COUNTERFACTUAL QUERY FILTERING FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation supervises student-generated responses with teacher predictions, but teacher–student disagreement alone does not reveal how strongly those predictions depend on the query. We propose Counterfactual Query Filtering (CQF) to allocate supervision using query sensitivity at fixed student prefixes. CQF compares frozen-teacher predictions under the original query and one fixed anchor per domain, retaining the most sensitive positions for original-query distillation without query-specific rewrites. Across eight mathematics and code benchmarks and four teacher–student configurations, retaining approximately 10% of positions improves the macro-average over full distillation by 0.76–2.58 percentage points in single-run comparisons, also outperforming count-, position-, and advantage-mass-matched random controls. At a shared 0.6B checkpoint, CQF updates align more closely with independent reference gradients and reduce reference loss more than count- and position-matched random selection at equal parameter displacement. Across methods, local diagnostic rankings differ from terminal accuracy rankings, distinguishing update utility from final performance.

1 INTRODUCTION

On-policy distillation (OPD) uses teacher predictions to train on student-generated responses, addressing the mismatch between offline demonstrations and the student's own prefixes (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025). The remaining question is where to apply supervision. Both the query and accumulated response condition the teacher, so teacher–student disagreement can persist even where the explicit query has little influence. We study whether query sensitivity at a fixed response prefix can identify teacher signals worth retaining.

Selective distillation uses student uncertainty, teacher–student disagreement, and local teachability to allocate supervision (Xu et al., 2026; Wang et al., 2026). Query contrasts provide a complementary criterion by testing how predictions change when the question changes. CREDIT contrasts teacher predictions across queries in a batch, and CROP rescores fixed student prefixes under original, paraphrased, and condition-changing prompts (Shen et al., 2026; Li et al., 2026). We ask whether a single anchor shared within a domain can provide this selection signal without a separate semantic rewrite for each training query.

Counterfactual Query Filtering (CQF; Figure 1) uses one fixed anchor for code and one for mathematics. The student generates once under the original query. A frozen teacher then scores exactly the same response prefixes under both the original query and the domain anchor. Holding the prefix fixed isolates the change in query context at each realized prediction point. Low cosine similarity marks positions whose predictive distributions respond most strongly to the query replacement. CQF retains these positions and applies the original-query OPD loss, separating the choice of supervision positions from the supervision target. This anchor-relative score can also reflect incompatibility between the anchor and the retained prefix.

At a 7K response cap and 100 training steps, retaining approximately 10% of positions improves the eight-benchmark macro-average over Full OPD and count-matched random selection across four teacher–student configurations. Count-, position-, and advantage-mass-matched controls test whether generic properties of sparse supervision reproduce these gains. We also compare adapted selective-distillation methods under their respective loss and feedback protocols.

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.

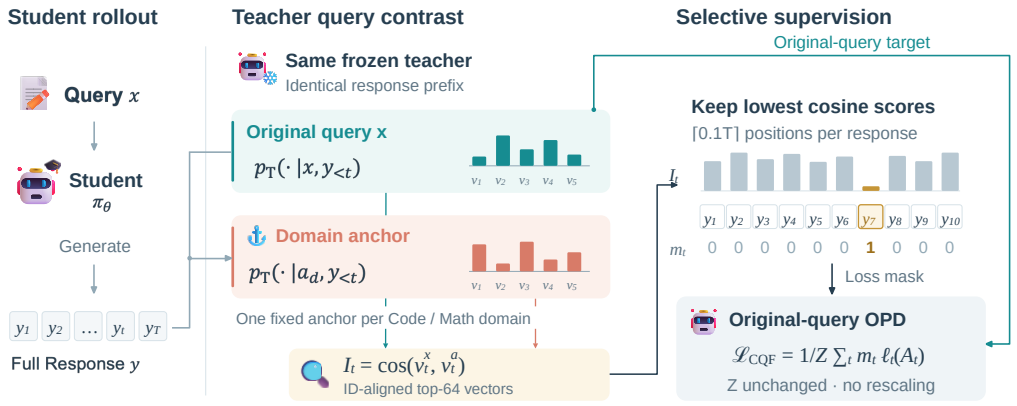

Figure 1: CQF selects supervision through a shared-anchor query contrast. The student generates once under x, and a frozen teacher scores each response prefix under x and a fixed domain anchor. Token-ID-aligned top-64 distributions define a mask retaining the 10% lowest-cosine positions. This mask gates the original-query OPD loss while preserving the pre-selection token denominator, with no rescaling. Response tokens, teacher distributions, and masks are schematic.

We also examine the updates produced by the selected supervision at a shared checkpoint. Compared with count- and position-matched random masks, CQF yields updates that align more closely with independent reference gradients and reduce reference loss more at equal parameter displacement. Comparing methods and changing the source of the advantage estimates reveal that lower gradient conflict, greater local loss reduction, and higher terminal accuracy need not coincide.

2 Related Work

On-policy and selective distillation. GKD addresses distribution mismatch in student-generated responses, while MiniLLM emphasizes reverse KL (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025). DistiLLM combines skew divergences with adaptive off-policy training, and DistiLLM-2 uses different objectives for teacher- and student-generated responses (Ko et al., 2024; 2025). SWITCH selectively introduces teacher intervention during generation (Koo et al., 2025). TIP and TA-OPD select token supervision using informativeness and teachability (Xu et al., 2026; Wang et al., 2026). IW-OPD addresses position bias by reweighting dense OPD advantages using cumulative absolute teacher–student log-probability gaps along the prefix (Xie et al., 2026). CQF preserves student trajectories and original-query targets, selecting loss positions through teacher predictions under different query contexts.

Counterfactual input dependence. PMI-FAITH measures document dependence to assess faithfulness, while PMI-DECODE and context-aware decoding use context contrasts to guide generation (Nandwani et al., 2023; Shi et al., 2024). In distillation, CREDIT uses batch-contrastive query replacement to assign input-specific credit in feedback-conditioned self-distillation (Shen et al., 2026). CROP uses meaning-preserving paraphrases to calibrate teacher sensitivity to condition-changing prompts (Li et al., 2026). CQF shares CROP's use of teacher contrasts at fixed student prefixes to mask original-query supervision. It replaces query-specific rewrites with a shared domain anchor and evaluates the resulting selector through matched controls and local update diagnostics.

Gradient interference. PCGrad, CAGrad, and Nash-MTL address task-gradient interactions through projection, conflict-averse optimization, and bargaining (Yu et al., 2020; Liu et al., 2021; Navon et al., 2022). Learning example weights from clean validation loss links training directions to an external objective (Ren et al., 2018). GUPO studies query-group gradients in GRPO through gradient uncertainty (Guo et al., 2026). CQF derives its training mask from teacher predictions and uses reference gradients only for evaluation. Our diagnostics separate interference among update

queries from alignment with independent tasks and from reference-loss changes after equally sized displacements in parameter space.

3 COUNTERFACTUAL QUERY FILTERING

3.1 SCORING A FIXED PREFIX UNDER TWO QUERY CONTEXTS

Let x denote a query, $y = (y_1, \ldots, y_T)$ a response sampled from student π_θ, and p_T a frozen teacher. Each query domain $d(x)$ has a fixed anchor question $a_{d(x)}$. The original and reference branches share the system message, chat template, thinking configuration, and student response. Only the final user question is replaced in the reference branch. At each valid response position, the two teacher predictions are given by

$$p_t^x(v) = p_T(v \mid x, y_{<t}), \qquad p_t^a(v) = p_T(v \mid a_{d(x)}, y_{<t}). \tag{1}$$

The student generates once under x, after which both teacher branches score the same prefixes. Replacing the explicit query leaves any query information already encoded in the response prefix intact. The contrast therefore measures anchor-relative sensitivity conditional on that realized prefix.

3.2 TOKEN-ALIGNED SPARSE DISTRIBUTION SIMILARITY

To compare distributions whose top tokens may differ, we align them by token ID. Let K_t^c contain the top-64 token IDs of p_t^c for $c \in \{x, a\}$. The sparse vectors and their cosine similarity are

$$v_t^c(j) = p_t^c(j)\mathbf{1}\{j \in K_t^c\}, \qquad I_t = \frac{\langle v_t^x, v_t^a \rangle}{\|v_t^x\|_2 \|v_t^a\|_2 + \varepsilon}, \qquad S_t = 1 - I_t. \tag{2}$$

We use $\varepsilon = 10^{-8}$ and compute the numerator over token IDs shared by the two supports. In these truncated distributions, I_t measures anchor-relative invariance and S_t measures sensitivity.

For a response r with $T_r > 0$ valid positions, CQF retains the $k_r = \lceil qT_r \rceil$ positions with the lowest I_t and masks the remaining distillation signals:

$$m_{r,t} = \mathbf{1}\{t \in \mathrm{BottomK}(I_{r,:}, k_r)\}, \qquad \widetilde{A}_{r,t} = m_{r,t} A_{r,t}. \tag{3}$$

Here $A_{r,t}$ is the original-query signal defined below. The principal configuration uses $q = 0.1$, with 0.3 and 0.5 evaluated as alternatives. Ties follow the top-k operator's implementation. "Low10" denotes retention of the lowest-cosine, and thus highest-sensitivity, positions; High10 retains the opposite end of the score distribution.

3.3 DISTILLATION OBJECTIVE AND NORMALIZATION

The distillation signal is the clipped teacher–student log-probability difference under the original query. At each loss pass, training recomputes the actor log probability and detaches the signal $A_t = \mathrm{sg}[\mathrm{clip}(\log p_T(y_t \mid x, y_{<t}) - \log \pi_\theta(y_t \mid x, y_{<t}), -10, 10)]$. Let Z count valid response tokens in the batch before filtering, and let π_{old} denote the sampling policy. With $\widetilde{A}_t = m_t A_t$ and $r_t(\theta) = \pi_\theta(y_t \mid x, y_{<t}) / \pi_{\mathrm{old}}(y_t \mid x, y_{<t})$, the PPO-style clipped objective (Schulman et al., 2017) takes the form

$$\mathcal{L}_m = -\frac{1}{Z} \sum_t \min\Big(r_t(\theta)\widetilde{A}_t, \mathrm{clip}(r_t(\theta), 1 - \epsilon_-, 1 + \epsilon_+)\widetilde{A}_t \Big), \tag{4}$$

with an additional dual-clipping guard for negative advantages. We use $\epsilon_- = 0.2$ and $\epsilon_+ = 0.28$, with implementation details in Appendix A. CQF constructs its advantage entirely from teacher and student log probabilities under the original query.

The no-rescale setting preserves selected advantages and the pre-selection count Z. Filtering changes gradient magnitude and direction; Adam state and gradient clipping further affect the optimizer step. Count- and normalization-matched random masks control for sparse supervision. Section 5 compares local directions at equal parameter displacement.

Anchor choice and computation. The principal configuration uses one Pass0 anchor per Code/Math domain from teacher-specific calibrated banks (Appendix A). Pass0 denotes zero verified successes in 16 teacher samples. CQF processes every response position and adds teacher scoring, so 10% retention describes supervision density rather than a 90% computational saving.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and data. We use Qwen3 students with 0.6B, 1.7B, and 4B parameters (Yang et al., 2025) and two teachers: Qwen3-30B-A3B-Thinking-2507 (T30) and Qwen3-4B-Thinking-2507 (T4). The main configuration is T30 $\rightarrow$ 0.6B. Our mixed dataset contains 25,276 Eurus-2 Code queries and 17,398 processed DAPO-Math queries, for a total of 42,674 (Cui et al., 2025; Yu et al., 2025). Training uses verl (Sheng et al., 2024), with 64 queries per batch, four student responses per query, a learning rate of 10^{-6}, and rollout temperature and top-p both set to 1. Students are initialized from their released Qwen3 checkpoints. The Qwen3-4B student is distinct from the Thinking-2507 teacher. We evaluate terminal checkpoints without validation-based selection. Main results use one training run per configuration, and Appendix F reports three CQF runs at 2K.

Budgets. We study response-cap/training-step pairs of 2048/350, 4096/175, and 7168/100, abbreviated 2K, 4K, and 7K. Their common cap–step product of 716,800 matches a nominal generation upper bound. Actual tokens, computation, sampled queries, and update counts vary. Each setting has its own Full OPD baseline; cross-setting comparisons assess the joint cap/step schedule.

Evaluation. We evaluate on AIME24, AIME25, AMC23, HMMT25, HumanEval, MBPP, LiveCodeBench (LCB), and MBPP+. The code suites follow established benchmarks (Chen et al., 2021; Austin et al., 2021; Liu et al., 2023; Jain et al., 2024). Scores average sampled correctness within queries and then across queries, rather than counting any successful attempt among n. Math uses 16 samples per query, HumanEval/MBPP/MBPP+ use eight, and LCB uses one. Math and Code average their four benchmarks; Mean equally weights both domains, equivalently all eight benchmarks. Evaluation uses a shared 38,912-token generation cap, independent of training (Appendix B). Scores and differences are computed before rounding to two decimals; improvements are percentage points.

4.2 MAIN RESULTS AND MATCHED CONTROLS

Table 1 compares Full OPD, CQF (Low10), and random controls under a common loss. Rand matches CQF's retained count without rescaling advantages. CQF achieves a higher eight-benchmark macro-average than Full in all four configurations and exceeds Rand by 1.69, 1.32, 0.81, and 2.05 points in table order. Rand also improves over Full in both T4 configurations, showing that sparsity and its associated optimization effects contribute to the observed gains.

Position-matched random selection (PMR) matches CQF's counts within normalized response-position bins. In table order, its Means are 39.68, 40.74, 59.66, and 74.20, trailing CQF by 1.64, 2.09, 1.33, and 2.24 points. Advantage-mass-matched random selection (AMR) randomly permutes positions and selects the prefix whose cumulative $|A_t|$ is closest to CQF's retained mass, obtaining 39.43 in the main setting. Neither separate control recovers CQF's Mean or matches its full optimizer update. AMR's count varies with the mass target instead of staying at 10%.

The aggregate gains include variation across benchmarks. In the main setting, CQF improves Math by 2.68 points and Code by 0.50, while LCB falls from 35.45 to 33.65. For T30 $\rightarrow$ 4B, CQF has the highest Mean (76.44), while PMR leads on LCB (72.32).

4.3 COMPARISON WITH ADAPTED SELECTION METHODS

Table 2 compares Full and CQF with adapted implementations of TIP, TA-OPD, CREDIT-random10 (CREDIT-A), and student-entropy selection, using each teacher to distill into 0.6B. All methods use mixed CodeMath data, 7K/100 steps, and the same evaluation protocol. The adapted methods normalize by selected-token count and use unclamped advantages. CREDIT-A additionally uses ground-truth feedback and an external frozen teacher. CQF preserves the pre-selection denominator

Table 1: Benchmark results (%) at 7K/100 steps, with one run per configuration. Mean equally weights the eight benchmarks. All methods use the same advantage clipping and pre-selection loss denominator. The text defines each matched control.

Teacher Student	Method	Math				Code				Overall
		AIME24	AIME25	AMC23	HMMT25	HumanEval	MBPP	LCB	MBPP+	**Mean**
T30 ↓ 0.6B	Full	16.04	21.46	56.41	10.21	66.23	53.02	**35.45**	59.03	39.73
	Rand	15.00	22.29	56.41	12.29	66.01	51.95	33.36	59.72	39.63
	PMR	12.50	**23.33**	56.25	12.29	66.23	52.58	33.36	60.91	39.68
	AMR	14.17	20.42	56.25	12.71	65.47	52.23	34.03	60.19	39.43
	CQF	**16.46**	**23.33**	**60.47**	**14.58**	**66.46**	**54.15**	33.65	**61.47**	**41.32**
T4 ↓ 0.6B	Full	15.42	22.50	61.09	11.04	62.58	54.67	33.74	61.01	40.26
	Rand	**17.29**	24.38	62.03	14.37	65.32	52.78	35.73	60.25	41.52
	PMR	15.62	21.46	62.19	12.92	**66.84**	52.50	34.31	60.09	40.74
	AMR	16.04	22.50	58.75	10.00	63.26	51.23	**36.11**	58.27	39.52
	CQF	16.46	**24.58**	**65.31**	**17.50**	66.08	**54.83**	35.45	**62.47**	**42.83**
T4 ↓ 1.7B	Full	44.79	37.08	83.28	21.88	86.20	72.28	51.85	73.05	58.80
	Rand	**47.92**	39.17	85.16	23.33	86.36	71.85	54.60	73.02	60.17
	PMR	45.21	38.33	83.44	21.04	85.98	73.45	**55.45**	**74.37**	59.66
	AMR	43.75	39.17	**85.47**	20.21	86.51	72.47	55.07	73.18	59.48
	CQF	47.50	**40.83**	84.53	**24.79**	**87.04**	**74.28**	54.69	74.21	**60.98**
T30 ↓ 4B	Full	67.50	**66.04**	96.41	43.12	93.75	88.85	71.00	78.80	75.68
	Rand	65.62	62.08	97.34	37.08	94.05	**88.95**	70.90	79.10	74.39
	PMR	63.75	61.67	95.47	38.96	94.13	88.48	**72.32**	78.80	74.20
	AMR	65.83	62.50	**98.28**	42.08	94.21	88.20	70.52	78.84	75.06
	CQF	**71.67**	64.79	97.50	**43.54**	**94.44**	88.78	71.09	**79.73**	**76.44**

and clamps advantages. CQF, TIP, and Entropy select positions within each response, whereas TA-OPD and CREDIT-A select across the batch. These comparisons capture the combined effects of selection, loss, and supervision (Appendix E).

Entropy selection. Entropy ranks valid response positions by the student's full-vocabulary predictive entropy, $H_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid x, y_{<t}) \log \pi_\theta(v \mid x, y_{<t})$, retaining the $\max(1, \lfloor 0.1T \rfloor)$ highest-entropy positions in each response of length T. This criterion prioritizes student uncertainty under the original query, whereas CQF measures teacher-side query sensitivity. Selected positions receive the original-query distillation signal with the baseline loss settings described above.

CQF achieves the highest observed Mean under both teachers: 42.83 with T4, compared with 41.75 for Entropy, and 41.32 with T30, compared with 40.71 for CREDIT-A. The margins over these strongest evaluated alternatives are 1.09 and 0.61 points, computed before rounding. These point estimates compare complete method configurations. The common-loss comparisons in Table 1 provide a complementary view of mask selection.

4.4 Response budgets, retention, and anchors

Figure 2 shows that Low10 improves Mean over Full across all three response-cap/training-step schedules. At 2K/350, Full obtains 38.88 and Low10 obtains 42.20 (+3.32). The corresponding scores are 38.09 and 41.06 (+2.97) at 4K/175, and 39.73 and 41.32 (+1.59) at 7K/100. Low10 also achieves the highest Mean among the Pass0 Low10/30/50 configurations at each schedule.

Reversing the score ranking reduces performance in the main setting. High10/30/50 obtain Means of 34.64/36.71/37.86, compared with 41.32/40.90/39.91 for Low10/30/50. The results favor retaining low-cosine positions in these settings, although mask choice also changes the retained advantage mass and the magnitude of the resulting update.

Performance does not vary monotonically across the tested anchor calibration bins. At 7K, Low10 with Pass0/75/100 obtains 41.32/41.51/41.54, while at 2K it obtains 42.20/41.22/41.34. Each bin

Table 2: Adapted selection methods for the 0.6B student at 7K/100 steps. Each row reports one training run, and Mean averages eight benchmarks (%). CREDIT-A denotes the fixed-teacher CREDIT-random10 adaptation. Loss normalization, selection budgets, and feedback follow the protocols described in the text.

Teacher Student	Method	Math				Code				Overall
		AIME24	AIME25	AMC23	HMMT25	HumanEval	MBPP	LCB	MBPP+	**Mean**
T30 ↓ 0.6B	Full	16.04	21.46	56.41	10.21	66.23	53.02	**35.45**	59.03	39.73
	TIP	15.21	22.29	59.69	11.25	59.30	50.90	34.50	59.03	39.02
	TA-OPD	**16.46**	23.12	59.53	12.92	66.08	52.73	35.26	59.03	40.64
	CREDIT-A	14.58	**23.33**	57.34	13.33	**70.12**	53.40	33.93	59.66	40.71
	Entropy	15.62	21.25	60.16	**14.58**	67.15	53.00	34.03	59.76	40.69
	CQF	**16.46**	**23.33**	**60.47**	**14.58**	66.46	**54.15**	33.65	**61.47**	**41.32**
T4 ↓ 0.6B	Full	15.42	22.50	61.09	11.04	62.58	54.67	33.74	61.01	40.26
	TIP	**17.92**	22.50	61.09	11.04	64.10	54.30	33.93	59.46	40.54
	TA-OPD	15.83	23.96	63.44	11.67	61.36	50.95	35.36	56.12	39.83
	CREDIT-A	17.50	23.33	62.34	11.88	**66.16**	53.73	35.36	60.09	41.30
	Entropy	17.50	23.75	65.16	14.58	64.79	53.33	35.07	59.82	41.75
	CQF	16.46	**24.58**	**65.31**	**17.50**	66.08	**54.83**	**35.45**	**62.47**	**42.83**

contains one anchor per domain. The sweep therefore compares specific anchor pairs, with anchor identity and calibrated difficulty varying together (Appendix C).

Measured training cost. The principal 7K CQF and Full runs use 106.29 and 94.44 allocated GPU-hours, respectively, corresponding to a 12.5% increase under the measured execution conditions. Actor-update times are similar at 42.48 and 41.88 seconds per step because both methods process every response token. The runs use different nodes and exclude offline anchor calibration, so these measurements do not isolate teacher-scoring overhead (Appendix F).

5 Local Update Diagnostics

5.1 A frozen local update

At checkpoint θ_0, we cache student log probabilities and freeze the sampled-token advantage:

$$A_t = \mathrm{sg}[\mathrm{clip}(\log p_T(y_t \mid x, y_{<t}) - \log \pi_{\theta_0}(y_t \mid x, y_{<t}), -10, 10)]\,. \tag{5}$$

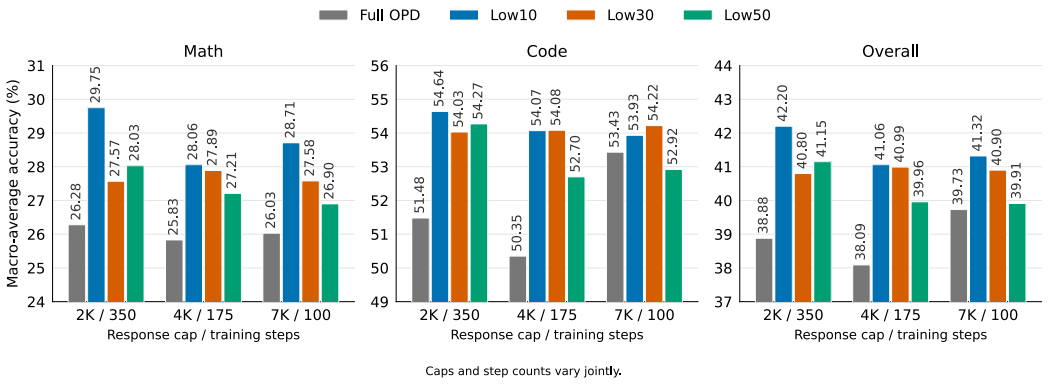

Figure 2: Terminal scores for T30 → 0.6B under three response-cap/training-step schedules, with one run per configuration. Low10/30/50 use Pass0 anchors without rescaling. Y-axes are truncated, and response caps and update counts change together.

For a batch containing Z valid response tokens before filtering, let $m_t \in \{0, 1\}$ be a frozen selection mask. In a shared parameter space $\mathcal{P}$, the token and batch update directions are

$$u_t = \frac{A_t}{Z} \nabla_{\mathcal{P}} \log \pi_\theta(y_t \mid x, y_{<t})\big|_{\theta_0}, \qquad U_m = \sum_t m_t u_t. \tag{6}$$

U_m is the negative gradient of the surrogate with frozen advantages. We compute it in FP32 on fixed BF16-generated trajectories, with the policy ratio set to one and optimizer state omitted (Appendix A). Matched controls use CQF's denominator; other methods use the losses in Appendix E. Dropped tokens remain in the prefix, so losses at later selected positions can still backpropagate through their representations.

5.2 Interference is distinct from task utility

We sum the response directions for query i into $U_i^{(m)}$. Following the negative-inner-product criterion for gradient conflict (Yu et al., 2020), we define the normalized negative interaction share:

$$C^-_{\text{mag}}(m) = \frac{\sum_{i<j} \max\{0, -\langle U_i^{(m)}, U_j^{(m)} \rangle\}}{\sum_{i<j} |\langle U_i^{(m)}, U_j^{(m)} \rangle|}. \tag{7}$$

Self-pairs are excluded. We treat comparisons with near-zero directions or denominators as undefined and report coverage. This statistic measures interactions between query groups and is invariant to common positive rescaling, although token selection can change the relative weights of queries.

Agreement between query directions need not imply progress on the task. Following validation-gradient alignment (Ren et al., 2018), let $d_{\text{task}} = -\nabla_{\mathcal{P}} L_{\text{ref}}(\theta_0)$, where L_{ref} is query-balanced NLL on independent, screened reference solutions. With local smoothness, nonzero update and reference directions, and a sufficiently small common radius ρ,

$$L_{\text{ref}}(\theta_0) - L_{\text{ref}}\left(\theta_0 + \rho \frac{U_m}{\|U_m\|}\right) = \rho \|d_{\text{task}}\| \cos(U_m, d_{\text{task}}) + O(\rho^2). \tag{8}$$

The first-order term links reference-loss improvement to directional alignment (Pruthi et al., 2020). A shared displacement radius separates this directional utility from update magnitude.

Equation 8 applies separately to each reference loss. E2 measures alignment on the direction set; E3 measures the loss change after displacement on a disjoint query set. E3 therefore tests whether directional utility transfers to independent reference samples.

5.3 Matched-mask comparisons at a shared checkpoint

We evaluate Equation 6 at the Full-OPD T30 $\rightarrow$ 0.6B Step100 checkpoint on four disjoint batches, each containing 16 queries (eight per domain, four responses per query). The 256 trajectories contain 1,730,682 tokens; 229 reach the 7168-token truncation limit, so the results primarily concern long, incomplete responses. Gradients cover the last transformer block and final normalization (15,731,968 FP32 parameters). Random and Position results average three masks within each batch. Variation across these batches measures sensitivity to update queries, not training-seed variability.

The dataset's 1,024-query validation split contains only Code and substantially overlaps training, so we construct independent reference sets for this diagnostic. Each domain has disjoint sets of 32 E2, eight calibration, and 64 E3 queries, giving 208 screened reference solutions across the two domains. All are disjoint from the 64 update queries and the checkpoint's OPD training data. E2 queries define reference gradients, calibration queries determine the shared displacement radius using the Full direction, and E3 queries measure loss changes on held-out reference samples. NLL is averaged over tokens within each solution and then across queries. Code solutions pass the supplied tests; Math solutions pass answer checks and AI-assisted screening (Appendix D). These probes use the frozen local update defined above, with policy ratio one and no optimizer state.

E1: query-group interactions. Low10 has lower C^-_{mag} than Random10-NR and Position10-NR in every batch (Table 3). It also falls below Full except in batch 3 (0.1554 versus 0.1495). The aggregate directions of retained and discarded tokens remain positively aligned (Appendix D), distinguishing query-group conflict from opposition between these two aggregates.

Table 3: Four-batch final-block means. E1: negative interaction share; E2: reference-gradient cosine; E3: reference NLL improvement at common $\rho = 0.01$. Lower E1 means less conflict; higher E2/E3 means greater task utility. All three TopAbs10-NR metrics are computed on the same frozen masks and the same four update batches.

	E1	E2 alignment		E3 ΔNLL	
Method	C^{-}_{mag}	Code	Math	Code	Math
Full	0.3042	0.2428	0.3135	0.0095	0.0087
Low10	0.1434	0.3139	0.3942	0.0114	0.0114
Random10-NR	0.3947	0.1499	0.1962	0.0058	0.0056
Position10-NR	0.4301	0.0991	0.1381	0.0039	0.0037
TopAbs10-NR	0.3643	-0.0633	-0.1186	-0.0032	-0.0036

E2: alignment with an independent task objective. Low10 aligns more closely with the reference direction than either matched sparse control in every batch and domain. It also exceeds Full in four Code and three Math batches. The independent reference solutions give this comparison an objective beyond agreement among update queries.

E3: utility under equal parameter displacement. All methods and batches share the independently calibrated radius $\rho = 0.01$ and its half, 0.005; parameters are restored before each displacement. At ρ, two-domain mean NLL gains are Low10 0.011429, Full 0.009114, Random 0.005660, and Position 0.003812. Low10 exceeds Random and Position in every batch and domain at both radii. Relative to Full, its gain is slightly smaller on Code batch 0 at ρ (a margin of -0.000019), but larger on Math and on the two-domain mean in every batch.

Large advantage mass does not imply useful supervision. TopAbs10-NR retains 64.04% of absolute advantage mass, compared with Low10's 23.65%, at the same token budget and denominator. Yet its E1 is higher than Low10's in every batch, its mean alignment is negative in both domains, and its mean two-domain NLL gain is -0.003381. Low10 has higher E2 in every batch and domain, and higher E3 at both radii. Retained advantage mass therefore does not explain the local utility of selection (Appendix D).

Full-parameter analysis. Extending the analysis to all 596,049,920 parameters preserves Low10's advantage over Full and Random in both domains for E2 and for E3 at both radii. Low10's Code/Math alignment is 0.1521/0.2553, compared with 0.1077/0.1355 for Full. At the independently calibrated full-space radius $\rho = 0.01$, two-domain mean NLL gains are Low10 0.047924, Full 0.031012, and Random 0.016618. Because full-space and final-block displacements have different geometries, we compare methods within each space. This comparison uses one preselected batch and one Random mask (Appendix D).

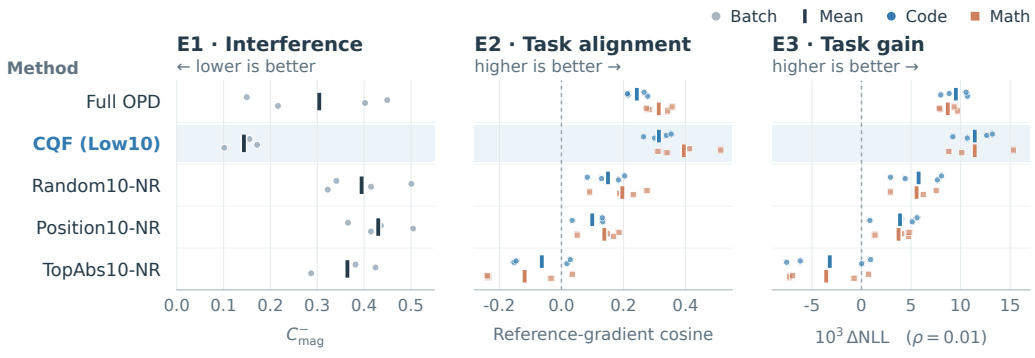

Figure 3: Local diagnostics on four shared update batches. Small markers show individual batches; short bold vertical marks show means. E3 uses the common radius $\rho = 0.01$ and displays $10^3\Delta$NLL. All panels use FP32 final-block parameters and the original frozen advantage.

5.4 SENSITIVITY AND COMPARISONS OF COMPLETE METHODS

Student log probabilities from BF16 generation and FP32 recomputation differ by 0.094919 nats on average in absolute value, changing the advantage sign state at 9.6099% of positions. To test sensitivity to this discrepancy, we replace the student term with the recorded serving policy π_{gen}:

$$A_t^{\text{gen}} = \text{sg}[\text{clip}(\log p_T(y_t \mid x, y_{<t}) - \log \pi_{\text{gen}}(y_t \mid x, y_{<t}), -10, 10)], \tag{9}$$

Trajectories, teacher scores, FP32 Jacobians, reference sets, and the main selectors' masks remain fixed; only Top-$|A|$10 is reranked. The same radii satisfy the independent calibration criteria.

Low10 retains higher E2/E3 than Random and Position in every batch and domain, while its E1 exceeds Full in all four batches: the mean Low-minus-Full difference changes from -0.1608 to $+0.0531$ (Appendix D). Direction cosines across advantage definitions range from -0.2760 to -0.0765 for Full and from 0.0709 to 0.2162 for Low10. Thus, the utility advantage over matched sparse controls persists despite substantial direction changes and a reversal of the conflict ordering with Full. This intervention varies the advantage within an FP32 local surrogate and does not reproduce BF16 training updates.

Table 4: Comparison of complete methods at the shared T30 → 0.6B Full Step100 checkpoint: four-batch final-block means, with E3 at $\rho = 0.01$. Lower E1 and higher E2/E3 are preferable. Terminal Mean averages eight benchmarks from separate 7K/100 training runs. CREDIT-A denotes the fixed-teacher CREDIT-random10 adaptation.

	E1	E2 alignment		E3 ΔNLL		Terminal
Method	C_{mag}^{-}	Code	Math	Code	Math	Mean (%)
Full	0.3042	0.2428	0.3135	+0.009522	+0.008706	39.73
TIP	0.3210	0.0549	0.0182	+0.000852	+0.000565	39.02
TA-OPD	0.0217	0.3879	0.4772	+0.013893	+0.014035	40.64
CREDIT-A	0.1610	0.3363	0.4369	+0.013032	+0.012249	40.71
Entropy	0.3698	-0.0681	-0.1414	-0.003596	-0.004446	40.69
CQF	0.1434	0.3139	0.3942	+0.011429	+0.011429	41.32

Local utility and terminal accuracy yield different rankings. Table 4 compares the methods' complete losses. Full and CQF clip advantages, the four adapted baselines do not, and CREDIT-A adds ground-truth feedback. Positive scalar denominators cancel in these diagnostics (Appendix E), but the remaining loss differences prevent attribution to masks alone. TA-OPD has the best mean E1/E2/E3 and a lower terminal Mean than CQF. Entropy worsens NLL in every batch, domain, and radius, yet exceeds Full in terminal Mean. These differing rankings show that local directional utility at the T30 → 0.6B checkpoint does not by itself explain the final accuracy gains.

6 DISCUSSION AND CONCLUSION

CQF allocates original-query OPD supervision through a shared domain anchor. Its Mean exceeds Full and count-matched random selection across four model pairs, with position- and advantage-mass-matched controls further supporting the selector. Additional teacher scoring changes supervision allocation while retaining full-trajectory processing.

The evaluation covers Qwen3 models on Math and Code, with one training run per main configuration and no direct CROP comparison. Comparisons with adapted methods combine selection scores with their associated loss choices. Evaluating one anchor per domain/bin leaves variation within each calibration bin unresolved, and the fidelity of top-64 scores to full-vocabulary rankings remains an open question for this selector.

At the shared checkpoint, CQF updates improve the independent reference objective more than matched sparse controls. Changing the advantage estimates reverses the mean conflict ranking relative to Full, while CQF retains its utility advantage over matched sparse controls. Terminal evaluation and these local probes therefore answer different questions: whether a training procedure improves benchmark performance, and whether a frozen update direction makes progress on an independent reference objective. How these local differences translate into training gains through successive optimizer updates remains unresolved.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

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. URL https://arxiv.org/abs/2107.03374.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. URL https://arxiv.org/abs/2502.01456.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Peizheng Guo, Jianqi Zhang, Xingyu Zhang, Yun Fan, Jiahuan Zhou, Changwen Zheng, and Wenwen Qiang. GUPO: Gradient uncertainty-aware policy optimization for post-training large language models. *arXiv preprint arXiv:2608.17411*, 2026. URL https://arxiv.org/abs/2608.17411.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2105.09938.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Jahyun Koo, Yerin Hwang, Yongil Kim, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. SWITCH: Studying with teacher for knowledge distillation of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3733–3746. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.206. URL https://aclanthology.org/2025.findings-naacl.206/.

Enhan Li, Junhao He, and Hongyang Du. CROP: Task relevance via counterfactuals for selective on-policy distillation. *arXiv preprint arXiv:2608.13387v3*, 2026. URL https://arxiv.org/abs/2608.13387v3. Version 3, revised September 16, 2026.

Yujia Li, David Choi, Junyoung Chung, et al. Competition-level code generation with AlphaCode. *arXiv preprint arXiv:2203.07814*, 2022. URL https://arxiv.org/abs/2203.07814.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*, 2023. URL https://arxiv.org/abs/2305.01210.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, October 2025. doi: 10.64434/tml.20251026. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi, and Luis Lastras. Pointwise mutual information based metric and decoding strategy for faithful generation in document grounded dialogs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10335–10347. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.639. URL https://aclanthology.org/2023.emnlp-main.639/.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 2022. URL https://proceedings.mlr.press/v162/navon22a.html.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343. PMLR, 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation. *arXiv preprint arXiv:2605.11613*, 2026. URL https://arxiv.org/abs/2605.11613.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024. URL https://arxiv.org/abs/2409.19256.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-short.69. URL https://aclanthology.org/2024.naacl-short.69/.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026. URL https://arxiv.org/abs/2605.26844.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026. doi: 10.48550/arXiv.2606.22600. URL https://arxiv.org/abs/2606.22600.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026. URL https://arxiv.org/abs/2604.14084.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

AI Use Statement

AI assisted code development and verification, diagnostic design, experiment execution, result checking and interpretation, and manuscript drafting, polishing, translation, and layout. Mathematical reference solutions received AI-assisted screening without expert human review (Appendix D). The authors are responsible for the results, analysis, and manuscript.

A Implementation and Reproducibility

Diagnostic precision and sparse scoring. We evaluate the student policy and Jacobian in FP32 on fixed BF16-generated trajectories. F32 checkpoint parameters are quantized for serving, so sampling and diagnostic probabilities can differ. Fixing the policy ratio at one isolates the local surrogate from multiple optimizer steps, clipping gates away from ratio one, and Adam history. Zero-padding both top-64 distributions over their token-ID union (at most 128 entries) produces the vocabulary-coordinate representation in Equation 2.

Separation of training and diagnostics. Training uses the `k1` sampled log-probability loss in policy-gradient mode with a standard clipped policy loss. Each loss pass recomputes actor log probabilities to form the detached advantage. The diagnostic instead caches these values at the chosen checkpoint to define the frozen-A surrogate. The policy ratio compares the current actor with the old sampling policy. Dual clipping for negative advantages uses the effective distillation-path value of 3.0, independently of the actor-level configuration value of 10. At ratio one, this guard leaves Equation 6 unchanged.

Gradient and optimizer conventions. CQF and its matched controls preserve the pre-filter valid-token denominator when aggregating the token-mean loss. In distributed execution, framework scaling implements this global normalization across partitions. The selector zeros local loss contributions while retaining all input tokens. Training applies optimizer state and gradient clipping, whereas the equal-displacement probe directly changes parameters in the specified space. A common scalar factor leaves cosine and Equation 7 unchanged, but relative query weights and optimizer dynamics can affect the comparison. The selected direction depends jointly on the teacher-based mask and the student's parameter-space Jacobian, motivating the empirical diagnostics. Without a lower-learning-rate Full control, optimization-scale effects remain unresolved.

Anchors and calibration. Queries use Code or Math anchors according to their data source. Pass0, Pass25, Pass50, Pass75, and Pass100 denote 0, 4, 8, 12, and 16 verified successes among 16 teacher samples, with one active question per domain/bin. T4 uses a bank recalibrated for that teacher. Calibration uses temperature 0.6, top-p 0.95, top-k 20, and a 63,488-token generation cap. The presence penalty is 1.5 for Code and zero for Math. These settings differ from student training rollouts. Each anchor comparison preserves the surrounding template and supplied response prefix. Because the reference prompt may be incompatible with the forced prefix, the anchor-bin comparisons measure the selector's sensitivity to this choice.

Controls. Random-NR uniformly selects the same ceiling-based number of positions in each response without rescaling. Position-NR matches Low's retained count within normalized response-position bins. AMR randomly orders valid positions and selects the prefix whose sum of absolute advantages is closest to Low's retained mass. Mass matching determines its retained count, and the "10" suffix identifies the Low10 reference. The archived evaluation also includes Random-Length, which uses absolute-advantage rescaling. The matched controls reported here follow the protocols defined above.

Computational accounting. Separate scoring requests obtain original-query sampled-token teacher log probabilities and original/anchor top-k distributions. Overhead depends on serving, batching, and cache reuse. Appendix F reports generated tokens, retained positions, actor time, and allocated GPU-hours for full runs; separate teacher-branch timings are unavailable.

B Evaluation Protocol and Data Provenance

The full evaluation covers 82 configurations and checkpoints on eight benchmarks, including four alternative selectors for each of T30 and T4 distilling into the 0.6B student. All tables and differences use unrounded benchmark scores, with equal weight for each benchmark. Archived project manifests record model checkpoints and evaluation protocols, with file hashes linking configurations to their results.

Code training queries come from the Code portion of Eurus-2-RL-Data released with PRIME (Cui et al., 2025). Math queries come from the open-r1 processed DAPO-Math-17k dataset, which deduplicates and reformats DAPO problems before local conversion to the verl format (Yu et al., 2025). The dataset identifiers are `PRIME-RL/Eurus-2-RL-Data` and `open-r1/DAPO-Math-17k-Processed`. The archived project data manifest records source paths, row counts, the deterministic mixing permutation, and shard hashes.

The evaluator's `average_at_n_pct` statistic averages correctness across samples within each problem and then across problems. It estimates sampled success, whereas pass@n counts a problem as solved if any attempt succeeds. Evaluations share random seeds but retain model-dependent sampling noise. Seed0 and Seed42 directory labels identify runs, not paired training replicas. Archived base-model entries differ in template or path and include a Step100 identifier unrelated to OPD training. We restrict training comparisons to checkpoints evaluated under the common protocol. These comparisons do not establish gains over an initial student evaluated with a matched protocol.

Data splitting for the diagnostic. We shuffle the 42,674 training queries with seed 20260817. The dataset's 1,024-query validation split contains only Code and substantially overlaps the training set. We therefore construct independent reference sets for the diagnostic. Each domain contributes 32 direction queries, 64 evaluation queries, and eight calibration queries, yielding 208 screened

Table 5: Evaluation settings shared across models. Training rollout settings are specified in the main text.

Setting	Math	Code
Samples per query	16	8 (LCB: 1)
Temperature / top-p / top-k	0.6 / 0.95 / 20	0.6 / 0.95 / 20
Presence penalty	0	1.5
Generation cap / context cap	38,912 / 40,960	38,912 / 40,960
Evaluation shard seeds	42–49	42–49
Thinking	enabled	enabled
LCB version	—	release_v6

reference solutions. These sets are disjoint from the 64 update queries and the checkpoint's OPD training data. Appendix D describes their construction, screening, and sources.

C Anchor Calibration Sweep

For T30 $\rightarrow$ 0.6B, the full anchor-bin sweep below compares two training schedules. The best-performing anchor pair changes with the schedule.

Table 6: Low10 anchor-bin sweep for the main teacher–student configuration. Each bin uses one anchor pair, with one question per domain. Anchor identity and calibrated difficulty vary together.

Setting	Pass0	Pass25	Pass50	Pass75	Pass100
2K/350 Mean	42.20	41.74	41.32	41.22	41.34
7K/100 Mean	41.32	40.61	40.59	41.51	41.54

D Gradient Protocol and Full-Parameter Analysis

Independent reference construction. We check diagnostic update and reference queries against the checkpoint's 42,674 OPD training queries, anchor bank, and main evaluation questions. We exclude canonical exact matches and pairs with at least 60% coverage by shared 64-character substrings. Update, direction, calibration, and evaluation queries are mutually disjoint. This check covers the study corpora, excluding base-model pretraining data.

The 104 Code references comprise 101 CodeContests questions and three APPS questions (Li et al., 2022; Hendrycks et al., 2021), paired with programs supplied by the datasets. Retained programs pass every supplied test under Python hash seeds 0 and 1, including CodeContests public, private, and generated tests. The NLL target is the tested program with its recorded delimiters. Selection depends on the harness's input/output and execution-time restrictions as well as test performance. Passing this finite test suite establishes correctness on the tested inputs.

The 104 Math references are complete, unchanged solutions from the local OpenThoughts-derived Math corpus. Screening uses source correctness labels, final-answer matching, and documented AI assistance, without expert-human review or formal proof verification. Incorrect or uncertain solutions are replaced in fixed candidate order within each preassigned split, independently of gradients. Written solutions are NLL targets. Manifests record source revisions or local file/row identifiers, exclusions, program/solution hashes, and acceptance decisions.

Calibration and numerical validation. In the controlled mask comparison, CQF, Random-NR, and Position-NR share the frozen A, the per-response token count rounded up, and the original batch-token denominator. Position matching uses ten normalized response-position bins. Calibration uses only the Full direction of batch 0 and a separate set of eight queries per domain. We consider radii $\{0.01, 0.03, 0.1, 0.3, 1.0\}$ in ascending order. Both the candidate radius and its half must produce a resolvable loss change and agree with the first-order prediction within 25%. The signal threshold is the larger of 10^{-6} and ten times the repeated no-update NLL discrepancy. The

first radius, 0.01, satisfies these criteria and is shared across methods. The evaluation set is used only to measure task gains.

The controlled mask comparison contains 72 E3 perturbations (nine method/seed instances, two radii, four batches); the full-parameter analysis contains six. We restore the initial parameters before each displacement. The maximum relative error in displacement norm is below 9×10^{-7} in the main space and 5.6×10^{-6} in the full space; repeated no-update NLL differences are zero for every query. On a small model, we validate query aggregation and compare gradients with direct autograd and frozen-signal finite differences. For each diagnostic batch, we check that the retained and discarded directions add to the full direction on its first response, verify parameter restoration, and measure the actual displacement. All diagnostics use FP32 parameters and full-vocabulary softmax with TF32 disabled.

Retained and discarded aggregate directions have positive cosines of 0.3109–0.5438 across four batches. This alignment coexists with Low10's lower negative interaction share.

Table 7: Full-parameter E2/E3 on the preselected first update batch. All 596,049,920 parameters participate; $\rho = 0.01$ is calibrated independently of the main subspace. Random uses seed 17.

	E2 alignment		E3 ΔNLL	
Method	Code	Math	Code	Math
Full	0.1077	0.1355	0.0319	0.0302
Low10	0.1521	0.2553	0.0439	0.0520
Random10-NR (seed 17)	0.0560	0.0772	0.0164	0.0168

At the independently calibrated full-space radius $\rho = 0.01$, two-domain mean NLL gains are Low10 0.047924, Full 0.031012, and Random 0.016618. The ordering persists at half the radius: Code/Math gains are 0.023092/0.028038 for Low10, 0.016348/0.015495 for Full, and 0.008440/0.008702 for Random seed 17. Because full-space and final-block displacements have different geometries, we compare methods within each space.

Absolute-advantage selection. We test advantage-magnitude selection through TopAbs10-NR query-group interactions, using cached E2/E3 masks, clipped FP32 advantages, all 256 responses, and the original batch-token denominator. Reusing these masks preserves the selections at tied top-k boundaries. Each of the 16 query directions sums its four response directions. Stored hashes verify the parameters, teacher scores, response/query order, and result identities. Recomputed student log probabilities for sampled tokens match the cache within absolute/relative tolerances $3 \times 10^{-5}/10^{-6}$.

The sum of query directions must match the aggregate TopAbs batch direction within relative L_2 error 5×10^{-5} and cosine at least 0.999999. Across all four batches, the measured relative error is zero and the cosine is within 10^{-12} of one. We compute E1 from the resulting 16×16 Gram matrices; all 120 unordered off-diagonal pairs are defined, with no near-zero query vectors. Table 8 reports individual batches and the mean of their four conflict ratios, rather than a ratio from pooled Gram matrices. TopAbs E1 is evaluated in the final-block subspace with FP32 advantages; full-parameter and generation-time-advantage E1 are not evaluated.

Table 8: TopAbs10-NR and Low10 E1 on shared update batches. The final columns report TopAbs's retained absolute-advantage percentage and defined-pair coverage.

Batch	TopAbs E1	Low10 E1	Low$-$TopAbs	$\lvert A\rvert$ retained (%)	Pairs (%)
0	0.2869	0.1015	-0.1854	64.95	100
1	0.3644	0.1715	-0.1929	62.98	100
2	0.4243	0.1454	-0.2790	63.56	100
3	0.3817	0.1554	-0.2263	64.66	100
Mean	0.3643	0.1434	-0.2209	64.04	100

At matched token counts and with the same frozen advantage and denominator, TopAbs retains 64.04% of absolute advantage mass, compared with Low10's 23.65%. Its mean E1 is 0.3643, and

the Low10-minus-TopAbs difference is -0.2209. Low10 exceeds TopAbs in E2 for every batch and domain, and in E3 for every batch, domain, and radius. TopAbs nevertheless improves NLL in batch 3 in both domains at both radii, despite negative mean E2 and E3. Selecting for advantage magnitude therefore fails to recover CQF's local utility at this checkpoint.

Advantage definition and direction sensitivity. The main advantage subtracts the student log probability recomputed in FP32; the alternative uses the BF16 serving log probability recorded for the sampled token during generation. Across 1,730,682 tokens, 166,316 sign states change (9.6099%); 145,950 are strict positive/negative flips (8.4331%), with the rest involving exact zero. We verify token indexing with a prefix-only next-token check and deliberately shifted-logit controls. The probability sources differ in checkpoint quantization, backend, prefill, and decoding behavior, so their discrepancy reflects these factors jointly.

Table 9: Sensitivity to the student log probability used in A. Entries are four-batch means of Low10 minus each control; random seeds are averaged within batch. E1 favors negative differences; E3 favors positive differences. E3 averages Code/Math at $\rho = 0.01$. Against Full, conflict ranking reverses; task-gain ranking persists.

	E1: $\Delta C^{-}_{\mathrm{mag}}$		E3: ΔNLL margin	
Control	A^{FP32}	A^{gen}	A^{FP32}	A^{gen}
Full	-0.160779	+0.053149	+0.002315	+0.005543
Random10-NR	-0.251252	-0.073449	+0.005769	+0.002776
Position10-NR	-0.286681	-0.057356	+0.007617	+0.003570

Table 10 shows substantial direction changes: Full has negative cosine between advantage definitions in all four batches. The original radii satisfy calibration for the alternative Full direction in every batch. TopAbs E2/E3 use masks reranked under the alternative advantage; its E1 is excluded. Fixed FP32 Jacobians and reference objectives isolate advantage sensitivity within the local surrogate, where E2/E3 gains over matched sparse controls persist. This does not reconstruct actor training or an Adam step.

Table 10: Direction changes when replacing A^{FP32} by A^{gen}. Ranges span four batches, including all three seeds for random/position controls. Norm ratio is alternative/original before E3 normalization. TopAbs reranks its mask; other masks are identical.

	Cosine between directions		Norm ratio	
Method	Min	Max	Min	Max
Full	-0.2760	-0.0765	1.3991	2.2155
Low10	0.0709	0.2162	1.2708	1.5062
Random10-NR	0.1630	0.5797	0.9451	1.7272
Position10-NR	0.2871	0.5520	1.0315	1.7154
TopAbs10-NR	0.2652	0.7236	2.4840	2.7160

E Cross-Method Diagnostics and Terminal Accuracy

Shared frozen state and method-specific signals. We evaluate TIP, Entropy, TA-OPD, the CREDIT-random10 adaptation (CREDIT-A), and Full-Unclipped on the same four 16-query batches, with four responses per query. All methods share the Full-OPD Step100 checkpoint, the FP32 final-block and normalization parameters, reference sets, and radii 0.01 and 0.005. The 56 E3 perturbations cover seven method/seed instances, two radii, and four batches. We verify cached trajectories, teacher scores, parameter identities, and reference hashes before aggregation; repeated no-update NLL discrepancies are zero. Each method retains its loss, unlike the fixed-advantage, fixed-denominator mask comparison.

The diagnostic token selectors use the same implementation as mixed-domain training. TIP and Entropy retain $\max(1, \lfloor 0.1T \rfloor)$ tokens per response. TIP uses its soft-or rule with full-vocabulary

student entropy and KL statistics conditioned on the teacher's top-64 support; Entropy keeps the highest-entropy tokens. TA-OPD keeps $\lceil 0.1N \rceil$ tokens across the batch using the top-16-union approximation and teacher top-64 scores, with missing log probabilities set to -30 and normalization quantiles 0.05/0.95. These three methods use the unclamped sampled-token difference $\log p_T - \log \pi_{\mathrm{FP32}}$ and the selected-token denominator. Full-Unclipped applies the same difference to all valid response positions, isolating the effect of removing advantage clipping under full-token supervision.

CREDIT-A uses a fixed external teacher with ground-truth feedback, random selection of $\lceil 0.1N \rceil$ batch tokens, and the unclamped advantage $\log p_T^+ - \log \pi_{\mathrm{FP32}} - 0.1 \log p_T^-$. Positive and negative teacher contexts share the original query's ground-truth feedback, capped at 1024 tokens; the negative context substitutes a different query from the same batch. One fixed negative assignment (seed 42) is shared by all four responses to each query. Mask seeds 17/42/123 vary token sampling only and are averaged within batch before the four-batch mean. This adaptation uses sampled-token supervision and a fixed external teacher with additional feedback, whereas CREDIT uses an EMA self-teacher and a vocabulary-distribution objective.

Local and terminal method rankings. Table 4 reports the cross-method diagnostics and terminal scores. At $\rho = 0.01$, two-domain NLL gains are 0.013964 for TA-OPD, 0.012640 for CREDIT-A, and 0.011429 for CQF. TA-OPD exceeds CQF's two-domain E3 in all four batches, while CREDIT-A exceeds it in three. TIP improves the two-domain NLL in two batches and worsens it in two, for mean 0.000709. Entropy worsens NLL in all four batches, both domains, and both radii, with a two-domain mean of -0.004021 at $\rho = 0.01$. These local rankings differ from the single-run terminal rankings, leaving the link between one-step NLL gains and terminal accuracy unresolved.

Advantage-clipping control. Full-Unclipped is evaluated only as a diagnostic control, with no terminal benchmark score. Its E1 is 0.3118, Code/Math E2 is 0.2355/0.3042, and Code/Math E3 at $\rho = 0.01$ is 0.009201/0.008480. Its two-domain mean NLL gain is 0.008840, close to Full's 0.009114; the cross-method differences persist.

A common positive scalar denominator cancels from E1, E2, and equal-displacement E3, while it scales the gradient passed to the training optimizer. Its effect over an optimization trajectory is therefore outside these normalized diagnostics. All gradient measurements use the T30 checkpoint; the T4 $\rightarrow$ 0.6B experiments measure terminal benchmark accuracy.

F Training Variability and Cost

Low10 across training seeds. Table 11 reports three Low10 runs at 2K/350: average Mean 41.75 and sample standard deviation 0.43, computed from two-decimal summaries. Seed 42 scores come from evaluation summaries; seeds 123 and 2024 come from source-hashed evaluation logs with aggregate scores and no per-example records. The seed 42 result of 42.25 and the 42.20 result in Figure 2 come from separate runs; the latter's source summary does not record its training seed. These unpaired 2K/350 runs leave main-comparison 7K uncertainty unmeasured. The four diagnostic batches replicate query sampling, not training.

Table 11: T30 $\rightarrow$ 0.6B Low10/Pass0/NR runs at 2K/350 across training seeds. Percentages; these runs are unpaired with Full/random repeats.

Seed	Math	Code	Mean
42	29.31	55.19	42.25
123	28.80	54.31	41.56
2024	27.88	55.03	41.45

Training cost. Table 12 covers 100 steps of the principal T30 $\rightarrow$ 0.6B 7K runs. Allocated GPU-hours equal Slurm elapsed time times eight GPUs (four student/actor, four teacher), measuring reservation rather than utilization. They include initialization, checkpointing, and internal validation, but exclude benchmark evaluation and offline anchor calibration. Runs use different nodes; Full and

Low save once, Random five times. Low consumes 12.5% more GPU-hours than Full. Separating anchor-scoring cost from execution differences requires individual teacher-branch timings.

Table 12: Training cost and token counts for 7K/100-step runs. Token counts are in millions (M); retained positions are a percentage of valid response positions. Actor time is seconds per step. Allocated GPU-hours use the documented eight-GPU setup.

Method	Response tokens (M)	Retained (%)	Actor s/step	GPU-hours
Full	172.35	100.000	41.88	94.44
Low10-NR	174.21	10.003	42.48	106.29
Random10-NR	171.41	10.004	41.81	94.86

Low10 retains about 10% of loss positions, while actor-update time remains close to Full. Generation and model processing still operate on every trajectory, so sparse supervision does not yield a proportional reduction in measured compute. The rollout-stage timer combines asynchronous generation, scoring, and orchestration.