j\le N,\ B\in\mathcal{B}} K_{j,B} < \infty$. The split design $\mathcal{G}_M$ is independent of the benchmark items and their execution randomness. There exist deterministic integers m_M, ℓ_M and constants $\rho_{\mathrm{score}}, \rho_{\mathrm{eval}} \in (0,1)$ such that $|D_r^{\mathrm{score}}| = m_M$, $|E_r| = \ell_M$, $m_M/M \to \rho_{\mathrm{score}}$, and $\ell_M/M \to \rho_{\mathrm{eval}}$ for every $r \le R$. All score variables take values in $[0,1]$.*

For each item i, let $\mathcal{R}_i$ collect W_i together with all execution randomness used to score item i across splits, systems, budgets, and retained artifacts. The tuples $\mathcal{R}_1, \dots, \mathcal{R}_M$ are i.i.d. and independent of $\mathcal{G}_M$. No independence is required among the coordinates within a fixed $\mathcal{R}_i$.

There exist constants $\delta_0 > 0$ and $C_g < \infty$ such that, for all sufficiently large M and all r, j, B, the selector $g_{j,B}$ is twice continuously differentiable on $\{x : \|x - s_{r,j,M}(B)\| \le \delta_0\}$, with first and second derivatives uniformly bounded by C_g. The weights ω_r are $\mathcal{G}_M$-measurable, nonnegative, and sum to one.

B.2 Coordinate formulas and plug-in contributions

The main text uses item-level contributions only conceptually. We now give the exact coordinate formulas used in the formal statements and proofs. Let

$$d_U = 2R \sum_{j\le N,\ B\in\mathcal{B}} K_{j,B} \tag{12}$$

denote the dimension of the stacked score table. For the score vectors in (4) and (5), define the coordinatewise item-level contributions

$$\Gamma^S_{i,r,j,k,M}(B) = \frac{M}{|D_r^{\mathrm{score}}|}\mathbf{1}\{i \in D_r^{\mathrm{score}}\}\Big(Z_j\big(W_i, a_{j,B,k}, \xi^S_{irjBk}\big) - s_{r,j,k,M}(B)\Big), \tag{13}$$

$$\Gamma^T_{i,r,j,k,M}(B) = \frac{M}{|E_r|}\mathbf{1}\{i \in E_r\}\Big(Z_j\big(W_i, a_{j,B,k}, \xi^T_{irjBk}\big) - t_{r,j,k,M}(B)\Big), \tag{14}$$

and their block vectors

$$\Gamma^S_{i,r,j,M}(B) = \big(\Gamma^S_{i,r,j,1,M}(B), \dots, \Gamma^S_{i,r,j,K_{j,B},M}(B)\big)^\top, \tag{15}$$

$$\Gamma^T_{i,r,j,M}(B) = \big(\Gamma^T_{i,r,j,1,M}(B), \dots, \Gamma^T_{i,r,j,K_{j,B},M}(B)\big)^\top. \tag{16}$$

The stacked vector $\sqrt{M}\mathbb{U}_M = \frac{1}{\sqrt{M}}\sum_{i=1}^M \Gamma_{i,M}$ is obtained by stacking the blocks in (15) and (16) over the finite index set $\{(r,j,B) : r \le R,\ j \le N,\ B \in \mathcal{B}\}$ under any fixed deterministic ordering. The coordinate definitions imply the score-table identities

$$\sqrt{M}\big(\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big) = \frac{1}{\sqrt{M}}\sum_{i=1}^M \Gamma^S_{i,r,j,M}(B), \tag{17}$$

$$\sqrt{M}\big(\widehat{T}_{r,j}(B) - t_{r,j,M}(B)\big) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \Gamma^{T}_{i,r,j,M}(B). \tag{18}$$

For the selected repeated-split estimator, define the population item-level contribution

$$\psi_{i,j,B,M} = \sum_{r=1}^{R} \omega_r \Big[q_{r,j,M}(B)^\top \Gamma^{T}_{i,r,j,M}(B) + t_{r,j,M}(B)^\top Dg_{j,B}\big(s_{r,j,M}(B)\big)\Gamma^{S}_{i,r,j,M}(B) \Big]. \tag{19}$$

The first term is the direct held-out evaluation contribution of item W_i. The second term is the first-order selection contribution: when W_i appears in a scoring subset, it perturbs the selection scores and hence the splitwise weights through the derivative of $g_{j,B}$.

Stack these contributions over the finite system-budget grid as

$$\mathbf{\Psi}_{i,M} = \{\psi_{i,j,B,M}\}_{j \le N,\ B \in \mathcal{B}}, \qquad \Omega_M = \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}\big[\mathbf{\Psi}_{i,M} \mathbf{\Psi}_{i,M}^\top \mid \mathcal{G}_M\big]. \tag{20}$$

For the computable bootstrap contributions, define the sample-centered coordinatewise analogues

$$\widehat{\Gamma}^{S}_{i,r,j,k,M}(B) = \frac{M}{|D^{\mathrm{score}}_r|} \mathbf{1}\{i \in D^{\mathrm{score}}_r\} \Big(Z_j\big(W_i, a_{j,B,k}, \xi^{S}_{irjBk}\big) - \widehat{S}_{r,j,k}(B) \Big), \tag{21}$$

$$\widehat{\Gamma}^{T}_{i,r,j,k,M}(B) = \frac{M}{|E_r|} \mathbf{1}\{i \in E_r\} \Big(Z_j\big(W_i, a_{j,B,k}, \xi^{T}_{irjBk}\big) - \widehat{T}_{r,j,k}(B) \Big), \tag{22}$$

and the corresponding block vectors

$$\widehat{\Gamma}^{S}_{i,r,j,M}(B) = \big(\widehat{\Gamma}^{S}_{i,r,j,1,M}(B), \dots, \widehat{\Gamma}^{S}_{i,r,j,K_{j,B},M}(B)\big)^\top, \tag{23}$$

$$\widehat{\Gamma}^{T}_{i,r,j,M}(B) = \big(\widehat{\Gamma}^{T}_{i,r,j,1,M}(B), \dots, \widehat{\Gamma}^{T}_{i,r,j,K_{j,B},M}(B)\big)^\top. \tag{24}$$

The plug-in contribution estimator used in the bootstrap subsection is

$$\widehat{\psi}_{i,j,B,M} = \sum_{r=1}^{R} \omega_r \Big[\widehat{q}_{r,j}(B)^\top \widehat{\Gamma}^{T}_{i,r,j,M}(B) + \widehat{T}_{r,j}(B)^\top Dg_{j,B}\big(\widehat{S}_{r,j}(B)\big)\widehat{\Gamma}^{S}_{i,r,j,M}(B) \Big]. \tag{25}$$

Here the first term is the sample analogue of the direct held-out contribution, and the second term is the sample analogue of the first-order selection contribution. We also write

$$\widehat{\mathbf{\Psi}}_{i,M} = \{\widehat{\psi}_{i,j,B,M}\}_{j \le N,\ B \in \mathcal{B}}, \qquad \mathcal{H}_M = \sigma\big(\mathcal{G}_M, \widehat{\mathbf{\Psi}}_{1,M}, \dots, \widehat{\mathbf{\Psi}}_{M,M}\big). \tag{26}$$

B.3 Conditional convergence convention and score-table CLT

For completeness, we state the conditional convergence convention used in the formal results below. We write $X_M \rightsquigarrow X$ conditionally on $\mathcal{G}_M$ in probability if

$$\sup_{h \in BL_1} \big| \mathbb{E}\big[h(X_M) \mid \mathcal{G}_M\big] - \mathbb{E}\big[h(X)\big] \big| \to_p 0, \tag{27}$$

where BL_1 is the class of real-valued functions bounded by one and Lipschitz with constant at most one.

Lemma 1 (Exact score-table representation)**.** *Under Assumption 1,*

$$\sqrt{M}\,\mathbb{U}_M = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \Gamma_{i,M}, \tag{28}$$

where $\Gamma_{i,M}$ is the stacked vector obtained from (13) *and* (14)*. Conditional on $\mathcal{G}_M$, the vectors $\Gamma_{1,M}, \dots, \Gamma_{M,M}$ are independent, mean zero, and satisfy*

$$\max_{1 \le i \le M} \mathbb{E}\big[\|\Gamma_{i,M}\|^2 \mid \mathcal{G}_M\big] \le C \tag{29}$$

for a deterministic constant $C < \infty$ and all sufficiently large M. Consequently,

$$\max_{r,j,B,k} M\,\mathbb{E}\Big[\big(\widehat{S}_{r,j,k}(B) - s_{r,j,k,M}(B)\big)^2 \mid \mathcal{G}_M \Big] \le C \tag{30}$$

and

$$\max_{r,j,B,k} M\,\mathbb{E}\Big[\big(\widehat{T}_{r,j,k}(B) - t_{r,j,k,M}(B)\big)^2 \mid \mathcal{G}_M \Big] \le C. \tag{31}$$

Proof. For a scoring coordinate,

$$\widehat{S}_{r,j,k}(B) = \frac{1}{|D_r^{\mathrm{score}}|} \sum_{i=1}^{M} \mathbf{1}\{i \in D_r^{\mathrm{score}}\} Z_j\big(W_i, a_{j,B,k}, \xi^S_{irjBk}\big). \tag{32}$$

Subtracting $s_{r,j,k,M}(B)$ and multiplying by $\sqrt{M}$ gives

$$\begin{aligned}\sqrt{M}\big(\widehat{S}_{r,j,k}(B) - s_{r,j,k,M}(B)\big) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \frac{M}{|D_r^{\mathrm{score}}|} \mathbf{1}\{i \in D_r^{\mathrm{score}}\} \\ \times \Big(Z_j\big(W_i, a_{j,B,k}, \xi^S_{irjBk}\big) - s_{r,j,k,M}(B)\Big),\end{aligned} \tag{33}$$

which is the corresponding coordinate of $M^{-1/2}\sum_{i=1}^{M}\Gamma_{i,M}$. The same calculation with (5) gives the held-out coordinates. Stacking all coordinates proves (28).

Conditional on $\mathcal{G}_M$, the split membership indicators and denominators are deterministic. The vector $\Gamma_{i,M}$ depends only on W_i and the execution randomness attached to item i. By Assumption 1, these item-level tuples are independent across i, and therefore $\Gamma_{1,M}, \ldots, \Gamma_{M,M}$ are conditionally independent.

For mean zero, consider a scoring coordinate. If $i \notin D_r^{\mathrm{score}}$, then the corresponding contribution is zero. If $i \in D_r^{\mathrm{score}}$, then the included summands in (32) are conditionally identically distributed given $\mathcal{G}_M$, so

$$s_{r,j,k,M}(B) = \mathbb{E}\Big[Z_j\big(W_i, a_{j,B,k}, \xi^S_{irjBk}\big) \mid \mathcal{G}_M\Big]. \tag{34}$$

The same argument applies to held-out coordinates. Hence

$$\mathbb{E}\big[\Gamma_{i,M} \mid \mathcal{G}_M\big] = 0. \tag{35}$$

The dimension d_U is fixed by Assumption 1. Also, the split fractions stay away from zero, so there exists $L < \infty$ such that

$$\max_{r \le R} \frac{M}{|D_r^{\mathrm{score}}|} \le L, \qquad \max_{r \le R} \frac{M}{|E_r|} \le L \tag{36}$$

for all sufficiently large M. Since scores lie in $[0,1]$, every coordinate of $\Gamma_{i,M}$ is bounded by L in absolute value. Thus $\|\Gamma_{i,M}\|^2 \le C$ for a deterministic constant $C < \infty$, proving (29).

Finally, by (33), conditional independence, and mean zero,

$$\begin{aligned}M\,\mathbb{E}\Big[\big(\widehat{S}_{r,j,k}(B) - s_{r,j,k,M}(B)\big)^2 \mid \mathcal{G}_M\Big] &= \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}\Big[\big(\Gamma^S_{i,r,j,k,M}(B)\big)^2 \mid \mathcal{G}_M\Big] \\ &\le C.\end{aligned} \tag{37}$$

The proof of (31) is identical. □

Assumption 2 (Score-table covariance stabilization)**.** *Let*

$$\Sigma_M = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}\big[\Gamma_{i,M}\Gamma_{i,M}^{\top} \mid \mathcal{G}_M\big]. \tag{38}$$

There exists a deterministic positive semidefinite matrix Σ such that

$$\|\Sigma_M - \Sigma\|_{\mathrm{op}} \to_p 0. \tag{39}$$

Proposition 1 (Score-table CLT)**.** *Under Assumptions 1 and 2,*

$$\sqrt{M}\,\mathbb{U}_M \rightsquigarrow N(0,\Sigma) \qquad \textit{conditionally on } \mathcal{G}_M \textit{ in probability}. \tag{40}$$

In particular, $\sqrt{M}\,\mathbb{U}_M = O_p(1)$.

Proof. By Lemma 1,

$$\sqrt{M}\,\mathbb{U}_M = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \Gamma_{i,M}, \tag{41}$$

where, conditional on $\mathcal{G}_M$, the summands are independent, mean zero, and uniformly bounded.

Fix a deterministic vector $u \in \mathbb{R}^{d_U}$ and define

$$Y_{i,M}(u) = \frac{1}{\sqrt{M}} u^\top \Gamma_{i,M}. \tag{42}$$

Then

$$\sum_{i=1}^{M} \mathbb{E}\big[Y_{i,M}(u)^2 \mid \mathcal{G}_M\big] = u^\top \Sigma_M u. \tag{43}$$

By Assumption 2, this variance converges in probability to $u^\top \Sigma u$.

The Lindeberg condition is immediate. Lemma 1 and bounded scores imply that there exists $C_0 < \infty$ such that $\max_i \|\Gamma_{i,M}\| \le C_0$ for all sufficiently large M. Hence

$$\max_{1 \le i \le M} |Y_{i,M}(u)| \le \frac{\|u\| C_0}{\sqrt{M}} \to 0, \tag{44}$$

which implies the conditional Lindeberg condition for every fixed u.

Along any subsequence there is a further subsequence on which $\|\Sigma_M - \Sigma\|_{\mathrm{op}} \to 0$ almost surely. On that further subsequence, the conditional Lindeberg–Feller theorem gives

$$u^\top \sqrt{M}\,\mathbb{U}_M \rightsquigarrow N(0, u^\top \Sigma u) \qquad \text{conditionally on } \mathcal{G}_M \tag{45}$$

almost surely. Since the dimension is fixed, Cramér–Wold gives (40). The bound $\sqrt{M}\,\mathbb{U}_M = O_p(1)$ follows from the same display. □

B.4 Selection-aware linearization

Lemma 2 (Selector linearization)**.** *Under Assumption 1,*

$$\max_{r \le R,\ j \le N,\ B \in \mathcal{B}} \Big\| \widehat{q}_{r,j}(B) - q_{r,j,M}(B) - Dg_{j,B}\big(s_{r,j,M}(B)\big)\big(\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big) \Big\| = o_p(M^{-1/2}). \tag{46}$$

Proof. By Lemma 1, each coordinate of $\widehat{S}_{r,j}(B) - s_{r,j,M}(B)$ has conditional second moment bounded by C/M. Since the index set is finite,

$$\max_{r,j,B} \big\| \widehat{S}_{r,j}(B) - s_{r,j,M}(B) \big\| = O_p(M^{-1/2}). \tag{47}$$

Fix r, j, B and write $\widehat{S} = \widehat{S}_{r,j}(B)$, $s = s_{r,j,M}(B)$, $g = g_{j,B}$, and $h = \widehat{S} - s$. Assumption 1 gives a neighborhood of s on which g is twice continuously differentiable with uniformly bounded second derivatives. On the event $\{\|h\| \le \delta_0\}$, Taylor's theorem yields

$$g(s+h) = g(s) + Dg(s)h + R(h), \qquad \|R(h)\| \le C\|h\|^2, \tag{48}$$

with C uniform over the finite index set. By (47), the event on which all relevant $\|h\|$ are at most δ_0 has probability tending to one. Hence

$$\begin{aligned} &\max_{r,j,B} \Big\| \widehat{q}_{r,j}(B) - q_{r,j,M}(B) - Dg_{j,B}\big(s_{r,j,M}(B)\big)\big(\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big) \Big\| \\ &\qquad \le C \left(\max_{r,j,B} \big\| \widehat{S}_{r,j}(B) - s_{r,j,M}(B) \big\| \right)^2 = O_p(M^{-1}), \end{aligned} \tag{49}$$

which proves the claim. □

Lemma 3 (Selected split-level expansion)**.** *Under Assumption 1, jointly over $r \le R$, $j \le N$, and $B \in \mathcal{B}$,*

$$\begin{aligned}\widehat{Y}_{r,j}(B) - \mathbb{E}\big[\widehat{Y}_{r,j}(B) \mid \mathcal{G}_M\big] &= q_{r,j,M}(B)^\top \big(\widehat{T}_{r,j}(B) - t_{r,j,M}(B)\big) \\ &\quad + t_{r,j,M}(B)^\top Dg_{j,B}\big(s_{r,j,M}(B)\big)\big(\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big) \\ &\quad + o_p(M^{-1/2}).\end{aligned} \tag{50}$$

Proof. Fix r, j, B and abbreviate

$$\widehat{S} = \widehat{S}_{r,j}(B),\ s = s_{r,j,M}(B),\ \widehat{T} = \widehat{T}_{r,j}(B),\ t = t_{r,j,M}(B),\ A = Dg_{j,B}(s),\ \widehat{q} = \widehat{q}_{r,j}(B),\ q = q_{r,j,M}(B). \tag{51}$$

By Lemma 2,

$$\widehat{q} = q + A(\widehat{S} - s) + R_q, \qquad \|R_q\| = o_p(M^{-1/2}). \tag{52}$$

The proof of Lemma 2 also gives the sharper bound $\|R_q\| = O_p(M^{-1})$. Moreover, since g maps into a simplex, the scores are bounded, and A is uniformly bounded, the Taylor remainder is uniformly bounded off the local event. Combining this boundedness with the conditional second-moment bound for $\widehat{S} - s$ gives

$$\mathbb{E}\big[\|R_q\| \mid \mathcal{G}_M\big] = O_p(M^{-1}), \qquad \mathbb{E}\big[\|R_q\|^2 \mid \mathcal{G}_M\big] = O_p(M^{-1}). \tag{53}$$

Expanding $\widehat{Y} = \widehat{q}^\top \widehat{T}$ gives

$$\begin{aligned}\widehat{Y} &= \big(q + A(\widehat{S} - s) + R_q\big)^\top \big(t + (\widehat{T} - t)\big) \\ &= q^\top t + q^\top(\widehat{T} - t) + t^\top A(\widehat{S} - s) + \mathcal{R},\end{aligned} \tag{54}$$

where

$$\mathcal{R} = t^\top R_q + \big(A(\widehat{S} - s)\big)^\top (\widehat{T} - t) + R_q^\top (\widehat{T} - t). \tag{55}$$

Since q, t, A are $\mathcal{G}_M$-measurable and s, t are the conditional means,

$$\mathbb{E}\big[q^\top(\widehat{T} - t) \mid \mathcal{G}_M\big] = 0, \qquad \mathbb{E}\big[t^\top A(\widehat{S} - s) \mid \mathcal{G}_M\big] = 0. \tag{56}$$

Therefore

$$\widehat{Y} - \mathbb{E}\big[\widehat{Y} \mid \mathcal{G}_M\big] = q^\top(\widehat{T} - t) + t^\top A(\widehat{S} - s) + \mathcal{R} - \mathbb{E}\big[\mathcal{R} \mid \mathcal{G}_M\big]. \tag{57}$$

It remains to show that the centered remainder is $o_p(M^{-1/2})$. Lemma 1 implies

$$\|\widehat{S} - s\| = O_p(M^{-1/2}), \qquad \|\widehat{T} - t\| = O_p(M^{-1/2}), \tag{58}$$

jointly over the finite index set, and also gives conditional second moments of order M^{-1}. The first remainder term is $O_p(M^{-1})$ with conditional expectation $O_p(M^{-1})$ by (53). For the second term, Cauchy–Schwarz and the conditional second-moment bounds give

$$\mathbb{E}\Big[\big|\big(A(\widehat{S} - s)\big)^\top (\widehat{T} - t)\big| \mid \mathcal{G}_M\Big] = O_p(M^{-1}), \tag{59}$$

and the term itself is $O_p(M^{-1})$. The third term is handled similarly using (53) and the conditional second-moment bound for $\widehat{T} - t$. Thus

$$\mathcal{R} - \mathbb{E}\big[\mathcal{R} \mid \mathcal{G}_M\big] = o_p(M^{-1/2}). \tag{60}$$

Substituting (60) into (57) proves the lemma. □

Lemma 4 (Aggregation over repeated splits)**.** *Under Assumption 1, jointly over $j \le N$ and $B \in \mathcal{B}$,*

$$\sqrt{M}\Big(\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)\Big) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \psi_{i,j,B,M} + o_p(1). \tag{61}$$

If $\Omega_M = \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}\big[\boldsymbol{\Psi}_{i,M}\boldsymbol{\Psi}_{i,M}^\top \mid \mathcal{G}_M\big]$ satisfies $\|\Omega_M - \Omega\|_{\mathrm{op}} \to_p 0$, then

$$\mathbf{Z}_M \rightsquigarrow N(0, \Omega) \qquad \textit{conditionally on } \mathcal{G}_M \textit{ in probability}. \tag{62}$$

Proof. By definition of the target and the $\mathcal{G}_M$-measurability of the weights,

$$\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B) = \sum_{r=1}^{R} \omega_r \Big(\widehat{Y}_{r,j}(B) - \mathbb{E}\big[\widehat{Y}_{r,j}(B) \mid \mathcal{G}_M\big] \Big). \tag{63}$$

Multiplying by $\sqrt{M}$ and substituting Lemma 3 yields

$$\begin{aligned}\sqrt{M}\Big(\widetilde{\theta}_{j,R}(B) - \theta_{j,M}^{\mathrm{RS}|\mathcal{G}}(B)\Big) = &\sum_{r=1}^{R} \omega_r q_{r,j,M}(B)^\top \sqrt{M}\big(\widehat{T}_{r,j}(B) - t_{r,j,M}(B)\big) \\ &+ \sum_{r=1}^{R} \omega_r t_{r,j,M}(B)^\top Dg_{j,B}\big(s_{r,j,M}(B)\big) \sqrt{M}\big(\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big) \\ &+ o_p(1). \end{aligned} \tag{64}$$

Using (17) and (18), the right-hand side becomes

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} \sum_{r=1}^{R} \omega_r \Big[q_{r,j,M}(B)^\top \Gamma_{i,r,j,M}^{T}(B) + t_{r,j,M}(B)^\top Dg_{j,B}\big(s_{r,j,M}(B)\big) \Gamma_{i,r,j,M}^{S}(B) \Big] + o_p(1). \tag{65}$$

The inner sum is exactly $\psi_{i,j,B,M}$ in (19), proving (61).

For the conditional Gaussian limit, let $d = N|\mathcal{B}|$ and fix $u \in \mathbb{R}^d$. Define

$$Z_{i,M}(u) = \frac{1}{\sqrt{M}} u^\top \mathbf{\Psi}_{i,M}. \tag{66}$$

Conditional on $\mathcal{G}_M$, the variables $Z_{1,M}(u), \ldots, Z_{M,M}(u)$ are independent and mean zero, because each $\mathbf{\Psi}_{i,M}$ is a $\mathcal{G}_M$-measurable linear transformation of $\Gamma_{i,M}$. Their variance sum is

$$\sum_{i=1}^{M} \mathbb{E}\big[Z_{i,M}(u)^2 \mid \mathcal{G}_M\big] = u^\top \Omega_M u. \tag{67}$$

By $\|\Omega_M - \Omega\|_{\mathrm{op}} \to_p 0$, this variance converges in probability to $u^\top \Omega u$. Since scores, shortlist sizes, and selector derivatives are uniformly bounded, there exists $C_\Psi < \infty$ such that

$$\max_{1 \le i \le M} \|\mathbf{\Psi}_{i,M}\| \le C_\Psi \tag{68}$$

for all sufficiently large M. Hence $\max_i |Z_{i,M}(u)| \le \|u\| C_\Psi / \sqrt{M} \to 0$, giving the conditional Lindeberg condition. The conditional Lindeberg–Feller theorem, followed by Cramér–Wold and the usual subsequence argument for convergence in probability of the conditional laws, gives

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} \mathbf{\Psi}_{i,M} \rightsquigarrow N(0, \Omega) \qquad \text{conditionally on } \mathcal{G}_M \text{ in probability.} \tag{69}$$

Combining (69) with (61) proves (62). □

Theorem 3 (Formal selection-aware linearization)**.** *Under Assumption 1, jointly over $j \le N$ and $B \in \mathcal{B}$,*

$$\mathbf{Z}_M = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \mathbf{\Psi}_{i,M} + o_p(1). \tag{70}$$

Furthermore, if there exists a deterministic positive semidefinite matrix Ω such that $\|\Omega_M - \Omega\|_{\mathrm{op}} \to_p 0$, then

$$\mathbf{Z}_M \rightsquigarrow N(0, \Omega) \qquad \textit{conditionally on } \mathcal{G}_M \textit{ in probability}. \tag{71}$$

Proof. The linear representation (70) is exactly the first conclusion of Lemma 4. The conditional Gaussian limit (71) is the second conclusion of Lemma 4. □

B.5 Multiplier bootstrap validity

Theorem 4 (Formal multiplier bootstrap validity)**.** *Work under Assumption 1, and suppose there exists a deterministic positive semidefinite matrix Ω such that $\|\Omega_M - \Omega\|_{\mathrm{op}} \to_p 0$. Assume further that the computable contributions satisfy*

$$\frac{1}{M}\sum_{i=1}^{M} \big\|\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big\|^2 \to_p 0. \tag{72}$$

Let

$$\bar{\widehat{\mathbf{\Psi}}}_M = \frac{1}{M}\sum_{i=1}^{M} \widehat{\mathbf{\Psi}}_{i,M}, \qquad \widehat{\mathbf{G}}^*_M = \frac{1}{\sqrt{M}}\sum_{i=1}^{M} \zeta_i\big(\widehat{\mathbf{\Psi}}_{i,M} - \bar{\widehat{\mathbf{\Psi}}}_M\big), \tag{73}$$

where $\zeta_1, \dots, \zeta_M \overset{\text{i.i.d.}}{\sim} N(0,1)$ are independent of the data. Then, with $G \sim N(0,\Omega)$ and $d = N|\mathcal{B}|$,

$$\sup_{h \in BL_1(\mathbb{R}^d)} \Big|\mathbb{E}\big[h(\widehat{\mathbf{G}}^*_M) \mid \mathcal{H}_M\big] - \mathbb{E}\big[h(G)\big]\Big| \to_p 0. \tag{74}$$

Moreover,

$$\sup_{h \in BL_1(\mathbb{R}^d)} \Big|\mathbb{E}\big[h(\widehat{\mathbf{G}}^*_M) \mid \mathcal{H}_M\big] - \mathbb{E}\big[h(\mathbf{Z}_M) \mid \mathcal{G}_M\big]\Big| \to_p 0. \tag{75}$$

Proof. Use the notation in (73). Conditional on $\mathcal{H}_M$, the vectors $\widehat{\mathbf{\Psi}}_{i,M} - \bar{\widehat{\mathbf{\Psi}}}_M$ are fixed and the multipliers are i.i.d. standard Gaussian. Hence

$$\widehat{\mathbf{G}}^*_M \mid \mathcal{H}_M \sim N\big(0, \widehat{\Omega}^*_M\big), \tag{76}$$

where

$$\widehat{\Omega}^*_M = \frac{1}{M}\sum_{i=1}^{M} \big(\widehat{\mathbf{\Psi}}_{i,M} - \bar{\widehat{\mathbf{\Psi}}}_M\big)\big(\widehat{\mathbf{\Psi}}_{i,M} - \bar{\widehat{\mathbf{\Psi}}}_M\big)^\top. \tag{77}$$

Define the infeasible centered covariance

$$\bar{\mathbf{\Psi}}_M = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Psi}_{i,M}, \qquad \widetilde{\Omega}_M = \frac{1}{M}\sum_{i=1}^{M} \big(\mathbf{\Psi}_{i,M} - \bar{\mathbf{\Psi}}_M\big)\big(\mathbf{\Psi}_{i,M} - \bar{\mathbf{\Psi}}_M\big)^\top. \tag{78}$$

We first show that $\widetilde{\Omega}_M$ is close to Ω. Since the $\mathbf{\Psi}_{i,M}$ are conditionally independent, mean zero, and uniformly bounded, each entry of

$$\frac{1}{M}\sum_{i=1}^{M} \mathbf{\Psi}_{i,M}\mathbf{\Psi}_{i,M}^\top - \Omega_M \tag{79}$$

has conditional variance of order M^{-1}. The dimension is fixed, so

$$\left\|\frac{1}{M}\sum_{i=1}^{M} \mathbf{\Psi}_{i,M}\mathbf{\Psi}_{i,M}^\top - \Omega_M\right\|_{\mathrm{op}} \to_p 0. \tag{80}$$

Moreover,

$$\mathbb{E}\big[\|\sqrt{M}\,\bar{\mathbf{\Psi}}_M\|^2 \mid \mathcal{G}_M\big] = \mathrm{tr}(\Omega_M) = O_p(1), \tag{81}$$

where the last equality follows from $\|\Omega_M - \Omega\|_{\mathrm{op}} \to_p 0$. Hence $\bar{\mathbf{\Psi}}_M\bar{\mathbf{\Psi}}_M^\top = o_p(1)$, and

$$\|\widetilde{\Omega}_M - \Omega\|_{\mathrm{op}} \to_p 0. \tag{82}$$

We next compare $\widehat{\Omega}^*_M$ with $\widetilde{\Omega}_M$. Let

$$a_{i,M} = \widehat{\mathbf{\Psi}}_{i,M} - \bar{\widehat{\mathbf{\Psi}}}_M, \qquad b_{i,M} = \mathbf{\Psi}_{i,M} - \bar{\mathbf{\Psi}}_M. \tag{83}$$

Then

$$\begin{aligned}\|\widehat{\Omega}_M^* - \widetilde{\Omega}_M\|_{\mathrm{op}} &\le \frac{1}{M}\sum_{i=1}^{M} \|a_{i,M} - b_{i,M}\|\,\|a_{i,M}\| + \frac{1}{M}\sum_{i=1}^{M} \|b_{i,M}\|\,\|a_{i,M} - b_{i,M}\| \\ &\le \left(\frac{1}{M}\sum_{i=1}^{M} \|a_{i,M} - b_{i,M}\|^2\right)^{1/2} \left(\frac{1}{M}\sum_{i=1}^{M} \|a_{i,M}\|^2\right)^{1/2} \\ &\quad + \left(\frac{1}{M}\sum_{i=1}^{M} \|b_{i,M}\|^2\right)^{1/2} \left(\frac{1}{M}\sum_{i=1}^{M} \|a_{i,M} - b_{i,M}\|^2\right)^{1/2}. \end{aligned} \tag{84}$$

Since

$$a_{i,M} - b_{i,M} = \big(\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big) - \big(\bar{\widehat{\mathbf{\Psi}}}_M - \bar{\mathbf{\Psi}}_M\big), \tag{85}$$

the plug-in condition of $\frac{1}{M}\sum_{i=1}^{M} \big\|\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big\|^2 \to_p 0$ implies

$$\frac{1}{M}\sum_{i=1}^{M} \|a_{i,M} - b_{i,M}\|^2 \to_p 0. \tag{86}$$

The remaining two factors in (84) are $O_p(1)$: this is immediate for the $b_{i,M}$ factor from uniform boundedness, and for the $a_{i,M}$ factor from the condition of $\frac{1}{M}\sum_{i=1}^{M} \big\|\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big\|^2 \to_p 0$. Thus

$$\|\widehat{\Omega}_M^* - \widetilde{\Omega}_M\|_{\mathrm{op}} \to_p 0. \tag{87}$$

Combining (82) and (87) gives

$$\|\widehat{\Omega}_M^* - \Omega\|_{\mathrm{op}} \to_p 0. \tag{88}$$

Let $U \sim N(0, I_d)$, $d = N|\mathcal{B}|$, be independent of everything else, and write

$$A_M = (\widehat{\Omega}_M^*)^{1/2}, \qquad A = \Omega^{1/2}. \tag{89}$$

The conditional law of $\widehat{\mathbf{G}}_M^*$ given $\mathcal{H}_M$ is the law of $A_M U$, while $G \sim N(0, \Omega)$ has the law of AU. Since matrix square roots are continuous on the cone of positive semidefinite matrices in fixed dimension, (88) implies $\|A_M - A\|_{\mathrm{op}} \to_p 0$. Therefore

$$\sup_{h \in BL_1} \Big|\mathbb{E}\big[h(\widehat{\mathbf{G}}_M^*) \mid \mathcal{H}_M\big] - \mathbb{E}\big[h(G)\big]\Big| \le \|A_M - A\|_{\mathrm{op}}\, \mathbb{E}\|U\| \to_p 0. \tag{90}$$

This proves that $\sup_{h \in BL_1} \Big|\mathbb{E}\big[h(\widehat{\mathbf{G}}_M^*) \mid \mathcal{H}_M\big] - \mathbb{E}\big[h(G)\big]\Big| \to_p 0$. The validity statement $\sup_{h \in BL_1} \Big|\mathbb{E}\big[h(\widehat{\mathbf{G}}_M^*) \mid \mathcal{H}_M\big] - \mathbb{E}\big[h(\mathbf{Z}_M) \mid \mathcal{G}_M\big]\Big| \to_p 0$ follows by the triangle inequality and Theorem 3. □

The plug-in condition (72) is stated abstractly in Theorem 4. The next proposition gives a sufficient condition for this requirement in the smooth-selector regime.

B.6 Plug-in consistency for the computable contributions

Proposition 2 (Sufficient condition for plug-in consistency)**.** *Work under Assumption 1. Suppose there exist constants $L_1, L_2 < \infty$ such that, for every $j \le N$, $B \in \mathcal{B}$, and all $x, y \in [0,1]^{K_{j,B}}$,*

$$\|Dg_{j,B}(x)\|_{\mathrm{op}} \le L_1, \qquad \|Dg_{j,B}(x) - Dg_{j,B}(y)\|_{\mathrm{op}} \le L_2\|x - y\|. \tag{91}$$

Then

$$\frac{1}{M}\sum_{i=1}^{M} \big\|\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big\|^2 = O_p(M^{-1}), \tag{92}$$

and in particular $\frac{1}{M}\sum_{i=1}^{M} \big\|\widehat{\mathbf{\Psi}}_{i,M} - \mathbf{\Psi}_{i,M}\big\|^2 \to_p 0$ holds.

Proof. By Lemma 1, the coordinatewise conditional second moments of $\widehat{S}_{r,j}(B) - s_{r,j,M}(B)$ and $\widehat{T}_{r,j}(B) - t_{r,j,M}(B)$ are of order M^{-1}. Since the index set is finite,

$$\big\|\widehat{S}_{r,j}(B) - s_{r,j,M}(B)\big\| = O_p(M^{-1/2}), \qquad \big\|\widehat{T}_{r,j}(B) - t_{r,j,M}(B)\big\| = O_p(M^{-1/2}) \tag{93}$$

jointly over the finite index set. The bound (91) gives

$$\big\|\widehat{q}_{r,j}(B) - q_{r,j,M}(B)\big\| = O_p(M^{-1/2}),\ \big\|Dg_{j,B}\big(\widehat{S}_{r,j}(B)\big) - Dg_{j,B}\big(s_{r,j,M}(B)\big)\big\|_{\mathrm{op}} = O_p(M^{-1/2}). \tag{94}$$

Since the scores are bounded and the split fractions stay away from zero, there exists $C_\Gamma < \infty$ such that

$$\max_{i,r,j,B}\Big(\|\Gamma^S_{i,r,j,M}(B)\|, \|\Gamma^T_{i,r,j,M}(B)\|, \|\widehat{\Gamma}^S_{i,r,j,M}(B)\|, \|\widehat{\Gamma}^T_{i,r,j,M}(B)\|\Big) \le C_\Gamma \tag{95}$$

for all sufficiently large M. Moreover,

$$\widehat{\Gamma}^S_{i,r,j,k,M}(B) - \Gamma^S_{i,r,j,k,M}(B) = \frac{M}{|D^{\mathrm{score}}_r|}\mathbf{1}\{i \in D^{\mathrm{score}}_r\}\big(s_{r,j,k,M}(B) - \widehat{S}_{r,j,k}(B)\big), \tag{96}$$

and similarly for $\widehat{\Gamma}^T_{i,r,j,k,M}(B) - \Gamma^T_{i,r,j,k,M}(B)$. Therefore,

$$\frac{1}{M}\sum_{i=1}^{M}\big\|\widehat{\Gamma}^S_{i,r,j,M}(B) - \Gamma^S_{i,r,j,M}(B)\big\|^2 = O_p(M^{-1}), \tag{97}$$

$$\frac{1}{M}\sum_{i=1}^{M}\big\|\widehat{\Gamma}^T_{i,r,j,M}(B) - \Gamma^T_{i,r,j,M}(B)\big\|^2 = O_p(M^{-1}). \tag{98}$$

For fixed r, j, B, define

$$U_{i,r,j,B,M} = q_{r,j,M}(B)^\top \Gamma^T_{i,r,j,M}(B), \qquad \widehat{U}_{i,r,j,B,M} = \widehat{q}_{r,j}(B)^\top \widehat{\Gamma}^T_{i,r,j,M}(B). \tag{99}$$

Using (94), (95), and (98),

$$\frac{1}{M}\sum_{i=1}^{M}\big|\widehat{U}_{i,r,j,B,M} - U_{i,r,j,B,M}\big|^2 = O_p(M^{-1}). \tag{100}$$

Likewise define

$$V_{i,r,j,B,M} = t_{r,j,M}(B)^\top Dg_{j,B}\big(s_{r,j,M}(B)\big)\Gamma^S_{i,r,j,M}(B), \tag{101}$$

$$\widehat{V}_{i,r,j,B,M} = \widehat{T}_{r,j}(B)^\top Dg_{j,B}\big(\widehat{S}_{r,j}(B)\big)\widehat{\Gamma}^S_{i,r,j,M}(B). \tag{102}$$

The same bounds, together with (97), imply

$$\frac{1}{M}\sum_{i=1}^{M}\big|\widehat{V}_{i,r,j,B,M} - V_{i,r,j,B,M}\big|^2 = O_p(M^{-1}). \tag{103}$$

Finally,

$$\widehat{\psi}_{i,j,B,M} - \psi_{i,j,B,M} = \sum_{r=1}^{R}\omega_r\Big[\big(\widehat{U}_{i,r,j,B,M} - U_{i,r,j,B,M}\big) + \big(\widehat{V}_{i,r,j,B,M} - V_{i,r,j,B,M}\big)\Big]. \tag{104}$$

Since R is fixed and the weights are nonnegative and sum to one,

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M}\big|\widehat{\psi}_{i,j,B,M} - \psi_{i,j,B,M}\big|^2 \le{}& 2\sum_{r=1}^{R}\omega_r\frac{1}{M}\sum_{i=1}^{M}\big|\widehat{U}_{i,r,j,B,M} - U_{i,r,j,B,M}\big|^2 \\ &+ 2\sum_{r=1}^{R}\omega_r\frac{1}{M}\sum_{i=1}^{M}\big|\widehat{V}_{i,r,j,B,M} - V_{i,r,j,B,M}\big|^2 = O_p(M^{-1}).\end{aligned} \tag{105}$$

Summing over the finite index set (j, B) proves (92). □

Remark 1. *The multiplier bootstrap uses one multiplier per benchmark item rather than one multiplier per split because the same item can appear in different roles across repeated splits and because the same evaluation pool is used across systems and budgets. The item-level vector $\widehat{\mathbf{\Psi}}_{i,M}$ collects all these roles before resampling. Using a common multiplier for the whole vector preserves the dependence structure needed for simultaneous bands and fixed contrasts.*

C Additional experiments

This appendix collects three sets of supplementary results that support the main paper's claims across all four (subject, tuner) configurations: (i) Track I theory-validation studies in controlled simulation (Section C.1); (ii) per-model SIREN-vs-baseline comparisons on real data (Section C.2); and (iii) per-model and per-budget PromptEval-vs-SIREN comparisons (Section C.3).

C.1 Theory Validation

Track I validates the three core theoretical results in controlled settings where ground truth is known by construction. All three studies use a common Bernoulli item-response DGP: item i has difficulty $\delta_i \sim \text{Uniform}(-2, 2)$, artifact k has quality q_k, and $Z_{ik} \sim \text{Bernoulli}(\sigma(q_k - \delta_i))$.

C.1.1 Study A: Coverage Validation of the Multiplier Bootstrap

Study A validates the three central operational claims of Theorems 1 and 2 in a controlled setting where the ground truth is tractable by Monte Carlo: (i) the multiplier bootstrap on the plug-in influence function $\widehat{\psi}$ achieves nominal coverage for the repeated-split estimator $\widetilde{\theta}$; (ii) its confidence bands shrink at the theoretical $M^{-1/2}$ rate; (iii) it reproduces a nonparametric item-bootstrap's coverage and width at a fraction of the computational cost.

Data generating process. Items have i.i.d. difficulties $\delta_i \sim \text{Uniform}(-2, 2)$ and K artifacts have qualities equally spaced over $[0, 0.3]$; scores are $Z_{ik} \sim \text{Bernoulli}(\sigma(q_k - \delta_i))$. The selector is softmax with τ=0.1, the split fraction is ρ=0.5, and the bootstrap uses B_{boot}=500 Gaussian multipliers. Ground truth $\theta^\star = \mathbb{E}[\widetilde{\theta}]$ is estimated by N_{gt}=3,000 independent Monte Carlo replications; empirical coverage is then measured over N_{sim}=2,000 trials per configuration, giving a Monte Carlo standard error of approximately SE ≈ 0.5 pp at the nominal 95% level.

Main result: calibrated coverage across M, K, and R. Figure 2 shows the coverage sweep at R=5 across $M \in \{100, 200, 500, 1000, 2000\}$ and $K \in \{2, 5, 10\}$. Coverage ranges from 93.9% to 95.9%, with a maximum deviation of 1.1 percentage points from nominal (Table 6). The right panel confirms that CI width tracks the theoretical $M^{-1/2}$ scaling on a log–log plot: the empirical log–log slopes are -0.510, -0.524, and -0.520 for K=2, 5, 10, all within 5% of the theoretical -0.5, and the three curves run parallel as predicted. In the practitioner regime of $M \geq 500$ the coverage is always within 0.6 pp of nominal and the width is well below 0.1, showing that the bootstrap is usable at sample sizes typical of LLM benchmarks.

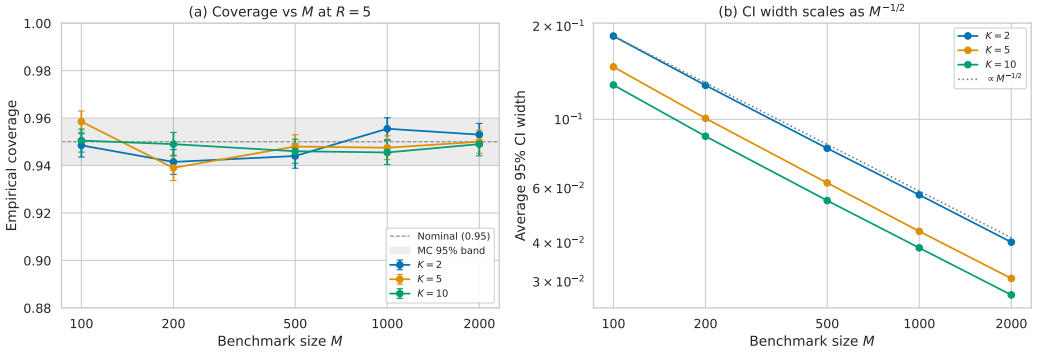

Figure 2: **Study A: Coverage and width across M at R=5.** (a) Empirical coverage of the multiplier-bootstrap 95% CI. Error bars are ±SE over N_{sim}=2,000 trials; the grey band is the Monte Carlo 95% tolerance region around the nominal level. (b) Log–log plot of average CI width; dotted line shows the reference $M^{-1/2}$ slope.

Effect of the number of repeated splits. Figure 3 examines the effect of R at fixed M=500 and K=10. Coverage is stable across $R \in \{1, 3, 5, 10\}$ (all within 1 pp of nominal), confirming that

Table 6: Study A: Average CI width of the SIREN multiplier bootstrap at R=5.

	Average CI width		
M	K=2	K=5	K=10
100	0.186	0.148	0.129
200	0.129	0.101	0.088
500	0.081	0.062	0.055
1000	0.057	0.044	0.038
2000	0.040	0.031	0.027

the influence-function bootstrap is already first-order valid at R=1. What R does buy is efficiency: increasing R from 1 to 5 contracts the CI width by 22% (from 0.0705 to 0.0547), but doubling R further to 10 contracts it by only another 4% (to 0.0525). The curvature reflects the fact that the held-out sample at a given R is not R-fold independent but merely R-fold reweighted, so the additional splits add vanishing information once R is moderate. This matches the practitioner recommendation $R \in [5, 10]$ used throughout the remainder of the paper.

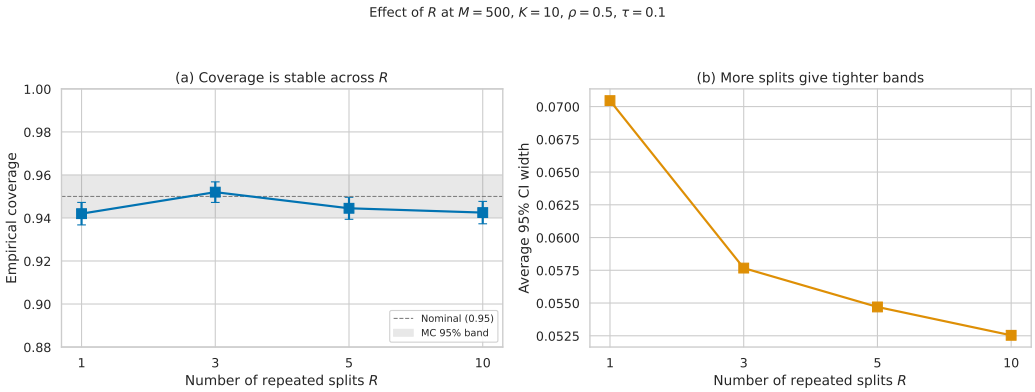

Figure 3: **Study A: Effect of the number of repeated splits** R at fixed M=500, K=10. (a) Coverage is stable across R (all within the MC 95% band). (b) Width contracts quickly up to R=5, with diminishing returns beyond.

SIREN matches a nonparametric item-bootstrap at a fraction of the cost. A practitioner could, in principle, bypass the influence-function machinery and instead resample items with replacement and rerun the entire repeated-split pipeline on each resample—a nonparametric item bootstrap that respects split dependence by construction. This comparison checks whether SIREN's influence-function approximation agrees with the item-bootstrap intervals up to Monte Carlo noise. Figure 4 (a) confirms this: at K=10, R=5 and $M \in \{200, 500, 1000\}$, both methods cover within the MC band (SIREN: 94.6–95.0%; item-bootstrap: 95.0–96.0%), and their average widths agree to within 0.8% in every configuration (Figure 4 (b), ratios 0.994–1.008). The methods differ in computational cost. On a representative configuration (M=500, K=10, R=5, B_{boot}=1000), the SIREN multiplier bootstrap runs in 7.5 ms, while the nonparametric item bootstrap takes 281 ms — a 37.6× speedup from replacing B_{boot} rerun pipelines with a single length-M dot product per resample. The gap grows with R, K, and B_{boot}: at the LLM-evaluation scale of the main MMLU-Pro experiments (Section 4.3) the same speedup reaches 35× over a paired bootstrap and $\sim 10^5\times$ over the IRT-based PromptEval. Beyond wall-clock efficiency, the influence function $\widehat{\psi}$ in (25) decomposes additively into a held-out evaluation contribution and a derivative-weighted selection contribution, which lets the same bootstrap deliver a variance decomposition by source and paired CIs across budget pairs — neither of which has any counterpart in the item-resample baseline.

Summary. Across 22 trial-configurations totaling more than 40,000 simulated trials, the SIREN multiplier bootstrap is empirically calibrated at the nominal level, scales with the theoretical $M^{-1/2}$ rate, and reproduces a nonparametric item-bootstrap's inference at a $\sim 38\times$ speedup. This establishes

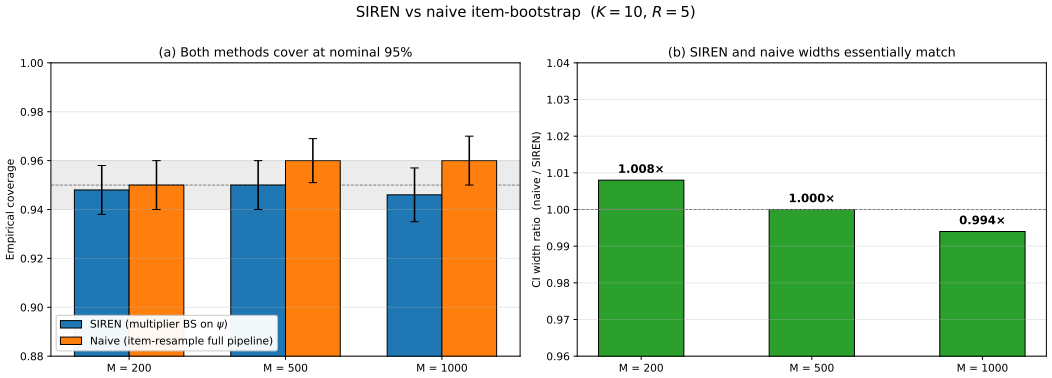

Figure 4: **Study A: SIREN versus a nonparametric item-bootstrap baseline** at K=10, R=5. (a) Both methods cover at the nominal level (error bars: $\pm 1.96 \cdot$SE). (b) CI width ratios are within $\pm 0.8\%$ of unity. The item-bootstrap is $\sim 38\times$ slower because it reruns the full repeated-split pipeline on every resample, while SIREN runs it once and bootstraps a length-M vector.

a clean empirical baseline before Studies B and C turn to the regimes (near-tie hard selection, same-data best-of reporting) where the theorem's regularity conditions are designed to fail.

C.1.2 Study B: Near-Tie Nonregularity of Hard Selection

Study B validates three operational claims about hard-selection nonregularity in the same two-artifact setting where it predicts failure: (i) hard argmax selection systematically undercovers when the top two artifacts are close in quality; (ii) soft (softmax) selection, whose smooth Jacobian is exactly the regularity condition required by Theorem 1, remains calibrated across all gaps; (iii) the winner-instability diagnostic $\widehat{\pi}_{\text{win}}$ serves as a practical trigger for switching between the two regimes.

Data generating process. Items have i.i.d. difficulties $\delta_i \sim \text{Uniform}(-2, 2)$. The two artifacts have qualities $q_1 = 0.5 + \Delta$ and $q_2 = 0.5$, so that Δ directly controls the population margin. Scores are $Z_{ik} \sim \text{Bernoulli}(\sigma(q_k - \delta_i))$. We fix M=500, R=5, ρ=0.5, τ=0.1, and B_{boot}=500, sweep Δ over 17 values from 0.00 to 0.80, and use N_{sim}=2,000 trials and N_{gt}=3,000 Monte Carlo replications per configuration (MC SE ≈ 0.5 pp at nominal coverage). For each method, ground truth $\theta^\star$ is computed separately by taking the expectation of that method's $\widetilde{\theta}$ under the same DGP, so that coverage measures whether each method's CI covers *its own* procedure-level target.

Main result: a U-shaped undercoverage curve for hard selection. Figure 5 (a) shows the coverage sweep. Hard-selection coverage traces a clear U-shape: it is close to nominal at Δ=0 (where the two artifacts are identical, so misselection is harmless), drops to a worst of 89.8% at Δ=0.20 and 0.30 (5.2 percentage points below nominal), and recovers to within MC noise once $\Delta \geq 0.35$. Soft selection, in contrast, holds between 93.7% and 96.0% coverage across the entire sweep. Table 7 reports representative points. The U-shape is the empirical signature of hard-selection nonregularity: at Δ=0 the two artifacts are interchangeable; at large Δ the winner is stable; between these two extremes is a window where the winner flips frequently yet misselection carries a real cost, and that is exactly where hard bootstrap fails.

Mechanism: hard bootstrap estimates the wrong variance. Figure 6 visualizes exactly why hard selection fails. For each gap we compare the true standard deviation of $\widetilde{\theta}$ across trials (solid) with the standard deviation implied by the average CI width (dashed; half-width divided by 1.96, i.e., the Gaussian-CI interpretation of the bootstrap quantile). Under soft selection the two curves track each other to within Monte Carlo noise at every Δ, confirming Theorem 2. Under hard selection the two curves diverge in the near-tie window: the true std climbs from ≈ 0.023 at Δ=0 to a peak of ≈ 0.027 around Δ=0.25, while the bootstrap-estimated std remains essentially flat near 0.022. The shaded region is the *missed variance*: a real source of randomness—the discrete jump in $\widehat{k}^\star$ across splits—that the plug-in influence function $\widehat{\psi}^{\text{hard}}$ is blind to because the argmax selector has zero

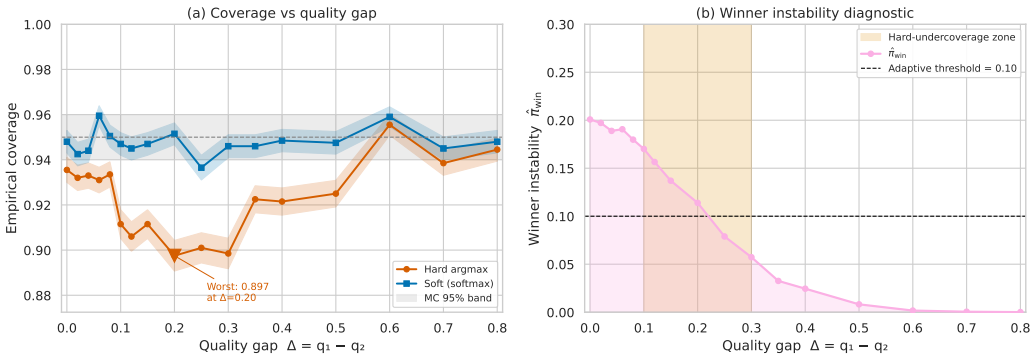

Figure 5: **Study B: Hard-selection coverage fails in the near-tie regime.** (a) Empirical coverage vs. quality gap Δ; error bands are $\pm$SE over N_{sim}=2,000 trials, grey band is the Monte Carlo 95% tolerance region around the nominal level. Hard (orange) drops to 89.8% at Δ=0.20; soft (blue) is stable across the sweep. (b) Winner instability $\widehat{\pi}_{\text{win}}$ and the adaptive threshold 0.10. The shaded orange band marks the empirical hard-undercoverage zone $\Delta \in [0.10, 0.30]$ and aligns with the regime where $\widehat{\pi}_{\text{win}}$ crosses the threshold from above.

Table 7: Study B: Empirical coverage (%) and winner instability at selected gap levels (M=500, K=2, R=5, Monte Carlo SE $\approx$ 0.5 pp at N_{sim}=2,000). Hard-selection coverage drops up to 5.2 pp below nominal; soft is stable.

Gap Δ	Hard (%)	Soft (%)	Adaptive (%)	$\widehat{\pi}_{\text{win}}$
0.00	93.6	94.8	93.8	0.201
0.06	93.1	96.0	94.1	0.191
0.10	91.1	94.7	92.8	0.170
0.15	91.1	94.7	93.2	0.137
0.20	89.8	95.2	93.5	0.114
0.30	89.8	94.6	93.5	0.057

derivative almost everywhere. At Δ=0.20 the bootstrap underestimates the std by approximately 14%, with the maximum miss of $\approx$ 15% occurring at Δ=0.25; both lie squarely inside the empirical undercoverage zone $\Delta \in [0.10, 0.30]$.

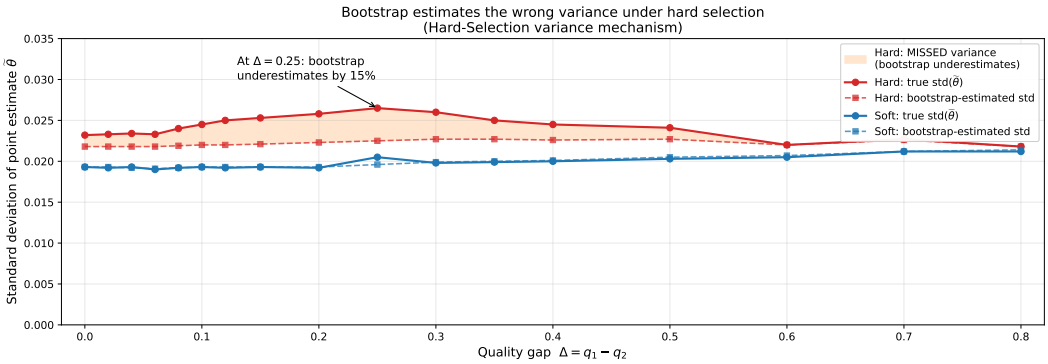

Figure 6: **Study B: Why hard-selection bootstrap undercovers.** The orange shaded area is the variance that the hard influence function $\widehat{\psi}^{\text{hard}}$ does not capture: the discontinuous jump of the argmax winner across splits. Soft selection has a non-zero Jacobian and therefore estimates its own variance correctly (blue lines track each other).

Instability-triggered adaptive selection. Figure 7 evaluates an instability-triggered adaptive rule: on each trial, use hard selection when the winner-instability frequency $\widehat{\pi}_{\text{win}}$ (the fraction of repeated

splits whose argmax differs from the across-split majority winner) satisfies $\widehat{\pi}_{\text{win}} \leq 0.10$, and switch to soft otherwise. Across the sweep, adaptive coverage improves on hard by 2–4 pp in the near-tie window, recovering to 93.5% at the worst gap Δ=0.20 (vs. 89.8% for hard). At Δ=0.40, the width ratios relative to hard selection are $\bar{w}_{\text{adapt}}/\bar{w}_{\text{hard}} = 0.99$ and $\bar{w}_{\text{soft}}/\bar{w}_{\text{hard}} = 0.89$ (Figure 7(b)). However, the rule does *not* fully close the gap at boundary points: at Δ=0.20 the population $\widehat{\pi}_{\text{win}}$ is 0.114, just above the threshold, so individual trials oscillate across the decision boundary and the ensemble coverage sits between hard and soft. Soft selection has the most stable coverage in this sweep. The adaptive rule does not inherit the smooth-selector guarantee because it switches discontinuously.

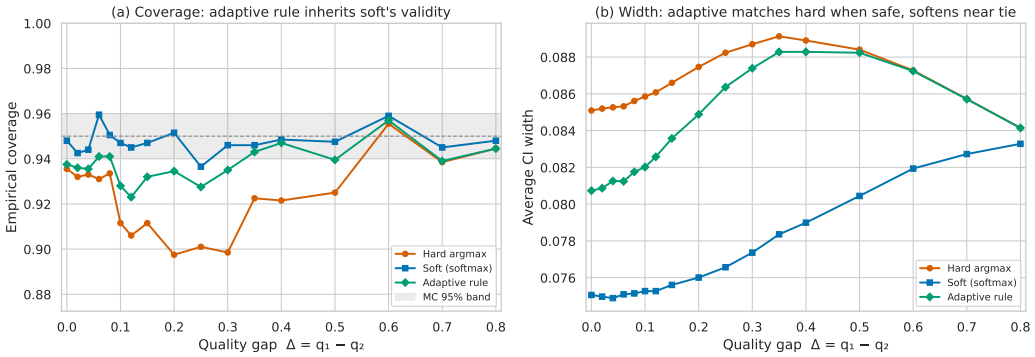

Figure 7: **Study B: Instability-triggered adaptive rule.** (a) Coverage of hard, soft, and adaptive across Δ. Adaptive (green) improves on hard near ties but does not fully match soft at boundary points where $\widehat{\pi}_{\text{win}} \approx 0.10$. (b) Average CI width: adaptive coincides with hard in the stable regime (large Δ) and pulls toward soft near ties, giving a dynamic trade-off between width and coverage.

Summary. Across 17 gap levels and 2,000 trials each, hard argmax selection is empirically nonregular in a narrow but operationally relevant window around $\Delta \approx 0.10$–0.30, where its bootstrap CIs lose up to 5.2 pp of coverage. The variance discrepancy arises because the plug-in influence function for argmax has zero derivative and therefore misses the selection-jump component of the true variance ($\approx 15\%$ underestimation of $\text{std}(\widetilde{\theta})$ at the peak). Soft selection includes the selection contribution through its Jacobian and maintains coverage close to nominal across the sweep. The winner-instability diagnostic identifies the sensitive region, but the adaptive rule retains some undercoverage near its switching threshold. Together, Studies A and B show that Theorem 1's smoothness assumption is relevant to interval calibration in the near-tie regime.

C.1.3 Study C: Same-Data Optimism

Study C empirically validates the same-data optimism phenomenon in a setting that is adversarial to any method relying on same-data best-of reporting: two systems with *identical* true performance but unequal search budgets. The question is whether the method under evaluation avoids declaring a spurious winner, or whether it inflates the score of whichever system happened to try more artifacts.

Data generating process. Items have i.i.d. difficulties $\delta_i \sim \text{Uniform}(-2, 2)$, and every artifact in every system shares the identical quality $q_k \equiv 0.5$, so all systems have common population mean $\theta^\star \approx 0.596$. We fix M=500, R=5, ρ=0.5, τ=0.1, and B_{boot}=500, with N_{sim}=2,000 trials and N_{gt}=10,000 Monte Carlo replications per configuration. The bias sweep (Table 8 and Figure 8 (left)) varies $H_A \in \{3, 5, 10, 20, 50\}$ against a fixed comparator System B with H_B=3 artifacts under shared item draws, so that bias and false-winner rate are measured on the same paired comparison. The unequal-search bar chart (Figure 8 (right)) sweeps six (H_A, H_B) pairs with independent item draws per system, the regime of practical interest where two teams report on disjoint evaluation runs.

Main result: same-data bias tracks the $\sqrt{2 \log H}$ theory. Table 8 and the left panel of Figure 8 report the optimism bias $\mathbb{E}[\widehat{\theta} - \theta^\star]$ of the two estimators on System A. Same-data best-of reporting inflates System A's score from $+1.5$ pp at H_A=3 to $+4.0$ pp at H_A=50, and its growth tracks the

Table 8: Study C: Same-data optimism (M=500, all artifacts identical, $\theta^\star \approx 0.596$, comparator H_B=3, paired item draws). The final columns report the frequency of declaring System A the winner. Same-data reporting compares point estimates, whereas SIREN uses nonoverlapping 95% CIs. The 50% symmetry reference applies to the point-estimate comparison at equal search effort.

	Optimism bias (pp)		FWR: A wins (%)	
H_A	Same-data	SIREN	Same-data	SIREN
3	$+1.5$	-0.2	48.3	0.0
5	$+2.3$	0.0	62.1	0.0
10	$+3.1$	$+0.1$	76.4	0.0
20	$+3.7$	$+0.1$	86.3	0.0
50	$+4.0$	-0.3	94.0	0.0

theoretical rate $\sigma\sqrt{2\log H_A}/\sqrt{M}$ (dashed line) up to a constant factor; the empirical bias sits below the theoretical curve because the maximum-of-Gaussians bound is conservative in finite-M Bernoulli scores. The repeated-split estimator $\widetilde{\theta}$ shows bias bounded by 0.4 pp across all library sizes, because its held-out evaluation is independent of the development scores used to form the softmax weights.

Mechanism: the winner of a noisy ensemble is biased upward. The root cause is an elementary extreme-value fact. For independent Gaussian estimation errors with standard error $\sigma/\sqrt{M}$, the expected maximum grows on the order of $\sigma\sqrt{2\log H}/\sqrt{M}$. Same-data best-of reporting picks the arg-max *and* reports its value on the same data that identified it as the winner, so the reported score inherits the entire extreme-value inflation. The repeated-split estimator evaluates the softmax weights on held-out items, separating the scores used to choose the weights from the scores used to report performance.

Effect of unequal search effort on rankings. The practical consequence appears when the two systems have unequal H. Same-data reporting inflates both systems' scores, but inflates the larger-library system more, creating a spurious gap between the two reports. At H_A=50 vs. H_B=3 in the paired-item table, this gap is $+4.0 - (+1.5) \approx 2.5$ pp of apparent advantage for System A, despite *identical* ground truth. The FWR columns of Table 8 translate this into false rankings: the probability that same-data reporting declares System A the winner climbs from 48.3% at H_A=3 (ties, near the 50% fair baseline) to $\mathbf{94.0}\%$ at H_A=50. The right panel of Figure 8 shows the same effect for the more practical regime in which two teams evaluate on independent item draws: across six (H_A, H_B) pairs, the win rate of the system with the larger search library ranges from 70.7% at $(20, 5)$ to 87.6% at $(100, 5)$, all well above the fair 50% baseline. SIREN's 95% CIs overlap in every reported trial, so the interval rule does not declare a winner. The two rules use different evidence thresholds, and overlapping intervals do not establish equivalence.

Summary. A benchmark report that compares two systems via their same-data best-of scores is not measuring procedure quality; it is partly measuring who searched harder. Across 5 paired library sizes and 6 unpaired pairings totaling 22,000 trials, the false-winner rate under same-data reporting grows from chance-level 48% to near-certain 94% as the search-effort imbalance widens, in qualitative agreement with the $\sqrt{2\log H}$ extreme-value theory. SIREN reports no winner under its interval rule in these simulations, while the point-estimate comparisons show the ranking effect of unequal search effort.

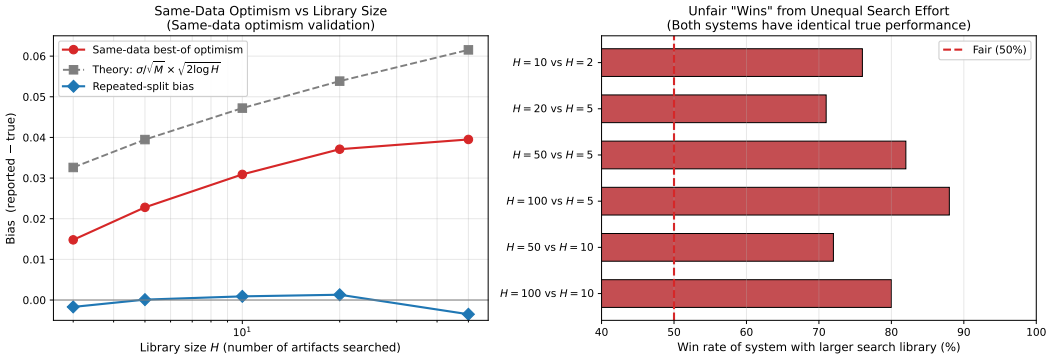

Figure 8: **Study C: same-data optimism validation.** *Left*: Optimism bias vs. library size H_A at fixed H_B=3 (paired item draws). Same-data reporting (red) produces bias growing roughly as $\sigma\sqrt{2\log H}/\sqrt{M}$ (dashed); the repeated-split estimator (blue) has near-zero bias. *Right*: When two systems with identical true quality search over unequal libraries (independent item draws), same-data reporting declares the system with the larger library the winner 71–88% of the time across six pairings, despite ground-truth equality.

C.2 Per-model SIREN-vs-baseline comparisons

For each (subject, tuner) cell we report the per-model directional calls and point estimates of SIREN against the four internal baselines: M1/M2 (naive max, Wald), M3 (single-split holdout), and M4 (R-split t-CI). Tables 9–11 give the per-model breakdown for all four cells. Across the four subject–tuner cells, SIREN has the highest aggregate directional accuracy under the common reporting reference in three cells and ties M1/M2 and M4 on Law with DSPy. The aggregated directional accuracy across these per-model breakdowns appears in Tables 1 and 2 of the main text.

Table 9: Unified per-model comparison across budgets on MMLU-Pro Math with the **random search** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\text{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\text{sign}(\theta^\star_A - \theta^\star_B)$. M1 and M2 are merged because they are numerically identical in point estimate and directional call.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.329	0.192	0.325	0.244	0.225	0.245	0.284	0.262	0.232	0.163	0.153
$500K$	$\theta^\star_A$	0.294	0.175	0.213	0.145	0.147	0.201	0.243	0.240	0.189	0.124	0.130
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	M1/M2	0.371 ✗	0.234 ✗	0.324 ✓	0.203 ✓	0.197 ✓	0.259 ✗	0.292 ✗	0.254 ✓	0.233 ✗	0.158 ✓	0.145 ✓
	M3	0.342 ✗	0.233 ✗	0.304 ✓	0.188 ✓	0.200 ✓	0.246 ✗	0.275 ✓	0.236 ✓	0.209 ✓	0.158 ✓	**0.129** ✓
	M4	0.365 ✗	0.235 ✗	0.323 ✓	0.198 ✓	0.183 ✓	0.252 ✗	0.284 ✗	0.243 ✓	0.224 ✓	0.151 ✓	0.134 ✓
	SIREN	**0.291** ✓	**0.174** ✓	**0.213** ✓	**0.144** ✓	**0.147** ✓	**0.202** ✓	**0.241** ✓	**0.238** ✓	**0.187** ✓	**0.125** ✓	**0.129** ✓
$1.5M$	$\theta^\star_A$	0.346	0.198	0.311	0.211	0.162	0.244	0.264	0.247	0.221	0.154	0.136
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	M1/M2	0.373 ✓	0.230 ✓	0.331 ✗	0.233 ✓	0.197 ✓	0.264 ✗	0.292 ✗	0.272 ✗	0.237 ✗	0.189 ✗	0.149 ✓
	M3	0.342 ✓	0.225 ✓	0.327 ✗	0.233 ✓	0.190 ✓	**0.244** ✓	0.275 ✓	0.259 ✓	0.209 ✓	0.188 ✗	0.129 ✓
	M4	0.359 ✓	0.222 ✓	0.329 ✗	0.227 ✓	0.191 ✓	0.256 ✗	0.289 ✗	0.265 ✗	0.227 ✓	0.191 ✗	**0.137** ✓
	SIREN	**0.343** ✓	**0.199** ✓	**0.312** ✓	**0.213** ✓	**0.163** ✓	0.245 ✗	**0.265** ✓	**0.246** ✓	**0.218** ✓	**0.155** ✓	**0.135** ✓
$3M$	$\theta^\star_A$	0.345	0.196	0.313	0.212	0.164	0.244	0.263	0.239	0.218	0.155	0.137
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	M1/M2	0.373 ✓	0.237 ✓	0.329 ✗	0.226 ✓	0.201 ✓	0.264 ✗	0.289 ✗	0.267 ✗	0.241 ✗	0.171 ✗	0.162 ✗
	M3	**0.342** ✓	0.220 ✓	0.323 ✓	**0.214** ✓	0.176 ✓	**0.244** ✓	0.270 ✓	0.214 ✓	0.227 ✓	0.169 ✗	0.158 ✗
	M4	0.359 ✓	0.238 ✓	0.317 ✓	0.217 ✓	0.189 ✓	0.255 ✗	0.273 ✓	0.251 ✓	0.230 ✓	0.163 ✗	0.154 ✗
	SIREN	**0.342** ✓	**0.196** ✓	**0.313** ✓	0.215 ✓	**0.165** ✓	0.246 ✗	**0.265** ✓	**0.238** ✓	**0.216** ✓	**0.156** ✓	**0.136** ✓
$6.5M$	$\theta^\star_A$	0.353	0.200	0.319	0.213	0.187	0.239	0.257	0.248	0.227	0.153	0.146
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	M1/M2	0.373 ✓	0.237 ✓	0.322 ✓	0.233 ✓	0.219 ✓	0.264 ✗	0.280 ✓	0.269 ✗	0.241 ✗	0.171 ✗	0.163 ✗
	M3	0.343 ✓	0.220 ✓	0.297 ✓	0.219 ✓	0.206 ✓	0.244 ✓	0.251 ✓	**0.248** ✓	0.214 ✓	0.171 ✗	0.155 ✗
	M4	0.359 ✓	0.235 ✓	0.311 ✓	0.220 ✓	0.208 ✓	0.261 ✗	0.269 ✓	0.260 ✓	0.232 ✗	0.161 ✓	0.157 ✗
	SIREN	**0.350** ✓	**0.201** ✓	**0.320** ✓	**0.214** ✓	**0.189** ✓	**0.243** ✓	**0.255** ✓	0.246 ✓	**0.225** ✓	**0.153** ✓	**0.146** ✓

Table 10: Unified per-model comparison across budgets on MMLU-Pro Law with the **random search** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\mathrm{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$. M1 and M2 are merged because they are numerically identical in point estimate and directional call.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.359	0.318	0.345	0.315	0.227	0.269	0.311	0.234	0.282	0.225	0.185
$500K$	$\theta^\star_A$	0.317	0.293	0.324	0.294	0.222	0.261	0.300	0.224	0.272	0.230	0.172
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B
	M1/M2	0.364 ✗	0.312 ✓	0.330 ✓	0.314 ✓	0.227 ✗	0.275 ✗	0.315 ✗	0.237 ✗	0.285 ✗	0.234 ✓	0.185 ✗
	M3	0.377 ✗	0.325 ✗	0.335 ✓	0.317 ✗	0.238 ✗	0.281 ✗	0.309 ✓	0.236 ✗	0.283 ✗	0.246 ✓	0.196 ✗
	M4	0.359 ✗	0.318 ✓	0.330 ✓	0.305 ✓	0.220 ✓	0.272 ✗	0.303 ✓	**0.226** ✓	**0.271** ✓	**0.229** ✓	0.186 ✗
	SIREN	**0.317** ✓	**0.300** ✓	**0.328** ✓	**0.291** ✓	**0.223** ✓	**0.262** ✓	**0.298** ✓	0.220 ✓	0.267 ✓	0.234 ✓	**0.176** ✓
$1.5M$	$\theta^\star_A$	0.308	0.298	0.324	0.283	0.223	0.265	0.304	0.235	0.274	0.217	0.165
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B	A<B	A<B
	M1/M2	0.364 ✗	0.318 ✗	0.338 ✓	0.325 ✗	0.251 ✗	0.275 ✗	0.320 ✗	0.244 ✓	0.285 ✗	0.240 ✗	0.185 ✗
	M3	0.377 ✗	0.311 ✓	0.329 ✓	0.339 ✗	0.259 ✗	0.271 ✗	0.309 ✓	**0.232** ✗	0.253 ✓	0.248 ✗	0.140 ✓
	M4	0.359 ✗	0.318 ✗	0.329 ✓	0.323 ✗	0.245 ✗	0.263 ✓	**0.304** ✓	0.230 ✗	0.264 ✓	0.234 ✗	0.176 ✓
	SIREN	**0.309** ✓	**0.305** ✓	**0.326** ✓	**0.283** ✓	**0.226** ✓	**0.265** ✓	0.303 ✓	**0.232** ✗	**0.270** ✓	**0.220** ✓	**0.168** ✓
$3M$	$\theta^\star_A$	0.324	0.294	0.325	0.281	0.224	0.265	0.308	0.241	0.273	0.230	0.173
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B	A>B	A<B
	M1/M2	0.364 ✗	0.318 ✗	0.339 ✓	0.327 ✗	0.249 ✗	0.277 ✗	0.328 ✗	0.261 ✓	0.280 ✓	0.246 ✓	0.185 ✗
	M3	0.377 ✗	0.327 ✗	0.317 ✓	0.301 ✓	0.244 ✗	0.257 ✓	0.321 ✗	0.251 ✓	0.253 ✓	0.257 ✓	0.194 ✗
	M4	0.359 ✗	0.321 ✗	0.330 ✓	0.321 ✗	0.236 ✗	0.262 ✓	0.320 ✗	0.246 ✓	0.259 ✓	0.235 ✓	0.181 ✓
	SIREN	**0.326** ✓	**0.302** ✓	**0.327** ✓	**0.279** ✓	**0.225** ✓	**0.264** ✓	**0.308** ✓	**0.238** ✓	**0.270** ✓	**0.232** ✓	**0.176** ✓
$6.5M$	$\theta^\star_A$	0.318	0.294	0.325	0.288	0.240	0.261	0.309	0.242	0.279	0.224	0.176
	True dir.	A<B	A<B	A<B	A<B	A>B	A<B	A<B	A>B	A<B	A<B	A<B
	M1/M2	0.335 ✓	0.303 ✓	0.342 ✓	0.330 ✗	0.265 ✓	0.284 ✗	0.319 ✗	0.258 ✓	0.292 ✗	0.240 ✗	0.185 ✗
	M3	0.323 ✓	0.287 ✓	0.339 ✓	0.355 ✗	0.267 ✓	0.297 ✗	0.305 ✓	0.251 ✓	0.283 ✗	0.246 ✗	0.190 ✗
	M4	0.328 ✓	**0.298** ✓	0.338 ✓	0.328 ✗	0.261 ✓	0.284 ✗	0.305 ✓	**0.244** ✓	**0.278** ✓	0.234 ✗	0.180 ✓
	SIREN	**0.320** ✓	0.300 ✓	**0.329** ✓	**0.288** ✓	**0.240** ✓	**0.262** ✓	**0.308** ✓	0.239 ✓	0.276 ✓	**0.225** ✗	**0.179** ✓

Table 11: Unified per-model comparison across budgets on MMLU-Pro Law with the **DSPy** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\mathrm{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$. M1 and M2 are merged because they are numerically identical in point estimate and directional call.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.340	0.312	0.334	0.328	0.219	0.256	0.302	0.251	0.257	0.201	0.169
$500K$	$\theta^\star_A$	0.349	0.327	0.357	0.329	0.244	0.284	0.317	0.256	0.279	0.236	0.166
	True dir.	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A<B
	M1/M2	0.360 ✓	0.333 ✓	0.361 ✓	0.339 ✓	0.250 ✓	0.296 ✓	0.323 ✓	**0.261** ✓	0.286 ✓	0.246 ✓	0.177 ✗
	M3	0.371 ✓	0.335 ✓	0.329 ✗	0.351 ✓	**0.246** ✓	0.279 ✓	0.321 ✓	0.238 ✗	0.269 ✓	0.228 ✓	0.176 ✗
	M4	**0.350** ✓	0.332 ✓	0.351 ✓	**0.325** ✗	0.248 ✓	0.292 ✓	0.312 ✓	0.242 ✗	0.271 ✓	**0.233** ✓	0.173 ✗
	SIREN	0.346 ✓	**0.331** ✓	**0.356** ✓	**0.325** ✗	0.247 ✓	**0.284** ✓	**0.314** ✓	0.247 ✗	**0.273** ✓	0.231 ✓	**0.166** ✓
$1.5M$	$\theta^\star_A$	0.353	0.331	0.359	0.323	0.243	0.284	0.323	0.258	0.285	0.237	0.172
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	M1/M2	0.374 ✓	0.341 ✓	0.372 ✓	0.330 ✗	0.248 ✓	0.290 ✓	0.342 ✓	0.269 ✓	0.294 ✓	0.253 ✓	0.180 ✓
	M3	0.387 ✓	**0.335** ✓	0.363 ✓	0.341 ✗	0.250 ✓	0.289 ✓	0.349 ✓	0.246 ✗	0.277 ✓	**0.234** ✓	0.176 ✓
	M4	0.366 ✓	0.337 ✓	0.365 ✓	**0.319** ✓	**0.240** ✓	0.281 ✓	0.336 ✓	**0.253** ✓	0.280 ✓	0.247 ✓	**0.171** ✓
	SIREN	**0.350** ✓	**0.335** ✓	**0.356** ✓	0.318 ✓	**0.246** ✓	**0.284** ✓	**0.320** ✓	0.250 ✗	**0.281** ✓	**0.234** ✓	0.174 ✓
$3M$	$\theta^\star_A$	0.345	0.323	0.356	0.326	0.248	0.284	0.325	0.258	0.279	0.238	0.175
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	M1/M2	0.358 ✓	0.339 ✓	0.362 ✓	0.339 ✗	0.258 ✓	0.294 ✓	0.342 ✓	0.269 ✓	0.290 ✓	0.248 ✓	0.187 ✓
	M3	0.379 ✓	0.329 ✓	0.343 ✓	0.339 ✗	0.267 ✓	0.287 ✓	0.349 ✓	0.242 ✗	0.265 ✓	0.255 ✓	0.186 ✓
	M4	0.349 ✓	0.337 ✓	0.351 ✓	**0.327** ✓	**0.249** ✓	0.287 ✓	0.334 ✓	**0.254** ✓	**0.277** ✓	**0.240** ✓	**0.178** ✓
	SIREN	**0.344** ✓	**0.326** ✓	**0.352** ✓	0.321 ✓	0.251 ✓	**0.283** ✓	**0.322** ✓	0.251 ✗	0.275 ✓	0.235 ✓	**0.178** ✓
$6.5M$	$\theta^\star_A$	0.346	0.323	0.357	0.324	0.244	0.282	0.322	0.259	0.283	0.240	0.176
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	M1/M2	0.358 ✓	0.339 ✓	0.362 ✓	0.333 ✗	0.251 ✓	0.294 ✓	0.342 ✓	0.269 ✓	0.291 ✓	0.248 ✓	0.187 ✓
	M3	0.379 ✓	0.329 ✓	0.343 ✓	**0.325** ✓	**0.246** ✓	0.287 ✓	0.349 ✓	0.242 ✗	0.265 ✓	0.255 ✓	0.162 ✗
	M4	0.349 ✓	0.337 ✓	0.349 ✓	0.321 ✓	0.240 ✓	0.286 ✓	0.334 ✓	0.250 ✗	0.273 ✓	**0.241** ✓	**0.176** ✓
	SIREN	**0.345** ✓	**0.326** ✓	**0.353** ✓	0.320 ✓	0.248 ✓	**0.281** ✓	**0.320** ✓	**0.251** ✓	**0.278** ✓	0.237 ✓	0.177 ✓

C.3 Per-model PromptEval-vs-SIREN comparisons

We next report the same per-model breakdown for our external baseline, PromptEval [28], sweeping its cell-observation fraction $f \in \{0.05, 0.10, 0.20, 0.40, 0.60, 0.80, 1.00\}$. Tables 12–14 give the

per-model point estimates and directional calls for each (f, B) combination across all four (subject, tuner) cells. A consistent picture emerges: PromptEval's signed error transitions from strongly negative at small f (because the Rasch model shrinks strong prompts toward the mean when the score matrix is sparse) through a positive plateau at large f (where dense observation reintroduces the same selection bias as M1), while SIREN remains within ± 0.21 pp of the Monte Carlo target at every budget across all four cells.

Table 12: Unified PromptEval [28] vs SIREN comparison across budgets on MMLU-Pro Math with the **random search** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\mathrm{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.329	0.192	0.325	0.244	0.225	0.245	0.284	0.262	0.232	0.163	0.153
$500K$	$\theta^\star_A$	0.294	0.175	0.213	0.145	0.147	0.201	0.243	0.240	0.189	0.124	0.130
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	PE f=0.05	0.260✓	0.147✓	0.283✓	0.140✓	0.120✓	0.173✓	0.194✓	0.201✓	0.176✓	0.097✓	0.125✓
	PE f=0.10	0.309✓	0.191✓	0.261✓	0.160✓	0.170✓	0.219✓	0.252✓	0.264✗	0.212✓	0.119✓	0.149✓
	PE f=0.20	0.348✗	0.241✗	0.357✗	0.219✓	0.221✓	0.269✗	0.286✗	0.298✗	0.240✗	0.149✓	0.170✗
	PE f=0.40	0.360✗	0.228✗	0.349✗	0.218✓	0.211✓	0.261✗	0.283✓	0.267✗	0.242✗	0.154✓	0.157✗
	PE f=0.60	0.370✗	0.233✗	0.332✗	0.215✓	0.199✓	0.258✗	0.293✗	0.258✓	0.234✗	0.155✓	0.152✓
	PE f=0.80	0.372✗	0.232✗	0.327✗	0.205✓	0.198✓	0.260✗	0.299✗	0.253✓	0.233✗	0.155✓	0.149✓
	PE f=1.00 (M1)	0.371✗	0.234✗	0.324✓	0.203✓	0.197✓	0.259✗	0.292✗	0.254✓	0.233✗	0.158✓	0.145✓
	SIREN	**0.291**✓	**0.174**✓	**0.213**✓	**0.144**✓	**0.147**✓	**0.202**✓	**0.241**✓	**0.238**✓	**0.187**✓	**0.125**✓	**0.129**✓
$1.5M$	$\theta^\star_A$	0.346	0.198	0.311	0.211	0.162	0.244	0.264	0.247	0.221	0.154	0.136
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	PE f=0.05	0.296✗	0.180✗	0.274✓	0.163✓	0.147✓	0.200✓	0.204✓	0.206✓	0.199✓	0.122✓	0.099✓
	PE f=0.10	**0.349**✓	0.216✓	0.318✓	0.227✓	0.184✓	0.256✗	0.276✓	0.264✗	0.237✗	0.166✗	0.149✓
	PE f=0.20	0.396✓	0.241✓	0.352✗	0.238✓	0.240✗	0.283✗	0.290✗	0.292✗	0.259✗	0.199✗	0.158✗
	PE f=0.40	0.383✓	0.228✓	0.328✗	0.233✓	0.209✓	0.286✗	0.307✗	0.285✗	0.244✗	0.189✗	0.151✓
	PE f=0.60	0.374✓	0.229✓	0.325✗	0.232✓	0.204✓	0.270✗	0.302✗	0.274✗	0.236✗	0.189✗	0.148✓
	PE f=0.80	0.373✓	0.224✓	0.330✗	0.234✓	0.198✓	0.260✗	0.292✗	0.271✗	0.237✗	0.184✗	0.146✓
	PE f=1.00 (M1)	0.373✓	0.230✓	0.331✗	0.233✓	0.197✓	0.264✗	0.292✗	0.272✗	0.237✗	0.189✗	0.149✓
	SIREN	**0.343**✓	**0.199**✓	**0.312**✓	**0.213**✓	**0.163**✓	**0.245**✗	**0.265**✓	**0.246**✓	**0.218**✓	**0.155**✓	**0.135**✓
$3M$	$\theta^\star_A$	0.345	0.196	0.313	0.212	0.164	0.244	0.263	0.239	0.218	0.155	0.137
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	PE f=0.05	0.366✓	0.144✗	0.275✓	0.159✓	0.104✓	0.213✓	0.208✓	0.202✓	0.187✓	0.117✓	0.116✓
	PE f=0.10	0.369✓	0.206✓	0.323✓	0.218✓	0.172✓	0.275✗	0.282✓	0.263✗	0.236✗	0.169✗	0.150✓
	PE f=0.20	0.382✓	0.250✓	0.340✗	0.251✗	0.212✓	0.280✗	0.302✗	0.282✗	0.246✗	0.194✗	0.179✗
	PE f=0.40	0.397✓	0.247✓	0.326✗	0.230✓	0.204✓	0.288✗	0.292✗	0.277✗	0.253✗	0.188✗	0.175✗
	PE f=0.60	0.391✓	0.243✓	0.330✗	0.231✓	0.214✓	0.265✗	0.296✗	0.264✗	0.249✗	0.176✗	0.162✗
	PE f=0.80	0.375✓	0.234✓	0.330✗	0.226✓	0.204✓	0.268✗	0.291✗	0.263✗	0.242✗	0.174✗	0.164✗
	PE f=1.00 (M1)	0.373✓	0.237✓	0.329✗	0.226✓	0.201✓	0.264✗	0.289✗	0.267✗	0.241✗	0.171✗	0.162✗
	SIREN	**0.342**✓	**0.196**✓	**0.313**✓	**0.215**✓	**0.165**✓	**0.246**✗	**0.265**✓	**0.238**✓	**0.216**✓	**0.156**✓	**0.136**✓
$6.5M$	$\theta^\star_A$	0.353	0.200	0.319	0.213	0.187	0.239	0.257	0.248	0.227	0.153	0.146
	True dir.	A>B	A>B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B
	PE f=0.05	0.287✗	0.179✗	0.274✓	0.172✓	0.138✓	0.199✓	0.200✓	0.193✓	0.200✓	0.119✓	0.117✓
	PE f=0.10	0.365✓	0.221✓	0.325✗	0.225✓	0.206✓	0.262✗	0.277✓	0.252✓	0.244✗	0.165✗	0.166✗
	PE f=0.20	0.400✓	0.245✓	0.329✗	0.235✓	0.250✗	0.277✗	0.295✗	0.277✗	0.249✗	0.183✗	0.168✗
	PE f=0.40	0.377✓	0.251✓	0.321✓	0.236✓	0.230✗	0.283✗	0.288✗	0.276✗	0.245✗	0.177✗	0.169✗
	PE f=0.60	0.372✓	0.246✓	0.327✗	0.231✓	0.237✗	0.266✗	0.276✓	0.264✗	0.243✗	0.171✗	0.164✗
	PE f=0.80	0.373✓	0.239✓	0.328✗	0.236✓	0.228✗	0.273✗	0.283✓	0.270✗	0.243✗	0.174✗	0.160✗
	PE f=1.00 (M1)	0.373✓	0.237✓	0.322✓	0.233✓	0.219✓	0.264✗	0.280✓	0.269✗	0.241✗	0.171✗	0.163✗
	SIREN	**0.350**✓	**0.201**✓	**0.320**✓	**0.214**✓	**0.189**✓	**0.243**✓	**0.255**✓	**0.246**✓	**0.225**✓	**0.153**✓	**0.146**✓

Table 13: Unified PromptEval [28] vs SIREN comparison across budgets on MMLU-Pro Law with the **random search** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\mathrm{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.359	0.318	0.345	0.315	0.227	0.269	0.311	0.234	0.282	0.225	0.185
$500K$	$\theta^\star_A$	0.317	0.293	0.324	0.294	0.222	0.261	0.300	0.224	0.272	0.230	0.172
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B
	PE f=0.05	0.248 ✓	0.162 ✓	0.157 ✓	0.141 ✓	0.119 ✓	0.172 ✓	0.137 ✓	0.087 ✓	0.163 ✓	0.121 ✗	0.128 ✓
	PE f=0.10	0.254 ✓	0.169 ✓	0.190 ✓	0.192 ✓	0.146 ✓	0.209 ✓	0.178 ✓	0.112 ✓	0.225 ✓	0.122 ✗	0.123 ✓
	PE f=0.20	0.311 ✓	0.227 ✓	0.244 ✓	0.234 ✓	0.177 ✓	0.283 ✗	0.204 ✓	0.164 ✓	0.280 ✓	0.158 ✗	0.154 ✓
	PE f=0.40	0.362 ✗	**0.299** ✓	0.339 ✓	0.318 ✗	0.243 ✗	0.275 ✗	0.302 ✓	0.235 ✗	0.291 ✗	0.235 ✓	0.196 ✗
	PE f=0.60	0.360 ✗	0.301 ✓	0.335 ✓	0.323 ✗	0.229 ✗	0.285 ✗	**0.301** ✓	0.230 ✓	0.282 ✗	0.237 ✓	0.197 ✗
	PE f=0.80	0.357 ✓	0.311 ✓	0.335 ✓	0.321 ✗	0.228 ✗	0.284 ✗	0.313 ✗	0.236 ✗	0.278 ✓	0.236 ✓	0.192 ✗
	PE f=1.00 (M1)	0.364 ✗	0.312 ✓	0.330 ✓	0.314 ✓	0.227 ✗	0.275 ✗	0.315 ✗	0.237 ✗	0.285 ✗	**0.234** ✓	0.185 ✗
	SIREN	**0.317** ✓	0.300 ✓	**0.328** ✓	**0.291** ✓	**0.223** ✓	**0.262** ✓	0.298 ✓	**0.220** ✓	**0.267** ✓	**0.234** ✓	**0.176** ✓
$1.5M$	$\theta^\star_A$	0.308	0.298	0.324	0.283	0.223	0.265	0.304	0.235	0.274	0.217	0.165
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B	A<B	A<B
	PE f=0.05	0.236 ✓	0.257 ✓	0.243 ✓	0.303 ✓	0.187 ✓	0.225 ✓	0.265 ✓	0.206 ✗	0.265 ✓	0.191 ✓	0.128 ✓
	PE f=0.10	0.325 ✓	**0.300** ✓	0.320 ✓	0.305 ✓	**0.225** ✓	0.277 ✗	0.334 ✗	0.256 ✓	0.299 ✗	0.241 ✗	0.174 ✓
	PE f=0.20	0.345 ✓	0.322 ✗	0.352 ✗	0.345 ✗	0.246 ✗	0.287 ✗	0.354 ✗	0.274 ✓	0.305 ✗	0.270 ✗	0.207 ✗
	PE f=0.40	0.353 ✓	0.318 ✓	0.351 ✗	0.345 ✗	0.247 ✗	0.279 ✗	0.338 ✗	0.253 ✓	0.293 ✗	0.253 ✗	0.197 ✗
	PE f=0.60	0.365 ✗	0.317 ✓	0.341 ✓	0.330 ✗	0.238 ✗	0.278 ✗	0.325 ✗	0.248 ✓	0.289 ✗	0.248 ✗	0.191 ✗
	PE f=0.80	0.362 ✗	0.322 ✗	0.340 ✓	0.323 ✗	0.252 ✗	0.279 ✗	0.321 ✗	0.243 ✓	0.284 ✗	0.243 ✗	0.190 ✗
	PE f=1.00 (M1)	0.364 ✗	0.318 ✗	0.338 ✓	0.325 ✗	0.251 ✗	0.275 ✗	0.320 ✗	0.244 ✓	0.285 ✗	0.240 ✗	0.185 ✗
	SIREN	**0.309** ✓	0.305 ✓	**0.326** ✓	**0.283** ✓	0.226 ✓	**0.265** ✓	**0.303** ✓	**0.232** ✗	**0.270** ✓	**0.220** ✓	**0.168** ✓
$3M$	$\theta^\star_A$	0.324	0.294	0.325	0.281	0.224	0.265	0.308	0.241	0.273	0.230	0.173
	True dir.	A<B	A<B	A<B	A<B	A<B	A<B	A<B	A>B	A<B	A>B	A<B
	PE f=0.05	0.314 ✓	0.284 ✓	0.344 ✓	0.296 ✓	0.218 ✓	0.247 ✓	0.277 ✓	0.274 ✓	0.254 ✓	0.204 ✗	0.145 ✓
	PE f=0.10	0.350 ✓	0.308 ✓	0.351 ✗	0.329 ✗	0.261 ✗	0.282 ✗	0.327 ✗	0.274 ✓	0.296 ✗	0.242 ✓	0.189 ✗
	PE f=0.20	0.372 ✗	0.333 ✗	0.360 ✗	0.326 ✗	0.267 ✗	0.297 ✗	0.329 ✗	0.279 ✓	0.308 ✗	0.258 ✓	0.199 ✗
	PE f=0.40	0.362 ✗	0.315 ✓	0.363 ✗	0.338 ✗	0.251 ✗	0.276 ✗	0.333 ✗	0.275 ✓	0.306 ✗	0.258 ✓	0.187 ✗
	PE f=0.60	0.367 ✗	0.312 ✓	0.350 ✗	0.342 ✗	0.248 ✗	0.283 ✗	0.327 ✗	0.273 ✓	0.281 ✓	0.252 ✓	0.186 ✗
	PE f=0.80	0.358 ✓	0.317 ✓	0.346 ✗	0.331 ✗	0.250 ✗	0.280 ✗	0.330 ✗	0.269 ✓	0.283 ✗	0.248 ✓	0.185 ✗
	PE f=1.00 (M1)	0.364 ✗	0.318 ✗	0.339 ✓	0.327 ✗	0.249 ✗	0.277 ✗	0.328 ✗	0.261 ✓	0.280 ✓	0.246 ✓	0.185 ✗
	SIREN	**0.326** ✓	**0.302** ✓	**0.327** ✓	**0.279** ✓	**0.225** ✓	**0.264** ✓	**0.308** ✓	**0.238** ✓	**0.270** ✓	**0.232** ✓	**0.176** ✓
$6.5M$	$\theta^\star_A$	0.318	0.294	0.325	0.288	0.240	0.261	0.309	0.242	0.279	0.224	0.176
	True dir.	A<B	A<B	A<B	A<B	A>B	A<B	A<B	A>B	A<B	A<B	A<B
	PE f=0.05	0.250 ✓	0.248 ✓	0.287 ✓	0.271 ✓	0.155 ✗	0.226 ✓	0.288 ✓	0.231 ✗	0.234 ✓	0.209 ✓	0.170 ✓
	PE f=0.10	0.341 ✓	0.306 ✓	0.344 ✓	0.318 ✗	0.223 ✗	0.277 ✗	0.331 ✗	0.269 ✓	0.287 ✗	0.245 ✗	0.213 ✗
	PE f=0.20	0.355 ✓	0.330 ✗	0.347 ✗	0.353 ✗	0.269 ✓	0.298 ✗	0.358 ✗	0.285 ✓	0.309 ✗	0.255 ✗	0.215 ✗
	PE f=0.40	0.354 ✓	0.311 ✓	0.346 ✗	0.338 ✗	0.269 ✓	0.279 ✗	0.330 ✗	0.272 ✓	0.317 ✗	0.249 ✗	0.204 ✗
	PE f=0.60	0.344 ✓	0.311 ✓	0.344 ✓	0.331 ✗	0.269 ✓	0.281 ✗	0.326 ✗	0.262 ✓	0.299 ✗	0.243 ✗	0.194 ✗
	PE f=0.80	0.337 ✓	0.306 ✓	0.344 ✓	0.324 ✗	0.260 ✓	0.284 ✗	0.322 ✗	0.264 ✓	0.292 ✗	0.243 ✗	0.187 ✗
	PE f=1.00 (M1)	0.335 ✓	0.303 ✓	0.342 ✓	0.330 ✗	0.265 ✓	0.284 ✗	0.319 ✗	0.258 ✓	0.292 ✗	0.240 ✗	0.185 ✗
	SIREN	**0.320** ✓	**0.300** ✓	**0.329** ✓	**0.288** ✓	**0.240** ✓	**0.262** ✓	**0.308** ✓	**0.239** ✓	**0.276** ✓	**0.225** ✗	**0.179** ✓

Table 14: Unified PromptEval [28] vs SIREN comparison across budgets on MMLU-Pro Law with the **DSPy** tuner. The untuned-pipeline-default reference $\theta^\star_B$ is fixed across budgets and shown once at the top. For each budget, $\theta^\star_A$ is the common soft-rule reference, and the method rows report scores $\hat{\theta}_A$ compared with it. The ✓/✗marker indicates whether $\mathrm{sign}(\hat{\theta}_A - \theta^\star_B)$ agrees with the true direction $\mathrm{sign}(\theta^\star_A - \theta^\star_B)$.

B	Row	Qwen3	Phi-3.5	Qwen2.5-7B	Llama3.1	Yi-1.5	InternLM	Qwen2-7B	Qwen2.5-3B	GLM-4	Mistr-v0.3	Mistr-v0.1
Ref.	$\theta^\star_B$	0.340	0.312	0.334	0.328	0.219	0.256	0.302	0.251	0.257	0.201	0.169
$500K$	$\theta^\star_A$	0.349	0.327	0.357	0.329	0.244	0.284	0.317	0.256	0.279	0.236	0.166
	True dir.	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A>B	A<B
	PE f=0.05	0.237✗	0.179✗	0.207✗	0.168✗	0.106✗	0.139✗	0.153✗	0.153✗	0.174✗	0.122✗	0.108✓
	PE f=0.10	0.223✗	0.197✗	0.269✗	0.218✗	0.145✗	0.202✗	0.211✗	0.195✗	0.213✗	0.152✗	0.114✓
	PE f=0.20	0.314✗	0.259✗	0.365✓	0.317✗	0.182✗	0.261✓	0.285✗	0.240✗	0.295✓	0.215✓	0.156✓
	PE f=0.40	0.376✓	0.337✓	0.372✓	0.347✓	0.252✓	0.296✓	0.331✓	**0.259**✓	0.293✓	0.249✓	0.197✗
	PE f=0.60	0.365✓	**0.330**✓	0.361✓	0.342✓	0.252✓	0.287✓	0.323✓	0.268✓	0.289✓	0.244✓	0.189✗
	PE f=0.80	0.364✓	0.332✓	0.362✓	0.338✓	0.249✓	0.289✓	0.321✓	**0.259**✓	0.289✓	0.248✓	0.176✗
	PE f=1.00 (M1)	0.360✓	0.333✓	0.361✓	0.339✓	0.250✓	0.296✓	0.323✓	0.261✓	0.286✓	0.246✓	0.177✗
	SIREN	**0.346**✓	0.331✓	**0.356**✓	**0.325**✗	**0.247**✓	**0.284**✓	**0.314**✓	0.247✗	**0.273**✓	**0.231**✓	**0.166**✓
$1.5M$	$\theta^\star_A$	0.353	0.331	0.359	0.323	0.243	0.284	0.323	0.258	0.285	0.237	0.172
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	PE f=0.05	0.261✗	0.196✗	0.287✗	0.300✓	0.163✗	0.163✗	0.210✗	0.162✗	0.197✗	0.181✗	0.135✗
	PE f=0.10	0.277✗	0.246✗	0.314✗	0.284✓	0.200✗	0.219✗	0.254✗	0.189✗	0.243✗	0.221✓	0.135✗
	PE f=0.20	0.365✓	0.340✓	0.374✓	0.343✗	0.276✓	0.295✓	0.335✓	0.279✓	0.300✓	0.254✓	0.194✓
	PE f=0.40	0.378✓	0.336✓	0.369✓	0.340✗	0.250✓	0.289✓	0.330✓	0.282✓	0.304✓	0.259✓	0.191✓
	PE f=0.60	0.368✓	0.339✓	0.371✓	0.335✗	0.255✓	0.294✓	0.335✓	0.276✓	0.297✓	0.246✓	0.188✓
	PE f=0.80	0.376✓	0.336✓	0.370✓	0.333✗	0.250✓	0.291✓	0.342✓	0.274✓	0.295✓	0.251✓	0.188✓
	PE f=1.00 (M1)	0.374✓	0.341✓	0.372✓	0.330✗	0.248✓	0.290✓	0.342✓	0.269✓	0.294✓	0.253✓	0.180✓
	SIREN	**0.350**✓	**0.335**✓	**0.356**✓	**0.318**✓	**0.246**✓	**0.284**✓	**0.320**✓	**0.250**✗	**0.281**✓	**0.234**✓	**0.174**✓
$3M$	$\theta^\star_A$	0.345	0.323	0.356	0.326	0.248	0.284	0.325	0.258	0.279	0.238	0.175
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	PE f=0.05	0.257✗	0.192✗	0.286✗	0.256✓	0.152✗	0.167✗	0.216✗	0.196✗	0.182✗	0.159✗	0.149✗
	PE f=0.10	0.284✗	0.239✗	0.299✗	0.272✓	0.208✗	0.226✗	0.252✗	0.198✗	0.219✗	0.192✗	0.156✗
	PE f=0.20	0.356✓	**0.326**✓	0.376✓	0.343✗	0.281✓	0.304✓	0.335✓	0.270✓	0.288✓	0.248✓	0.205✓
	PE f=0.40	0.363✓	0.340✓	0.377✓	0.337✗	0.272✓	0.298✓	0.337✓	0.276✓	0.290✓	0.247✓	0.198✓
	PE f=0.60	0.356✓	0.334✓	0.374✓	0.333✗	0.260✓	0.295✓	0.334✓	**0.261**✓	0.287✓	0.256✓	0.188✓
	PE f=0.80	0.356✓	0.337✓	0.366✓	0.339✗	0.264✓	0.294✓	0.339✓	0.264✓	0.286✓	0.251✓	0.185✓
	PE f=1.00 (M1)	0.358✓	0.339✓	0.362✓	0.339✗	0.258✓	0.294✓	0.342✓	0.269✓	0.290✓	0.248✓	0.187✓
	SIREN	**0.344**✓	**0.326**✓	**0.352**✓	**0.321**✓	**0.251**✓	**0.283**✓	**0.322**✓	0.251✗	**0.275**✓	**0.235**✓	**0.178**✓
$6.5M$	$\theta^\star_A$	0.346	0.323	0.357	0.324	0.244	0.282	0.322	0.259	0.283	0.240	0.176
	True dir.	A>B	A>B	A>B	A<B	A>B	A>B	A>B	A>B	A>B	A>B	A>B
	PE f=0.05	0.220✗	0.206✗	0.270✗	0.260✓	0.148✗	0.166✗	0.233✗	0.194✗	0.184✗	0.184✗	0.129✗
	PE f=0.10	0.268✗	0.263✗	0.305✗	0.274✓	0.185✗	0.215✗	0.281✗	0.190✗	0.223✗	0.229✓	0.148✗
	PE f=0.20	0.361✓	0.354✓	0.373✓	0.342✗	0.258✓	0.295✓	0.341✓	0.277✓	0.308✓	0.268✓	0.207✓
	PE f=0.40	0.363✓	0.344✓	0.380✓	0.358✗	0.254✓	0.291✓	0.351✓	0.277✓	0.297✓	0.252✓	0.202✓
	PE f=0.60	0.359✓	0.337✓	0.373✓	0.344✗	0.259✓	0.291✓	0.342✓	0.278✓	0.298✓	0.252✓	0.190✓
	PE f=0.80	0.360✓	0.338✓	0.364✓	0.334✗	0.260✓	0.294✓	0.341✓	0.280✓	0.292✓	0.249✓	0.184✓
	PE f=1.00 (M1)	0.358✓	0.339✓	0.362✓	0.333✗	0.251✓	0.294✓	0.342✓	0.269✓	0.291✓	0.248✓	0.187✓
	SIREN	**0.345**✓	**0.326**✓	**0.353**✓	**0.320**✓	**0.248**✓	**0.281**✓	**0.320**✓	**0.251**✓	**0.278**✓	**0.237**✓	**0.177**✓

C.3.1 Per-budget directional accuracy summaries

The per-budget summary tables below collapse each (f, B) cell to a single $(\mathrm{dir}, \mathrm{bias})$ pair averaged over the 11 models, making the f-dependent bias trade-off easier to read at a glance. Table 5 for Math with the DSPy tuner appears in the main text alongside the discussion of the PromptEval comparison; the corresponding tables for the other three (subject, tuner) cells are presented here. The U-shape pattern across f is consistent across all four cells: SIREN's mean signed error stays bounded by 0.21 pp at every budget, while every PromptEval fraction f reports a bias of at least 0.74 pp in absolute value.

Table 15: PromptEval [28] per-budget directional accuracy on **MMLU-Pro Math** for Random Search.

	500K		1.5M		3M		6.5M	
f	dir	bias	dir	bias	dir	bias	dir	bias
0.05	**11/11**	−1.67	9/11	−3.66	**10/11**	−3.59	9/11	−4.20
0.10	10/11	+1.87	7/11	+1.35	7/11	+1.61	6/11	+1.52
0.20	3/11	+6.36	3/11	+4.14	3/11	+3.94	3/11	+3.33
0.40	4/11	+5.72	5/11	+3.20	4/11	+3.54	4/11	+2.84
0.60	5/11	+5.43	5/11	+2.65	4/11	+3.06	4/11	+2.34
0.80	5/11	+5.29	5/11	+2.33	4/11	+2.58	4/11	+2.43
1.00	6/11	+5.19	5/11	+2.50	4/11	+2.49	6/11	+2.09
SIREN	**11/11**	**−0.09**	**10/11**	**+0.01**	**10/11**	**+0.01**	**11/11**	**+0.02**

Table 16: PromptEval [28] per-budget directional accuracy on **MMLU-Pro Law** for Random Search.

	500K		1.5M		3M		6.5M	
f	dir	bias	dir	bias	dir	bias	dir	bias
0.05	10/11	−11.58	**10/11**	−3.52	10/11	−0.74	9/11	−3.51
0.10	10/11	−9.01	7/11	+1.46	4/11	+2.45	4/11	+1.80
0.20	9/11	−4.30	2/11	+3.74	2/11	+3.54	3/11	+3.81
0.40	4/11	+1.68	3/11	+3.02	3/11	+2.96	4/11	+2.85
0.60	5/11	+1.56	3/11	+2.51	4/11	+2.55	5/11	+2.25
0.80	5/11	+1.65	2/11	+2.40	4/11	+2.34	5/11	+1.89
1.00	4/11	+1.50	2/11	+2.24	4/11	+2.10	5/11	+1.77
SIREN	**11/11**	**+0.05**	**10/11**	**+0.11**	**11/11**	**+0.07**	**10/11**	**+0.10**

Table 17: PromptEval [28] per-budget directional accuracy on **MMLU-Pro Law** for DSPy.

	500K		1.5M		3M		6.5M	
f	dir	bias	dir	bias	dir	bias	dir	bias
0.05	1/11	−12.72	1/11	−8.30	1/11	−8.60	1/11	−8.76
0.10	1/11	−9.14	2/11	−5.34	1/11	−5.56	2/11	−5.22
0.20	5/11	−2.31	**10/11**	+1.70	**10/11**	+1.57	10/11	+2.08
0.40	**10/11**	+1.49	**10/11**	+1.46	**10/11**	+1.59	10/11	+1.93
0.60	**10/11**	+0.96	**10/11**	+1.24	**10/11**	+1.08	10/11	+1.50
0.80	**10/11**	+0.74	**10/11**	+1.26	**10/11**	+1.12	10/11	+1.27
1.00	**10/11**	+0.77	**10/11**	+1.10	**10/11**	+1.13	10/11	+1.04
SIREN	9/11	**−0.21**	**10/11**	**−0.17**	**10/11**	**−0.20**	**11/11**	**−0.21**

D Additional empirical checks

D.1 Interval centering and the selection contribution

We first isolate the interval center by assigning all artifacts equal population quality, so hard and soft selection have the same target. We use $M = 500$, $R = 5$, scoring fraction 0.5, selector temperature $\tau = 0.1$, and 2,000 trials per configuration. As K increases from 1 to 50, M2's signed bias increases from -0.10 to $+4.29$ pp and its 95% coverage decreases from 95.7% to 49.3%. At $K = 50$, 50.7% of intervals lie entirely above the truth, with no downward misses; the reported standard error is 1.58 times the empirical standard deviation. This points to the shifted selected-winner center, rather than insufficient width, as the main problem in this setting. In the same $K = 50$ setting, SIREN-centered Wald and multiplier intervals using the full item-level contribution achieve 95.1% and 94.4% coverage, respectively, with nearly identical widths.

We next center every interval at the same SIREN estimate and vary only the variance calculation. Across a pre-specified quality-gap sweep, the evaluation-only Wald ablation omits the scoring-subset effect through the selector. Its coverage ranges from 91.7% to 95.6%, and its reported standard error reaches 0.88 of the empirical standard deviation. The full-contribution Wald and multiplier intervals remain close in coverage and width (Table 18). Thus the selection contribution matters for calibration even after correcting the center. For one pointwise coordinate, the full-contribution Wald and Gaussian multiplier intervals behave similarly; the multiplier construction also supports simultaneous bands and correlated system-budget comparisons.

Table 18: Coverage over the quality-gap sweep with the same SIREN point estimate. All intervals have nominal 95% coverage.

Variance calculation	Coverage range (%)
Evaluation-only Wald	91.7–95.6
Full-contribution Wald	94.7–95.8
Full-contribution multiplier	94.3–95.3

D.2 Estimation under each method's reporting target

We reanalyze all 176 Math/Law cells using a fresh-data target for each method's hard or soft reporting rule. We use the empirical item-row distribution and $10,000$ independent redraws with replacement, retaining fixed scoring-position masks. Selected artifacts are valued by their full-table mean scores. The common-reference tables instead use permutations and complementary splits without replacement, so the two references need not coincide.

All Math/Law tables use the audited point estimates from this reanalysis. For Math/random search, M1/M2 uses the full-matrix maximum, correcting the earlier first-column calculation; the four Qwen3-8B configurations use the same split implementation as the other cells. We compute directions from unrounded scores and count exact default ties as incorrect. These corrections reconcile M1/M2 with full-observation PromptEval.

Table 19 summarizes estimator accuracy under those references. M1/M2 and SIREN have similar mean absolute errors, 0.22 and 0.24 pp, and the same directional accuracy, while SIREN has the mean signed error closest to zero. The results do not show that M1/M2 estimate SIREN's target. They show that proximity to a common soft-rule reference and accuracy for a method's own target are different comparisons. Under the common reference, M1/M2, M3, M4, and SIREN have MAEs of 1.85, 1.67, 1.30, and 0.24 pp. SIREN has $167/176$ correct directions under that reference and $165/176$ under its fresh-data target. The two changed directions are GLM-4 on Math with DSPy at 1.5M and 6.5M, where the fresh-data reference lies just above the default and the permutation reference lies just below it.

Table 19: Fresh-data reanalysis of 176 cells under each method's own reporting target. Errors are in percentage points. Bold identifies the mean signed error closest to zero, smallest mean absolute error (MAE), and best directional accuracy, including ties.

Method	Mean signed error	MAE	Correct direction
M1/M2	$+0.22$	$\mathbf{0.22}$	$\mathbf{165/176}$
M3	-0.66	1.28	$126/176$
M4	-0.44	0.56	$148/176$
SIREN	$\mathbf{-0.05}$	0.24	$\mathbf{165/176}$

D.3 Comparison with nested cross-validation

We compare SIREN with five-fold nested cross-validation on MMLU-Pro Law using Qwen3-8B, fixed-library random search, and $B = 500\text{K}$. Each fold holds out 220 or 221 outer-test items. Within the outer-training portion, we pre-specify 100 development items and 200 disjoint SIREN reporting items. We rerun the complete search-and-selection procedure under four tuner seeds per fold and freeze the returned hard artifact, shortlist, and soft-rule weights before reading the outer-test scores. All 1,101 items serve as outer test once, giving 20 complete tuner reruns with the original development size.

Table 20: Nested cross-validation and SIREN with matched development size. Scores are percentages, and differences are percentage points.

Quantity	Result
Nested-CV hard-artifact outer-test mean	32.38
SIREN reporting mean	32.69
Matched outer-test soft-rule mean	32.03
SIREN minus matched soft-rule mean	$+0.66$ pp
Outer-test soft minus hard mean	-0.35 pp
Complete tuner reruns	20

The matched comparison for SIREN is its reporting estimate against the outer-test value of the same soft rule. The $+0.66$ pp difference is a reporting gap against a finite outer-test sample, rather than exact population error. The -0.35 pp soft-minus-hard difference concerns the deployment rules. The point estimates are close in this configuration, but do not establish equivalence. Full nested cross-validation evaluates the complete hard tune-then-deploy procedure by rerunning tuning inside each outer fold. SIREN provides conditional inference after one completed search. M3 and M4 reuse a frozen shortlist and therefore are holdout reporting baselines, rather than full nested cross-validation.

D.4 Fresh development samples and tuner reruns

We examine the relation between the conditional reporting target and the full tuning procedure on MMLU-Pro Law using Qwen3-8B, an 18-configuration fixed-library random-search tuner, and $B = 500\text{K}$. The primary design uses five pre-specified development samples and two tuner seeds per sample ($5D \times 2U$). Each run uses 100 development items and freezes its returned hard artifact and shortlist before SIREN uses a separate 200-item reporting pool. A further disjoint 200-item audit pool evaluates the soft rule and the returned hard artifact.

The fixed run is the pre-specified `D00_U00` run of this experiment. It was fixed before the repeated-tuning results were examined, is included in the repeated-tuning averages, and differs from the original open-ended-tuner run. As a post-result sensitivity check, we add two further pre-generated tuner seeds to each of the same five development samples ($5D \times 4U$), without adding development, reporting, or audit items. Both designs therefore have five independent development samples. The joint bootstrap resamples development samples, then tuner seeds within each sampled development set, and uses common item resamples across runs that share reporting and audit pools.

The primary fixed-run reporting gap is $34.96 - 34.73 = +0.23$ pp. The fixed-soft minus repeated-soft gap is $+0.33$ pp, and the repeated soft-minus-hard gap is -0.80 pp, giving a fixed-soft minus repeated-hard gap of -0.47 pp. Adding the reporting gap gives a SIREN-estimate minus repeated-

Table 21: Fixed-run reporting and repeated tuning. Scores are percentages. Differences and confidence intervals are in percentage points and are computed before rounding. The fixed-soft value is measured on the audit pool.

Quantity	Primary: $5D \times 2U$	Sensitivity: $5D \times 4U$
Fixed-run SIREN estimate	34.96	34.96
Fixed-run soft-rule audit value	34.73	34.73
Repeated-tuning soft-rule mean	34.40	34.31
Repeated-tuning hard-artifact mean	35.20	35.08
Fixed-soft minus repeated-hard gap	-0.47	-0.34
95% joint bootstrap CI	$[-2.98, 1.89]$	$[-2.59, 1.85]$

hard difference of -0.24 pp. With four seeds, the two audit-based components are $+0.42$ and -0.76 pp, giving -0.34 pp. The confidence intervals are too wide to establish equivalence within the pre-specified ± 1 pp margin. The experiment separates fixed-run reporting error, variation across tuner runs, and the difference between deployment rules; it does not identify the conditional soft-rule target with the original hard tune-then-deploy target.

D.5 Stochastic decoding and fractional scores

We freeze a pre-specified five-artifact Qwen3-8B Law shortlist and generate ten outputs for each artifact on each of 1,001 items at decoding temperature 1.0, giving 50,050 generations. Among item-artifact pairs, 24.4% produce more than one valid answer and 11.6% change between correct and incorrect across the ten decodes. We evaluate one sampled decode per item and the average of two sampled decodes, which gives scores in $\{0, 0.5, 1\}$. Each setting uses a reporting target computed from 10,000 independent redraws and 2,000 further trials for interval coverage, with 2,000 multiplier draws per trial.

Table 22: Bootstrap calibration under real stochastic execution. Coverage is measured over repeated trials for the reporting target induced by the frozen output table.

Setting	Bias (pp)	95% coverage (%)	Reported SE / empirical SD
One stochastic decode	-0.04	95.05	1.004
Mean of two stochastic decodes	-0.04	95.05	1.007

The multiplier intervals remain calibrated for both scoring rules. All executions belonging to one benchmark item remain together in its item-level contribution. The experiment checks the reporting target induced by the frozen real-output table; it does not rerun the tuner or establish calibration for every stochastic task.

D.6 Additional MMLU-Pro domains

We apply the original eleven-model random-search framework to MMLU-Pro Philosophy and Economics at $B \in \{500\text{K}, 1.5\text{M}\}$, with $K = 10$, $R = 10$, scoring fraction 0.5, and $\tau = 1$. Each model-budget cell has a 100,000-redraw Monte Carlo reporting reference. Table 23 uses the original common-reference convention. For M1/M2, M3, and M4, differences from this soft-rule reference can include both estimation error and differences between reporting rules. MAE averages absolute cellwise differences, rather than taking the absolute value of mean signed bias.

The reference direction is negative in $40/44$ cells. SIREN preserves all $4/4$ positive tuned gains and identifies $39/40$ regressions, so its directional accuracy is not explained by always predicting the majority direction. All 44 SIREN intervals contain their corresponding Monte Carlo references. We report this as interval containment across heterogeneous cells, rather than a repeated-sampling coverage estimate.

Table 23: Additional domains under the common reporting reference. Signed bias and MAE are in percentage points. Bold identifies the best direction, signed bias closest to zero, and smallest MAE within each subject or aggregate.

Subject	Method	Correct direction	Mean signed bias	MAE
Philosophy	M1/M2	$6/22$	$+2.93$	2.93
	M3	$13/22$	$+1.44$	2.86
	M4	$15/22$	$+1.31$	1.71
	SIREN	$\mathbf{21/22}$	$\mathbf{+0.09}$	$\mathbf{0.37}$
Economics	M1/M2	$13/22$	$+2.89$	2.89
	M3	$13/22$	$+2.59$	2.91
	M4	$16/22$	$+2.20$	2.32
	SIREN	$\mathbf{22/22}$	$\mathbf{+0.15}$	$\mathbf{0.28}$
All	M1/M2	$19/44$	$+2.91$	2.91
	M3	$26/44$	$+2.02$	2.88
	M4	$31/44$	$+1.76$	2.02
	SIREN	$\mathbf{43/44}$	$\mathbf{+0.12}$	$\mathbf{0.32}$

D.7 Code generation scored by execution

We evaluate HumanEval+ Mini through EvalPlus [23], built on the 164-task HumanEval benchmark [7]. Using Qwen3-8B, we treat 18 decoding temperatures from 0.1 to 1.8 as deployable artifacts and generate two programs per task-artifact pair, giving $5{,}904$ programs scored by executable unit tests. Each task-seeded realization uses 100 development tasks, 64 disjoint reporting tasks, development-only top-$K = 10$ retention, and a $10{,}000$-redraw reporting reference. The 164 realizations per budget overlap in tasks and are descriptive realizations, rather than independent models or datasets.

Table 24: Execution-scored code generation under the common reporting reference. Signed bias and MAE are in percentage points. Bold identifies the best direction, signed bias closest to zero, and smallest MAE at each budget.

Budget cap	Method	Correct direction	Mean signed bias	MAE
500K	M1/M2	$64/164$	$+2.60$	2.61
	M3	$73/164$	$+0.12$	4.25
	M4	$104/164$	-0.49	1.91
	SIREN	$\mathbf{115/164}$	$\mathbf{-0.08}$	$\mathbf{1.39}$
1.5M	M1/M2	$71/164$	$+2.69$	2.69
	M3	$80/164$	-0.20	4.18
	M4	$108/164$	-0.43	1.94
	SIREN	$\mathbf{113/164}$	$\mathbf{-0.10}$	$\mathbf{1.37}$

SIREN is closest to the specified reporting reference under both signed bias and MAE. The distinction matters for M3 at 500K, whose signed bias is $+0.12$ pp but whose MAE is 4.25 pp. As in the additional-domain comparison, the baseline differences from the common reference are not pure errors for each method's own target. The 1.5M row uses a token cap and the full library, which is exhausted after approximately 0.89 to 0.98 million actual tokens. The study extends the evidence to open-ended program outputs scored by execution, while retaining a fixed artifact library.