Item-Level Coreset Selection Is Implicit Mixture Optimization: A Q-Matrix Diagnostic for Code Supervised Fine-Tuning

Anonymous ACL submission

Abstract

Item-level coreset selectors fail to consistently beat uniform-random sampling on general supervised fine-tuning (SFT) pools. Recent critiques document the failure but offer no mechanism, and none evaluates code-SFT. We project each selector onto its induced source-mixture: the fraction of selected items drawn from the more algorithmic of two source corpora. On a 98,672-item pool combining Evol-Instruct and KodCode, induced source-mixture predicts downstream BCB-Hard performance better than the nominal scoring criterion. At Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base, with three random seeds per cell, neither Cherry-LLM IFD nor DEITA Pareto-dominates random pool sampling. DEITA gains 2.03 percentage points (pp) on BCB-Hard at the larger cell but loses 5.05 pp on HumanEval+ and 0.96 pp on MBPP+; only 0.6% of its selected items come from the algorithmic source. Stratified-random sampling at the diagnosed source-mixture lifts paired BCB-Hard pass@1 by 0.90 pp and 2.03 pp at the two cells, using a one-line algorithm and no scoring model; both lifts are exploratory at this seed count. CRUXEval-O is flat across all 15 SFT configurations, bounding the mechanism to synthesis-style benchmarks. Code and artifacts: https://anonymous.4open.science/r/qmatrix-codesft-9B15.

1 Introduction

Two recent critiques of instruction-tuning coreset selection, Diddee and Ippolito (2025)'s *Chasing Random* and Ivison et al. (2025a)'s large-scale comparison, show that widely adopted item-level scoring methods including Cherry-LLM IFD (Li et al., 2024), DEITA (Liu et al., 2024), and AlpaGasus (Chen et al., 2024) fail to consistently beat uniform-random sampling on general SFT pools; Ivison et al. (2025a) report that 4 of 7 such methods *drop* in downstream performance as the pool grows from 200K to 5.8M items. Both papers are empirical observations. Neither offers a mechanism for *why* item-level scoring criteria are so easily matched by random sampling, and neither evaluates the code-SFT regime where these methods are most often deployed in practice.

From scoring to projection. We propose a Q-matrix diagnostic for item-level coreset selectors. Our code-SFT pool is 98,672 items merged from Evol-Instruct (Wei et al., 2023) and KodCode (Xu et al., 2025). Each item is tagged with the knowledge components (KCs) it exercises, drawn from an 82-KC ontology that spans algorithmic primitives (e.g., dynamic programming) and software-engineering primitives (e.g., exception handling); stacking these tag vectors gives a sparse binary matrix $\mathbf{Q}$ (§3.1). From $\mathbf{Q}$, any selector that returns a subset of k items projects to a single point in (α, KCcov) space, where α is the fraction of selected items drawn from the more algorithmic source (KodCode) and KCcov is the number of distinct KCs covered. We treat this projection as the diagnostic axis for downstream code-SFT performance and evaluate Cherry-LLM IFD, DEITA, our own coverage maximizer (MaxCov), and stratified-random sampling (StratRand) at a fixed selection budget of $k = 5{,}000$ items against multi-seed uniform random sampling on BCB-Hard (Zhuo et al., 2025). Each selector lands at a different induced α because none reads source identity. BCB-Hard pass@1 tracks that α position more closely than KC coverage or the selector's nominal criterion.

Headline results. At our two code-SFT cells (Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base, $n = 3$ seeds each), neither Cherry-LLM nor DEITA Pareto-dominates random pool sampling on the code-generation channels. Cherry-LLM sits within seed noise of pool on every channel. DEITA lifts BCB-Hard by 2.03 pp at Qwen2.5-

Coder-3B-Base but loses 5.05 pp on HumanEval+ and 0.96 pp on MBPP+, a channel-trade consistent with its $\alpha \to 0.006$ position on the response surface. STRATRAND at the diagnosed α lifts paired BCB-Hard by 0.90 pp at Qwen2.5-1.5B-Base ($d_z \approx 2.0$, $t_{(2)} = 3.47$, two-sided $p \approx 0.074$, 95% CI $[-0.22, +2.02]$) and 2.03 pp at Qwen2.5-Coder-3B-Base ($d_z = 1.73$, two-sided $p \approx 0.096$, 95% CI $[-0.88, +4.93]$), matching or exceeding every item-level method we test on code-generation channels with a one-line algorithm and no scoring model. We treat these lifts as exploratory at $n = 3$ and report direction and paired effect size alongside two-sided 95% CIs throughout.

Contributions. We make three contributions to the analysis and practice of code-SFT data selection.

1. **We propose a Q-matrix diagnostic** (§3.1) that projects any item-level coreset selector to a single point in (α, KCcov) space on a two-source code-SFT pool, where α is the induced fraction of KodCode items and KCcov is per-KC coverage over an 82-KC ontology.

2. **We demonstrate that on two code-SFT cells, induced α tracks BCB-Hard performance more closely than the nominal scoring criterion or KC coverage** (§5.1–§5.2). Within our own MAXCOV family, the variant with broader KC coverage (80/82 vs. 66/82) scores *lower* on BCB-Hard because its α drifts toward pool. Source-mixture, not KC coverage, carries the signal.

3. **We show that a one-line stratified-random sampler at the diagnosed α matches or exceeds every item-level method we test** (§5.3), lifting paired BCB-Hard by 0.90 pp at Qwen2.5-1.5B-Base and 2.03 pp at Qwen2.5-Coder-3B-Base, with selection-time cost comparable to Cherry-LLM IFD scoring and no scoring model.

Scope envelope. The lever is regime-bounded: at Qwen2.5-Coder-1.5B-Base and Qwen2.5-3B-Base the diagnosed α does not surface a BCB-Hard lift (§5.4), and the trace benchmark CRUXEval-O is flat across all 15 SFT configurations we run (§5.5), locating the mechanism in synthesis-style channels rather than in execution-trace reasoning. The constructive recipe is light: a one-shot LLM-tagged content profile per source, a small α sweep, then stratified-random sampling at the chosen mixture (§6.4).

2 Related Work

Item-level selection for instruction tuning, and its critiques. Cherry-LLM's IFD score (Li et al., 2024) bootstraps a small "cherry" subset using perplexity differences. DEITA (Liu et al., 2024) scores items by complexity-×-quality with a REPR-Filter for diversity. Chen et al. (2024) (AlpaGasus) uses an LLM-as-judge rating filter; Xia et al. (2024b) (LESS) selects by gradient-based influence. Zhou et al. (2023) (LIMA) demonstrates the small-coreset regime with hand-curated data. Du et al. (2023) (MoDS) stages quality, coverage, and necessity but does not isolate source mixture as an explicit axis.

The most direct prior to this paper is Diddee and Ippolito (2025) (*Chasing Random*, NAACL Findings 2025), which compares AlpaGasus, Cherry-LLM, DEITA, and Longest-Response against uniform- and strict-random baselines on LLaMA-7B across FLAN/Dolly/Evol/Alpaca pools, finding that "selection strategies generalize poorly, often failing to consistently outperform even random baselines". Ivison et al. (2025a) extend this with a large-scale study, observing that 4 of 7 methods including IFD and Top-PPL *drop* as the pool grows from $200K$ to $5.8M$ items. Neither paper evaluates the code-SFT regime nor offers a mechanism for the failure. Xia et al. (2025) concludes diversity-over-quality at scale.

We contribute the code-SFT cell and a single-mechanism account that subsumes both directions of failure (Cherry-LLM locked near pool, DEITA pushed far from it) under the same diagnostic.

Source-mixture optimization (mostly pretraining). DoReMi (Xie et al., 2023), RegMix (Liu et al., 2025), and Sheared LLaMA (Xia et al., 2024a) optimize *pretraining* mixtures. For SFT mixture specifically, Tulu (Wang et al., 2023) reports that no single dataset or fixed combination dominates across 12 instruction sources, without sweeping ratios; the Tulu 3 release (Lambert et al., 2025) extends this with hand-curated source ratios at frontier scale but does not characterise the implicit mixture induced by item-level scorers. LASER (Mirza et al., 2025) stratifies SFT across task categories (Math/Coding/etc.) rather than source datasets. IDEAL (Ming et al., 2026) and concurrent SFT-mixing work (Li et al., 2025)

target multi-capability mixtures. None of these analyzes item-level selectors as implicit mixture shifters.

Code-SFT data papers. Magicoder/OSS-Instruct (Wei et al., 2023) ships the Evol-Instruct pool and the Magicoder prompt format we adopt. WaveCoder (Yu et al., 2024) argues task-mix is decisive for code IT. KodCode (Xu et al., 2025) provides the verified-triplet code-instruction corpus used here. None of WaveCoder, OpenCoder (Huang et al., 2025), SelfCodeAlign (Wei et al., 2024), or XFT (Ding et al., 2024) performs a controlled α sweep on BCB-Hard or another code-completion benchmark. The Qwen2.5-Coder technical report (Hui et al., 2024) provides the model family.

Q-matrices and cognitive diagnosis applied to LLMs. Q-matrices originate in cognitive diagnosis and IRT (Tatsuoka, 2005; de la Torre, 2009). We preserve the binary-incidence structure but do not estimate a cognitive-diagnosis measurement model: learner-response data, item parameters, and latent attribute inference are absent.[1] The closest application to LLM data is CDS (Zhao et al., 2025), which uses a Q-KC matrix to drive *synthesis* of weakness-targeted data. We differ by (i) using the Q-matrix to *diagnose* selectors rather than drive synthesis or selection, (ii) connecting the per-method KC footprint to source-mixture, and (iii) proposing no new selection objective: the constructive alternative is unstructured stratified sampling at the diagnosed α.

Format and pretraining as confounds. Sclar et al. (2024) demonstrates up to 76-pp accuracy spread from prompt-format choice on instruction-tuned models, motivating our format-axis treatment as a hyperparameter rather than a contribution. Allen-Zhu and Li (2025) argues pretraining shapes the selectable feature manifold; we appeal to this for the regime cells where code-pretraining saturates the lever.

3 Methodology

3.1 Source pool and Q-matrix construction

Our SFT pool is the union of Evol-Instruct from Magicoder (Wei et al., 2023) (78,258 items) and KodCode (Xu et al., 2025) (20,414 items), totalling 98,672 items.[2] The two corpora have visibly different programming-distribution profiles: KodCode emphasizes algorithmic and graph-theoretic problems, while Evol-Instruct leans toward Python idioms and library use.

Knowledge-component ontology. We define an ontology of $K = 82$ knowledge components, drawn from a hybrid of (a) LeetCode topic tags (66 KCs) covering algorithmic primitives such as `dynamic_programming`, `topological_sort`, `segment_tree`; and (b) ACM CC2020-aligned paradigm-level units (16 KCs) covering software-engineering primitives such as `exception_handling`, `regex`, `generators`, `file_io`. The ontology deliberately excludes library-specific tags and task-type indicators so that the same KC vocabulary maps cleanly to both corpora.

Annotation. Each item i is annotated with a sparse vector $\mathbf{q}_i \in \{0,1\}^{82}$ over KCs by an LLM annotator (Appendix D); mean active KCs per item is 2.42; 80/82 KCs are non-empty in the pool. Stacked rows give the sparse incidence matrix $\mathbf{Q} \in \{0,1\}^{|\mathcal{D}|\times K}$.

Per-source KC marginals. For α to be a meaningful lever, the two sources need to differ in what KCs they emphasise; if the marginals were identical, any mixture would be content-equivalent. Let $p_e, p_k \in \Delta^{K-1}$ be the column-normalized KC marginals over Evol-Instruct and KodCode. We measure $\mathrm{KL}(p_k \| p_e) = 0.431$ nats, $\mathrm{KL}(p_e \| p_k) = 0.444$ nats, and $\mathrm{JS}(p_e, p_k) = 0.086$ nats (cosine 0.937, ℓ_1 distance 0.555). The profiles are overlapping but non-identical, which is the structural precondition for α to shift the SFT manifold at all.

Mixture parameter α. For any selected subset $\mathcal{S} \subseteq \mathcal{D}$, define $\alpha(\mathcal{S}) = |\mathcal{S} \cap KodCode|/|\mathcal{S}|$. The natural pool ratio is $\alpha_{\text{pool}} = 20{,}414/98{,}672 = 0.207$. We treat α as the diagnostic axis and report it for every selector.

3.2 Selection algorithms

We compare five families of selectors at fixed budget k.

StratRand (α) draws $\lceil \alpha \cdot k \rceil$ items uniformly without replacement from KodCode and the rest uniformly from Evol-Instruct; the single hyperparameter is α. Setting $\alpha = \alpha_{\text{pool}}$ recovers uniform

[1] We use the term descriptively for the $\{0,1\}^{|\mathcal{D}|\times K}$ structure only.

[2] We exclude any LeetCode-derived items that overlap our held-out evaluations; see Appendix E.

random sampling over the pool (henceforth "Random").

MaxCov is our own coverage-aware construction, included as an instance of how an explicitly coverage-aware item-level selector lands on the response surface. It runs a submodular greedy with sqrt-saturating per-KC gain,

$$\Delta_i = \sum_{c=1}^{K} \sqrt{u_c} - \sqrt{\max(u_c - q_{i,c}, 0)}, \quad (1)$$

where u_c is unmet demand for KC c relative to per-KC quotas of $k \cdot (p_{\text{pool}})_c$. We compare two implementations of the gain term that place the sqrt-concavity at different points in the per-KC accounting: **MaxCov-v1** anchors the saturation on attained per-KC coverage and tolerates per-KC overshoot in exchange for α flexibility, while **MaxCov-v2** anchors on residual unmet demand and is our canonical concave-in-demand form. The two variants land at distinct (α, KCcov) points by construction. At $k = 5K$ on Qwen2.5-1.5B-Base, v1 covers $66/82$ KCs with $\alpha = 0.36$, while v2 covers $80/82$ KCs with $\alpha = 0.31$. The pair gives a within-method handle on coverage-mixture decoupling that we use in §5.2.

Cherry-LLM IFD (Li et al., 2024). We compute Instruction-Following Difficulty $\text{IFD}_i = \text{PPL}(\mathbf{r}_i \mid \mathbf{x}_i)/\text{PPL}(\mathbf{r}_i)$ on the un-fine-tuned base model and select the top-k by IFD.[3]

DEITA (Liu et al., 2024). Items are scored as $\text{complexity}_\phi(\mathbf{x}) \times \text{quality}_\psi(\mathbf{x}, \mathbf{r})$ via the released DEITA scorers; a REPR-Filter then enforces embedding-space diversity. We use the original released artifacts with our pool.

3.3 Q-matrix as diagnostic, not objective

A central methodological move of this paper is to use the Q-matrix as a *diagnostic projection* rather than a selection objective. For each selector π producing subset $\mathcal{S}_\pi$, we compute $\alpha(\mathcal{S}_\pi)$, the projected per-KC marginal $\bar{q}_\pi \in \Delta^{K-1}$, and the divergence from p_e and p_k. Each selector is then a single point in (α, KCcov) space, which lets us ask a falsifiable question: *do selectors with the same α but different per-KC marginals achieve different benchmark scores?* In our cells, §5.2 reports that matched-α StratRand samples and item-level selectors land within seed noise of each other, so the KC-marginal difference does not appear to carry signal beyond what α already captures.

[3] Cherry-LLM as originally proposed (Li et al., 2024) computes IFD via a pre-experienced model trained on a small bootstrap subset and admits an optional downstream filter. We use the bare IFD score with no bootstrap and no filter, which is the IFD-baseline convention used in subsequent coreset comparison work; we refer to it as "Cherry-LLM IFD" throughout for consistency with that literature.

4 Experimental Setup

Models. We use four base models from the Qwen2.5 family: Qwen2.5-Coder-3B-Base, Qwen2.5-3B-Base, Qwen2.5-Coder-1.5B-Base, Qwen2.5-1.5B-Base (Hui et al., 2024; Yang et al., 2024). All experiments use the Base variants; the Coder variants have additional code-domain pretraining over the Base variants of the same scale. None of the four has undergone instruction tuning.

Training. We fine-tune with LoRA $r = 8$, $\alpha = 16$ over the 7 attention/MLP target modules, learning rate 2×10^{-5} with cosine decay and 10% warmup, effective batch size 32, 2 epochs. Training uses bf16 mixed precision on a single 141GB H200 GPU; each $5K$-item LoRA cell completes in $\lesssim 3$ GPU-hours. Three-seed cells use seeds $\{42, 43, 44\}$.

Prompt format. We use the Magicoder `@@Instruction / @@Response` format from Wei et al. (2023) as the canonical format for all training reported in the main paper; the format ablation is treated as a hyperparameter (Appendix C).

Benchmarks. We report BCB-Hard (Zhuo et al., 2025) hard subset, $n = 148$ practical API-grounded code completions, scored by sandbox execution. Secondary code-completion benchmarks (HumanEval+, MBPP+; Liu et al., 2023) are in Appendix G. The trace-channel benchmark is CRUXEval-O output prediction (Gu et al., 2024), $n = 800$. All benchmarks are evaluated with greedy decoding under vLLM 0.6.6; pass@1 is reported.

Cell coverage. The four base models do not appear uniformly across tables; we summarise which cells each table reports. F1 (Tables 1, 2) is multi-seed at Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base ($n = 3$ each). F2 (Table 3) reports single-seed BCB-Hard (seed 42) at Qwen2.5-1.5B-Base; multi-seed α-footprint stability for all item-level selectors is in Appendix J. F3 (Tables 4, 5) is multi-seed at both Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base. The Scope grid (Table 6) is multi-seed at Qwen2.5-1.5B-Base and Qwen2.5-

Coder-3B-Base; the Qwen2.5-Coder-1.5B-Base and Qwen2.5-3B-Base rows are single-seed boundary cells (Qwen2.5-3B-Base detail in Appendix B). All training uses the Magicoder format and $k = 5K$ unless noted.

Reproducibility. Code, selection artifacts, Q-matrix, and trained adapter weights are released at https://anonymous.4open.science/r/qmatrix-codesft-9B15.

5 Results

We organise the results around three findings. F1 is the empirical observation that neither Cherry-LLM nor DEITA Pareto-dominates random pool sampling on the code-generation channels at our two code cells (§5.1): Cherry-LLM within seed noise of pool everywhere, and DEITA's BCB-Hard lift at Qwen2.5-Coder-3B-Base paired with large HumanEval+ +/MBPP+ + losses. F2 is the source-mixture account that organises both directions of failure under one variable (§5.2). F3 is the constructive consequence and its regime envelope (§5.3–§5.4). We report two-sided p-values, two-sided 95% CIs, and paired Cohen's d_z throughout. F3 results are exploratory at $n = 3$ seeds; we report direction and effect size alongside CIs that include zero from the wide $t_{(2)}$ tails. Negative or null cells, including the CRUXEval-O channel-flat result and the Qwen2.5-3B-Base null, are reported in the same place as the positive ones.

5.1 F1: item-level selectors do not Pareto-dominate pool on code-SFT, at two cells

Tables 1 and 2 report two multi-seed head-to-heads: the original Qwen2.5-1.5B-Base cell, and a cross-cell replication at Qwen2.5-Coder-3B-Base.

Cross-cell replication. At Qwen2.5-Coder-3B-Base, neither selector Pareto-dominates the matched-format Random@α_{pool} baseline (Appendix I). Cherry-LLM sits within seed noise of pool on every channel and below pool on HumanEval+ + (-1.02), MBPP+ + (-0.89), and CRUXEval-O (-0.38); its only positive is $+0.22$ on BCB-Hard, below σ. DEITA loses -5.05 on HumanEval+ + and -0.96 on MBPP+ + while landing $+2.03$ on BCB-Hard, a channel-trade consistent with its $\alpha \to 0.006$ position on the response surface (§5.2) rather than a free win. Both directions match Qwen2.5-1.5B-Base: source-blind item-level scores do not appear to recover the position deficit their induced α imposes. The general-SFT critiques of Diddee and Ippolito (2025) and Ivison et al. (2025a) describe failure to beat random; on the two code-SFT cells we test, neither item-level selector Pareto-dominates pool on the synthesis channels, with the BCB-Hard-positive direction (DEITA at Qwen2.5-Coder-3B-Base) coming paired with large HumanEval+ +/MBPP+ + losses that the α projection accounts for.

Selector	α	BCB-Hard	HumanEval+ +	MBPP+ +	$\Delta_{BCB-Hard}$
Random Pool	0.21	4.05 (1.17)	46.34 (0.62)	50.79 (0.65)	—
StratRand $_{0.50}$	0.50	**4.96 (0.39)**	**49.19 (0.35)**	**54.67 (0.81)**	**+0.90**
Cherry-LLM IFD	0.30	**2.93 (1.70)**	**47.60 (0.60)**	**52.47 (0.51)**	**−1.13**
DEITA	0.006	**3.38 (0.68)**	47.80 (0.35)	**49.63 (0.40)**	**−0.68**

Table 1: **F1 at Qwen2.5-1.5B-Base + Magicoder, $k = 5K$, 3 seeds each.** Mean (σ) pass@1. Cherry-LLM and DEITA both land below Random Pool on BCB-Hard; the mixed HumanEval+ +/MBPP+ + picture is consistent with them landing closer to pool-default on those channels. The StratRand $_{0.50}$ reference row shows that explicit α control at the diagnosed mixture lifts every code-generation channel above pool at the same budget (§5.3). Paired $\Delta_{BCB-Hard}$ effect sizes vs. Random are sub-σ at this cell alone (Cherry-LLM $d_z = -0.52$, DEITA $d_z = -0.38$; StratRand $d_z \approx +2.0$; full effect-size table in Appendix K); the F1 claim rests on directional consistency across this cell and Qwen2.5-Coder-3B-Base (Table 2).

Selector	BCB-Hard	HumanEval+ +	MBPP+ +	$\Delta_{BCB-Hard}$
Random Pool	8.11 (0.00)	72.15 (0.29)	65.96 (0.33)	—
StratRand $_{0.50}$	**10.14 (1.17)**	**73.80 (1.04)**	**67.10 (0.66)**	**+2.03**
Cherry-LLM IFD	8.33 (1.41)	71.13 (0.96)	65.07 (0.25)	+0.22
DEITA	10.14 (0.68)	67.10 (0.00)	65.00 (0.17)	+2.03

Table 2: **F1 cross-cell at Qwen2.5-Coder-3B-Base + Magicoder, $k = 5K$, 3 seeds each.** Mean (σ) pass@1. Both Random Pool and the diagnosed StratRand($\alpha = 0.50$) are shown as reference columns. Cherry-LLM lands below pool on HumanEval+ +/MBPP+ + and is within seed noise on BCB-Hard; DEITA lifts BCB-Hard by $+2.03$ but loses -5.05 on HumanEval+ + and -0.96 on MBPP+ +. StratRand at the diagnosed mixture is the only selector that lifts every code-generation channel relative to pool. Paired effect sizes vs. Random Pool are in Appendix K; DEITA's $+BCB-Hard/-HumanEval+$ + channel-trade is large in both directions ($d_z = +3.00$ on BCB-Hard, $d_z = -14.4$ on HumanEval+ +; two-sided $p \leq 0.04$).

5.2 F2: the mechanism is uncontrolled source-mixture shift

Table 3 reports the Q-matrix decomposition of each selector at $k = 5K$ on Qwen2.5-1.5B-Base. The right two columns project each selector to a point in (α, KCcov) space.

Selector	α	KCcov	BCB-Hard
Random Pool	0.21	74/82	4.05
StratRand ($\alpha=0.21$)	0.21	71/82	4.05
Cherry-LLM IFD ($k=5K$)	0.30	77/82	2.93
Cherry-LLM IFD ($k=10K$)	0.28	79/82	2.03
DEITA ($k=5K$)	0.006	73/82	3.38
MaxCov-v1	0.36	66/82	4.05
MaxCov-v2 (canonical)	0.31	80/82	2.70
StratRand ($\alpha=0.36$)	0.36	71/82	4.05

Table 3: **F2: every selector projects to an α, and benchmark score tracks α position more closely than KC coverage.** Qwen2.5-1.5B-Base + Magicoder. Single-seed (seed 42) for Cherry-LLM, DEITA, MaxCov; matched single-seed StratRand reference points. Multi-seed Cherry-LLM/DEITA in Table 1; multi-seed α footprint stability for all four item-level selectors in Appendix J (Cherry-LLM/DEITA: $\sigma_\alpha=0$; MaxCov-v2: $\sigma_\alpha=0.007$; MaxCov-v1: $\sigma_\alpha=0.060$). The two MaxCov variants place the sqrt-concavity of the gain term at different points in the per-KC accounting (§3.2) and land at distinct (α, KCcov) points by construction; we use the pair as a within-method handle on coverage-mixture decoupling.

Selectors silently choose α. None of Cherry-LLM, DEITA, or MaxCov reads source identity, yet each lands on a sharply different α. Cherry-LLM locks near the pool ratio ($\alpha\approx0.30$ at $k=5K$ and $k=10K$) and BCB-Hard does not move with budget. DEITA's REPR-Filter aggressively prunes the denser KodCode embedding cluster, pushing $\alpha \to 0.006$. MaxCov-v1 lands at $\alpha \approx 0.31$ on average (seed-42 realization 0.36, $\sigma_\alpha=0.060$; Appendix J) as a side-effect of per-KC quotas.

Within-method coverage-mixture decoupling. Moving from MaxCov-v1 to MaxCov-v2 raises KC coverage from 66/82 to 80/82 while α moves from 0.36 to 0.31; BCB-Hard *drops* from 4.05 to 2.70. A StratRand sample at the same $\alpha=0.36$ but with random per-KC mass matches v1 at 4.05, so at matched α the item-level ranking does not add measurable lift. In this within-method comparison the variant that covers more KCs but lands further from the favourable α scores lower, which is the pattern we would expect if α position rather than KC coverage is the dimension carrying signal.

Cross-channel signature of α-position. The α position also predicts per-channel divergence. At Qwen2.5-Coder-3B-Base, DEITA's $\alpha = 0.006$ leaves a near-pure Evol-Instruct diet: HumanEval+ + and MBPP+ + (closer to Evol-Instruct style) regress, while BCB-Hard lifts +2.03 pp from the small surviving KodCode-style residual at higher complexity. Cherry-LLM's $\alpha = 0.30$ sits close enough to $\alpha_{\text{pool}}=0.21$ that all four channels track pool to within σ.

α	s42	s43	s44	Mean	σ
0.21 (pool)	3.38	3.38	5.41	4.05	1.17
0.40	5.41	3.38	6.08	**4.96**	1.41
0.50	4.73	4.73	5.41	**4.96**	0.39
0.60	4.05	4.73	6.08	**4.96**	1.03

Table 4: **F3a: StratRand sweep at Qwen2.5-1.5B-Base, 3 seeds $\times$ 4 α.** BCB-Hard pass@1. Interior $\alpha\in[0.40, 0.60]$ forms a flat plateau at 4.96, paired +0.90 pp vs. pool ($d_z\approx2.0$; $t_{(2)}=3.47$, two-sided $p\approx0.074$, 95% CI $[-0.22, +2.02]$; exploratory).

Selector	BCB-Hard	HumanEval+ +	MBPP+ +	CRUXEval-O
StratRand $_{0.50}$	**10.14 (1.17)**	**73.80 (1.04)**	**67.10 (0.66)**	39.50 (0.21)
Cherry-LLM IFD	8.33 (1.41)	71.13 (0.96)	65.07 (0.25)	39.04 (0.07)
DEITA	10.14 (0.68)	67.10 (0.00)	65.00 (0.17)	39.46 (0.19)
$\Delta_{\text{StratRand}-Cherry-LLM}$	+1.80	+2.67	+2.03	+0.46
$\Delta_{\text{StratRand}-DEITA}$	+0.00	+6.70	+2.10	+0.04

Table 5: **F3 cross-cell three-method head-to-head at Qwen2.5-Coder-3B-Base + Magicoder, $k = 5K$, 3 seeds each.** Mean (σ) pass@1. StratRand at the diagnosed mixture matches or exceeds both item-level selectors on every code-generation channel; CRUXEval-O (trace channel) is flat across selectors at $\sim$39.0, range < 0.5 pp (§5.5). A matched-format `Random@`$\alpha_{\texttt{pool}}$ row at this cell is reported in Appendix I. Paired effect sizes for the $\Delta_{\text{StratRand}-Cherry-LLM}$ and $\Delta_{\text{StratRand}-DEITA}$ rows are in Appendix K (most are large; MBPP+ + and DEITA-HumanEval+ + comparisons reach two-sided $p \leq 0.05$). BCB-Hard $\Delta_{\text{StratRand}-DEITA}=0$ exactly; the F3 BCB-Hard lift at this cell is established against the matched-format Random pool baseline (App. I, $d_z=+1.73$) rather than against DEITA.

5.3 F3: direct α control matches or exceeds item-level selectors

If α position is the dimension carrying signal (§5.2), then a selector that sets α directly should reach what item-level scorers reach indirectly. StratRand is precisely that selector: one hyperparameter, no scoring model, no IFD-style compute, and a sampling step that runs in seconds. Tables 4 and 5 report the multi-seed StratRand sweep at Qwen2.5-1.5B-Base and the cross-cell confirmation at Qwen2.5-Coder-3B-Base.

Interior plateau, paired lift. Three interior α values produce identical multi-seed means at 4.96 vs. pool 4.05. Per-seed paired lifts $\{+1.35, +0.90, +0.45\}$ give mean +0.90, paired

Cell	Pool	Best α	Δ
Qwen2.5-1.5B-Base + Mag	4.05	4.96 @ 0.50	$+\mathbf{0.90}^{\dagger}$
Qwen2.5-Coder-1.5B-Base + Mag	6.08	6.08 @ $\{0.40, 0.50\}$	0.00
Qwen2.5-Coder-3B-Base + Mag	8.11	10.14 @ 0.50	$+2.03^{\ddagger}$
Qwen2.5-3B-Base + Mag	—	null (LoRA & FFT)	≈ 0

Table 6: **Regime grid.** BCB-Hard pass@1. †paired-seed $d_z \approx 2.0$, two-sided $p \approx 0.074$, 95% CI $[-0.22, +2.02]$ (exploratory). ‡Qwen2.5-Coder-3B-Base pool is matched-format Random@α_{pool}, 3-seed; pool $\sigma_{BCB-Hard} = 0$ (same 12/148 pass count across distinct generations); paired $d_z = 1.73$, two-sided $p \approx 0.096$, 95% CI $[-0.88, +4.93]$. Full pool baseline at this cell in Appendix I. The Qwen2.5-3B-Base cell is in Appendix B.

SEM 0.26, $d_z \approx 2.0$ (very large), $t_{(2)} = 3.47$, two-sided $p \approx 0.074$, 95% CI $[-0.22, +2.02]$. The single-seed view shows an apparent peak at $\alpha = 0.50$ (Appendix A); multi-seed reveals it as a broad plateau. The CI crosses zero from the $n = 3$ wide-tailed $t_{(2)}$ rather than from the effect size; F3 is exploratory.

Cross-cell three-method head-to-head. At Qwen2.5-Coder-3B-Base, StratRand($\alpha = 0.50$) matches DEITA on BCB-Hard and exceeds both selectors on HumanEval+ + ($\Delta = +2.67/+6.70$) and MBPP+ + ($\Delta = +2.03/+2.10$; Table 5). Deltas are large relative to within-method σ (≤ 1.41); on HumanEval+ +, StratRand sits roughly 6σ above DEITA. CRUXEval-O is flat at ~ 39.0 (range 0.46 pp), per §5.5.

5.4 Scope: where the lever fires

Table 6 summarises the four model cells across which we run StratRand (or its MaxCov equivalent at the same α).

The lever moves BCB-Hard at two cells (Qwen2.5-1.5B-Base, Qwen2.5-Coder-3B-Base) and is quiet at two others. We keep both quiet cells in the grid because each is informative about *when* the recipe applies: Qwen2.5-Coder-1.5B-Base (the code-pretrained 1.5B) tests the saturation upper bound, and Qwen2.5-3B-Base (3B Base without code pretraining) tests the capacity-without-prior corner. At Qwen2.5-Coder-1.5B-Base the interior α matches pool to three decimals at $\alpha \in \{0.40, 0.50\}$ and *drops* by -1.13 pp at $\alpha = 0.60$, consistent with the code-pretrained model already having absorbed the KodCode-style mass. At Qwen2.5-3B-Base (a 3B Base without code pretraining), the lever does not surface under either LoRA or full fine-tuning (Appendix B). We read these two quiet cells as a regime envelope: the recipe applies where the SFT mixture has slack to move the model, and we offer a two-axis hypothesis for that envelope in §6.2.

5.5 Channel specificity

CRUXEval-O output-prediction (Appendix H, Table 8) is flat across all 15 SFT configurations: within each base size the selector-induced range is < 1 pp (within binomial noise); the inter-base gap is ~ 10 pp (size effect). The same lever that produces multi-seed lifts on BCB-Hard/HumanEval+ +/MBPP+ + does not move CRUXEval-O. The mechanism is bounded to synthesis-style channels; §6.3 discusses the implication.

6 Discussion

6.1 Why item-level methods fail on code-SFT

The mixture parameter α is a property of the *selected set*, not of any item. Source-blind item-level scores can only move α as a side-effect of item-content preferences. Cherry-LLM's IFD correlates with surface features (length, token rarity) which are roughly source-agnostic in our pool, so the top-k tail is approximately uniform in source and α stays near α_{pool} at both $k = 5K$ and $10K$. DEITA's REPR-Filter aggressively prunes the denser KodCode embedding cluster, pushing $\alpha \to 0.006$, the inverse of what moves α into the favourable interior. A source-aware variant of any per-item scorer is mechanically equivalent to StratRand at the diagnosed α in the limit, and the item-level component adds no further BCB-Hard win in any cell we test.

Relation to LESS and other gradient methods. Xia et al. (2024b) scores items by gradient-similarity to a validation set. Our framework predicts: with a validation set whose α is far from the favourable region, LESS locks at the validation α and underperforms StratRand inside the region; with a BCB-Hard-style validation set, LESS should locate the lever because gradient-similarity is benchmark-aware. This is the only known route by which an item-level method implicitly reads source.

Relation to prior critiques. Where Diddee and Ippolito (2025) and Ivison et al. (2025a) observe *that* item-level methods fail to beat random on general SFT, we offer a candidate mechanism that organises both directions of failure under a single

variable (the induced α position) and report a non-Pareto-dominance pattern on the code-SFT synthesis channels at the two base cells we test: Cherry-LLM within seed noise of pool, DEITA trading a BCB-Hard gain at Qwen2.5-Coder-3B-Base for HumanEval+ +/MBPP+ + losses.

6.2 Regime: when does the lever have slack?

Table 6 shows a diagonal pattern that admits a two-axis reading. **Pretrain-coverage slack**: at Qwen2.5-1.5B-Base, the base has limited code pretraining and KodCode adds new algorithmic coverage; lifting α from 0.21 to 0.50 shifts the SFT manifold toward BCB-Hard-rewarded distributions. At Qwen2.5-Coder-1.5B-Base, the model is already code-pretrained on a related distribution: KodCode is largely redundant and $\alpha = 0.60$ *hurts* by -1.13 pp. **Capacity headroom**: at Qwen2.5-Coder-3B-Base, a $5K$ budget leaves capacity to absorb harder KodCode-dominant items; at Qwen2.5-3B-Base, capacity exists but the code-pretraining prior does not, and a $5K$ KodCode-heavy budget is insufficient under both LoRA and FFT (Appendix B). We report this two-axis reading as hypothesis, not theory: the four cells are consistent with it but do not falsify alternatives.

6.3 Why CRUXEval-O does not move

BCB-Hard, HumanEval+ +, and MBPP+ + probe synthesis (write a function that passes tests); CRUXEval-O probes tracing (predict output given function and input). SFT on Evol-Instruct $\cup$ KodCode shifts $p(\mathbf{r} \mid \mathbf{x})$ over response style, but $p(\text{trace} \mid \text{code})$ is set during pretraining. Our mixture lever moves the former, not the latter; using SFT mixture optimization to improve trace benchmarks is on the wrong axis.

6.4 Practical implications

At the 1.5B–3B scale, the recommended intervention is: (i) tag each source with a coarse content profile ($\sim$ 20–80 KCs; a single-shot LLM tagger suffices); (ii) confirm sources are distinguishable (JS $\gtrsim 0.05$ nats); (iii) run a small StratRand (α) sweep at low budget; (iv) train at the discovered α with stratified-random sampling. Selection-time cost is comparable to running Cherry-LLM IFD on the pool, and item-level scoring offers no overhead-justified gain in any cell we test.

7 Conclusion

Item-level coreset selection on code-SFT at the 1.5B–3B scale appears, in our cells, to work or fail through implicit source-mixture shift more than through item-level quality or KC coverage. A Q-matrix decomposition over an 82-KC ontology makes the projection explicit: every selector maps to an induced α as a side-effect of source-blind scoring, and in our experiments benchmark performance tracks α position more closely than the selector's nominal criterion. Neither Cherry-LLM nor DEITA Pareto-dominates random pool sampling at Qwen2.5-1.5B-Base or Qwen2.5-Coder-3B-Base: Cherry-LLM stays within seed noise of pool on every code-generation channel, and DEITA's BCB-Hard lift at Qwen2.5-Coder-3B-Base ($+2.03$ pp) is paired with losses of -5.05 pp on HumanEval+ + and -0.96 pp on MBPP+ +. Within our MaxCov family, the variant with broader KC coverage scores lower on the benchmark because its α drifts away from the favourable region. Stratified-random sampling at the diagnosed α matches or exceeds every item-level method we test, lifting BCB-Hard by paired $+0.90$ pp at Qwen2.5-1.5B-Base and reaching 10.14 at Qwen2.5-Coder-3B-Base. The lever is regime-bounded by the cells we report; the diagnostic and the recipe transfer to any pool whose source content profiles are distinguishable.

Limitations

Multi-seed coverage. F1 is multi-seed at two cells ($n = 3$ at Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base, Tables 1, 2). The F3 StratRand sweep is multi-seed at Qwen2.5-1.5B-Base and the cross-cell F3 head-to-head at Qwen2.5-Coder-3B-Base is $n = 3$ per selector across three benchmarks (Table 5). The F2 multi-seed α-footprint table (Appendix J) confirms each selector lands at a characteristic α region; the within-method MaxCov v1-vs-v2 BCB-Hard comparison is single-seed. The seed-42 MaxCov-v1 sample sits in the upper tail of its $\sigma_\alpha = 0.060$ distribution, so we prefer the matched-α StratRand control in Table 3 as the cleaner decoupling of α from item-level signal.

Pool reference at Qwen2.5-Coder-3B-Base. The matched-format Random@α_{pool} baseline at Qwen2.5-Coder-3B-Base (Appendix I) gives the same direction as the StratRand reference column in Tables 2 and 5: Cherry-LLM within σ of

pool, DEITA Pareto-dominated by pool on the synthesis channels, and STRATRAND the only selector lifting all three.

Two SFT sources and pool scale. The mixture analysis is two-domain (Evol-Instruct ∪ KodCode); the one-parameter cleanness of α would degrade on higher-dimensional simplexes. Our pool is $98K$ items vs. 1–$10M$ in commercial code-SFT; we expect the mechanism (item-level $\rightarrow$ uncontrolled α shift) to persist, since it is a property of source heterogeneity rather than pool size, but the diagnosed $\alpha^\star$ may shift.

Method coverage. We evaluate Cherry-LLM, DEITA, MAXCOV, and STRATRAND. LESS (Xia et al., 2024b), RDS+ (Ivison et al., 2025b), TAG-COS, and AlpaGasus (Chen et al., 2024) are not in this revision; §6.1 states the framework's falsifiable prediction for LESS.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2025. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *The Thirteenth International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Jimmy de la Torre. 2009. Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1):115–130.

Harshita Diddee and Daphne Ippolito. 2025. Chasing random: Instruction selection strategies fail to generalize. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1943–1957, Albuquerque, New Mexico. Association for Computational Linguistics.

Yifeng Ding, Jiawei Liu, Yuxiang Wei, and Lingming Zhang. 2024. XFT: Unlocking the power of code instruction tuning by simply merging upcycled mixture-of-experts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12941–12955, Bangkok, Thailand. Association for Computational Linguistics.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *ArXiv*, abs/2311.15653.

Alex Gu, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. 2024. CRUXEval: A benchmark for code reasoning, understanding and execution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16568–16621. PMLR.

Siming Huang, Tianhao Cheng, Jason Klein Liu, Weidi Xu, Jiaran Hao, Liuyihan Song, Yang Xu, Jian Yang, Jiaheng Liu, Chenchen Zhang, Linzheng Chai, Ruifeng Yuan, Xianzhen Luo, Qiufeng Wang, Yuan-Tao Fan, Qingfu Zhu, Zhaoxiang Zhang, Yang Gao, Jie Fu, and 7 others. 2025. OpenCoder: The open cookbook for top-tier code large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33167–33193, Vienna, Austria. Association for Computational Linguistics.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. *Preprint*, arXiv:2409.12186.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025a. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025b. Large-scale data selection for instruction tuning. *Preprint*, arXiv:2503.01807.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Yuan Li, Zhengzhong Liu, and Eric P. Xing. 2025. Data mixing optimization for supervised fine-tuning of large language models. In *Forty-second International Conference on Machine Learning*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of

large language models for code generation. *Preprint*, arXiv:2305.01210.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2025. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Chenlin Ming, Chendi Qu, Qizhi Pei, Zhuoshi Pan, Yu Li, Xiaoming Duan, Lijun Wu, and Conghui He. 2026. IDEAL: Data equilibrium adaptation for multi-capability language model alignment. In *The Fourteenth International Conference on Learning Representations*.

Paramita Mirza, Lucas Weber, and Fabian Küch. 2025. Laser: Stratified selective sampling for instruction tuning with dedicated scoring strategy. *Preprint*, arXiv:2505.22157.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Kikumi Tatsuoka. 2005. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20:345 – 354.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Preprint*, arXiv:2306.04751.

Yuxiang Wei, Federico Cassano, Jiawei Liu, Yifeng Ding, Naman Jain, Zachary Mueller, Harm de Vries, Leandro Von Werra, Arjun Guha, and LINGMING ZHANG. 2024. Selfcodealign: Self-alignment for code generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Empowering code generation with oss-instruct. In *International Conference on Machine Learning*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024a. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. LESS: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. 2025. Rethinking data selection at scale: Random selection is almost all you need. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2698–2711, Suzhou, China. Association for Computational Linguistics.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. 2025. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. *Preprint*, arXiv:2503.02951.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 24 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhaojian Yu, Xin Zhang, Ning Shang, Yangyu Huang, Can Xu, Yishujie Zhao, Wenxiang Hu, and Qiufeng Yin. 2024. WaveCoder: Widespread and versatile enhancement for code large language models by instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5140–5153, Bangkok, Thailand. Association for Computational Linguistics.

Haokun Zhao, Jinyi Han, Jiaqing Liang, Yanghua Xiao, Xiaojun Meng, and Jiansheng Wei. 2025. CDS: Data synthesis method guided by cognitive diagnosis theory. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8370–8393, Vienna, Austria. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, and 14 others. 2025.

Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *The Thirteenth International Conference on Learning Representations*.

A Single-seed five-point sweep at Qwen2.5-1.5B-Base

Selector	α	KCcov	HumanEval+	HumanEval+ +	MBPP	MBPP+ +	BCB-Hard
Random Pool	0.21	74/82	50.61	46.34	60.32	53.44	3.38
StratRand $\alpha=0.21$	0.21	71/82	52.44	47.56	60.32	53.17	4.05
StratRand $\alpha=0.30$	0.30	71/82	51.22	45.12	61.11	53.97	4.73
StratRand $\alpha=0.45$	0.45	72/82	55.49	48.78	61.90	55.29	**5.41**
StratRand $\alpha=0.55$	0.55	72/82	54.88	49.39	62.17	55.56	**5.41**
StratRand $\alpha=0.70$	0.70	71/82	56.71	50.00	62.70	55.03	3.38
MaxCov-v1	0.36	66/82	54.27	48.17	59.26	51.32	4.05
MaxCov-v2 (canonical)	0.31	80/82	53.05	47.56	59.79	50.79	2.70
Cherry-LLM IFD	0.30	77/82	51.22	46.95	59.79	52.65	2.03
Cherry-LLM IFD ($k=10K$)	0.28	79/82	53.66	48.17	60.32	52.65	2.03
DEITA	0.006	73/82	52.44	47.56	60.32	51.32	3.38

Table 7: **Full single-seed sweep at Qwen2.5-1.5B-Base + Magicoder, $k=5K$, seed 42.** The single-seed view shows an apparently concave BCB-Hard curve with peak at $\alpha \in [0.45, 0.55]$ delivering $+2.03$ pp over Random. Multi-seed (Table 4) corrects this to a flat interior plateau.

B Qwen2.5-3B-Base null cell

At Qwen2.5-3B-Base (3B Base, no code pretraining) + Magicoder, the lever does not surface. Under LoRA $r=8$ at $k=5K$, StratRand at $\alpha=0.50$ does not lift BCB-Hard above pool. A full-fine-tuning (FFT) replication at the same budget produces the same null. We report this as a single-cell scope point: at this capacity-without-code-prior cell, $5K$ KodCode-heavy items is insufficient to move BCB-Hard under either parameter-efficient or full SFT.

C Format ablation

We treat prompt format as a hyperparameter (Sclar et al., 2024). Magicoder `@@ Instruction` / `@@ Response` (Wei et al., 2023) is used throughout the main paper. The alternative default `### Problem:` / `### Solution:` format produces qualitatively similar mixture-vs-BCB-Hard response surfaces at Qwen2.5-1.5B-Base, with absolute pass rates shifted downward by ~ 2 pp; the F1 negative direction for Cherry-LLM/DEITA is preserved across formats.

D Annotation details

The 82-KC ontology is in `qmatrix/ontology/kc_ontology.yaml`. Each item is annotated by an LLM tagger (`annotator_v06`) with a primary KC and zero or more secondary KCs (mean 2.42 active KCs/item). Annotation calibration was checked against a 200-item human-tagged sample; per-KC F1 is reported in Appendix F.

E Contamination check

We exclude any LeetCode-derived items in KodCode whose problem statements overlap our held-out LeetCode-style evaluations. See `qmatrix/contamination/` for the exclusion list.

F Per-KC F1 against human

Annotation F1 averages 0.78 across 82 KCs on a 200-item human-tagged sample. Per-KC F1 distribution and the 20% lower-confidence KCs are listed in the artifact release.

G Additional evaluations

HumanEval+, MBPP++ pass@1 for the F3 Qwen2.5-1.5B-Base sweep are included in Table 7. Multi-seed HumanEval+ +/MBPP+ + at the F1 Qwen2.5-1.5B-Base and Qwen2.5-Coder-3B-Base cells are in the main tables (Tables 1, 2, 5).

H CRUXEval-O channel-flat data

Cell + Selector	Mean (σ)	Range
Qwen2.5-1.5B-Base Cherry-LLM IFD	28.67 (0.38)	28.25–29.00
Qwen2.5-1.5B-Base DEITA	28.79 (0.08)	28.75–28.88
Qwen2.5-Coder-3B-Base StratRand $_{0.50}$	39.50 (0.21)	39.25–39.62
Qwen2.5-Coder-3B-Base Cherry-LLM IFD	39.04 (0.07)	39.00–39.12
Qwen2.5-Coder-3B-Base DEITA	39.46 (0.19)	39.25–39.62

Table 8: **CRUXEval-O is flat across selectors at both base cells.** pass@1, $n=800$, 3 seeds per row. Within each base size, the selector-induced range is < 1 pp (within binomial noise). The lever moves synthesis benchmarks but not the execution-trace channel (§5.5).

I Matched-format pool baseline at Qwen2.5-Coder-3B-Base

Table 9 reports a multi-seed (3-seed) Random@α_{pool} baseline at Qwen2.5-Coder-3B-Base + Magicoder + $k = 5K$. This is the matched-format Random reference for the F1 cross-cell and F3 cross-cell claims, complementing StratRand ($\alpha = 0.50$) as a second reference column.

Selector	BCB-Hard	HumanEval+ +	MBPP+ +	CRUXEval-O
Random@α_{pool}	8.11 (0.00)	72.15 (0.29)	65.96 (0.33)	39.42 (0.15)
Cherry-LLM IFD	8.33 (1.41)	71.13 (0.96)	65.07 (0.25)	39.04 (0.07)
DEITA	10.14 (0.68)	67.10 (0.00)	65.00 (0.17)	39.46 (0.19)
StratRand $_{0.50}$	**10.14** (**1.17**)	**73.80** (**1.04**)	**67.10** (**0.66**)	39.50 (0.21)
$\Delta_{Cherry-LLM-\text{pool}}$	+0.22	−1.02	−0.89	−0.38
$\Delta_{DEITA-\text{pool}}$	+2.03	−5.05	−0.96	+0.04
$\Delta_{\text{StratRand}-\text{pool}}$	**+2.03**	**+1.65**	**+1.14**	+0.08

Table 9: **Matched-format Random@α_{pool} at Qwen2.5-Coder-3B-Base, with cross-method Δ.** 3 seeds, $k = 5K$, Magicoder format, LoRA $r = 8$. Cherry-LLM is within seed noise of pool across all four channels (no positive lift exceeds σ). DEITA is Pareto-dominated by pool on the synthesis-style channels (-5.05 HumanEval+ +, -0.96 MBPP+ +, $+0.04$ CRUXEval-O) and lifts BCB-Hard $+2.03$ pp; the cross-channel signature follows the $\alpha = 0.006$ position on the response surface (§5.2). StratRand at the diagnosed mixture is the only selector that lifts *every* code-generation channel relative to pool ($+1.1$ to $+2.0$ pp). CRUXEval-O is flat to within 0.5 pp across all four selectors (channel-null, §5.5). Pool $\sigma_{BCB-Hard} = 0$ reflects all three pool seeds passing the same $12/148$ BCB-Hard tasks despite distinct generations (verified by per-seed SHA256 of generated samples).

J Multi-seed stability of induced α footprints

A natural concern about F2 is whether each selector's induced α position is itself a stable property of the selector or a seed-dependent artifact. We rerun the selection step at three seeds $\{42, 43, 44\}$ at $k = 5000$ on the same 98,672-item pool and report the spread.

Implication for F2. The multi-seed footprint confirms that each item-level selector has a characteristic α region rather than drifting across seeds. Cherry-LLM/DEITA's deterministic selection means their α is a population-level property, not a seed-realization. MaxCov-v2's tight $\sigma_\alpha = 0.007$ indicates that the canonical concave-in-demand form not only raises KC coverage but also stabilises the mixture-induction behaviour: the v2 algorithm reliably lands at $\alpha \approx 0.30$ on this pool. MaxCov-v1's wider $\sigma_\alpha = 0.060$ reflects the higher tie-break sensitivity of the coverage-anchored gain. The within-method comparison in §5.2 therefore captures two coupled changes (KC coverage rises and α moves toward pool), and v1's wider α band means the single-seed v1 BCB-Hard value should be read as one sample of a distribution rather than a sharp point estimate.

Selector	α (mean $\pm\sigma$)	KCcov (mean $\pm\sigma$)
Random Pool	0.2015 (static)	—
Cherry-LLM IFD	0.3068 ± 0.0000	77 ± 0
DEITA	0.0064 ± 0.0000	73 ± 0
MaxCov-v1	0.307 ± 0.060	62.3 ± 2.3
MaxCov-v2	0.300 ± 0.007	80 ± 0

Table 10: **Multi-seed α and KCcov footprints, $k = 5000$ on the 98,672-item pool, no-LCD convention.** Cherry-LLM/DEITA selection is deterministic in our implementation (no rng or tie-breaking randomness), so $\sigma = 0$ trivially; MaxCov uses a seeded numpy RNG. The induced α for MaxCov-v2 is tightly concentrated ($\sigma = 0.007$, KCcov invariant at 80); v1's coverage-anchored gain is markedly noisier ($\sigma_\alpha = 0.060$, KCcov ± 2.3), as the coverage-saturating form is more sensitive to greedy tie-breaks. The seed-42 MaxCov-v1 sample in Table 3 ($\alpha = 0.36$) sits at the upper tail of the v1 distribution and is the specific realization used to train the as-reported adapter. The qualitative footprint claim holds across all three seeds: v1 lands closer to the pool end of the response surface and v2 lands closer to the diagnosed interior.

K Paired effect sizes for all main-table Δ comparisons

Table 11 compiles paired Cohen's d_z, paired SEM, two-sided $t_{(2)}$, two-sided p, and 95% two-sided confidence intervals for every Δ reported in the main tables. We treat all comparisons as exploratory at $n = 3$ seeds (see analysis-discipline statement at the start of §5). d_z is invariant to the sign convention; p-values and CIs are two-sided.

Cell / channel	Comparison	Δ	SEM	d_z	p_{two}	95% CI
Qwen2.5-1.5B-Base BCB-Hard	Cherry-LLM − Pool	−1.13	1.25	−0.52	0.464	[−6.52, +4.27]
Qwen2.5-1.5B-Base BCB-Hard	DEITA − Pool	−0.68	1.03	−0.38	0.580	[−5.12, +3.76]
Qwen2.5-1.5B-Base BCB-Hard	StratRand plateau − Pool	+0.90	0.26	+2.00	0.074	[−0.22, +2.02]
Qwen2.5-Coder-3B-Base BCB-Hard	Cherry-LLM − Pool	+0.22	0.81	+0.16	0.809	[−3.27, +3.72]
Qwen2.5-Coder-3B-Base BCB-Hard	DEITA − Pool	+2.03	0.39	+3.00	0.035	[+0.35, +3.70]
Qwen2.5-Coder-3B-Base BCB-Hard	StratRand$_{0.50}$ − Pool	+2.03	0.68	+1.73	0.096	[−0.88, +4.93]
Qwen2.5-Coder-3B-Base HumanEval+ +	Cherry-LLM − Pool	−1.02	0.42	−1.42	0.134	[−2.81, +0.77]
Qwen2.5-Coder-3B-Base HumanEval+ +	DEITA − Pool	−5.05	0.20	−14.35	0.002	[−5.93, −4.18]
Qwen2.5-Coder-3B-Base HumanEval+ +	StratRand$_{0.50}$ − Pool	+1.65	0.72	+1.31	0.151	[−1.47, +4.76]
Qwen2.5-Coder-3B-Base MBPP+ +	Cherry-LLM − Pool	−0.89	0.10	−4.96	0.013	[−1.34, −0.45]
Qwen2.5-Coder-3B-Base MBPP+ +	DEITA − Pool	−0.96	0.17	−3.25	0.030	[−1.69, −0.23]
Qwen2.5-Coder-3B-Base MBPP+ +	StratRand$_{0.50}$ − Pool	+1.14	0.35	+1.91	0.081	[−0.34, +2.62]
Qwen2.5-Coder-3B-Base BCB-Hard	StratRand$_{0.50}$ − Cherry-LLM	+1.80	1.48	+0.70	0.347	[−4.55, +8.16]
Qwen2.5-Coder-3B-Base BCB-Hard	StratRand$_{0.50}$ − DEITA	+0.00	0.78	+0.00	1.000	[−3.36, +3.36]
Qwen2.5-Coder-3B-Base HumanEval+ +	StratRand$_{0.50}$ − Cherry-LLM	+2.67	0.75	+2.04	0.071	[−0.58, +5.91]
Qwen2.5-Coder-3B-Base HumanEval+ +	StratRand$_{0.50}$ − DEITA	+6.70	0.60	+6.45	0.008	[+4.12, +9.28]
Qwen2.5-Coder-3B-Base MBPP+ +	StratRand$_{0.50}$ − Cherry-LLM	+2.03	0.30	+3.96	0.021	[+0.76, +3.31]
Qwen2.5-Coder-3B-Base MBPP+ +	StratRand$_{0.50}$ − DEITA	+2.10	0.29	+4.20	0.018	[+0.86, +3.34]

Table 11: **Paired effect sizes for all main-table Δ comparisons.** $n = 3$ matched seeds per row; $t_{(2)}$ and two-sided p. The StratRand plateau row at Qwen2.5-1.5B-Base is the F3 headline (§5.3); the others correspond to Tables 1, 2, 5, 9.

L Use of AI assistance

We used Anthropic's Claude (Claude Code) as an AI coding assistant during this project, in two roles. (i) *Co-drafting under author supervision*: the first author drove paper structure, framing, and all em-

pirical claims; Claude was used to draft and polish prose, suggest reorganisations, and pressure-test arguments through iterative dialogue. (ii) *Code generation*: portions of the data-processing, selection, training, and evaluation code (e.g., selector implementations, sweep scripts, plotting) were drafted with Claude's assistance and then edited and integrated by the first author. All AI-generated text and code were reviewed, edited, and verified by the first author before inclusion in the paper or the released artifact, and the first author takes full responsibility for the content of this submission.