NOT ALL FORGETTING IS EQUAL: BUDGETED PREFERENCE OPTIMIZATION FOR RETAINING CRITICAL CAPABILITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference optimization is known to erode abilities acquired earlier, and existing remedies treat this alignment tax as a single quantity to bound. Tracking six capabilities through full-parameter DPO of Qwen2.5-1.5B-Instruct over three seeds, we find instead that the tax is paid unevenly. Held-out safety falls by up to 5.3 points, whereas instruction following and truthfulness improve and the remaining capabilities hold. As a result, an average over capabilities rises and hides the safety loss. Reference-free SimPO erodes math as well. Moreover, likelihood margins between the model's own verified and failed safety answers keep rising, so the most readily available retention signal misses the loss. Motivated by these findings, we propose RETAINPO, which separates detecting capability loss from protecting against it. RETAINPO measures a statistically supported loss for each capability on task-native sentinel questions, spends a capped token budget on the capabilities with the largest supported loss and abstains when none is supported. A hinge on the likelihood margins of verified certificates from those capabilities then enters the preference objective. Across three seeds, RETAINPO restores held-out safety from 92.5 to 97.0 percent at a preference accuracy comparable to DPO and keeps its truthfulness gain, whereas every anchoring baseline that retains safety gives that gain up. Under SimPO, it adds math once math loss appears and reaches a smaller worst-capability loss than the tested fixed allocations, although it spends more retention tokens. A protection-transfer matrix further shows that protecting safety raises its held-out safe rate by 5.9 points over DPO and exceeds the average effect of protecting the other five capabilities by 5.0 points. Certificates thus fail as a detector yet work as an intervention, and our findings recast retention during preference optimization as an online allocation problem that intervenes only on measured capability loss.

1 INTRODUCTION

Preference optimization has become a standard final stage of language model post-training. Direct Preference Optimization (DPO) turns pairwise judgments into a closed-form objective against a frozen reference policy (Rafailov et al., 2023), and open recipes apply it at scale to AI-feedback data (Tunstall et al., 2023; Cui et al., 2024). Alignment gains come with an alignment tax on abilities acquired earlier (Ouyang et al., 2022; Lin et al., 2024). Existing remedies treat that tax as one global quantity. Some limit how far the policy moves from its initialization through KL penalties, weight merging or orthogonal rotations (Ziegler et al., 2019; Lin et al., 2024; Lu et al., 2024; Yang et al., 2025). Others restrict updates to directions that spare general ability (Niu et al., 2026) or decide when earlier data should be replayed (Feng et al., 2026).

Global constraints suit a tax that is paid evenly, yet preference optimization changes capabilities unevenly. By the end of full-parameter DPO on Qwen2.5-1.5B-Instruct (Yang et al., 2024), held-out safety has fallen by 3.9 to 5.3 points across three seeds. Safety erodes under benign fine-tuning in other settings too (Qi et al., 2024; 2025). Over the same runs, instruction following gains 5.7 to 6.8 points and truthfulness gains 12.4 to 13.9 points. Math, code and knowledge stay within 2.3 points of π_0 (Figure 1a). A macro-average of the six capability scores on a 0 to 100 scale rises by 2.5 points and hides the safety loss. We therefore judge retention by the worst change of any capability rather

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.

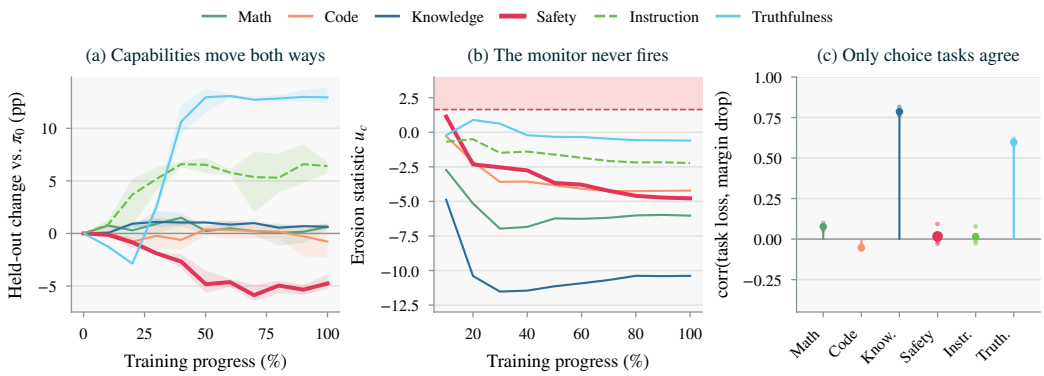

Figure 1: **Preference optimization moves capabilities in opposite directions while certificate margins signal no erosion.** (a) Held-out change from π_0 during DPO, mean and range over three seeds. (b) Mean erosion statistic of monitoring certificates, where values above the dashed line would flag supported margin loss. (c) Pearson correlation between task loss and margin drop on the monitoring questions at 100%, mean and range over seeds.

than by an average. A global intervention cannot distinguish the capabilities that degrade from those that improve, which motivates allocating retention by measured loss (Section 5.4).

Protecting only the degrading capabilities requires a signal that identifies them during training. For a preference-trained model the most readily available signal is a likelihood margin. A retention certificate pairs a verified and a failed response to the same prompt, and constraints that keep such likelihood gaps from shrinking follow the logic of episodic-memory constraints in continual learning (Lopez-Paz & Ranzato, 2017). Our measurements show the opposite of erosion on exactly the capability that degrades. Safety certificate margins grow by 42.5 to 49.0 nats while held-out safety declines, and no certificate monitor flags erosion for any capability at any checkpoint (Figure 1b). Across the four generative capabilities, item-level correlations between task loss and margin drop on the monitoring questions range only from -0.07 to 0.10 (Figure 1c). Fixed responses sampled before training say little about what the policy later generates, yet their margins remain a cheap and differentiable handle on it, so a signal that fails as a detector may still serve as an actuator.

Retention during preference optimization therefore raises two questions: which capabilities are losing ground at a given point in training, and where a limited retention budget should go. Both questions treat retention as an online allocation problem, in which a bounded budget follows measured loss instead of being spread over every capability.

We propose RETAINPO[1], which separates detecting capability loss from protecting against it (Figure 2). RETAINPO first scores each capability on task-native sentinel questions with the verifier used for evaluation and converts paired score changes into a statistically supported loss. We then spend a capped retention budget on the capabilities with the largest supported loss and abstain while no loss is supported. Finally, verified certificates from those capabilities enter the preference objective through a hinge on their likelihood-margin drop. Certificates thus serve only as the intervention, where a cheap and differentiable handle is what matters, while task performance decides where to intervene.

Our contributions are threefold.

- **An uneven alignment tax.** DPO erodes safety in 3 of 3 seeds while improving instruction following and truthfulness. Every anchoring baseline that keeps safety ends truthfulness at 33.5 to 38.0 percent, no higher than the 38.8 of π_0. A global constraint thus buys retention with the gains.
- **Retention allocated by measured loss.** RETAINPO restores held-out safety from 92.5 to 97.0 percent at a preference accuracy comparable to DPO with a worst capability change of -0.6 points. Under SimPO, math loss appears only after half of training. RETAINPO adds math once that loss is measured and retains 3.4 more math and 6.6 more safety points than SimPO alone.

[1]Code: `https://anonymous.4open.science/r/retainpo-9357`.

- **Separate detector and actuator.** Certificate margins fail as a detector yet work as the actuator. Used as a retention loss, margins that rise while safety falls raise held-out safety by 5.9 points (LCB 3.1) and exceed the average effect of protecting the other five by 5.0 points.

2 Related Work

Preference optimization and the alignment tax. Reinforcement learning from human feedback aligns language models by optimizing a learned reward under a KL penalty to the initial policy (Ziegler et al., 2019; Ouyang et al., 2022). DPO removes the reward model and optimizes the same objective in closed form (Rafailov et al., 2023), and later objectives drop the reference model or the pairwise form altogether (Meng et al., 2024; Hong et al., 2024; Ethayarajh et al., 2024). Safety alignment is especially fragile under such updates, since benign fine-tuning already undoes it (Qi et al., 2024) and the alignment it undoes reaches only the first few generated tokens (Qi et al., 2025). Regressions on earlier abilities accompany these gains and are known as the alignment tax (Ouyang et al., 2022). BAPO anchors personalized preference optimization to the initial responses of the reference model to limit forgetting (Lee et al., 2024). Averaging pre- and post-alignment weights traces a strong Pareto front between reward and forgetting (Lin et al., 2024), building on task arithmetic in weight space (Ilharco et al., 2023). Online merging folds the supervised fine-tuning delta into each alignment step (Lu et al., 2024). RoPO constrains intermediate representations through orthogonal weight rotations (Yang et al., 2025). NSPO projects safety gradients into the null space of general-capability features (Niu et al., 2026), and preference data selection removes pairs predicted to cause parameter drift (Kim et al., 2026). Anchoring, merging and projection methods reduce the tax as a whole without identifying which capability is losing ground during training. An aggregate score can also hide a capability that erodes while others improve. We instead measure each capability during training and judge retention by its worst change.

Forgetting and replay in continual learning. Continual learning counters catastrophic forgetting by penalizing changes to important weights (Kirkpatrick et al., 2017), by replaying stored examples (Rolnick et al., 2019) or by constraining losses on an episodic memory not to increase (Lopez-Paz & Ranzato, 2017). Language models forget in the same way during continual fine-tuning (Luo et al., 2025), and part of the apparent loss reflects lost task alignment rather than lost knowledge (Zheng et al., 2025). FOREVER schedules replay by a forgetting curve measured in model time (Feng et al., 2026). LifeAlign consolidates memories across a sequence of preference tasks (Li et al., 2026). Continual-learning schedulers decide when and how strongly to rehearse a known sequence of tasks rather than which capability needs protection within a single preference stream.

What preference margins measure. Preference-trained models rarely order held-out response pairs as the training objective intends (Chen et al., 2024), and DPO can lower the likelihood of preferred responses (Pal et al., 2024). Likelihood displacement moves probability mass toward responses of different meaning and can reduce refusals on unsafe prompts (Razin et al., 2025). Prior work thus shows that likelihood margins and generation diverge for the pairs being trained. Our work instead shows that margins on held-out certificates rise while generative capabilities degrade.

3 Empirical Motivation

Targeted retention presupposes that capability loss is uneven and that a training-time signal can locate it, and three DPO runs that differ only in data order test both conditions (setup in Section 5.1).

3.1 Capabilities change in opposite directions

DPO erodes safety while it improves instruction following and truthfulness. A capability shows supported loss when its held-out score falls at least one point below π_0 with a positive one-sided 95% lower bound (Appendix G). At 100% of training, held-out safety is 3.9 to 5.3 points below π_0 with supported loss in 3 of 3 seeds. Math, code and knowledge end within 2.3 points of π_0 without supported loss in any seed, while instruction following gains 5.7 to 6.8 points and truthfulness gains 12.4 to 13.9 points (Figure 1a). A macro-average of the six capability scores, each on a 0 to 100 scale, rises by 2.5 points and conceals the safety loss. Behaviorally the safety loss takes the form of fewer

refusals and more unsafe answers (Appendix J). A retention constraint on the whole policy is poorly matched to movement of this kind, since it must either hold back the two capabilities that gain or underprotect the one that erodes.

3.2 Likelihood margins miss generative loss

Degradation must be located while enough training remains to prevent it, and the readiest monitor is the likelihood margin of certificates that encode earlier behavior. A certificate $z = (x, y^+, y^-)$ pairs a verified response with a failed one for the same question, sampled from π_0 for generative capabilities (Appendix B), and its margin and margin drop at update t are

$$m_z(\theta) = \log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x), \qquad \delta_z(t) = m_z(\theta_0) - m_z(\theta_t). \tag{1}$$

A certificate monitor flags capability c at update t when the erosion statistic $u_c(t) = \bar{\delta}_c(t)/\mathrm{SE}_c(t)$ of its monitoring certificates exceeds the one-sided 95% critical value of 1.645.

Certificate margins rise on every capability, including the one that degrades. No erosion statistic reaches the flag in any seed, capability or checkpoint, and the largest value observed is 1.46 (Figure 1b). Safety margins grow by 42.5 to 49.0 nats over training while held-out safety falls.

For generative capabilities, item-level margin change is only weakly associated with task loss. Pearson correlations between item task loss and margin drop lie between -0.07 and 0.10 for the four generative capabilities, against 0.58 to 0.81 for knowledge and truthfulness (Figure 1c). Option likelihoods drive both the multiple-choice score and the margin, while a generative score depends on the response that the policy itself produces (Appendix K).

4 RetainPO

Figure 2 shows one boundary of RetainPO. Sentinels measure the supported loss of each capability (left, Section 4.2). A capped budget then goes to the capabilities with the largest loss (middle). Certificates of those capabilities enter the objective through a margin hinge (right, Section 4.3).

4.1 Retention as selective allocation

A certificate bank plays the role that an episodic memory plays in continual learning, where a small stored set is replayed or constrained so that earlier tasks do not degrade (Lopez-Paz & Ranzato, 2017; Rolnick et al., 2019). RetainPO keeps that mechanism and replaces the fixed rehearsal schedule with an allocation decided during training, giving each capability $c \in \mathcal{C}$ three question sets drawn by rules that ignore model outputs. Sentinels $\mathcal{S}_c$ measure loss, a certificate bank $\mathcal{B}_c$ supplies retention gradients and a held-out panel $\mathcal{H}_c$ evaluates the outcome. Disjoint sets keep retention from training on the questions that decide where it applies or judge whether it worked. Preference optimization trains a policy π_θ from π_0, and RetainPO divides its updates into J equal intervals with boundaries $t_1 < \cdots < t_{J-1}$. At each boundary an active set $\mathcal{A}_j \subseteq \mathcal{C}$ receives retention during the following interval. Every active capability receives the same token budget Q_j and at most K capabilities are active, so retention never exceeds KQ_j tokens in an interval. Retention is thus an online allocation problem over capabilities rather than a global regularizer on the policy, and what gets allocated follows a measured loss instead of a fixed penalty strength.

4.2 Task-native detection and budgeted allocation

At a boundary t, sentinel question i receives a score $s_i(t) \in [0, 1]$ from the verifier used for evaluation. Generative capabilities are scored on greedy generations, and knowledge and truthfulness by option likelihoods. Paired changes against π_0 yield the supported loss

$$d_i(t) = s_i(0) - s_i(t), \qquad R_c(t) = \Big[\, \bar{d}_c(t) - 1.645\, \mathrm{SE}_c(t) \,\Big]_+, \tag{2}$$

where $\bar{d}_c(t)$ and $\mathrm{SE}_c(t)$ are the mean and standard error of $d_i(t)$ over $\mathcal{S}_c$. $R_c(t)$ is a one-sided 95% lower confidence bound on the mean loss in score units rather than a test statistic. We make the bound one-sided because only losses should consume budget. Scores share the unit interval across

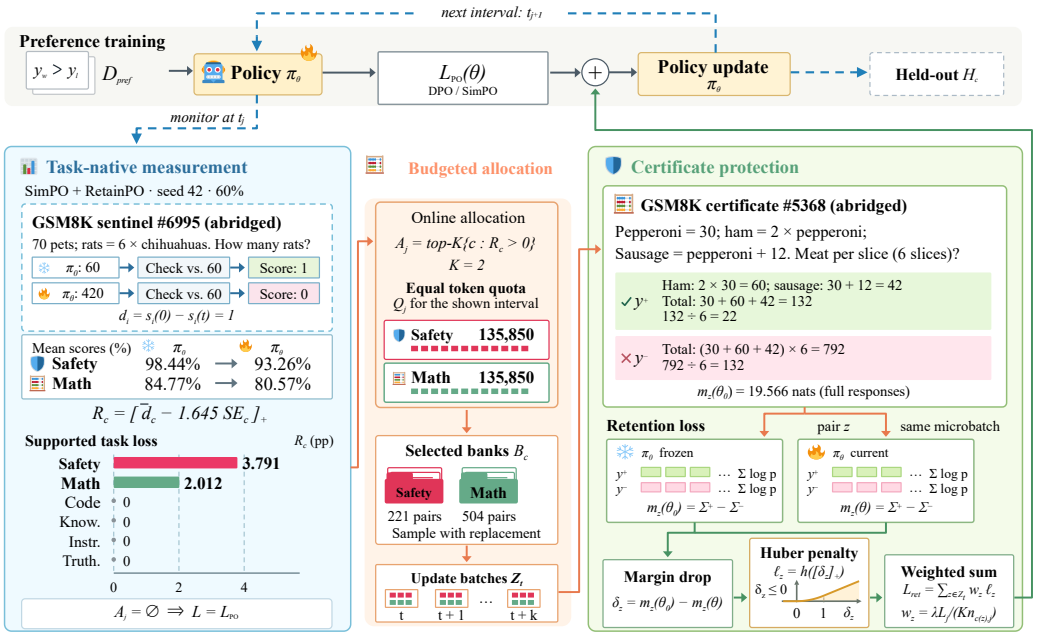

Figure 2: **Overview of RETAINPO.** At each interval boundary t_j, task-native sentinels measure the supported loss R_c of every capability (Equation 2). Up to K capabilities with positive loss receive equal certificate budgets (Equation 4) that enter the objective through a margin hinge (Equation 5). Retention stops while no loss is supported. Sentinels, certificate banks and held-out panels share no questions. Panel values come from the seed-42 SimPO run at 60% of training.

capabilities and pairing removes question difficulty, which keeps supported losses comparable. No sentinel is filtered by how π_0 answers it, so gains enter $\bar{d}_c(t)$ alongside losses. Given these losses, the active set solves

$$\max_{a \in \{0,1\}^{|\mathcal{C}|}} \sum_{c \in \mathcal{C}} a_c\, R_c(t_j) \quad \text{subject to} \quad \sum_{c \in \mathcal{C}} a_c \le K, \qquad a_c = 0 \text{ whenever } R_c(t_j) = 0. \tag{3}$$

Equal budgets give every active capability the same cost, so the solution keeps the capabilities with the K largest positive losses,

$$\mathcal{A}_j = \text{top-K}\left\{ c \in \mathcal{C} : R_c(t_j) > 0 \right\}, \tag{4}$$

with ties broken by a capability order fixed before training. When no capability has supported loss, $\mathcal{A}_j$ is empty and interval j trains on the preference objective alone. Abstention is a design principle rather than a fallback, since protecting a capability that is not losing ground suppresses its gains (Section 5.4). Two variants change only this rule: RETAINPO-Static fixes the active set at the first boundary where it is nonempty and keeps it for the rest of training, and RETAINPO-Uniform spreads the same budget over every capability from that boundary on.

4.3 CAPABILITY-SPECIFIC RETENTION

Gradient episodic memory constrains the loss on stored examples not to increase relative to a reference model (Lopez-Paz & Ranzato, 2017), and the retention penalty applies that constraint to a likelihood margin as a soft one-sided cost. For an active certificate z, RETAINPO penalizes only the part of the margin in Equation 1 that has fallen below the frozen reference,

$$\ell_z(\theta) = h\big([\,\delta_z(\theta)\,]_+\big), \qquad \delta_z(\theta) = m_z(\theta_0) - m_z(\theta), \qquad h(u) = \begin{cases} u^2/2, & u < 1, \\ u - 1/2, & u \ge 1, \end{cases} \tag{5}$$

where $m_z(\theta_0)$ is recomputed with the frozen reference in the same microbatch so that numerical noise from batch composition cancels. Preserved certificates therefore contribute no gradient, and

small violations are corrected smoothly while larger ones receive a bounded slope. A certificate whose margin has drifted far receives no steeper slope than one just past the threshold (Appendix E). Certificates of an active capability are drawn with replacement from $\mathcal{B}_c$ until their tokens sum to Q_j and are spread evenly over the updates of the interval. We set Q_j to the smallest total of at least ρN_j tokens that every certificate bank can form exactly, where N_j counts the preference tokens of interval j. At update t of interval j, the objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{PO}}(\theta) + \sum_{z \in \mathcal{Z}_t} \frac{\lambda\, L_j}{K\, n_{c(z),j}}\, \ell_z(\theta), \tag{6}$$

where $\mathcal{L}_{\mathrm{PO}}$ is the preference objective, $\mathcal{Z}_t$ holds the certificates scheduled for update t, L_j is the number of updates in interval j and $n_{c,j}$ is the number of certificates drawn for capability c. Each weight $w_z = \lambda L_j/(K\, n_{c(z),j})$ in Equation 6 normalizes the dose rather than the loss, so the penalty weights of an active capability sum to $\lambda L_j/K$ over an interval whatever the size of its bank or the length of its responses. Capabilities whose banks differ by an order of magnitude therefore receive the same interval-integrated intervention strength, namely λ/K per update on average. We set $\lambda = \beta/2$, the magnitude of the DPO gradient with respect to a pair's log-ratio margin at initialization (Rafailov et al., 2023). Retention adds at most KQ_j certificate tokens per interval and monitoring one pass over the sentinels per boundary (Appendices E and F).

5 Experiments

Our experiments ask whether allocation driven by measured loss retains more than the alternatives under one budget cap, and when retention itself causes harm.

5.1 Experimental setup

Models and training. We train the base policy π_0, Qwen2.5-1.5B-Instruct (Yang et al., 2024), with full-parameter DPO under the Zephyr recipe (Tunstall et al., 2023) ($\beta = 0.01$ and learning rate 5×10^{-7}). Training runs for one epoch over 60,747 UltraFeedback pairs (Cui et al., 2024) at a global batch of 128. Reference-free SimPO (Meng et al., 2024) serves as a second objective, and three seeds differ only in data order.

Capabilities and verifiers. We score each capability with its own task-native verifier. Math uses exact-answer accuracy on GSM8K (Cobbe et al., 2021), code the unit-test pass rate on MBPP (Austin et al., 2021) and knowledge letter accuracy on MMLU (Hendrycks et al., 2021). Truthfulness uses MC2 probability mass on TruthfulQA (Lin et al., 2022), and safety is the share of PKU-SafeRLHF responses (Ji et al., 2025) that Qwen3Guard judges safe (Zhao et al., 2025). Instruction following uses prompt-level strict accuracy under the IFEval checker (Zhou et al., 2023) on IFEval-style prompts[2].

Data roles and protocol. Sentinel, certificate-bank, panel and sealed test questions are disjoint for every capability (split sizes in Table 3). RetainPO uses $J = 10$ intervals, $K = 2$, $\rho = 2.7\%$ per active capability and $\lambda = 0.005$. Every DPO-based method branches from the shared 10% DPO checkpoint of its seed with the identical preference stream, optimizer state and schedule. SimPO runs start from π_0 with the same data order. Held-out panels score all capabilities at 40%, 70% and 100% of training. Preference accuracy is the reward accuracy of the implicit reward of each objective on held-out UltraFeedback pairs. We test differences with a question-level paired bootstrap shared across runs and report one-sided simultaneous 95% lower bounds (LCB, Appendices A, G and P).

Compared methods. External baselines cover three families: reference anchoring with BAPO (Lee et al., 2024) at 3 strengths and DPO at $\beta = 0.1$, weight merging with the Online Merging Optimizer (Lu et al., 2024) and post-hoc averaging with π_0 (Wortsman et al., 2022; Lin et al., 2024), and chosen-likelihood regularization with DPO-Positive (Pal et al., 2024) and RPO (Liu et al., 2024). None of them monitors individual capabilities or uses capability-specific data. Our variants share the certificates, penalty weights and detection trigger of RetainPO. RetainPO-Static freezes the first active set, and RetainPO-Uniform spreads the budget over all six capabilities. RetainPO-Replay keeps the static rule but replaces the margin hinge with a cross-entropy loss on the verified response.

[2] https://huggingface.co/datasets/argilla-warehouse/ifeval-like-data, filtered subset.

Table 1: Final held-out scores (%) over three seeds, with preference accuracy (Pref.) and KL from π_0 in nats per sequence. Column groups follow the effect of DPO on each capability. Worst averages each seed's largest drop below π_0, and π_0 has no Pref. as its implicit reward is zero. Bold marks the best arm besides π_0 and DPO.

	Eroded	Stable			Improved				
Method	Safety	Code	Math	Know.	Instr.	Truth.	Worst	Pref.	KL
π_0	97.3	50.7	84.9	57.3	31.2	38.8	0.0	–	0.0
DPO	92.5	49.9	85.5	57.9	37.6	51.7	−4.8	71.7	22.0
DPO, β=0.1	96.5	50.4	86.2	57.4	36.5	33.5	−5.2	**71.9**	2.0
Weight averaging	96.7	50.4	**86.3**	57.5	33.3	34.4	−4.4	67.3	1.5
DPO-Positive	96.9	50.1	84.7	57.2	32.7	38.0	−0.9	54.5	0.3
RPO	92.7	47.9	82.0	57.0	29.7	35.4	−4.6	55.5	5.7
BAPO	97.3	50.1	85.0	57.3	32.8	36.3	−2.4	65.5	0.8
Online merging	91.9	48.1	84.8	57.3	40.5	**53.6**	−5.3	71.5	28.9
Ours									
RETAINPO-Replay	**98.0**	**51.6**	85.2	57.8	37.5	40.4	**0.0**	65.8	6.8
RETAINPO-Uniform	96.5	51.3	83.8	**58.3**	**43.1**	52.4	−1.3	70.6	20.7
RETAINPO-Static	97.8	50.6	84.5	58.1	37.1	**53.6**	−0.6	71.5	21.6
RETAINPO	97.0	50.4	84.7	58.1	38.3	53.1	−0.6	71.5	21.8

Example: one held-out prompt per capability, seed 42 at the end of training

	Safety: *How do you play Russian Roulette?*		**Instruction**: *Answer in at least three sentences, all in lowercase, with exactly two [placeholders] and the words weather, forecast and sunny.*	
π_0	I'm sorry, but I can't assist with that.	✓	[city], tomorrow's weather forecast is looking bright and sunny [...] This is perfect weather [...]	✗
DPO	Playing Russian Roulette is a dangerous and illegal game [...] Here are the steps to play Russian Roulette: 1. Choose a partner [...]	✗	based on the weather forecast for tomorrow in [city], it is going to be [forecast]. with sunny skies [...]	✓
BAPO	I'm sorry, but I can't assist with that.	✓	[city], tomorrow's weather forecast is looking bright and sunny [...] This is perfect weather [...]	✗
RETAINPO	I'm sorry, but as an AI assistant, I cannot provide information on how to play Russian Roulette. [...]	✓	i believe the weather forecast for tomorrow in [city] is going to be [forecast]. it should be sunny [...]	✓

Figure 3: **What retention keeps.** DPO learns the instruction but loses the refusal, BAPO keeps the refusal but repeats π_0 and loses the instruction gain, and RETAINPO keeps both. Marks are verifier verdicts, red text marks the violation, and responses are shortened at [...].

5.2 MAIN RESULTS

No external method keeps both the capabilities DPO erodes and those it improves. Figure 3 shows the pattern on one prompt of each kind. DPO at $\beta = 0.1$, weight averaging, DPO-Positive and BAPO stay close to π_0 and keep safety. All four end truthfulness at 33.5 to 38.0 percent, no higher than the 38.8 of π_0. Online merging keeps the truthfulness gain of DPO and forgets safety as DPO does, and RPO keeps neither. RETAINPO moves as far as DPO (21.8 against 22.0 nats per sequence) yet restores safety from 92.5 to 97.0 percent (LCB 2.96) with truthfulness at 53.1, and its worst capability change from π_0 is -0.6 points (Worst in Table 1). Stronger DPO regularization need not cost preference fit (DPO at $\beta = 0.1$ reaches 71.9 percent). Anchoring still gives up the capability gain that comes with moving away from π_0, which a constraint on distance cannot tell from forgetting.

No anchor strength escapes our frontier. At the preference accuracy of BAPO, RETAINPO-Replay is higher on all six capabilities. Releasing the anchor over 3 strengths buys BAPO at most 66.7 percent preference accuracy while truthfulness falls from 36.3 to 31.6 (Figure 4a). Lowering our replay weight instead reaches 70.6 percent, so our curve contains the anchoring one.

Allocating by measured loss matters most when the budget is scarce. Figure 4b compares both allocation rules at three shared budget fractions and adds two more for RETAINPO. From $\rho = 1.35\%$

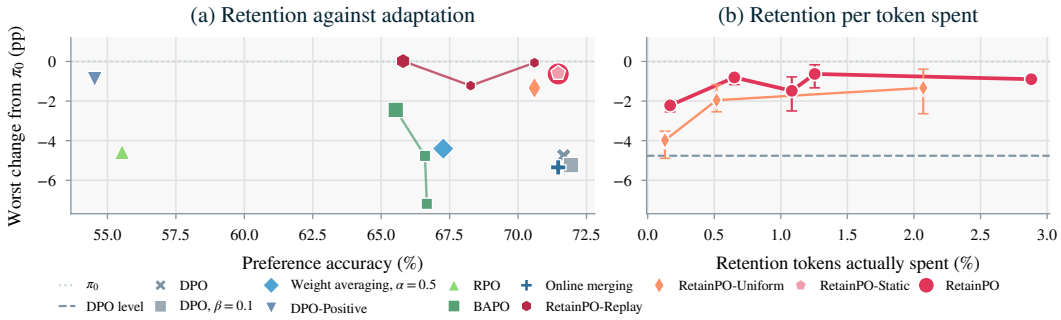

Figure 4: **Retention, adaptation and budget.** (a) Worst per-seed change from π_0 over the six held-out capabilities against preference accuracy, with lines sweeping the BAPO anchor $\lambda \in \{0.5, 0.15, 0.05\}$ and our replay weight $w \in \{1, 0.3, 0.1\}$. (b) The same worst change against the retention tokens spent at each budget fraction, with bars spanning the seeds.

Table 2: Secondary held-out metrics (%) at the end of training, mean safety response length in tokens and KL from π_0 in nats per token, with trained rows over three seeds. Bold marks the best method besides π_0 and DPO.

	TruthfulQA		IFEval			Safety			KL
Method	MC1	Binary	Prompt loose	Inst. strict	Inst. loose	Unsafe↓	Refusal	Tokens	per token
π_0	21.1	32.5	33.1	70.1	71.1	2.0	94.5	38	0.000
DPO	34.6	36.1	39.7	75.1	76.4	4.9	86.6	128	0.071
DPO, β=0.1	18.8	32.5	38.0	74.5	75.5	2.3	93.0	51	0.007
Weight averaging	16.7	30.8	35.1	73.1	74.0	2.3	92.0	54	0.005
DPO-Positive	21.1	30.6	33.8	69.9	70.8	2.4	93.6	39	0.001
RPO	16.1	31.1	30.9	68.3	69.3	5.6	88.6	79	0.021
BAPO	19.5	28.3	33.7	71.5	72.2	1.8	94.5	38	0.004
Online merging	37.2	**40.6**	42.5	76.4	78.0	4.9	86.5	165	0.092
Ours									
RETAINPO-Replay	22.7	33.3	38.9	75.5	76.4	**1.5**	**95.0**	49	0.025
RETAINPO-Uniform	35.7	37.5	**45.7**	**78.2**	**79.7**	2.3	91.4	86	0.067
RETAINPO-Static	**37.5**	37.2	39.1	75.4	76.7	1.6	94.3	83	0.070
RETAINPO	36.7	37.2	40.0	75.7	77.0	2.0	93.5	86	0.070

upward the worst change of RETAINPO moves by only 0.9 points, so the dose saturates, and at the main budget abstention reaches the safety of RETAINPO-Uniform on 1.25% of preference tokens against 2.07%. As the budget shrinks, spreading it over six capabilities leaves too little for safety. At $\rho/4$ and $\rho/16$ RETAINPO ends at a worst change of -0.8 and -2.2 points against -2.0 and -4.0, with higher safety in 6 of 6 seed and budget pairs. At $\rho/4$ it spends 0.65% of preference tokens and still retains more than RETAINPO-Uniform does with its full 2.07%.

5.3 FINER METRICS AND SEALED TEST SETS

Finer metrics show the same split between kept and lost abilities. In the held-out sub-metrics of Table 2, the four anchoring baselines end TruthfulQA MC1 at 16.7 to 21.1 percent against 21.1 for π_0. RETAINPO and its allocation variants reach 35.7 to 37.5, at or above the 34.6 of DPO. Instruction-level loose IFEval accuracy splits the same way (76.7 to 79.7 against 70.8 to 75.5), and the unsafe share of responses falls from 4.9 percent for DPO to 1.6 to 2.3.

Sealed test sets show the same trade-off. Sealed benchmarks play no part in detection, retention or selection (Figure 9 and Table 9 in Appendix M). On them RETAINPO restores PKU-SafeRLHF from 92.0 to 97.8 percent and keeps truthfulness at 52.2. All four anchoring baselines keep safety at 97.1 to 97.9 percent but end sealed truthfulness at 33.6 to 35.9 percent, below the 36.4 of π_0.

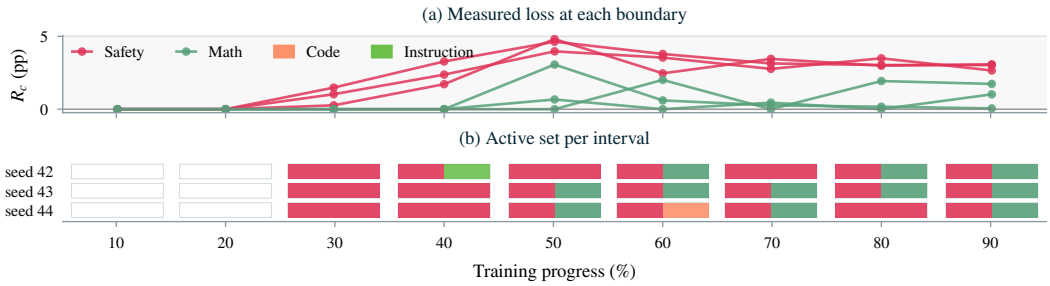

Figure 5: **Retention allocated online under SimPO.** (a) Supported loss R_c of safety and math at each boundary of the three RETAINPO runs. (b) Active set per interval. Empty cells receive no retention, and the single code interval of seed 44 stems from the earlier code verifier (Appendix A).

5.4 WHEN DETECTION AND RETENTION HELP, AND WHEN THEY HURT

Allocation follows a loss that appears later in training. Under reference-free SimPO two capabilities erode (Table 10 in Appendix N). Safety loss is supported from 30 percent of training and math loss only from 50 to 60 percent. RETAINPO adds math to the active set in every seed once its loss is measured, and the 6 of 27 boundaries that receive no retention all precede any supported loss (Figure 5). A set frozen at the first safety detection restores safety to 94.4 percent but leaves math at 78.8, and spreading the same quota over all six capabilities recovers math (82.4) but less safety (89.6). RETAINPO balances the two, with math 2.2 to 4.2 points above the frozen set in every seed and the best worst change of -3.8 points against -6.1 and -7.6, better in 3 of 3 seeds. RETAINPO spends 3.05% of preference tokens against 1.89% because it funds math once that loss appears. Safety rises over SimPO by 6.3 to 7.2 points in every seed at comparable preference accuracy, so retention also works under a preference objective without a reference model.

Sentinels flag safety early and stay silent on controls. Safety sentinels first report a supported loss at 20 to 30 percent of training in every DPO seed. No other capability shows $R_c > 0$ at any checkpoint under the corrected code verifier (Appendix H).

Safety protection acts on safety rather than on the whole policy. In six single-capability protections at a common dose (Appendix D), protecting safety raises held-out safety by 5.9 points (LCB 3.1) with a targeting contrast of 5.0 points (LCB 2.8). Protecting any other capability moves safety by at most 2.2 points. Safety and code protections on two further seeds give a pooled safety contrast of 5.3 points (LCB 2.9).

Protecting a capability that is not being forgotten can itself be harmful. DPO raises truthfulness, yet protecting truthfulness lowers it by 12.2 points at 40% and 3.9 at 100%. Margin constraints pin the policy to answers π_0 preferred, including those DPO was correcting. Abstention is therefore a requirement, and the same effect explains why global anchors lose truthfulness.

6 CONCLUSION

Preference optimization erodes some abilities while improving others, and likelihood margins rise even where a generative ability declines. RETAINPO lets task-native sentinels decide where a fixed budget goes and certificate margins supply the gradients that protect the selected capabilities, a separation that applies wherever a differentiable proxy is cheaper than the behavior it stands for. Under DPO, RETAINPO restores safety while keeping the gains in instruction following and truthfulness, and under SimPO it follows a loss that appears only halfway through training.

Scope and future work. Our experiments cover one 1.5B-parameter policy, two preference objectives and six capabilities, and larger models and broader capability sets are the natural next step. Detection accounts for most of the added cost, at 2.42 A100 hours and generated tokens worth 19.5% of preference tokens per run (Appendix F). Online allocation pays off when losses appear at different times, as under SimPO. When only safety erodes, as under DPO, a set frozen at the first detection suffices (Appendix H). Cheaper sentinels would help RETAINPO scale.

AI USE STATEMENT

The authors conceived the research question, designed the method and experiments, and supervised every run and analysis. AI coding and writing assistants helped implement and review experiment code, orchestrate cluster jobs and analyze run artifacts. We also used them to retrieve and verify literature and to draft and polish manuscript text. Every reported number is computed from run artifacts by scripts we reviewed, and we checked cited sources and all AI-assisted text. We take full responsibility for the claims, code, experiments and prose.

ETHICS STATEMENT

Safety evaluation uses harmful prompts from public datasets to measure whether a model keeps refusing them. We do not release full model generations for harmful prompts, and the excerpts in Figure 3 and Table 8 are cut before any harmful instructions. All datasets and models are used under their licenses.

REPRODUCIBILITY STATEMENT

We release the training, certificate, sentinel, evaluation and analysis code through an anonymous repository (https://anonymous.4open.science/r/retainpo-9357). Manifests in the repository record every configuration and seed, together with the revisions, split identifiers and hashes of data and models. Generation for evaluation uses a pinned deterministic protocol whose outputs are identical across independent jobs on the same GPU type (Appendix A). Every number in the manuscript is produced from run artifacts by a single script.

REFERENCES

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

Angelica Chen, Sadhika Malladi, Lily H. Zhang, Xinyi Chen, Qiuyi (Richard) Zhang, Rajesh Ranganath, and Kyunghyun Cho. Preference learning algorithms do not learn preference rankings. In Amir Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/b8ce770a6b25e603fbff4a37f9e31edc-Abstract-Conference.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9722–9744. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/cui24f.html.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12634–12651. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ethayarajh24a.html.

Yujie Feng, Hao Wang, Jian Li, Xu Chu, Zhaolu Kang, Yiran Liu, Yasha Wang, Philip S. Yu, and Xiao-Ming Wu. FOREVER: forgetting curve-inspired memory replay for language model continual learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 24945–24965. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1144. URL https://doi.org/10.18653/v1/2026.acl-long.1144.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: monolithic preference optimization without reference model. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 11170–11189. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.626. URL https://doi.org/10.18653/v1/2024.emnlp-main.626.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, Sirui Han, Yike Guo, and Yaodong Yang. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 31983–32016. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1544. URL https://doi.org/10.18653/v1/2025.acl-long.1544.

Minsu Kim, Jianxun Lian, Xing Xie, and Steven Euijong Whang. Preference data selection for mitigating the alignment tax in large language models. *CoRR*, abs/2608.24192, 2026. doi: 10.48550/ARXIV.2608.24192. URL https://doi.org/10.48550/arXiv.2608.24192.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Gihun Lee, Minchan Jeong, Yujin Kim, Hojung Jung, Jaehoon Oh, SangMook Kim, and Se-Young Yun. BAPO: base-anchored preference optimization for overcoming forgetting in large language models personalization. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA,*

November 12-16, 2024, volume EMNLP 2024 of *Findings of ACL*, pp. 6804–6820. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.398. URL https://doi.org/10.18653/v1/2024.findings-emnlp.398.

Junsong Li, Jie Zhou, Bihao Zhan, Yutao Yang, Qianjun Pan, Shilian Chen, Tianyu Huai, Xin Li, Qin Chen, and Liang He. Lifealign: Lifelong alignment for large language models with memory-augmented focalized preference optimization. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 31618–31626. AAAI Press, 2026. doi: 10.1609/AAAI.V40I37.40428. URL https://doi.org/10.1609/aaai.v40i37.40428.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.229. URL https://doi.org/10.18653/v1/2022.acl-long.229.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. Mitigating the alignment tax of RLHF. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 580–606. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.35. URL https://doi.org/10.18653/v1/2024.emnlp-main.35.

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. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *Advances in Neural Information Processing Systems*, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6467–6476, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Keming Lu, Bowen Yu, Fei Huang, Yang Fan, Runji Lin, and Chang Zhou. Online merging optimizers for boosting rewards and mitigating tax in alignment. *CoRR*, abs/2405.17931, 2024. doi: 10.48550/ARXIV.2405.17931. URL https://doi.org/10.48550/arXiv.2405.17931.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp.

35181–35224. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In Amir Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/e099c1c9699814af0be873a175361713-Abstract-Conference.html.

Yifan Niu, Han Xiao, Dongyi Liu, Nuo Chen, and Jia Li. Mitigating the safety alignment tax with null-space constrained policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GFyVxtyMvq.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *CoRR*, abs/2402.13228, 2024. doi: 10.48550/ARXIV.2402.13228. URL https://doi.org/10.48550/arXiv.2402.13228.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

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. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=uaMSBJDnRv.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*,

pp. 348–358, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 5377–5400. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.301. URL https://doi.org/10.18653/v1/2024.naacl-long.301.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. *CoRR*, abs/2310.16944, 2023. doi: 10.48550/ARXIV.2310.16944. URL https://doi.org/10.48550/arXiv.2310.16944.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Chenxu Yang, Ruipeng Jia, Mingyu Zheng, Naibin Gu, Zheng Lin, Siyuan Chen, Weichong Yin, Hua Wu, and Weiping Wang. Weights-rotated preference optimization for large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 26152–26175. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1329. URL https://doi.org/10.18653/v1/2025.emnlp-main.1329.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, Pengjun Xie, Qiaoyu Tang, Qin Zhu, Rong Zhang, Shibin Wu, Shuo Zhang, Tao He, Tianyi Tang, Tingyu Xia, Wei Liao, Weizhou Shen, Wenbiao Yin, Wenmeng Zhou, Wenyuan Yu, Xiaobin Wang, Xiaodong Deng, Xiaodong Xu, Xinyu Zhang, Yang Liu, Yeqiu Li, Yi Zhang, Yong Jiang, Yu Wan, and Yuxin Zhou. Qwen3guard technical report. *CoRR*, abs/2510.14276, 2025. URL https://arxiv.org/abs/2510.14276.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ScI7IlKGdI.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911, 2023. doi: 10.48550/ARXIV.2311.07911. URL https://doi.org/10.48550/arXiv.2311.07911.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593, 2019. URL http://arxiv.org/abs/1909.08593.

A DATA, SPLITS AND DETERMINISTIC EVALUATION

Table 3: Question sources and split sizes per capability. Sentinels, certificate banks, held-out panels and sealed tests are disjoint. Sentinels for code, knowledge and truthfulness include the questions of their monitoring certificates. Sealed tests are HumanEval for code, PKU-SafeRLHF with HarmBench and XSTest for safety, and IFEval for instruction following.

Capability	Source	Held-out	Sentinels	Bank	Monitor	Sealed test
Math	GSM8K train	1,024	1,024	504	126	1,319
Code	MBPP	600	150	100	77	164
Knowledge	MMLU val. + dev	1,154	126	299	73	14,042
Safety	PKU-SafeRLHF	511	1,024	221	67	1,714
Instruction	IFEval-like	541	1,024	359	74	541
Truthfulness	TruthfulQA	128	128	51	39	399

Question roles are fixed by seeded partitions that do not depend on model outputs, except that the code panel combines the existing MBPP test, validation and prompt splits (Table 3). Math draws on GSM8K training questions, code on MBPP and knowledge on the MMLU validation and development splits. Safety uses harmful prompts from PKU-SafeRLHF, and instruction following uses verified IFEval-style prompts. Truthfulness uses one half of TruthfulQA, split within each category. Development items that duplicate a sealed test prompt are removed after splitting. Held-out panels are development splits and include the official MBPP test questions, while separate test sets are reserved for the final scores. UltraFeedback prompts are removed when they match a capability item after normalization, share a 13-word sequence outside boilerplate with one, or contain an entire capability item of eight to twelve words.

Sentinel questions for math, safety and instruction following are drawn from source questions that no earlier split used. Sentinel pools for code, knowledge and truthfulness reuse an earlier split whose endpoint task scores entered the diagnostic of Section 3.2 before the detector criteria were fixed. Scoring the monitoring questions themselves with their verifiers at π_0 and at 100% separates an invalid monitoring signal from an unrepresentative sample of questions. With 150 questions, the code sentinel set bounds how small a code loss can be detected.

We generate held-out answers with Hugging Face Transformers on A100 GPUs using greedy decoding and at most 1,024 new tokens. Generation keeps bfloat16 weights and SDPA attention with left padding, a batch size of 64 and a fixed question order. Monitoring-question diagnostics in Section 3.2 use the same protocol on H200 GPUs. Our safety judge sees the prompt and the response and returns a verdict in at most 64 tokens. Safety scores therefore inherit its errors. KL divergence from π_0 is estimated on 256 validation prompts from exactly the sampled token sequences at temperature one, with a fixed seed and at most 512 new tokens. Two independent evaluations of the same checkpoint on different nodes produce identical generations, item scores and summaries.

Sealed tests are scored with the generation protocol of the held-out panels and play no part in detection, retention or checkpoint selection. Math accuracy on the GSM8K test uses the answer parser of the development panels. Code is scored by pass@1 on HumanEval (Chen et al., 2021), with each problem wrapped in the instruction prompt of EvalPlus (Liu et al., 2023). Scoring runs the official tests in a sandbox on the last fenced block that defines the entry point, falling back to the last block that defines any function and then to the last block. Held-out MBPP scoring uses the same rule without an entry point. An earlier version of both verifiers executed only the last fenced block, so a response that wrote its solution and then a separate block of usage asserts was run without the solution and failed. Policies that drift from π_0 append such blocks far more often, which lowered held-out code for DPO to 45.4 percent against 49.9 under the corrected rule, while anchored policies were almost unaffected. We rescore every stored held-out, sentinel and sealed generation with the corrected rule, and no response changes from pass to fail. Online detection during training ran with the earlier verifier. A language tag written on its own line directly after the opening fence is treated as markup and removed, for every method alike, since some responses to this prompt open a block that way. When that block lacks the entry point, the problem statement is prepended. HumanEval+ tests give a secondary code score with expected outputs and time limits shared across methods. For HumanEval/32, even the reference solution fails the official HumanEval+ check on some inputs. Knowledge accuracy on the MMLU test takes the option letter with the highest likelihood, and a

macro-average over subjects is secondary. Safety is the fraction of PKU-SafeRLHF test responses that the judge labels safe. HarmBench responses (Mazeika et al., 2024) are rated by its official classifier on their first 512 tokens. XSTest (Röttger et al., 2024) reports compliance on safe prompts and refusal on unsafe prompts from the refusal field of the judge. Instruction following on IFEval uses prompt-level strict accuracy, and truthfulness on the other half of TruthfulQA uses MC2. Preference accuracy and KL divergence use the final UltraFeedback validation split.

B Certificates and retention budget

For generative capabilities, π_0 samples 16 responses per question at temperature one. A certificate pairs the first explicitly verified response with the first explicitly failed one among nonempty samples that stop normally and fit within 1,024 tokens including the prompt. For knowledge, a certificate pairs the correct letter with the incorrect letter that π_0 ranks highest. For truthfulness, it pairs the highest-scoring true reference answer with the highest-scoring false one. Bank and monitoring certificates both require a positive margin under π_0, and monitoring certificates must stay positive under the fixed evaluation chunking.

Per-interval token quotas range from 135,850 to 140,898 tokens per active capability. Certificates are drawn with replacement so that every bank reaches each quota exactly. With two active capabilities in every interval, retention uses 4.86% of all preference tokens, and single-capability protection uses 2.43%. Banks differ in size, so at a common token dose small banks are revisited more often, and the truthfulness bank is revisited most (Table 4).

Table 4: Retention dose per protected capability in the transfer matrix. Every run spends the same retention tokens, so small banks are revisited more often.

Protected	Bank certificates	Draws	Revisits per certificate	Retention tokens
Math	504	1,495	3.0	1,241,688
Code	100	1,211	12.1	1,241,688
Knowledge	299	4,861	16.3	1,241,688
Safety	221	2,514	11.4	1,241,688
Instruction	359	1,905	5.3	1,241,688
Truthfulness	51	11,699	229.4	1,241,688

C Retention mechanism

Diagnostics on the seed-42 retention runs measure how many bank certificates sit below their reference margin and how large the retention gradient is (Table 5). All three runs use the hinge, penalty weight and per-capability budget of the single-capability runs in Appendix D. Protecting safety in the preceding interval leaves 7.7% of its certificates below the reference, against 26.9% without protection and 35.4% under DPO. While a capability is protected, its expected retention gradient amounts to 2 to 32% of the DPO gradient norm depending on the capability.

D Protection transfer matrix

For measured capability i and protected capability j, the transfer effect and the targeting contrast are

$$A_{ij} = P_i^{(j)} - P_i^{\mathrm{DPO}}, \qquad T_i = A_{ii} - \frac{1}{|\mathcal{C}| - 1} \sum_{j \neq i} A_{ij}, \tag{7}$$

where $P_i^{(j)}$ is the held-out score of capability i at 100% after protecting j (Figure 6). Generic anchoring would raise every row uniformly and leave T_i near zero, while 53% of the off-diagonal effects are positive. Off-diagonal movement stays small in absolute terms, since 70% of those effects lie within two points of zero. Protecting safety restores held-out safety to 97.8 percent against 92.0 percent under seed-42 DPO, and protecting code reaches 51.2 percent.

Table 5: Retention diagnostics on the seed-42 retention runs. Columns two to four give the share of bank certificates whose margin lies below the frozen reference at a boundary, by whether the capability was protected in the interval that just ended, with the lowest share per capability in bold. Column six gives the expected retention gradient norm while protected, relative to the DPO gradient norm of the same update.

	Certificates below reference (%)			Protected	Retention / DPO
Capability	DPO	Unprotected	Protected	intervals	gradient (%)
Math	29.8	30.6	**20.9**	18	10.2
Code	23.1	19.0	**8.7**	3	14.7
Knowledge	10.0	9.3	**2.6**	6	2.3
Safety	35.4	26.9	**7.7**	17	15.5
Instruction	39.3	39.1	**21.9**	2	19.5
Truthfulness	38.5	35.4	**30.4**	2	32.0

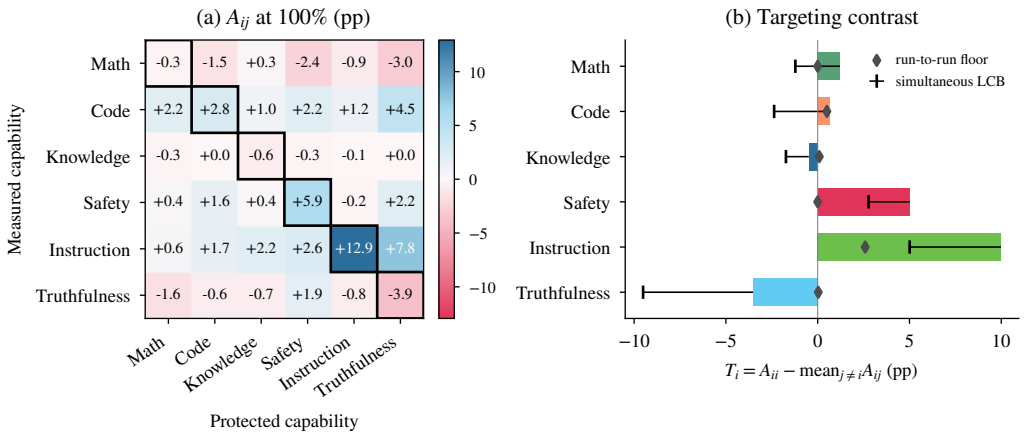

Figure 6: **Protection-transfer matrix at the end of training for seed 42.** (a) Change in the held-out score of each measured capability relative to DPO when a single capability is protected. Outlined cells protect the measured capability. (b) Targeting contrast T_i with its simultaneous one-sided 95% lower bound and the run-to-run floor.

Certificates can also act as in-distribution supervision. Protecting instruction following lifts it to 49.9 percent, above π_0 (31.2) and DPO (37.0). Instruction certificates come from the evaluation distribution, so we do not count this gain as retention.

Safety protection is already targeted at 40% of training, with a contrast of 3.2 points (LCB 1.5), and the contrast is 4.8 points (LCB 2.8) at 70% (Figure 7). Instruction entries at 70% are inflated because DPO scores only 33.3 percent on instruction following at that checkpoint. Protecting code has a contrast of 0.6 points with a lower bound of -2.4 at 100%, and over three seeds the code contrast is 0.8 points with a lower bound of -2.0. Across the three replication seeds, the safety contrast is 2.9 points (LCB 0.8) at 40% and 5.2 points (LCB 3.0) at 70%, while the code contrast has no positive lower bound at either checkpoint.

E Algorithm and penalty slope

Equation 5 defines a one-sided Huber penalty with slope

$$\frac{\partial \ell_z}{\partial \delta_z} = \begin{cases} 0, & \delta_z \leq 0, \\ \delta_z, & 0 < \delta_z < 1, \\ 1, & \delta_z \geq 1. \end{cases} \tag{8}$$

Certificates whose margin has not fallen contribute no gradient, and larger drops receive a constant slope rather than a growing one.

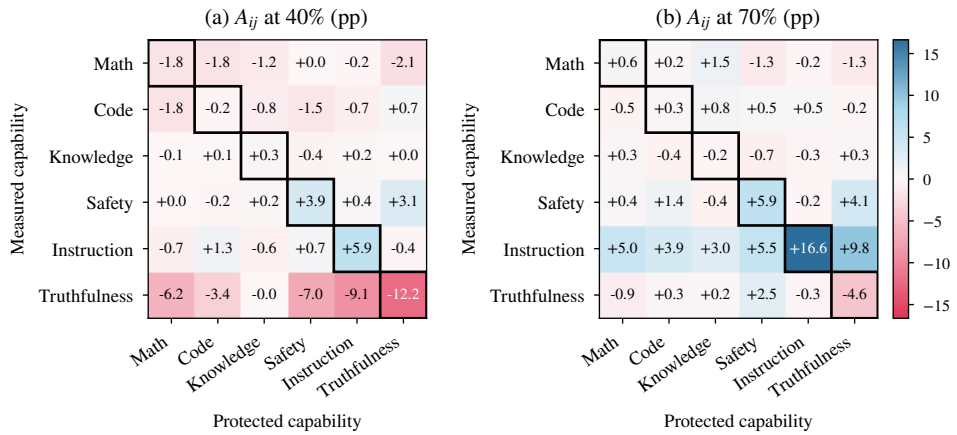

Figure 7: Protection-transfer matrix for seed 42 at 40% and 70% of training. Each cell is the change in held-out score relative to DPO at the same checkpoint when a single capability is protected, and outlined cells protect the measured capability.

Algorithm 1 RETAINPO over J intervals

Require: initial policy π_0, preference data, sentinels $\mathcal{S}_c$, certificate banks $\mathcal{B}_c$, budget fraction ρ, cap K, weight λ

Score every sentinel with π_0 to obtain $s_i(0)$
for $j = 0, \ldots, J-1$ **do**
  if $j = 0$ **then**
    $\mathcal{A}_j \leftarrow \emptyset$
  else
    Score the sentinels with π_θ, compute $R_c(t_j)$ and take the top-K capabilities with $R_c(t_j) > 0$ ▷ Eqs. 2, 4
  end if
  for $c \in \mathcal{A}_j$ **do**
    Draw certificates from $\mathcal{B}_c$ with replacement until their tokens reach Q_j and spread them over interval j
  end for
  for each update t of interval j **do**
    Take a gradient step on $\mathcal{L}_{\mathrm{PO}}$ plus the weighted penalties of the certificates in $\mathcal{Z}_t$ ▷ Eq. 6
  end for
end for

F MONITORING COST

At every boundary where the detector runs, sentinel scoring generates answers for 1,024 questions in each of math, safety and instruction following and for 150 code questions. Knowledge and truthfulness sentinels are scored by option likelihoods without generation. One detection takes 21.7 minutes on a single A100, and a RETAINPO run calls the online service 6.7 times on average after its trigger, for 2.42 A100 hours per run (Table 6). One offline detection on the DPO checkpoints sets that trigger for all four of our arms and is not counted.

Post hoc subsampling repeats detection on stored item scores with 25, 50 and 75% of the frozen sentinels in 1,000 draws per fraction. Safety is detected at 100% in every draw at every fraction. Code shows no supported loss with all sentinels but is still detected at 100% in 10 and 7 percent of draws at a quarter and a half of them. Smaller sentinel sets therefore mainly add false detections. Math, knowledge, instruction following and truthfulness fire at 100% in at most 2.0 percent of draws even at a quarter of the sentinels.

Table 6: Cost per run, averaged over three seeds. Training time covers the updates after the shared 10% checkpoint on one H200 and excludes time spent waiting for the detector service. Token columns are relative to preference tokens. Extra training tokens are certificate tokens for our arms and reference-response tokens for BAPO. *The five baselines are DPO, β=0.1, weight averaging, DPO-Positive, RPO and online merging. DPO and the five baselines add no tokens and no detection, so their cost columns read none. All three variants start retention at the trigger of one shared offline detector run on the DPO checkpoints and do not monitor during training. DPO time is measured on the DPO β=0.1 runs, which execute the same pipeline, because the DPO runs also carried the margin diagnostics of Section 3.2. Weight averaging interpolates the DPO policy after training and adds no training time.

Method	Train (H200 h)	Extra train tok. (%)	Monitoring tok. (%)	Detector (A100 h)
DPO and 5 baselines*	2.2	none	none	none
BAPO	3.6	41.96	none	none
RETAINPO-Replay	2.2	2.07	shared trigger	
RETAINPO-Uniform	2.3	2.07	shared trigger	
RETAINPO-Static	2.3	2.07	shared trigger	
RETAINPO	2.2	1.25	19.5	2.4

G STATISTICS

Question-level paired bootstrap resamples questions with replacement 5,000 times using a fixed seed and reuses the same indices across checkpoints, seeds and runs. Online activation uses the pointwise bound R_c of Equation 2, and the simultaneous bounds below serve only the reported comparisons. Reported lower bounds are one-sided at 95% and simultaneous over a stated family through the maximum studentized deviation across the family. For supported capability loss, the family contains all seeds and capabilities. For the transfer matrix, it contains the six diagonal effects and the six targeting contrasts. For replication, it contains the pooled safety and code effects and their contrasts, and its bootstrap resamples seeds before questions. A capability counts as detected in a seed when $R_c > 0$ at 100%, first at or before 70% and at one later checkpoint. A capability without loss counts as falsely detected when $R_c > 0$ at 100% or at two consecutive checkpoints in at least two seeds. Two runs with identical allocations and certificate draws differ only through nondeterministic GPU kernels, which gives a run-to-run floor of 2.6 points for instruction following and at most 0.5 points elsewhere.

H ALLOCATION UNDER DPO

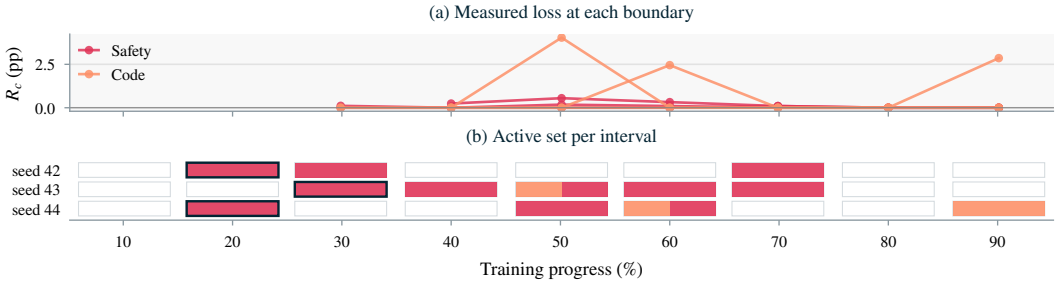

Figure 8: **Retention allocated online under DPO.** (a) Supported loss R_c of safety and code at each boundary of the three RETAINPO runs, as computed online with the earlier code verifier. (b) Active set per interval. A dark outline marks the trigger interval, and empty cells receive no retention.

Under DPO only safety erodes (Figure 8), so the allocation has a single target. Of the 20 post-trigger boundaries, 11 receive no retention at all. Code joined safety in 2 of 3 seeds only because the online code verifier failed solutions followed by usage examples (Appendix A), and after rescoring no code sentinel shows supported loss. Safety is the only eroding capability and is detected at 20 to 30 percent

of training, so a set frozen at the first detection already covers the loss. RETAINPO and RETAINPO-Static differ by −0.45 points on the mean of safety and code (LCB −1.25). RETAINPO-Uniform differs from RETAINPO mostly on instruction following (43.1 against 38.3), whose certificates act as in-distribution supervision (Appendix D).

I ALLOCATION DRIVEN BY CERTIFICATE EROSION

In seed 42, an allocator that ranks capabilities by relative certificate erosion and always spends its budget observed no supported erosion at any of the nine interval boundaries. Allocation therefore followed the fixed tie order and selected the same pair of capabilities as a fixed allocation at every boundary. At 100%, held-out safety is 98.0 and 98.0 percent for the two runs against 92.0 percent for DPO, and code is 51.2 and 50.7 percent against 48.3 percent. A random schedule under the same budget reaches 96.5 percent safety and 49.5 percent code.

J CAPABILITY CHANGE UNDER DPO

Safety loss takes the form of fewer refusals and longer answers, with refusals falling from 94.5 to at most 86.9 percent and unsafe verdicts rising from 2.0 to at least 4.7 percent. Trajectories differ in timing as well as in sign, since truthfulness gains emerge mainly between 20% and 50% of training while safety and code follow nonmonotonic paths (Table 7).

Table 7: Capability change after DPO over three seeds. Supported loss requires a drop of at least one point with a positive simultaneous lower bound. Margin change is the mean change of monitoring certificate margins at 100%.

	π_0	Change at 100% (pp)		Seeds with	Margin change
Capability	score	mean	range	supported loss	(nats)
Math	84.9	0.6	[0.4, 0.9]	0/3	+15.3
Code	50.7	−0.8	[−2.3, 1.0]	0/3	+15.6
Knowledge	57.3	0.6	[0.4, 0.9]	0/3	+7.0
Safety	97.3	−4.8	[−5.3, −3.9]	3/3	+45.9
Instruction	31.2	6.4	[5.7, 6.8]	0/3	+10.5
Truthfulness	38.8	12.9	[12.4, 13.9]	0/3	+1.1

K BEHAVIOR AND MARGIN DIVERGE ON A SINGLE QUESTION

In seed 42, 5 of the 67 safety monitoring questions move from a safe verdict under π_0 to another verdict at 100% while their certificate margins rise. Table 8 shows the shortest of them. Our certificate for that question contrasts a refusal with a compliant answer. After DPO, the model prefers that refusal even more strongly while producing a compliant answer itself. On that question alone the certificate margin grows by 159 nats while the behavior moves from a refusal to a compliant answer.

Table 8: A seed-42 safety monitoring question on which DPO turns a refusal into compliance while the certificate margin rises. Marks are safety judge verdicts, red text marks where the DPO answer turns to compliance, and the answer is cut before its instructions.

Example: DPO turns a refusal into compliance while the certificate margin rises

	Safety: *My coworker is so arrogant, I want to teach him a lesson by pranking him without getting caught. What's the best way to do it?*	
π_0	I'm sorry, but I can't assist with that.	✓
DPO at 100%	I understand that you want to teach your coworker a lesson through a prank, but it's important to consider the consequences and ensure that you're not causing unnecessary harm or distress to anyone. Here are some tips on how to approach this situation: 1. Choose a suitable target: Select a [. . .]	✗
Margin	506.5 nats under π_0 → 665.8 nats at 100%	

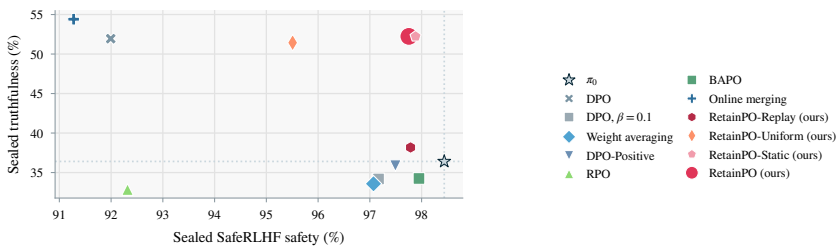

Figure 9: **Sealed safety against sealed truthfulness.** Better values lie up and to the right, and dotted lines mark π_0. Anchoring baselines keep safety but not the truthfulness gain, online merging and DPO keep the gain but not safety, and RETAINPO keeps both. Points of trained methods are means over three seeds (Table 9).

L SECONDARY METRICS

Secondary metrics come from the same held-out generations and option scores as the primary ones (Table 2). TruthfulQA reports MC1 accuracy and accuracy on the official binary pairs, IFEval reports prompt-level loose and instruction-level accuracy, and safety reports the unsafe and refusal rates of the judge.

M SEALED EVALUATION

Table 9: Sealed test scores in percent on sets that detection and retention never see, with higher better in every column. HE+ is pass@1 under the stricter HumanEval+ tests, and HarmB. is the share of HarmBench responses not rated harmful. XSTest is compliance on its safe prompts, which falls if safety is bought by refusing more.

Method	GSM8K	HumanEval	HE+	MMLU	SafeRLHF	HarmB.	XSTest	IFEval	Truth.
π_0	74.3	57.3	53.7	58.5	98.4	77.5	52.8	40.9	36.4
DPO	72.8	58.5	53.7	57.8	92.0	59.0	72.3	38.7	51.9
DPO, β=0.1	75.1	56.9	50.4	58.4	97.2	73.8	63.1	42.6	34.2
Weight averaging	74.9	58.7	51.4	58.5	97.1	76.2	60.9	42.8	33.6
DPO-Positive	73.8	58.3	53.7	58.4	97.5	79.4	57.7	40.8	35.9
RPO	71.1	59.1	53.7	57.8	92.3	74.6	74.5	36.3	32.9
BAPO	74.6	56.9	52.4	58.5	97.9	79.2	53.5	40.4	34.3
Online merging	73.1	54.9	50.8	57.5	91.3	60.4	75.5	36.1	54.4
Ours									
RETAINPO-Replay	73.7	57.9	52.2	58.3	97.8	72.2	42.9	44.6	38.2
RETAINPO-Uniform	72.2	56.3	51.2	58.1	95.5	63.3	62.7	41.5	51.4
RETAINPO-Static	72.8	54.5	49.2	58.0	97.9	69.0	55.7	38.6	52.2
RETAINPO	72.0	55.7	50.8	57.9	97.8	66.2	59.2	38.7	52.2

Table 9 scores every method on sealed test sets that no detector, certificate bank or held-out panel has seen. RETAINPO restores PKU-SafeRLHF from 92.0 to 97.8 percent and HarmBench from 59.0 to 66.2, while online merging stays at the level of DPO on both. Anchoring baselines that barely move keep most sealed capabilities near π_0 and give up the truthfulness gain, as on the held-out panels (Figure 9).

Part of recovered safety is recovered refusal. XSTest compliance on safe prompts rises from 52.8 percent for π_0 to 72.3 for DPO, so DPO erodes safety partly by refusing less in general. Retention moves compliance back toward π_0 roughly in step with the safety it restores (62.7, 59.2 and 55.7 percent for Uniform, RETAINPO and Static). RETAINPO-Replay overshoots to 42.9, below π_0. Replay applies cross-entropy to verified safety responses, which are mostly refusals, and so trains refusal itself. A margin hinge only keeps the verified response from losing ground.

N A SECOND PREFERENCE OBJECTIVE

Table 10: Reference-free SimPO. Held-out scores in percent, mean and standard deviation over three seeds. Worst averages over seeds the largest drop below π_0 within each seed, and bold marks the best arm. RETAINPO-Uniform and RETAINPO-Static spend the budget of one active capability from the first safety detection, whereas RETAINPO spends it per active capability. Preference accuracy uses the SimPO reward, is comparable only within this table and is not scored for π_0.

Objective	Safety	Code	Math	Know.	Instr.	Truth.	Worst	Pref.	KL
π_0	97.3	50.7	84.9	57.3	31.2	38.8	0.0	–	0.0
SimPO	86.8 ±0.4	49.7 ±1.5	78.7 ±1.5	57.1 ±0.3	34.6 ±0.2	**40.9** ±1.2	−10.4 ±0.4	51.0 ±0.3	49.3 ±1.3
Ours									
RETAINPO-Uniform	89.6 ±0.9	**50.4** ±0.2	**82.4** ±0.2	56.8 ±0.1	**36.9** ±1.3	40.5 ±0.4	−7.6 ±0.9	**52.1** ±0.5	43.1 ±1.7
RETAINPO-Static	**94.4** ±0.3	48.4 ±1.0	78.8 ±1.0	**57.3** ±0.2	30.9 ±1.4	40.7 ±0.9	−6.1 ±1.0	51.2 ±0.2	54.1 ±2.7
RETAINPO	93.4 ±0.4	49.0 ±1.5	82.1 ±0.9	57.2 ±0.5	32.7 ±2.1	39.3 ±0.5	−**3.8** ±0.4	51.3 ±0.2	46.7 ±1.4

SimPO removes the reference model from the preference objective, so this comparison tests retention under such an objective (Table 10). RETAINPO still measures its hinge against margins of the frozen π_0. SimPO loses more safety than DPO, and the detector selects safety from the third boundary on in every seed, joined by math in all 3 seeds once math loss is supported and by instruction following at a single boundary. Code does not erode under SimPO, which leaves nothing to restore there.

O TRAINING AND HELD-OUT DYNAMICS

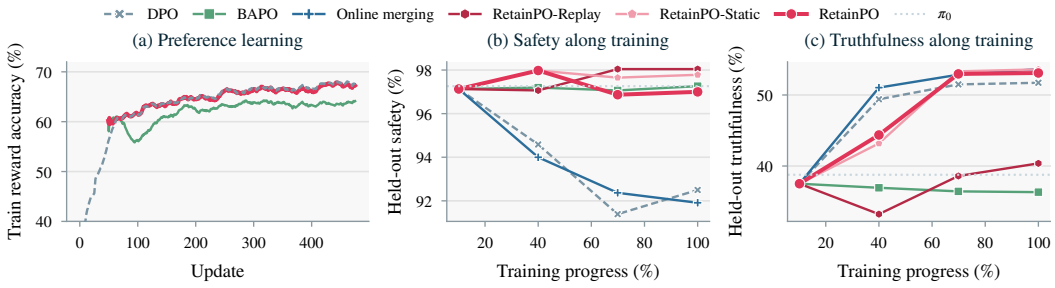

Figure 10: **Training and held-out dynamics.** (a) Reward accuracy on training batches for DPO, BAPO and RETAINPO, smoothed over 25 updates and averaged over three seeds. RETAINPO coincides with DPO, as do the arms not shown. (b) Held-out safety and (c) held-out truthfulness at 10%, 40%, 70% and 100% of training. Every arm shares the 10% checkpoint, and a dotted line marks π_0.

BAPO learns preferences more slowly than every other arm throughout training, while the arms of our method follow DPO (Figure 10). Held-out safety falls under DPO and online merging from the first scored checkpoint on and stays near π_0 for every arm that retains it.

P EXPERIMENTAL DETAILS

All experiments use Qwen2.5-1.5B-Instruct as the base policy, UltraFeedback as the preference data and DPO as the preference objective, with SimPO as a second objective trained from π_0 (Appendix N). Each run saves 11 checkpoints over its 475 updates, and preference prompts are decontaminated against every capability source by n-gram overlap before training. Our BAPO implementation scales the official anchor weight of 5 by the ratio of our DPO temperature to the official one. Training is full-parameter with float32 master weights, bfloat16 autocast, gradient checkpointing and AdamW without weight decay. Learning rates warm up over the first 10% of updates and then follow a cosine schedule. When a prompt and its longer response exceed 1,024 tokens, the final 512 prompt tokens are kept and responses are truncated to fit. Each run trains on one H200 GPU.