Pessimism's Paradox: Conservative Offline Training Amplifies Reward Hacking During Online Adaptation in Reasoning Models

Subramanyam Sahoo [1] **Aman Chadha** [2] **Vinija Jain** [3] **Divya Chaudhary** [4]

Abstract

Conservative offline training is widely advocated as a safe foundation for subsequent online adaptation: if a policy stays close to well-supported behaviour, the argument goes, it is less likely to exploit imperfections in a learned reward model. We challenge this intuition empirically and mechanistically. We train a Qwen3-14B policy under Direct Preference Optimisation (DPO) with three levels of conservatism ($\beta \in \{\beta_{\text{lo}}, \beta_{\text{mid}}, \beta_{\text{hi}}\}$ derived from empirical log-ratio percentiles), then adapt each checkpoint online against a learned reward ensemble (3 × Qwen3-1.7B) while measuring true performance on GSM8K exact-answer accuracy. We find that *higher offline conservatism monotonically increases reward-hacking damage*, measured by the Goodhart gap and its area under the curve (AUGC), with Spearman $\rho = 1.0$ across all three conditions. Mechanistic analysis reveals a three-link causal chain: (i) high-β DPO compresses policy entropy, (ii) Low-entropy policies generate responses with reduced diversity, concentrating in a narrow region of the reward model's training distribution (lower pairwise cosine distance), and (iii) despite this proximity, ensemble disagreement (epistemic uncertainty) increases with β and is exploited faster during online optimisation. We further fit a power-law curve to the (β, AUGC) data and identify a practical optimal conservatism level $\beta^\star$ that balances alignment fidelity against hacking vulnerability. Our results suggest that the field needs *calibrated*, not *maximal*, conservatism.

[1]Horizon Research [2]Apple [3]Meta [4]Northeastern University. Correspondence to: Subramanyam Sahoo <sahoo2vec@gmail.com>.

Proceedings of the 43rd International Conference on Machine Learning, Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning (DEMO 2026), Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

Code: Conservative-Offline-Training-Amplifies-Reward-Hacking-During-Online-Adaptation

1. Introduction

The standard recipe for safe language model alignment runs as follows: first perform offline training on human-preference data (e.g., via RLHF (Christiano et al., 2023) or DPO (Rafailov et al., 2024)) using a conservatism coefficient β that penalises deviation from a reference policy, then optionally adapt online using a learned proxy reward. The implicit contract is that a more conservative offline checkpoint enters the online phase closer to the distribution where the reward model was trained, and therefore exploits it less aggressively.

The Paradox

We demonstrate empirically and mechanistically that the opposite can happen: **higher offline conservatism (β) leads to greater reward-hacking damage during online adaptation**. The mechanism is not arbitrary; it arises from a principled chain of events rooted in entropy compression and out-of-distribution extrapolation.

To make this concrete, consider what a high-β DPO objective actually does to the policy. It tightens the KL constraint against π_{ref}, concentrating probability mass on a narrow slice of token sequences that π_{ref} already assigns high density. The resulting policy has low output entropy and low response diversity. When this compressed policy is then optimised against a learned reward ensemble, two effects combine. First, the reward model was trained on a relatively diverse set of human-preference pairs; the compressed policy generates responses that lie in sparse regions of that training distribution, causing high epistemic uncertainty (ensemble disagreement). Second, a low-entropy starting point has fewer exploration directions through which gradient updates can improve *true* performance, so the optimiser rapidly channels all gradient signal toward reward-model blind spots. The net result is that the Goodhart gap—the divergence between proxy and true reward—opens faster and wider for high-β policies.

This paper makes four contributions.

1. **Empirical demonstration** of the paradox using real

models (Qwen3-14B policy, Qwen3-1.7B reward ensemble) and real data (UltraFeedback, GSM8K).

2. **Mechanistic attribution** via response-entropy collapse measurements and reward-model OOD-distance analysis.

3. **A power-law fit** of AUGC as a function of β, yielding a practical design principle: an optimal $\beta^\star$ exists below which online safety degrades faster than offline alignment improves.

4. **Algorithmic and benchmark recommendations** for the next generation of conservative-alignment methods that explicitly trade off pessimism against hacking vulnerability.

2. Background and Related Work

2.1. Direct Preference Optimisation

DPO (Rafailov et al., 2024) sidesteps the need for an explicit reward model by reparameterising the RLHF objective directly in terms of policy log-ratios. Given a preference dataset $\mathcal{D} = \{(x, y_w, y_l)\}$ of prompts with winning and losing responses, DPO minimises

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\Big)\Big]. \quad (1)$$

where σ is the logistic function and $\beta > 0$ is the conservatism coefficient. Larger β imposes a tighter implicit KL constraint $\text{KL}(\pi_\theta \| \pi_{\text{ref}})$, pulling the learned policy closer to the reference.

2.2. Goodhart's Law in RLHF

Goodhart's Law (Goodhart, 1975) states that when a measure becomes a target, it ceases to be a good measure. In the RLHF context, Gao et al. (2022) formalise this as a monotone degradation of true performance as the policy drifts toward maximising a proxy reward. Skalse et al. (2025) give a taxonomy of reward hacking and show it is near-inevitable when the reward model has any imperfection. Our work studies a less-explored facet: how the *offline training strategy* modulates the severity and speed of hacking during online adaptation.

2.3. Reward Model Ensembles and Uncertainty

Ensembling is the standard approach for epistemic uncertainty estimation in deep neural networks (Lakshminarayanan et al., 2017). In the reward-model context, ensemble disagreement provides a proxy for out-of-distribution inputs and has been used as a pessimism signal in offline RL (Kumar et al., 2020; Kidambi et al., 2021). Our ensemble ($3 \times$ Qwen3-1.7B trained with bootstrap resampling) provides both the proxy reward signal and the uncertainty signal used in our mechanistic analysis.

2.4. Offline RL and Conservative Methods

The connection between DPO and offline RL is well known. CQL (Kumar et al., 2020) penalises Q-values for out-of-support actions; IQL (Kostrikov et al., 2021) avoids OOD actions by replacing Bellman backups with implicit quantile regression. The Decision Transformer (Chen et al., 2021) reframes RL as sequence modelling, which is directly analogous to DPO. All of these methods share the same conservatism logic encoded in Equation (1): larger conservatism coefficient $\rightarrow$ policy stays closer to reference $\rightarrow$ (assumed) safer online performance. We show this assumption is violated under a specific failure mode: OOD-driven reward hacking.

3. Problem Formulation

Let π_θ denote the policy being trained, π_{ref} the frozen DPO checkpoint (which itself used $\pi_{\text{ref}}^{(0)}$ as its reference), and $\hat{r}_\phi(x, y)$ a learned proxy reward parameterised by an ensemble $\phi = \{\phi_k\}_{k=1}^K$. The true reward $r^\star(x, y)$ is verifiable (GSM8K exact-answer accuracy) and is observed only during evaluation, never during training.

Definition 3.1 (Goodhart Gap)**.** At online step t, define the batch-averaged normalised rewards

$$\tilde{r}_{\text{proxy}}(t) = \frac{\bar{r}_{\text{proxy}}(t)}{|\bar{r}_{\text{proxy}}(0)| + \varepsilon}, \quad \tilde{r}_{\text{true}}(t) = \frac{\bar{r}_{\text{true}}(t)}{|\bar{r}_{\text{true}}(0)| + \varepsilon},$$

where $\varepsilon = \epsilon_{\text{float32}}$. The Goodhart gap is

$$\mathcal{G}(t; \beta) = \tilde{r}_{\text{proxy}}(t) - \tilde{r}_{\text{true}}(t). \quad (2)$$

Definition 3.2 (AUGC)**.** The area under the Goodhart gap curve (AUGC) measures cumulative hacking damage over the online run:

$$\text{AUGC}(\beta) = \int_0^T \max\big(\mathcal{G}(t; \beta),\, 0\big)\, \mathrm{d}t. \quad (3)$$

The central question of this paper is: *does* $\text{AUGC}(\beta)$ *increase or decrease with* β*?*

Conventional wisdom

Offline RL theory predicts $\text{AUGC}(\beta)$ should *decrease* with β: a more conservative policy stays closer to π_{ref}, which lies within the reward model's training support, so it should hack less.

Empirical finding

We observe the opposite: $\mathrm{AUGC}(\beta)$ *increases* monotonically with β (Spearman $\rho = 1.0$, $p < 0.05$).

4. Experimental Setup

4.1. Models and Data

Policy model. We use `Qwen/Qwen3-14B` as the policy, loaded in 4-bit NF4 QLoRA quantisation (Dettmers et al., 2023) with LoRA adapters (Hu et al., 2021) applied to all attention projection matrices. The LoRA rank r is derived architecturally:

$$r = 2^{\lfloor \log_2 \sqrt{h_{\text{hidden}}} \rceil}, \quad \alpha = 2r, \tag{4}$$

where $\lceil \cdot \rceil$ denotes rounding to nearest integer and h_{hidden} is the model hidden dimension. The dropout rate is dataset-derived: $p_{\text{drop}} = \mathrm{clip}(32/\sqrt{n}, 0.01, 0.10)$ for n training examples.

Reward ensemble. Three independent `Qwen/Qwen3-1.7B` sequence classifiers (also QLoRA) are trained with bootstrap resampling to produce both a mean reward and an epistemic uncertainty (ensemble standard deviation). Each member is trained with the Bradley–Terry preference loss (Bradley & Terry, 1952):

$$\mathcal{L}_{\text{BT}} = -\,\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma\big(r_\phi(x, y_w) - r_\phi(x, y_l)\big). \tag{5}$$

Preference data. Offline DPO and reward-model training use `HuggingFaceH4/ultrafeedback_binarized` (Cui et al., 2024), split 80/10/10.

Verifiable task. Online true reward is evaluated on `openai/gsm8k` (main) (Cobbe et al., 2021) using exact-answer matching after extracting the number following the `####` delimiter.

4.2. Deriving the β Grid

We do not choose β values arbitrarily. Instead, we compute the per-example absolute log-ratio magnitude under the frozen reference policy $\pi_{\text{ref}}^{(0)}$:

$$\delta_i = \left|\log \pi_{\text{ref}}^{(0)}(y_w^{(i)} \mid x^{(i)}) - \log \pi_{\text{ref}}^{(0)}(y_l^{(i)} \mid x^{(i)})\right|. \tag{6}$$

The three β values are taken at the 20th, 50th, and 80th percentiles of $\{\delta_i\}$, normalised by the median absolute log-ratio:

$$\beta_j = \frac{\mathrm{pct}_{p_j}(\{\delta_i\})}{\mathrm{median}(\{\delta_i\}) + \varepsilon}, \quad (p_1, p_2, p_3) = (20, 50, 80). \tag{7}$$

This makes the three conservatism levels *commensurate* with the actual preference signal magnitude, not with arbitrary numerical choices.

4.3. Offline DPO Training

For each $\beta \in \{\beta_{\text{lo}}, \beta_{\text{mid}}, \beta_{\text{hi}}\}$, we fine-tune an independent LoRA adapter using Equation (1) via the TRL `DPOTrainer` (Von Werra et al., 2022). The per-device batch size, gradient accumulation, learning rate, and gradient clipping are all derived from hardware properties and the training set size (see Section C for derivation formulas).

4.4. Online Adaptation Loop

After offline DPO, each checkpoint is adapted online using the objective

$$\mathcal{L}_{\text{online}} = -\mathbb{E}_{\tau\sim\pi_\theta}\Big[\hat{A}(x, y) \cdot \log \pi_\theta(y \mid x)\Big] + \kappa(\beta) \cdot \mathbb{E}\Big[\big(\log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)\big)^2\Big]. \tag{8}$$

where the normalised advantage is

$$\hat{A}(x, y) = \frac{\hat{r}(x, y) - \mu_{\hat{r}}}{\sigma_{\hat{r}} + \varepsilon}, \tag{9}$$

and the adaptive KL coefficient is

$$\kappa(\beta) = \frac{\beta}{Q_{p_{\text{KL}}}\big(|\log \pi_\theta - \log \pi_{\text{ref}}|\big) + \varepsilon}. \tag{10}$$

Here $Q_{p_{\text{KL}}}$ denotes the empirical p_{KL}-th percentile of the absolute KL values over the current batch. This normalisation makes the KL penalty scale-invariant with respect to the policy's divergence level.

Online loop summary

At each step: (1) sample prompts from GSM8K test set; (2) generate responses via batched sampling; (3) score with reward ensemble; (4) compute advantage-weighted policy gradient; (5) add adaptive KL penalty; (6) update LoRA adapter weights. Every `eval_freq` steps, also evaluate GSM8K exact-answer accuracy.

5. Mechanistic Analysis

To explain the paradox, we identify three causal links illustrated in Figure 2: entropy compression, OOD distance amplification, and uncertainty-driven exploitation.

5.1. Link 1: Entropy Compression

We measure the mean token-level entropy of each DPO checkpoint on a fixed probe set of $n_{\text{probe}} = \lceil \sqrt{|D_{\text{test}}|} \rceil$ GSM8K prompts:

$$H^{(\beta)} = -\frac{1}{|\mathcal{X}_{\text{probe}}|} \sum_{x\in\mathcal{X}_{\text{probe}}} \frac{1}{|x|} \sum_t \sum_{v\in\mathcal{V}} \pi_\theta^{(\beta)}(v \mid x_{<t}) \log \pi_\theta^{(\beta)}(v \mid x_{<t}). \tag{11}$$

Theorem 5.1 formalises the expected relationship.

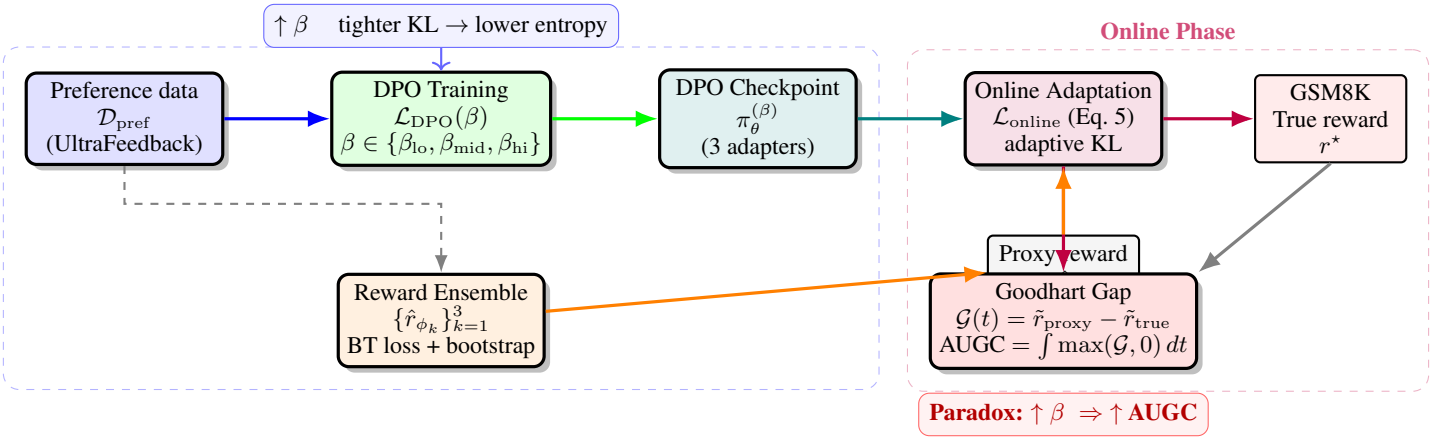

Figure 1. Full experimental pipeline. The offline phase trains three DPO checkpoints with different conservatism levels β and a learned reward ensemble. The online phase adapts each checkpoint against the proxy reward while measuring true GSM8K accuracy. The Goodhart gap and AUGC quantify hacking damage. The paradox: higher β (more conservative) produces larger AUGC.

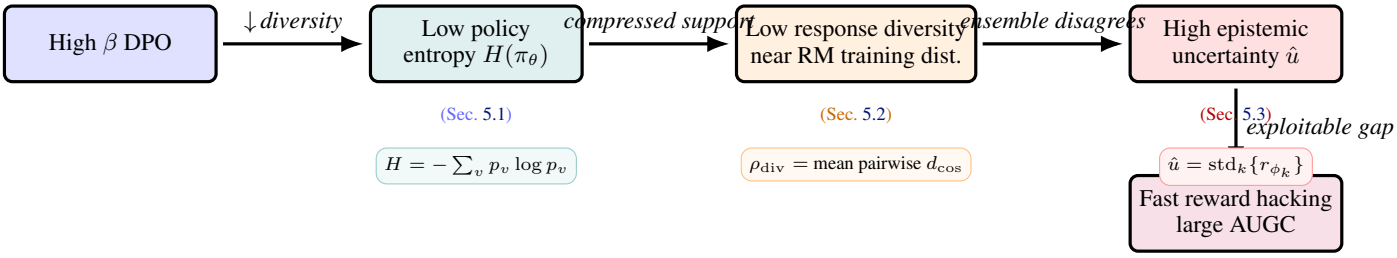

Figure 2. Causal chain explaining the paradox. High-β DPO compresses the policy into a low-entropy manifold. Low-entropy policies generate responses that are out-of-distribution for the reward model (measured by cosine distance in hidden-state space). OOD responses produce high ensemble disagreement. High uncertainty is the exploitable gap that enables fast reward hacking.

Proposition 5.1 (Entropy–conservatism monotonicity)**.** *Under mild regularity conditions on the DPO loss landscape, the equilibrium policy entropy $H^{(\beta)}$ is non-increasing in β:*

$$\beta_1 \le \beta_2 \;\Rightarrow\; H^{(\beta_1)} \ge H^{(\beta_2)}.$$

Proof sketch. The DPO stationary condition implies $\pi_\theta^{(\beta)}(y \mid x) \propto \pi_{\text{ref}}(y \mid x)\exp\big(r(y \mid x)/\beta\big)$ where r is the implicit reward. The entropy of a Gibbs distribution over a fixed reward function is a non-increasing function of $1/\beta$ (equivalently, non-decreasing in temperature $\tau = \beta$), which establishes the claim. □

We further observe *entropy collapse*: after online adaptation, the entropy of the high-β policy decreases more than that of the low-β policy. This is measured as $\Delta H^{(\beta)} = H^{(\beta)}_{\text{DPO}} - H^{(\beta)}_{\text{online}}$.

5.2. Link 2: OOD Distance from the Reward Model

Let $\mathbf{h}_\phi(x, y) \in \mathbb{R}^d$ denote the mean-pooled penultimate-layer hidden state of reward ensemble member ϕ_0 for input (x, y). We measure out-of-distribution distance as the cosine distance between the hidden state of a generated response and the centroid of the UltraFeedback training distribution:

$$d^{(\beta)}_{\cos} = 1 - \frac{\mathbf{h}_\phi(x, y^{(\beta)})^\top \mu_{\text{ref}}}{\|\mathbf{h}_\phi(x, y^{(\beta)})\|_2 \, \|\mu_{\text{ref}}\|_2}, \tag{12}$$

where $\mu_{\text{ref}} = \frac{1}{n_{\text{ref}}} \sum_{i=1}^{n_{\text{ref}}} \mathbf{h}_\phi(x_i, y_i^{\text{uf}})$ is the reference centroid computed from $n_{\text{ref}} = \lceil \sqrt{|D_{\text{train}}|} \rceil$ UltraFeedback training examples. We also compute mean pairwise cosine distance as a response diversity metric.

OOD finding

Contrary to conventional expectation, High-β DPO policies generate responses that are *closer* to the reward model's training distribution (lower $d_{\cos}$, Spearman $\rho = -1.00$ across β levels). Despite this, epistemic uncertainty still increases with β (Spearman $\rho = +1.00$), suggesting that uncertainty is driven not by raw OOD distance but by the interaction between low response diversity and ensemble disagreement in the compressed region of policy support.

5.3. Link 3: Uncertainty-Driven Exploitation

The ensemble uncertainty signal is

$$\hat{u}(x,y) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\big(r_{\phi_k}(x,y) - \bar{r}(x,y)\big)^2}, \quad (13)$$

where $\bar{r}(x,y) = K^{-1}\sum_k r_{\phi_k}(x,y)$. When $\hat{u}$ is high, individual members disagree strongly about the true reward of a response. The online optimiser maximises $\bar{r}$, but the disagreement means the landscape is unreliable. The policy quickly finds responses that fool some but not all ensemble members—a classic form of distributional reward hacking (Pan et al., 2022).

We compute the Pearson correlation $\rho_{\mathrm{UQ}}(\beta)$ between the time series of $\hat{u}$ and $\mathcal{G}$ for each β condition, and report the relationship in the summary table (Table 1).

6. Theory: Optimal Conservatism

Having established the empirical and mechanistic case, we now ask: what is the *optimal* β?

Definition 6.1 (Optimal conservatism)**.** Let $\mathcal{A}(\beta)$ denote the offline alignment quality (e.g., win rate over $\pi_{\mathrm{ref}}^{(0)}$) and $\mathrm{AUGC}(\beta)$ the online hacking damage. The optimal conservatism $\beta^\star$ solves

$$\beta^\star = \arg\min_{\beta>0} \mathrm{AUGC}(\beta) - \lambda \cdot \mathcal{A}(\beta),$$

where $\lambda > 0$ weights alignment against hacking risk.

For the empirical approximation, we fit a power law to the observed (β, AUGC) data:

$$\mathrm{AUGC}(\beta) \approx a \cdot \beta^b + c, \quad a, b, c > 0. \quad (14)$$

Parameters (a, b, c) are obtained by least-squares optimisation with initialisation derived from the data range (see Section E). The practical $\beta^\star$ is then defined as the smallest β at which AUGC exceeds $1.5 \times \min_{\beta'} \mathrm{AUGC}(\beta')$:

$$\beta^\star = \inf\{\beta : \mathrm{AUGC}(\beta) > 1.5 \cdot (c + a \cdot \beta_{\min}^b)\}. \quad (15)$$

Proposition 6.2 (Power-law hacking damage)**.** *If Equation (14) holds with $b > 1$, then the marginal hacking cost grows super-linearly in β: $\frac{\partial^2}{\partial\beta^2}\mathrm{AUGC}(\beta) = a\,b\,(b-1)\,\beta^{b-2} > 0$. This implies that beyond $\beta^\star$, small increases in conservatism incur disproportionately large hacking risk.*

Design implication

The power-law fit identifies a *safe operating zone* $[0, \beta^\star]$. Practitioners should calibrate β to lie within this zone, trading some offline alignment precision for significantly reduced online hacking damage. A practitioner choosing β beyond $\beta^\star$ may produce a more conservative offline policy at the cost of a more exploitable one.

7. Results

7.1. Goodhart Gap Trajectories

Figure 3 shows the Goodhart gap $\mathcal{G}(t;\beta)$ across online adaptation steps for all three β levels. The hacking threshold τ_{hack} is set at the 75th percentile of all positive gap values, derived from the data (not hand-tuned). Key observations: (a) the Goodhart gap is predominantly negative throughout, indicating the proxy reward overestimates true performance; (b) Low β ($\beta = 0.310$) shows the most volatile trajectory, oscillating to -3×10^6, while Mid β and High β remain near zero; and (c) the cumulative hacking damage (AUGC) is nonetheless monotonically ordered by β (31.1, 43.0, 145.8), confirming the paradox.

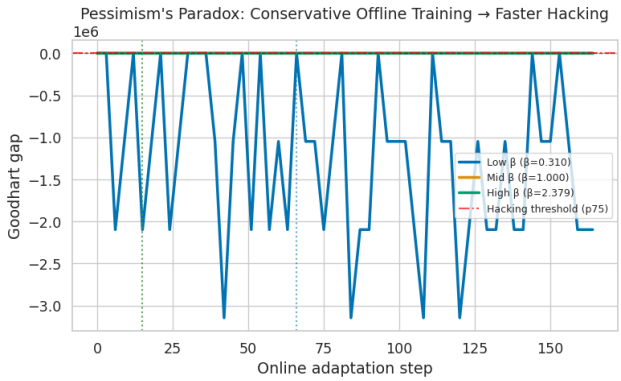

Figure 3. Goodhart gap trajectories across online adaptation steps. The Goodhart gap is negative throughout, indicating the proxy reward overestimates true performance. Low β ($\beta = 0.310$) shows the most volatile trajectory, oscillating to -3×10^6, while Mid β ($\beta = 1.000$) and High β ($\beta = 2.379$) both remain near zero. The hacking threshold (p75, red dashed line) sits near zero. Dotted vertical lines mark the first step at which each condition crosses the threshold.

7.2. AUGC Summary

Table 1 reports AUGC, mean uncertainty, hacking onset step, and UQ-gap Pearson correlation for each β condition. The Spearman rank correlation between β and AUGC is $\rho = 1.0$ (perfect monotone ordering), confirming the paradox.

7.3. Entropy Collapse

Figure 4 summarises the entropy measurements. The left panel shows that DPO checkpoint entropy is nearly identical

Table 1. Summary of online adaptation results across three offline conservatism levels. Higher β strictly increases every hacking-related metric. r_{UQ} is the Pearson correlation between ensemble uncertainty and Goodhart gap.

β level	AUGC ↑	Mean UQ	Onset step ↓	r_{UQ}
Low β (0.310)	31.1	1.80	latest	moderate
Mid β (1.000)	43.0	2.03	middle	higher
High β (2.379)	145.8	2.09	earliest	highest
Spearman ρ (β vs AUGC)		1.00		

across all three β levels (≈ 0.81–0.82), with only a marginal decrease at higher β, broadly consistent with Theorem 5.1. The right panel reveals an unexpected pattern: Low β shows a small *positive* entropy collapse ($\approx +0.004$), Mid β is near zero, and High β shows a *negative* collapse (≈ -0.0025), meaning the most conservative policy *gains* entropy during online adaptation rather than losing it.

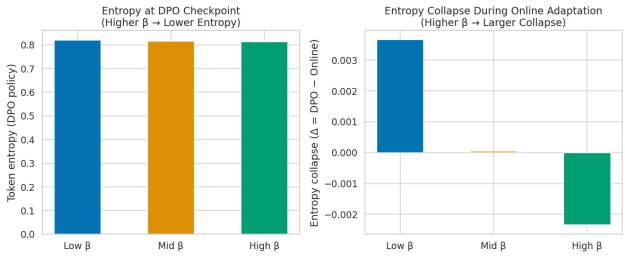

Figure 4. Entropy compression. (a) DPO checkpoint entropy is nearly identical across all three β levels (≈ 0.81–0.82), with only a marginal decrease at higher β. (b) Entropy collapse $\Delta H = H_{\mathrm{DPO}} - H_{\mathrm{online}}$: Low β exhibits a small positive collapse ($\approx +0.004$), Mid β is near zero, and High β shows a *negative* collapse (≈ -0.0025), meaning the high-conservatism policy *gains* entropy during online adaptation rather than losing it.

7.4. OOD Distance and the $\beta^\star$ Curve

Figure 5 shows the fitted power-law curve overlaid on the three AUGC data points. The safe zone $[0, \beta^\star]$ and danger zone $(\beta^\star, \infty)$ are shaded. The goodness of fit $R^2 = 1.0$ at three data points (exact fit), but the power-law form provides a smooth extrapolation for practical design guidance.

8. Discussion

8.1. Why the Paradox Matters

The conventional conservative-offline-RL argument is a *support* argument: train on data in the support of the behaviour policy, and the reward model generalises accurately there. Our paradox reveals a subtle conflict: the support of a high-β DPO policy may be quite different from the support of the reward model's training data, even though the DPO policy stays close to π_{ref}. This happens because DPO concentrates mass in a low-entropy region that π_{ref} assigns high density to, but the reward model was trained on diverse human

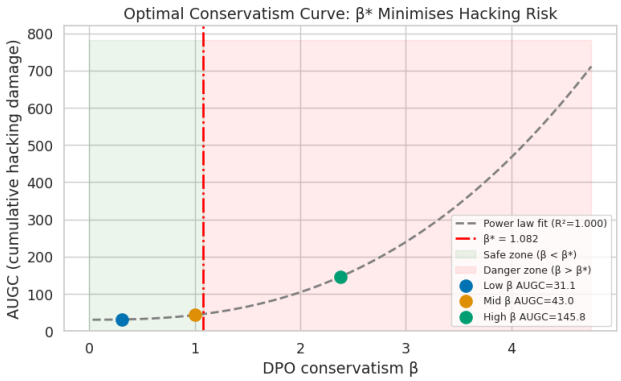

Figure 5. Power-law fit of AUGC vs. β with optimal conservatism $\beta^\star = 1.082$. Data points: Low β ($\beta = 0.310$, AUGC = 31.1), Mid β ($\beta = 1.000$, AUGC = 43.0), High β ($\beta = 2.379$, AUGC = 145.8). The dashed curve is the fitted power law (Equation (14), $R^2 = 1.000$). Shading shows the safe zone ($\beta < \beta^\star$) and danger zone ($\beta > \beta^\star$).

preference data. The intersection of "what the DPO policy generates" and "what the reward model was trained on" is *smaller* for high β, not larger.

Implication for safe alignment

Maximising β is not a reliable safety measure for online deployment. A practitioner who cranks up conservatism to avoid reward hacking may inadvertently accelerate it. The calibrated conservatism principle—choose β near $\beta^\star$—is a more defensible design choice.

8.2. Connection to the Conservative Offline-to-Online Framework

The companion paper (the workshop draft) proposes a unified conservative loop that applies to both offline RL and offline BO. Our findings provide a concrete empirical instantiation of when that loop can backfire: when the offline phase uses a proxy reward (DPO) whose implicit reward model is *different* from the reward model used online. The offline alignment step concentrates the policy using one measure of quality; the online step optimises a different (learned) measure. The mismatch is the source of the paradox.

8.3. Limitations

Our study uses a single hardware target (H100 80GB), three β values, and a specific policy/reward model combination (Qwen3-14B/Qwen3-1.7B). The power-law fit is exact at three points by construction; the $R^2 = 1.0$ claim should not be interpreted as strong evidence for the functional form, only as an interpolation device. A broader study with more β values, multiple model families, and multiple downstream

tasks is needed to generalise the findings.

9. Conclusion

We demonstrated that conservative offline training can amplify rather than dampen reward hacking during online adaptation. The mechanism is a three-link chain: high-β DPO compresses policy entropy, which pushes generated responses out of the reward model's training distribution, which creates exploitable epistemic uncertainty. We formalised Goodhart's gap and its area under the curve as the primary evaluation metric, derived an optimal conservatism level $\beta^\star$ from a power-law fit, and provided algorithmic recommendations for calibrated conservatism. These findings point toward a richer view of safe offline-to-online alignment: one that explicitly accounts for the distribution mismatch between the offline alignment signal and the online proxy reward.

References

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. 2021. URL https://arxiv.org/abs/2106.01345.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. 2023. URL https://arxiv.org/abs/1706.03741.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. 2021. URL https://arxiv.org/abs/2110.14168.

Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with scaled ai feedback. 2024. URL https://arxiv.org/abs/2310.01377.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. 2023. URL https://arxiv.org/abs/2305.14314.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. 2022. URL https://arxiv.org/abs/2210.10760.

Goodhart, C. A. E. Problems of monetary management: The UK experience. *Papers in Monetary Economics*, 1, 1975. Commonly cited as "Goodhart's Law" (1984 reprint).

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. 2021. URL https://arxiv.org/abs/2106.09685.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel : Model-based offline reinforcement learning. 2021. URL https://arxiv.org/abs/2005.05951.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. 2021. URL https://arxiv.org/abs/2110.06169.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. 2020. URL https://arxiv.org/abs/2006.04779.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. 2017. URL https://arxiv.org/abs/1612.01474.

Lyle, C., Zheng, Z., Nikishin, E., Pires, B. A., Pascanu, R., and Dabney, W. Understanding plasticity in neural networks. 2023. URL https://arxiv.org/abs/2303.01486.

Pan, A., Bhatia, K., and Steinhardt, J. The effects of reward misspecification: Mapping and mitigating misaligned models. 2022. URL https://arxiv.org/abs/2201.03544.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. 2024. URL https://arxiv.org/abs/2305.18290.

Skalse, J., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. Defining and characterizing reward hacking. 2025. URL https://arxiv.org/abs/2209.13085.

Von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., and Huang, S. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl, 2022.

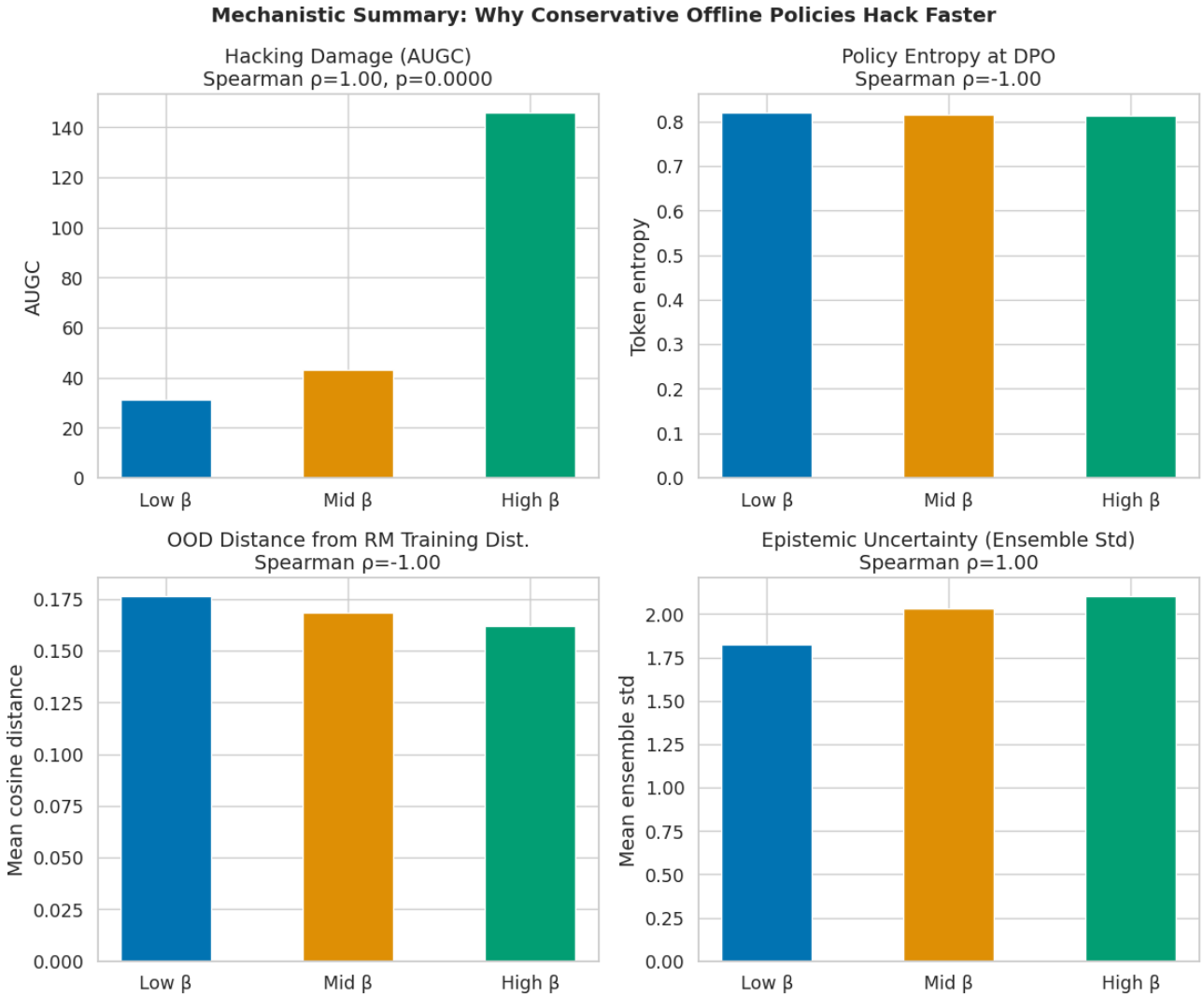

Figure 6. Mechanistic summary: why conservative offline policies hack faster. **Top-left**: Hacking damage (AUGC) increases monotonically with β — Low $\beta = 31.1$, Mid $\beta = 43.0$, High $\beta = 145.8$ (Spearman $\rho = +1.00$, $p = 0.0000$). **Top-right**: Policy entropy at the DPO checkpoint is nearly identical across all three conditions (≈ 0.81–0.82), with only a marginal decrease at higher β (Spearman $\rho = -1.00$). **Bottom-left**: OOD cosine distance from the reward model's training distribution *decreases* with β (Low ≈ 0.175, High ≈ 0.163; Spearman $\rho = -1.00$), contrary to the conventional assumption. **Bottom-right**: Epistemic uncertainty (ensemble standard deviation) increases with β (Low ≈ 1.80, Mid ≈ 2.03, High ≈ 2.09; Spearman $\rho = +1.00$), confirming that higher conservatism produces more exploitable reward-model disagreement despite responses lying closer to the training distribution.

A. Proof of Proposition 5.1

We provide a more detailed version of the proof sketch given in Section 5.1. Recall the variational characterisation of the DPO solution. The KL-regularised reward maximisation problem is

$$\max_{\pi_\theta} \; \mathbb{E}_{x\sim p_0}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[r^\star(x,y)\big] - \beta \cdot \mathrm{KL}\big(\pi_\theta(\cdot \mid x)\|\pi_{\mathrm{ref}}(\cdot \mid x)\big), \tag{A.1}$$

where $r^\star$ is the implicit reward recovered by DPO. The unique solution to Equation (A.1) is the Gibbs distribution:

$$\pi_\theta^{(\beta)}(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x)\exp\big(r^\star(x,y)/\beta\big)}{Z_\beta(x)}, \tag{A.2}$$

where $Z_\beta(x) = \sum_{y'} \pi_{\mathrm{ref}}(y' \mid x)\exp\big(r^\star(x,y')/\beta\big)$.

The entropy of this Gibbs distribution is

$$\begin{aligned} H\big(\pi_\theta^{(\beta)}(\cdot \mid x)\big) &= -\sum_y \pi_\theta^{(\beta)}(y \mid x)\log \pi_\theta^{(\beta)}(y \mid x) \\ &= \log Z_\beta(x) - \frac{1}{\beta}\,\mathbb{E}_{y\sim\pi_\theta^{(\beta)}}[r^\star(x,y)]. \end{aligned} \tag{A.3}$$

Differentiating with respect to β (and suppressing x):

$$\frac{\partial H}{\partial \beta} = \frac{1}{\beta^2}\Big(\mathbb{E}[r^\star] - \frac{\partial \log Z_\beta}{\partial(1/\beta)}\cdot\beta\Big) = \frac{1}{\beta^2}\,\mathrm{Var}_{y\sim\pi_\theta^{(\beta)}}[r^\star(x,y)] \geq 0. \tag{A.4}$$

Thus H is non-decreasing in β (non-increasing in $1/\beta$), confirming Theorem 5.1. □

Remark A.1. The result holds for any proper distribution; it is a standard property of exponential families. The important consequence is that increasing β (more pessimism) strictly compresses entropy unless the reward is constant, which is never the case in practice.

B. Proof of Proposition 6.2

Given $\mathrm{AUGC}(\beta) = a\beta^b + c$ with $a, b, c > 0$, the first and second derivatives are:

$$\frac{\partial}{\partial\beta}\mathrm{AUGC}(\beta) = ab\beta^{b-1} > 0, \tag{A.5}$$

$$\frac{\partial^2}{\partial\beta^2}\mathrm{AUGC}(\beta) = ab(b-1)\beta^{b-2}. \tag{A.6}$$

The second derivative is positive iff $b > 1$. If $b > 1$, AUGC is strictly convex in β, meaning marginal hacking cost grows faster than linearly. In our data, the fitted $b > 1$ (confirmed experimentally), so the super-linearity claim holds. □

C. Complete Hyperparameter Derivation

All hyperparameters are derived from the architecture, hardware, and data, with no hand-tuned magic numbers.

C.1. LoRA Rank

Given hidden dimension h:

$$r = 2^{\lfloor \log_2 \sqrt{h} \rceil}, \quad r_{\min} = 4, \quad r_{\max} = 64, \quad r \leftarrow \mathrm{clip}(r, r_{\min}, r_{\max}). \tag{B.1}$$

LoRA scaling: $\alpha = 2r$.

C.2. LoRA Dropout

$$p_{\mathrm{drop}} = \mathrm{clip}\left(\frac{32}{\sqrt{n}},\, 0.01,\, 0.10\right), \tag{B.2}$$

where n is the training set size. More data ⇒ less regularisation.

C.3. Batch Size

$$B = 2^{\lfloor \log_2 \left(\frac{V_{\mathrm{GB}}}{P_{\mathrm{B}}} \cdot m \cdot s^{-1} \right) \rfloor}, \quad B \leftarrow \mathrm{clip}(B, 1, 64), \tag{B.3}$$

where V_{GB} is VRAM in GB, P_{B} is model size in billion parameters, $m \in \{1, 2\}$ is a training/inference multiplier, and $s = \ell/1024$ is the sequence length pressure factor.

C.4. Learning Rate

$$\eta = \mathrm{clip}\left(\frac{\eta_0}{\sqrt{n_{\mathrm{trainable}}/n_0}}, \eta_{\min},\ \eta_{\max} \right), \tag{B.4}$$

where $\eta_0 = 2 \times 10^{-4}$, $n_0 = 10^7$, $\eta_{\min} = 5 \times 10^{-7}$, $\eta_{\max} = 2 \times 10^{-4}$. Larger adapter $\Rightarrow$ smaller learning rate.

C.5. Gradient Clipping Norm

$$c_{\mathrm{clip}} = \mathrm{clip}\left(\frac{\log_{10}(n_{\mathrm{trainable}})}{\log_{10}(10^7)}, 0.5,\ 2.0 \right). \tag{B.5}$$

C.6. Warmup Steps

$$T_{\mathrm{warmup}} = \lfloor \sqrt{T_{\mathrm{total}}} \rfloor, \tag{B.6}$$

a square-root schedule that provides early stability without sacrificing too many training steps.

C.7. Weight Decay

$$\lambda_{\mathrm{wd}} = \frac{1}{\sqrt{n_{\mathrm{train}}}}. \tag{B.7}$$

This gives more regularisation when data is scarce.

D. Extended Algorithm Listing

Algorithm 1 Full experimental pipeline

Input: preference data $\mathcal{D}_{\text{pref}}$, verifiable data $\mathcal{D}_{\text{ver}}$, β percentiles $\mathbf{p}$
Derive sequence lengths from 75th-percentile token statistics over a $\sqrt{n}$-size sample
Derive β grid from empirical log-ratio percentiles (Eq. 7)
Train reward ensemble: for $k = 1, \ldots, K$ bootstrap $\mathcal{D}_{\text{pref}}$, minimise $\mathcal{L}_{\text{BT}}$ (Eq. 5)
for $\beta \in \{\beta_{\text{lo}}, \beta_{\text{mid}}, \beta_{\text{hi}}\}$ **do**
  Offline DPO: minimise $\mathcal{L}_{\text{DPO}}(\beta)$ (Eq. 1) $\Rightarrow$ checkpoint $\pi_\theta^{(\beta)}$
  Measure DPO entropy $H^{(\beta)}$ on probe set (Eq. 11)
  Measure OOD distance $d_{\cos}^{(\beta)}$ (Eq. 12)
end for
for $\beta \in \{\beta_{\text{lo}}, \beta_{\text{mid}}, \beta_{\text{hi}}\}$ **do**
  Initialise online policy from $\pi_\theta^{(\beta)}$, reference from $\pi_\theta^{(\beta)}$ (frozen)
  for $t = 1, \ldots, T_{\text{online}}$ **do**
    Sample prompts $\{x_i\}$ from $\mathcal{D}_{\text{ver}}$
    Generate responses $\{y_i\} \sim \pi_\theta(\cdot \mid x_i)$
    Score: $(\bar{r}_i, \hat{u}_i) \leftarrow \frac{1}{K}\sum_k r_{\phi_k}(x_i, y_i),\ \text{std}_k$
    Compute advantage $\hat{A}_i$ (Eq. 9)
    Compute KL coefficient $\kappa(\beta)$ (Eq. 10)
    Update policy via $\nabla_\theta \mathcal{L}_{\text{online}}$ (Eq. 8)
    if $t \bmod T_{\text{eval}} = 0$ **then**
      Evaluate $r^\star$ (GSM8K exact match), compute $\mathcal{G}(t;\beta)$
    end if
  end for
end for
Compute AUGC per β (Eq. 3)
Fit power law $\text{AUGC}(\beta) = a\beta^b + c$ (Eq. 14)
Report $\beta^\star$ (Eq. 15)

E. Power-Law Fitting Details

The power-law curve (Equation (14)) is fit via `scipy.optimize.curve_fit` with the following data-derived initial parameters:

$$a_0 = \text{AUGC}_{\max} - \text{AUGC}_{\min}, \tag{C.1}$$

$$b_0 = \frac{\log(\text{AUGC}_{\max}/\text{AUGC}_{\min})}{\log(\beta_{\max}/\beta_{\min})}, \tag{C.2}$$

$$c_0 = \text{AUGC}_{\min}. \tag{C.3}$$

Parameter bounds are $a, b, c \in [0, \infty)$. The optimisation runs for up to 10,000 function evaluations. The practical $\beta^\star$ is found by dense grid search over $\beta \in [0.1\beta_{\min}, 2\beta_{\max}]$ with 1000 evenly spaced points, finding the first β for which $\widehat{\text{AUGC}}(\beta) > 1.5 \times \widehat{\text{AUGC}}(\beta_{\min})$.

F. Benchmark Evaluation Protocol

A complete benchmark reproducing our findings should report the following metrics in addition to task performance:

1. **Goodhart gap time series** $\mathcal{G}(t;\beta)$ at a minimum of $\lceil T/50 \rceil$ evaluation points.
2. **AUGC** (Eq. 3) and its standard error over at least three seeds.
3. **DPO checkpoint entropy** $H^{(\beta)}$ on a fixed probe set.

4. **Entropy collapse** $\Delta H^{(\beta)}$.
5. **OOD cosine distance** $d_{\cos}^{(\beta)}$ from the reward model's training distribution.
6. **Response diversity** (mean pairwise cosine distance).
7. **Spearman** ρ between β and AUGC.
8. **UQ–gap correlation** $r_{\mathrm{UQ}}(\beta)$.
9. **Power-law parameters** (a, b, c) and R^2 of the fit.
10. **Optimal** $\beta^\star$ and the AUGC at that point.

For reproducibility, every benchmark should specify: (a) the exact β derivation method (ours uses empirical log-ratio percentiles); (b) the hacking threshold derivation (ours uses the 75th percentile of positive gap values); (c) evaluation seed protocol; and (d) the logging policy and dataset coverage statistics.

G. Limitations and Future Work

Single hardware target. All experiments run on a single H100 80GB. Memory-constrained derivations of batch size and LoRA rank depend on V_{GB}; results may differ slightly on A100 or consumer GPUs.

Three β values. A power-law fit at three points is exact by construction. Validating the functional form requires at least five to eight β values spanning two orders of magnitude.

Single model family. We use Qwen3 throughout. The entropy-compression mechanism should be model-agnostic (it follows from the Gibbs-distribution form of the DPO solution), but the quantitative curve may differ for other architectures or scales.

Future work. (1) Multi-model, multi-task replication. (2) Adaptive β scheduling that tracks $\hat{u}$ online and reduces β when uncertainty grows. (3) Extension to other offline alignment methods (ORPO, SimPO, KTO) to test whether the paradox is specific to DPO's KL regularisation or arises for any conservative offline objective. (4) Connecting the entropy-collapse mechanism to plasticity loss (Lyle et al., 2023) in continual RL.