SMALLER MODELS, BETTER REJECTS: PREFERENCE DISTILLATION SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference distillation commonly treats a teacher response as preferred and the student's own response as rejected. This practice rests on two assumptions: the student's own failures are the most informative negatives, and rejects must come from a model as large as the student, which makes reject generation increasingly costly as students scale. We find that neither holds: across students from 7B to 72B, smaller frozen models generate rejects with less inference compute yet train stronger students than the student's own rejects, both before and after sequence-level knowledge distillation, on code generation and mathematical reasoning. To analyze this finding, we ask which reject distribution most improves a given student and derive a finite-horizon utility bound in a linearized feature model of DPO. The bound characterizes a favorable region of reject distributions and motivates three interventions. First, because net transfer in the bound is linear in source mixtures, we randomly mix rejects from a smaller model and from the student-scale model; performance rises as the smaller model's share grows. Second, because the bound predicts that task content independent of the prompt survives when rejects are detached from their prompts, we reassign rejects to other prompts and further shuffle their code tokens; both still outperform length-matched gibberish, so part of the gain comes from task structure rather than from errors specific to each prompt. Third, the analysis shows that selecting candidates with lower likelihood under the SeqKD reference helps when higher-likelihood candidates carry less useful contrast, as may happen for near misses that share features with correct responses. We reselect each source's candidate rejects by this likelihood; lower-likelihood selections outperform higher-likelihood ones for every source. Together, these results suggest a simple design principle: effective rejects preserve task structure while limiting coupling to the reference policy, and smaller frozen models provide both at low cost.

1 INTRODUCTION

Knowledge distillation transfers the capabilities of a large teacher model to a smaller student (Hinton et al., 2015) and has become a standard way to build capable large language models at lower cost (Guo et al., 2025; Tunstall et al., 2023). When the teacher is a proprietary model accessible only through its outputs, black-box knowledge distillation trains a student to imitate the teacher's responses (Ye et al., 2025; Chen et al., 2026c). Beyond imitation, several preference-based distillation methods instead construct supervision from teacher and student responses, treating the teacher output as preferred and the student output as rejected (Zhang et al., 2024; Li et al., 2024; Gu et al., 2025; Chen et al., 2026c). SODA applies this idea in a static pipeline with two stages (Chen et al., 2026c). The student first learns from teacher responses through sequence-level knowledge distillation (SeqKD (Kim & Rush, 2016)) and then undergoes DPO (Rafailov et al., 2023) on fixed teacher and student responses. Using the student's own response as the reject seems intuitive because its failures appear to be the most relevant alternatives from which it should learn.

This choice raises both a scaling question and a data question. As the student grows, generating self rejects requires sampling from the same increasingly large model. Yet the student's own failures, although verified incorrect, may not provide the most useful negative supervision: after SeqKD on teacher responses, they can be near misses that are already likely under the DPO reference, which may leave little for DPO to contrast. Fig. 1 summarizes our main finding. Across Qwen2.5 Bai et al.

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.

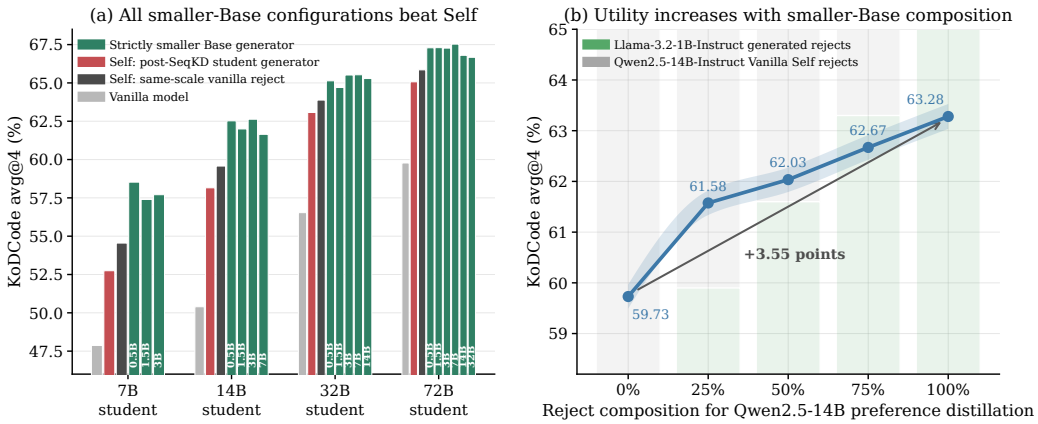

Figure 1: Smaller Base models provide stronger rejects, and their advantage varies systematically with source composition. **Left:** KoDCode avg@4 for the original vanilla model, both Self constructions, and every smaller Base source available to each 7B–72B student. All 18 smaller Base configurations exceed both post-SeqKD Self and vanilla Self. **Right:** a randomized mixture intervention on a matched 14B population. Avg@4 rises from 59.73 to 63.28 as the share of Llama-1B Base rejects increases from 0% to 100%.

(2023) student models scaling from 7B to 72B, every tested *smaller Base* source, a frozen vanilla model with strictly fewer parameters than the student, trains a stronger student than either Self construction: *vanilla Self*, the original model at the student scale, or *post-SeqKD Self*, the reference policy that initializes DPO. We establish this phenomenon through a large scaling study on verifiable code generation and math reasoning tasks. Across both tasks, smaller Base rejects improve over Self at every student scale, while requiring less inference compute.

Why should the cheaper frozen source also provide stronger negative supervision? This phenomenon turns reject construction into an inverse design problem: preference optimization determines how the student changes for a given reject distribution, whereas reject construction asks the reverse question of which reject distribution to supply so that the student improves the most (Rafailov et al., 2023; Won et al., 2025; Pan et al., 2026). We analyze this problem with a linearized feature model of DPO around the initialization. The model scores each reject source by how much its rejects move training toward better task performance and by what fraction of this movement points the wrong way, and it predicts improvement when the former is large and the latter is small. Applied to three ways of constructing rejects, this theoretical analysis motivates three interventions on code generation.

First, the analysis shows that the net effect of a source mixture is linear in its mixing weights, which motivates testing whether utility rises with the smaller Base share. We randomly mix smaller Base and vanilla Self rejects for a 14B student. Code generation performance rises from 59.73% to 63.28% as the Llama-1B share grows from 0% to 100% (Fig. 1b), so utility changes systematically with source composition. Second, the analysis shows that reassigning rejects to other prompts, which keeps the same pool of reject texts but breaks each reject's link to its own prompt, preserves the effect carried by content independent of the prompt. This motivates reassigning smaller Base rejects to other prompts and further permuting their code tokens. Reassigned rejects outperform length-matched gibberish at every student scale, while gibberish adds little beyond continued training on the chosen responses across all students. Even after permutation destroys token order, syntax, and prompt correspondence, the rejects still outperform gibberish, so part of the gain comes from task-characteristic code structure rather than from errors specific to each prompt. Third, the analysis shows that rejects with lower likelihood under the SeqKD reference help when higher-likelihood rejects offer less useful contrast, and smaller Base rejects are less likely than Self rejects under this reference. To test this, we sample a fixed pool of candidates from each source for the 14B student and select rejects that favor either lower or higher reference likelihood. Lower-likelihood selections outperform higher-likelihood ones for every source; for the 1.5B source, avg@4 reaches 63.14% compared with 61.88%. Together, these results suggest that useful rejects preserve task structure while limiting coupling to the reference policy, and smaller Base models provide both at low cost.

Our contributions are threefold. First, we establish a robust scaling phenomenon in preference distillation: across 7B–72B students, rejects from smaller frozen Base models require less inference compute yet train stronger students than the student's own rejects. Second, we cast reject source selection as inverse data design and analyze it in a linearized feature model of DPO. A finite-horizon bound characterizes a favorable region of reject distributions and motivates three construction interventions. Third, through these interventions we identify two observable properties of useful rejects, task-relevant structure and limited coupling to the reference policy, which yield a simple design principle for reject construction.

1.1 Related Work

Preference-based distillation. Preference-based distillation transfers supervision from stronger models through comparisons between responses. Existing methods rank responses from external generators with AI feedback (Tunstall et al., 2023) or treat teacher responses as preferred to student responses under ranking, distributional, or listwise objectives (Zhang et al., 2024; Li et al., 2024; Gu et al., 2025). SODA, closest to our setting, applies DPO to fixed teacher and student pairs after SeqKD (Chen et al., 2026c). These methods establish effective forms of preference distillation, but couple the response source to the prescribed training pipeline, while we study the reject-generating distribution as an independent design variable.

Pair selection and optimization. Other work selects or weights observed pairs, or couples data generation with training through online sampling (Huang et al., 2025; Qi et al., 2025; Lin et al., 2026; Belakaria et al., 2025; Yang et al., 2026), or reshapes how optimization acts on each pair (Yuan et al., 2025; Chen et al., 2026b;a; Mouiche, 2026; Liu et al., 2026). These studies motivate pair-level and optimization-based explanations of data utility. We test such explanations, but the resulting diagnostics do not consistently recover the advantage of strictly smaller Base rejects over Self rejects.

Distribution-level data design. DID derives a rejected-response sampling distribution from the differential information between target and reference policies (Won et al., 2025); our analysis instead characterizes useful reject distributions by their effect on task utility. Cao et al. (2026) show that stronger teachers need not always yield better students and address this mismatch with a curriculum over progressively stronger teachers. Pan et al. (2026) find that chosen-response quality often dominates contrastiveness and on-policy mixing. Source identity and composition also matter: multi-model pairs can induce superficial shortcuts in safety alignment (Wang et al., 2025), pairs from weaker generators can still improve a stronger learner (Geng et al., 2025; Lee et al., 2026), and mixing on-policy and off-policy pairs has task-dependent effects (Li & Khashabi, 2025). Unlike work that changes complete pairs or the chosen-response source, we hold prompts, chosen responses, initialization, objective, and training budget fixed while varying reject-source identity, relative scale, source composition, and reference coupling.

2 Preference Distillation Setting

For a student s, SeqKD trains on $\mathcal{D}_{\mathrm{KD}} = \{(x_i^{\mathrm{KD}}, y_i^{\mathrm{KD}})\}$ and produces S, which initializes the DPO policy and serves as its frozen reference. On a disjoint prompt set, reject construction q produces the preference dataset

$$\mathcal{D}_{\mathrm{pref}}(q) = \{(x_j^{\mathrm{pref}}, y_j^{+}, y_{j,q}^{-})\}, \tag{1}$$

where y_j^{+} is a verified teacher response and $y_{j,q}^{-}$ is the reject produced by q. Given a preference triple (x, y^{+}, y^{-}), DPO minimizes

$$\ell_{\mathrm{DPO}}(\theta; x, y^{+}, y^{-}) = -\log \sigma\left(\beta\left[\log \frac{\pi_\theta(y^{+} \mid x)}{S(y^{+} \mid x)} - \log \frac{\pi_\theta(y^{-} \mid x)}{S(y^{-} \mid x)}\right]\right), \tag{2}$$

where σ is the sigmoid function and $\beta > 0$ controls the scale of the reference-relative preference margin. Training minimizes the average loss over $\mathcal{D}_{\mathrm{pref}}(q)$.

Let $\mathrm{PD}_s(q)$ denote the policy obtained after the fixed DPO training protocol. Within each student scale, all variants share the prompts, chosen responses, SeqKD initialization, objective, optimizer, training schedule, training budget, and evaluation protocol. Only the reject construction changes. For two constructions q and q', we define their *reject-source effect* as

$$\Delta_s(q, q') = \mathrm{Perf}(\mathrm{PD}_s(q)) - \mathrm{Perf}(\mathrm{PD}_s(q')), \tag{3}$$

where Perf denotes the task performance metric. This contrast measures the dataset-level effect of replacing one reject construction with another under the shared protocol. The following section develops a linearized feature model to analyze how reject composition affects DPO updates and finite-horizon utility.

3 Reject Source Selection as Inverse Data Design

We study reject construction as an inverse design problem: how should we choose the reject distribution to improve a given student? We analyzes DPO on constructed preference pairs through a linearized feature model, characterizes each reject source by two transfer coordinates, and derives three construction interventions from this characterization, which we evaluate in §4. The full construction, formal statements, and proofs appear in Appx. §F.

3.1 A linearized feature model of Reject Utility

A construction q from §2 changes only the distribution $Q(y^- \mid x)$ of its rejects; the preference-prompt distribution p, the distribution $T(y^+ \mid x)$ of verified teacher responses, and the SeqKD reference S are shared across constructions. Let $\phi(x, y) \in \mathbb{R}^d$ be fixed response features and $v \in \mathbb{R}^d$ a trainable coefficient vector, and consider

$$\pi_v(y \mid x) = \frac{S(y \mid x) \exp\big(v^\top \phi(x, y)\big)}{\mathbb{E}_{y' \sim S(\cdot|x)} \exp(v^\top \phi(x, y'))}, \qquad \pi_0 = S. \tag{4}$$

Taking $\phi(x, y) = \nabla_\theta \log \pi_\theta(y \mid x)\big|_{\theta_0}$, the score gradients of the reference network at its SeqKD parameters θ_0, makes π_v the network's first-order model around the shared DPO initialization, following the local linearization perspective on neural training (Lee et al., 2019) (§F.2). In this geometry a reject can share features with the chosen response and contrast along others (Yuan et al., 2025; Razin et al., 2025): for a task requiring `max(xs)`, the reject `min(xs)` contrasts on maximum versus minimum selection, whereas `max(0, max(xs))` shares maximum selection and differs only in an erroneous zero lower bound (detailed illustration in §F.1).

Let $\delta = \phi(x, y^+) - \phi(x, y^-)$, and write $\mathbb{E}_Q[\cdot]$ for the expectation over triples $(x, y^+, y^-) \sim p(x)\, T(y^+ \mid x)\, Q(y^- \mid x)$. The shared normalizer cancels, so a pair's DPO margin in Eq. 2 is $\beta v^\top \delta$, and full-batch gradient descent with step sizes $\eta_t > 0$ follows

$$v_{t+1}^Q = v_t^Q + \eta_t F_Q(v_t^Q), \qquad F_Q(v) = \beta\, \mathbb{E}_Q\big[\sigma(-\beta v^\top \delta)\, \delta\big], \qquad v_0^Q = 0. \tag{5}$$

Each pair pushes v along δ with a weight that decays as its margin grows; the construction enters the dynamics only through the distribution of δ.

3.2 Feature Transfer and a Favorable Region

Eq. 5 specifies how a construction moves v; we now ask how that movement translates into task performance, and for which reject distributions it helps. We measure utility by expected task reward, denoted as $U(v) = \mathbb{E}_{x \sim p_{\mathrm{eval}}, y \sim \pi_v(\cdot|x)}[r(x, y)]$, where p_{eval} is the evaluation-prompt distribution and the reward $r(x, y) \in [0, 1]$ scores task success, and let $u = \nabla U(0) \neq 0$ and $\widehat{u} = u/\|u\|$, the direction in which an initial update most improves utility. A reject is useful or harmful only relative to what it replaces, so we fix a comparison reject distribution $Q_0(y \mid x)$ with the same p and T. Let $\bar{\phi}_0(x) = \mathbb{E}_{y \sim Q_0(\cdot|x)} \phi(x, y)$, and score a reject by the transfer score

$$k(x, y) = \big\langle \widehat{u},\, \bar{\phi}_0(x) - \phi(x, y) \big\rangle, \tag{6}$$

positive when substituting y for the comparison reject moves the initial update further along the utility direction. Two coordinates summarize a source:

$$a_Q = \mathbb{E}_{x \sim p,\, y \sim Q} |k(x, y)|, \qquad c_Q = \frac{\mathbb{E}_{x \sim p,\, y \sim Q}[-k(x, y)]_+}{a_Q}, \qquad \kappa_Q = \mathbb{E}\, k = a_Q(1 - 2c_Q), \tag{7}$$

with $[w]_+ = \max(w, 0)$, all expectations over $x \sim p$ and $y \sim Q$, and $c_Q = 0$ when $a_Q = 0$. Here a_Q is the transfer mass (how much the source moves the update along $\widehat{u}$ at all) and c_Q the adverse

share (the fraction of that movement pointing the wrong way), so κ_Q is the net transfer that survives cancellation. Because prompts and chosen responses are shared, the initial field difference obeys the exact identity $\langle u, F_Q(0) - F_{Q_0}(0)\rangle = \frac{\beta}{2}\|u\|\kappa_Q$. Over a finite horizon, this initial advantage carries through training:

Proposition 1 (Finite-horizon transfer bound)**.** *Under regularity conditions shared by the compared constructions (bounded features suffice; §F.3), H steps with positive step sizes η_t and total step size $\tau_H = \sum_{t=0}^{H-1} \eta_t$ give*

$$G_H(Q) = U(v_H^Q) - U(v_H^{Q_0}) \;\geq\; K_H\, a_Q(1 - 2c_Q) - E_H, \qquad K_H = \frac{\beta\tau_H\|u\|}{2}, \tag{8}$$

where $E_H = O(\tau_H^2)$ collects field drift and utility curvature and is given explicitly in §F.3.

For a target gain $\epsilon > 0$, the bound implies $G_H(Q) \geq \epsilon$ on the favorable region

$$\mathcal{F}_{H,\epsilon} = \left\{ Q : 0 \leq c_Q < \frac{1}{2}, \quad a_Q \geq \frac{\epsilon + E_H}{K_H(1 - 2c_Q)} \right\}. \tag{9}$$

The two conditions trade off: a larger adverse share c_Q requires a larger transfer mass a_Q. Because K_H is linear in the total step size τ_H while $E_H = O(\tau_H^2)$, scaling all step sizes by a sufficiently small factor makes the bound in Eq. 8 positive for any source with $\kappa_Q > 0$, and membership in $\mathcal{F}_{H,\epsilon}$ is exactly the condition that this bound is at least ϵ.

3.3 Construction Interventions and Hypotheses

The coordinates a_Q and c_Q are not directly observable. We therefore translate them into three construction operators, each of which moves κ_Q in a direction fixed by an explicit hypothesis about the source, and state the hypotheses as P1–P3. For a construction q inducing Q, an operator $\mathcal{O}$ yields a construction $q_{\mathcal{O}}$ inducing $\mathcal{O}Q$, whose effect is the causal contrast $\Delta_s(q_{\mathcal{O}}, q)$ of Eq. 3.

P1: source mixtures. Randomized source assignment implements

$$\mathcal{M}_{\mathbf{w}}(Q_1, \ldots, Q_m) = \sum_{s=1}^{m} w_s Q_s, \qquad \mathbf{w} \in \Delta^{m-1}, \tag{10}$$

with Δ^{m-1} the probability simplex. Because k is fixed by the common reference, utility, and comparison distribution, $\kappa_{\mathcal{M}_{\mathbf{w}}} = \sum_s w_s \kappa_{Q_s}$ and $a_{\mathcal{M}_{\mathbf{w}}} = \sum_s w_s a_{Q_s}$: net transfer is affine in $\mathbf{w}$, which motivates testing whether utility rises along the mixture path as the smaller Base share grows (§4.2).

P2: prompt reassignment and lexical permutation. With $\overline{Q}(y) = \mathbb{E}_{x' \sim p} Q(y \mid x')$ the prompt-averaged reject distribution, reassignment applies

$$(\mathcal{P}_\rho Q)(y \mid x) = (1 - \rho)Q(y \mid x) + \rho\, \overline{Q}(y). \tag{11}$$

For an additive decomposition $\phi = \phi_{\mathrm{dom}}(y) + \phi_{\mathrm{pair}}(x, y)$, the mean of ϕ_{dom} is invariant under $\mathcal{P}_1$ (§F.4), so the transfer it carries survives, and lexical permutation removes all but its order-invariant part. This motivates testing whether both retain utility over length-matched gibberish (§4.3).

P3: reference-likelihood reselection. Within a generator-specific candidate bank C_x, let $\tilde{s}_j$ standardize the length-normalized reference score $s_j = |y_j|^{-1} \log S(y_j \mid x)$ within the bank. The reselection rule samples

$$q_\gamma(j \mid C_x) = \frac{\exp(-\gamma \tilde{s}_j)}{\sum_l \exp(-\gamma \tilde{s}_l)}, \tag{12}$$

inducing the reject distribution Q_γ, with $\gamma > 0$ repelling from the reference, $\gamma < 0$ attracting, and $\gamma = 0$ native. Then $d\kappa_{Q_\gamma}/d\gamma = -\mathbb{E}_{x \sim p} \mathrm{Cov}_{j \sim q_\gamma}(\tilde{s}_j, k)$ (§F.4), so repulsion raises net transfer whenever candidates with higher reference likelihood transfer less. The construction hypothesis is that this holds within model-generated candidate banks: near misses that share features with the correct response sit closer to the reference yet supply less contrast, as in the two-feature example of §F.1. This motivates testing whether repulsion outperforms attraction within fixed candidate banks (§4.4).

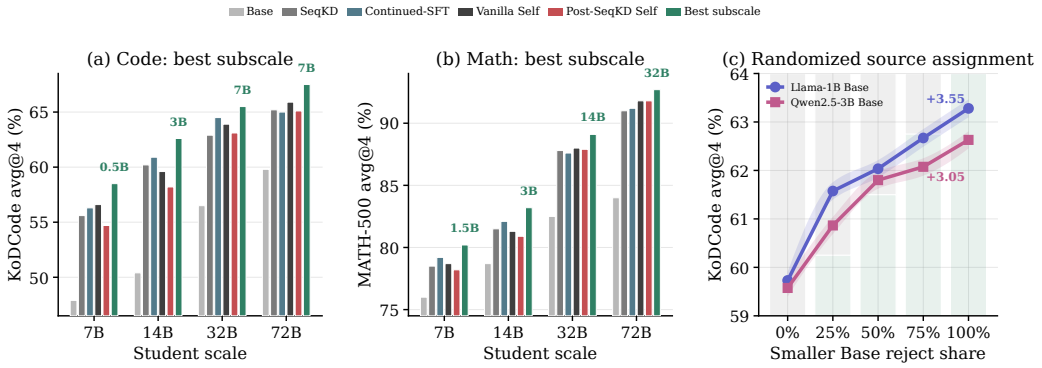

Figure 2: Smaller Base models improve preference distillation across code generation and mathematical reasoning. **(a, b)** Absolute avg@4 for the original vanilla model, SeqKD, Continued-SFT, both Self constructions, and the best observed smaller Base source for each 7B–72B student. Labels above the green bars identify the displayed reject source. **(c)** Two randomized source assignment interventions vary the proportions of smaller Base and vanilla Self rejects for a 14B student. The green and gray background shows the dataset composition. Performance increases with the smaller Base share for both source families.

4 Experiments

4.1 Experimental Settings

We study the two stage preference-distillation pipeline introduced by SODA (Chen et al., 2026c). A Qwen2.5 student first learns execution verified teacher responses through SeqKD and then undergoes DPO on a disjoint preference set. Within each student scale, all DPO variants share the SeqKD initialization, prompts, chosen responses, objective, optimization budget, and evaluation protocol. Only reject construction changes. A *smaller Base source* is a frozen vanilla model with strictly fewer parameters than the student. We compare these sources with vanilla Self and post-SeqKD Self. Vanilla Self uses the matching instruction tuned model, while post-SeqKD Self uses the policy that initializes DPO. We write Self for both when the distinction is immaterial. Our main scaling study covers Qwen2.5-Instruct students at 7B, 14B, 32B, and 72B (Bai et al., 2023). The code line uses KoDCode (Xu et al., 2025). SeqKD trains on 75,752 prompts with execution verified GPT-4o responses (OpenAI, 2024). DPO uses a disjoint preference split of 24,248 prompts. Each prompt has a separately generated and verified GPT-4o response as chosen, together with a reject from the designated frozen source. Every reject is verified incorrect: it fails at least one unit test. We evaluate on 5,000 held out KoDCode problems and four external benchmarks: BigCodeBench-Complete, BigCodeBench-Instruct (Zhuo et al., 2025), HumanEval (Chen et al., 2021), and MBPP (Austin et al., 2021). The primary in domain endpoint is avg@4, the mean execution success over four sampled responses. The mathematical reasoning line uses verifier correct DeepSeek-R1 trajectories from OpenR1-Math-220k (Guo et al., 2025; Open R1, 2025). These trajectories provide the teacher responses for SeqKD and the chosen responses for preference training. Every reject is sampled from the designated source and verified incorrect by the answer verifier. We evaluate avg@4 on MATH-500 (Hendrycks et al., 2021) and the 2024 and 2025 AIME problems (Mathematical Association of America, 2025). Complete split construction, training, decoding, and evaluation details in Appx. §A.

4.2 RQ1: How Should Reject Generation Scale with the Student?

Smaller Base sources consistently outperform Self. Fig. 2 summarizes the best observed smaller Base endpoint at every student scale on both code generation and mathematical reasoning. Across the complete code scaling matrix, all smaller Base configurations outperform both Self constructions on avg@4 (Tab. 3). The gains are substantial across both task families. For example, rejects generated by Qwen2.5-3B-Instruct improve the 14B student's avg@4 over vanilla Self by 3.1 points on code generation and 1.9 points on mathematical reasoning. The best source varies with the student and task. Llama Base sources also remain competitive with the best Qwen2.5 smaller Base source

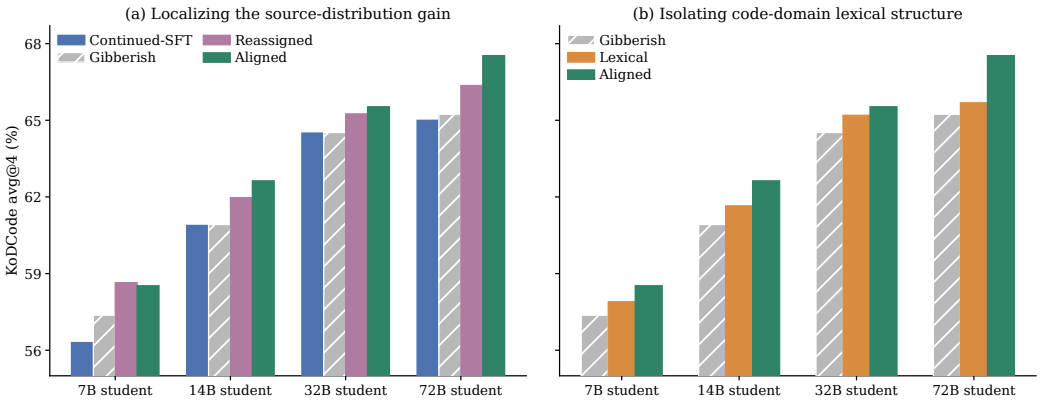

Figure 3: Localizing and decomposing the utility of smaller Base rejects. **(a)** Absolute KoDCode avg@4 across four control settings at every student scale. Continued-SFT trains on the preference-stage chosen responses alone. Gibberish supplies length-matched artificial rejects. Prompt reassignment preserves real Base rejects while removing their original prompt correspondence, and Aligned restores that correspondence. **(b)** Lexical permutation preserves the generated code-token inventory of prompt-reassigned Base rejects while destroying token order, syntax, and coherent program semantics. Aligned shows the original prompt-matched Base construction.

(§C.1). Complete in-domain and external code results are reported in Tabs. 3 and 4, and the complete mathematical reasoning results are reported in Tab. 5. These sources are also cheaper: on the code preference prompts, we estimate generation compute using parameter count adjusted by output length. For smaller Base sources, this estimate is 0.9% to 50.1% of vanilla Self at the same student scale, corresponding to a $2.0\times$ to $108.3\times$ reduction. The complete cost analysis and measured GPU runtimes appear in Appx. §B.

Randomized source assignment controls performance. The pure source comparison changes an entire reject dataset at once. We therefore apply the randomized source assignment operator of Eq. 10 (P1 in §3.3), varying the proportions of smaller Base and vanilla Self rejects on a matched 14B training population. As shown in Fig. 2c, increasing the share of Llama-1B Base rejects from 0% to 100% raises code avg@4 from 59.73% to 63.28%. Utility therefore changes systematically with source composition rather than only between the two pure endpoints. The Qwen2.5-3B intervention follows the same ordering from 59.58% to 62.63%, and a 72B midpoint lies between its two pure source endpoints. Full composition results are reported in §D.1.

Takeaway 1. *Reject generation need not scale with the student. Across the tested 7B to 72B range, smaller Base models use less inference compute and provide stronger rejects than either Self construction.*

4.3 RQ2: Where Does the Smaller Base Gain Reside?

The scaling study identifies a stable source effect, but it does not determine whether the gain requires negatives at all, whether any suppressible text is sufficient, or whether the gain depends on the exact prompt and reject pairing. We address these possibilities with direct construction interventions, realizing the two operators of P2 (§3.3), first localizing the gain to the reject source distribution and then identifying a structural component carried by that distribution.

Real source distributions retain utility without prompt correspondence. We construct four controls at every student scale using the strongest in domain Base source for that student. Continued-SFT starts from the SeqKD model and trains on the DPO-stage chosen responses alone. Gibberish replaces every reject with length-matched artificial text, serving as the comparison construction for P2. Prompt reassignment assigns real Base rejects to different prompts, realizing $\mathcal{P}_1$ in Eq. 11, with construction-specific normalization and filtering as described below. The aligned setting restores the original prompt and reject correspondence. As shown in Fig. 3a, Gibberish closely tracks

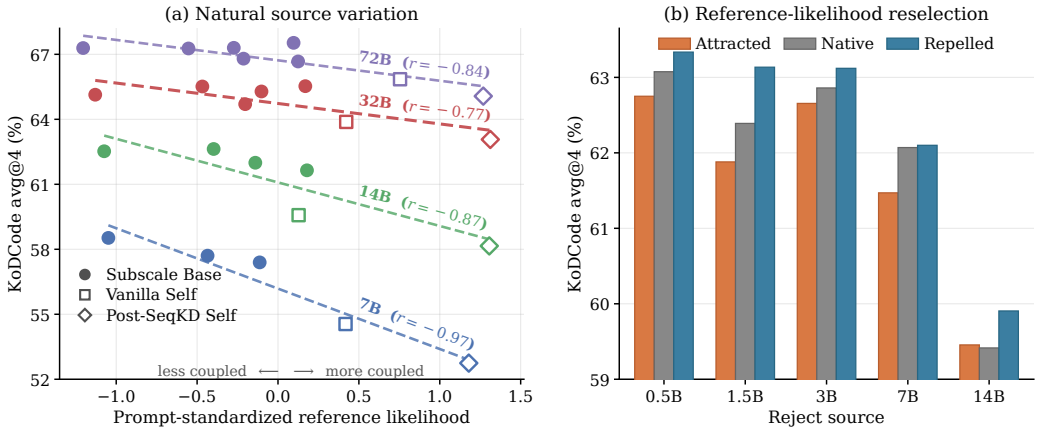

Figure 4: Reference likelihood tracks and controls reject utility. **(a)** Each point is a natural reject source scored by its mean length-normalized likelihood under the corresponding student's SeqKD reference and standardized within prompt. Lines are fitted separately for each student scale; labels report Pearson correlations. Filled circles are smaller Base sources, squares are vanilla Self, and diamonds are post-SeqKD Self. **(b)** For a fixed Qwen2.5-14B-Instruct student, each source-specific candidate bank is reselected toward lower reference likelihood (Repelled), without likelihood preference (Native), or toward higher reference likelihood (Attracted).

Continued-SFT at 14B, 32B, and 72B, with a larger gain at 7B. Prompt-reassigned Base rejects improve over Gibberish at every scale. The largest gain appears for the 7B student, where reassignment improves avg@4 over Gibberish by 1.28 points. These controls remove the original prompt correspondence; some also apply interface normalization and execution filtering. Separate exact-derangement controls preserve the original reject-text multiset. Together they show that part of the gain does not require the original prompt and reject pairing and can come from features of the reject text alone; the incremental effect of restoring alignment depends on the student and source. Complete endpoints, paired contrasts, and reward trajectories appear in §D.2.

Code-domain lexical structure carries corrective value. We next permute the lexical tokens within every prompt-reassigned Base reject, keeping only the token inventory, the order-invariant part of the text feature in P2. This operation preserves the generated code-token inventory while destroying token order, Python syntax, coherent programs, and prompt correspondence. Despite this destruction, Lexical significantly outperforms Gibberish at every student scale, as shown in Fig. 3b. The largest gain occurs for the 14B student, where preserving only the generated code-token inventory improves avg@4 by 0.73 points. Code-domain lexical structure therefore carries corrective value even without an executable program, coherent failure semantics, or prompt correspondence. An AST-based intervention and complete paired contrasts are reported in §D.2.

Takeaway 2. *The utility of real Base rejects survives removal of exact prompt correspondence, and code-domain lexical structure recovers a substantial part of this distribution-level gain across every student scale.*

4.4 RQ3: What Makes a Reject Distribution Useful?

RQ2 shows that useful rejects retain structure from the code domain. We further analyze what distinguishes smaller Base sources from Self within this structured output family. The analysis in §3 identifies likelihood under the SeqKD reference as a property of the reject distribution that we can manipulate directly, and P3 in §3.3 predicts an advantage for repulsion over attraction when higher-likelihood candidates transfer less. We test this in two steps. First, we ask whether reference likelihood organizes the utility of naturally generated reject sources across student scales. Second, we manipulate this quantity directly by reselecting from fixed candidate banks.

Reference likelihood organizes the natural source boundary. For every student, we score each reject source under that student's exact SeqKD reference on the same preference prompts. As shown in Fig. 4a, higher reference likelihood is associated with lower downstream avg@4 at every student scale. The strongest relationship occurs for the 7B student, with a Pearson correlation of -0.97. Smaller Base sources consistently combine lower coupling with higher utility, while vanilla Self and post-SeqKD Self combine higher coupling with lower utility. Reference likelihood thus organizes the stable source boundary between smaller Base and Self rejects. Complete source measurements and robustness analyses are reported in §D.3.

Lower reference coupling improves utility under controlled reselection. To separate coupling from the choice of generator, we fix each reject source and change only which of its responses we use. For each prompt, the source generates eight candidates, and we score each one by its per-token log likelihood under the 14B student's SeqKD reference. From these candidates we pick one reject in three ways (Eq. 12): Repelled favors low likelihood, Native picks at random, and Attracted favors high likelihood. A prompt is then kept only if all three selected rejects run to completion, pass exactly the same fraction of unit tests, and differ in length by at most 32 tokens. The generator, candidate bank, student initialization, and DPO protocol remain fixed within each source. Fig. 4b shows that lower reference likelihood improves utility in our candidate banks: Repelled produces the strongest endpoint for every reject source, while Attracted selection reduces utility for each smaller Base source. The largest separation occurs for the 1.5B source, where Repelled reaches 63.14% avg@4 compared with 61.88% for Attracted, a gain of 1.26 points. Because reselection changes reference likelihood without changing the generator or its candidate bank, this result supports treating reference coupling as an operational property of the reject distribution rather than only a proxy for model scale. Manipulation checks, raw endpoints, paired effects, and interaction contrasts for this intervention are reported in §D.4.

Together, RQ2 and RQ3 suggest why smaller Base sources are effective. Gibberish lacks the code-domain structure that carries corrective value. Self rejects preserve this structure but remain strongly coupled to the policy used as the DPO reference. Smaller Base sources have both properties: they retain code-domain structure while remaining less coupled to the reference. Reference repulsion lowers coupling further and improves utility. Additionally, optimization diagnostics, including reject fields, reward margins, likelihood trajectories, and RC-DPO, do not recover the smaller Base over Self ordering (Appx. §E).

Takeaway 3. *Across natural reject sources, lower reference coupling separates smaller Base from Self rejects. Within fixed source-specific candidate banks, reference repulsion improves downstream utility. Task-related structure and limited reference coupling are the two theory-motivated source properties that we identify empirically.*

5 Conclusion

In this paper, we study how rejected responses should be constructed for preference distillation, rather than treating the student's own failures as a fixed default. Across students from 7B to 72B, smaller Base models produce stronger rejects than both Self constructions while requiring less inference compute. Randomized source assignment further shows that downstream performance changes systematically with reject source composition. Reject generation therefore need not scale with the student, and a smaller Base model provides a simple and effective source in the settings we study.

Our analysis derives a favorable region of reject distributions and motivates two source properties that our interventions then identify empirically. Arbitrary gibberish provides no residual benefit beyond continued training on chosen responses, whereas real Base rejects retain substantial utility after their original prompt correspondence is removed. Lexical structure from the code domain carries part of this gain at the distribution level even without coherent program semantics. Among incorrect outputs generated by models, reference likelihood then separates useful sources: smaller Base rejects are less coupled to the SeqKD reference than Self rejects, and directly selecting rejects with lower reference likelihood improves downstream utility within fixed candidate banks. Together, these results identify reject construction as an inverse data design problem: in our experiments, effective rejects preserve task structure while avoiding excessive coupling to the reference policy. Model scale exposes this regime, but the broader opportunity is to control these properties directly without increasing the cost of reject generation.

AI USE STATEMENT

We used generative AI tools to polish the writing of this paper, including grammar, wording, and readability. We did not use generative AI tools for research ideation, experimental design, running experiments, data analysis, theoretical derivations, or literature search. The authors reviewed every AI-assisted edit against the underlying results and take full responsibility for the content of the paper.

ETHICS STATEMENT

This work does not involve human subjects, crowdsourcing, or the collection of new personal data. All training and evaluation data come from publicly released datasets and benchmarks. All models are publicly released Qwen2.5 and Llama models, and teacher responses come from GPT-4o and from DeepSeek-R1. We use these resources under their respective licenses and terms of use. Our experiments deliberately collect incorrect model outputs, but these outputs serve only as rejects that the student learns to avoid, and they are restricted to programming and mathematics problems. Model-generated code is executed only to verify correctness against unit tests. The authors declare no competing interests.

REPRODUCIBILITY STATEMENT

§4 describes the pipeline, the reject sources, and the controls shared by all compared variants. Appx. §A gives the data sources, the split sizes of each training stage, the reject verification procedure, and the evaluation protocol and metrics; decoding settings appear in the caption of Tab. 3. Appx. §B reports the cost of reject generation and the measured H100 generation time. Appx. §C and Appx. §D report every numerical result behind the figures, including the construction of each intervention and its prompt population. Appx. §E defines the diagnostics used in the mechanism analyses. The complete theoretical construction, formal assumptions, and proofs appear in Appx. §F.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Syrine Belakaria, Joshua Kazdan, Charles Marx, Chris Cundy, Willie Neiswanger, Sanmi Koyejo, Barbara E Engelhardt, and Stefano Ermon. Sharpe ratio-guided active learning for preference optimization in rlhf. *arXiv preprint arXiv:2503.22137*, 2025.

Jincheng Cao, Fanzhi Zeng, Leqi Liu, and Aryan Mokhtari. Curriculum learning-guided progressive distillation in large language models. *arXiv preprint arXiv:2605.11260*, 2026.

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.

Shaolong Chen, Madalina Ciobanu, Qingqing Mao, and Ritankar Das. Adadpo: Self-adaptive direct preference optimization with balanced gradient updates. *arXiv preprint arXiv:2605.28440*, 2026a.

Wei Chen, Yubing Wu, Junmei Yang, Delu Zeng, Qibin Zhao, John Paisley, Min Chen, and Zhou Wang. Towards disentangled preference optimization dynamics: Suppress the loser, preserve the winner. *arXiv preprint arXiv:2604.18239*, 2026b.

Xiwen Chen, Jingjing Wang, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Yueyue Deng, Hejian Sang, Zhipeng Wang, Alborz Geramifard, and Feng Luo. Soda: Semi on-policy black-box distillation for large language models. *arXiv preprint arXiv:2604.03873*, 2026c.

Scott Geng, Hamish Ivison, Chun-Liang Li, Maarten Sap, Jerry Li, Ranjay Krishna, and Pang Wei Koh. The delta learning hypothesis: Preference tuning on weak data can yield strong gains. *arXiv preprint arXiv:2507.06187*, 2025.

Yanggan Gu, Junzhuo Li, Sirui Huang, Xin Zou, Zhenghua Li, and Xuming Hu. Capturing nuanced preferences: Preference-aligned distillation for small language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15959–15973, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL `http://arxiv.org/abs/1503.02531`.

Kexin Huang, Junkang Wu, Ziqian Chen, Xue Wang, Jinyang Gao, Bolin Ding, Jiancan Wu, Xiangnan He, and Xiang Wang. Larger or smaller reward margins to select preferences for llm alignment? In *Forty-second International Conference on Machine Learning*, 2025.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Chia-Hsuan Lee, Mingyang Zhou, Renkun Ni, Zelei Cheng, Sihui Dai, Supriyo Chakraborty, Shixiong Zhang, Sambit Sahu, and William Campbell. Decomposing the delta: What do models actually learn from preference pairs? *arXiv preprint arXiv:2604.08723*, 2026.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Tianjian Li and Daniel Khashabi. Simplemix: Frustratingly simple mixing of off-and on-policy data in language model preference learning. *arXiv preprint arXiv:2505.02363*, 2025.

Yixing Li, Yuxian Gu, Li Dong, Dequan Wang, Yu Cheng, and Furu Wei. Direct preference knowledge distillation for large language models. *arXiv preprint arXiv:2406.19774*, 2024.

Xiaoqiang Lin, Arun Verma, Zhongxiang Dai, Daniela Rus, See-Kiong Ng, and Bryan Kian Hsiang Low. Activedpo: Active direct preference optimization for sample-efficient alignment. In *International Conference on Learning Representations*, volume 2026, pp. 21618–21635, 2026.

Jilong Liu, Yonghui Yang, Pengyang Shao, Wenjian Tao, Hao Zhan, Haokai Ma, Wei Qin, and Richang Hong. Compassdpo: Dynamics-controlled direct preference optimization for robust safety alignment. *arXiv preprint arXiv:2603.07211*, 2026.

Mathematical Association of America. American invitational mathematics examination. MAA American Mathematics Competitions, 2025. URL https://maa.org/student-programs/amc/.

Inoussa Mouiche. Gradient-gated dpo: Stabilizing preference optimization in language models. *arXiv preprint arXiv:2605.02626*, 2026.

Open R1. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

OpenAI. GPT-4o system card, 2024. URL https://openai.com/index/gpt-4o-system-card/.

Yu Pan, Zhongze Cai, Huaiyang Zhong, Guanting Chen, and Chonghuan Wang. What matters in data for dpo? *Advances in Neural Information Processing Systems*, 38:44689–44716, 2026.

Xuan Qi, Rongwu Xu, and Zhijing Jin. Difficulty-based preference data selection by dpo implicit reward gap. *arXiv preprint arXiv:2508.04149*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 24791–24834, 2025.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Yifan Wang, Runjin Chen, Bolian Li, David Cho, Yihe Deng, Ruqi Zhang, Tianlong Chen, Zhangyang Wang, Ananth Grama, and Junyuan Hong. More is less: The pitfalls of multi-model synthetic preference data in dpo safety alignment. *arXiv preprint arXiv:2504.02193*, 2025.

Yunjae Won, Hyunji Lee, Hyeonbin Hwang, and Minjoon Seo. Differential information distribution: A bayesian perspective on direct preference optimization. *arXiv preprint arXiv:2505.23761*, 2025.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, 2025.

Junming Yang, Ning Xu, Biao Liu, Shiqi Qiao, and Xin Geng. Alignment through meta-weighted online sampling: Bridging the gap between data generation and preference optimization. In *International Conference on Learning Representations*, volume 2026, pp. 119593–119620, 2026.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Hui Yuan, Yifan Zeng, Yue Wu, Huazheng Wang, Mengdi Wang, and Liu Leqi. A common pitfall of margin-based language model alignment: Gradient entanglement. In *International Conference on Learning Representations*, volume 2025, pp. 5278–5297, 2025.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Haorui Wang, Zhen Qin, Feng Han, Jialu Liu, Simon Baumgartner, Michael Bendersky, and Chao Zhang. Plad: Preference-based large language model distillation with pseudo-preference pairs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15623–15636, 2024.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025.

A Experimental Details

Our main scaling study covers Qwen2.5 students at 7B, 14B, 32B, and 72B. We train and evaluate separate lines for verifiable code generation and mathematical reasoning.

Code training line. The main scaling experiment uses KoDCode (Xu et al., 2025). SeqKD trains on 75,752 prompts with execution-verified teacher solutions. The preference stage uses a disjoint set of 24,248 prompts, each paired with a separately generated, execution-verified teacher response y^+ and a reject y_q^- from the designated frozen source. Each reject is verified incorrect by execution: it fails at least one unit test. All Continued-SFT and DPO variants within a student scale use the same preference prompts and y^+ responses; only DPO uses rejects, and only the reject source changes across DPO variants.

Math training line. We use a separate OpenR1-Math line to test whether residual value beyond Continued-SFT also appears in mathematical reasoning. Students are first trained on filtered verifier-correct reasoning trajectories (33,524 prompts for SeqKD) and then receive Continued-SFT or preference training on OpenR1-Math preferences (12,288 prompts for DPO). Each reject is sampled from the designated source and retained only if the automatic answer verifier marks its final answer incorrect.

Code evaluation. The in-domain test contains 5,000 KoDCode problems disjoint from both training stages. We report greedy pass@1 and four stochastic responses per problem: avg@4 is the mean execution success rate, and pass@4 is the fraction of problems solved at least once. For KoDCode-trained models, external evaluation uses BigCodeBench-Complete (BCB-C) and BigCodeBench-Instruct (BCB-I) (Zhuo et al., 2025), which contain the same 1,140 problems under different prompt formats, together with HumanEval (Chen et al., 2021) (164 problems) and MBPP (Austin et al., 2021) (257 problems). The reported external average weights each benchmark by its number of problems and is a descriptive summary.

Math evaluation. OpenR1-Math-trained models are evaluated on external benchmarks: MATH-500 (Hendrycks et al., 2021) and AIME (Mathematical Association of America, 2025) using an automatic answer verifier. We report avg@4 over four sampled responses.

Training hyperparameters. Tab. 1 lists the training settings of the code line. Within each student scale, all Continued-SFT and DPO variants use the same settings; only the training objective and the reject source differ.

Table 1: Training hyperparameters of the code line. All runs use full-parameter training with AdamW, a cosine learning rate schedule with 10% linear warmup, bf16 precision, gradient checkpointing, DeepSpeed ZeRO-3, and seed 42. Continued-SFT uses the same settings as DPO.

Setting	SeqKD	DPO
Learning rate	5×10^{-6}	5×10^{-7}
DPO β	–	0.1
Epochs	1	1
Gradient accumulation	16	16
Effective batch (7B–72B)	256	256
Max sequence length	2,560	2,560
Gradient clipping	0.2	0.2

Table 2: Complete output-length-adjusted parameter-token cost comparison. Decode and total costs are percentages of same-scale vanilla Self. The reduction factor is the inverse of the total-cost ratio.

Student	Smaller Base source	Parameter ratio	Reject tokens	Decode cost	Total cost	Reduction
7B	0.5B	6.5%	11.317M	10.6%	8.2%	12.2×
7B	1.5B	20.3%	6.366M	18.7%	19.6%	5.1×
7B	3B	40.5%	7.074M	41.5%	40.9%	2.4×
14B	0.5B	3.3%	11.317M	5.1%	4.1%	24.3×
14B	1.5B	10.5%	6.366M	9.0%	9.8%	10.2×
14B	3B	20.9%	7.074M	20.0%	20.5%	4.9×
14B	7B	51.6%	6.908M	48.2%	50.1%	2.0×
32B	0.5B	1.5%	11.317M	2.6%	2.0%	50.9×
32B	1.5B	4.7%	6.366M	4.6%	4.7%	21.3×
32B	3B	9.4%	7.074M	10.3%	9.8%	10.2×
32B	7B	23.2%	6.908M	24.9%	23.9%	4.2×
32B	14B	45.1%	7.387M	51.6%	47.7%	2.1×
72B	0.5B	0.7%	11.317M	1.3%	0.9%	108.3×
72B	1.5B	2.1%	6.366M	2.3%	2.2%	45.5×
72B	3B	4.2%	7.074M	5.2%	4.6%	21.8×
72B	7B	10.5%	6.908M	12.5%	11.2%	8.9×
72B	14B	20.3%	7.387M	25.8%	22.4%	4.5×
72B	32B	45.1%	6.460M	50.1%	47.0%	2.1×

B Reject Generation Cost Analysis

We measure reject-generation cost on the exact 24,248 selected responses used by the code preference stage. Let P_q denote the number of unique parameters in reject generator q, N_x the total number of prompt tokens, and N_q^- the total number of generated reject tokens. We define the decode and total parameter-token proxies as

$$C_{\text{decode}}(q) = P_q N_q^-, \qquad C_{\text{total}}(q) = P_q(N_x + N_q^-). \tag{13}$$

The first quantity measures autoregressive output work, while the second also charges prompt prefill to each generator. Both account for source-dependent response length and are reported as compute proxies rather than measured FLOPs. The shared prompts contain 9.536 million Qwen2.5 tokens, or 393.3 tokens per prompt on average.

Parameter count alone is insufficient because reject lengths differ across sources: the 0.5B source averages 466.7 tokens per reject, while the other Qwen2.5 sources average 240 to 305 tokens. The proxies in Tab. 2 account for this difference, and each cost is normalized by the same-scale vanilla Self generator used for the corresponding student.

Across all strict-subscale configurations, the total-cost proxy ranges from 0.9% to 50.1% of same-scale vanilla Self, corresponding to reductions of $2.0\times$ to $108.3\times$. Relative to post-SeqKD Self, the range is 0.9% to 55.1%, corresponding to reductions of $1.8\times$ to $108.3\times$.

We additionally calibrate the proxy with a matched H100 generation sweep in which each source generates eight candidates for the same 24,248 prompts with vLLM, temperature 0.6, and a 2,048-token output cap. Counting only the GPUs used for tensor-parallel inference, the 0.5B, 1.5B, 3B, and 7B sources require 1.22, 0.53, 0.64, and 0.84 H100 hours, compared with 2.37 for the 14B generator. The measured runtime therefore follows the direction of the proxy.

C Complete Evaluation Results

This section reports the numerical results underlying the main experiment figures. Within each student scale, all preference-stage variants share the prompts, chosen responses, SeqKD initialization, objective, optimizer budget, and evaluation protocol; only the reject source changes. Tab. 3 reports KoDCode held-out pass@1, avg@4, and pass@4 for every reject construction, and Tab. 4 reports external pass@1 on BCB-C, BCB-I, HumanEval, and MBPP. Tab. 5 reports MATH-500 and AIME avg@4; among the 14 Qwen configurations with a smaller Base source, $13/14$ exceed Continued-SFT on MATH-500 and $12/14$ on AIME.

Table 3: KoDCode held-out results. pass@1 uses greedy decoding; avg@4 is the mean success rate over four stochastic samples, and pass@4 is the fraction of problems solved by at least one sample. Stochastic decoding uses temperature 0.6 and top-$p = 1.0$. All values are percentages. Bold marks the best result within each student and metric.

		pass@1				Stochastic	
Method	Reject Source	Overall	Easy	Medium	Hard	avg@4	pass@4
Student: Qwen2.5-7B-Instruct							
Vanilla	–	48.9	62.3	45.3	40.2	47.9	59.8
SeqKD	–	58.8	70.4	56.2	50.7	55.6	69.4
Continued-SFT	–	59.6	71.6	57.0	52.0	56.3	70.0
DPO	0.5B	**60.5**	**72.0**	**57.5**	53.1	**58.5**	**70.5**
	1.5B	59.8	70.8	57.0	52.7	57.4	70.2
	3B	60.1	71.2	56.7	**53.5**	57.7	70.0
	7B(*Vanilla Self*)	58.9	70.1	56.0	51.6	54.6	68.9
	7B(*Post-SeqKD Self*)	58.6	69.5	56.0	50.7	52.7	69.9
Student: Qwen2.5-14B-Instruct							
Vanilla	–	50.9	63.3	47.4	43.1	50.4	63.4
SeqKD	–	61.7	72.6	58.8	54.7	60.2	72.4
Continued-SFT	–	62.5	73.7	59.6	55.3	60.9	72.7
DPO	0.5B	64.2	**74.9**	61.9	56.5	62.5	73.5
	1.5B	63.6	73.2	61.0	57.5	62.0	73.2
	3B	64.3	74.8	61.6	57.6	**62.6**	**74.0**
	7B	**64.4**	73.9	**62.2**	**57.9**	61.6	73.1
	14B(*Vanilla Self*)	62.0	72.7	59.1	55.0	59.6	72.3
	14B(*Post-SeqKD Self*)	63.3	73.5	60.4	57.1	58.2	73.7
Student: Qwen2.5-32B-Instruct							
Vanilla	–	57.3	67.9	53.9	51.3	56.5	67.6
SeqKD	–	64.1	74.5	61.2	57.4	62.9	73.7
Continued-SFT	–	66.0	75.4	65.0	58.1	64.5	74.5
DPO	0.5B	65.8	75.2	62.5	**60.8**	65.1	74.8
	1.5B	66.0	73.6	62.8	60.7	64.7	74.4
	3B	66.2	75.4	63.4	**60.8**	**65.5**	74.9
	7B	**66.9**	75.9	64.8	60.5	**65.5**	**76.1**
	14B	66.6	**76.1**	64.2	60.2	65.3	75.6
	32B(*Vanilla Self*)	65.4	75.2	62.8	59.0	63.9	75.1
	32B(*Post-SeqKD Self*)	65.8	75.0	**65.3**	58.4	63.1	76.0
Student: Qwen2.5-72B-Instruct							
Vanilla	–	60.0	70.1	57.1	53.7	59.8	67.3
SeqKD	–	65.1	75.3	63.7	56.7	65.2	75.4
Continued-SFT	–	66.5	76.3	64.5	59.5	65.0	75.9
DPO	0.5B	68.3	77.8	**66.1**	61.7	67.3	76.6
	1.5B	68.5	76.9	66.0	**63.6**	67.3	76.3
	3B	**68.6**	**77.9**	66.0	62.8	67.3	76.6
	7B	68.3	77.8	65.8	62.2	**67.5**	**77.2**
	14B	68.1	76.9	65.2	63.1	66.8	76.3
	32B	68.2	77.1	65.5	62.9	66.7	76.6
	72B(*Vanilla Self*)	68.3	77.3	65.9	62.3	65.9	76.6
	72B(*Post-SeqKD Self*)	68.0	76.5	**66.1**	61.9	65.1	76.4

Table 4: External pass@1 across four code-generation benchmarks. Weighted Avg. weights each benchmark by its number of problems: 1,140 each for BCB-C and BCB-I, 164 for HumanEval, and 257 for MBPP. All values are percentages. Bold marks the best result within each student and column.

Method	Reject Source	BCB-C	BCB-I	HumanEval	MBPP	Weighted Avg.
Student: Qwen2.5-7B-Instruct						
Base	–	42.3	29.1	84.1	79.0	42.8
SeqKD	–	43.4	32.5	83.3	79.1	44.6
Continued-SFT	–	43.9	33.6	82.3	79.1	45.2
DPO	0.5B	**45.2**	35.4	**84.8**	78.2	**46.6**
	1.5B	45.0	33.3	82.3	**80.2**	45.7
	3B	44.8	34.0	80.5	77.4	45.5
	7B(*Vanilla Self*)	42.6	33.2	79.9	76.7	44.1
	7B((*Post-SeqKD Self*))	40.9	30.1	80.5	79.4	42.4
Student: Qwen2.5-14B-Instruct						
Base	–	48.2	35.5	82.9	79.0	47.9
SeqKD	–	49.1	37.3	82.8	81.4	49.2
Continued-SFT	–	49.4	39.2	81.7	82.5	50.2
DPO	0.5B	50.0	**40.4**	82.3	82.5	51.0
	1.5B	49.3	40.2	**84.1**	**83.3**	50.8
	3B	**52.4**	40.3	83.5	82.5	**52.0**
	7B	48.9	39.2	81.7	81.3	49.9
	14B(*Vanilla Self*)	48.4	37.4	80.5	80.2	48.7
	14B(*Post-SeqKD Self*)	46.0	37.2	86.0	81.3	48.1
Student: Qwen2.5-32B-Instruct						
Base	–	52.8	38.6	89.6	**87.2**	52.3
SeqKD	–	53.2	39.9	86.7	86.8	52.8
Continued-SFT	–	53.0	41.8	87.2	86.4	53.5
DPO	0.5B	52.7	40.6	87.8	85.3	52.8
	1.5B	53.3	41.7	86.0	85.9	53.5
	3B	54.9	42.1	**90.9**	85.3	54.6
	7B	**56.3**	42.8	87.2	86.8	**55.4**
	14B	53.2	**44.0**	86.6	85.3	54.4
	32B(*Vanilla Self*)	54.4	41.1	87.8	86.4	53.9
	32B(*Post-SeqKD Self*)	51.0	41.4	88.4	84.8	52.4
Student: Qwen2.5-72B-Instruct						
Base	–	54.2	43.4	87.8	89.2	54.8
SeqKD	–	55.1	43.7	86.5	87.4	55.3
Continued-SFT	–	55.9	**44.3**	86.0	86.4	55.7
DPO	0.5B	55.8	43.7	87.8	88.6	55.8
	1.5B	56.1	43.2	**90.2**	89.4	55.9
	3B	55.7	42.5	89.6	**89.5**	55.4
	7B	56.8	43.9	87.8	89.0	56.3
	14B	**57.6**	43.9	89.0	89.0	**56.7**
	32B	56.6	42.9	89.6	88.2	55.8
	72B(*Vanilla Self*)	56.3	40.8	86.0	87.4	54.5
	72B(*Post-SeqKD Self*)	54.5	42.4	88.4	86.8	54.5

Table 5: MATH-500 and AIME avg@4 over four samples.

Method	Reject Source	MATH-500 avg@4	AIME avg@4
Teacher (DeepSeek-R1)	–	97.3	74.9
	Student: Qwen2.5-72B-Instruct		
Vanilla	–	84.0	15.8
SeqKD	–	91.0	37.9
Continued-SFT	–	91.2	37.5
DPO	0.5B	91.6	40.0
	7B	91.5	37.1
	14B	91.7	**40.4**
	32B	**92.7**	40.4
	72B(*Vanilla Self*)	91.8	35.4
	72B(*Post-SeqKD Self*)	91.8	37.5
	Llama-1B	91.7	34.2
	Llama-3B	92.0	39.6
	Llama-8B	92.0	35.4
	Student: Qwen2.5-32B-Instruct		
Vanilla	–	82.5	13.8
SeqKD	–	87.8	29.2
Continued-SFT	–	87.6	29.2
DPO	0.5B	88.4	32.1
	7B	88.4	**35.4**
	14B	89.1	32.1
	32B(*Vanilla Self*)	88.0	30.8
	32B(*Post-SeqKD Self*)	87.9	30.4
	Llama-1B	88.6	29.2
	Llama-3B	**89.5**	32.5
	Llama-8B	89.1	32.5
	Student: Qwen2.5-14B-Instruct		
Vanilla	–	78.7	13.3
SeqKD	–	81.5	18.8
Continued-SFT	–	82.1	19.6
DPO	0.5B	83.1	**22.9**
	1.5B	82.9	**22.9**
	3B	83.2	20.0
	7B	82.0	**22.9**
	14B(*Vanilla Self*)	81.3	20.0
	14B(*Post-SeqKD Self*)	80.9	19.6
	Llama-1B	83.9	20.8
	Llama-3B	**84.6**	20.4
	Llama-8B	83.9	21.3
	Student: Qwen2.5-7B-Instruct		
Vanilla	–	**76.0**	7.9
SeqKD	–	78.5	10.8
Continued-SFT	–	79.2	11.7
DPO	0.5B	79.9	16.7
	1.5B	80.2	**22.9**
	3B	79.6	11.7
	7B(*Vanilla Self*)	78.7	14.2
	7B(*Post-SeqKD Self*)	78.2	12.9

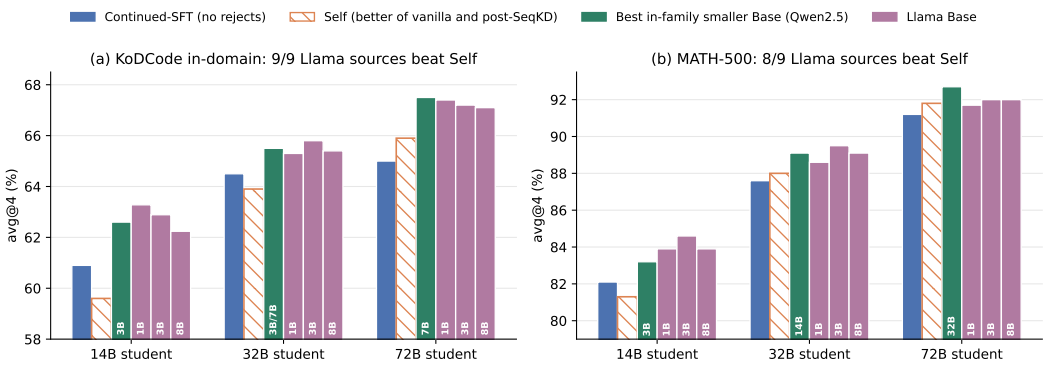

Figure 5: Cross-family reject sources. Bars report avg@4 on (a) KoDCode in-domain and (b) MATH-500 for the 14B, 32B, and 72B Qwen2.5 students. Self is the better of vanilla Self and post-SeqKD Self. Labels on the green bars identify the best Qwen2.5 smaller Base source; labels on the purple bars identify the Llama source.

C.1 Cross-Family Reject Sources

To test whether the smaller Base advantage depends on generating rejects within the student's model family, we replace the Qwen2.5 reject source with Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, or Llama-3.1-8B-Instruct for the 14B, 32B, and 72B Qwen2.5 students. All three are smaller Base sources for these students. Fig. 5 compares them with Continued-SFT, the better of the two Self constructions, and the best Qwen2.5 smaller Base source at the same student scale.

Llama rejects outperform Self in all nine code configurations and in eight of nine MATH-500 configurations; the exception is Llama-1B for the 72B student on MATH-500 (91.7% versus 91.8%).

Table 6: Complete randomized source assignment results. The Base share is the proportion of prompts assigned rejects from the indicated smaller Base source. All remaining prompts use vanilla Self rejects from the student-scale model.

Student	Smaller Base source	Base share	Code avg@4	Code pass@4
14B	Llama-3.2-1B-Instruct	0%	59.730	72.380
		25%	61.575	73.000
		50%	62.035	72.760
		75%	62.670	73.380
		100%	63.280	73.180
14B	Qwen2.5-3B-Instruct	0%	59.575	72.280
		25%	60.865	73.100
		50%	61.800	73.860
		75%	62.075	73.300
		100%	62.630	73.960
72B	Qwen2.5-7B-Instruct	0%	65.850	76.560
		50%	66.685	76.580
		100%	67.530	77.240

Table 7: KoDCode avg@4 across the reject-control ladder. Prompt-reassigned arms use real Base rejects reassigned across prompts. Each row is one completed prompt-reassignment experiment and reports its Continued-SFT, gibberish, reassigned, and aligned settings.

Student	Reject source	Continued-SFT	Gibberish	Prompt-reassigned	Aligned
7B	Qwen2.5-0.5B-Instruct	56.30	57.37	58.65	58.53
14B	Qwen2.5-3B-Instruct	60.89	60.92	61.97	62.63
32B	Qwen2.5-7B-Instruct	64.50	64.52	65.25	65.53
72B	Qwen2.5-7B-Instruct	65.00	65.24	66.36	67.53

D Construction Intervention Details

This section reports the complete results of the construction interventions in §3.3: randomized source assignment (P1), prompt reassignment and lexical permutation (P2), and reference likelihood reselection (P3), together with the natural source coupling analysis that motivates P3. Within each intervention, all arms share the student, prompts, chosen responses, SeqKD initialization, optimizer schedule, and evaluation protocol; only the reject construction changes.

D.1 Randomized Source Assignment

We implement the randomized source assignment operator in Eq. 10 with three interventions (Tab. 6). For the 14B student, rejects come from vanilla Self or from either Llama-3.2-1B-Instruct or Qwen2.5-3B-Instruct; for the 72B student, they come from vanilla Self or Qwen2.5-7B-Instruct. In every intervention, avg@4 rises with the Base share: from 59.730 to 63.280 with Llama-3.2-1B-Instruct, from 59.575 to 62.630 with Qwen2.5-3B-Instruct, and from 65.850 to 67.530 with Qwen2.5-7B-Instruct, passing through 66.685 at 50%. Source composition therefore controls the attainable utility continuously across model families and student scales.

D.2 Prompt Reassignment and Lexical Permutation

Tab. 7 reports the Continued-SFT, Gibberish, prompt-reassigned, and aligned endpoints at every student scale, and Fig. 6 shows the corresponding implicit reward trajectories.

At every student scale, lexical permutation starts from the corresponding prompt-reassigned Base rejects. It preserves the generated code-token inventory while destroying token order, Python syntax, coherent program semantics, and prompt correspondence (Tab. 8). Lexical permutation improves avg@4 over Gibberish at all four scales and recovers between 40.0% and 92.5% of the real prompt-reassigned gain. This identifies code-domain lexical structure as a cross-scale component of Base

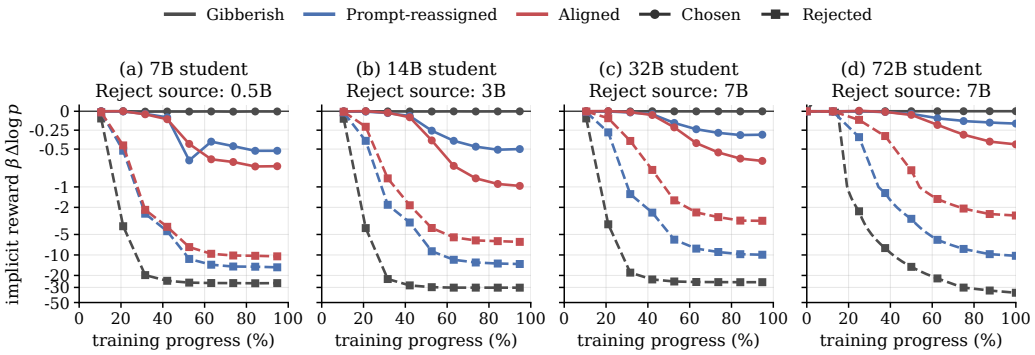

Figure 6: Implicit reward dynamics for gibberish, prompt reassigned, and prompt-matched Base rejects at each student scale. Gibberish is suppressed most strongly but does not recover the utility of real Base rejects.

Table 8: Cross-scale lexical-structure interventions. Recovery is the Lexical gain over Gibberish divided by the real prompt-reassigned gain over Gibberish.

Student	Gibberish	Lexical	Real reassigned	Lexical − Gibberish	Recovery
7B	57.370	57.900	58.645	+0.530	41.6%
14B	60.920	61.650	61.970	+0.730	69.5%
32B	64.520	65.195	65.250	+0.675	92.5%
72B	65.235	65.685	66.360	+0.450	40.0%

reject utility. At 14B, an additional AST scrambling intervention reaches 61.840 avg@4. It preserves Python syntax and common AST constructs while disrupting program semantics.

D.3 Reference Coupling of Natural Sources

Tab. 9 reports every point used in the natural-source analysis. For each student scale, the likelihood measurements use the same 24,248 preference prompts and the exact SeqKD model used as the DPO reference. Downstream utility uses the same 5,000 held-out problems and four samples per problem. The negative association is preserved under prompt standardization and every leave-one-source-out analysis (Tab. 10). Reference likelihood therefore organizes the Base-versus-Self boundary.

D.4 Reference Likelihood Reselection

For each source, we sample eight candidates for each preference prompt and score their length-normalized likelihood under the 14B SeqKD reference within the prompt-specific bank. Shared Gumbel draws construct reference-repelled, native, and reference-attracted datasets from each fixed bank. A prompt is then kept only if all 3 selected rejects run to completion, pass exactly the same fraction of unit tests, and differ in length by at most 32 tokens. We retain the intersection of prompts satisfying these criteria across all reject sources, yielding a shared set of 20,136 prompts used for every source and all three selection rules. Repelled selection improves on Native for all five sources (Tab. 11). Attracted selection reduces utility for all four smaller Base sources. Combining the 14B repel effect with the attracted effect from each smaller source yields positive double-dissociation interactions of +0.815, +1.000, +0.695, and +1.090 points for 0.5B, 1.5B, 3B, and 7B sources. Across the 15 arms, after centering coupling and utility within each generator, their Pearson and Spearman associations are 0.868 and 0.936, respectively.

Table 9: Natural reject-source measurements underlying Fig. 4a. Each source is scored on the same 24,248 preference prompts under the corresponding student's SeqKD reference. Mean log likelihood is length normalized. Prompt z is the mean reference likelihood after standardization within each prompt. Utility is KoDCode avg@4 on 5,000 held-out problems.

Student	Reject source	Source class	Mean log S/token	Prompt z	Mean tokens	avg@4
7B	Qwen2.5-0.5B-Instruct	Subscale Base	-0.3725	-1.048	466.7	58.525
7B	Qwen2.5-1.5B-Instruct	Subscale Base	-0.2527	-0.113	262.5	57.395
7B	Qwen2.5-3B-Instruct	Subscale Base	-0.2851	-0.435	291.8	57.705
7B	Qwen2.5-7B-Instruct	Vanilla Self	-0.1790	$+0.418$	284.9	54.550
7B	post-SeqKD 7B	Post-SeqKD Self	-0.0915	$+1.179$	243.7	52.745
14B	Qwen2.5-0.5B-Instruct	Subscale Base	-0.3848	-1.074	466.7	62.525
14B	Qwen2.5-1.5B-Instruct	Subscale Base	-0.2680	-0.140	262.5	61.995
14B	Qwen2.5-3B-Instruct	Subscale Base	-0.2922	-0.398	291.8	62.630
14B	Qwen2.5-7B-Instruct	Subscale Base	-0.2206	$+0.179$	284.9	61.645
14B	Qwen2.5-14B-Instruct	Vanilla Self	-0.2255	$+0.127$	304.7	59.575
14B	post-SeqKD 14B	Post-SeqKD Self	-0.0874	$+1.306$	240.9	58.155
32B	Qwen2.5-0.5B-Instruct	Subscale Base	-0.3957	-1.130	466.7	65.135
32B	Qwen2.5-1.5B-Instruct	Subscale Base	-0.2784	-0.203	262.5	64.700
32B	Qwen2.5-3B-Instruct	Subscale Base	-0.3045	-0.467	291.8	65.515
32B	Qwen2.5-7B-Instruct	Subscale Base	-0.2234	$+0.169$	284.9	65.530
32B	Qwen2.5-14B-Instruct	Subscale Base	-0.2570	-0.102	304.7	65.285
32B	Qwen2.5-32B-Instruct	Vanilla Self	-0.1921	$+0.422$	266.4	63.880
32B	post-SeqKD 32B	Post-SeqKD Self	-0.0834	$+1.312$	242.0	63.070
72B	Qwen2.5-0.5B-Instruct	Subscale Base	-0.4034	-1.204	466.7	67.290
72B	Qwen2.5-1.5B-Instruct	Subscale Base	-0.2840	-0.273	262.5	67.300
72B	Qwen2.5-3B-Instruct	Subscale Base	-0.3130	-0.553	291.8	67.270
72B	Qwen2.5-7B-Instruct	Subscale Base	-0.2290	$+0.096$	284.9	67.530
72B	Qwen2.5-14B-Instruct	Subscale Base	-0.2685	-0.213	304.7	66.800
72B	Qwen2.5-32B-Instruct	Subscale Base	-0.2262	$+0.124$	266.4	66.670
72B	Qwen2.5-72B-Instruct	Vanilla Self	-0.1452	$+0.754$	239.6	65.850
72B	post-SeqKD 72B	Post-SeqKD Self	-0.0821	$+1.269$	239.3	65.070

Table 10: Cross-source association between mean reference likelihood and downstream avg@4. Exact p-values enumerate all assignments of the observed utilities to sources; p_- is the one-sided test for a negative association. Prompt z uses within-prompt standardized reference likelihood. Leave-one-out (LOO) reports the range of Pearson correlations after removing each source in turn.

Student	n	r	ρ	p_-	p_2	Prompt-z r	LOO r range
7B	5	-0.967	-1.000	0.0083	0.0083	-0.968	$[-0.988, -0.920]$
14B	6	-0.868	-0.886	0.0083	0.0083	-0.871	$[-0.909, -0.645]$
32B	7	-0.752	-0.464	0.0298	0.0484	-0.770	$[-0.863, -0.405]$
72B	8	-0.836	-0.690	0.0057	0.0076	-0.842	$[-0.931, -0.684]$

Table 11: Reference-likelihood reselection for the Qwen2.5-14B-Instruct student. Coupling entries are the negative mean length-normalized log likelihood under the 14B SeqKD reference, so larger values indicate lower reference coupling. Utility entries are KoDCode avg@4; the final three columns give the within-source contrasts. Native arms are trained on the reselection candidate banks, so their values differ from the natural source endpoints in Tab. 9.

	Coupling ($-\log S$/token)			avg@4			Contrast		
Reject source	Repelled	Native	Attracted	Repelled	Native	Attracted	R $-$ N	A $-$ N	R $-$ A
Qwen2.5-0.5B-Instruct	0.690	0.519	0.359	63.335	63.075	62.750	$+0.260$	-0.325	$+0.585$
Qwen2.5-1.5B-Instruct	0.487	0.344	0.238	63.135	62.390	61.880	$+0.745$	-0.510	$+1.255$
Qwen2.5-3B-Instruct	0.492	0.363	0.260	63.120	62.860	62.655	$+0.260$	-0.205	$+0.465$
Qwen2.5-7B-Instruct	0.320	0.243	0.183	62.100	62.070	61.470	$+0.030$	-0.600	$+0.630$
Qwen2.5-14B-Instruct	0.363	0.263	0.191	59.905	59.415	59.455	$+0.490$	$+0.040$	$+0.450$

E Additional Explanation Boundaries

The direct construction interventions provide the main evidence. This section reports complementary diagnostics that test whether optimization summaries recover the same source ordering.

E.1 DPO Diagnostic Definitions

We define the DPO diagnostics used in the fixed-pair and RC-DPO analyses. DPO induces the implicit reward (Rafailov et al., 2023; Yuan et al., 2025)

$$\widehat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}. \tag{14}$$

We write $\widehat{r}_\theta^+$ and $\widehat{r}_\theta^-$ for the chosen and rejected implicit rewards. At DPO initialization, $\pi_{\theta_0} = \pi_{\mathrm{ref}}$, so every pair has zero implicit margin and logistic weight $1/2$. Define the mean chosen field and the mean rejected field induced by source q as

$$F^+ = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log \pi_{\theta_0}(y_i^+ \mid x_i), \qquad F_q^- = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log \pi_{\theta_0}(y_{i,q}^- \mid x_i). \tag{15}$$

The initialization gradient is

$$\nabla_\theta \mathcal{L}_{\mathrm{DPO}}(\theta_0) = -\frac{\beta}{2}\left(F^+ - F_q^-\right). \tag{16}$$

For exact random sampling from the reference, the score-function identity gives

$$\mathbb{E}_{y \sim \pi_{\mathrm{ref}}(\cdot|x)}[\nabla_\theta \log \pi_{\theta_0}(y \mid x)] = 0. \tag{17}$$

For the empirical field measurement, let $L_{i,q}^-$ denote the number of rejected-response tokens, and let P denote a fixed CountSketch projection of the score gradient with respect to the final-layer parameters. We report the per-token rejected-score field

$$\widetilde{F}_q^- = \frac{1}{n}\sum_{i=1}^{n} P\left(\frac{1}{L_{i,q}^-}\nabla_\theta \log \pi_{\theta_0}(y_{i,q}^- \mid x_i)\right), \qquad N_q^- = \|\widetilde{F}_q^-\|_2. \tag{18}$$

Under gradient descent, $-F_q^-$ is the first-step component that lowers rejected likelihood. The projection norm is used only for within-student comparisons under the same tangent space.

For a fixed chosen or rejected response, define the sequence log-likelihood displacement at checkpoint t as

$$\Delta\ell_{i,t}^{\pm} = \log \pi_{\theta_t}(y_i^{\pm} \mid x_i) - \log \pi_{\theta_0}(y_i^{\pm} \mid x_i). \tag{19}$$

Following RC-DPO (Chen et al., 2026b), Pathway III contains pairs for which $\Delta\ell_{i,t}^+ \geq 0$ and $\Delta\ell_{i,t}^- \leq 0$.

RC-DPO rescales the backward incentives that produce these dynamics. For batch-mean sequence log probabilities $\ell_t^{\pm}$, standard DPO uses $z_t = (\ell_t^+ - \ell_{\mathrm{ref}}^+) - (\ell_t^- - \ell_{\mathrm{ref}}^-)$ and $\mathcal{L}_t = -\log \sigma(\beta z_t)$. Let c_t and d_t be log-EMA-smoothed norms of $\nabla\ell_t^+$ and $\nabla\ell_t^-$ on a trainable output-head subspace, and set $\alpha_t = \sqrt{d_t/c_t}$. RC-DPO uses

$$\widetilde{\ell}_t^+ = \alpha_t \ell_t^+ + (1-\alpha_t)\,\mathrm{sg}(\ell_t^+), \qquad \widetilde{\ell}_t^- = \alpha_t^{-1}\ell_t^- + (1-\alpha_t^{-1})\,\mathrm{sg}(\ell_t^-). \tag{20}$$

The forward log probabilities are unchanged, while the backward pass scales the chosen and rejected gradients by α_t and α_t^{-1}, respectively.

E.2 Reject-Score Fields

The student-local score-field analysis uses 128 shared prompts and two independent CountSketch projections. Sequence-sum and per-token norms are reported separately, and all comparisons are made within a student and normalization view. The complete 14B and 72B measurements appear in Tab. 12.

Table 12: Student-local reject-score fields at the shared preference- optimization initialization. Norms are averaged over two independent sketches. Differences are source minus post-SeqKD Self. Per-token norms and differences are scaled by 10^3.

Reject source	Mean-field norm	Δ field	Coherence	Per-token norm	Δ per-token
14B student					
post-SeqKD Self (14B)	1.559	–	0.171	9.47	–
0.5B	3.146	+1.587	0.171	16.85	+7.38
1.5B	2.119	+0.561	0.153	16.96	+7.49
3B	2.829	+1.270	0.199	17.69	+8.22
vanilla 7B	2.163	+0.604	0.166	13.31	+3.84
vanilla Self (14B)	2.278	+0.720	0.182	13.47	+4.00
72B student					
post-SeqKD Self (72B)	1.200	–	0.200	7.14	–
0.5B	2.146	+0.946	0.171	11.51	+4.38
1.5B	1.565	+0.365	0.161	11.93	+4.79
3B	1.865	+0.665	0.191	11.63	+4.49
vanilla 7B	1.499	+0.299	0.168	9.06	+1.93
14B	1.599	+0.399	0.174	9.29	+2.15
32B	1.548	+0.348	0.182	9.63	+2.49
vanilla Self (72B)	1.669	+0.469	0.227	11.81	+4.67

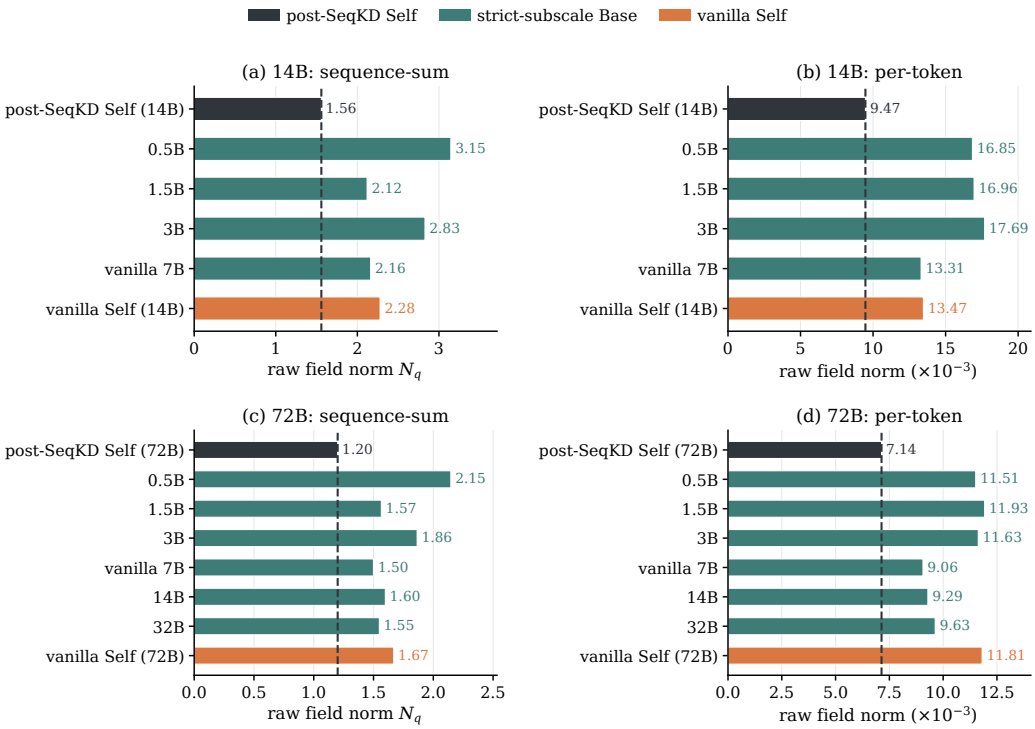

Figure 7: Empirical rejected-response fields at the shared DPO initialization for the 14B and 72B students. Post-SeqKD Self has an attenuated field relative to smaller Base sources, while vanilla Self overlaps the smaller Base range.

Post-SeqKD Self has the weakest aggregate reject field at both verified student scales. Vanilla Self restores field strength but remains closer to post-SeqKD Self in field space and execution failures. Smaller Base sources combine an active field with greater displacement. Since vanilla Self overlaps the Base field range while producing lower downstream utility, field strength alone does not explain the source class gap.

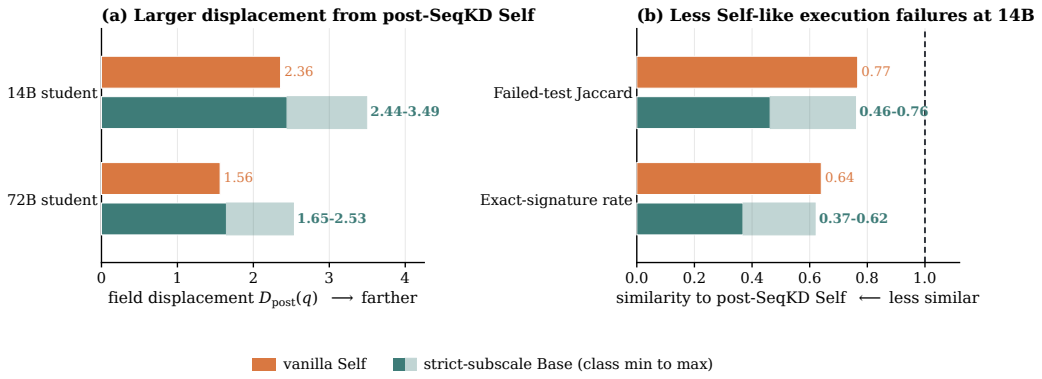

Figure 8: Differences between Self and smaller Base reject classes. The panels report score-field displacement from post-SeqKD Self and overlap in failed execution tests. These measurements characterize the source classes but do not rank their downstream utility.

Table 13: Endpoint fixed-pair dynamics and KoDCode outcomes. Path III is the percentage of prompts that preserve the chosen response while suppressing the source's own reject. Regressions count problems solved by the shared preference-optimization initialization but not by the final model. The lower panel reports descriptive correlations across the seven source endpoints.

Reject	Chosen $\Delta\ell$	Path III (%)	Regressions
post-SeqKD Self	-6.79	5.3	288
0.5B	-6.67	11.7	262
1.5B	-10.92	7.8	324
3B	-10.19	12.5	247
7B	-8.80	12.3	215
vanilla Self	-8.23	9.6	299

Fixed-pair statistic	Gain r/ρ	Recoveries r/ρ	Regressions r/ρ
Chosen mean displacement	$+0.023/+0.071$	$-0.014/+0.071$	$-0.041/-0.107$
Chosen preservation rate	$+0.505/+0.643$	$-0.106/-0.036$	$-0.716/-0.786$
Pathway-II rate	$-0.511/-0.667$	$+0.109/+0.072$	$+0.726/+0.811$
Own-reject suppression depth	$+0.352/+0.107$	$+0.292/+0.321$	$-0.137/-0.071$
Cross-source suppression	$+0.360/+0.429$	$+0.204/+0.536$	$-0.233/-0.179$
Own-specific suppression	$+0.289/-0.036$	$+0.278/+0.357$	$-0.075/+0.071$

Here r and ρ denote Pearson and Spearman correlations, respectively. These seven-source associations describe completed runs; they are not prompt-level selection results.

E.3 Fixed-Pair Dynamics and RC-DPO

Fixed-pair displacement and Pathway III follow Eq. 19. Every checkpoint is evaluated on the same bank of 512 prompts and fixed responses, so movement across checkpoints reflects policy change rather than resampling. Endpoint statistics and KoDCode regressions for the 14B source sweep are reported in Tab. 13.

RC-DPO directly changes the chosen and rejected gradient balance. For the 3B Base and vanilla Self sources it increases fixed-bank Pathway III, while the held out changes do not follow a common positive direction. The corresponding fixed-pair effects and source by objective interactions appear in Tab. 14. The implicit reward trajectories are shown in Fig. 10, with absolute held-out endpoints in Tab. 15.

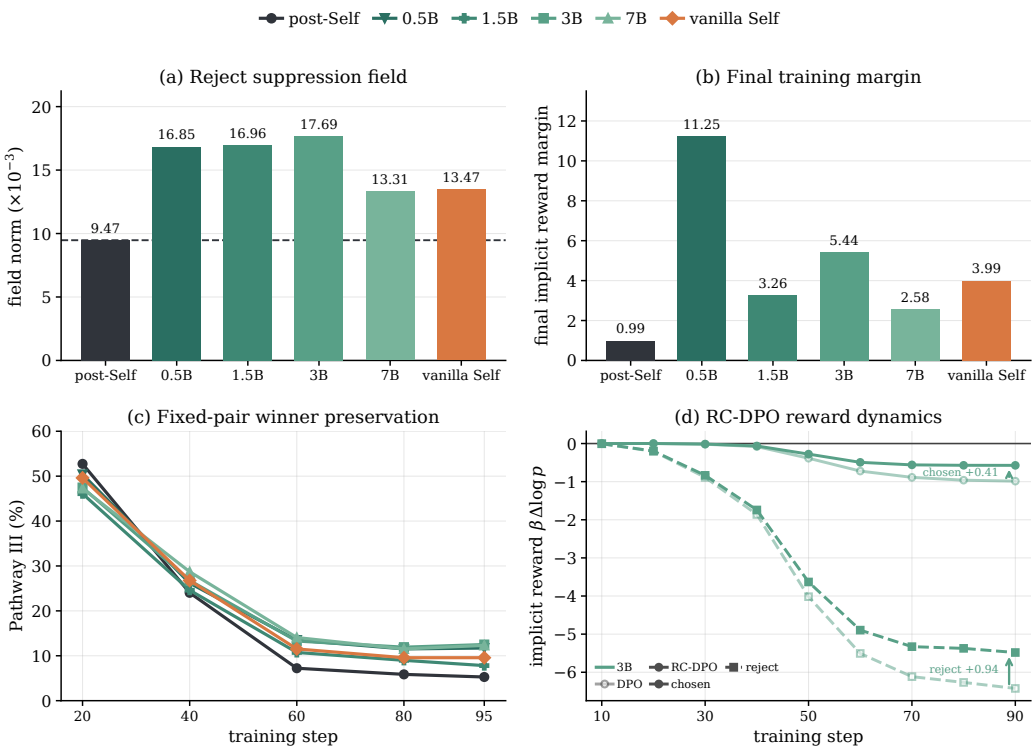

Figure 9: Training diagnostics for the 14B student. Panels report the initialization reject field, final implicit reward margin, winner preservation on fixed pairs, and the effect of RC-DPO on chosen and rejected rewards. None consistently separates smaller Base rejects from both Self controls.

Table 14: Reject source × objective factorial on the fixed 512-prompt bank. Likelihood displacements are sequence sums relative to the shared SeqKD reference. RC effects and interactions are percentage points.

Reject source	Objective	Chosen $\Delta\ell$	Own-reject $\Delta\ell$	Path III (%)	RC effect on Path III
post-SeqKD Self	DPO	−6.787	−16.337	5.3	–
	RC-DPO	−11.650	−22.545	1.0	−4.30
3B Base	DPO	−10.185	−63.464	12.5	–
	RC-DPO	−5.890	−53.974	18.6	+6.05
vanilla Self	DPO	−8.230	−44.353	9.6	–
	RC-DPO	−5.711	−39.027	12.9	+3.32

Path-III source contrast	DPO gap	RC-DPO gap	Source × RC interaction
3B Base − post-Self	+7.23	+17.58	+10.35
vanilla Self − post-Self	+4.30	+11.91	+7.62
3B Base − vanilla Self	+2.93	+5.66	+2.73

Every cell uses identical response IDs and token counts; no response is truncated. All six policies suppress their own reject on 100% of fixed pairs. RC-DPO raises Path III for the raw sources but lowers it for Self, so objective calibration widens rather than removes the source gap.

Table 15: Held-out performance under standard DPO and RC-DPO for the 14B student. Values are absolute percentages.

Reject source	Objective	KoDCode pass@1	BCB-C avg@4	BCB-C pass@4
post-SeqKD Self	DPO	63.32	45.70	65.35
	RC-DPO	62.16	44.96	64.65
3B Base	DPO	64.34	51.12	65.44
	RC-DPO	64.34	50.86	64.91
vanilla Self	DPO	61.96	49.06	63.95
	RC-DPO	62.52	48.84	64.12

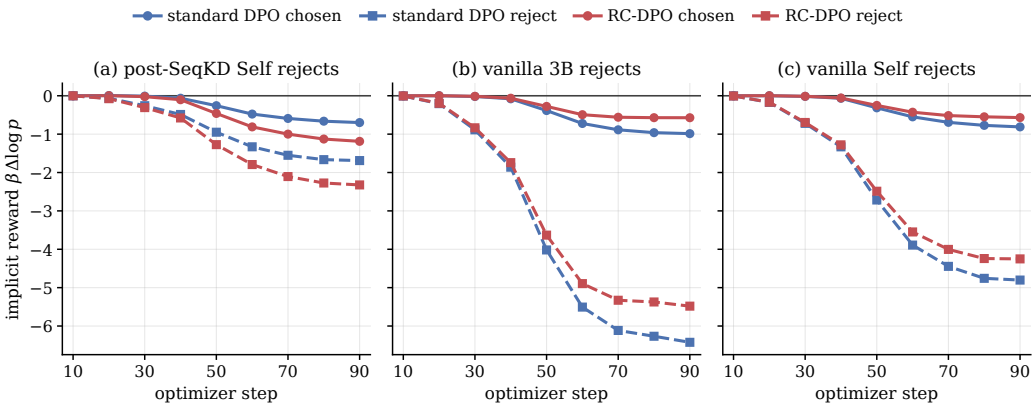

Figure 10: Reward dynamics under standard DPO and RC-DPO for the 14B student. Panels show post-SeqKD Self, 3B Base, and vanilla Self rejects. Solid lines track chosen implicit reward and dashed lines track rejected implicit reward. RC-DPO changes the likelihood trajectory in a source-dependent manner.

F Reject Construction and Optimization Analyses

F.1 Illustrative Code Example with Two Features

Consider returning the maximum element of a nonempty integer list, including lists whose elements are all negative. The following constructed responses illustrate two distinct features.

Correct response y_+.

```
def solve(xs):
    return max(xs)
```

Incorrect response y_e: minimum selection.

```
def solve(xs):
    return min(xs)
```

Incorrect response y_c: an erroneous zero lower bound.

```
def solve(xs):
    return max(0, max(xs))
```

On `xs = [-3, -1]`, the outputs are -1, -3, and 0, respectively. Define a two-dimensional feature vector whose first coordinate distinguishes maximum from minimum selection and whose second coordinate marks the zero lower bound:

$$\phi_+ = (1, 0), \qquad \phi_e = (-1, 0), \qquad \phi_c = (1, 1), \qquad \phi_0 = (0, 0). \tag{21}$$

Here y_0 is a task-neutral text. With $v = (v_1, v_2)$, the change in response score is $v^\top \phi_y$. The parameter v_1 favors maximum selection, while v_2 favors the extra zero lower bound.

The chosen–rejected feature differences are

$$\delta_e = (2, 0), \qquad \delta_c = (0, -1), \qquad \delta_0 = (1, 0). \tag{22}$$

At initialization, the corresponding single-pair DPO increments are $\eta\beta(1, 0)$, $\frac{\eta\beta}{2}(0, -1)$, and $\frac{\eta\beta}{2}(1, 0)$. The pair involving y_e reinforces maximum selection. The pair involving y_c cancels along the shared maximum-selection coordinate and corrects the error along the second coordinate. Thus sharing a feature and supplying a useful contrast can occur within the same rejected response.

In this example, with $Q_0 = \delta_{y_0}$ and $\widehat{u} = (\widehat{u}_1, \widehat{u}_2)$, the transfer scores are $k_e = \widehat{u}_1$ and $k_c = -\widehat{u}_1 - \widehat{u}_2$. Their values depend on the utility gradient under the reference. The two-dimensional assignment therefore separates overlap on one feature from the response's net contribution across all features.

F.2 Derivation of the Vector-Feature DPO Model

F.2.1 Policy Construction

Let $S(y \mid x)$ be the frozen SeqKD reference policy. For each prompt x, let $\mathcal{Y}_x$ denote its response support, on which $S(y \mid x) > 0$. We assign each response a fixed feature vector $\phi(x, y) \in \mathbb{R}^d$ and introduce a trainable coefficient vector $v \in \mathbb{R}^d$, initialized at zero. Define the response score

$$f_v(x, y) = \log S(y \mid x) + v^\top \phi(x, y). \tag{23}$$

The score change is linear in v. Normalizing the exponentiated scores gives

$$\begin{aligned} Z_v(x) &= \sum_{y' \in \mathcal{Y}_x} S(y' \mid x) \exp\big(v^\top \phi(x, y')\big), \\ \pi_v(y \mid x) &= \frac{S(y \mid x) \exp\big(v^\top \phi(x, y)\big)}{Z_v(x)}. \end{aligned} \tag{24}$$

Thus, π_v is a softmax over the scores f_v. Equivalently, $Z_v(x) = \mathbb{E}_{y' \sim S(\cdot|x)}[\exp(v^\top \phi(x, y'))]$.

We assume that $Z_v(x)$ is finite in the parameter region considered and that differentiation can be interchanged with the sums below. Finite response support with finite features satisfies these conditions; bounded features also ensure a finite normalizer on countable support.

At initialization,

$$Z_0(x) = \sum_y S(y \mid x) = 1, \qquad \pi_0(y \mid x) = S(y \mid x). \tag{25}$$

The model therefore starts from the same reference policy as DPO. During training, S and ϕ remain fixed, and only v changes.

F.2.2 Connection to Local Linearization

Let π_θ denote the original network, with SeqKD parameters θ_0 and $S = \pi_{\theta_0}$. When v represents displacement in the same parameter space, the corresponding network parameters are $\theta_0 + v$. For a twice continuously differentiable log probability, Taylor expansion around θ_0 gives

$$\log \pi_{\theta_0 + v}(y \mid x) = \log S(y \mid x) + v^\top g(x, y) + O(\|v\|^2), \tag{26}$$

where

$$g(x, y) = \nabla_\theta \log \pi_\theta(y \mid x)|_{\theta=\theta_0}. \tag{27}$$

The expansion is pointwise in (x, y); a uniform remainder requires corresponding uniform derivative bounds.

Choosing $\phi = g$ uses the network's initial score gradients as fixed response features. Exponentiating the linear part of Eq. 26 gives the unnormalized weights $S(y \mid x) \exp(v^\top g(x, y))$. Normalizing these weights yields Eq. 24. This construction uses initial gradient features throughout training, following the local linearization perspective on neural training (Lee et al., 2019).

To verify its first-order correspondence with the network, first differentiate the log normalizer:

$$\begin{aligned} \nabla_v \log Z_v(x) &= \frac{\sum_{y'} S(y' \mid x) \exp(v^\top \phi(x, y')) \phi(x, y')}{Z_v(x)} \\ &= \mathbb{E}_{y' \sim \pi_v(\cdot|x)}[\phi(x, y')]. \end{aligned} \tag{28}$$

Consequently,

$$\nabla_v \log \pi_v(y \mid x) = \phi(x, y) - \mathbb{E}_{y' \sim \pi_v(\cdot|x)}[\phi(x, y')]. \tag{29}$$

For the choice $\phi = g$, the reference-weighted feature mean is zero:

$$\begin{aligned} \mathbb{E}_{y \sim S(\cdot|x)}[g(x, y)] &= \sum_y \pi_{\theta_0}(y \mid x)\ \nabla_\theta \log \pi_\theta(y \mid x)|_{\theta=\theta_0} \\ &= \sum_y \nabla_\theta \pi_\theta(y \mid x)|_{\theta=\theta_0} \\ &= \nabla_\theta \sum_y \pi_\theta(y \mid x)\Big|_{\theta=\theta_0} = 0. \end{aligned} \tag{30}$$

Here the response support is fixed locally in θ, and the interchange of differentiation and summation is assumed valid. Combining Eq. 29 with Eq. 30 gives

$$\nabla_v \log \pi_v(y \mid x)|_{v=0} = g(x, y) = \nabla_\theta \log \pi_\theta(y \mid x)|_{\theta=\theta_0}. \tag{31}$$

Thus the constructed policy matches both the network's response probabilities and its log-probability gradients at initialization.

The modeling assumption is that these response features remain fixed as v is trained. The subsequent derivations are exact for this fixed-feature family. Its approximation to the original network beyond initialization depends on the higher-order terms in Eq. 26.

F.2.3 Shared Features and Response Contrasts

For a chosen response y^+ and a reject y^- to the same prompt, define

$$\delta(x, y^+, y^-) = \phi(x, y^+) - \phi(x, y^-). \tag{32}$$

Their log-probability ratio satisfies

$$\log \frac{\pi_v(y^+ \mid x)}{\pi_v(y^- \mid x)} = \log \frac{S(y^+ \mid x)}{S(y^- \mid x)} + v^\top \delta. \tag{33}$$

If the two responses have identical feature values in coordinate j, then $\delta_j = 0$, and changing v_j does not change their probability ratio. If their feature values differ, changing v_j changes the ratio according to δ_j. A pair can therefore share features in some coordinates and contrast in others.

For example, the illustrative assignment $\phi(x, y^+) = (1, 1)$ and $\phi(x, y^-) = (1, -1)$ gives $\delta = (0, 2)$. The first coordinate changes both response scores equally, whereas the second changes their relative score. Such assignments illustrate the model's geometry. When ϕ consists of network score gradients, its coordinates instead represent sensitivities to network parameters. §F.1 gives a concrete code example with two features.

F.2.4 DPO Margin and Gradient

For a preference triple (x, y^+, y^-), define the reference-relative DPO margin

$$m_v = \beta \left[\log \frac{\pi_v(y^+ \mid x)}{S(y^+ \mid x)} - \log \frac{\pi_v(y^- \mid x)}{S(y^- \mid x)} \right], \qquad \beta > 0. \tag{34}$$

By Eq. 24,

$$\log \frac{\pi_v(y \mid x)}{S(y \mid x)} = v^\top \phi(x, y) - \log Z_v(x). \tag{35}$$

The chosen and rejected responses share the same prompt, so their normalizers cancel:

$$\begin{aligned} m_v &= \beta \left[v^\top \phi(x, y^+) - \log Z_v(x) - v^\top \phi(x, y^-) + \log Z_v(x) \right] \\ &= \beta v^\top \delta. \end{aligned} \tag{36}$$

A positive margin means that the chosen-to-rejected probability ratio has increased relative to the reference.

The pairwise DPO loss is

$$\ell(v; x, y^+, y^-) = -\log \sigma(m_v), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}. \tag{37}$$

Using

$$\frac{d}{dm} \big[-\log \sigma(m) \big] = -\sigma(-m), \qquad \nabla_v m_v = \beta \delta, \tag{38}$$

the pairwise gradient is

$$\nabla_v \ell(v; x, y^+, y^-) = -\beta \sigma(-\beta v^\top \delta) \delta. \tag{39}$$

Hence a gradient-descent step on this pair changes v by

$$-\eta \nabla_v \ell = \eta \beta \sigma(-\beta v^\top \delta) \delta. \tag{40}$$

The update follows the feature difference δ. Its scalar weight decreases continuously as the margin increases, since

$$\frac{d}{dm} \sigma(-m) = -\sigma(-m)\sigma(m) < 0. \tag{41}$$

At initialization, every pair has zero margin and weight $\sigma(0) = 1/2$.

F.2.5 The Reject-Dependent Update Field

Using the preference-prompt law p, the verified chosen-response law T, and the reject law Q induced by the construction, define the preference law

$$\mathcal{D}_Q(x, y^+, y^-) = p(x)T(y^+ \mid x)Q(y^- \mid x), \tag{42}$$

and write $\mathbb{E}_Q$ for expectation under this law. The population DPO objective is

$$\mathcal{L}_Q(v) = \mathbb{E}_Q[\ell(v; x, y^+, y^-)], \tag{43}$$

where the response feature contrast δ depends on the sampled triple as in Eq. 32. Assuming differentiation and expectation can be interchanged, Eq. 39 gives

$$\nabla_v \mathcal{L}_Q(v) = -\beta \mathbb{E}_Q\left[\sigma(-\beta v^\top \delta)\delta\right]. \tag{44}$$

Define the negative-gradient field

$$F_Q(v) = -\nabla_v \mathcal{L}_Q(v) = \beta \mathbb{E}_Q\left[\sigma(-\beta v^\top \delta)\delta\right]. \tag{45}$$

Full-batch gradient descent therefore follows

$$v_{t+1}^Q = v_t^Q + \eta_t F_Q(v_t^Q), \qquad v_0^Q = 0, \tag{46}$$

with positive step sizes η_t. For a fixed finite preference dataset, replacing $\mathbb{E}_Q$ by the empirical average yields the corresponding empirical objective and exact full-batch update.

At initialization,

$$F_Q(0) = \frac{\beta}{2}\mathbb{E}_Q[\delta]. \tag{47}$$

For a comparison reject distribution Q_0, define

$$\bar{\phi}_Q(x) = \mathbb{E}_{y^- \sim Q(\cdot|x)}[\phi(x, y^-)], \qquad \bar{\phi}_0(x) = \mathbb{E}_{y^- \sim A(\cdot|x)}[\phi(x, y^-)]. \tag{48}$$

Because the prompt and chosen distributions are shared, their contributions cancel in the initial field difference:

$$F_Q(0) - F_{Q_0}(0) = \frac{\beta}{2}\mathbb{E}_{x\sim p}\left[\bar{\phi}_0(x) - \bar{\phi}_Q(x)\right]. \tag{49}$$

This identity connects reject construction to the initial update used in the transfer analysis.

More generally, with p, T, and ϕ fixed, reject construction affects $F_Q(v)$ through the induced distribution of δ. At initialization, only its mean enters the update. As training proceeds, the margin-dependent weights make the update depend on its broader distribution.

F.3 Vector DPO Dynamics and the Finite-Horizon Bound

Policy family and feature gradients. Use the policy and update field derived in §F.2, with the stated regularity conditions. Write $g_v(x, y) = \nabla_v \log \pi_v(y \mid x)$, whose centered-feature expression is given in Eq. 29. For the bounds below, assume finite response sets and bounded features, or response spaces on which the corresponding uniform bounds exist.

Bounds on the complete vector field. The field in Eq. 5 satisfies

$$\|F_Q(v)\| \le M_Q := \beta \mathbb{E}_Q\|\delta\|, \tag{50}$$

$$\nabla F_Q(v) = -\beta^2 \mathbb{E}_Q\left[\sigma(-\beta v^\top \delta)\sigma(\beta v^\top \delta)\delta\delta^\top\right]. \tag{51}$$

In particular, a global Lipschitz constant is

$$L_Q = \frac{\beta^2}{4}\left\|\mathbb{E}_Q[\delta\delta^\top]\right\|_{\mathrm{op}}. \tag{52}$$

This matrix retains interactions among all feature coordinates. For $\|\delta\| \le D$, common bounds over all compared sources are $M = \beta D$ and $L = \beta^2 D^2/4$. For a fixed source family, the maxima of M_Q and L_Q give tighter common constants.

For the utility U defined in §3.2,

$$u = \mathbb{E}_{x\sim p_{\mathrm{eval}}} \mathrm{Cov}_{y\sim S(\cdot|x)}\big(r(x, y), \phi(x, y)\big). \tag{53}$$

If $\|\phi\| \le B$, then $\|g_v\| \le 2B$ and

$$\nabla^2 U(v) = \mathbb{E}_{x\sim p_{\mathrm{eval}}, y\sim\pi_v}\left[(r(x, y) - U_x(v))g_v(x, y)g_v(x, y)^\top\right], \tag{54}$$

where $U_x(v) = \mathbb{E}_{\pi_v(\cdot|x)} r(x, y)$. Hence $L_U = 4B^2$ is a valid global bound for rewards in $[0, 1]$. A smaller local curvature bound can be used on the trajectory ball.

Formal conditions. We prove Proposition 1 in the following form: the compared constructions Q and Q_0 share the constants M, L, and L_U above, $u = \nabla U(0) \neq 0$, and Eq. 46 runs for H steps with step sizes $\eta_t > 0$ and total step size $\tau_H = \sum_{t=0}^{H-1} \eta_t$. The conclusion is Eq. 8 with E_H given explicitly in Eq. 62.

Step 1: extract initial transfer. Taking the inner product of Eq. 49 with u and using the transfer score k of Eq. 6 gives the exact identity

$$\langle u, F_Q(0) - F_{Q_0}(0)\rangle = \frac{\beta}{2}\|u\|\kappa_Q. \tag{55}$$

Splitting k into positive and negative parts, with expectations over $x \sim p$ and $y \sim Q$,

$$\kappa_Q = \mathbb{E}[k]_+ - \mathbb{E}[-k]_+, \qquad a_Q = \mathbb{E}[k]_+ + \mathbb{E}[-k]_+, \qquad a_Q c_Q = \mathbb{E}[-k]_+, \tag{56}$$

so $\kappa_Q = a_Q - 2a_Q c_Q = a_Q(1 - 2c_Q)$.

Step 2: control vector trajectory drift. Let $\tau_t = \sum_{j<t} \eta_j$. Since $\|F_Q\| \leq M$, all iterates obey $\|v_t^Q\| \leq M\tau_t$. Writing $R_Q = v_H^Q - \tau_H F_Q(0)$, Lipschitz continuity gives

$$\|R_Q\| \leq \sum_{t=0}^{H-1} \eta_t \|F_Q(v_t^Q) - F_Q(0)\| \tag{57}$$

$$\leq LM \sum_{t=0}^{H-1} \eta_t \tau_t = \frac{LM}{2}\left(\tau_H^2 - \sum_{t=0}^{H-1} \eta_t^2\right). \tag{58}$$

The same bound holds for Q_0. It controls the full vector remainder, including directions orthogonal to the initial utility gradient.

Step 3: compare endpoint utilities. Taylor's inequality on the trajectory ball gives

$$|U(v) - U(0) - u^\top v| \leq \frac{L_U}{2}\|v\|^2. \tag{59}$$

Applying the lower inequality to v_H^Q and the upper inequality to $v_H^{Q_0}$ yields

$$G_H(Q) \geq u^\top (v_H^Q - v_H^{Q_0}) - \frac{L_U}{2}\left(\|v_H^Q\|^2 + \|v_H^{Q_0}\|^2\right) \tag{60}$$

$$\geq \tau_H u^\top (F_Q(0) - F_{Q_0}(0)) - \|u\|(\|R_Q\| + \|R_{Q_0}\|) - L_U M^2 \tau_H^2. \tag{61}$$

Substituting the bounds on $\|R_Q\|$ and $\|R_{Q_0}\|$ and collecting the correction terms proves Proposition 1 with

$$E_H = \|u\| LM \left(\tau_H^2 - \sum_{t=0}^{H-1} \eta_t^2\right) + L_U M^2 \tau_H^2. \tag{62}$$

The correction is a worst case over the compared constructions and does not vanish as $Q \to Q_0$. If a source has $\kappa_Q > 0$ and the schedule is scaled by $\lambda > 0$, then $K_H = \lambda K_*$ and $E_H = \lambda^2 E_*$. For sufficiently small λ, the bound is positive. Any target ϵ up to that positive bound makes Eq. 9 nonempty.The coordinates a_Q and c_Q are evaluated at the DPO initialization, and the bound is informative only while τ_H is small enough that E_H does not dominate. Within that range it orders constructions by κ_Q and fixes the direction of each intervention's effect, which is what P1–P3 test; it does not give the magnitude of the endpoint gain.

F.4 Intervention Identities and Implementation

The interventions instantiate the comparison distribution Q_0 of §3.2 as follows: randomized mixtures compare against the replaced Self source, the structural controls compare against length-matched gibberish, and reference-likelihood reselection compares against native selection ($\gamma = 0$).

Source composition. Since k is fixed by the common reference, utility, and comparison distribution, randomized source assignment gives

$$\kappa_{\mathcal{M}_{\mathbf{w}}} = \sum_s w_s \kappa_{Q_s}, \qquad a_{\mathcal{M}_{\mathbf{w}}} = \sum_s w_s a_{Q_s}, \qquad a_{\mathcal{M}_{\mathbf{w}}} c_{\mathcal{M}_{\mathbf{w}}} = \sum_s w_s a_{Q_s} c_{Q_s}. \tag{63}$$

The adverse share mixes with transfer-strength weights. Uniform M, L, L_U across the source family make the gain lower bound affine in source weights. The actual endpoint includes the source-dependent trajectory and curvature terms bounded in the proof.

Prompt reassignment. For an additive decomposition $\phi(x, y) = \phi_{\mathrm{dom}}(y) + \phi_{\mathrm{pair}}(x, y)$, independent reassignment preserves $\mathbb{E}_Q \phi_{\mathrm{dom}}(y)$ and its contribution to κ_Q. The prompt-dependent component can change. An exact finite-dataset derangement preserves the empirical reject-text multiset and removes all original pairings. The prompt-mismatch controls additionally apply interface normalization and execution filtering, producing a transformed set. An order-invariant lexical component of ϕ_{dom} is also preserved by lexical permutation of its token inventory. Whether this component supplies favorable transfer is the construction hypothesis examined by the structural controls.

Reference likelihood selection. The standardized candidate scores are

$$s_j = \frac{1}{|y_j|} \log S(y_j \mid x), \qquad \tilde{s}_j = \frac{s_j - \overline{s}_{C_x}}{\mathrm{sd}_{C_x}(s)}, \tag{64}$$

and reselection samples q_γ as in Eq. 12. Let $k_j = k(x, y_j)$ and hold Q_0, u, and each candidate bank fixed. Differentiating Eq. 12 gives

$$\frac{d\kappa_{Q_\gamma}}{d\gamma} = -\mathbb{E}_{x\sim p} \operatorname{Cov}_{j\sim q_\gamma(\cdot|C_x)}(\tilde{s}_j, k_j). \tag{65}$$

A negative averaged covariance increases the leading transfer term as repulsion strengthens. This is the explicit connection condition between the observed reference score and the vector model's utility-aligned geometry.

Positive γ favors lower-reference-likelihood candidates, negative γ favors higher-likelihood candidates, and $\gamma = 0$ gives native selection. Shared Gumbel variables couple sampling randomness across arms. The binary-wrong sweep retains failing candidates with their observed partial test-pass fractions. The 0.5B/1.5B/3B arms use this binary filter; the 7B/14B arms use the earlier candidate-bank filtering protocol.

F.5 Connection to Gradient Interference

At initialization, write $g^+ = g_0(x, y^+)$ and $g^- = g_0(x, y^-)$. For one pair,

$$\Delta v = \frac{\eta\beta}{2}(g^+ - g^-) = \frac{\eta\beta}{2}\delta. \tag{66}$$

The first-order change in the chosen log probability is $\frac{\eta\beta}{2}(\|g^+\|^2 - \langle g^+, g^-\rangle)$, as in gradient-interference analyses (Yuan et al., 2025; Razin et al., 2025). The utility gradient in Eq. 53 instead aggregates reward-relevant score directions over evaluation prompts. The transfer score projects the change in negative supervision onto this aggregate direction, connecting training constructions to evaluation utility.