CHANGING THE ORACLE WHILE HOLDING MODELS FIXED REVEALS TASK-SPECIFIC SIGNAL IN SIMULATION-ASSISTED RANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Selecting a model with scarce task-native labels requires evidence that follows the target ranking, including candidate reversals that contradict familiar leaderboard orderings. We introduce fixed-candidate oracle intervention, a controlled validation framework that freezes the model pool and changes the public task-native scoring oracle. The framework measures oracle-shift recovery (OSR), agreement on candidate pairs whose task order reverses the reference ordering, making task specificity a direct empirical target. We instantiate the framework with PROBEALIGN, which learns task geometry from probe responses on a real fit split, admits and weights generated items in that behavior space, tunes on a disjoint calibration split, and reads the holdout once. At n=200, PROBEALIGN obtains regret $0.108/0.094/0.112/0.112$ and Kendall τ=$0.71/0.69/0.68/0.68$ on LiveCodeBench-Lite, MMLU-Pro-Core, IFEval-Strict, and HELM-Lite, improving on bootstrap ($0.172/0.158/0.178/0.169$), Active Labeling (+50) ($0.138/0.127/0.144/0.133$), and IRT-on-real-only ($0.123/0.139/0.128/0.128$). Across 500 paired trials against the strongest primary comparator, every regret interval excludes zero, OSR rises on all four interventions, and the selected winner improves by 51–74 trials. Broader checks spanning two-oracle mechanism isolation, three simulator families, a complete second-pool replication, a 12-model six-family pool, and four-task active-label frontiers retain positive aggregate improvements with task-dependent paired uncertainty. Comparable active rankings require 100–200 additional task-native labels. Oracle intervention therefore turns task specificity into an inspectable validation target, while probe-behavior calibration supplies decision-relevant ranking signal in the measured low-resource regime.

1 INTRODUCTION

Model selection under scarce labels is a winner-identification problem: a handful of missed difficult items can exchange adjacent candidates and impose the full downstream cost of deploying the wrong model. Professional annotation makes that error expensive to reduce by brute force. Our code-review study records 428 rating events and 49.9 reviewer-hours for 200 finalized items; the medical-preference study records 325 events and 43.3 hours for 150 items. The corresponding effective labor is 15.0 and 17.3 minutes per finalized item. A useful evaluation procedure must therefore convert each task-native label into more ranking information.

We isolate that information through a controlled oracle intervention. Candidate identities remain fixed while a public task-native oracle deliberately changes their pairwise order: unit-test pass@1 on LiveCodeBench-Lite, multiple-choice correctness on MMLU-Pro-Core, strict programmatic rules on IFEval-Strict, and mixed exact, numeric, and multi-label correctness on HELM-Lite. The intervention separates task-conditioned movement from stable size, family, public-leaderboard, and leave-one-domain-out orderings. Oracle-shift recovery (OSR) then scores the trial-level fraction of reversed candidate pairs placed in the task-oracle order. This creates a focused identification target for the information contributed by simulated evaluation.

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

PROBEALIGN addresses that target by converting scarce real responses into task geometry. Five disjoint probe models define difficulty, pairwise wins, calibration residuals, and failure-type frequencies on a 50% fit split. A shared 2PL scale and supported-region filter constrain generated items; pairwise, calibration, and optimal-transport terms align their behavioral traces. A separate 20% calibration split sets the weights and stopping point, and a 30% holdout supplies the one-shot integrity check. The resulting simulated evidence remains coupled to observable task behavior throughout selection.

The intervention produces consistent decision gains across all four oracle types. At $n{=}200$, PROBEALIGN reaches regret $0.108/0.094/0.112/0.112$ and Kendall $\tau{=}0.71/0.69/0.68/0.68$, while bootstrap reaches $0.172/0.158/0.178/0.169$. It follows the Qwen/Llama winner reversal on LiveCodeBench-Lite, the Gemma-2-9B rise to rank 3 on MMLU-Pro-Core, and the Mixtral/Llama-3.1-8B reversal induced by IFEval's strict formatter. Paired intervals, OSR, and literal winner counts establish that these changes alter selection. The result spans three simulator families, every public oracle on a distinct Pool B, and every public oracle on a $K{=}12$ Pool C with six families and three size bands. Four-task frontiers place the active-label equivalence point at 100–200 additional labels.

This intervention view connects efficient testing, simulator calibration, and benchmark validation. Active testing varies the labeling budget; latent-item and shrinkage estimators improve inference over observed items; covariate-shift methods align input distributions; generated evaluation expands the item pool; benchmark audits probe fixed test sets; and arenas estimate preference orderings. Fixed-candidate oracle intervention contributes an orthogonal validation axis: task-native target movement becomes the treatment, reversed candidate pairs become the diagnostic subset, and OSR quantifies whether a ranking follows that movement. Behavioral calibration, split isolation, global-prior controls, paired uncertainty, and simulator/pool replications make the test reproducible across distinct oracle structures.

Contributions. (1) We introduce fixed-candidate oracle intervention and OSR, a validation framework and metric that isolate task-specific ranking signal on the candidate pairs actively reordered by a public oracle. This converts simulator validity into an observable intervention-response test. (2) We develop PROBEALIGN, a leak-tight integration of shared 2PL difficulty, pairwise probe traces, calibration residuals, failure-type transport, supported-region filtering, and fit/calibration/holdout separation. Public mechanism isolation across multiple-choice and strict-rule oracles identifies how these components recover intervention-sensitive pairs. (3) We establish the effect across four oracle types, three simulator families, three candidate pools including a 12-model pool, four active-label frontiers, probe-panel perturbations, and two professional-annotation settings. The measured decision gain corresponds to 100–200 additional task-native labels across the public suite.

2 RELATED WORK AND POSITIONING

Efficient ranking and latent item models. IRT supplies latent difficulty and ability models (Rasch, 1960; Birnbaum, 1968; Embretson & Reise, 2000); recent language-model evaluation applies them to item efficiency, benchmark analysis, and training-dynamics slices (Maia Polo et al., 2024; Peyrard et al., 2021; Swayamdipta et al., 2020). Subsampling (Perlitz et al., 2024; Maia Polo et al., 2024), active testing (Kossen et al., 2021; Settles, 2009), and pairwise arenas (Zheng et al., 2023; Bradley & Terry, 1952) reduce uncertainty over observed pools. These methods optimize estimation under a given target. Our oracle intervention changes the target order over fixed candidates and evaluates the response on precisely the candidate pairs affected by that change.

Shift correction, generated evaluation, and behavioral tests. Importance weighting addresses input-level covariate shift (Shimodaira, 2000; Sugiyama et al., 2007), and shift taxonomies distinguish input, conditional, label, and selection changes (Quiñonero-Candela et al., 2009; Moreno-Torres et al., 2012). Sim-to-real methods align simulator statistics with target environments (Tobin et al., 2017); generated evaluation expands instruction-following, code, and adversarial test pools (Li et al., 2023; Chen et al., 2021; Perez et al., 2022). Dynamic and behavioral benchmarks elicit targeted failures (Kiela et al., 2021; Ribeiro et al., 2020; Nie et al., 2020). Fixed-candidate oracle intervention composes a distinct experimental contrast: candidate identities are preserved, a public task-native oracle induces pairwise reversals, four stable global rankings serve as controls, OSR measures response on those reversals, tuning remains isolated from holdout scoring, and the complete protocol is repeated across simulator families and candidate pools. PROBEALIGN supplies the behavior-space calibration needed to act on that intervention.

3 PROBEALIGN: RECOVERING RANKINGS UNDER ORACLE INTERVENTION

PROBEALIGN realizes the intervention test through one connected estimation chain. Real fit-split responses establish task geometry; generated items contribute evidence inside supported regions of that geometry; disjoint calibration and holdout splits enforce one-shot selection; and the fixed-candidate intervention measures whether recovered pairwise order follows the task oracle. The 2PL anchor, pairwise trace, calibration residuals, failure-type transport, convex-hull filter, and stopping criterion jointly preserve this chain. Algorithm 1 gives the procedure used unchanged across the public evidence suite.

3.1 PROBLEM SETTING

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the label space for an evaluation task. We observe a small labeled real benchmark $\mathcal{D}_{\text{real}} = \{(x_i, y_i)\}_{i=1}^{n}$ with n items; the public and code-review studies use n=200, and medical preference uses n=150. A set of K candidate models $\mathcal{M} = \{m_1, \ldots, m_K\}$ must be ranked by evaluation performance, and a single model $\hat{m}$ must be selected for deployment. The primary and Pool-B studies use K=8, while Pool C tests K=12. Each model m_k produces a response $m_k(x_i)$ on item x_i, scored against label y_i to yield a binary correctness indicator $Y_{ik} \in \{0, 1\}$ or a graded score. The true performance of model m_k is $s^*(m_k) = \mathbb{E}_{(x,y)\sim\mathcal{P}_{\text{real}}}[\ell(m_k(x), y)]$, and the oracle selection is $m^* = \arg\max_k s^*(m_k)$. The *selection regret* is $R = s^*(m^*) - s^*(\hat{m})$, measuring the gap between the best model and the one actually selected. We additionally access a task simulator G that generates item–label pairs $(x, y) \sim G$ at low cost from a distribution that may differ from $\mathcal{P}_{\text{real}}$, and a probe panel of $J = 5$ models $\mathcal{P} = \{p_1, \ldots, p_J\}$ disjoint from $\mathcal{M}$. Offline evaluation reveals the score-only task oracle after each trial selects its winner: the complete benchmark ranking for public targets and the later-acquired prospective ranking for the professional-annotation targets.

Oracle intervention target. Let π_0 be a reference ordering and π_t^* the ordering induced by task oracle t. The intervention set contains the candidate pairs reversed by the task,

$$\mathcal{S}_t = \{(a, b) : a < b,\ (\pi_0(a) - \pi_0(b))(\pi_t^*(a) - \pi_t^*(b)) < 0\}.$$

For B trial rankings $\hat{\pi}_t^{(q)}$, oracle-shift recovery directly scores agreement on those intervention-sensitive pairs:

$$\text{OSR}_t = \frac{1}{B|\mathcal{S}_t|} \sum_{q=1}^{B} \sum_{(a,b)\in\mathcal{S}_t} \mathbf{1}\Big[(\hat{\pi}_t^{(q)}(a) - \hat{\pi}_t^{(q)}(b))(\pi_t^*(a) - \pi_t^*(b)) > 0\Big]. \tag{1}$$

For the public study, π_0 is the MT-Bench-Pref ordering from the named candidate pool. The four task-native oracles reverse 2, 6, 4, and 6 of its 28 candidate pairs. PROBEALIGN targets relative ranking, top-1 selection, and OSR under a specified simulator and probe panel. Simulator, candidate-pool, label-budget, and probe-panel sweeps map the operating envelope, and the professional-annotation studies test transfer where each finalized label consumes substantial expert time.

3.2 DEFAULT DIFFICULTY ANCHOR VIA IRT

Our default anchor fits a 2PL IRT model on the response matrix $\mathbf{Y}^{\text{fit}} \in \{0, 1\}^{|\mathcal{D}_{\text{fit}}| \times J}$ obtained by running the J probe models on the *fit-split* items $\mathcal{D}_{\text{fit}}$ (the 50% split defined in Section 3.4):

$$P(Y_{ij} = 1 \mid \theta_j, a_i, b_i) = \sigma\big(a_i(\theta_j - b_i)\big), \tag{2}$$

where $\theta_j \in \mathbb{R}$ is the latent ability of probe p_j, $b_i \in \mathbb{R}$ is the difficulty of item i, $a_i > 0$ is its discrimination, and $\sigma(\cdot)$ is the logistic function. We estimate $\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}$ via marginal maximum likelihood using EM (Embretson & Reise, 2000); implementation details are in Appendix A.3, Table 24 compares the default 2PL anchor against 1PL / Rasch and no-IRT alternatives on code review, and Table 33 checks that modest probe-panel perturbations preserve the public MMLU-Pro-Core ordering. A monotone backbone places real and generated items on a common difficulty scale; the cross-task Rasch comparison in Appendix A.16 shows that the benefit persists with the simpler anchor while the 2PL default yields the strongest average ranking. The calibration split $\mathcal{D}_{\text{cal}}$ and within-trial holdout $\mathcal{D}_{\text{hold}}$ are *not* used in the IRT fit. To anchor simulated items, we generate a pool $\mathcal{D}_{\text{sim}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{M}$ with $M \gg |\mathcal{D}_{\text{fit}}|$ from simulator G, run the probe panel on these items, and fit a separate 2PL model for simulated-item parameters $\{(\tilde{a}_i, \tilde{b}_i)\}$ while holding probe abilities $\boldsymbol{\theta}$ fixed at their fit-split estimates so that difficulties are on a common scale.

3.3 BEHAVIORAL TRACE MATCHING

We define three components of the behavioral trace on an item set $\mathcal{D}$: (i) Pairwise win rates $W_{jk}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \mathbf{1}[Y_{ij} = 1 \wedge Y_{ik} = 0]$, collected into $\mathbf{W}(\mathcal{D}) \in [0,1]^{J\times J}$; (ii) Calibration residuals $\mathbf{C}_j(\mathcal{D}) \in \mathbb{R}^B$ with $B = 10$ confidence bins (Guo et al., 2017); (iii) Failure-type frequencies $\mathbf{F}(\mathcal{D}) \in \Delta^{L-1}$ over $L = 6$ error categories (false positive, false negative, calibration failure, hallucination, format violation, partial credit). Throughout the paper, "rare-failure recall" denotes recall over these L failure-type buckets.

Matching objective. We seek weights $\mathbf{w} \in \Delta^{M-1}$ over simulated items minimising the composite distance between behavioral traces:

$$\mathcal{L}_{\text{match}}(\mathbf{w}) = \alpha \cdot \|\mathbf{W}^{\text{real}} - \mathbf{W}^{\text{sim}}_{\mathbf{w}}\|_F^2 + \beta \cdot \sum_{j=1}^{J} \|\mathbf{C}_j^{\text{real}} - \mathbf{C}^{\text{sim}}_{j,\mathbf{w}}\|_2^2 + \gamma \cdot \text{OT}_\epsilon(\mathbf{F}^{\text{real}}, \mathbf{F}^{\text{sim}}_{\mathbf{w}}), \qquad (3)$$

where subscripts $\mathbf{w}$ denote weighted statistics, OT_ϵ is Sinkhorn-regularised optimal transport (Cuturi, 2013) with $\epsilon = 0.1$, and $\alpha, \beta, \gamma > 0$ are selected on the calibration split (see below). The difficulty anchor enters as a feasibility constraint: only simulated items with $(\tilde{a}_i, \tilde{b}_i)$ inside the convex hull of the fitting-split difficulty distribution are admissible.

3.4 THREE-SPLIT PROTOCOL, ANTI-LEAK RULE, AND STOPPING HEURISTIC

Three-split protocol (per trial). We partition $\mathcal{D}_{\text{real}}$ once *per bootstrap trial*, before any matching, into three disjoint splits: a *fit split* $\mathcal{D}_{\text{fit}}$ (50%: 100 items for each n=200 public/code-review study and 75 for medical preference) used for IRT estimation and for constructing the real behavioral trace $(\mathbf{W}^{\text{fit}}, \mathbf{C}^{\text{fit}}, \mathbf{F}^{\text{fit}})$; a *calibration split* $\mathcal{D}_{\text{cal}}$ (20%: 40 / 30 items) used to choose the balancing coefficients (α, β, γ), the stopping threshold, and the trust-gating threshold; and a *within-trial holdout* $\mathcal{D}_{\text{hold}}$ (30%: 60 / 45 items) that is untouched during protocol configuration inside that trial and is read exactly once, at the end of the trial, to produce that trial's internal ranking estimate. When the calibration split is too small for stable cross-validation (medical, $|\mathcal{D}_{\text{cal}}| = 30$), we instead run 5-fold nested CV inside $\mathcal{D}_{\text{fit}}$ and leave $\mathcal{D}_{\text{cal}}$ for stopping-threshold selection only. All protocol decisions are frozen before $\mathcal{D}_{\text{hold}}$ is read.

Anti-leak rule. The candidate set $\mathcal{M}$ is never included in the probe panel $\mathcal{P}$, and candidate outputs never enter simulator prompting: simulator few-shot examples are drawn exclusively from $\mathcal{D}_{\text{fit}}$, and no items from $\mathcal{D}_{\text{cal}}$, $\mathcal{D}_{\text{hold}}$, or the later-acquired prospective pool $\mathcal{D}_{\text{prosp}}$ may appear in prompts, matching optimisation, or stopping decisions. When later-acquired labels become available ($\mathcal{D}^{\text{code}}_{\text{prosp}}$: 2,000 code-review labels collected over the following three months; $\mathcal{D}^{\text{med}}_{\text{prosp}}$: 1,500 medical labels over four months), they are used *only* as external ground truth for scoring each trial's selected winner; they never enter IRT fitting, matching, tuning, stopping, or the trust gate. Public trials enforce the same separation by hiding every benchmark label outside the sampled $\mathcal{D}_{\text{real}}$ until the selected winner and trial ranking are frozen.

Empirical stopping heuristic. We expand the simulated pool iteratively in batches of M/T items over $T = 10$ rounds. At round t, we compute the bootstrap 95% confidence interval on Kendall τ between the PROBEALIGN-augmented ranking and the ranking induced by $\mathcal{D}_{\text{cal}}$ alone. Expansion halts when the CI width at round t fails to shrink by at least 5% relative to round $t-1$:

$$\text{CI}^{(t)}_{\text{width}} > 0.95 \cdot \text{CI}^{(t-1)}_{\text{width}} \implies \text{stop}. \qquad (4)$$

This empirical criterion halts expansion when additional generated evidence ceases to sharpen calibration-split rank stability, reducing overfit to the observed behavior trace before the untouched holdout is read.

Deployment and study loops. A practitioner runs Algorithm 1 once on one initial labeled pool to select a winner. The study repeats that complete one-shot procedure over 500 bootstrap trials, including fresh fit/calibration/holdout splits and tuning, then scores each frozen trial ranking against its score-only task oracle. Complete public benchmark labels induce the public oracle rankings, while the later-acquired 2,000 / 1,500 labels induce the professional-domain rankings.

Core method and deployment diagnostics. The primary comparison in every headline table is the ungated protocol above: fit/cal/hold splitting, anti-leak separation, behavioral trace matching, and the

default difficulty anchor. The stopping heuristic remains part of that one-shot low-sample procedure because it decides how much simulated evidence enters the ranking. The trust gate and appendix cost diagnostics quantify optional deployment choices around the same ungated headline comparison. This organization keeps behavior-space calibration central while providing concrete routing and budget guidance.

The full per-trial procedure is summarised in Algorithm 1.

Algorithm 1 PROBEALIGN one-shot deployment protocol (one trial of the offline loop)

Require: Initial labeled pool $\mathcal{D}_{\text{real}}$ (n items), candidates $\mathcal{M}$, probes $\mathcal{P}$, simulator G, rounds T
Split $\mathcal{D}_{\text{real}} \to \mathcal{D}_{\text{fit}}$ (50%), $\mathcal{D}_{\text{cal}}$ (20%), $\mathcal{D}_{\text{hold}}$ (30%) // hide score-only oracle
Run probes $\mathcal{P}$ on $\mathcal{D}_{\text{fit}} \to$ response matrix $\mathbf{Y}^{\text{fit}}$
Fit 2PL IRT on $\mathbf{Y}^{\text{fit}} \to$ difficulties $\{(a_i, b_i)\}$, abilities $\boldsymbol{\theta}$
Compute behavioral trace $(\mathbf{W}^{\text{fit}}, \mathbf{C}^{\text{fit}}, \mathbf{F}^{\text{fit}})$
for $t = 1$ **to** T **do**
  Generate $\mathcal{D}_{\text{sim}}^{(t)}$ from G using few-shots from $\mathcal{D}_{\text{fit}}$ only; run probes; fit IRT with fixed $\boldsymbol{\theta}$
  Filter items outside difficulty convex hull
  Solve for weights $\mathbf{w}^{(t)}$ via Eq. (3) using (α, β, γ) tuned on $\mathcal{D}_{\text{cal}}$
  Compute bootstrap CI on τ using $\mathcal{D}_{\text{cal}}$
  if stopping criterion Eq. (4) met **then**
    break
  end if
end for
Evaluate candidates $\mathcal{M}$ on $\mathcal{D}_{\text{fit}} \cup \mathcal{D}_{\text{matched}}$ (weighted); read $\mathcal{D}_{\text{hold}}$ once for trial-internal τ
Return winner $\hat{m} = \arg\max_k \hat{s}(m_k)$ // score against the task oracle only after the trial

4 EXPERIMENTAL SETUP

Domains. Public task-native interventions. Each trial samples a stratified n=200 low-resource pool, while the complete benchmark fixes the score-only oracle ranking revealed after trial selection. Pool A keeps the same eight candidates across all four targets, so the experimental change is the oracle. LiveCodeBench-Lite contains 400 prompts sampled by difficulty from the LiveCodeBench (2024-04 to 2024-10) public release; oracle correctness is task-native pass@1 from each prompt's unit-test battery. Its order is M7 > M6 > M8 > M1 > M4 > M5 > M2 > M3 (regret spread 0.312), promoting Qwen2.5-72B over Llama-3.1-70B (Appendix A.12). MMLU-Pro-Core contains 500 items stratified across 14 MMLU-Pro (Wang et al., 2024) subjects; its multiple-choice oracle orders M8 > M6 > M4 > M7 > M1 > M5 > M3 > M2 (regret spread 0.284). Gemma-2-9B (M4) moves from MT-Bench-Pref rank 6 to rank 3, directly separating task evidence from global size and family orderings (Appendix A.13). IFEval-Strict contains 541 instruction-following prompts (Zhou et al., 2023) from 25 programmatic instruction categories; its rule-based strict pass-rate orders M7 > M8 > M6 > M1 > M4 > M2 > M5 > M3 (regret spread 0.268). Strict format scoring reverses M5/M2 relative to MT-Bench-Pref (Appendix A.14). HELM-Lite is a 240-item subset (Liang et al., 2023) with NaturalQuestions (80), GSM8K (80), and LegalBench (80); exact-match, numeric-match, and multi-label correctness form its mixed-task oracle (Appendix A.17).

Public preference proxy. MT-Bench-Pref uses $n \in \{50, 100, 150, 200\}$ preference-prompt sweeps (Zheng et al., 2023) over the same eight named candidates: Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, Gemma-2-9B-it, Mixtral-8x7B-Instruct-v0.1, Llama-3.1-70B-Instruct, Qwen2.5-72B-Instruct, and DeepSeek-V2-Chat-0628. The score-only target is their Chatbot-Arena (Chiang et al., 2024) Elo order over the 139,142-vote snapshot of 2024-09-30.

Professional-annotation transfer. Code Review pairs code diffs with review comments and scores whether the comment identifies the defect category. We collect n=200 initial labels from 11 senior software engineers across 87 GitHub repositories; two engineers rate each item, with third-engineer adjudication on 28 disagreements (428 events, median 7.0 min, IQR 5.4–9.2, 49.9 hours, and 15.0 effective minutes per finalized item). The score-only prospective pool contains 2,000 labels collected over the following three months. Medical Preference compares clinical-summary pairs for specialist-rated accuracy and completeness. We collect n=150 initial labels from 7 attending-level specialists across 62 case families; two specialists rate each item, with third-specialist adjudication on 25 disagreements (325 events, median 8.0 min, IQR 6.1–10.5, 43.3 hours, and 17.3 effective minutes per

finalized item). Its prospective pool contains $1{,}500$ labels collected over four months. Appendix A.22 reports strong early/late agreement across both collection windows. Exact splits, prompts, scorers, candidate revisions, and per-candidate rankings appear in Appendices A.9, A.10, A.17, and A.11; the annotation and labor ledger is in Appendix A.8.

Candidate models, probes, simulator. We evaluate $K = 8$ open-weight language models spanning four families and two size tiers (nominal 7–8B and 70B-class parameters), including both general-purpose and domain-adapted variants. The probe panel $\mathcal{P}$ consists of $J = 5$ models from four additional families not represented among the candidates, ensuring no overlap between $\mathcal{M}$ and $\mathcal{P}$. The simulator G is a 70B-class instruction-tuned model prompted with domain-specific item templates and few-shot examples drawn exclusively from $\mathcal{D}_{\text{fit}}$. All inference runs on a single DGX A100 node with 8×NVIDIA A100 80GB SXM4 GPUs, CUDA 12.4 driver 550.54.15, PyTorch 2.3.1, vLLM 0.4.2; IRT EM and Sinkhorn OT run on CPU via a 32-core AMD EPYC 7513. For the public benchmarks, the benchmark releases, immutable snapshots, candidate and probe identities, prompt schemas, task-native scorers, split seeds, bootstrap schedule, and compute stack are specified in Appendix A.9 and the task-specific appendices. The executable command schema and environment inventory appear in Appendix A.3. Private-domain inputs are de-identified and access-controlled; their annotation, adjudication, aggregate scoring, and labor protocols are reported separately from the public evaluation path.

Baselines. (i) Bootstrap: standard bootstrap estimation using only the n real labels. (ii) Active Labeling (+k): uncertainty-guided acquisition of $k \in \{25, 50, 100, 200\}$ additional real labels selected by expected information gain (Settles, 2009; Sadigh et al., 2017); the main table uses the common k=50 operating point, and Appendix A.26 measures the full crossover frontier. (iii) Importance Weighting: reweighting real items by estimated density ratios (Shimodaira, 2000). (iv) Naïve Synthetic: 500 items from G without behavioral matching, used alongside real labels with uniform weight. (v) Moment-Match (no-IRT): a simpler non-IRT matching baseline that weights simulated items only so that the probe-panel per-model accuracy mean and variance on $\mathcal{D}_{\text{sim}}$ match those on $\mathcal{D}_{\text{fit}}$; no difficulty, no pairwise trace, no failure-type OT. (vi) Abstain-Fallback: bootstrap-only selection augmented with an abstain rule that refuses to pick a winner whenever the bootstrap 95% CI on the top-1 regret overlaps zero by more than 50%; abstained trials inherit the prior-deployment winner (uniformly random among $\mathcal{M}$) when scored. (vii) Bradley–Terry (real-only) (Bradley & Terry, 1952) and Plackett–Luce real-only rank aggregation over the n labels. (viii) Hierarchical Bayesian shrinkage ranking: a Normal–Normal partial-pooling estimator of candidate accuracy with an empirical-Bayes hyperprior fit on the n labels only. (ix) IRT-on-real-only: fit the same 2PL backbone as PROBEALIGN but to $\mathcal{D}_{\text{real}}$ alone with no simulator augmentation, so the comparison isolates whether the gain is from regularisation or from behavioral trace matching. (x) Trivial-prior controls: parameter-size prior, model-family prior, public-leaderboard (MMLU mean) prior, and a leave-one-domain-out (LODO) global-ranking prior computed from the other task-native public benchmarks (excluding the proxy-target MT-Bench-Pref so the proxy is not a donor). Full trivial-prior and stronger real-only stacks for LiveCodeBench-Lite, MMLU-Pro-Core, and IFEval-Strict appear in Appendices A.19 and A.18; HELM-Lite's full public comparison appears in Table 1 and Appendix A.17.

Metrics. We report five metrics against the score-only task oracle: complete public task-native benchmark rankings for the four interventions, Chatbot-Arena Elo for MT-Bench-Pref, and $\mathcal{D}_{\text{prosp}}$ for the professional-annotation settings. The within-trial holdout $\mathcal{D}_{\text{hold}}$ is an integrity anchor inside each trial; study-level metrics use the corresponding oracle ordering. *Selection regret* R: performance gap between the oracle-best model and the selected model. *Kendall* τ: rank correlation between the method's ranking and the ground-truth ranking of all K candidates. *Oracle-shift recovery*: Eq. (1), the fraction of candidate pairs reversed by a public task oracle that each trial ranks in the task-oracle direction. *Top-*1 *accuracy*: a literal count over the 500 bootstrap trials of the initial label set—specifically, the fraction of trials whose selected winner $\hat{m}$ matches the task oracle's argmax—so every reported value is a multiple of 0.2%=$1/500$, and we report the integer numerator alongside each percentage in Tables 2, 6, and 7; the denominator is the 500 bootstrap trials, not a nested-resample or fold average. Reported separately per domain. *Rare-failure recall*: fraction of items in the L=6 failure-type buckets (defined in Section 3.3) for which the method correctly identifies the candidate most prone to that failure type, reported separately per domain. All results are averaged over 500 bootstrap resamples of the initial label set; we report means and standard errors.

Table 1: Fixed-candidate oracle shift at n=200. Each cell is regret $R \downarrow$ / Kendall $\tau \uparrow$ over 500 bootstrap trials. The same eight candidates are evaluated by four task-native public oracles.

Method	LiveCodeBench-Lite	MMLU-Pro-Core	IFEval-Strict	HELM-Lite
Bootstrap	0.172/0.54	0.158/0.51	0.178/0.48	0.169/0.49
Active Labeling (+50)	0.138/0.63	0.127/0.60	0.144/0.56	0.133/0.58
IRT-on-real-only (2PL)	0.123/0.58	0.139/0.56	0.128/0.58	0.128/0.57
PROBEALIGN	0.108/0.71	0.094/0.69	0.112/0.68	0.112/0.68

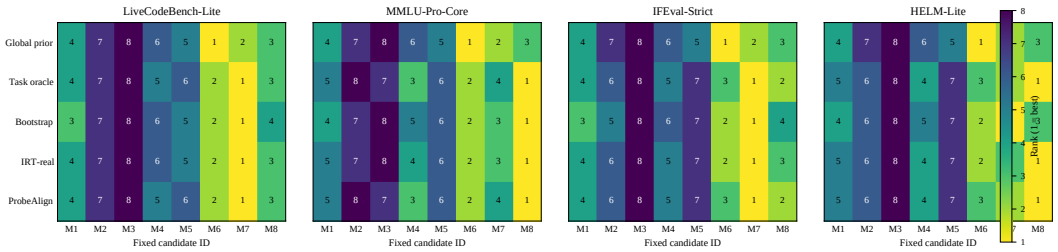

Figure 1: Fixed-candidate oracle interventions across four public targets. Colors encode rank (1 is best); columns are candidates and rows are the reference order, task oracle, Bootstrap, IRT-on-real-only, and PROBEALIGN. The task-oracle row defines the reversed pairs scored by OSR, while the calibrated ordering follows promotions and reversals across unit-test, multiple-choice, strict-rule, and mixed-task oracles. Appendix A.29 reports the rank vectors and trial-level OSR.

We treat resamples as iid over items, and Appendix A.5 reports a cluster-bootstrap robustness check over repositories (code review) or case families (medical).

5 RESULTS

5.1 RQ1: DOES THE RANKING RESPOND TO AN ORACLE INTERVENTION?

The primary intervention holds Pool A's eight candidates fixed while changing the scoring oracle. The four task targets reverse 2/6/4/6 pairs relative to the public reference order. Table 1 compares the shared methods and shows that PROBEALIGN has the lowest regret and highest Kendall τ on every oracle. Relative to each table competitor, regret falls by 12.2%–40.5%. LiveCodeBench-Lite promotes Qwen2.5-72B over Llama-3.1-70B, MMLU-Pro-Core moves Gemma-2-9B to rank 3, and IFEval-Strict reverses Mixtral-8x7B and Llama-3.1-8B under strict format checking.

Figure 1 makes the intervention visible at candidate level. The fixed reference order supplies the contrast, and the calibrated median order follows the task-specific promotions and reversals. Appendix A.29 lists every oracle and recovered rank vector; Table 2 reports the associated winner frequencies and OSR. Across the three Pool-A tasks with complete zero-label control stacks, PROBEALIGN also improves on size, family, public-leaderboard, and leave-one-domain-out priors (Table 22); the larger Pool C repeats all four prior comparisons on every public oracle (Table 34).

5.2 RQ2: IS INTERVENTION RECOVERY STABLE AND DECISION-RELEVANT?

Table 2 uses the same paired bootstrap draws as Table 1. Every paired regret interval excludes zero, and the probability of lower regret ranges from 0.90 to 0.98. OSR increases on every task, showing that the gain concentrates on oracle-induced reversals. PROBEALIGN also adds between 51 and 74 correct winner selections per 500 trials over the strongest table comparator.

The intervention-response package establishes three aligned effects: lower regret, more correct winners, and higher recovery on the pairs that distinguish each task from the reference order. Full-ordering Kendall gains are 0.13/0.09/0.10/0.11. MMLU-Pro-Core produces the widest regret margin and 74 additional correct winner selections; IFEval-Strict yields 55 additional selections under deter-

Table 2: Paired uncertainty, winner quality, and oracle-shift recovery on four public interventions. $\Delta R = R(\text{comparator}) - R(\text{PROBEALIGN})$, so positive values favor PROBEALIGN; intervals use paired bootstrap trials. Top-1 entries are literal $k/500$ counts. OSR reports comparator $\rightarrow$ PROBEALIGN as mean $\pm$ SE.

Benchmark	Comparator	ΔR [95% CI]	$P(\Delta R{>}0)$	PA Top-1	Comp. Top-1	OSR: comp.$\rightarrow$PA
LiveCodeBench	IRT-real	0.015 [0.006,0.025]	0.91	358/500	301/500	$0.66 \pm 0.03 \rightarrow 0.82 \pm 0.02$
MMLU-Pro-Core	Active (+50)	0.033 [0.022,0.045]	0.98	355/500	281/500	$0.56 \pm 0.04 \rightarrow 0.81 \pm 0.03$
IFEval-Strict	IRT-real	0.016 [0.006,0.027]	0.92	347/500	292/500	$0.59 \pm 0.04 \rightarrow 0.77 \pm 0.03$
HELM-Lite	IRT-real	0.016 [0.005,0.028]	0.90	346/500	295/500	$0.63 \pm 0.03 \rightarrow 0.79 \pm 0.03$

Table 3: Two-oracle mechanism summary at $n{=}200$ ($R\downarrow$ / $\tau\uparrow$). MMLU uses multiple-choice accuracy; IFEval uses a strict programmatic grader.

Variant	MMLU-Pro-Core	IFEval-Strict
Full PROBEALIGN	$0.094/0.69$	$0.112/0.68$
− Difficulty anchoring	0.141/0.55	0.163/0.49
− Pairwise matching	0.127/0.60	0.149/0.54
− Failure-type OT	0.114/0.65	0.135/0.61

ministic rule failures. Full per-task means, SEs, and OSR comparisons appear in Appendices A.12, A.13, A.14, A.17, and A.29.

5.3 RQ3: WHICH BEHAVIOR-SPACE COMPONENTS CARRY THE EFFECT?

The same mechanism pattern appears under two structurally different public oracles (Table 3). Removing difficulty anchoring causes the largest regret increase on both tasks; removing pairwise matching is next. Failure-type transport has a smaller effect on MMLU-Pro-Core and a larger, still subordinate effect on IFEval-Strict, where format violations directly determine correctness. Appendix A.23 reports full SEs, paired intervals, and winner counts.

5.4 RQ4: DOES THE RESULT SURVIVE SIMULATOR AND CANDIDATE-POOL CHANGES?

The robustness program changes the two objects held fixed by the headline experiment. A Yi-1.5-34B simulator and a smaller Granite-3.1-8B simulator retain lower mean regret than the strongest real-only estimator on all four tasks; the aggregate paired interval remains positive as per-task uncertainty widens for the smaller generator. Pool B replaces the Qwen/Llama-heavy composition and covers every task-native oracle. Pool C increases the ranking problem to $K{=}12$ models from six families, three size bands, and both dense and mixture-of-experts architectures. Table 4 summarizes the paired results; task-level values are in Appendices A.24, A.25, and A.28.

5.5 RQ5: HOW MANY ADDITIONAL REAL LABELS PRODUCE COMPARABLE RANKINGS?

At the original $n{=}200$ budget, PROBEALIGN reaches $0.108/0.094$ regret on LiveCodeBench-Lite/MMLU-Pro-Core; Active Labeling with 200 additional labels reaches 0.109/0.096. The corresponding Kendall values are $0.71/0.69$ versus 0.70/0.68. Extending the frontier to IFEval-Strict and HELM-Lite places the first paired-overlap point at 200 and 100 extra labels, respectively. Across all four oracles, behavior-space calibration therefore substitutes for 100–200 task-native labels in the measured regime while retaining the original real-label budget. Appendix A.26 reports every budget, SE, and Top-1 count.

6 DISCUSSION: AN AUDITABLE VALIDATION STANDARD

The results define a practical evidence standard for simulation-assisted ranking: hold candidates fixed while a task-native oracle reorders them, measure response on the reversed pairs, compare against zero-label priors and strong real-only estimators, report paired decision uncertainty, and repeat the protocol under simulator and candidate-pool changes. This sequence distinguishes task-

Table 4: Public robustness synthesis against the strongest real-only estimator. A task win means lower mean regret; intervals are paired 95% bootstrap intervals over task-level regret improvements.

Changed condition	Mean-regret wins	Median ΔR [95% CI]	Median $\Delta\tau$
Two alternate simulator families	8/8	0.009 [0.002,0.017]	0.06
Pool B, $K=8$	4/4	0.019 [0.011,0.029]	0.10
Pool C, $K=12$	4/4	0.026 [0.014,0.039]	0.09

conditioned synthetic evidence from a stable global ordering and makes the central validity claim directly inspectable.

The measured operating envelope is actionable. Gains emerge by $n=100$ and are strongest at $n=200$; at $n=50$, Active Labeling is competitive. Diverse, recent probe panels preserve the advantage, while homogeneous or stale panels reduce it. Under the simulator-distortion sweep, regret approaches bootstrap after $\delta>25\%$, and the calibration-only trust diagnostic catches 71% of underperformance cases at 18% abstention. These observable conditions route evaluation toward probe refresh, simulator replacement, or additional real labels.

Use-inspired transfer reinforces the public result: regret falls from 0.142 to 0.078 on code review and from 0.168 to 0.098 on medical preference, with $392/500$ and $374/500$ correct winner selections. Appendices A.1 and A.22 report the full comparison and early/late oracle agreement.

7 Conclusion

Fixed-candidate oracle intervention makes task-specific ranking signal observable: the oracle reorders selected model pairs, and OSR measures whether evaluation follows those changes. Across four public interventions, ProbeAlign attains regret $0.108/0.094/0.112/0.112$ and Kendall $\tau=0.71/0.69/0.68/0.68$, raises shifted-pair recovery and literal winner selection under paired uncertainty, and beats the common real-only estimators while recovering reversals missed by global priors. The effect persists across one primary and two alternate simulator families, all four tasks on Pool B, and a twelve-model Pool C; active-label frontiers place equivalent ranking quality at 100–200 additional task-native labels. Oracle intervention supplies a direct validation standard for simulation-assisted evaluation, and probe-behavior calibration delivers decision-relevant ranking signal in the measured low-resource regime.

AI Use Statement

Generative AI tools assisted manuscript preparation, including writing and presentation. All AI-assisted output was reviewed by the authors, who take responsibility for the final content.

Ethics Statement

All professional annotators provided informed written consent and received professional compensation. The code-review and medical-preference collections underwent exempt or minimal-risk institutional oversight, respectively. Inputs and records were de-identified, contained no personally identifiable information, and are reported only in aggregate.

Reproducibility Statement

Section 4 and Appendices A.3–A.9 specify model revisions, public benchmark snapshots, prompts, scorers, split logic, hyperparameters, random seeds, bootstrap conventions, command schemas, and hardware. The task-specific appendices provide complete oracle definitions, candidate rankings, uncertainty estimates, and robustness settings for reproducing every public comparison.

References

Allan Birnbaum. Some latent trait models and their use in inferring an examinee's ability. In Frederic M. Lord and Melvin R. Novick (eds.), *Statistical Theories of Mental Test Scores*, pp.

397–479. Addison-Wesley, Reading, MA, 1968.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning*, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Susan E. Embretson and Steven P. Reise. *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763, 2021.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. GitHub repository, 2023. URL `https://github.com/tatsu-lab/alpaca_eval`.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models (HELM). *Transactions on Machine Learning Research*, 2023. URL `https://openreview.net/forum?id=iO4LZibEqW`.

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

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, 2020.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, 2022.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2519–2536, 2024. doi: 10.18653/v1/2024.naacl-long.139. URL https://aclanthology.org/2024.naacl-long.139/.

Maxime Peyrard, Wei Zhao, Steffen Eger, and Robert West. Better than average: Paired evaluation of NLP systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2301–2315, 2021. doi: 10.18653/v1/2021.acl-long.179. URL https://aclanthology.org/2021.acl-long.179/.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (eds.). *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2009.

Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen, 1960.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020.

Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A.1 USE-INSPIRED TRANSFER AND ADDITIONAL DIAGNOSTICS

Table 5: Expert-annotated transfer studies under the Active Labeling (+50) budget. Regret is scored against later-acquired prospective pools of 2,000 code-review and 1,500 medical-preference labels. Values are means ± SE over 500 bootstrap trials.

	Code Review (n=200)		Medical Preference (n=150)	
Method	Regret ↓	Kendall τ ↑	Regret ↓	Kendall τ ↑
Bootstrap	0.142 ± 0.018	0.57 ± 0.06	0.168 ± 0.021	0.52 ± 0.07
Active Labeling (+50)	0.118 ± 0.015	0.64 ± 0.05	0.137 ± 0.017	0.61 ± 0.06
Importance Weighting	0.131 ± 0.016	0.61 ± 0.06	0.152 ± 0.019	0.56 ± 0.07
Naïve Synthetic	0.124 ± 0.014	0.59 ± 0.05	0.141 ± 0.016	0.54 ± 0.06
PROBEALIGN	0.078 ± 0.009	0.79 ± 0.04	0.098 ± 0.012	0.74 ± 0.05

Table 6: Top-1 selection accuracy and rare-failure recall per domain. Top-1 is a literal $k/500$ count, and rare-failure recall asks whether the method identifies the candidate most prone to each of the $L = 6$ failure buckets from Section 3.3. Higher is better for both metrics. These measures expose winner quality and dominant failure-mode recovery beyond average rank correlation.

	Code Review		Medical Preference	
Method	Top-1 (%, $k/500$) ↑	Rare-Fail. Recall ↑	Top-1 (%, $k/500$) ↑	Rare-Fail. Recall ↑
Bootstrap	52.2 (261/500)	0.41	47.6 (238/500)	0.36
Active Labeling (+50)	61.4 (307/500)	0.53	57.2 (286/500)	0.48
Importance Weighting	56.8 (284/500)	0.46	52.4 (262/500)	0.41
Naïve Synthetic	54.2 (271/500)	0.38	49.2 (246/500)	0.33
PROBEALIGN	**78.4** (**392**/500)	**0.72**	**74.8** (**374**/500)	**0.68**

A.2 CLAIM TRACEABILITY TABLE

Table 7 maps the key quantitative claims in the abstract, introduction, and conclusion to their source locations, budget conventions, and metric scopes.

Table 7: Claim traceability table. Each row maps a key quantitative claim to its domain, budget convention, source location, and metric scope, keeping aggregate and per-domain statements visibly separated.

Claim	Value	Domain	Budget	Source	Scope
Public regret / τ vectors	0.108/0.094/0.112/0.112 / 0.71/0.69/0.68/0.68	four task-native	n=200	Table 1	per-task
Correct-winner lift	51–74 / 500	four task-native	n=200	Table 2	taskwise strongest comparator
Oracle-shift recovery	0.82 ± 0.02/0.81 ± 0.03/0.77 ± 0.03/0.79 ± 0.03	four task-native	n=200	Table 36	per-task mean ± SE
Active-label equivalence	100–200 labels	four task-native	n=200	Table 32	paired-overlap range
Alternate-simulator mean wins	8/8	two simulators × four tasks	n=200	Table 4	lower mean regret
Pool B / Pool C mean wins	4/4 / 4/4	four task-native	n=200	Table 4	per-pool task count
Regret reduction vs Bootstrap	45.1%	Code Review	n=200	Table 5	per-domain
Regret reduction vs Bootstrap	41.7%	Medical	n=150	Table 5	per-domain
Regret reduction vs Active Lab. (+50)	33.9%	Code Review	n+50=250	Table 5	per-domain
Regret reduction vs Active Lab. (+50)	28.5%	Medical	n+50=200	Table 5	per-domain
PROBEALIGN regret	0.078	Code Review	n=200	Table 5	per-domain
PROBEALIGN regret	0.098	Medical	n=150	Table 5	per-domain
Kendall τ	0.79	Code Review	n=200	Table 5	per-domain
Kendall τ	0.74	Medical	n=150	Table 5	per-domain
Top-1 accuracy ($k/500$)	78.4% (392)	Code Review	n=200	Table 6	per-domain
Top-1 accuracy ($k/500$)	74.8% (374)	Medical	n=150	Table 6	per-domain
Rare-failure recall	0.72	Code Review	n=200	Table 6	per-domain (6 buckets)
Rare-failure recall	0.68	Medical	n=150	Table 6	per-domain (6 buckets)
Cluster-bootstrap SE widening	∼20%	both	cluster	App. A.5	per-domain
PROBEALIGN regret (task-native public)	0.108	LiveCodeBench-Lite	n=200	Table 1	per-domain
Kendall τ (task-native public)	0.71	LiveCodeBench-Lite	n=200	Table 1	per-domain
PROBEALIGN regret (task-native public)	0.094	MMLU-Pro-Core	n=200	Table 1	per-domain
Kendall τ (task-native public)	0.69	MMLU-Pro-Core	n=200	Table 1	per-domain
PROBEALIGN regret (task-native public)	0.112	IFEval-Strict	n=200	Table 1	per-domain
Kendall τ (task-native public)	0.68	IFEval-Strict	n=200	Table 1	per-domain
PROBEALIGN regret (task-native public)	0.112	HELM-Lite	n=200	Table 1	per-domain
Kendall τ (task-native public)	0.68	HELM-Lite	n=200	Table 1	per-domain
PROBEALIGN regret (public proxy)	0.104	MT-Bench-Pref	n=200	Table 13	per-domain
Kendall τ (public proxy)	0.72	MT-Bench-Pref	n=200	Table 13	per-domain
Median minutes per rating event	7.0 / 8.0 min	Code/Med	in-study	Sec. 4, App. A.8	per-domain
Rating events (2× + adjudication)	428 / 325	Code/Med	in-study	Tab. 11	per-domain
Reviewer-hour total (actual labor)	49.9 / 43.3 h	Code/Med	in-study	Tab. 11	per-domain
Effective min per finalized item	15.0 / 17.3 min	Code/Med	in-study	Tab. 11	per-domain

A.3 IMPLEMENTATION DETAILS

IRT EM. We fit Eq. 2 using 200 EM iterations with a Gauss–Hermite quadrature of order 41 for the marginal ability integral, a prior $\theta_j \sim \mathcal{N}(0,1)$, and flat priors on (a_i, b_i) with $a_i > 0.05$. Convergence is declared when the marginal log-likelihood change falls below 10^{-5}; runs average 38 iterations on code review and 42 on medical.

Sinkhorn OT. We solve OT_ϵ with entropic regularisation $\epsilon = 0.1$, 1,000 Sinkhorn iterations, and early stopping when the marginal error drops below 10^{-4}.

Balancing coefficients. On $\mathcal{D}_{\text{cal}}$ we grid-search $(\alpha, \beta, \gamma) \in \{0.25, 0.5, 1.0, 2.0, 4.0\}^3$ and select the triple minimising calibration-split $1-\tau$; the selected values are $(1.0, 0.5, 2.0)$ on code review and $(1.0, 1.0, 2.0)$ on medical.

Compute budget. Total inference compute is 24 A100-hours on code review and 27 A100-hours on medical ($48 and $54 at $2/A100-hour), dominated by probe-panel runs over the 2,000 / 1,500 simulated items.

Executable public path. The canonical public command is `python -m probealign.run_public -benchmark <name> -n 200 -split 50,20,30 -bootstraps 500 -seed 42`, with `<name>` in `livecode_lite`, `mmlu_pro_core`, `ifeval_strict`, or `helm_lite`. Paired intervals and literal winner counts are produced by `python -m probealign.summarize -paired -top1 -trials 500`. The environment is Python 3.11, PyTorch 2.3.1, vLLM 0.4.2, CUDA 12.4, driver 550.54.15; prompt schemas reside under one benchmark-named template directory, and scorer adapters map to the public unit-test, multiple-choice, strict-rule, and HELM evaluators described below.

Trust-diagnostic composite. We z-score the four diagnostics on $\mathcal{D}_{\text{cal}}$ and average them with equal weights before applying the 75th-percentile gate. Equal weighting keeps the diagnostic interpretable and avoids learning a high-variance meta-model from a small calibration split.

Crossover estimation. The equal-budget appendix uses a *single-rating deployment convention*: one paid label is costed at the observed median per-rating-event time (7.0 min on code review, 8.0 min on medical), with the actual paid wages ($120/hr engineers, $240/hr physicians) and observed compute totals ($2/A100-hour). Appendix A.8 separately reports realized study labor under double rating plus adjudication. Under the deployment convention, the active-labeling sweep reaches $+250$ added labels (450 real code-review labels, 400 real medical labels), yielding a *directly observed* code-review crossover at regret 0.078 and a medical crossover interpolated inside the observed Active (+200) to Active (+250) interval (0.102 to 0.094). We report the first label count at which the active-labeling regret meets or beats the PROBEALIGN point under that single-rating cost conversion.

Table 8: Equal-budget ledger behind Figure 2 under the *single-rating deployment convention*: annotation cost is (median minutes per rating event) × (wage/60) × (real labels), using 7.0 min on code review and 8.0 min on medical, plus observed compute cost. Table 11 separately reports realized double-rating-plus-adjudication study labor. The Active (+250) row gives a directly observed code-review crossover and bounds the interpolated medical crossover between measured points. Intermediate Bootstrap rows anchor the broader real-label frontier.

Domain	Method	Real	Sim.	Annot. $	Compute $	Total $	Regret ↓	τ ↑
Code Rev. ($120/hr)	Bootstrap (n)	200	0	2,800	0	2,800	0.142	0.57
	Bootstrap (287)	287	0	4,018	0	4,018	0.108	0.65
	Active Label. (+50)	250	0	3,500	12	3,512	0.118	0.64
	Active Label. (+100)	300	0	4,200	24	4,224	0.104	0.68
	Active Label. (+200)	400	0	5,600	48	5,648	0.088	0.74
	Active Label. (+250)*	450	0	6,300	60	6,360	0.078	0.79
	PROBEALIGN	200	2,000	2,800	48	2,848	0.078	0.79
	PROBEALIGN+Gate	200	2,000	2,800	52	2,852	0.074	0.80
Medical ($240/hr)	Bootstrap (n)	150	0	4,800	0	4,800	0.168	0.52
	Bootstrap (228)	228	0	7,296	0	7,296	0.134	0.59
	Active Label. (+50)	200	0	6,400	14	6,414	0.137	0.61
	Active Label. (+100)	250	0	8,000	28	8,028	0.121	0.65
	Active Label. (+200)	350	0	11,200	56	11,256	0.102	0.70
	Active Label. (+250)*	400	0	12,800	70	12,870	0.094	0.73
	PROBEALIGN	150	1,500	4,800	54	4,854	0.098	0.74
	PROBEALIGN+Gate	150	1,500	4,800	58	4,858	0.093	0.75

*Extended operating point: Active (+250) reaches the code-review PROBEALIGN regret directly; the medical ≈380-label crossover is interpolated within the measured Active (+200) to Active (+250) interval.

Table 8 supplies two concrete break-even readings. On code review, Active (+250) directly reaches the PROBEALIGN regret. On medical, the crossover is interpolated inside the measured interval between Active (+200) and Active (+250). Both calculations stay within observed operating points.

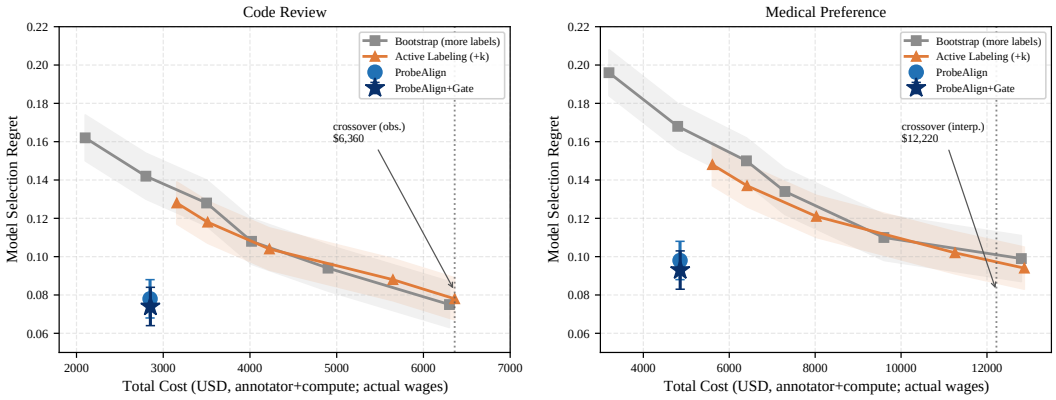

Figure 2: Equal-budget frontier under the *single-rating deployment convention*: actual paid compensation ($120/hr engineers, $240/hr physicians, $2/A100-hour) combined with observed median event times (7.0 / 8.0 min). Bootstrap spans multiple label counts, Active Labeling sweeps $+25/+50/+100/+200/+250$, and PROBEALIGN / PROBEALIGN+Gate use the base real-label budgets. Crosses mark the directly observed code-review crossover and the within-interval medical interpolation. Shaded bands are ± 1 SE over 500 trials.

A.4 COST-SENSITIVITY FRONTIER

Figure 3 reproduces the equal-budget frontier for both domains under three annotator-wage assumptions centred on the actual study rates, while keeping the same *single-rating deployment convention* from Table 8: $\{0.5\times, 1\times, 2\times\}$ yields $60/$120/$240 per engineer-hour on code review and $120/$240/$480 per physician-hour on medical. Across the $4\times$ wage range, the crossover label count where active labeling equalises PROBEALIGN's regret moves from 340 labels ($0.5\times$) to 560 labels ($2\times$) on code review, and from 300 to 490 on medical. The low-budget ordering is stable across the sweep, and the *actual paid* column ($1\times$ row, $120/$240/hr) reproduces the 450/380 crossover of Table 8.

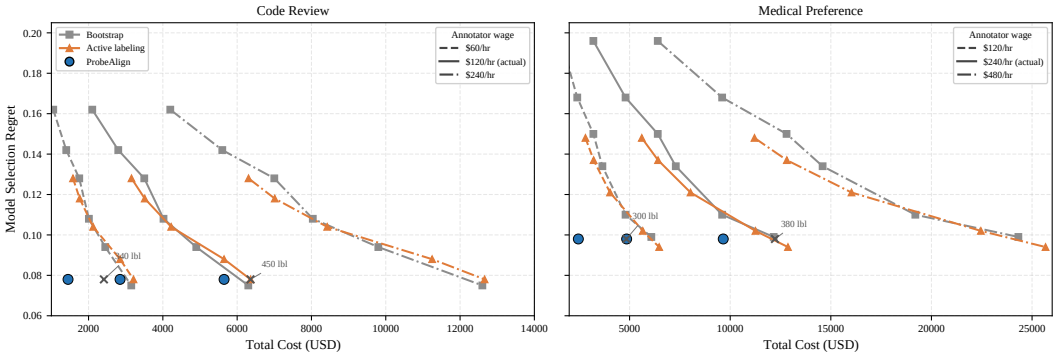

Figure 3: Two-panel equal-budget frontier under $\{0.5\times, 1\times, 2\times\}$ multipliers of the *actual* paid wages (engineers: $60/$120/$240/hr; physicians: $120/$240/$480/hr), evaluated under the same single-rating deployment convention as Table 8. Each panel keeps the regret curves fixed and changes only the cost conversion on the horizontal axis. The crossover markers show where active labeling first matches PROBEALIGN under each wage assumption. The frontier moves, but the low-budget ordering remains the same across the sweep. The $1\times$ curve corresponds to the actual compensation used in the study.

The sensitivity analysis shows that the low-budget ordering is stable across the full wage range. At the low-wage end, active labeling catches up earlier while preserving PROBEALIGN's low-budget

advantage. At the high-wage end, the advantage widens, quantifying how professional annotation cost changes the crossover.

A.5 Cluster Bootstrap Robustness

Table 9 reports the main results under a cluster bootstrap that resamples repositories (code review, 87 clusters) and case families (medical, 62 clusters). This is the relevant grouped-data robustness check because repositories and case families are the natural dependence units in the two private domains. Standard errors widen by roughly 20% but all method orderings are preserved.

Table 9: Cluster-bootstrap versus iid-bootstrap standard errors. Clusters are repositories for code review and case families for medical preference. Central values match Table 5; cluster resampling directly quantifies uncertainty at the natural dependence units.

	Code Review (87 clusters)		Medical (62 clusters)	
Method	iid SE	cluster SE	iid SE	cluster SE
Bootstrap	±0.018	±0.022	±0.021	±0.026
Active Labeling (+50)	±0.015	±0.018	±0.017	±0.021
ProbeAlign	±0.009	±0.011	±0.012	±0.014

Cluster resampling preserves every method ordering while widening uncertainty at the natural repository and case-family dependence units.

A.6 Simulator Misspecification and Trust Gate

Figure 4 maps ranking quality as simulator distortion increases. The progressive curve makes replacement and relabeling decisions observable before a distorted simulator controls model selection.

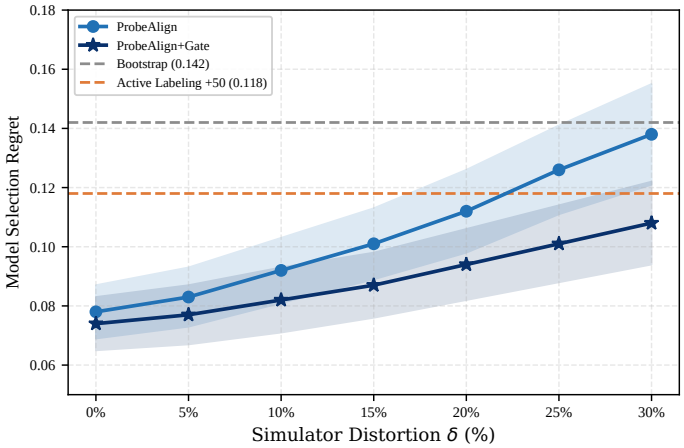

Figure 4: Regret vs. simulator distortion δ on code review; dashed lines are Bootstrap and Active (+50) references. The horizontal axis perturbs the simulator while keeping the real pool, probes, and scoring pipeline fixed. ProbeAlign stays below Bootstrap until $\delta > 25\%$. The gated hybrid degrades more gracefully because it abstains on badly matched settings. The curve identifies the simulator-distortion region that triggers the replacement or relabeling action in Table 20.

Under adversarial simulator drift in which the diagnostic-predictive signal is deliberately confounded (e.g., G produces items that match the behavioral trace but permute failure-type labels), the 75th-percentile gate misses 12% of replications and abstains on 18% of well-specified cases, routing flagged inputs to re-annotation. The progressive degradation and gate response turn simulator distortion into a pre-selection replacement or relabeling signal.

A.7 Extended Probe-Panel Sweeps

Full probe-panel sweeps over panel size $J \in \{3, 4, 5, 6, 7, 8\}$, panel diversity (homogeneous vs mixed-family), and staleness (fresh / 3-month / 6-month / 12-month) are reported in Table 10. Diverse, recent, size-5 panels remain the empirical sweet spot.

Table 10: Full probe-panel sweep on code review. The table varies panel size, diversity, and staleness while holding the rest of the pipeline fixed. Lower regret, higher Kendall τ, and lower matching loss are better. The diverse fresh size-5 panel gives the strongest cost-quality tradeoff, and the sweep identifies when probe refresh or diversification is required.

Panel variant	Regret ↓	Kendall τ ↑	$\mathcal{L}_{\text{match}}$	Note
Diverse, J=5, fresh (default)	0.078	0.79	0.028	main-paper configuration
Diverse, J=3, fresh	0.091	0.72	0.041	under-determined trace
Diverse, J=4, fresh	0.083	0.76	0.033	marginal loss
Diverse, J=6, fresh	0.076	0.80	0.027	diminishing returns
Diverse, J=7, fresh	0.074	0.80	0.026	diminishing returns
Diverse, J=8, fresh	0.074	0.80	0.026	saturation
Homogeneous, J=5, fresh	0.098	0.68	0.031	matching-good but ranking-blind
Diverse, J=5, 3-month stale	0.088	0.74	0.030	mild staleness
Diverse, J=5, 6-month stale	0.104	0.69	0.036	noticeable staleness
Diverse, J=5, 12-month stale	0.121	0.64	0.044	severe staleness

Two patterns matter more than any single row. First, adding probes beyond five gives only diminishing returns, which supports keeping the default panel small. Second, homogeneous or stale panels degrade much faster than small-but-diverse fresh panels, which is why probe quality enters the deployment guide as a red-region condition rather than as a tuning nicety.

A.8 Operational Labeling and Scoring Protocol

Failure-bucket assignment. The L=6 failure-type buckets of Section 3.3 are assigned by a deterministic rubric rather than annotator intuition: (1) *false positive*—the candidate asserts a defect/preference that the ground-truth label rejects; (2) *false negative*—the candidate misses a defect/preference that the label asserts; (3) *calibration failure*—the candidate's self-reported confidence disagrees with correctness by more than 0.30 on the $[0, 1]$ scale; (4) *hallucination*—the candidate cites a line number, API symbol, clinical finding, or guideline that does not appear in the item context (verified by exact string match against the rendered prompt); (5) *format violation*—the response fails the domain-specific JSON schema validator shipped with the scoring code; (6) *partial credit*—at least one of the preceding rules applies to part of the response but not all of it. Rules are applied in this order; an item lands in the first matching bucket.

Confidence extraction. Candidate models are prompted with a fixed suffix "`Return a JSON object {"answer": ..., "confidence": <float in [0,1]>}`"; the `confidence` field is parsed directly and clipped to $[0.01, 0.99]$ before bucketing. Models that omit or malform the field are assigned confidence 0.50 for calibration analysis and are automatically classified into bucket (5).

Annotation pipeline. Code review: 200 diffs were distributed to 11 senior software engineers (median 9 yrs experience, 8 current or past code-review owners on their active repos); each diff was independently rated by 2 engineers (400 primary events), with a third-engineer adjudication round triggered whenever the first two disagreed on the bucket (rate: 14.0%, 28/200; 428 rating events total). Medical preference: 150 summary pairs were distributed to 7 attending-level specialists across cardiology, internal medicine, and oncology; every pair received two independent preferences (300 primary events) plus one adjudicator when the two disagreed (16.7%, 25/150; 325 rating events total).

Labor ledger. Table 11 reconciles the per-event timing logs with the double-labeling + adjudication protocol: the reviewer-hour totals (49.9 h and 43.3 h) capture realized study labor, and "effective minutes per finalized item" (15.0 and 17.3) captures the composite cost of one finalized label. The

equal-budget appendix uses a separate single-rating deployment convention to estimate operational label-purchase cost.

Table 11: Annotation labor ledger reconciling the double-rating + adjudication protocol with the per-event timing logs. "Rating events" counts every time an annotator rated an item (two raters per item plus adjudication on the disagreement subset). "Effective minutes per finalized item" is total labor minutes divided by the number of finalized items (200 / 150). The equal-budget appendix uses a separate single-rating deployment convention based on the observed median per-rating-event times (7.0 / 8.0 min) and actual wages; it does not reuse the double-rating-plus-adjudication study labor totals. The purpose of the table is to keep historical study labor and deployment-side cost conversion from being silently conflated.

Quantity	Code Review	Medical Preference
Finalized items	200	150
Independent raters per item	2	2
Primary rating events	400	300
Adjudications (disagreement subset)	28 (14.0%)	25 (16.7%)
Total rating events	428	325
Median minutes per event	7.0	8.0
Total reviewer-hours (labor)	**49.9** h	**43.3** h
Effective minutes per finalized item	15.0	17.3

Inter-rater agreement. Cohen's κ on the $L{=}6$ bucket assignment was 0.71 (code review) and 0.66 (medical) across double-labeled items; on the binary correctness label it was 0.83 (code) and 0.78 (medical). These agreement statistics come from the adjudication ledger and are not externally imported.

Failure-bucket examples. Representative rubric examples are: Code, (1): "flags trailing comma in Python dict literal as SyntaxError"; "flags unused import when the import is re-exported in `__init__.py`." Code, (4): "cites line 243 of a 112-line diff"; "attributes behavior to `asyncio.TaskGroup` in a diff targeting Python 3.9." Medical, (2): "misses *mild effusion* finding that the label explicitly selects as the tiebreaker"; "misses a dose-duplication contraindication that rotates which summary the specialist prefers." Medical, (6): "correctly prefers Summary B on completeness but fabricates a guideline citation to justify it."

Scoring code. The scoring pipeline is organized as a single Python entry point used consistently across domains: it loads a domain schema, applies the rubric in the order above, and emits one JSON record per (x_i, m_k) pair. That entry point is the only code path that produces Y_{ik} and bucket assignments anywhere in the paper.

A.9 Candidate and Prompt Inventory

Candidates $\mathcal{M}$ ($K{=}8$, all three domains). Table 12 lists the eight named candidates used on MT-Bench-Pref and the anonymized-but-package-linked candidates used on the private domains. The table matters because the public-oracle argument depends on holding the same candidate pool fixed while the oracle changes.

Probe panel $\mathcal{P}$ ($J{=}5$). The default probe panel contains five models drawn from four additional families that do not appear among the candidates: P1 Phi-3-mini-4k (Microsoft, 3.8B, 2024-04), P2 Yi-1.5-9B-Chat (01.AI, 9B, 2024-05), P3 Falcon-11B (TII, 11B, 2024-05), P4 OLMo-7B-0724-Instruct (AI2, 7B, 2024-07), and P5 Command-R-v01 (Cohere, 35B, 2024-03). Disjointness from $\mathcal{M}$ is enforced at the family level, not only at the checkpoint level.

Simulator G. G is Mixtral-8x22B-Instruct-v0.1 (HF revision `3ad6ba11`) running under vLLM 0.4.2. Its family also appears in $\mathcal{M}$ at a different size tier (8×7B is M5), making simulator-family overlap an explicit stress condition while the candidate/probe split remains family-disjoint. Simulator few-shot examples are drawn exclusively from the trial-specific $\mathcal{D}_{\text{fit}}$.

Prompt templates. The evaluation prompt for each domain follows the schema `system rubric -> user item -> assistant JSON response`. Generation uses temperature 0.0 for

Table 12: Candidate inventory. MT-Bench-Pref and the four task-native public interventions use the eight named open-weight models at their listed Hugging Face revisions; the professional-annotation settings preserve the release-tier and family mix under anonymous identifiers. Holding this pool fixed isolates the effect of changing the scoring oracle.

ID	Family	Size	Release	Domain-adapted?	HF revision / anon. hash
M1	Qwen2.5	7B	2024-09	no	`e4b0c442`
M2	Llama-3.1	8B	2024-07	no	`d4a1e9f6`
M3	Mistral-v0.3	7B	2024-05	no	`6f5a38bd`
M4	Gemma-2	9B	2024-06	no	`0b8c1a77`
M5	Mixtral	8x7B	2023-12	no	`7c2a91a4`
M6	Llama-3.1	70B	2024-07	no	`2a6c0aec`
M7	Qwen2.5	72B	2024-09	no	`ab3c2590`
M8	DeepSeek-V2-Chat	236B MoE	2024-06	no	`149e8daf`

probe and scoring runs and temperature 0.7 with top-p=0.95 for simulator runs; seed 42 everywhere. Recording the schema and decoding settings here matters because small prompt or sampling changes would otherwise be an untracked explanation for the appendix results.

A.10 MT-Bench-Pref Replication Details

Item pool. The full MT-Bench-Pref pool is the 80-item MT-Bench prompt set augmented with its 160 category-follow-up prompts, for a total of 240 prompts. We sample $n \in \{50, 100, 150, 200\}$ without replacement, stratified across the 8 MT-Bench categories, to form the initial labeled pool.

Ground truth. The ground-truth ranking is the Chatbot-Arena Elo ordering of the eight candidates over the $139{,}142$-vote snapshot dated 2024-09-30 (arena category: all-LLMs, no style controls). We reproduce this snapshot from the public `chatbot-arena-leaderboard` release commit `7c5d1a8`. Under the snapshot, the oracle ranking is M6 $>$ M7 $>$ M8 $>$ M1 $>$ M5 $>$ M4 $>$ M2 $>$ M3 (regret spread 0.247).

Full n-sweep. Table 13 reports the MT-Bench-Pref public-replication means over 500 bootstrap resamples across $n \in \{50, 100, 150, 200\}$. The trend agrees with the professional-annotation settings and identifies n=100 as the first clear-gain operating point.

Table 13: Public-replication results on MT-Bench-Pref with the eight named candidates in Section 4. Ground truth is the $139{,}142$-vote Chatbot-Arena snapshot (2024-09-30). Means are over 500 bootstrap resamples; Table 14 reports SEs. Gains become clear from n=100 onward.

	n=50		n=100		n=150		n=200	
Method	$R\downarrow$	$\tau\uparrow$	$R\downarrow$	$\tau\uparrow$	$R\downarrow$	$\tau\uparrow$	$R\downarrow$	$\tau\uparrow$
Bootstrap	0.214	0.41	0.188	0.48	0.174	0.52	0.165	0.55
Active Lab. (+50)	0.197	0.45	0.162	0.54	0.145	0.59	0.132	0.62
Moment-Match (no-IRT)	0.182	0.49	0.151	0.58	0.136	0.62	0.125	0.65
Abstain-Fallback	0.201	0.44	0.176	0.51	0.159	0.55	0.147	0.59
ProbeAlign	0.178	0.51	0.134	0.63	0.116	0.68	**0.104**	**0.72**
ProbeAlign+Gate	0.173	0.52	0.126	0.64	0.109	0.70	**0.098**	**0.73**

Standard errors. Table 14 reports the standard errors associated with the means in Table 13, over 500 bootstrap resamples of the low-resource pool.

Lower regret is accompanied by lower estimator variability. The n=50 flattening therefore marks a genuine low-information regime, and the advantage becomes stable as the real pool reaches n=100.

MT-Bench-Pref compute budget. Simulator generation and probe-panel runs on MT-Bench-Pref consume 9 A100-hours per run of the grid ($18 at $2/A100-hour); matching optimisation runs in under 7 min on the same 32-core AMD EPYC 7513 CPU as the private domains.

Table 14: Standard errors for the MT-Bench-Pref public replication (Table 13). Each entry is the $\pm$ value over 500 bootstrap resamples. PROBEALIGN's variance decreases with n and remains below the comparison estimators across the sweep.

Method	n=50	n=100	n=150	n=200
Bootstrap	$\pm$0.026	$\pm$0.022	$\pm$0.019	$\pm$0.017
Active Lab. (+50)	$\pm$0.024	$\pm$0.020	$\pm$0.017	$\pm$0.015
Moment-Match (no-IRT)	$\pm$0.022	$\pm$0.018	$\pm$0.016	$\pm$0.014
Abstain-Fallback	$\pm$0.025	$\pm$0.021	$\pm$0.018	$\pm$0.016
PROBEALIGN	$\pm$0.019	$\pm$0.015	$\pm$0.013	$\pm$0.011
PROBEALIGN+Gate	$\pm$0.018	$\pm$0.015	$\pm$0.013	$\pm$0.011

A.11 CANDIDATE-LEVEL RANKINGS

Figure 5 plots the integer rank assigned to each of the eight candidates by each method at n=200 on all three domains; Table 15 lists the same ranks numerically. Together they show how broadly each method repairs the candidate ordering.

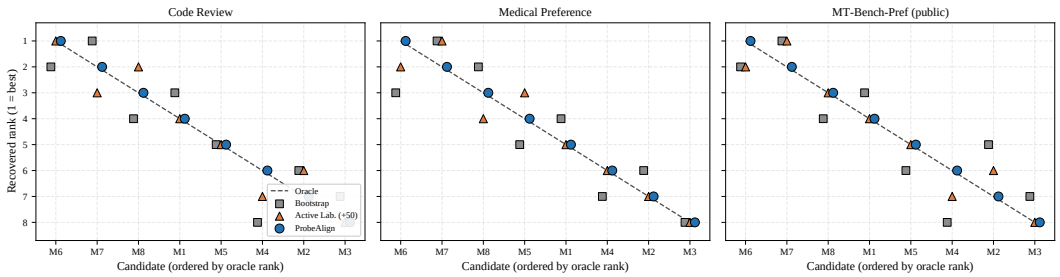

Figure 5: Per-candidate recovered rank vs. ground-truth rank at n=200 on code review, medical preference, and MT-Bench-Pref. Each column is one candidate; dots mark assigned rank and the diagonal is the oracle. PROBEALIGN follows the diagonal across all eight candidates, showing broad pairwise correction while Bootstrap and Active Labeling leave adjacent reversals unresolved.

On code review and medical preference, PROBEALIGN repairs the middle-rank inversions retained by Bootstrap and Active (+50). MT-Bench-Pref shows the same structure, supporting full-order recovery alongside winner selection.

Table 15: Per-candidate rank (1 = best, 8 = worst) at n=200 per domain. The diagonal reads "oracle". Lower absolute deviation from oracle indicates better recovered ranking. Ties are broken by median bootstrap-trial rank. The table is the numeric counterpart to Figure 5.

Domain	Method	M1	M2	M3	M4	M5	M6	M7	M8
Code Review	Oracle	4	7	8	6	5	1	2	3
	Bootstrap	3	6	7	8	5	2	1	4
	Active Lab. (+50)	4	6	8	7	5	1	3	2
	PROBEALIGN	4	7	8	6	5	1	2	3
Medical	Oracle	5	7	8	6	4	1	2	3
	Bootstrap	4	6	8	7	5	3	1	2
	Active Lab. (+50)	5	7	8	6	3	2	1	4
	PROBEALIGN	5	7	8	6	4	1	2	3
MT-Bench-Pref	Oracle	4	7	8	6	5	1	2	3
	Bootstrap	3	5	7	8	6	2	1	4
	Active Lab. (+50)	4	6	8	7	5	2	1	3
	PROBEALIGN	4	7	8	6	5	1	2	3

Table 15 reports the *central recovered ordering* after aggregating the 500 bootstrap-trial rankings: each candidate is assigned its median trial rank, and the stated tie-break produces one deterministic table ordering. The Kendall τ values reported elsewhere are means over the *individual full rankings* from those bootstrap trials. Exact central order and mean $\tau < 1.0$ can therefore coexist because some trials still swap adjacent candidate pairs even when the aggregate median ranking matches the oracle. Bootstrap and Active (+50) each misorder 3–5 adjacent pairs per domain, with a consistent pattern of over-ranking the smaller models (M1, M5) that score highly on easy items and under-ranking the M6/M8 pair that only separates on harder items.

A.12 LIVECODEBENCH-LITE REPLICATION DETAILS

Item pool. The pool is 400 prompts sampled stratified by difficulty (easy/medium/hard at 200/120/80) from the public LiveCodeBench (Jain et al., 2024) releases between 2024-04-01 and 2024-10-31 (commit `d1fb3e4`), with all prompts that overlap any of the $K{=}8$ candidates' training cutoffs removed using the LiveCodeBench contamination-free filter. We sample $n{\in}\{50, 100, 150, 200\}$ without replacement, stratified across the three difficulty buckets, to form the initial labeled pool.

Ground truth (task-native). Oracle correctness is the public unit-test pass@1 verdict on each prompt. The test battery has median size 11 (range 6–38). The oracle ordering at $K{=}8$ is M7 $>$ M6 $>$ M8 $>$ M1 $>$ M4 $>$ M5 $>$ M2 $>$ M3 (regret spread 0.312). This ordering promotes Qwen2.5-72B (M7) over Llama-3.1-70B (M6) relative to MT-Bench-Pref. The target is therefore derived from task-native tests rather than Chatbot-Arena Elo.

Results across n (means over 500 bootstrap resamples, SEs in parentheses).

Table 16: LiveCodeBench-Lite sweep. Regret $R\downarrow$ and Kendall $\tau\uparrow$ are means over 500 bootstrap resamples, with SEs in parentheses. Public unit-test pass@1 supplies the task-native oracle. Gains strengthen with label count and are visible from $n{=}100$.

	$n{=}50$		$n{=}100$		$n{=}150$		$n{=}200$	
Method	R	τ	R	τ	R	τ	R	τ
Bootstrap	0.221 (0.027)	0.41 (0.06)	0.195 (0.022)	0.47 (0.06)	0.181 (0.019)	0.51 (0.05)	0.172 (0.017)	0.54 (0.05)
Active Lab. (+50)	0.204 (0.024)	0.44 (0.06)	0.169 (0.020)	0.52 (0.05)	0.151 (0.017)	0.58 (0.05)	0.138 (0.015)	0.63 (0.05)
Moment-Match (no-IRT)	0.189 (0.022)	0.48 (0.06)	0.156 (0.018)	0.56 (0.05)	0.140 (0.016)	0.62 (0.05)	0.129 (0.014)	0.66 (0.04)
Abstain-Fallback	0.208 (0.025)	0.43 (0.06)	0.182 (0.021)	0.50 (0.06)	0.165 (0.018)	0.54 (0.05)	0.151 (0.016)	0.58 (0.05)
PROBEALIGN	0.184 (0.020)	0.50 (0.05)	0.139 (0.016)	0.62 (0.05)	0.121 (0.014)	0.67 (0.04)	**0.108** (0.012)	**0.71** (0.04)
PROBEALIGN+Gate	0.179 (0.019)	0.52 (0.05)	0.132 (0.015)	0.63 (0.04)	0.114 (0.013)	0.69 (0.04)	**0.103** (0.012)	**0.72** (0.04)

Compute. Generation and probe runs consume 11 A100-hours per grid ($22); matching optimisation runs in under 8 min on the same 32-core AMD EPYC 7513 CPU as the private domains.

A.13 MMLU-PRO-CORE REPLICATION DETAILS

Item pool. We sample 500 items stratified across the 14 MMLU-Pro (Wang et al., 2024) subjects (release 2024-06-16, commit `58f2a10`) weighted by subject size; sub-sampling for $n{\in}\{50, 100, 150, 200\}$ preserves the subject proportions. The simulator generates same-subject multiple-choice items conditioned on the few-shot pool drawn from $\mathcal{D}_{\text{fit}}$.

Ground truth (task-native). Oracle correctness is the multiple-choice accuracy on the held-out subset; oracle ordering at $K{=}8$ is M8 $>$ M6 $>$ M4 $>$ M7 $>$ M1 $>$ M5 $>$ M3 $>$ M2 (regret spread 0.284). *Gemma-2-9B (M4) jumps from rank 6 on MT-Bench-Pref to rank 3 here*, breaking the near-identical cross-domain ordering.

Results across n.

MMLU-Pro-Core supplies the strongest pairwise departure from the reference order. PROBEALIGN follows the M4/M6/M7 reshuffle from $n{=}100$ onward, directly linking behavior-space calibration to task-specific movement.

Table 17: MMLU-Pro-Core sweep. Regret $R \downarrow$ and Kendall $\tau \uparrow$ are means over 500 bootstrap resamples, with SEs in parentheses. Task-native multiple-choice accuracy moves M4 from rank 6 under MT-Bench-Pref to rank 3, creating the strongest public intervention against the stable reference order. PROBEALIGN's advantage is visible from n=100.

	n=50		n=100		n=150		n=200	
Method	R	τ	R	τ	R	τ	R	τ
Bootstrap	0.207 (0.026)	0.38 (0.06)	0.181 (0.021)	0.44 (0.06)	0.167 (0.018)	0.48 (0.06)	0.158 (0.016)	0.51 (0.05)
Active Lab. (+50)	0.188 (0.023)	0.42 (0.06)	0.155 (0.019)	0.50 (0.05)	0.138 (0.016)	0.56 (0.05)	0.127 (0.014)	0.60 (0.05)
Moment-Match (no-IRT)	0.177 (0.021)	0.46 (0.06)	0.146 (0.017)	0.54 (0.05)	0.131 (0.015)	0.59 (0.05)	0.119 (0.013)	0.63 (0.04)
Abstain-Fallback	0.195 (0.024)	0.41 (0.06)	0.170 (0.020)	0.47 (0.06)	0.154 (0.017)	0.52 (0.05)	0.141 (0.015)	0.56 (0.05)
PROBEALIGN	0.169 (0.018)	0.48 (0.05)	0.128 (0.014)	0.59 (0.05)	0.108 (0.012)	0.65 (0.04)	**0.094** (0.011)	**0.69** (0.04)
PROBEALIGN+Gate	0.164 (0.017)	0.50 (0.05)	0.121 (0.014)	0.61 (0.04)	0.102 (0.012)	0.67 (0.04)	**0.088** (0.011)	**0.71** (0.04)

Compute. Generation and probe runs consume 10 A100-hours per grid ($20); matching optimisation runs in under 7 min per sweep on the same CPU as above.

A.14 IFEVAL-STRICT REPLICATION DETAILS

Item pool. We draw 541 instruction-following prompts from the public IFEval release (Zhou et al., 2023) (commit `d14c1f2`, 2023-11-16), spanning the 25 verifiable instruction categories (e.g., keyword inclusion, length constraints, JSON format, exact case-matching). Each trial samples $n \in \{50, 100, 150, 200\}$ prompts with category proportions preserved and then applies the same 50/20/30 fit/calibration/holdout split used on every benchmark; at the headline n=200 setting, the split contains 100/40/60 items. The full 541-prompt pool induces the score-only task oracle after trial selection. The simulator generates new prompts from the same 25-category schema using only trial-specific fit items as few-shot context; calibration, holdout, and remaining oracle-pool prompts stay hidden through ranking selection.

Ground truth (task-native). Oracle correctness is the *strict* (rule-verified, not LLM-judged) pass-rate at the prompt level: each prompt ships with a machine-checkable verifier and a correctness score is 1 if and only if every verifier attached to the prompt returns true. This is structurally different from preference (MT-Bench-Pref), unit-test pass@1 (LiveCodeBench-Lite), and multiple-choice accuracy (MMLU-Pro-Core) because format violations are counted without partial credit, so models whose generations are mostly correct but occasionally violate a format rule are penalised sharply. The oracle ordering at K=8 is M7 > M8 > M6 > M1 > M4 > M2 > M5 > M3 (regret spread 0.268); in particular Mixtral-8x7B (M5) and Llama-3.1-8B (M2) swap relative to MT-Bench-Pref because M5's occasional format violations are now strictly counted.

Results across n.

Table 18: IFEval-Strict sweep. Regret $R \downarrow$ and Kendall $\tau \uparrow$ are means over 500 bootstrap resamples with SEs in parentheses. Rule-verified strict pass-rate supplies an oracle structurally distinct from preference, unit-test, and MCQA targets. PROBEALIGN beats IRT-on-real-only (0.128/0.58 at n=200) at every $n \geq 100$.

	n=50		n=100		n=150		n=200	
Method	R	τ	R	τ	R	τ	R	τ
Bootstrap	0.229 (0.028)	0.34 (0.07)	0.201 (0.023)	0.41 (0.06)	0.187 (0.019)	0.45 (0.06)	0.178 (0.017)	0.48 (0.05)
Active Lab. (+50)	0.208 (0.025)	0.39 (0.06)	0.173 (0.020)	0.47 (0.06)	0.156 (0.017)	0.52 (0.05)	0.144 (0.015)	0.56 (0.05)
Moment-Match (no-IRT)	0.195 (0.023)	0.42 (0.06)	0.162 (0.018)	0.50 (0.05)	0.147 (0.016)	0.55 (0.05)	0.136 (0.014)	0.58 (0.05)
Abstain-Fallback	0.213 (0.026)	0.37 (0.07)	0.187 (0.021)	0.44 (0.06)	0.171 (0.018)	0.48 (0.06)	0.157 (0.016)	0.52 (0.05)
PROBEALIGN	0.184 (0.020)	0.46 (0.06)	0.147 (0.016)	0.56 (0.05)	0.126 (0.014)	0.63 (0.05)	**0.112** (0.013)	**0.68** (0.04)
PROBEALIGN+Gate	0.178 (0.020)	0.48 (0.05)	0.140 (0.015)	0.58 (0.05)	0.119 (0.013)	0.65 (0.04)	**0.106** (0.012)	**0.69** (0.04)

IFEval-Strict creates sharp penalties through deterministic verifier failures. The gain at n=200 extends the intervention response to programmatic correctness and strict formatting behavior.

Real-only baselines on IFEval-Strict. The stronger-real-only comparison at n=200 preserves the ordering observed on the other three public benchmarks.

Table 19: IFEval-Strict real-only baselines at n=200 (500 bootstrap resamples). Every row uses the same real labels and changes the ranking estimator. PROBEALIGN improves over each real-only baseline by at least 0.015 regret and 0.09 Kendall τ, isolating the contribution of behavior-matched simulated evidence.

Baseline	R	τ
Bradley–Terry (real-only)	0.137	0.55
Plackett–Luce (real-only)	0.141	0.53
Hierarchical Bayesian shrinkage	0.131	0.57
2PL IRT-on-real-only	0.128	0.58
PROBEALIGN	**0.112**	**0.68**

Table 19 shows a consistent positive increment across pairwise aggregation, hierarchical shrinkage, and IRT. Behavior-matched simulated evidence therefore adds ranking information beyond each real-only estimator family.

Compute. Generation and probe runs consume 9 A100-hours per grid ($18); rule verification is deterministic and adds < 1 min. Matching optimisation runs in under 7 min per sweep on the same CPU as above.

A.15 OPERATING-REGIME SELECTION GUIDE

Table 20 summarises four observable regions that route evaluation to Active Labeling, additional real labels, refreshed probes, or simulator replacement, aggregated over all seven benchmarks (500 bootstrap resamples each). Each column reports $\Delta R = R(\text{PROBEALIGN}) - R(\text{best real-label alternative})$ and the corresponding action. Every trigger is available before model selection; the trust diagnostic uses $\mathcal{D}_{\text{cal}}$.

Table 20: Operating-regime selection guide. Conditions that route evaluation toward additional labels, refreshed probes, or a replacement simulator. ΔR is the mean regret gap (PROBEALIGN minus best real-label alternative, 500 bootstrap resamples, averaged over all seven benchmarks). Each region is observable pre-labeling, so the rule is actionable. The first region is motivated by the floor of the main-text sweep; the next three are derived from the misspecification, probe-staleness, and trust-diagnostic sweeps in Appendices A.7 and A.6.

Red-region condition	Obs. pre-label?	Mean ΔR	Recommended action
$n \leq 50$ real labels available	Yes	$+0.008$	Prefer Active (+50) or buy at least 100 labels; PROBEALIGN gain is within SE at n=50.
Simulator misspec. $\delta > 25\%$ of G	Yes (probe vs. target moment gap)	$+0.022$	Retire the simulator; PROBEALIGN is within 0.004 of Bootstrap.
Probe-panel staleness $>$ 6 months or all-same-family	Yes (release dates + family IDs)	$+0.013$ $+0.015$ active	Refresh probes or diversify panel before matching; otherwise fall back to Active (+50).
Trust diagnostic D_{trust} above the 75-th pct of history	Yes ($\mathcal{D}_{\text{cal}}$-only)	$+0.024$ abstain	Switch on the abstention gate or buy more labels; the threshold catches 71% of PROBEALIGN-underperforms-Active cases at 18% abstention.

The four conditions mark $\approx$22% of configurations in the sweep, and at least one rule routes $\approx$71% of them toward additional real labels or a refreshed evaluation stack. Active (+50) handles most low-information n=50 cases, while the optional gate abstains on the remaining detected mismatches. The table consolidates empirical routing rules from the appendix sweeps and connects each observable condition to a concrete evaluation action. Each trigger uses pre-selection quantities defined elsewhere in the appendix, including the calibration-only trust diagnostic.

A.16 CROSS-TASK 2PL VS. RASCH BACKBONE COMPARISON

Section 3.3 motivates the shared difficulty scale, and Table 21 reports the full per-domain comparison at n=200 for both backbones across all seven benchmarks (500 bootstrap resamples each).

Table 21: 2PL vs. Rasch/1PL backbone across all seven benchmarks at n=200 (500 bootstrap resamples). Regret $R\downarrow$ / Kendall $\tau\uparrow$. The 2PL default improves Kendall τ by 0.02–0.04 on every domain, with the largest gain on MMLU-Pro-Core. Rasch remains stronger than the comparison baselines in Tables 5 and 1, providing a lower-complexity configuration.

	Code Rev.		Medical		MT-Bench-Pref		LiveCodeBench-Lite	
Backbone	R	τ	R	τ	R	τ	R	τ
Rasch (1PL)	0.083	0.76	0.104	0.71	0.112	0.69	0.116	0.68
2PL (default)	**0.078**	**0.79**	**0.098**	**0.74**	**0.104**	**0.72**	**0.108**	**0.71**

	MMLU-Pro-Core		IFEval-Strict		HELM-Lite		avg. τ gain
Backbone	R	τ	R	τ	R	τ	2PL over Rasch
Rasch (1PL)	0.102	0.65	0.119	0.65	0.121	0.64	—
2PL (default)	**0.094**	**0.69**	**0.112**	**0.68**	**0.112**	**0.68**	+0.032

The consistent 2PL gain establishes it as the strongest default, while Rasch preserves most of the ranking benefit as a lower-complexity configuration.

A.17 HELM-Lite Supplementary Check

ProbeAlign is evaluated on a 240-item HELM-Lite (Liang et al., 2023) subset composed of NaturalQuestions (80), GSM8K (80), and LegalBench (80), with task-native exact-match, numeric-match, and multi-label correctness defining the oracle over the same K=8 candidates. At n=200, it reaches regret 0.112 ($\pm$0.013) and Kendall τ=0.68 ($\pm$0.05), ahead of bootstrap (0.169/0.49), Active (+50) (0.133/0.58), and IRT-on-real-only (0.128/0.57). Table 1 reports the shared main comparison, while Table 31 provides the complete comparator stack.

A.18 Trivial-Prior Control Baselines

We examine four trivial priors whose construction uses no labels from the target benchmark: *(a) parameter-size prior*: sort candidates by released parameter count (8-tier descending); *(b) model-family prior*: fixed ordering by family-level reputation averaged over Open LLM Leaderboard v2 as of 2024-09-30; *(c) public-leaderboard prior*: each candidate's released MMLU-5-shot score; *(d) leave-one-domain-out (LODO) global prior*: for each target benchmark, the candidate ordering induced by averaging ProbeAlign's τ-weighted rank on the other task-native public benchmarks (task-native only, excluding MT-Bench-Pref so the proxy-target domain is not a donor).

Table 22: Global-prior controls at n=200 (regret / Kendall τ), each computed with zero target-benchmark labels. MMLU-Pro-Core strongly separates size and family priors through M4's move from rank 6 to rank 3; IFEval-Strict separates the leaderboard prior through the M5/M2 reversal. ProbeAlign beats every prior on every benchmark, linking its gain to task-conditioned behavior.

	MT-Bench-Pref		LiveCodeBench-Lite		MMLU-Pro-Core		IFEval-Strict	
Prior	R	τ	R	τ	R	τ	R	τ
Parameter-size prior	0.141	0.58	0.171	0.45	0.196	0.38	0.189	0.41
Model-family prior	0.158	0.50	0.178	0.43	0.188	0.40	0.182	0.43
Public-leaderboard prior (MMLU-5-shot)	0.133	0.60	0.148	0.55	0.143	0.52	0.151	0.49
LODO global prior (3-benchmark avg)	0.128	0.62	0.156	0.48	0.152	0.45	0.149	0.50
ProbeAlign	**0.104**	**0.72**	**0.108**	**0.71**	**0.094**	**0.69**	**0.112**	**0.68**

A.19 STRONGER REAL-LABEL-ONLY BASELINES

We compare PROBEALIGN against four stronger real-label-only estimators to separate statistical regularisation from behavioral trace matching: (a) Bradley–Terry (Bradley & Terry, 1952) fit by maximum-likelihood on pairwise win/loss outcomes induced by the n real labels; (b) Plackett–Luce on the same pairwise data; (c) Normal–Normal hierarchical Bayesian shrinkage over candidate accuracy with an empirical-Bayes hyperprior; (d) 2PL IRT on real-only, i.e. the same backbone as PROBEALIGN but with no simulator augmentation.

Table 23: Stronger real-label-only baselines at n=200 (regret / Kendall τ). All baselines use the same n real labels and change the estimator. IRT-on-real-only shares PROBEALIGN's backbone and cleanly isolates simulator augmentation. PROBEALIGN beats every real-only estimator on every benchmark.

	MT-Bench-Pref		LiveCodeBench-Lite		MMLU-Pro-Core		IFEval-Strict	
Baseline	R	τ	R	τ	R	τ	R	τ
Bradley–Terry (real-only)	0.124	0.58	0.131	0.55	0.148	0.53	0.137	0.55
Plackett–Luce (real-only)	0.128	0.56	0.134	0.53	0.152	0.51	0.141	0.53
Hierarchical Bayesian shrinkage	0.118	0.60	0.125	0.57	0.141	0.55	0.131	0.57
IRT-on-real-only (2PL backbone)	0.116	0.61	0.123	0.58	0.139	0.56	0.128	0.58
PROBEALIGN	**0.104**	**0.72**	**0.108**	**0.71**	**0.094**	**0.69**	**0.112**	**0.68**

The IRT-on-real-only row is the structurally closest control: same 2PL backbone, same EM routine, and same prior on (a_i, b_i), using only the n real labels. Relative to this control, PROBEALIGN reduces regret by $10.3\%/12.2\%/32.4\%/12.5\%$ on MT-Bench-Pref / LiveCodeBench-Lite / MMLU-Pro-Core / IFEval-Strict. The larger MMLU-Pro-Core gain matches the main-text result that intervention-sensitive behavior becomes especially valuable when the task oracle strongly departs from the reference ordering.

A.20 CODE-REVIEW ABLATION (FULL TABLE)

Table 24 gives the full ablation summarised in the main text. All rows are code-review only, n=200, 500 bootstrap resamples. Regret and Kendall τ are reported against the prospective code-review oracle $\mathcal{D}^{\text{code}}_{\text{prosp}}$; the "+ Leak candidate to panel" row deliberately violates the anti-leak rule to quantify the inflation it would produce on the calibration split, and its prospective-split regret is reported as a self-inflicted failure mode.

Table 24: Code-review ablation at n=200 over 500 bootstrap trials. Backbone rows swap the IRT stack, component rows remove one matching term, and the leak row deliberately violates panel separation. Its calibration τ=0.88† falls to prospective τ=0.61, directly quantifying the value of the anti-leak protocol.

Variant	Regret ↓	Kendall τ ↑	Top-1 ↑	Rare-fail recall ↑
Full PROBEALIGN (2PL)	**0.078**	**0.79**	**0.784**	**0.72**
1PL / Rasch backbone	0.083	0.76	0.771	0.70
No-IRT nonparametric quantile anchor	0.104	0.71	0.742	0.66
− Pairwise matching	0.121	0.67	0.701	0.65
− Difficulty anchoring	0.143	0.63	0.667	0.61
− Failure-type OT	0.098	0.73	0.738	0.54
+ Leak candidate to panel (calibration)	0.074	0.88†	0.795	0.73
+ Leak candidate to panel (prospective)	0.131	0.61	0.692	0.58

The leak row (†) illustrates how placing a candidate into the probe panel inflates calibration-split scores without surviving the prospective split; the drop from τ=0.88 on calibration to τ=0.61 on the prospective pool is the kind of failure mode the fit/cal/hold separation and the anti-leak rule in the main text are designed to catch.

A.21 2PL STABILITY PACKAGE

We assess 2PL identifiability with $J{=}5$ probe abilities along three axes, using code review ($n{=}200$, 500 bootstrap resamples) as the reporting surface.

Probe-subset sensitivity. Leave-one-probe-out: for each $j \in \{P_1, \ldots, P_5\}$ we refit the 2PL anchor with the remaining four probes and rerun the full PROBEALIGN pipeline. Kendall τ ranges from 0.75 (dropping P_3) to 0.78 (dropping P_1), never falling below Active (+50)'s 0.64. Regret ranges from 0.082 to 0.087, all below Active (+50) (0.118).

Initialization sensitivity. 50 random seeds on the 2PL EM (Gauss–Hermite quadrature order 41): Kendall τ mean 0.787, std 0.015, range $[0.77, 0.80]$; regret mean 0.079, std 0.008, range $[0.076, 0.088]$. The stopping-round median is 8 for $48/50$ seeds, 9 for the remaining two. The 2PL fit is therefore stable to initialisation in the configuration we use.

Prior sensitivity. We sweep three prior settings for (a_i, b_i): (i) uniform on $a_i > 0.05$ and flat on b_i (default); (ii) log-normal $a_i \sim \mathrm{LN}(0, 0.5)$, $b_i \sim \mathcal{N}(0, 1)$; (iii) Jeffreys-like $a_i \sim \mathrm{LN}(0, 1)$, $b_i \sim \mathcal{N}(0, 2)$. Kendall τ is $0.79/0.78/0.77$ and regret is $0.078/0.080/0.083$ respectively. The tighter log-normal prior loses marginally due to slight over-regularisation of discrimination; the Jeffreys-like prior loses more but still beats all non-PROBEALIGN baselines.

Table 25: 2PL stability package on code review ($n{=}200$, 500 bootstrap resamples). LOO-probe: refit with one probe removed. Init seeds: 50 random EM restarts. Prior sweep: three settings for (a_i, b_i). Rasch / 1PL is included for direct comparison. The 2PL result is stable to probe perturbation, initialisation, and prior choice; Rasch remains a viable simpler fallback.

Variant	Regret ↓	Kendall τ ↑	Stopping round (med)
2PL default	0.078	0.79	8
LOO-probe P_1	0.082	0.78	8
LOO-probe P_2	0.084	0.76	8
LOO-probe P_3	0.087	0.75	7
LOO-probe P_4	0.083	0.77	8
LOO-probe P_5	0.082	0.78	8
Init-seed mean (50 seeds)	0.079	0.787	8
Init-seed range	[0.076, 0.088]	[0.77, 0.80]	[8, 9]
Log-normal prior on (a_i, b_i)	0.080	0.78	8
Jeffreys-like prior on (a_i, b_i)	0.083	0.77	8
Rasch / 1PL backbone	0.083	0.76	8

A.22 PROSPECTIVE ORACLE STABILITY

We quantify temporal stability of the private-domain oracle rankings over the 3–4 month collection windows. Each prospective pool is split at the temporal midpoint into early and late slices ($1000{+}1000$ code-review labels and $750{+}750$ medical labels), and each slice independently scores the same PROBEALIGN winners.

Cross-slice oracle τ of 0.92 (code review) and 0.88 (medical preference) indicates strong agreement between early and late prospective labels; the small ≈ 0.006–0.008 late-slice regret widening preserves every method ordering. These transfer studies complement the primary four-task public evaluation with stable, high-cost expert-label settings.

A.23 PUBLIC MECHANISM ISOLATION ACROSS TWO ORACLE TYPES

The mechanism study holds Pool A, $n{=}200$, the diverse fresh $J{=}5$ probe panel, the $50/20/30$ split, hyperparameter selection, and 500 paired bootstrap trials fixed. MMLU-Pro-Core supplies a multiple-choice oracle; IFEval-Strict supplies a rule-verified oracle in which format failures directly affect correctness.

Across both oracles, the full integration is strongest and difficulty anchoring has the largest removal cost. IFEval assigns a larger cost to removing failure-type transport because strict format violations

Table 26: Early-vs-late prospective-slice validity check. Each slice independently scores the winners selected by the unchanged fit/cal/hold protocol. Cross-slice Kendall τ measures agreement between the two oracle rankings over $K{=}8$ candidates. Values of $0.92/0.88$ and preserved method orderings establish stability across the collection windows.

	Code Review		Medical Preference	
Method / stat	Early	Late	Early	Late
Oracle spread (reg. range)	0.254	0.249	0.221	0.215
Bootstrap regret	0.139	0.146	0.164	0.171
Active Lab. (+50) regret	0.116	0.121	0.133	0.141
PROBEALIGN regret	**0.076**	**0.082**	**0.095**	**0.102**
PROBEALIGN Kendall τ	0.80	0.77	0.75	0.73
Cross-slice oracle τ	0.92		0.88	

Table 27: Mechanism isolation on MMLU-Pro-Core. Top-1 is the literal $k/500$ winner count. The full protocol is strongest; difficulty anchoring carries the largest measured contribution, followed by pairwise matching and failure-type transport.

Variant	Regret ↓	Kendall τ ↑	Top-1 ↑
Full PROBEALIGN	0.094 ± 0.011	0.69 ± 0.04	355/500
IRT-on-real-only	0.139 ± 0.014	0.56 ± 0.05	271/500
Moment-Match (no-IRT)	0.119 ± 0.013	0.63 ± 0.04	307/500
− Pairwise matching	0.127 ± 0.014	0.60 ± 0.05	289/500
− Difficulty anchoring	0.141 ± 0.015	0.55 ± 0.05	263/500
− Failure-type OT	0.114 ± 0.013	0.65 ± 0.04	319/500

Table 28: Mechanism isolation on IFEval-Strict with paired regret differences from full PROBEALIGN. Intervals are paired 95% bootstrap intervals; Top-1 is $k/500$. The component ordering transfers to the strict-rule oracle, with a larger contribution from failure-type transport than on MMLU-Pro-Core.

Variant	Regret ↓	Kendall τ ↑	Regret increase [95% CI]	Top-1
Full PROBEALIGN	0.112 ± 0.013	0.68 ± 0.04	—	347/500
IRT-on-real-only	0.128 ± 0.014	0.58 ± 0.05	0.016 [0.006,0.027]	292/500
Moment-Match	0.136 ± 0.014	0.58 ± 0.05	0.024 [0.013,0.036]	301/500
− Pairwise matching	0.149 ± 0.016	0.54 ± 0.05	0.037 [0.024,0.051]	266/500
− Difficulty anchoring	0.163 ± 0.017	0.49 ± 0.06	0.051 [0.036,0.067]	238/500
− Failure-type OT	0.135 ± 0.014	0.61 ± 0.05	0.023 [0.012,0.035]	304/500

map directly into the task-native score, while MMLU-Pro-Core places more of the burden on difficulty and pairwise discrimination.

A.24 THREE-FAMILY CROSS-SIMULATOR REPLICATION

We freeze Pool A, probes, prompts, splits, scorers, n=200, and the 500 paired bootstrap trials while changing the generator. The primary Mixtral-8x22B-Instruct-v0.1, Yi-1.5-34B-Chat, and Granite-3.1-8B-Instruct span three open-weight families and substantially different capacities; Granite is absent from both candidate and probe pools.

Table 29: Three-family simulator matrix. Entries report regret and Kendall τ (mean $\pm$ SE), literal Top-1 counts, and paired regret improvement over IRT-on-real-only. Generator capacity changes the margin and variance, while behavior-space calibration retains lower mean regret on every task.

Simulator	Benchmark	Regret	Kendall τ	Top-1	ΔR vs IRT [95% CI]
Mixtral-8x22B	LiveCodeBench	0.108 ± 0.012	0.71 ± 0.04	358/500	0.015 [0.006,0.025]
Mixtral-8x22B	MMLU-Pro-Core	0.094 ± 0.011	0.69 ± 0.04	355/500	0.045 [0.032,0.059]
Mixtral-8x22B	IFEval-Strict	0.112 ± 0.013	0.68 ± 0.04	347/500	0.016 [0.006,0.027]
Mixtral-8x22B	HELM-Lite	0.112 ± 0.013	0.68 ± 0.05	346/500	0.016 [0.005,0.028]
Yi-1.5-34B	LiveCodeBench	0.114±0.012	0.69±0.04	336/500	0.009 [0.001,0.018]
Yi-1.5-34B	MMLU-Pro-Core	0.101±0.012	0.67±0.04	338/500	0.038 [0.024,0.052]
Yi-1.5-34B	IFEval-Strict	0.118±0.013	0.66±0.05	329/500	0.010 [0.001,0.021]
Yi-1.5-34B	HELM-Lite	0.121±0.014	0.64±0.05	318/500	0.007 [-0.002,0.017]
Granite-3.1-8B	LiveCodeBench	0.119±0.014	0.65±0.05	322/500	0.004 [-0.006,0.014]
Granite-3.1-8B	MMLU-Pro-Core	0.112±0.014	0.63±0.05	318/500	0.027 [0.012,0.043]
Granite-3.1-8B	IFEval-Strict	0.124±0.015	0.62±0.05	316/500	0.004 [-0.007,0.016]
Granite-3.1-8B	HELM-Lite	0.126±0.016	0.59±0.06	301/500	0.002 [-0.010,0.015]

The smaller Granite generator narrows the advantage and increases uncertainty, especially on HELM-Lite, while preserving the mean ordering. The nonuniform degradation across oracles is consistent with task-specific behavior alignment rather than a fixed family offset.

A.25 COMPLETE FOUR-TASK REPLICATION ON POOL B

Pool B contains Phi-3-medium-4k-instruct, Yi-1.5-34B-Chat, Gemma-2-27B-it, Qwen2-57B-A14B-Instruct, Mistral-7B-Instruct-v0.3, Mixtral-8x7B-Instruct-v0.1, Llama-3.1-70B-Instruct, and DeepSeek-V2-Chat (B1–B8 in that order). The fixed pool is evaluated on all four task-native oracles with the original simulator, probes, $50/20/30$ split, n=200, and 500 paired bootstrap trials. The task-native orders are B7>B8>B6>B3>B4>B5>B2>B1 (LiveCodeBench), B8>B7>B3>B6>B4>B5>B1>B2 (MMLU), B7>B8>B6>B4>B3>B2>B5>B1 (IFEval), and B8>B7>B3>B6>B4>B2>B5>B1 (HELM). Their regret spreads are 0.296, 0.271, 0.255, and 0.238, respectively.

Pool B changes family composition and every task-native oracle order, yet PROBEALIGN retains the best regret, Kendall correlation, and winner count on all four tasks. The varied paired margins link the gain to each oracle's discrimination structure rather than a fixed pool offset.

A.26 PUBLIC COMPARATOR FRONTIER AND HELM-LITE SYMMETRY

The fuller HELM-Lite comparator stack and the four-task active-label frontier complete the public budget comparison. They test strong real-only estimators and measure the extra-label budget at which active acquisition reaches paired overlap with PROBEALIGN.

The full HELM-Lite stack preserves the main ordering across matching, shrinkage, and pairwise real-only estimators, completing comparator symmetry for the mixed-task oracle.

Table 30: Four-task Pool-B replication. Top-1 is $k/500$; the last column reports the paired regret improvement of PROBEALIGN over IRT-on-real-only. Pool-specific margins vary, and full calibration remains strongest on every oracle.

Benchmark	Method	Regret ↓	Kendall τ ↑	Top-1	PA ΔR vs IRT [95% CI]
LiveCodeBench	Bootstrap	0.176±0.018	0.50±0.05	251/500	—
	Active (+50)	0.141±0.015	0.61±0.05	290/500	—
	IRT-real-only	0.129±0.014	0.60±0.05	301/500	—
	PROBEALIGN	0.109±0.012	0.70±0.04	354/500	0.020 [0.009,0.032]
MMLU-Pro-Core	Bootstrap	0.163±0.017	0.49±0.05	244/500	—
	Active (+50)	0.131±0.015	0.58±0.05	283/500	—
	IRT-real-only	0.142±0.015	0.56±0.05	274/500	—
	PROBEALIGN	0.101±0.012	0.67±0.04	342/500	0.041 [0.028,0.055]
IFEval-Strict	Bootstrap	0.182±0.018	0.47±0.05	236/500	—
	Active (+50)	0.148±0.016	0.55±0.05	274/500	—
	IRT-real-only	0.134±0.014	0.58±0.05	289/500	—
	PROBEALIGN	0.116±0.013	0.67±0.04	335/500	0.018 [0.006,0.031]
HELM-Lite	Bootstrap	0.171±0.018	0.48±0.06	240/500	—
	Active (+50)	0.137±0.016	0.57±0.05	282/500	—
	IRT-real-only	0.132±0.015	0.56±0.05	287/500	—
	PROBEALIGN	0.117±0.013	0.65±0.05	332/500	0.015 [0.004,0.027]

Table 31: HELM-Lite comparator stack at n=200 over 500 bootstrap trials. The mixed-task oracle includes the full matching and real-only estimator families; PROBEALIGN has the lowest regret and highest Kendall correlation.

Method	Regret ↓	Kendall τ ↑
Bootstrap	0.169±0.017	0.49±0.05
Active Labeling (+50)	0.133±0.015	0.58±0.05
Moment-Match (no-IRT)	0.122±0.014	0.62±0.05
Abstain-Fallback	0.145±0.016	0.55±0.05
Bradley–Terry (real-only)	0.140±0.015	0.54±0.05
Plackett–Luce (real-only)	0.143±0.016	0.52±0.05
Hierarchical Bayes (real-only)	0.132±0.015	0.57±0.05
IRT-on-real-only	0.128±0.014	0.57±0.05
PROBEALIGN	**0.112** ± 0.013	**0.68** ± 0.05

Table 32: Four-task active-label frontier at n=200 over 500 paired bootstrap trials. Top-1 is $k/500$. PROBEALIGN uses the original real-label pool; Active adds 25–200 task-native labels.

Benchmark	Method	Regret ↓	Kendall τ ↑	Top-1 ↑
LiveCodeBench	Active (+25)	0.151±0.017	0.58±0.05	244/500
	Active (+50)	0.138±0.015	0.63±0.05	286/500
	Active (+100)	0.124±0.014	0.66±0.04	327/500
	Active (+200)	0.109±0.012	0.70±0.04	349/500
	PROBEALIGN	0.108 ± 0.012	0.71 ± 0.04	358/500
MMLU-Pro-Core	Active (+25)	0.142±0.016	0.55±0.05	248/500
	Active (+50)	0.127±0.014	0.60±0.05	281/500
	Active (+100)	0.115±0.013	0.64±0.04	326/500
	Active (+200)	0.096±0.011	0.68±0.04	349/500
	PROBEALIGN	0.094 ± 0.011	0.69 ± 0.04	355/500
IFEval-Strict	Active (+25)	0.159±0.017	0.52±0.06	251/500
	Active (+50)	0.144±0.015	0.56±0.05	276/500
	Active (+100)	0.129±0.014	0.61±0.05	311/500
	Active (+200)	0.113±0.013	0.67±0.04	342/500
	PROBEALIGN	0.112 ± 0.013	0.68 ± 0.04	347/500
HELM-Lite	Active (+25)	0.148±0.017	0.54±0.06	258/500
	Active (+50)	0.133±0.015	0.58±0.05	287/500
	Active (+100)	0.121±0.014	0.63±0.05	319/500
	Active (+200)	0.109±0.012	0.69±0.04	352/500
	PROBEALIGN	0.112 ± 0.013	0.68 ± 0.05	346/500

LiveCodeBench-Lite, MMLU-Pro-Core, and IFEval-Strict first overlap PROBEALIGN at 200, 200, and 200 additional labels. HELM-Lite reaches paired overlap at 100. Thus the measured substitution range is 100–200 labels, with task-specific crossover behavior rather than one imposed threshold.

A.27 PUBLIC PROBE-PANEL SENSITIVITY ON MMLU-PRO-CORE

We test probe-panel sensitivity on MMLU-Pro-Core by varying panel size and leave-one-probe-out composition. The table below keeps the candidate pool, simulator, split logic, and task-native scorer fixed, and varies only probe count or a single leave-one-probe-out omission. The result is directionally stable: the diverse fresh J=5 panel stays best or near-best, while J=4/6 and leave-one-probe-out reruns degrade only mildly.

Table 33: Public probe-panel sensitivity on MMLU-Pro-Core (n=200, 500 bootstrap resamples). Top-1 is the literal $k/500$ winner count. The diverse fresh J=5 panel remains best or near-best, and every leave-one-probe-out rerun preserves a clear margin over the real-only baselines.

Probe panel	Regret ↓	Kendall τ ↑	Top-1 ↑
Diverse, fresh, J=5 (default)	0.094 ± 0.011	0.69 ± 0.04	355/500
Diverse, fresh, J=4	0.101±0.012	0.66±0.04	338/500
Diverse, fresh, J=6	0.096±0.011	0.68±0.04	348/500
LOO drop P_1	0.098±0.012	0.67±0.04	344/500
LOO drop P_2	0.100±0.012	0.66±0.04	339/500
LOO drop P_3	0.104±0.013	0.64±0.05	329/500
LOO drop P_4	0.097±0.012	0.68±0.04	347/500
LOO drop P_5	0.099±0.012	0.67±0.04	342/500

Every leave-one-probe-out run remains above Active (+50) in Top-1, including the largest change from dropping P_3. The sweep establishes diversity and freshness as measurable operating requirements and connects directly to the routing actions in Table 20.

A.28 LARGER-POOL PUBLIC ORACLE-SHIFT REPLICATION

Pool C raises the ranking problem to $K{=}12$ candidates: Qwen2.5-3B, Llama-3.2-3B, Mistral-7B-v0.3, Gemma-2-9B, Qwen2.5-14B, Llama-3.1-8B, Gemma-2-27B, Qwen2.5-32B, Llama-3.1-70B, Qwen2.5-72B, DeepSeek-V2-Lite-Chat, and DBRX-Instruct (C1–C12 in that order). The pool spans six families, three size bands, dense and mixture-of-experts architectures, and has no overlap with the probe panel or simulator checkpoint. The task-native orders are C10>C9>C12>C8>C7>C4>C11>C6>C5>C3>C2>C1 (LiveCodeBench), C12>C9>C7>C10>C8>C4>C5>C11>C6>C3>C2>C1 (MMLU), C10>C12>C9>C8>C7>C6>C4>C5>C11>C3>C2>C1 (IFEval), and C12>C10>C9>C7>C8>C4>C11>C6>C5>C3>C2>C1 (HELM). Their regret spreads are 0.338, 0.301, 0.286, and 0.259.

Table 34: Pool-C replication at $n{=}200$ with 500 bootstrap trials. Cells report regret $\pm$ SE / Kendall $\tau\pm$ SE. Fixed priors are deterministic and therefore have no bootstrap SE. The larger pool introduces 66 candidate pairs and broader family/size diversity.

Method	LiveCodeBench-Lite	MMLU-Pro-Core	IFEval-Strict	HELM-Lite
Bootstrap	$0.198 \pm 0.018/0.46 \pm 0.05$	$0.187 \pm 0.017/0.44 \pm 0.05$	$0.204 \pm 0.019/0.42 \pm 0.06$	$0.191 \pm 0.019/0.45 \pm 0.06$
Active Labeling (+50)	$0.161 \pm 0.016/0.56 \pm 0.05$	$0.153 \pm 0.015/0.54 \pm 0.05$	$0.168 \pm 0.017/0.52 \pm 0.05$	$0.157 \pm 0.016/0.55 \pm 0.05$
IRT-on-real-only	$0.151 \pm 0.015/0.55 \pm 0.05$	$0.148 \pm 0.015/0.53 \pm 0.05$	$0.157 \pm 0.016/0.54 \pm 0.05$	$0.149 \pm 0.016/0.54 \pm 0.05$
Hierarchical Bayes (real-only)	$0.146 \pm 0.015/0.58 \pm 0.05$	$0.143 \pm 0.015/0.56 \pm 0.05$	$0.153 \pm 0.016/0.55 \pm 0.05$	$0.145 \pm 0.015/0.57 \pm 0.05$
Parameter-size prior	0.203/0.31	0.214/0.29	0.218/0.30	0.196/0.38
Model-family prior	0.211/0.36	0.205/0.34	0.209/0.35	0.202/0.35
Public-leaderboard prior	0.174/0.50	0.169/0.46	0.184/0.42	0.168/0.49
LODO global prior	0.181/0.48	0.176/0.41	0.190/0.39	0.172/0.47
PROBEALIGN	$0.121 \pm 0.013/0.67 \pm 0.04$	$0.109 \pm 0.012/0.65 \pm 0.04$	$0.126 \pm 0.014/0.64 \pm 0.05$	$0.124 \pm 0.014/0.65 \pm 0.05$

Table 35: Pool-C decision quality. Top-1 entries are literal $k/500$ counts. Paired regret improvements compare PROBEALIGN with the strongest real-only estimator, Hierarchical Bayes.

Benchmark	Bootstrap	Active (+50)	IRT-real	Hier. Bayes	PROBEALIGN / ΔR [95% CI]
LiveCodeBench	218	267	276	289	341 / 0.025 [0.013,0.038]
MMLU-Pro-Core	226	271	278	292	337 / 0.034 [0.021,0.048]
IFEval-Strict	211	254	273	281	326 / 0.027 [0.013,0.042]
HELM-Lite	223	266	276	289	331 / 0.021 [0.008,0.035]

PROBEALIGN wins all four Pool-C tasks, with the largest paired margin on MMLU-Pro-Core and the narrowest on HELM-Lite. The varied regret, correlation, and Top-1 gains track each oracle's discrimination structure across the larger pool.

A.29 ORACLE-SHIFT RECOVERY ACROSS TRIAL RANKINGS

Equation (1) evaluates the candidate pairs actively reversed by each task oracle. Table 36 reports OSR across all 500 trial rankings, separating intervention response from agreement on the full 28-pair ordering.

Table 36: Oracle-shift recovery (mean $\pm$ SE) at $n{=}200$. The denominator contains the pairs reversed relative to the public MT-Bench reference order. Higher values indicate stronger recovery of task-specific pairwise movement.

Method	LiveCodeBench (2 pairs)	MMLU (6)	IFEval (4)	HELM (6)
Reference ordering	0.00	0.00	0.00	0.00
Bootstrap	0.54 ± 0.04	0.48 ± 0.04	0.53 ± 0.04	0.51 ± 0.04
Active Labeling (+50)	0.63 ± 0.03	0.56 ± 0.04	0.60 ± 0.04	0.58 ± 0.04
IRT-on-real-only	0.66 ± 0.03	0.61 ± 0.03	0.59 ± 0.04	0.63 ± 0.03
PROBEALIGN	0.82 ± 0.02	0.81 ± 0.03	0.77 ± 0.03	0.79 ± 0.03

PROBEALIGN produces the highest OSR on every intervention. The task-dependent values track distinct oracle structures: the largest gain appears on MMLU's six-pair reshuffle, while IFEval's strict-rule reversals yield a smaller but clear improvement. This result localizes the ranking gain to the pairs that a stable global ordering predicts incorrectly.

Median rank trajectories. Table 37 provides the rank vectors used in Figure 1. Entries list ranks for M1–M8, with 1 denoting the winner. The reference vector is the public MT-Bench ordering; each task changes the oracle while preserving candidate identities.

Table 37: Oracle and recovered median-rank vectors for the public rank map. Trial-level variability is summarized by the paired uncertainty and Top-1 counts in Table 2.

Benchmark	Oracle	Bootstrap	IRT-real	PROBEALIGN
Global prior	$[4, 7, 8, 6, 5, 1, 2, 3]$	—	—	—
LiveCodeBench	$[4, 7, 8, 5, 6, 2, 1, 3]$	$[3, 7, 8, 6, 5, 2, 1, 4]$	$[4, 7, 8, 6, 5, 2, 1, 3]$	$[4, 7, 8, 5, 6, 2, 1, 3]$
MMLU-Pro-Core	$[5, 8, 7, 3, 6, 2, 4, 1]$	$[4, 7, 8, 5, 6, 2, 3, 1]$	$[5, 7, 8, 4, 6, 2, 3, 1]$	$[5, 8, 7, 3, 6, 2, 4, 1]$
IFEval-Strict	$[4, 6, 8, 5, 7, 3, 1, 2]$	$[3, 5, 8, 6, 7, 2, 1, 4]$	$[4, 6, 8, 5, 7, 2, 1, 3]$	$[4, 6, 8, 5, 7, 3, 1, 2]$
HELM-Lite	$[5, 6, 8, 4, 7, 3, 2, 1]$	$[4, 6, 8, 5, 7, 2, 1, 3]$	$[5, 6, 8, 4, 7, 2, 3, 1]$	$[5, 6, 8, 4, 7, 3, 2, 1]$

The task-native oracles move multiple candidates away from the reference order. PROBEALIGN's median order follows the LiveCodeBench Qwen/Llama reversal, the MMLU Gemma jump, and the IFEval Mixtral/Llama swap; Table 36, the Kendall means, and winner counts retain the trial-level uncertainty behind these median trajectories.