Follow the Winners: Conservative Policy Improvement with the Cross-Entropy Method for Critic-Free RFT

Joery A. de Vries[1] **Neil D. Lawrence**[1,2] **Zhenwen Dai**[1]

[1]Trent AI Limited, London, United Kingdom
[2]Computer Laboratory, University of Cambridge, Cambridge, United Kingdom

{joery,neil,zhenwen}@trent.ai ndl21@cam.ac.uk

Abstract

Critic-free reinforcement fine-tuning (RFT) for agentic large language models is often done through GRPO or its derivatives, which compute a group baseline over repeated rollouts per prompt to reduce variance yet update relatively aggressively on these high-variance samples. We propose *Follow the Winners* (FTW), a critic-free policy-learning algorithm that adapts the cross-entropy method to RFT, replacing group rollouts with an ordinal filter on replay-buffer samples that yields polynomial concentration in the order statistic of returns. We derive FTW through a control-as-inference lens, which also recovers GRPO and DPO as specific modelling choices, identifying GRPO as risk-neutral while DPO and FTW share a bounded risk-seeking offset that FTW controls. We identify this offset as an inherent trade-off of variance reduction through ordinal filters on samples, whereas a critic model induces a different trade-off between bias and variance. Scaled to agentic LLM post-training, FTW matches GRPO and PPO on Sokoban and Search-R1 baselines, showing a viable trade-off from a value model or group rollouts to CPU memory.

1 Introduction

Reinforcement learning fine-tuning (RFT) of large language models (LLMs) for agentic tasks learns from trajectory returns that often exhibit high variance even under deterministic rewards due to long horizons, partial observability, and state aliasing (Kaelbling et al., 1998; Spaan, 2012; Ouyang et al., 2022; Yao et al., 2023). Policy-gradient methods such as PPO (Schulman et al., 2017) partly reduce variance with a learned critic, at the cost of GPU memory for a second model and possible bias depending on how we bootstrap (Schulman et al., 2016). Critic-free methods such as GRPO (Shao et al., 2024) instead compute a baseline from a group of repeated rollouts per prompt, which moves the cost to throughput and needs the environment to be reset at the exact same initial states, but still concentrates aggressively on cardinal returns (Mei et al., 2020a). A practical critic-free RFT algorithm should thus update more conservatively to reduce entrenchment on noisy samples and draw its variance reduction from an alternative source for stability without GRPO's or PPO's constraints.

We propose *Follow the Winners* (FTW), a critic-free policy-learning algorithm that induces polynomial concentration in the order statistic of returns. This behavior stems from our trajectory filtering procedure, which adapts the cross-entropy method (Rubinstein, 1997, 1999; Botev et al., 2013) to deep RL through minibatched filtering over a replay-buffer proposal that amortizes data generation and serves as the alternative source of variance reduction. We show that the well-known post-training algorithm DPO (Rafailov et al., 2023) and FTW share a bounded risk-seeking offset, which we identify as inherent to reducing variance through ordinal filters on samples. Furthermore, we show that GRPO is risk-neutral and that FTW controls its risk offset through its selection pressure.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Our analysis is grounded in the Control-as-Inference (CAI) framework (Ziebart, 2010; Levine, 2018; Kappen et al., 2012), which we extend by recasting critic-free RFT as inference over potentials defined on trajectory batches rather than individual trajectories. The resulting variational objective motivates the cross-entropy method as a simple sampling-and-projection scheme for optimization, from which we derive FTW. Under the same view, DPO emerges as the pairwise special case of a Plackett–Luce potential (Luce, 1959; Plackett, 1975; Rafailov et al., 2023), while GRPO (Shao et al., 2024) optimizes a Rao–Blackwellized upper bound on the evidence of a multinomial potential. This places GRPO, DPO, and FTW in a shared design space in which we can juxtapose their modelling decisions and risk offsets, a second theoretical axis of trade-offs next to the practical considerations.

We validate our theory predictions on diagnostic bandit problems and show that FTW extends as a general policy learning method in Brax continuous-control. Scaled to agentic LLM post-training, FTW trains more stably and with lower seed variance than REINFORCE++ (Hu et al., 2025), and matches GRPO and PPO on Sokoban and the published Search-R1 baselines but with our alternative variance-reduction scheme, i.e., no value model, group rollouts or environment duplication. The LLM results show that the combined effect of FTW's polynomially concentrating learning rule and replay-buffer proposal is a viable alternative to critics or group rollouts for counteracting variance.

2 Background

Large Language Models. A large language model (LLM) defines an autoregressive distribution over token sequences that typically uses an attention-based neural architecture (Vaswani et al., 2017). We write this as $p_\theta(X_{1:T}) = \prod_{t=1}^{T} p_\theta(X_t \mid X_{<t})$, where $X_{<t} = (X_1, \ldots, X_{t-1})$ denotes the prefix of previously generated tokens, X_t denotes the next token, and θ parameterizes the model. The parameters θ^*_{pre} are fitted through supervised pre-training. Given a dataset of example sequences $\mathcal{D} = \{x^{(i)}_{1:T}\}_{i=1}^{N}$, this corresponds to solving $\theta^*_{\text{pre}} = \arg\max_\theta \mathbb{E}_{x_{1:T}\sim\mu_\mathcal{D}} \ln p_\theta(X_{1:T} = x_{1:T})$. In practice, many additional techniques are used to stabilize and improve training, including positional encodings (Su et al., 2024), layer normalization (Ba et al., 2016), gradient clipping, and tokenization.

Reinforcement Learning for Agentic LLMs. Agentic problems are partially observable Markov decision processes (POMDPs, Kaelbling et al., 1998; Spaan, 2012), specified by states $s \in \mathcal{S}$, observations $o \in \Omega$, actions $a \in \mathcal{A}$, a reward function $R(s_t, a_t) \triangleq R_t$, transition dynamics $p(S_{t+1} \mid S_t = s, A_t = a)$, observation kernel $p(O_t|S_t = s)$ and a discount factor $\gamma \in [0, 1]$. In the LLM setting, observations and actions both occupy a textual medium, although the formulation naturally extends to mixed-modality settings. Given a pretrained LLM $p_\theta(x_{1:T})$ that maps histories to next-token predictions, we write it as a history-dependent policy $\pi_\theta(A_t \mid H_t)$, where $H_t = (O_0, A_0, \ldots, O_{t-1}, A_{t-1}, O_t)$ denotes the interaction history available before taking action A_t. Observations O_t can be viewed as externally provided context tokens, which are masked in the output likelihood as they are not generated by the policy itself. The policy induces the marginal,

$$p_{\pi_\theta}(H_T) = \int dS_{0:T}\, p(S_0)\, p(O_0 \mid S_0) \prod_{t=0}^{T-1} \pi_\theta(A_t \mid H_t)\, p(S_{t+1} \mid S_t, A_t)\, p(O_{t+1} \mid S_{t+1}). \quad (1)$$

The goal of reinforcement learning is then to find parameters θ^*_{RFT} that maximize expected return, initialized from pretrained weights θ^*_{pre}, i.e., $\theta^*_{\text{RFT}} = \arg\max_\theta \mathbb{E}_{p_{\pi_\theta}(H_T, S_{0:T})}[\sum_{t=0}^{T} \gamma^t R(S_t, A_t)]$. We write the returns starting at any t as $G_t = G(H_{t:T}) \sim \text{Law}(\sum_{i=t}^{T} R_i \mid H_{t:T})$. When rewards are immediately available they can also be absorbed in the observations for to enable adaptation.

Control-as-Inference. The Control-as-Inference (CAI) framework (Levine, 2018; Kappen et al., 2012) reframes the optimization of θ^*_{RFT} as a probabilistic inference problem. The central idea is to tilt a prior trajectory distribution p_{π_θ} by an energy function $\Phi : \mathcal{H} \to \mathbb{R}$ that biases towards the expected return. This is commonly written by introducing a binary optimality variable $\mathcal{O}_t \in \{0, 1\}$ with conditional likelihood $\Phi_{1:T} \triangleq p(\mathcal{O}_{1:T} = 1 \mid H_T) \propto \exp \mathbb{E}_{S_{0:T}|H_T} G(H_T) = \prod_{t=1}^{T} \exp \mathbb{E}_{S_t|H_t} R_t$, so that high return trajectories receive larger posterior mass. By Bayes' rule, at any step t this induces a posterior policy, which is optimality-conditioned from t to T,

$$p_{\pi_\theta}(A_t \mid H_t, \mathcal{O}_{t:T} = 1) = \pi_\theta(A_t \mid H_t)\, \frac{\mathbb{E}_{H_T\sim p_{\pi_\theta}(\cdot|H_t,A_t)}\left[\mathcal{Z}(H_T, A_t)^{-1} \exp \mathbb{E}_{S_{t:T}|H_T} G_t\right]}{\mathbb{E}_{H_T\sim p_{\pi_\theta}(\cdot|H_t)}\left[\mathcal{Z}(H_T)^{-1} \exp \mathbb{E}_{S_{t:T}|H_T} G_t\right]}, \quad (2)$$

which reweights the prior policy $\pi_\theta(A_t \mid H_t)$ according to the exponentiated expected return. The partition function $\mathcal{Z}$ normalizes the potential, and G is a deterministic function over the trajectories. A well known property of this posterior policy is that it is exactly the optimal policy for a KL-regularized augmentation of the base MDP (see Appendix E). In particular, it solves the variational optimization,

$$p_{\pi_\theta}(\cdot \mid \mathcal{O}_{t:T} = 1) = p_{q^*}(\cdot) = \arg\max_{q \in \Pi} \mathbb{E}_{p_q}[\ln p(\mathcal{O}_{t:T} = 1 \mid H_{t:T})] - \mathbb{E}_{p_q} D_{\mathrm{KL}}(q \,\|\, \pi_\theta)\,, \quad (3)$$

where $\Pi = \{\pi \mid \pi : \mathcal{H} \to \Delta(\mathcal{A})\}$ denotes the set of admissible policies. This makes the framework especially useful in deep RL, where regularized policy improvement has proven effective in practice.

The Cross-Entropy Method. To solve Eq. (3), algorithms such as MPO (Abdolmaleki et al., 2018) sample directly from Eq. (2) before projecting the resulting solution back onto a parametric model[1]. This reflects a broader *bootstrap-and-project* scheme: trajectories are drawn from the proposal $p_{\pi_\theta}(H_T)$, reweighted by an unnormalized potential $\Phi(\cdot)$ encoding the optimization direction, and the resulting Φ-tilted empirical distribution is projected back onto the parametric family of π_θ to define the next prior. A useful parallel can be drawn here to the cross-entropy method (CEM) for optimization (Rubinstein, 1997, 1999; Botev et al., 2013), see Algorithm 1. CEM similarly draws a batch $\{x_i\}_{i=1}^N \sim q_{\psi_t}$, scores each sample by $S(x_i)$, and iteratively concentrates q_{ψ_t} on high-scoring x_i in order to approach $\arg\max_x S(x)$. Concretely, it uses an indicator potential $\Phi(x) = \mathbf{1}_{S(x) \geq \mu_t}$, where μ_t is the $(1-\rho)$-quantile of S under q_{ψ_t} for $\rho \in (0,1)$. In practice, μ_t is estimated by the score of the $\lceil \rho N \rceil$-th best batch-sample. The indicator then retains the top-quantile samples, and the parametric family is refit to this elite set by cross-entropy minimization, typically in closed-form. We adapt this for deep-RL in Section 3.

Algorithm 1: Cross-Entropy Method.

Input: Proposal q_ψ, score S, batch size N, elite fraction $\rho \in (0,1)$, iterations T.

for $t = 1, \ldots, T$ **do**

- Sample $\{x_i\}_{i=1}^N \sim q_{\psi_t}$
- Score each sample, $s_i \leftarrow S(x_i)$
- Set threshold μ_t to the $\lceil \rho N \rceil$-th best s_i
- Form elite set $\mathcal{E}_t \leftarrow \{x_i : s_i \geq \mu_t\}$
- Project $\psi_{t+1} \leftarrow \arg\max_\psi \sum_{\mathcal{E}_t} \ln q_\psi(x)$

Return q_{ψ_T}

3 Follow the Winners: Model Derivation & Elite Filtering

We derive *Follow the Winners* (FTW) from a CAI problem in which optimality is specified through ordinal comparisons rather than cardinal returns. The idea is to perform inference not over a single trajectory, but over a finite batch of trajectories, and to then condition this batch on an ordinal observation which specifies trajectory preference. This yields an evidence lower bound whose optimization naturally suggests a sample-and-project method: tilt the sampling distribution toward trajectories that win local comparisons, then project the resulting distribution onto the policy class.

Model Derivation. Standard CAI formulates policy optimization as inference over individual trajectories H_T, typically by conditioning on binary optimality variables $\mathcal{O}_t \in \{0,1\}$. In agentic RFT, however, scalar trajectory returns can be sparse, noisy, and difficult to calibrate across tasks, while relative comparisons among sampled rollouts are often more reliable. We therefore replace inference over individual trajectories with inference over finite collections $\mathcal{B}_N = (H_T^1, \ldots, H_T^N)$. Under a base sampling policy π, a collection is distributed according to the product measure $p_\pi(\mathcal{B}_N) = \prod_{n=1}^N p_\pi(H_T^n)$. The collection $\mathcal{B}_N$ provides a local comparison set for defining ordinal feedback.

To encode this feedback, let $\mathcal{C}(\mathcal{B}_N) = (i_1, \ldots, i_N)$ denote the ranking induced over the collection, where $H_T^{i_1} \succ \cdots \succ H_T^{i_N}$. The binary optimality variable of standard CAI is then replaced by an ordinal observation variable $\mathcal{O}_T$, whose support depends on the likelihood model. For instance, under a Plackett–Luce model, $\mathcal{O}_T$ denotes a full permutation σ of the trajectories, while for a multinomial model, $\mathcal{O}_T \in \{1, \ldots, N\}$ denotes the identity of a preferred trajectory. We write $\Phi_T(\mathcal{B}_N)$ for the likelihood, or unnormalized potential, assigned to the batch $\mathcal{B}_N$ under the ordinal observation.

For simplicity, we focus on the sparse final-reward setting and condition only on $\mathcal{O}_T$, avoiding partial trajectory feedback through $\mathcal{O}_t$ for $t \neq T$. We further assume that the ranking rule $\mathcal{C}$ is consistent

[1] This is orthogonal to the constrained optimization of the Lagrangian in MPO, and the fact that MPO iteratively updates the reference policy π_θ as part of an expectation-maximization loop (Neal and Hinton, 1998).

Algorithm 2: Follow the Winners (FTW).

Input: Policies $\pi_\theta, \pi_{\text{ref}}$, buffer size M, batch size C, winners K, batches J, KL weight λ.

Initialize FIFO queue $\mathcal{B}_M \leftarrow \emptyset$ s.t. $|\mathcal{B}_M| \leq M$

while *not converged* **do**

- Collect (async) $H_{0:T} \sim \pi_{\theta_t}$ and append to $\mathcal{B}_M$
- **for** $j = 1, \ldots, J$ **do**
  - Sample $\mathcal{B}_C^{(j)} = \{H_{0:T}^{(i)}\}_{i=1}^{C} \sim \text{Unif}(\mathcal{B}_M)$
  - Score each $H_{0:T}^{(i)} \in \mathcal{B}_C^{(j)}$ by $G(H_{0:T}^{(i)})$
  - Elite-set $\mathcal{E}^{(j)} \leftarrow$ top-K trajectories under G
- $\nabla_\theta \mathcal{L} \leftarrow \mathbb{E}_{\bigcup_j \mathcal{E}^{(j)}}[\nabla_\theta \ln \pi_\theta] - \lambda \nabla_\theta D_{\text{KL}}$ Eq.(4)
- Project $\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \mathcal{L}$

Return π_θ

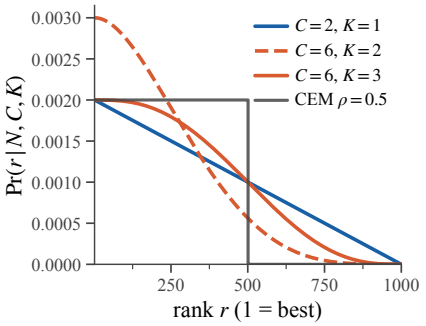


Figure 1: Normalized marginal occupancy of rank r under FTW's elite-filtering, Eq. (5), for $N = 1000$ and varying C, K. Larger C sharpens concentration on top-ranked trajectories; larger K softens it.

across subsets, satisfying independence of irrelevant alternatives (IIA), which implies a total ordering over the trajectory domain (McFadden, 1974; Luce, 1959). This assumption is important for our algorithmic construction: IIA ensures that the per-minibatch rankings used by FTW's elite-filtering step are consistent with an underlying total ordering of conditional returns.

Under our ordinal CAI construction, the marginal is lower-bounded as (compare with Eq. (3)):

$$\ln p_{\pi_\theta}(\mathcal{O}_T) = \ln \mathbb{E}_{p_{\pi_\theta}(\mathcal{B}_N)} \Phi_T(\mathcal{B}_N) \geq \mathbb{E}_{p_q}\left[\ln \Phi_T(\mathcal{B}_N) - N \cdot D_{\text{KL}}(q \,\|\, \pi_\theta)\right] = \mathcal{L}_\Phi(q, \pi_\theta). \quad (4)$$

The ELBO $\mathcal{L}_\Phi$ has a direct (regularized) policy-improvement interpretation. The potential Φ_T tilts the proposal q toward the preferred in-batch items, while the KL term constrains the update to the reference π_θ. Optimizing the bound therefore alternates between two simple operations: filtering better trajectories and projecting the tilted distribution onto the policy class. This structure is exactly what underpins the cross-entropy method (CEM) for optimization, as shown in Algorithm 1.

Algorithm Construction. FTW instantiates an adaptation of CEM for agentic RFT. As motivated in Section 1, we highlight three desiderata on the choice of proposal and ordinal potential relevant for the RFT setting. First, updates should be conservative: trajectory returns are high variance, and poor intermediate updates should not destabilize subsequent learning by collapsing onto noisy locally-strong trajectories. Second, variance reduction should come from the samples alone, for example through batch-level baselines (Hu et al., 2025), antithetic sampling, or rank transforms of returns, to avoid both a critic model and the environment duplication required for group rollouts. Third, the algorithm should reuse past trajectories to amortize data-collection costs and benefit from regularization and stability from replay (Mnih et al., 2015; Fedus et al., 2020; Arnal et al., 2026).

We show our FTW pseudocode in Algorithm 2 as our adaptation of CEM (Alg. 1). The proposal distribution q is the mixture of the h past policies, approximated in practice by the empirical occupancy of a FIFO replay buffer, $\hat{q} = \frac{1}{\min(h, t+1)} \sum_{i=\max(0,\, t-h+1)}^{t} \pi_{\theta_i}$, where h is the maximum age of samples retained in the buffer (Cornuet et al., 2012). Then, the closed-form projection step of classical CEM is replaced by a gradient-based projection onto the policy class. The main departure from standard CEM is how FTW constructs the elite set (filtering). Rather than sorting the entire replay buffer and retaining a global top $(1-\rho)$-quantile, FTW instead samples J independent minibatches of size C from the buffer and retains the top $K = \lceil \rho C \rceil$ per batch. The resulting union of $J \cdot K$ elites are then used in a cross-entropy gradient update. Thus, trajectories that are repeatedly preferred over other trajectories obtain larger weight in consecutive proposal distributions.

Filtering Probability. Our elite-filtering rule defines a combinatorial tilting distribution Φ over the replay buffer without explicit evaluation (which would require sorting the buffer). For a N-size buffer $\mathcal{B}_N$, let r denote the rank of an item under the buffer's theoretical ordering (lower is better).

Then, the probability that an item of rank r is selected as one of the K elites in a minibatch of size C,

$$\Pr(r \mid N, C, K) = \frac{1}{K\binom{N}{C}} \sum_{k=1}^{K} \binom{r-1}{k-1}\binom{N-r}{C-k}. \tag{5}$$

The derivation is given in Appendix D.2, Lemma 1. The induced potential Φ, as shown in Figure 1, is a degree-$(C-1)$ polynomial in r and is monotonically decreasing for all $K \geq 1$ with $K < C$ (Goldberg and Deb, 1991; David and Nagaraja, 2003). Increasing K spreads probability mass from the highest-ranked trajectories toward lower-ranked ones, softening selection pressure. The minibatch size C acts as a concentration parameter: $C = K$ recovers uniform sampling over the buffer, while increasing C for fixed K sharpens concentration on high-ranked trajectories. Together, the ratio $\rho = K/C$ controls the elite fraction and C controls how sharply that fraction is enforced. The limiting case $C = N$ with $J = 1$ recovers standard CEM filtering exactly, but amortizes data-collection poorly since the $N - K$ non-elite trajectories in the buffer are discarded each update. For intermediate $K < C < N$, the induced selection profile becomes smoother and more sigmoid-like, as illustrated for $C = 6$ and $K = 3$ in Figure 1. The special case $K = 1$, $C = 2$ reduces to binary tournament selection, connecting FTW to classical selection mechanisms in genetic algorithms (Goldberg, 1989; Goldberg and Deb, 1991). Thus, our polynomial profile meets two of the aforementioned desiderata, conservative updates and sample-only variance reduction, purely through the (C, K) filtering parameters.

4 Theoretical Analysis

This section diagnoses the characteristics of our algorithm, specifically its risk profile relative to baselines. We first recover GRPO and DPO within our extension of the CAI framework (Section 3) as the multinomial and Plackett-Luce potentials over sampled batches. We then use this recovery to derive the implicit risk profile of each method through the certainty equivalent of its selection rule. We show that GRPO is risk-neutral and that its otherwise (non-Jensen bounded) exponential multinomial potential is unboundedly risk-seeking. Then we show that DPO and FTW are boundedly risk-seeking toward a selection threshold that varies by the selection ratio K/C. We assume reader familiarity with GRPO and DPO and provide a self-contained primer in Appendix C.

4.1 Recovering GRPO and DPO

We treat the two potentials mentioned in Section 3 in turn. The Plackett–Luce potential at $|\mathcal{B}_N| = 2$ recovers DPO directly, while the multinomial potential admits a Jensen upper bound whose looser surrogate is exactly GRPO's group-mean advantage. We close the subsection by noting that, within our framework, the remaining difference between the two methods is the choice of baseline normalizer.

We first recover DPO through our formalism simply by plugging in the Plackett–Luce model for $N = 2$, under minor technical assumptions. Note that the authors of DPO (Rafailov et al., 2023) also sketch a Plackett–Luce generalization in their Appendix A.5, our result is complementary.

Corollary 1 (DPO from a N=2 PL)**.** *At $|\mathcal{B}_N| = 2$ with ordering $H_T^{(1)} \succ H_T^{(2)}$, the Plackett–Luce potential with weighting $g(H_T) = \exp(s_\theta(H_T)/\beta)$ and log-ratio score $s_\theta^{(i)} \triangleq \log \frac{\pi_\theta(H_T^{(i)})}{\pi_{ref}(H_T^{(i)})}$ reduces to $p(\mathcal{O}_T = (1,2) \mid \mathcal{B}_2) = \sigma(\frac{1}{\beta}(s_\theta^{(1)} - s_\theta^{(2)}))$, the Bradley–Terry sigmoid. The ELBO in Eq.* (4) *then recovers DPO's objective, with the prior π_θ in the KL term playing the role of DPO's reference π_{ref} and temperature β matching DPO's up to N=2 rescaling of our ELBO's KL (derived in App. E.5).*

The multinomial potential at general N requires more work. The log-partition $\ln \sum_k g_k$ in the ELBO couples all trajectories in the batch, so its policy gradient is dense. This can be seen by plugging the multinomial potential into Eq. (4) and averaging across all indices $i \in \{1, \ldots, N\}$ of $\mathcal{B}_N$,

$$\nabla_q \mathcal{L}_\Phi(q, \pi_\theta) \propto \mathbb{E}_{p_q}\left[\frac{1}{N}\sum_{n=1}^{N} \underbrace{(\ln g(n) - b_n)}_{\triangleq \alpha_n(\mathcal{B}_N)} \sum_{m=1}^{N} \nabla_q \ln p_q(H_T^m) - \nabla_q D_{\mathrm{KL}}(q \,\|\, \pi_\theta)\right], \tag{6}$$

which produces the double-summation over coefficients α_n and policy gradients. We sidestep this double-sum through a Rao–Blackwell argument that decouples the gradient into per-trajectory coefficients α_n, upon which a Jensen upper bound on the log-partition reveals GRPO's baseline.

Proposition 1 (GRPO as a Jensen upper bound)**.** *Under the multinomial potential with exponential weight $g(H_T) = \exp G(H_T)$, Rao–Blackwellization of Eq.* (6) *yields the per-trajectory form*

$$\nabla_q \widehat{\mathcal{L}_\Phi}(q, \pi_\theta) \propto \mathbb{E}_{p_q}\left[\frac{1}{N}\sum_{n=1}^{N} \alpha_n(\mathcal{B}_N) \sum_{t=1}^{T} \nabla \ln q(A_t^n \mid H_t^n)\right] - \nabla_q D_{\mathrm{KL}}(q \,\|\, \pi_\theta)\,, \tag{7}$$

with coefficients $\alpha_n(\mathcal{B}_N) = \ln g_n - \ln \sum_k g_k$ admitting the Jensen upper bound $\alpha_n \le G(H_T^n) - \frac{1}{N}\sum_k G(H_T^k)$, whose right-hand side is GRPO's group-mean advantage.

Proof sketch (full proof in Appendix D.1). Rao–Blackwell. The multinomial potential applies symmetrically to any item in $\mathcal{B}_N$, so we can hold one trajectory fixed and marginalize over the rest. This yields per-trajectory coefficients $\alpha_n(\mathcal{B}_N) = \ln g_n - \ln \sum_k g_k$, recovering an advantage form with log-partition baseline. *Jensen bound.* By concavity of $\ln$, $\alpha_n \le \ln g_n - \frac{1}{N}\sum_k \ln g_k$. Where the right-hand side reduces to $G(H_T^n) - \frac{1}{N}\sum_k G(H_T^k)$, which is GRPO's group-mean advantage. □

The Jensen bound comes at a cost: maximizing an upper bound does not guarantee maximization of the underlying objective. The gap is precisely the risk offset that the multinomial potential induces under uncertain returns, which we profile in the next subsection. More broadly, when viewed from this framework, the key difference between the multinomial and Plackett–Luce potentials, from which GRPO and DPO are positioned, is the choice of baseline normalizer. The multinomial potential applies a uniform group-mean across all items, while the Plackett–Luce model induces rank-dependent baselines that dampen updates at the extremes of the batch.

4.2 Implicit Risk Profiles of RFT Algorithms

The CAI posterior in Eq. (2) marginalizes the return G over the return law. In deterministic, perfectly observable settings this collapses to $\mathbb{E}_{S_{t:T}|H_T} \exp G(H_{t:T}) = \exp G(H_{t:T})$. Under aleatoric noise or epistemic uncertainty from partial observability, the marginalization on returns induces an implicit risk profile in the algorithm. In critic-based methods such as SAC, we learn a Q-function surrogate to move the expectation inside $\exp \mathbb{E}_{S_{t:T}|H_T} G_{t:T}$, in critic-free RFT we have no such fallback.

To make this risk-profile precise, we reduce each potential Φ to a per-entry selection weight $w_\Phi(H_T) = \mathbb{E}_{G|H_T}[\varphi(G)]$ with an increasing utility φ through Rao-Blackwellization. We then define the risk offset $\mathcal{R}_\varphi(H_T) = J_\varphi(H_T) - \mu(H_T)$, where $J_\varphi = \varphi^{-1}(w_\Phi)$ is the certainty equivalent (Pratt, 1964) of the entry's return (see the full setup in Appendix D.3). The offset illustrates the deviation from the conditional mean return $\mu = \mathbb{E}[G|H_T]$ as a function of its spread $\sigma^2 = \mathrm{Var}[G|H_T]$.

Proposition 2 (Risk profiles of RFT potentials under Gaussian noise)**.** *For Gaussian returns $G \mid H_T \sim \mathcal{N}(\mu, \sigma^2)$ and an assumed Gaussian buffer marginal $F_0 = \mathcal{N}(\mu_0, \sigma_0^2)$, the risk offset $\mathcal{R}_\varphi$ of each potential is as follows (proof and other return families in Appendix D.3):*

- *(a)* Multinomial potential & GRPO (CAI on cardinal returns): $\mathcal{R} = \frac{1}{2}\sigma^2$ *exactly. The Jensen step that positions GRPO (Prop. 1) replaces $\ln \sum_k \exp G_k$ with $\frac{1}{N}\sum_k G_k$, which is an affine utility, so $\mathcal{R} = 0$ for every return law. GRPO is risk-neutral.*
- *(b)* Bradley–Terry potential (DPO or Plackett–Luce at $N = 2$): $\mathcal{R} = (\mu - \mu_0)\big(\frac{\sigma_0}{\sqrt{\sigma_0^2+\sigma^2}} - 1\big)$ *exactly. The offset is bounded by $|\mu - \mu_0|$ and $J_\varphi \to \mu_0$ as $\sigma \to \infty$.*
- *(c)* Rank-based potential (FTW, top-K of C): $\mathcal{R} \approx (\mu - t)\big(\frac{\sigma_0 s_{C,K}}{\sqrt{\sigma^2+\sigma_0^2 s_{C,K}^2}} - 1\big)$*, using a normal approximation of the K-th order statistic. The same as (b) with μ_0 replaced by the threshold $t = \mu_0 + \sigma_0 a_{C,K}$, where $a_{C,K}$ and $s_{C,K}$ are the standardized mean and standard deviation of that order statistic. Aggressive filters $K \ll C$ have $a_{C,K} > 0$ and so $t > \mu_0$, mild filters $K \ge C/2$ have $a_{C,K} \le 0$ and $t \le \mu_0$, finally $(C, K) = (2, 1)$ recovers (b) exactly.*

For the ordinal potentials (b) and (c), the offset is paid in how often noise swaps preference orderings within the batch rather than in the magnitude of the returns, which is also what makes it bounded. This can be desirable when σ is epistemic, since for returns with μ below the attractor (μ_0 or t) the offset is positive and acts as an exploration term to encourage knowledge-seeking. However, when σ is aleatoric, the same offset biases the fixed point away from the mean-optimal policy, treating reward

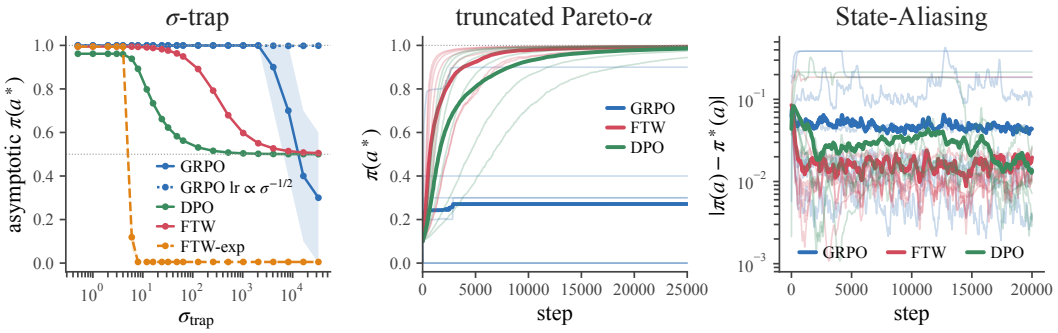


Figure 2: Diagnostic bandits isolating algorithm properties (Sec. 4). **Left (σ-trap):** asymptotic $\pi(a^*)$ vs. distractor variance. GRPO is only stable for large σ when proportionally decreasing the learning rate. **Center (Pareto-α):** under heavy tail returns, GRPO learning destabilizes whereas rank-based FTW/DPO remain stable. **Right (state-aliasing):** FTW achieves lower L1 distance $|\pi(a) - \pi^*(a)|$ between the learned and optimal action distributions in the POMDP. Left all lines show mean and 95% bootstrapped CIs over replay buffer sizes $N \in \{16, 64, 256, 1024, 4096\}$ and 10 seeds. Center and right: bold lines are averages over sampled learning-rates, translucent lines show sample trajectories.

noise as if it were information (de Vries et al., 2026). GRPO's risk-neutrality avoids both effects at the cost of neutrality. In FTW, the elite ratio K/C determines the attractor of FTW. Mild filters $K \geq C/2$ place it at or below the buffer mean μ_0, it equals DPO at $(C, K) = (2, 1)$, and aggressive filters $K \ll C$ place it above μ_0. For both FTW and DPO, the offset is bounded by $|\mu - t|$.

A separate consequence is the cardinal treatment of returns, which can be sensitive to the tail of the return distribution. The offset in (a) requires the moment-generating function of the return to be finite, which fails for heavy-tailed returns (Pareto-$\alpha \leq 1$, Student-t with $\nu \leq 2$). Although the Jensen bound that positions GRPO removes the offset itself, the same heavy tails destabilize GRPO's cardinal-return baseline numerically. Rank-based potentials don't suffer from this as they only depend on within-batch order statistics, which remain bounded for any continuous distribution.

5 Validation on Bandit & Deep Reinforcement Learning

We test the predictions of Section 4 on a controlled testbed before scaling up to LLM post-training. The predictions are that GRPO is risk-neutral but numerically unstable under heavy-tailed returns and that DPO and FTW are boundedly risk-seeking toward an elite threshold set by the selection ratio K/C, coinciding at $(C, K) = (2, 1)$. Furthermore, our FTW method is a general method that can be applied in the deep-RL setting, however its ordinal and conservative nature make it both slower and more stable in terms of the learning curves (compared to a baseline SAC).

Bandit settings. We construct three diagnostic tasks. The *σ-trap* is a multi-armed bandit in which a zero-mean distractor arm has much higher return variance than the optimal arm with mean 10, $\sigma_{\text{trap}} \gg \sigma^*$. A risk-neutral learner commits to the optimal arm, while a sufficiently risk-seeking learner gets distracted by the noise of the suboptimal arm as σ grows. Secondly, the *Pareto-α* bandit draws rewards from a truncated Pareto-α distribution and sweeps the tail index α to test whether each method's fixed point remains well-defined as the moment-generating function diverges at $\alpha \leq 1$. Finally, the *state-aliasing* task collapses two distinct latent states onto the same observation with a deterministic reward, so maintaining policy stochasticity is needed for low regret. This isolates the concentration rate from the risk profile. We describe the bandit setup in more detail in Appendix G.4.

The results in Figure 2 agree with the theory on all three items. In the σ-trap (left), GRPO converges on the optimal arm so long as its update rule remains numerically stable given the increasing gradient magnitude, which is only achieved when we decrease the learning rate as $\sigma^{-1/2}$. We also see that DPO is distracted earlier by noise than FTW, but both degrade to the same uniform policy eventually. This is the coin-flip limit which Prop. 2 predicts for bounded operators once the distractor's draws land above and below the optimal mean with equal probability (Appendix D.3). Although FTW at $(2, 1)$ shares DPO's risk offset, DPO's earlier collapse is due to its more aggressive exponential

concentration in the update rule, a dynamical entrenchment effect of DPO's loss which our fixed-point theory does not capture. As expected, the exponential potential over the buffer (FTW-exp) collapses onto the trap arm the fastest, see also Appendix H.2 for more variations of the elite-ratio parameters (C, K). On the Pareto-α bandit (center), GRPO fails to identify the optimal arm in the heavy-tailed regime, while DPO and FTW remain stable. This matches the prediction that the rank-based potential has a well-defined fixed point for any continuous return distribution whereas the cardinal-exponential potential does not. On the aliasing task (right), FTW achieves lower L1 distance $|\pi(a) - \pi^*(a)|$ between the learned and optimal action distributions compared to GRPO and DPO, when averaged over a learning-rate population. This is consistent with polynomial concentration retaining more stochasticity under partial observability. See Appendix H for additional Bandit results and Appendix F for a knowledge-seeking experiment left out of the main paper.

Deep RL. We also evaluate on Brax HalfCheetah (Freeman et al., 2021) to check that FTW functions as a more general deep-RL algorithm. Figure 3 compares FTW-K1-C64 and FTW-K4-C64 against SAC (Haarnoja et al., 2018). FTW is less sample-efficient than SAC in the early phase of training, as expected from the ordinal treatment of returns that discards return magnitude. By the end of the 10^9-step budget, FTW-K4-C64 reaches mean return comparable to SAC with visibly tighter seed-to-seed variation. Thus, our conservative ordinal update also extends to deep-RL settings without a critic, which is the regime of LLM post-training.

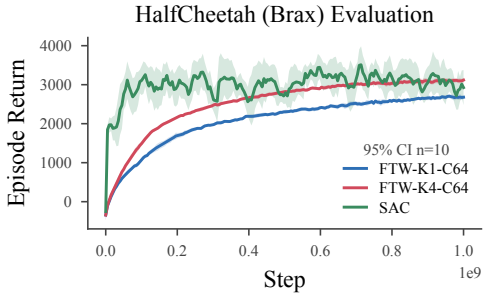
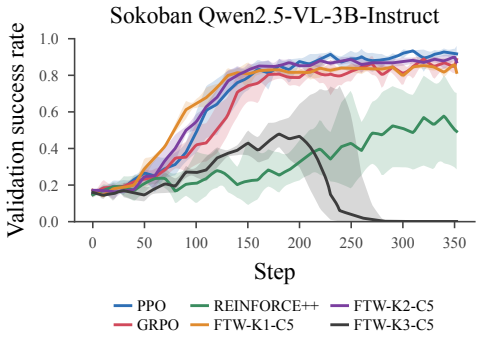


Figure 3: Evaluation learning curves on HalfCheetah after 1B steps of training. Shaded regions are 95% Gaussian CIs over 10 seeds.

6 LLM Post-Training Experiments



Qwen2.5-3B-Instruct

Method	Avg. Test Acc. %
Search-R1 (PPO)	32.5
Search-R1 (GRPO)	33.6
FTW-K1-C4	35.46 ± 0.15 SD

Figure 4: Sokoban test success rate curves (left, 128 puzzles, mean ± one standard deviation over 5 seeds) and final exact-match accuracy ± one standard deviation on the Search-R1 test set over 3 seeds (right, 50k+ examples). PPO and GRPO results are from Jin et al. (2025).

We evaluate Follow the Winners (FTW) on two agentic LLM post-training benchmarks: Sokoban (Schrader, 2018), a visual box-pushing puzzle that requires multi-turn planning, and Search-R1 (Jin et al., 2025), an open-domain question-answering task with a retrieval tool. We interleave chain-of-thought with tool or action outputs and use sparse terminal rewards together with a -0.1 penalty for formatting errors. We reuse the experimental setup of Feng et al. (2025), see Appendix G.4.

We ablate the FTW filtering operator for $C = 5$ and $K \in \{1, 2, 3\}$ against GRPO, PPO, and REINFORCE++ on Sokoban using the Qwen2.5 model-line (Qwen Team et al., 2025) with a replay buffer of size 2000. On Search-R1 we use FTW with $C = 4$ and $K = 1$ with a 10k replay buffer. GRPO uses $n = 8$ rollouts per prompt and is the only method that needs duplicated group rollouts.

Figure 4 (left) shows test success rate curves on Sokoban. For $K \in \{1, 2\}$ FTW clusters with GRPO and PPO in learning speed and performance, showing that FTW performs competitively despite a much simpler update, without PPO's value function model or GRPO's duplicated rollouts. Notably, for $K = 3$ FTW collapses after peaking around 0.4. Proposition 2 shows that this filter is relatively

mild, its elite threshold for selection sits exactly at the buffer mean. Under continuous returns this means that there should be monotonic improvement (cf. the $K{=}3$, $C{=}4$ bandit in Appendix H.2). Under Sokoban's binary returns, however, a minibatch with fewer than K successes fills its remaining elite slots with zero-return trajectories that the cross-entropy loss then imitates, whereas at $K = 1$ a single reward per minibatch would suffice. As the success rate falls this becomes more likely and the collapse reinforces itself. We attribute this to a finite-sample effect that larger batches and replay buffers dampen at the cost of slower learning (detailed in Appendix H.1). Finally, we also see that REINFORCE++, exhibits large seed variance and slower learning, implying that dropping group rollouts or a value function without some variance reduction replacement is insufficient on Sokoban.

The table in Figure 4 (right) reports average exact-match accuracy on the 50k+ Search-R1 test set from Jin et al. (2025). We see that our FTW variant matches (within noise) the published Search-R1 PPO and GRPO baselines reported by Jin et al. (2025). This again supports our claim that FTW is a viable alternative to PPO, which needs more than double the GPU resources, and to GRPO, which needs group duplication (Appendix G.2). However, we do caution that the bandit experiments isolate the theoretical mechanisms more cleanly than what the LLM post-training setting can provide.

7 Related Work

The CAI framework on which we build has a long history in stochastic optimal control (Kappen et al., 2012; Todorov, 2009; Rawlik et al., 2012; Theodorou et al., 2010) and reinforcement learning (Levine, 2018; Ziebart, 2010; Dayan and Hinton, 1997; Toussaint, 2009), where it has motivated methods such as MPO (Abdolmaleki et al., 2018), REPS (Peters et al., 2010), and SAC (Haarnoja et al., 2018). We extend this line by recasting the tilting potential as conditioning on permutations or winner indices over sampled batches, rather than on a binary variable for individual trajectories.

Beyond GRPO and DPO, our framework subsumes a broader family of bootstrap-and-project methods. RAFT (Dong et al., 2023) corresponds to FTW with $C = N$, $K = 1$, and a buffer refreshed every prompt. ReST (Gulcehre et al., 2023) and ReST-EM (Singh et al., 2024) bootstrap toward an annealed threshold potential before projecting back onto the model parameters. Variational best-of-n (Amini et al., 2025) shares the single-batch top-1 selection structure ($C = N$, $K = 1$, no replay), though its KL-regularized variational objective differs from our cross-entropy projection, and STaR (Zelikman et al., 2022) is closely related as a hard top-1 selection criterion. Listwise preference methods such as LiPO (Liu et al., 2025a), PRO (Song et al., 2024), and RSO (Liu et al., 2024) use the PL likelihood directly as a training objective on the policy. A complementary approach to damping concentration is generalized-entropy regularization through Tsallis entropy (Lee et al., 2020) or general greedification operators (Chan et al., 2022), which for the right choice of greedification can also achieves polynomial concentration through the regularizer rather than directly through the potential. Recent work by Wu et al. (2026) also reach a related GRPO/DPO correspondence in a bottom-up way by reframing GRPO's group-relative advantage as a contrastive loss. Our derivation arrives at the same $N = 2$ Bradley–Terry limit top-down as the first-stage marginal of the PL model under CAI. We provide a more comprehensive discussion of the literature and its positioning to our work in Appendix B.

8 Conclusion

We introduced Follow the Winners (FTW), a critic-free policy-learning algorithm for agentic LLM post-training that is adapted from the cross-entropy method for optimization. FTW induces polynomial concentration to limit aggressive policy updates which can lead to entrenchment, and uses a replay-buffer proposal to reduce sample variance without the need for a critic model or group rollouts. On agentic LLM benchmarks, FTW matches GRPO and PPO on Sokoban and the published Search-R1 baselines, and trains more stably and faster than REINFORCE++ on Sokoban. This supports our algorithm motivation that FTW offers a viable trade-off, spending CPU memory in place of a value model or the environment duplication that group rollouts require. To derive the algorithm, we extended the Control-as-Inference framework to encompass contemporary critic-free RFT methods by reinterpreting the tilting potential as conditioning on permutations or winner indices over sampled batches. This recovers DPO exactly under the Plackett–Luce potential and positions GRPO as the Jensen upper bound on the corresponding multinomial objective, exposing GRPO as risk-neutral while DPO and FTW share a bounded risk offset that FTW controls through its selection

pressure (Proposition 2). The framework places GRPO, DPO, and FTW on a common design axis, and provides a top-down basis for principled algorithm design in the critic-free RFT regime.

Limitations. FTW's risk offset acts as an exploration term under epistemic uncertainty, but the same offset also biases the fixed point away from the mean-optimal policy under aleatoric noise (Section 4.2), making FTW (and DPO) less suited to environments with large irreducible noise. The replay-buffer-based proposal amortizes data-collection costs and provides additional regularization, but at the price of memory overhead. That said, CPU memory is often cheaper and more accessible than stricter GPU requirements. Finally, our IIA assumption (Section 3) underlying the elite-filtering interpretation requires a total ordering on the trajectory domain, and thus needs further adaptation for non-transitive settings such as optimizing for a Nash equilibrium.

Acknowledgments and Disclosure of Funding

This work was funded by Trent AI Limited. The authors declare no competing interests.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a Posteriori Policy Optimisation. In *International Conference on Learning Representations*, 2018.

Hany Abdulsamad, Sahel Iqbal, and Simo Särkkä. Sequential Monte Carlo for Policy Optimization in Continuous POMDPs. arXiv:2505.16732, 2025.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662.

Riad Akrour, Marc Schoenauer, and Michèle Sebag. Preference-Based Policy Learning. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2011)*, volume 6911 of *Lecture Notes in Computer Science*, pages 12–27. Springer, 2011. doi: 10.1007/978-3-642-23780-5_11.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. Variational Best-of-N Alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

Brandon Amos and Denis Yarats. The Differentiable Cross-Entropy Method. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pages 291–302, 2020.

Charles Arnal, Vivien Cabannes, Taco Cohen, Julia Kempe, and Remi Munos. Efficient RL Training for LLMs with Experience Replay. arXiv:2604.08706, 2026.

Hagai Attias. Planning by Probabilistic Inference. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, volume R4 of *Proceedings of Machine Learning Research*, pages 9–16. PMLR, 2003.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. arXiv:1607.06450, 2016.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander Nicholas D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Forty-second International Conference on Machine Learning*, 2025.

Georges Bonnet. Transformations des signaux aléatoires à travers les systèmes non linéaires sans mémoire. *Annales des Télécommunications*, 19(9–10):203–220, 1964. doi: 10.1007/BF03014720.

Zdravko I. Botev, Dirk P. Kroese, Reuven Y. Rubinstein, and Pierre L'Ecuyer. The Cross-Entropy Method for Optimization. In *Handbook of Statistics, Volume 31: Machine Learning—Theory and Applications*, pages 35–59. Elsevier, 2013. doi: 10.1016/B978-0-444-53859-8.00003-5.

Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pages 193–200. MIT Press, 2007.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 129–136. ACM, 2007. doi: 10.1145/1273496.1273513.

Alan Chan, Hugo Silva, Sungsu Lim, Tadashi Kozuno, A. Rupam Mahmood, and Martha White. Greedification Operators for Policy Optimization: Investigating Forward and Reverse KL Divergences. *Journal of Machine Learning Research*, 23(253):1–79, 2022.

Olivier Chapelle and Lihong Li. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Jean-Marie Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

H. A. David and H. N. Nagaraja. *Order Statistics*. Wiley, New York, 3rd edition, 2003. doi: 10.1002/0471722162.

Peter Dayan and Geoffrey E. Hinton. Using Expectation-Maximization for Reinforcement Learning. *Neural Computation*, 9(2):271–278, 1997. doi: 10.1162/neco.1997.9.2.271.

Joery A. de Vries, Jinke He, Yaniv Oren, and Matthijs T. J. Spaan. Trust-Region Twisted Policy Improvement. In *Forty-second International Conference on Machine Learning*, 2025.

Joery A. de Vries, Jinke He, Yaniv Oren, Pascal R. van der Vaart, Mathijs M. de Weerdt, and Matthijs T. J. Spaan. VariBASed: Variational Bayes-Adaptive Sequential Monte-Carlo Planning for Deep Reinforcement Learning. arXiv:2602.18857, 2026.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment. *Transactions on Machine Learning Research*, 2023.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting Fundamentals of Experience Replay. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3061–3071. PMLR, 2020.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1587–1596. PMLR, 2018.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A Theory of Regularized Markov Decision Processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2160–2169. PMLR, 2019.

David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

David E. Goldberg and Kalyanmoy Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1994.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling. arXiv:2308.08998, 2023.

Emil J. Gumbel. *Statistical Theory of Extreme Values and Some Practical Applications*, volume 33 of *Applied Mathematics Series*. U.S. Government Printing Office, Washington, D.C., 1954.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1861–1870. PMLR, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2555–2565. PMLR, 2019.

Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing Critic-Free Policy Optimization with Global Advantage Normalization. arXiv:2501.03262, 2025.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of Bayesian neural nets via local linearization. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR, 2021.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. arXiv:2503.09516, 2025.

Norman L. Johnson, Samuel Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions*. Wiley Series in Probability and Statistics. Wiley-Interscience, New York, 1997.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101(1):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*, pages 267–274. Morgan Kaufmann, 2002.

Hilbert J. Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine Learning*, 87:159–182, 2012. doi: 10.1007/s10994-012-5278-7.

Mohammad Emtiyaz Khan and Håvard Rue. The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46, 2023.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE Samples, Get a Baseline for Free! ICLR 2019 Workshop on Deep RL Meets Structured Prediction, 2019a.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3499–3508. PMLR, 2019b.

Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, Mineui Hong, Jaein Kim, Yong-Lae Park, and Songhwai Oh. Generalized Tsallis Entropy Reinforcement Learning and Its Application to Soft Mobile Robots. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, 2020. doi: 10.15607/RSS.2020.XVI.036.

Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. arXiv:1805.00909, 2018.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. Statistical Rejection Sampling Improves Preference Optimization. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J. Liu, and Xuanhui Wang. LiPO: Listwise Preference Optimization through Learning-to-Rank. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2404–2420, Albuquerque, New Mexico, 2025a. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.121.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. In *2nd AI for Math Workshop @ ICML 2025*, 2025b.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, Shengyi Huang, Siran Yang, Jiamang Wang, Wenbo Su, and Bo Zheng. Tricks or Traps? A Deep Dive into RL for LLM Reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, New York, 1959.

Matthew Macfarlane, Edan Toledo, Donal John Byrne, Paul Duckworth, and Alexandre Laterre. SPO: Sequential Monte Carlo Policy Optimisation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Shie Mannor, Reuven Y. Rubinstein, and Yohai Gat. The Cross-Entropy Method for Fast Policy Search. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 512–519, 2003.

Daniel McFadden. Conditional Logit Analysis of Qualitative Choice Behavior. In *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, 1974.

Jincheng Mei, Chenjun Xiao, Bo Dai, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. Escaping the gravitational pull of softmax. In *Advances in Neural Information Processing Systems*, volume 33, pages 21130–21140, 2020a.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6820–6829. PMLR, 2020b.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Remi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and Efficient Off-Policy Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Radford M. Neal and Geoffrey E. Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants. In *Learning in Graphical Models*, pages 355–368. Springer Netherlands, Dordrecht, 1998. doi: 10.1007/978-94-011-5014-9_12.

Harrie Oosterhuis. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness (Extended Abstract). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), Sister Conferences Best Papers*, pages 5319–5323, 2022. doi: 10.24963/ijcai.2022/743.

Yaniv Oren, Moritz Akiya Zanger, Pascal R. Van der Vaart, Mustafa Mert Çelikok, Wendelin Boehmer, and Matthijs T. J. Spaan. Value Improved Actor Critic Algorithms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative Entropy Policy Search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612, Menlo Park, CA, USA, 2010. AAAI Press. doi: 10.1609/aaai.v24i1.7727.

Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic Planning with Sequential Monte Carlo methods. In *International Conference on Learning Representations*, 2019.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient Cross-Entropy Method for Real-time Planning. In *Proceedings of the 2020 Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pages 1049–1065. PMLR, 2021.

R. L. Plackett. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975. doi: 10.2307/2346567.

Aloïs Pourchot and Olivier Sigaud. CEM-RL: Combining Evolutionary and Gradient-Based Methods for Policy Search. In *International Conference on Learning Representations (ICLR)*, 2019.

John W. Pratt. Risk aversion in the small and in the large. *Econometrica*, 32(1/2):122–136, 1964. doi: 10.2307/1913738.

Robert Price. A useful theorem for nonlinear devices having Gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, 1958. doi: 10.1109/TIT.1958.1057444.

Qwen Team, 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, Tianyi Tang, 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. arXiv:2412.15115, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference. In *Proceedings of Robotics: Science and Systems VIII*, Sydney, Australia, 2012. doi: 10.15607/RSS.2012.VIII.045.

Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 1997. doi: 10.1016/S0377-2217(96)00385-2.

Reuven Y. Rubinstein. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Max-Philipp B. Schrader. gym-sokoban. `https://github.com/mpSchrader/gym-sokoban`, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 2015. PMLR.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *The Fourth International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. arXiv:1707.06347, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300, 2024.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, 2024.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024. doi: 10.1609/aaai.v38i17.29865.

Matthijs T. J. Spaan. Partially Observable Markov Decision Processes. In *Reinforcement Learning: State-of-the-Art*, pages 387–414. Springer, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-27645-3_12.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to Summarize from Human Feedback. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *Journal of Machine Learning Research*, 11(104):3137–3181, 2010.

William R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294, 1933. doi: 10.1093/biomet/25.3-4.285.

Emanuel Todorov. Linearly-solvable Markov Decision Problems. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pages 1369–1376. MIT Press, 2007.

Emanuel Todorov. Efficient Computation of Optimal Actions. *Proceedings of the National Academy of Sciences*, 106(28):11478–11483, 2009. doi: 10.1073/pnas.0710743106.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1049–1056. Association for Computing Machinery, 2009. doi: 10.1145/1553374.1553508.

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 2094–2100, 2016. doi: 10.1609/aaai.v30i1.10295.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ruiyi Wang and Prithviraj Ammanabrolu. A Practitioner's Guide to Multi-turn Agentic Reinforcement Learning. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding Self-Evolution in LLM Agents via Multi-Turn Reinforcement Learning. arXiv:2504.20073, 2025.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing Multi-Turn Reasoning in LLM Agents via Turn-Level Reward Design and Credit Assignment. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, Mark Coates, and Jian-Yun Nie. It Takes Two: Your GRPO Is Secretly DPO. arXiv:2510.00977, 2026.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1192–1199. ACM, 2008. doi: 10.1145/1390156.1390306.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A Minimalist Approach to LLM Reasoning: from Rejection Sampling to Reinforce. arXiv:2504.11343, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. arXiv:2503.14476, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling Interaction via the Principle of Maximum Causal Entropy. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1255–1262, Madison, WI, USA, 2010. Omnipress.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: Variational Bayes-Adaptive Deep RL via Meta-Learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

Appendix: Table of Contents

A Disclosure Statements **18**

B Extended Related Work **18**

B.1 Control as Inference. 18

B.2 Reinforcement Learning from Human Feedback (RLHF). 19

B.3 Learning to Rank and Listwise Preference Optimization. 19

B.4 Policy Improvement Operators for Critic-Free RL. 20

B.5 Cross-Entropy Method for Deep RL. 21

C GRPO and DPO Primer **21**

D Proofs **22**

D.1 Rao-Blackwell Argument and GRPO upper-bound 22

D.2 Probability of Selection under Elite-Filtering 22

D.3 Risk Offsets across Return Distributions 23

E Extended Derivation **28**

E.1 Preliminaries 28

E.2 Batch-Level CAI and Ordinal Potentials 29

E.3 The Plackett–Luce Model 31

E.4 The Multinomial (Winner) Model 31

E.5 Policy Gradients and Recovery of GRPO and DPO 31

F Belief-Policy Mirror Descent under Batch CAI **33**

G Experimental Setup **37**

G.1 Hardware 37

G.2 Compute and Memory Footprint 37

G.3 Software Stack 37

G.4 Main Environments 38

G.5 Hyperparameters 39

H Supplementary Results **42**

H.1 Training-time Success Rates for Sokoban and Search-R1 42

H.2 Varying FTW Selection Operators for the noisy Bandit-trap 42

H.3 Bandit Distribution Sweep 43

H.4 State-Aliasing under Learning-Rate Variation 45

H.5 FTW under Tie-Saturated Rewards 45

H.6 Epistemic Contextual Bandit 46

NeurIPS Paper Checklist **50**

A Disclosure Statements

LLM Usage. LLMs were used for writing assistance (language and notation), literature search, and experiment infrastructure via agentic coding tools (Claude Code & Codex). All ideas, derivations, and results are solely the authors'.

Broader Impact. Our work contributes a general-purpose policy-learning algorithm for critic-free reinforcement fine-tuning of large language models. The contribution is methodological and is not tied to any specific deployed application, dataset, or released model.

As with most advances in LLM post-training, improvements in agentic capability can in principle support beneficial applications (such as research assistants, accessibility tooling, and scientific discovery) as well as harmful ones (such as automated disinformation, scam generation, or surveillance). The conservative, polynomially-concentrating update rule introduced by FTW does not differentially advantage either direction, and we clearly document limitations (namely the risk-seeking profile under aleatoric noise) that can cause unintended effects in practice. Our experiments use publicly available pre-trained models from the Qwen2.5 family and benchmarks intended for research, and we do not release post-trained model weights.

Relative to GRPO-style baselines, FTW incurs additional memory overhead from the replay buffer. This is partly offset by amortized data-collection costs, so the net compute footprint depends on the specific deployment regime; we compare the compute and memory footprints of FTW, GRPO, and PPO in Appendix G.2. Running, debugging, and tuning large-scale RL experiments is computationally costly and carries a non-trivial environmental footprint, which we acknowledge as a known concern of this line of research.

Reproducibility. We provide full pseudocode for FTW in Algorithm 2 as the method is intentionally simple. Appendix G documents the full experimental setup, covering hardware (Appendix G.1), software (Appendix G.3), exact environment and benchmark specifications (Appendix G.4), and all hyperparameters (Appendix G.5). All experiments build on publicly available components: the Qwen2.5 model family, known environments like Sokoban, Search-R1, and Brax. Our code is a fork of Feng et al. (2025). Additional results supporting the main-text claims are reported in Appendix H. We do not release code, models, or data until the manuscript is accepted and internal legal procedures are completed.

B Extended Related Work

We situate our method relative to four strands of work: control as inference; RLHF and preference optimization; policy improvement in the critic-free setting; and CEM methods, together with a brief discussion of order statistics and their implications for our hyperparameter choices.

B.1 Control as Inference.

The Control-as-Inference framework (Levine, 2018; Ziebart, 2010; Ziebart et al., 2010; Kappen et al., 2012; Toussaint, 2009; Attias, 2003; Dayan and Hinton, 1997; Rawlik et al., 2012; Todorov, 2007, 2009; Theodorou et al., 2010) recasts sequential decision-making as probabilistic inference in a graphical model. Under specific modelling assumptions, the posterior in this formulation coincides with the optimal or soft-optimal control law, making tools from probabilistic inference and stochastic optimal control directly applicable to the control problem. This dual formulation enables an interchange of methods between the two formalisms: examples include sequential Monte Carlo and particle filtering for planning (Piché et al., 2019; Macfarlane et al., 2024; de Vries et al., 2025), expectation-maximization for policy search as used by REPS and MPO (Dayan and Hinton, 1997; Abdolmaleki et al., 2018; Peters et al., 2010), and amortized variational inference for control (Zintgraf et al., 2021; de Vries et al., 2026). A further merit is that the probabilistic interpretation extends naturally to the deep reinforcement learning setting with high data throughput, since the free-energy formulation yields natural regularization for approximate policy improvement (Haarnoja et al., 2018; Chan et al., 2022; Geist et al., 2019).

The formalism usually builds on the idea of exponentiated reward potentials that tilt a prior trajectory distribution so that the resulting posterior concentrates on high-return trajectories. Under specific

choices of potential and dynamics, this recovers maximum-entropy reinforcement learning (Ziebart et al., 2010) and, in the continuous-time setting with KL or quadratic control costs, a linearized form of the Hamilton–Jacobi–Bellman equation (Kappen et al., 2012; Todorov, 2009; Theodorou et al., 2010). The framework is usually applied in the Markov decision process setting (Levine, 2018); we instead develop a formulation for the partially observable case, for which related treatments can be found in Abdulsamad et al. (2025); de Vries et al. (2026). Our contribution differs from the standard single-trajectory exponential tilting view in two ways: we allow ordinal likelihoods over finite sets of sampled trajectories, and we replace exponential return-based tilting with a polynomial, order-statistic-based tilting mechanism. The combination with a tailored cross-entropy method for optimization yields the conservative, approximate, policy-iteration method from the main text.

B.2 Reinforcement Learning from Human Feedback (RLHF).

RLHF (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022) has driven much of the recent LLM post-training literature. The shift toward critic-free methods such as GRPO and its variants (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b, 2026), REINFORCE++ (Hu et al., 2025), and RLOO (Kool et al., 2019a; Ahmadian et al., 2024), and toward direct preference optimization without an explicit reward model such as DPO (Rafailov et al., 2023) and listwise extensions (Liu et al., 2025a; Song et al., 2024), has produced a fragmented landscape that our derivation seeks to unify. Methods specifically targeting multi-turn agentic settings such as StarPO (Wang et al., 2025), Multi-Turn GRPO (Wang and Ammanabrolu, 2025; Wei et al., 2025), or GiGPO (Feng et al., 2025) share our operational regime but treat the multi-turn structure through specialized advantage estimators or credit-assignment heuristics, rather than through a top-down probabilistic framework.

Within this landscape, our framework places GRPO-family and DPO-family methods on a common axis (subset size N and choice of likelihood) and identifies common heuristics such as group baselines, shared prefixes, and contrastive losses as consequences of specific top-down modelling choices rather than fundamental algorithmic differences. Although we discussed RAFT (Dong et al., 2023) and similar methods in the main text, the off-policy extension was introduced by RAFT++ (Xiong et al., 2025) which is parallel to our use of the replay buffer. A closely related thread is STaR (Zelikman et al., 2022), which bootstraps a reasoning policy by iteratively fine-tuning on self-generated rationales that lead to correct answers, and can be viewed as a degenerate form of bootstrap-and-project with a hard top-1 selection criterion. Recent work by Wu et al. (2026) reaches a related correspondence bottom-up by reframing GRPO's group-relative advantage as a contrastive loss, and shows a gradient-level equivalence in the $N = 2$ case. Our derivation arrives at the same $N{=}2$ Bradley-Terry limit top-down as the first-stage marginal of the PL model under CAI, and takes a broader perspective through the lens of tilting potentials (i.e., the Multinomial or Plackett-Luce).

B.3 Learning to Rank and Listwise Preference Optimization.

The Plackett-Luce (PL) family is one of several ordinal potentials our framework admits, and the one through which we recover DPO (Rafailov et al., 2023) as the $N{=}2$ special case. This places our work in contact with a long line of research on PL-based ranking outside of LLM alignment. Classical learning-to-rank (Cao et al., 2007; Xia et al., 2008; Oosterhuis, 2022) fits a parameterized scoring function to observed rankings via maximum likelihood under PL or related listwise losses. Our formulation is closest to PL-Rank (Oosterhuis, 2022), but differs in two ways. First, we do not parameterize the PL model on an external scoring function. Instead, our PL likelihood arises from the Control-as-Inference construction (Levine, 2018) over sampled trajectory collections, and the policy is updated through the resulting posterior rather than fit to PL labels directly. Second, the CAI derivation supplies a KL-regularization term against a reference policy that is absent in standard listwise learning-to-rank.

Preference-based policy learning (Akrour et al., 2011), and its deep RL version (Christiano et al., 2017), form another branch in reinforcement learning that learns a preference model on trajectory data as a surrogate for the reward. This surrogate is then optimized akin to a value function instead of the direct returns. DPO (Rafailov et al., 2023) collapses this two-stage procedure into a single objective under a Bradley-Terry model, and sketches a PL generalization in their Appendix A.3. Our derivation can be read as making their sketch more explicit. The PL likelihood is recovered top-down from CAI rather than imposed as a ranking loss, which yields the ELBO and its policy gradient for optimization as consequences rather than design choices.

Listwise preference optimization methods such as LiPO (Liu et al., 2025a) and PRO (Song et al., 2024) use the same PL likelihood as a training objective directly on the LLM's response distribution. Rather than fitting an external scoring function as in classical LTR, they treat the policy itself as the PL ranker over candidate responses, perhaps similar to standard learning-to-rank. The two methods differ in their loss construction. LiPO uses a rank-aware listwise loss inspired by LambdaRank (Burges et al., 2007), which weights each pair by its contribution to a target ranking metric. PRO uses sequential elimination with a Bradley-Terry-style contrast at each rank position, which is one approach to sample from the standard PL factorization. RSO (Liu et al., 2024) is closely related and uses Boltzmann acceptance to source preference data closer to the optimal policy, which connects to our (ordinal) exponential-tilting view in Section 2 but without the ELBO grounding.

B.4 Policy Improvement Operators for Critic-Free RL.

The classical policy improvement theorem (Howard, 1960; Sutton and Barto, 2018) states that replacing actions by those with higher Q-value under the current policy yields a non-degrading update. This guarantee is the foundation of approximate policy iteration, and it depends on having access to a Q-estimate against which to greedify. In conventional deep reinforcement learning that estimate comes from a parametric value function trained alongside the policy, and a substantial body of work studies how to effectively estimate Q (Mnih et al., 2015; van Hasselt et al., 2016; Munos et al., 2016) and how aggressively to move the policy toward its argmax (Kakade and Langford, 2002). For instance, Chan et al. (2022) formalize greedification through forward and reverse KL projections, showing that the choice of divergence materially affects stability. Oren et al. (2025) extend the view to actor-critic methods and show that more conservative greedification operators improve performance in the deep setting. Trust-region methods such as TRPO and PPO (Schulman et al., 2015, 2017), and relative-entropy policy search (Peters et al., 2010), take a complementary route by constraining the magnitude of the per-step policy update against a reference policy. On the theoretical side, Mei et al. (2020a,b); Beirami et al. (2025) show that softmax policy gradient and best-of-n methods concentrate exponentially fast on the argmax-action for the current value estimate, which can be desirable for swift convergence when the estimate is well-calibrated.

Critic-free RFT for LLMs breaks the assumption underlying this toolbox. Methods such as GRPO (Shao et al., 2024), REINFORCE++ (Hu et al., 2025), and RAFT (Dong et al., 2023) discard the value function for compute reasons and replace it with batch-relative statistics: the group-mean baseline in GRPO, the global advantage normalization in REINFORCE++, the top-1 winner in RAFT. The estimate against which the policy is greedified is no longer a stable function approximator but a Monte Carlo statistic over N samples per prompt. For small N this estimate is high-variance, and the exponential concentration of softmax policy gradients amplifies that variance into the policy update. This can cause *entrenchment*, where one trajectory with large positive empirical advantage receives an outsized push, and the policy commits to it before subsequent samples can correct the estimate. KL-regularization slows this commitment but does not eliminate it (Geist et al., 2019), since the asymptotic concentration profile of the soft-optimal policy is unchanged by the size of the KL coefficient. The view that conservative updates protect against estimator noise is well-established for value-based methods, where double Q-learning (van Hasselt et al., 2016) and clipped twin Q-targets (Fujimoto et al., 2018) address overestimation bias from value-function noise. Our argument transposes this concern to the policy side and to the critic-free setting.

Our work proposes a different lever for controlling concentration in this critic-free regime through the shape of the tilting potential. *Follow the Winners* induces a polynomial concentration in the order statistic of returns rather than the exponential concentration of softmax-based methods (see Section 3). In essence, our polynomial potential approach dampens the concentration target, and the KL regularizer bounds how quickly the policy moves toward it. A complementary mechanism for damping concentration is generalized-entropy regularization, where Tsallis entropy (Lee et al., 2020) replaces the standard KL or Shannon term and yields polynomial-tailed policies in the limit. This achieves polynomial concentration through the regularizer rather than the potential, and the two approaches are formally distinct: Tsallis-regularized policies are polynomial in the advantage relative to a reference, whereas our potential is polynomial in the order statistic of within-batch returns. The two could in principle be combined, although we leave that to future work.

One asymptotic benefit of the polynomial potential over the exponential is reduced sensitivity to noisy return estimations. Firstly, the update depends on within-batch ranking rather than on the magnitude

of the empirical advantage. Secondly, the implicit risk preference induced by exponential tilting grows quadratically with the noise scale of the empirical mean, whereas under a rank-based potential it grows only linearly and saturates at the upper quantile of the noise distribution. We sketch this argument informally in Appendix D. This effect is less crucial in critic-augmented settings, where value estimation can absorb part of the noise, but it is directly relevant in critic-free RFT, where the only fallback is the batch average itself.

B.5 Cross-Entropy Method for Deep RL.

The cross-entropy method (Rubinstein, 1997, 1999; Botev et al., 2013) has been applied to reinforcement learning along two main lines, neither of which matches our usage. The first treats CEM as a black-box policy-search method, where the elite-filtering step operates over policy parameters directly. Mannor et al. (2003) introduced this for fast policy search, and Pourchot and Sigaud (2019) extend it by combining CEM with gradient-based actor-critic updates to obtain a population-based hybrid. The second uses CEM as a short-horizon planner over learned dynamics models, where the elite-filtering step operates over candidate action sequences rather than policy parameters. PETS (Chua et al., 2018), PlaNet (Hafner et al., 2019), the differentiable CEM of Amos and Yarats (2020), and the sample-efficient iCEM of Pinneri et al. (2021) all fall into this category.

Our usage differs from both lines: *Follow the Winners* instead applies CEM elite-filtering as an optimization step for an ELBO derived from a Control-as-Inference graphical model, where the optimization targets either an amortized variational inference or expectation-maximization scheme. Furthermore, we approximate the elite set for the projection step as a collection of *trajectories* drawn from the replay buffer rather than parameters or action sequences. The CEM step then acts as the bootstrap-and-project component of an approximate policy-improvement operator, with the empirical cross-entropy projection mapping the filtered trajectories back onto the policy parameters via SGD. This connects the CEM elite-filtering operator to the order-statistics analysis of Section 3, which are not present in any of the prior CEM-RL applications. A practical consequence is that our method does not require a learned dynamics model (unlike planner-style CEM) and does not maintain a population of policies (unlike CEM-RL), which keeps it compatible with the critic-free RFT setting.

Order Statistics, Subset Selection, and Genetic Algorithms. The tournament-selection scheme used by Follow the Winners (Section 3, Lemma 1) draws on classical literature in order statistics and evolutionary computation. The marginal probability of selection by tournament rank is a standard order-statistics result (David and Nagaraja, 2003; Johnson et al., 1997), and tournament selection itself is a foundational operator in genetic algorithms (GAs) (Goldberg, 1989; Goldberg and Deb, 1991) where the trade-off between tournament size and selection pressure has been studied extensively. The $K = 1, C = 2$ case of our Eq. (13) recovers binary tournament selection exactly. Relative to the GA literature, our contribution is to embed tournament selection inside a probabilistic framework: the resulting empirical occupancy is shown in Section 3 to act as an implicit tilting distribution that targets the upper quantile of returns without explicit potential evaluation. This connects the GA-style operator to the cross-entropy method for optimization (Rubinstein, 1997, 1999; Botev et al., 2013) discussed in the previous subsection. When $C = N$, our scheme reduces to standard CEM elite-filtering applied to the buffer, making the GA-style and CEM-style perspectives literally equivalent in this limit. A separate but adjacent thread is sampling without replacement from a Plackett-Luce distribution via the Gumbel-top-k trick (Gumbel, 1954; Kool et al., 2019b), which we use to sample full PL rankings efficiently.

C GRPO and DPO Primer

For self-containedness, we shortly reintroduce the key components of GRPO and DPO in our notation, so that any reader jumping into our paper has them available.

GRPO. Group-Relative Policy Optimization (Shao et al., 2024) is a critic-free policy-gradient method built on PPO (Schulman et al., 2017). For each prompt it samples a group of N trajectories from the current policy, scores their returns $G_1, \dots, G_N$, and replaces the learned value baseline of actor-critic methods with the group statistics through the advantage $A_i = (G_i - \bar{G})/\sigma_G$, where $\bar{G}$ and σ_G are the group mean and standard deviation. The policy maximizes the PPO-style clipped

objective with a KL penalty toward a reference policy,

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\Big[\tfrac{1}{N}\sum_{i=1}^{N} \min\big(r_i(\theta)A_i,\ \text{clip}(r_i(\theta), 1\pm\epsilon)A_i\big)\Big] - \lambda\, D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\theta_{ref}}\big), \tag{8}$$

with importance ratio $r_i(\theta) = \pi_\theta(H^{(i)})/\pi_{\theta_{\text{old}}}(H^{(i)})$. Trajectories above the group mean are reinforced and those below are suppressed, which makes the update risk-neutral with respect to the return distribution but sensitive to its scale. The latter is also why variants like Dr. GRPO (Liu et al., 2025b) remove the standard-deviation normalization, since it undercredits good trajectories in high-variance groups and overcredits nearly indistinguishable ones in low-variance groups. Many further GRPO variants exist, as covered in Appendix B.

DPO. Direct Preference Optimization (Rafailov et al., 2023) learns from pairwise preferences instead of raw returns. It assumes a Bradley–Terry preference model (from the RLHF framework), in which a latent reward r generates preferences between two trajectories with probability $\Pr(H^{(w)} \succ H^{(l)}) = \sigma\big(r(H^{(w)}) - r(H^{(l)})\big)$. Substituting the closed-form solution of the KL-regularized RL objective, $r(H) = \beta \ln \frac{\pi_\theta(H)}{\pi_{\theta_{ref}}(H)}$, eliminates the reward model and yields the loss

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(H^{(w)},\, H^{(l)})}\Big[\ln \sigma\Big(\beta \ln \tfrac{\pi_\theta(H^{(w)})}{\pi_{\theta_{ref}}(H^{(w)})} - \beta \ln \tfrac{\pi_\theta(H^{(l)})}{\pi_{\theta_{ref}}(H^{(l)})}\Big)\Big], \tag{9}$$

which increases the likelihood of the preferred trajectory $H^{(w)}$ relative to the dispreferred $H^{(l)}$, anchored to the reference policy $\pi_{\theta_{ref}}$. In our experiments, preferences are induced by comparing sampled returns, and Appendix E.5 recovers both objectives top-down from the CAI framework.

D Proofs

D.1 Rao-Blackwell Argument and GRPO upper-bound

Proof of Proposition 1. Fix a trajectory H_T^n and consider the conditional law of $\mathcal{B}_N$ under $Q(\mathcal{B}_N) = \prod_{j=1}^{N} q(H_T^j)$ given that H_T^n is held fixed. Under this conditional law, the remaining $N-1$ batch elements are redrawn independently from q (or replay buffer). By definition,

$$\beta_n(H_T^n) = \mathbb{E}_{Q(\mathcal{B}_N)}[\alpha_n(\mathcal{B}_N) \mid H_T^n]. \tag{10}$$

Now let $\mathcal{B}_N^{(1)}, \ldots, \mathcal{B}_N^{(M)}$ be i.i.d. conditional batch completions around the fixed trajectory H_T^n. Then $\alpha_n(\mathcal{B}_N^{(m)})$ is an i.i.d. sequence, so by the law of large numbers,

$$\frac{1}{M}\sum_{m=1}^{M} \alpha_n(\mathcal{B}_N^{(m)}) \xrightarrow[M\to\infty]{} \mathbb{E}_{Q(\mathcal{B}_N)}[\alpha_n(\mathcal{B}_N) \mid H_T^n] = \beta_n(H_T^n). \tag{11}$$

Hence $\alpha_n(\mathcal{B}_N)$ is an unbiased Monte Carlo estimator of the per-sample Rao–Blackwell coefficient under conditional batch completion.

The Jensen bound follows by upper-bounding the coefficient α_n,

$$\alpha_n(\mathcal{B}_N) = \ln g_n - \ln \sum_{k=1}^{N} g_k \le \ln g_n - \frac{1}{N}\sum_{k=1}^{N} \ln g_k - \ln N \le \ln g_n - \frac{1}{N}\sum_{k=1}^{N} \ln g_k. \tag{12}$$

Then, plugging in the exponential potential over the cardinal returns $g_n = \exp G(H_T^n)$, we obtain $G(H_T^n) - \frac{1}{N}\sum_{k=1}^{N} G(H_T^k)$ for the right-hand side, which is exactly GRPO's group averaged baseline. □

D.2 Probability of Selection under Elite-Filtering

Lemma 1 (Elite-Filtering Marginal)**.** *Consider a buffer of N trajectories with a total ranking, where $r = 1$ denotes the highest-return trajectory. An elite-filtering step draws a subset of size C uniformly without replacement from the buffer and retains the top-K ranked trajectories as elites. The normalized marginal occupancy of the trajectory at rank r among the retained elites is*

$$\Pr(r \mid N, C, K) = \frac{1}{K\binom{N}{C}} \sum_{k=1}^{K} \binom{r-1}{k-1}\binom{N-r}{C-k}, \tag{13}$$

which sums to 1 over $r \in \{1, \ldots, N\}$.

Proof. **Step 1: Decompose by elite position.** Item r is retained iff it is included in the sampled subset and lies among the top-K ranks within that subset. It occupies exactly elite position $k \in \{1, \dots, K\}$ iff exactly $k-1$ of the other sampled items have rank $< r$ and exactly $C-k$ have rank $> r$. These events are mutually exclusive across k, so

$$\Pr(\text{item } r \text{ retained}) = \sum_{k=1}^{K} \Pr(\text{item } r \text{ at elite position } k).$$

Step 2: Apply the multivariate hypergeometric distribution. The buffer is partitioned into three groups: $\{r\}$ (size 1), the $r-1$ better items, and the $N-r$ worse items. By the multivariate hypergeometric distribution (Johnson et al., 1997, Ch. 6), the probability of drawing exactly 1 item from $\{r\}$, exactly $k-1$ from the $r-1$ better items, and exactly $C-k$ from the $N-r$ worse items in a uniform draw of C items without replacement is,

$$\Pr(\text{item } r \text{ occupies position } k) = \frac{\binom{1}{1}\binom{r-1}{k-1}\binom{N-r}{C-k}}{\binom{N}{C}} = \frac{\binom{r-1}{k-1}\binom{N-r}{C-k}}{\binom{N}{C}}.$$

Summing over $k = 1, \dots, K$ gives the unnormalized marginal $\sum_{k=1}^{K} \binom{r-1}{k-1}\binom{N-r}{C-k} / \binom{N}{C}$.

Step 3: Normalization. Summing over all buffer items and exchanging the order of summation,

$$\sum_{r=1}^{N}\sum_{k=1}^{K}\binom{r-1}{k-1}\binom{N-r}{C-k} = \sum_{k=1}^{K}\sum_{r=1}^{N}\binom{r-1}{k-1}\binom{N-r}{C-k} = \sum_{k=1}^{K}\binom{N}{C} = K\binom{N}{C},$$

where the inner sum follows from the Vandermonde identity $\sum_{r=1}^{N} \binom{r-1}{k-1}\binom{N-r}{C-k} = \binom{N}{C}$ (Graham et al., 1994, Ch. 5). Hence the unnormalized marginal sums to K over all r, and dividing by K yields a proper probability distribution summing to 1, confirming our result in Eq. (13). □

Boundary Cases. A few boundary cases are immediate. For $K = 1$, only the $k = 1$ term survives, giving $\Pr(r \mid N, C, 1) = \binom{N-r}{C-1} / \binom{N}{C}$, which is strictly decreasing in r and satisfies $\Pr(1 \mid N, C, 1) = C/N$. For $K = C$, Vandermonde's identity yields $\sum_{k=1}^{C} \binom{r-1}{k-1}\binom{N-r}{C-k} = \binom{N-1}{C-1}$, so $\Pr(r \mid N, C, C) = 1/N$ for all r, recovering uniform sampling.

Remark. Variations of the Marginal & Duplicates in Elite-Filtering.

Lemma 1 assumes trajectories in the buffer are distinct and that each elite-filtering step draws C items uniformly without replacement from the full buffer of size N, with repeated steps drawn independently. Duplicate trajectories, sampling with replacement within a draw, or sampling without replacement across repeated elite-filtering steps would slightly alter the marginal probabilities. We include Lemma 1 mainly to illustrate the induced tilting distribution underlying our policy optimization procedure; a full treatment of all such variants is outside the scope of this work.

D.3 Risk Offsets across Return Distributions

We compare the implicit risk preferences of the algorithms when the return $G(H_T)$ is genuinely stochastic given the observable trajectory, rather than reduced to its conditional mean as in Eq. (2). The noise may be aleatoric or epistemic, and the critic-free setting has no value function to fall back on as a surrogate for the mean return. We compute the risk offset of each algorithm for four concrete return families under a single recipe, summarized in Table 1.

Setup. Each buffer entry carries a trajectory H_T^i and a return $G_i \sim P(\cdot \mid H_T^i)$, drawn independently across entries. Write $\mu(H_T) = \mathbb{E}[G \mid H_T]$ and $\sigma^2(H_T) = \mathrm{Var}[G \mid H_T]$. When we resample data from the buffer, we treat sampled returns as i.i.d. from the population marginal F_0 induced by the buffer's occupancy (density f_0 where the return family is continuous). The buffer's empirical distribution estimates F_0 up to finite-sample error, and we assume for methods like GRPO that the buffer is simply refreshed at each iteration to fit within this framing. The ordinal operators compare returns, and discrete families produce ties, which we break uniformly at random. This tie-breaking is

captured by the randomized grade of a return value g, $U(g) = \Pr(G' < g) + V \cdot \Pr(G' = g)$ with $G' \sim F_0$ and $V \sim \mathrm{Unif}(0,1)$ refreshed at each comparison and $U(g) \in [0,1]$. In words, the grade is the fraction of buffer mass strictly below g, plus a uniformly random position within the tie mass at g, so comparing returns with tie coin-flips is the same as comparing grades. For continuous families the grade reduces to the buffer quantile $F_0(g)$, and the grade of a buffer draw is $\mathrm{Unif}(0,1)$.

Every batch-level potential $\Phi(\mathcal{B}_N)$ below reduces to a per-entry selection weight

$$w_\Phi(H_T) \;=\; \mathbb{E}_{G|H_T}[\varphi(G)], \tag{14}$$

with φ an increasing single-return utility. The reduction holds one entry H_T fixed and marginalizes the remaining batch members over the buffer (exactly like the Rao–Blackwell conditioning of Proposition 1). For example take the MNL/softmax, the batch weight $e^{G_n/\tau}/\sum_k e^{G_k/\tau}$ has a denominator shared by the whole batch that concentrates around its mean over batch completions, so to first order this reduces to w_Φ with $\varphi(g) = e^{g/\tau}$. The induced utilities are:

- **MNL-exp** (softmax on cardinal returns, temperature τ): $\varphi(g) = e^{g/\tau}$, and $\tau = 1$ recovers the potential $g_n = \exp G$ of Proposition 1.
- **GRPO**: The Jensen surrogate of the Rao-Blackwellized MNL-exp (Proposition 1) which scores each entry by the group-mean advantage, so $\varphi(g) = g/\tau$.
- **DPO** (pairwise): the winner of a pair is the higher grade in randomly drawn comparisons. Thus, an entry with grade $U(g)$ gives $\varphi(g) = \mathbb{E}_V[U(g)]$, the win rate averaged over ties.
- **FTW top-K/C** (ranks): the entry is retained iff at most $K-1$ of its $C-1$ competitors beat it, and each does so independently with probability $1-U(g)$, giving $\varphi(g) = \mathbb{E}_V[B_{C,K}(U(g))]$ with $B_{C,K}(u) = \Pr(\mathrm{Bin}(C-1, 1-u) \le K-1)$, which matches Lemma 1 in the large-buffer limit up to the entry-independent inclusion factor.
- **FTW softmax-on-ranks** (temperature τ): $\varphi(g) = \mathbb{E}_V[e^{U(g)/\tau}]$, the Boltzmann weight on ranks (determined through sorting returns).

Definition 1 (Certainty equivalent and risk offset)**.** *Let φ be an increasing utility. Define the certainty equivalent $J_\varphi(H_T) \triangleq \varphi^{-1}\big(\mathbb{E}_{G|H_T}[\varphi(G)]\big)$ and the risk offset as $\mathcal{R}_\varphi(H_T) \triangleq J_\varphi(H_T) - \mu(H_T)$.*

A few remarks. First, $\mathcal{R}_\varphi \neq 0$ requires a non-degenerate return law and a non-affine φ. Second, as the noise vanishes, $G \to \mu(H_T)$ and every cell of Table 1 gives $\mathcal{R} = 0$. Third, an affine utility commutes with the expectation, so $J_\varphi = \mu$ and $\mathcal{R}_\varphi = 0$ exactly, which settles the GRPO row in every column of Table 1. Then finally, the score can be ambiguous. For example, take two buffers with values $\{0, 3, 10\}$ and $\{0, 7, 10\}$: the grade would be equal for items 3 and 7 despite taking on different actual values. In words, ordinal operators resolve only the ordering of returns and not their values, so the inverse φ^{-1} covers an interval rather than a single number. In these cases, we write $0^\dagger$ whenever the operator still ranks entries properly by their mean return, to distinguish an inexpressible offset from a computed zero.

To derive generic risk profiles $\mathcal{R}$ for the algorithms/method variations on varying noise distributions, we first layout a general starting point in the following lemma. This is followed by a proposition where we derive each risk profile of interest.

Lemma 2 (Theory recipe)**.** *Let $w_\Phi = \mathbb{E}_{G|H_T}[\varphi(G)]$ and let $\sigma^2 = \mathrm{Var}[G \mid H_T]$ as in the setup.*

(i) For $\varphi(g) = e^{g/\tau}$, the certainty equivalent is the scaled log moment-generating function $J_\varphi = \tau \ln \mathbb{E}_{G|H_T}[e^{G/\tau}]$, exactly.

(ii) For φ twice differentiable near μ and $\sigma^2 \to 0$ we get the certainty-equivalent expansion,

$$J_\varphi \;=\; \mu + \frac{\sigma^2}{2}\frac{\varphi''(\mu)}{\varphi'(\mu)} + o(\sigma^2), \tag{15}$$

with Arrow–Pratt coefficient φ''/φ' (Pratt, 1964).

(iii) Grade utilities depend on returns only through $U(g)$, so both FTW and DPO are invariant to any strictly increasing transformation of returns.

Proof. (i) Simply invert the exponential. (ii) Take the Taylor expansion for $\varphi(G)$ around μ inside the expectation, the linear term vanishes since $\mathbb{E}[G-\mu]=0$, leaving $w_\Phi = \varphi(\mu) + \frac{1}{2}\sigma^2\varphi''(\mu) + o(\sigma^2)$, which works when the higher central moments are $o(\sigma^2)$. Then recall that $\varphi(J_\varphi) = w_\Phi$. For $\varphi'(\mu) \neq 0$, expanding $\varphi(J_\varphi)$ around μ gives $\varphi(\mu) + \varphi'(\mu)(J_\varphi - \mu) + o(\sigma^2)$. Simple rearrangement, cancelling $\varphi(\mu)$ and dividing by $\varphi'(\mu)$, gives the claim. (iii) $U(g)$ depends on g only through the events $\{G' < g\}$ and $\{G' = g\}$, which any strictly increasing transformation of returns preserves. □

Proposition 3 (Risk offsets per return family)**.** *For each distribution (column) in Table 1, denote the conditional return law as* $\mathrm{Law}(G_t \mid H_T)$ *and assume that the buffer marginal* F_0 *bootstraps from this parametric law, writing the buffer's implicit parameters as the subscript '0', so that* $F_0 = \mathcal{N}(\mu_0, \sigma_0^2)$ *in the Gaussian column, and likewise* p_0*,* λ_0*,* θ_0 *for the other columns. Then the risk offset* $\mathcal{R}_\varphi(H_T)$ *of each operator is as given in Table 1. Cells are exact unless marked* $\approx$ *(a normal approximation, of the Poisson family or of the elite threshold) or written with an explicit* $o(\sigma^2)$ *remainder.*

Table 1: Risk offsets $\mathcal{R} = J - \mu$ per operator (rows) and return family (columns) ordered from least aggressive (top) to most aggressive (bottom). Sparse rewards use scaled returns m and success rate p, such that $\mu = mp$ allowing the mean to be invariant to larger or smaller p for properly chosen m. Similarly, counts use $G = m \cdot \mathrm{Poi}(\lambda)$, so $\mu = m\lambda$ and $\sigma^2 = m^2\lambda$. The exponential family has rate θ and mean $1/\theta$ and requires the MNL-exp cell to satisfy $\theta\tau > 1$ to be defined. GRPO's zeros are exact, while $0^\dagger$ marks cells whose ranking is mean-consistent but whose offset is not expressible in return units (the certainty equivalent is interval-valued). The constants $a_{C,K}$ and $s^2_{C,K}$ are the mean and variance of the K-th largest value among $C-1$ buffer returns standardized to mean 0 and variance 1, $t = \mu_0 + \sigma_0 a_{C,K}$ denotes the elite threshold, H_n and $H_n^{(2)}$ are the harmonic numbers of first and second order, and f_0 is the density of the population marginal F_0.

	Gaussian $\mathcal{N}(\mu,\sigma^2)$	Sparse $m\,\mathrm{Ber}(p)$	Counts $m\,\mathrm{Poi}(\lambda)$	Exponential(θ)
GRPO	0	0	0	0
DPO	$(\mu-\mu_0)\Big(\frac{\sigma_0}{\sqrt{\sigma_0^2+\sigma^2}}-1\Big)$	$0^\dagger$	$\approx$ Gaussian cell, $\sigma^2 = m^2\lambda$	$\frac{1}{\theta_0}\ln\big(1+\frac{\theta_0}{\theta}\big)-\frac{1}{\theta}$
FTW top-K/C	$\approx(\mu-t)\Big(\frac{\sigma_0 s_{C,K}}{\sqrt{\sigma^2+\sigma_0^2 s^2_{C,K}}}-1\Big)$	$0^\dagger$	$\approx$ Gaussian cell, $\sigma = m\sqrt{\lambda}$	$\approx$ Gaussian cell, $(a_{C,K}, s_{C,K})$ via $H, H^{(2)}$
FTW smx-ranks	$\frac{\sigma^2}{2}\Big(\frac{f_0(\mu)}{\tau}-\frac{\mu-\mu_0}{\sigma_0^2}\Big)+o(\sigma^2)$	$0^\dagger$	$\approx$ Gaussian cell	$\frac{\sigma^2\theta_0}{2}\Big(\frac{e^{-\theta_0\mu}}{\tau}-1\Big)+o(\sigma^2)$
MNL-exp	$\frac{\sigma^2}{2\tau}$	$\tau\ln(1-p+pe^{m/\tau})-mp$	$\tau\lambda(e^{m/\tau}-1)-m\lambda$	$-\tau\ln\big(1-\frac{1}{\theta\tau}\big)-\frac{1}{\theta}, \theta\tau>1$

Proof. **GRPO row.** Conditioned on H_T, the batch mean averages the entry's own return with its $N-1$ competitors, $\mathbb{E}[\bar{G} \mid H_T] = \frac{1}{N}\mu(H_T) + \frac{N-1}{N}\mu_0$, so $\mathbb{E}[G-\bar{G} \mid H_T] = (1-\frac{1}{N})\,(\mu(H_T)-\mu_0)$. This is linear in $\mu(H_T)$ with a shared constant. Marginally over the buffer the advantage averages to zero, but the conditional average determines the ranking in again a linear fashion. Thus $\mathcal{R}=0$.

(Sparse) Bernoulli column. On the support $\{0, m\}$ any utility gives $w_\Phi = (1-p)\,\varphi(0) + p\,\varphi(m)$, which is affine in p. Thus, every operator ranks by the mean mp. The grade utilities are flat between any other value between the two payoffs and admit no unique J_φ, so their cells record the mean-rank as $0^\dagger$ rather than actually zero. The next step shows the only exception in the MNL-exp cell.

MNL-exp row. By Lemma 2(i), $J_\varphi = \tau \ln \mathbb{E}_{G|H_T}[e^{G/\tau}]$, this is the moment generating function and has a standard closed form for each of the considered distributions: Gaussian $e^{\mu/\tau+\sigma^2/2\tau^2}$, scaled Bernoulli $1-p+pe^{m/\tau}$, scaled Poisson $e^{\lambda(e^{m/\tau}-1)}$, exponential $(1-\frac{1}{\theta\tau})^{-1}$ for $\theta\tau>1$ and infinite otherwise. Subtracting the means gives the cells. All four reduce to $\sigma^2/2\tau$ at small noise. The only difference for MNL-exp in the (sparse) Bernoulli column is the $e^{m/\tau}$ factor, the weight is sensitive to the return's value, whereas the ordinal operators depend only on its rank.

DPO row. At a sure return g, the utility is an independent average over opponents, $\varphi(g) = \mathbb{E}_V[U(g)] = \Pr(G'<g) + \frac{1}{2}\Pr(G'=g)$ with $G' \sim F_0$, i.e., the win rate of g against any one buffer draw. Then, averaging over the entry's own return gives the win rate of the noisy variant, $w_\Phi = \mathbb{E}_{G|H_T}[\varphi(G)] = \Pr(G'<G) + \frac{1}{2}\Pr(G'=G)$ with G' independent of G. For continuous values $\Pr(G'=G)=0$.

Gaussian: The weight is equivalently $w_\Phi = \Pr(G'<G) = \Pr(G-G'>0)$. The difference of independent Gaussians is Gaussian, $G-G' \sim \mathcal{N}(\mu-\mu_0, \sigma^2+\sigma_0^2)$, so standardizing gives $w_\Phi = \Phi\big((\mu-\mu_0)/\sqrt{\sigma^2+\sigma_0^2}\big)$, with Φ here the standard normal CDF. The certainty equivalent

solves $\varphi(J_\varphi) = w_\Phi$, where for the sure return J_φ only the opponent's draw is random. Thus, its utility is the quantile $\varphi(J_\varphi) = \Phi\big((J_\varphi - \mu_0)/\sigma_0\big)$. Applying Φ^{-1} to both sides and solving for J_φ gives $J_\varphi = \mu_0 + \sigma_0(\mu - \mu_0)/\sqrt{\sigma^2 + \sigma_0^2}$, and subtracting μ factors into the cell $\mathcal{R} = (\mu - \mu_0)\big(\sigma_0/\sqrt{\sigma_0^2 + \sigma^2} - 1\big)$.

Exponential: The Gaussian steps carry over with two substitutions, namely the expressions for the win rate and the CDF. The win rate becomes $w_\Phi = \Pr(G > G') = \int_0^\infty e^{-\theta x}\, \theta_0 e^{-\theta_0 x}\, dx = \theta_0/(\theta + \theta_0)$. At the sure return J_φ only the opponent is random, so the utility becomes the CDF, $\varphi(J_\varphi) = \Pr(G' < J_\varphi) = 1 - e^{-\theta_0 J_\varphi}$. Setting this equal to w_Φ gives $e^{-\theta_0 J_\varphi} = \theta/(\theta + \theta_0)$, hence $J_\varphi = \frac{1}{\theta_0}\ln(1 + \theta_0/\theta)$, and subtracting the mean $1/\theta$ gives the table result.

Counts: Let $G = mk$ with k the count. Averaging the win-and-tie probability over the opponents gives the exact weight $w_\Phi = \sum_k \mathrm{Poi}_\lambda(k)\big[F_{\lambda_0}(k-1) + \frac{1}{2}\mathrm{Poi}_{\lambda_0}(k)\big]$, where $F_{\lambda_0}(k-1) = \Pr(G' < mk)$ is the Poisson CDF and the scale m cancels due to the invariance as per Lemma 2(iii). This sum has no closed form, and the utility is a step function. Thus the table only reports the Gaussian approximation with moment-matched laws $\mathcal{N}(m\lambda, m^2\lambda)$ and $\mathcal{N}(m\lambda_0, m^2\lambda_0)$, which is valid for large λ and λ_0.

FTW top-K/C row. An entry is retained iff its return exceeds the K-th largest of its $C-1$ competitors. To formalize this, we will standardize an entry's returns according to its own law $G = \mu + \sigma Z$ with $Z \triangleq (G - \mu)/\sigma$. This assumes nothing but gives us Z with mean 0, variance 1, while preserving the underlying family (Table 1). Each of the $C - 1$ competitors are then also standardized but by the buffer's (μ_0, σ_0). This holds exactly for the continuous columns, since every Gaussian standardizes to $\mathcal{N}(0, 1)$. For the exponential law we know that the mean and standard deviation are both $1/\theta$, so $Z = (G - 1/\theta)/(1/\theta) = \theta G - 1$, and $\theta G \sim \mathrm{Exp}(1)$ for any rate θ. This recipe fails for the discrete columns since a linear standardization can only fix the first two moments and their third and higher moments will still depend on the parameter.

Now write the random threshold as $T \triangleq \mu_0 + \sigma_0 Z_{(K)}$, the K-th largest competitor return, where $Z_{(K)}$ is the K-th largest of the $C - 1$ standardized competitor draws. Retention is the event $G \geq T$, so the utility is the threshold's CDF, $\varphi(g) = \Pr(T \leq g) \triangleq F_T(g)$, and the certainty equivalent solves $F_T(J_\varphi) = \mathbb{E}_{G|H_T}[F_T(G)]$. For the pairwise case $(C, K) = (2, 1)$ where T is a single buffer draw we obtain exactly the DPO setting. In every other case we have that the law of T is known exactly, $F_T(g) = B_{C,K}(F_0(g))$ with $B_{C,K}(u) = \Pr(\mathrm{Bin}(C - 1, 1 - u) \leq K - 1)$, but it is inconvenient and not Gaussian for $C > 2$. Moreover, $J_\varphi = F_T^{-1}(\mathbb{E}[F_T(G)])$ has no closed form. For larger C we therefore approximate T by a Gaussian, and any residual error caused by the higher moments of $Z_{(K)}$ (which is typically skewed with a longer tail on the high returns) is what the table's $\approx$ marks. Writing $a_{C,K} \triangleq \mathbb{E}[Z_{(K)}]$ and $s_{C,K}^2 \triangleq \mathrm{Var}[Z_{(K)}]$ for the mean and variance of the K-th largest of $C - 1$ standardized draws, we obtain $t \triangleq \mathbb{E}[T] = \mu_0 + \sigma_0 a_{C,K}$ and $\mathrm{sd}(T) = \sigma_0 s_{C,K}$. Approximating T by a Gaussian with these two exact moments allows us to reuse the DPO steps with (μ_0, σ_0) replaced by $(t, \sigma_0 s_{C,K})$. The weight becomes $w_\Phi = \Pr(G - T \geq 0) = \Phi\big((\mu - t)/\sqrt{\sigma^2 + \sigma_0^2 s_{C,K}^2}\big)$, since $G - T \sim \mathcal{N}(\mu - t, \sigma^2 + \sigma_0^2 s_{C,K}^2)$ for independent Gaussians. At the sure return only the threshold is random, so $\varphi(J_\varphi) = \Pr(T \leq J_\varphi) = \Phi\big((J_\varphi - t)/(\sigma_0 s_{C,K})\big)$. Setting the two equal, applying Φ^{-1}, and solving gives $J_\varphi = t + \sigma_0 s_{C,K}(\mu - t)/\sqrt{\sigma^2 + \sigma_0^2 s_{C,K}^2}$, and subtracting μ factors into

$$\mathcal{R} \;=\; (\mu - t)\left(\frac{\sigma_0 s_{C,K}}{\sqrt{\sigma^2 + \sigma_0^2 s_{C,K}^2}} - 1\right). \tag{16}$$

This is the same shrinkage form as the DPO row, with the buffer mean μ_0 attractor substituted by the elite threshold t. The remaining paragraphs show the $(a_{C,K}, s_{C,K})$ pair for each distribution.

Gaussian: Due to standardization, we have that $a_{C,K}$ and $s_{C,K}$ are simply the mean and standard deviation of the K-th largest among $C - 1$ independent $\mathcal{N}(0, 1)$ draws. Neither has a closed form, but both follow from one-dimensional quadrature against the order-statistic density, $a_{C,K} = \frac{(C-1)!}{(K-1)!\,(C-1-K)!}\int_{\mathbb{R}} z\,\Phi(z)^{C-1-K}(1 - \Phi(z))^{K-1}\,\mathrm{d}\Phi(z)$, where the same integral with z^2 in the integrand gives the second moment and $s_{C,K}^2$ follows by subtracting $a_{C,K}^2$. Concretely, this gives

example values $(a, s) = (0, 1)$ at $(2, 1)$, $(0.85, 0.75)$ at $(4, 1)$, $(0, 0.67)$ at $(4, 2)$, $(-0.85, 0.75)$ at $(4, 3)$, and $(1.35, 0.63)$ at $(8, 1)$, which we verified numerically. Note that $(4, 2)$ shares the attractor $t = \mu_0$ with $(2, 1)$, the median of three symmetric draws has mean zero, but its smaller $s_{C,K}$ makes the shrinkage factor smaller at every σ, so the pull toward μ_0 is stronger. At $(C, K) = (2, 1)$ the threshold is a single Gaussian buffer draw, so nothing is approximated, $t = \mu_0$, $s_{2,1} = 1$, and the cell coincides with the DPO row exactly. For very large C the mean grows like $\sqrt{2\ln(C/K)}$, in the sense that $a_{C,K}/\sqrt{2\ln(C/K)} \to 1$ as $C \to \infty$ (David and Nagaraja, 2003, Ch. 10).

Exponential: Both constants are closed-form. The K-th largest of $C - 1$ i.i.d. $\mathrm{Exp}(1)$ variables has mean $H_{C-1} - H_{K-1}$ and variance $H^{(2)}_{C-1} - H^{(2)}_{K-1}$, with $H^{(2)}_n \triangleq \sum_{i=1}^{n} i^{-2}$ (David and Nagaraja, 2003, Ch. 4), so standardizing gives $a_{C,K} = H_{C-1} - H_{K-1} - 1$ and $s^2_{C,K} = H^{(2)}_{C-1} - H^{(2)}_{K-1}$ at any finite C. The cell plugs these into the Gaussian coefficient, and it stays approximate despite the exact constants because exponential order statistics are more strongly skewed than normal ones, so the Gaussian law of T is coarser here. At $(2, 1)$ the pair is again $(0, 1)$ which follows DPO.

Counts: As noted before, the standardized shape of $m\,\mathrm{Poi}(\lambda)$ still depends on λ, so the location-scale step does not hold exactly for this family. We therefore replace the Poisson laws by their moment-matched normals, $\mathcal{N}(m\lambda, m^2\lambda)$ for the entry and $\mathcal{N}(m\lambda_0, m^2\lambda_0)$ for the buffer, which is accurate when λ and λ_0 are large. Under this substitution the Gaussian cell applies verbatim, with $\sigma = m\sqrt{\lambda}$ and the Gaussian pair $(a_{C,K}, s_{C,K})$. Although it is possible to compute the exact moments of the Poisson elite threshold, the cell's error comes from treating T as Gaussian rather than from the two constants, so plugging exact constants into the approximate formula would not make the result exact. For reference, this would give $a_{C,K} = (\mathbb{E}[k_{(K)}] - \lambda_0)/\sqrt{\lambda_0}$, where the mean follows from the tail sum $\mathbb{E}[k_{(K)}] = \sum_{j \geq 0} \Pr(\mathrm{Bin}(C - 1, q_j) \geq K)$ with $q_j = \Pr(\mathrm{Poi}(\lambda_0) > j)$ and $k_{(K)}$ the K-th largest of $C - 1$ i.i.d. $\mathrm{Poi}(\lambda_0)$ draws.

FTW softmax-on-ranks row. Start from the utility $\varphi(g) = \mathbb{E}_V[e^{U(g)/\tau}]$. In the continuous columns ties have probability zero, so the grade is the plain quantile $U(g) = F_0(g)$ and $\varphi(g) = e^{F_0(g)/\tau}$. For Lemma 2(ii) we differentiate, $\varphi' = \frac{f_0}{\tau}\,\varphi$ and $\varphi'' = \big(\frac{f_0^2}{\tau^2} + \frac{f_0'}{\tau}\big)\varphi$, so the Arrow–Pratt coefficient is $\varphi''/\varphi' = f_0/\tau + f_0'/f_0$ and the lemma gives $\mathcal{R} = \frac{\sigma^2}{2}\big(f_0(\mu)/\tau + (\ln f_0)'(\mu)\big) + o(\sigma^2)$. We can then insert the PDFs. *Gaussian:* $(\ln f_0)'(\mu) = -(\mu - \mu_0)/\sigma_0^2$, giving the cell. *Exponential:* $f_0(\mu) = \theta_0 e^{-\theta_0 \mu}$ and $(\ln f_0)' = -\theta_0$, giving the cell. *Counts:* the grade is a step function and the expansion cannot be applied directly. For the Poisson specifically, however, we can assume the normal limit $\mathcal{N}(m\lambda_0, m^2\lambda_0)$ and reuse the Gaussian result, with the caveat that λ_0 must be large. □

Discussion. Read top to bottom, Table 1 orders the operators by risk sensitivity, GRPO exactly neutral, the three grade operators boundedly risk-seeking, MNL-exp unboundedly so. GRPO's utility is affine, so it is exactly neutral in every column, this was also briefly shown empirically in Figure 2. It says nothing about learning stability due to exploding variance however. One takeaway from Prop. 3 is that the grade-based operators are always boundedly risk-seeking due to being scale agnostic. For FTW the attractor is the elite threshold $t = \mu_0 + \sigma_0 a_{C,K}$. Mild filters with K near C have $a_{C,K} \leq 0$ and put it at or below the buffer mean μ_0, DPO and the half filters $K = C/2$ have $a_{C,K} = 0$ and put it at the buffer mean exactly, and aggressive filters with $K \ll C$ have $a_{C,K} > 0$ and raise it above. The pull grows with the entry's own σ but saturates at the attractor, so no amount of noise scores an entry above t. MNL-exp uses the full return magnitude and is unboundedly risk-seeking, which also makes it fragile.

The Gaussian cell also predicts the σ-trap of Figure 2 (left). So long as both arms remain represented in the buffer, a noisy Gaussian distractor lands above or below the optimal mean with probability $\frac{1}{2}$ as its variance grows. Therefore, every ordinal comparison converges towards a coin flip and the distilled policy should approach a uniform distribution. Aggressive filters with $a_{C,K} > 0$ fail before this regime, because the buffer spread σ_0 grows with the distractor's noise, so already at moderate noise their threshold $t = \mu_0 + \sigma_0 a_{C,K}$ rises above the optimal mean. This excludes the near-deterministic arm more and more often, and eventually the noisy arm crowds the buffer. We show this in the supplementary results in Figure 6, where for aggressive (C, K) variants the recovered policy converges to the trap-arm.

We close with limitations of this theoretical section. Firstly, the Arrow-Pratt expansion behind the FTW softmax-on-ranks row holds in the small-noise regime $\sigma \ll \sigma_0$, which is not typically realistic,

whereas the other rows hold at every σ. Secondly, the elite threshold T is approximated by a normal to obtain a closed form, its exact law is skewed, so the FTW top-K/C cells are exact only in the pairwise Gaussian case. Thirdly, the buffer marginal is in general a mixture over conditional return laws, in practice these laws are not guaranteed homogenous. Our single-family assumption on F_0 is thus a homogeneity idealization. Lastly, on sparse or stepwise rewards the grade operators admit no unique certainty equivalent, the $0^\dagger$ cells. On these discrete rewards they are also jackpot-blind, at a fixed mean mp their weight for a rare large payoff collapses to that of a sure zero as $p \to 0$ (which implies that the jackpot m must grow), whereas MNL-exp instead diverges.

E Extended Derivation

E.1 Preliminaries

Let π_θ be a history-dependent parametric policy acting in a partially observable Markov decision process. We write the observable history at time t as $H_t = (O_0, A_0, \ldots, O_{t-1}, A_{t-1}, O_t)$, and the full observable trajectory as $H_{0:T} = (O_0, A_0, O_1, A_1, \ldots, O_T)$. The policy induces a distribution over observable trajectories by marginalizing over latent states,

$$p_{\pi_\theta}(H_{0:T}) = \int p(S_0)p(O_0 \mid S_0) \prod_{t=0}^{T-1} \pi_\theta(A_t \mid H_t)\, p(S_{t+1} \mid S_t, A_t)\, p(O_{t+1} \mid S_{t+1})\, dS_{0:T}. \quad (17)$$

Denote the set of admissible policies by $\Pi \triangleq \{\pi \mid \pi : \mathcal{H} \times \Theta \to \mathcal{P}(\mathcal{A})\}$, so that $\pi_\theta \in \Pi$.

Since rewards depend on the latent state-action process, the return is defined as

$$G(S_{0:T}, A_{0:T-1}) \triangleq \sum_{t=0}^{T} r(S_t, A_t) \in \mathbb{R}. \quad (18)$$

For notational convenience, we write $G(H_{0:T})$ for the return random variable induced by the latent state-action trajectory underlying $H_{0:T}$, i.e. $G(H_{0:T}) \sim \mathrm{Law}\left(\sum_{t=0}^{T} r(S_t, A_t) \,\middle|\, H_{0:T}\right)$. The objective is then

$$\theta^* \triangleq \arg\max_\theta \, \mathbb{E}_{p_{\pi_\theta}(S_{0:T}, H_{0:T})}[G(S_{0:T}, A_{0:T-1})]\,. \quad (19)$$

For convenience, in the infinite-horizon setting $T \to \infty$, assume that the transition kernel satisfies $\Pr(S_{t+1} = s_{absorb} \mid S_t, A_t) > 0$ for all $S_t \in \mathcal{S}$, where s_{absorb} is an absorbing state that transitions to itself with probability 1 and yields zero reward. Equivalently, one may work with a discount factor $\gamma \in [0, 1)$, which can be interpreted as an effective termination probability $p_{term} = 1 - \gamma$.

Define a binary optimality variable $\mathcal{O}_T \in \{0, 1\}$, which we use to condition the graphical model at the trajectory level. For simplicity, this corresponds to a sparse-reward setting in which rewards are evaluated only on completion. Most importantly, the optimality variable depends only on the return, so that

$$\mathcal{O}_T \perp\!\!\!\perp (S_{0:T}, H_{0:T}) \mid G(S_{0:T}, A_{0:T-1}). \quad (20)$$

We can then perform model selection by maximizing the evidence

$$\max_\theta p_{\pi_\theta}(\mathcal{O}_T = 1) = \max_\theta \int p_{\pi_\theta}(\mathcal{O}_T = 1, S_{0:T}, H_{0:T})\, dS_{0:T}\, dH_{0:T}. \quad (21)$$

For brevity, we write $\Pr(\mathcal{O}_T = 1) \equiv \Pr(\mathcal{O}_T)$, since the case $\mathcal{O}_T = 0$ is not of direct interest.

Remark. Meaning of the Optimality Variable $\mathcal{O}_T$.

The variable $\mathcal{O}_T$ serves as a soft optimality indicator at the trajectory level, and is best understood as a formal device rather than an observed quantity. Rather than directly maximizing expected return, we maximize the evidence $p_{\pi_\theta}(\mathcal{O}_T = 1)$, treating θ as a parameter of the generative model, an empirical Bayes perspective in which the policy is optimized by

maximizing the marginal likelihood of optimality. The actual values of $\mathcal{O}_T$ are never used directly; at optimality, π_θ would generate only high-return trajectories, so the outcome vector collapses to $[1, 1, \ldots, 1]$, while a suboptimal policy produces a mixture of zeros and ones reflecting the proportion of good trajectories.

KL-Regularized Policy Gradient. To derive optimization objectives, let $q_\phi(H_{0:T}) \in \Pi$ be a variational policy and lower-bound the log-evidence via Jensen's inequality,

$$\ln p_{\pi_\theta}(\mathcal{O}_T) = \ln \int p(\mathcal{O}_T \mid H_{0:T})\, p_{\pi_\theta}(H_{0:T})\, dH_{0:T}$$
$$\geq \mathbb{E}_{q_\phi}[\ln p(\mathcal{O}_T \mid H_{0:T})] - D_{\mathrm{KL}}(q_\phi(H_{0:T}) \,\|\, p_{\pi_\theta}(H_{0:T})) \triangleq \mathcal{L}(\theta, \phi). \tag{22}$$

Assuming an exponential likelihood,

$$p(\mathcal{O}_T \mid H_{0:T}) = \frac{\exp G(H_{0:T})}{Z(\pi_\theta)}, \qquad Z(\pi_\theta) = \mathbb{E}_{p_{\pi_\theta}}[\exp G(H_{0:T})]\,, \tag{23}$$

and noting that $Z(\pi_\theta)$ is constant with respect to the maximization over ϕ, we obtain the KL-regularized objective and its gradient,

$$\max_\phi \mathcal{L}(\theta, \phi) = \max_\phi \mathbb{E}_{q_\phi}[G(H_{0:T})] - D_{\mathrm{KL}}(q_\phi \,\|\, p_{\pi_\theta})\,, \tag{24}$$
$$\nabla_\phi \mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi}[G(H_{0:T})\, \nabla_\phi \ln q_\phi] - \nabla_\phi D_{\mathrm{KL}}(q_\phi \,\|\, p_{\pi_\theta})\,, \tag{25}$$

which recovers a policy-gradient objective regularized by the KL divergence between the variational policy and the model. A locally optimal policy can then be recovered through coordinate ascent by alternating until convergence:

$$\theta^* = \arg\max_\theta \mathcal{L}(\theta, \phi^*), \qquad \phi^* = \arg\max_\phi \mathcal{L}(\theta^*, \phi). \tag{26}$$

In our setting this can be implemented simply by setting $\theta^* \leftarrow \phi^*$, since both parameterize policies in Π. Under the assumptions that $\mathcal{L}$ is bounded above, each coordinate update is exact, and G is bounded (ensuring $Z(\pi_\theta) < \infty$), each step does not decrease $\mathcal{L}$, usually converging to a local optimum.

Gibbs identity. Maximizing the ELBO is equivalent to minimizing the KL divergence to the tilted posterior q^*, by the Gibbs identity

$$\mathbb{E}_{q_\phi}[G(H_{0:T})] - D_{\mathrm{KL}}(q_\phi \,\|\, p_{\pi_\theta}) = \ln Z(\pi_\theta) - D_{\mathrm{KL}}(q_\phi \,\|\, q^*)\,, \tag{27}$$
$$q^*(H_{0:T}) \triangleq \frac{p_{\pi_\theta}(H_{0:T}) \exp G(H_{0:T})}{Z(\pi_\theta)}, \tag{28}$$

so the evidence bound is tight exactly at q^*, provided the policy class contains it. Maximizing over ϕ therefore projects q^* onto the policy class, from which we can observe an alternative optimization strategy. Draws from q^* are simply draws from p_{π_θ} reweighted, or resampled, by the potential $\exp G$. The projection then becomes a cross-entropy fit on these tilted samples, which never needs a gradient of the potential. This duality between the policy gradient and the cross-entropy fit is what motivates FTW's sample-and-project approach.

E.2 Batch-Level CAI and Ordinal Potentials

We now extend the trajectory-level CAI of Appendix E.1 to inference over finite collections of trajectories, conditioned on ordinal feedback. This section develops the framework abstractly, subsequent subsections instantiate it with the Plackett–Luce model and multinomial model.

Batch-level ELBO (Eq. 4). We modify the base graphical model marginalizing not over the single trajectory H_T but a batch $\mathcal{B}_N = (H_T^1, \ldots, H_T^N)$ and the binary $\mathcal{O}_T \in \{0, 1\}$ with an ordinal observation $\mathcal{O}_T \in \mathbb{Z}_+$ or $\mathcal{O}_T \in \mathbb{S}_N$ where $\mathbb{S}_N$ is the symmetric group of size N (overloading notation) supported on rankings or winner indices, the evidence becomes $\ln p_{\pi_\theta}(\mathcal{O}_T) = \ln \mathbb{E}_{p_{\pi_\theta}(\mathcal{B}_N)}[\Phi_T(\mathcal{B}_N)]$, where $p_{\pi_\theta}(\mathcal{B}_N) = \prod_{n=1}^N p_{\pi_\theta}(H_T^n)$ by the i.i.d. batch construction (Section 3). Introducing a

variational $q \in \Pi$ at the trajectory level and lifting it to the batch as $p_q(\mathcal{B}_N) = \prod_n p_q(H_T^n)$, Jensen's inequality gives

$$\ln p_{\pi_\theta}(\mathcal{O}_T) \geq \mathbb{E}_{p_q(\mathcal{B}_N)}[\ln \Phi_T(\mathcal{B}_N)] - D_{\mathrm{KL}}(p_q(\mathcal{B}_N) \,\|\, p_{\pi_\theta}(\mathcal{B}_N)) \,. \tag{29}$$

The KL between product measures factorizes as $D_{\mathrm{KL}}(\prod_n p_q \,\|\, \prod_n p_{\pi_\theta}) = N \cdot D_{\mathrm{KL}}(p_q \,\|\, p_{\pi_\theta})$, recovering Eq. 4 from the main text. The factor N is not a tunable weight but a direct consequence of the i.i.d. batch construction, however, it can be absorbed into the KL coefficient λ in Algorithm 2 which is equivalent to adjusting the temperature of the likelihood $\ln \Phi_T(\mathcal{B}_N)$.

Ordinal feedback. We consider two cases for ordinal feedback on a batch $\mathcal{B}_N = \{H_{0:T}^{(i)}\}_{i=1}^N$:

1. **Full ranking.** $\mathcal{O}_T = \sigma \in \mathbb{S}_N$ specifies a complete permutation of $\mathcal{B}_N$. The likelihood is the Plackett–Luce (PL) model (§E.3), and methods such as DPO and LiPO operate in this regime.
2. **Winner-only.** $\mathcal{O}_T = n \in \{1, \ldots, N\} \subset \mathbb{Z}_+$ identifies a single preferred trajectory. The likelihood is the multinomial (MNL) model (§E.4), which arises as the first-stage marginal of PL, and methods such as GRPO operate in this regime.

Both yield well-defined ordinal CAI problems. The substantive question is whether ordinal feedback elicited on a sampled batch $\mathcal{B}_N$ is *globally consistent*. In other words, whether the preference structure on $\mathcal{B}_N$ agrees with an underlying total order on $\mathcal{H}$. This is established by Luce's choice axiom.

Consistency via Luce's Choice Axiom. Both the full-ranking PL likelihood and the winner-only MNL marginal inherit independence of irrelevant alternatives (IIA) from the PL family. For any two trajectories $a, b \in \mathcal{B}_N$, define the pairwise preference probability

$$p(a \succ b \mid \mathcal{B}_N) \;\equiv\; p(\mathcal{O}_T = a \mid \mathcal{O}_T \in \{a, b\}, \mathcal{B}_N). \tag{30}$$

Proposition 4 (Luce's choice axiom (Luce, 1959))**.** *Under the PL model with positive weights g, pairwise preferences are independent of the enclosing batch $\mathcal{B}_N$: for any $a, b \in \mathcal{B}_N$,*

$$p(a \succ b \mid \mathcal{B}_N) \;=\; \frac{g(a)}{g(a) + g(b)}. \tag{31}$$

Proof. By Bayes' rule, and using that $\{\mathcal{O}_T = a\}$ and $\{\mathcal{O}_T = b\}$ are mutually exclusive,

$$p(\mathcal{O}_T = a \mid \mathcal{O}_T \in \{a, b\}, \mathcal{B}_N) = \frac{p(\mathcal{O}_T = a \mid \mathcal{B}_N)}{p(\mathcal{O}_T = a \mid \mathcal{B}_N) + p(\mathcal{O}_T = b \mid \mathcal{B}_N)} \tag{32}$$

$$= \frac{g(a)/Z}{g(a)/Z + g(b)/Z} \;=\; \frac{g(a)}{g(a) + g(b)}, \tag{33}$$

where $Z = \sum_{i=1}^N g(H_{0:T}^{(i)})$ cancels. □

Since the right-hand side is independent of $\mathcal{B}_N$, ordinal feedback elicited on any batch is a consistent marginal of an underlying total ordering on $\mathcal{H}$. This has two operational implications.

For the PL likelihood (full ranking), IIA guarantees that the relative probability of any two rankings σ, σ' that agree on their top-k elements is independent of the remaining elements. Thus, subset rankings are consistent permutations of the full-domain ordering, so optimizing on any batch yields a gradient aligned with the full-domain objective.

For the MNL likelihood (winner-only), IIA guarantees that the winner identity is stable regardless of which other trajectories appear in $\mathcal{B}_N$: if $H_{0:T}^{(n)}$ is the best trajectory globally, it wins in any batch containing it. The winner-only model is thus a principled marginal of the full PL model.

Both guarantees mean that Monte Carlo approximation over sampled batches $\mathcal{B}_N^{(1)}, \ldots, \mathcal{B}_N^{(M)} \sim p_{\pi_\theta}$ is well-posed for either likelihood choice. This justifies the per-minibatch filtering operation of our algorithm FTW (Section 3) since per-batch rankings under either potential are consistent estimators of the full-domain ranking, IIA ensures the elite set is unbiased in expectation.

E.3 The Plackett–Luce Model

The Plackett–Luce (PL) model defines a probability distribution over rankings of a finite set of alternatives. Given a batch $\mathcal{B}_N = \{H_{0:T}^{(i)}\}_{i=1}^N$, each trajectory is assigned a positive weight $g(H_{0:T}^{(i)}) > 0$ through a weight function $g : \mathcal{H} \to \mathbb{R}_{>0}$ (for instance, $g(H_T^{(i)}) = \exp G(H_T^i)$). The probability of observing a complete ranking $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(N))$ is given by sequential elimination,

$$p(\mathcal{O}_T = \sigma \mid \mathcal{B}_N) \;=\; \prod_{n=1}^{N} \frac{g\Big(H_{0:T}^{(\sigma(n))}\Big)}{\sum_{k=n}^{N} g\Big(H_{0:T}^{(\sigma(k))}\Big)}. \tag{34}$$

Here $\sigma(1)$ denotes the top-ranked (winner) trajectory and $\sigma(N)$ the bottom-ranked. The first-stage marginal, which is the probability that trajectory n is ranked first, follows by summing over all rankings consistent with $\sigma(1) = n$, yielding the familiar softmax form

$$p(\mathcal{O}_T = n \mid \mathcal{B}_N) \;=\; \frac{g\Big(H_{0:T}^{(n)}\Big)}{\sum_{i=1}^{N} g\Big(H_{0:T}^{(i)}\Big)}. \tag{35}$$

If the generating policy is already optimal: $p(\mathcal{O}_T = n \mid \mathcal{B}_N) = 1/N$, and the ranking is uniform. The weight function g is a modeling choice that determines the inductive bias of the likelihood; independent of this choice, the PL model satisfies IIA (Proposition 4), so pairwise preferences are consistent across batches.

E.4 The Multinomial (Winner) Model

The winner-only model takes $p(\mathcal{O}_T = n \mid \mathcal{B}_N)$ as the probability that $H_{0:T}^{(n)}$ beats all others in $\mathcal{B}_N$,

$$p(\mathcal{O}_T = n \mid \mathcal{B}_N) \;\equiv\; p\Big(H_{0:T}^{(n)} \succ H_{0:T}^{(k)},\ \forall k \neq n \,\Big|\, \mathcal{B}_N\Big) \;=\; \frac{g^{(n)}}{\sum_{j=1}^{N} g^{(j)}}, \tag{36}$$

which is exactly the first-stage PL marginal (Eq. 35). The first-stage marginal of the full PL ranking model coincides with the standalone MNL choice model, so optimizing on winner-only feedback is principled and inherits the global consistency guarantees of PL.

Exchangeability. Since the batch construction $p_{\pi_\theta}(\mathcal{B}_N) = \prod_n p_{\pi_\theta}(H_T^n)$ is permutation-invariant, the batch indices $\{1, \ldots, N\}$ are a priori exchangeable. Conditioning on a specific winner index n is therefore arbitrary, and we instead work with the average over all N possible winner identities,

$$\ell(\mathcal{O}_T \mid \mathcal{B}_N) \;\triangleq\; \frac{1}{N} \sum_{n=1}^{N} p(\mathcal{O}_T = n \mid \mathcal{B}_N), \tag{37}$$

which substitutes into the ELBO (Eq. 29) to give the averaged form,

$$\widehat{\mathcal{L}}(\theta, \phi \mid \mathcal{B}_N) \;=\; \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{p_q(\mathcal{B}_N)}[\ln p(\mathcal{O}_T = n \mid \mathcal{B}_N)] - N \cdot D_{\mathrm{KL}}(p_q \,\|\, p_{\pi_\theta}). \tag{38}$$

This is the objective from which the per-trajectory MNL policy gradient is derived in §E.5.

E.5 Policy Gradients and Recovery of GRPO and DPO

We now derive the policy gradient under each likelihood with the Boltzmann weight function $g(H_{0:T}) = \exp G(H_{0:T})$, and use these gradients to recover GRPO and DPO as special cases. The discussion below derives the gradient directly from the per-trajectory log-likelihood. Throughout, we adopt the convention that $\sigma(1)$ denotes the top-ranked (winner) trajectory and $\sigma(N)$ the bottom-ranked.

Policy gradient under the full PL model. Plugging the Boltzmann weight into the log-likelihood of a complete ranking σ,

$$\ln p(\mathcal{O}_T = \sigma \mid \mathcal{B}_N) \;=\; \sum_{n=1}^{N} \Big[G^{(\sigma(n))} - C_n(\mathcal{B}_N) \Big], \qquad C_n(\mathcal{B}_N) \;\triangleq\; \ln \sum_{k=n}^{N} \exp G^{(\sigma(k))}. \tag{39}$$

Substituting into the ELBO (29) and applying the log-derivative trick,

$$\nabla_\phi \mathcal{L}(\theta, \phi \mid \mathcal{B}_N) = \mathbb{E}_{p_q(\mathcal{B}_N)} \left[\sum_{n=1}^{N} \Big(G^{(\sigma(n))} - C_n(\mathcal{B}_N) \Big) \nabla_\phi \ln p_q(\mathcal{B}_N) \right] - N \nabla_\phi D_{\mathrm{KL}}(q_\phi \,\|\, p_{\pi_\theta}). \tag{40}$$

The baselines $C_n(\mathcal{B}_N)$ shrink monotonically down the ranking, with $C_1(\mathcal{B}_N) \geq C_2(\mathcal{B}_N) \geq \cdots \geq C_N(\mathcal{B}_N) = G^{(\sigma(N))}$. This produces a non-uniform gradient signal across rank positions. At the bottom ($n = N$), the baseline reduces to $G^{(\sigma(N))}$ itself, so the advantage is exactly zero. At the top ($n = 1$), the log-sum-exp is dominated by the largest return $G^{(\sigma(1))}$, so the advantage is non-zero but small in magnitude. Middle-ranked trajectories carry the largest advantages, since their returns are smaller than the log-sum-exp but the log-sum-exp itself remains dominated by the top of the ranking. Gradient signal is therefore diluted at both extremes and concentrated on the middle of the ranking.

Policy gradient under the MNL model. Starting from the averaged ELBO of Eq. 38 and plugging in the Boltzmann weight, the per-winner log-likelihood is

$$\ln p(\mathcal{O}_T = n \mid \mathcal{B}_N) \;=\; G\Big(H_{0:T}^{(n)}\Big) - C(\mathcal{B}_N), \qquad C(\mathcal{B}_N) \;\triangleq\; \ln \sum_{i=1}^{N} \exp G\Big(H_{0:T}^{(i)}\Big). \tag{41}$$

Substituting and applying the log-derivative trick,

$$\nabla_\phi \mathcal{L}(\theta, \phi \mid \mathcal{B}_N) = \mathbb{E}_{p_q(\mathcal{B}_N)} \left[\frac{1}{N} \sum_{n=1}^{N} \Big(G\Big(H_{0:T}^{(n)}\Big) - C(\mathcal{B}_N) \Big) \nabla_\phi \ln p_q(\mathcal{B}_N) \right] - N \nabla_\phi D_{\mathrm{KL}}(q_\phi \,\|\, p_{\pi_\theta}). \tag{42}$$

Unlike typical policy gradient methods where the baseline allows advantages of both signs, our derivation yields $C(\mathcal{B}_N) \geq G(H_{0:T}^{(n)})$ for all n. All advantages are therefore non-positive, so the learning signal operates through repulsion, with the KL term anchoring q_ϕ to p_{π_θ}. In contrast to the full PL gradient, all N items share a single baseline $C(\mathcal{B}_N)$, so the relative magnitudes of the per-trajectory advantages are determined entirely by the spread of returns rather than by rank position.

Recovery of GRPO. Eq. 42 is structurally close to GRPO (Shao et al., 2024) but differs in two ways. The baseline is the log-sum-exp $C(\mathcal{B}_N)$ rather than the group mean $\mu_G = \frac{1}{N} \sum_i G^{(i)}$, and the KL coefficient carries an explicit factor of N. Both differences are addressed by results already established.

- **Baseline.** Proposition 1 (Appendix D.1) shows that the per-trajectory coefficient $\alpha_n = \ln g_n - \ln \sum_k g_k$ admits a Jensen upper bound. The right-hand side of this bound is exactly GRPO's group-mean advantage $G(H_T^n) - \frac{1}{N} \sum_k G(H_T^k)$. Applying this bound to Eq. 42 replaces $C(\mathcal{B}_N)$ with $\mu_G + \ln N$ and yields GRPO's advantage. The Jensen slack is precisely the risk bonus profiled in Proposition 2.
- **KL scaling.** The factor N on the KL term is a consequence of the i.i.d. batch construction (§E.2). It absorbs into the KL coefficient λ, equivalent to tuning GRPO's β.

A residual structural difference is that GRPO assumes trajectories in $\mathcal{B}_N$ share a common prompt prefix, corresponding to branched rollouts from a single state. Our derivation treats trajectories as fully independent draws from q_ϕ, which is strictly weaker. The $\ln N$ offset between the log-sum-exp and group-mean baselines is independent of ϕ and does not affect the critical points of the optimization.

Recovery of DPO. At $N = 2$, the full PL and winner-only likelihoods coincide, since there is only one ranking consistent with trajectory 1 winning. The shared log-likelihood reduces to a single pairwise comparison,

$$\ln p(\mathcal{O}_T = \sigma \mid \mathcal{B}_N)\big|_{N=2} \;=\; \ln p(\mathcal{O}_T = n \mid \mathcal{B}_N)\big|_{N=2} \;=\; \ln \frac{\exp G^{(1)}}{\exp G^{(1)} + \exp G^{(2)}}, \tag{43}$$

which is the Bradley–Terry log-likelihood. Under DPO's own reparameterization $g(H_T) = \exp(s_\theta(H_T)/\beta)$ with $s_\theta = \log \pi_\theta / \pi_{\mathrm{ref}}$, the right-hand side becomes $\ln \sigma\Big(\frac{1}{\beta}(s_\theta^{(1)} - s_\theta^{(2)})\Big)$, recovering DPO's loss. The remaining bookkeeping concerns the role of the reparameterization, the factor of 2 absorbed into the KL coefficient, and the identification of π_θ with π_{ref} via the EM coordinate-ascent argument of Appendix E.1.

Summary. Our derivation provides a holistic account of two prominent RFT objectives. Both GRPO and DPO optimize a KL-regularized ELBO over trajectory batches under a Boltzmann likelihood. They differ in batch size ($N = 2$ vs. $N > 2$) and choice of baseline (group-mean Jensen surrogate vs. exact pairwise normalizer). Within this framework, the apparent diversity of recent preference-optimization methods reduces to modeling choices on ℓ, g, β, and the projection step, rather than to fundamentally different algorithmic principles. We do not claim a complete reduction of all such methods here. The framework is permissive enough to accommodate them, but the substantive recovery results are GRPO and DPO.

F Belief-Policy Mirror Descent under Batch CAI

This section describes our graphical model extension for the contextual-bandit study (see also Appendix G.4). We extend our control-as-inference formulation for batch potentials to deal with uncertainty given past data, and use our FTW approach to perform mirror descent in policy-space with a Gaussian belief. This structure enables closed form policy iteration, where each mirror-descent step extracts maximum value from past data given what the learning operator allows.

Setup. We use an empirical Bayes type approach to fit a batch version of the variational distribution $q(a, W)$ to approximate our batch-CAI posterior $p(a_t|H_{1:t-1}, \mathcal{O}_{t:T})$ from Eq. (4). We write the policy as a linear softmax over bias-augmented features[2] $\bar{\phi}(x) = (1, \tilde{\phi}(x)) \in \mathbb{R}^{d+1}$ that depend on some context $x \in \mathcal{X}$, that is $\pi_W(a \mid x) \propto \exp\big((W\bar{\phi}(x))_a/\tau\big)$ with $W \in \mathbb{R}^{|\mathcal{A}|\times(d+1)}$ and temperature τ. Then, $\bar{q}$ represents the posterior predictive $\bar{q}(a_t \mid x_t, H_{1:t-1}) = \mathbb{E}_{q(W|H_{1:t-1})}[\pi_W(a_t \mid x_t)]$ where $q(W|H_{1:t-1})$ is the variational belief state given a past history of interactions $H_{1:t-1} = \{(x_i, a_i, r_i)\}_{i=1}^{t-1} \cup \{x_t\}$ for some reward r_i (see also Section 2 from the main text). Thus, sampling from $\bar{q}$ works through posterior sampling where we first draw a hypothesis W from the belief $q(W_t \mid H_{1:t-1})$, and then draw an action a_t from π_{W_t} given hypothesis W_t and context x_t. Write the batch $\mathcal{B}_N = \big((x_1, a_1), \ldots, (x_N, a_N)\big)$ as some sample originating from one proposal policy over multiple contexts, or from multiple actions on the same context. This gives the batch joint $\bar{q}(\mathcal{B}_N \mid \cdot) = \prod_i \bar{q}(a_i \mid x_i, \cdot)p(x_i)$ where '$\cdot$' implies any history and $p(x_i)$ is some arbitrary context-process. We use a simple Gaussian belief for $q(W \mid H_{1:t-1})$ that factorizes independently over the actions $q(W) \triangleq \prod_a \mathcal{N}(w_a; \mu_a, \Lambda_a^{-1})$ such that epistemic uncertainty over actions is perfectly captured but there is still generalization across contexts. We initialize each w_a at $\mu_a = 0$, $\Lambda_a = I$, which induces a uniform $\bar{q}$ since $\mathrm{Law}(W_0)$ induces exchangeability across arms.

The evidence lower-bound splits in two penalties. Applying Eq. (4) to the joint $q(\mathcal{B}_N, W) = q_W(\mathcal{B}_N)q(W)$ against the reference $p(\mathcal{B}_N, W) = \pi_W(\mathcal{B}_N)p(W)$, and splitting up the divergence gives the conditional ELBO,

$$\ln p_\pi(\mathcal{O}_t \mid H_{<t}) = \ln \mathbb{E}_{p_\pi(\mathcal{B}_N, W_t|H_{<t})}\Phi_t(\mathcal{B}_N) \geq \mathbb{E}_{p_q(\mathcal{B}_N, W_t|H_{<t})}[\ln \Phi_t(\mathcal{B}_N)] - KL(p_q\|p_\pi) \quad (44)$$

$$\mathcal{L}(q, \pi; H_{<t}) = \mathbb{E}_{q(W_t|H_{<t})}\left[\mathbb{E}_{q(\mathcal{B}_N|x_t, W_t)}[\ln \Phi_t(\mathcal{B}_N)] - N \cdot KL(q_W\|\pi_W)\right] - KL(q(W_t \mid H_{<t})\|p(W_t \mid H_{<t})) \quad (45)$$

The history $H_{<t}$ conditioning is important to emphasize since the contextual bandit assumes stationary transition dynamics, i.e., $x_{t+1} \perp\!\!\!\perp x_t$. This setting implies episodic control for the actions $\mathcal{B}_N$. As a result, the optimality variables $\mathcal{O}_{t+1} \perp\!\!\!\perp \mathcal{O}_t$ so long as we also assume that the hypotheses $W_t \sim q$ are also independent of $\mathcal{O}_t, \forall t$. This assumption is reasonable in a typical Bayesian filtering setting, it's analogous to assuming that the dynamics of the world (world-model) are independent of rewards you may try to maximize. The resulting lower-bound $\mathcal{L}(q, \pi; H_{<t})$ then contains the typical potential and policy KL-term that tilts the prior policy π towards the batch-potential at a fixed hypothesis W_t.

[2]The features are defined in Appendix G.4 as part of the environment description.

The addition is the belief term that accumulates evidence about which hypotheses are plausible. We will optimize q per factor by coordinate ascent.

Ascent in $q(\mathcal{B}_N \mid W)$. The joint $q(\mathcal{B}_N, W)$ splits into two factors for coordinate ascent, the policy q_W sits in an inner-optimization which is what we complete first. Assume we obtain some history $H_{<t}$ which we use to sample a hypothesis $W_t \sim q(W_t \mid H_{<t})$, then write the inner-ELBO as $\mathcal{L}(q, \pi; x_t)$. We solve the inner-optimization and explicitly write it as a function of W_t,

$$q^*(\mathcal{B}_N \mid W) \triangleq \frac{\pi_W \Phi}{Z_N(W)}, \qquad Z_N(W) \triangleq \mathbb{E}_{\pi_W}[\Phi], \tag{46}$$

$$\mathcal{L}(q_W, \pi_W; x_t) = \mathbb{E}_{q(\mathcal{B}_N|W)}\big[\ln \Phi\big] - N \cdot KL\big(q_W(a_t \mid x_t) \,\|\, \pi_W(a_t \mid x_t)\big) \tag{47}$$

$$= \mathbb{E}_{q(\mathcal{B}_N|W)}\Big[\ln \frac{\pi_W \Phi}{q(\mathcal{B}_N \mid W)}\Big] = \mathbb{E}_{q(\mathcal{B}_N|W)}\Big[\ln \frac{Z_N(W)\, q^*}{q(\mathcal{B}_N \mid W)}\Big] \tag{48}$$

$$= \ln Z_N(W) - KL\big(q(\mathcal{B}_N \mid W) \,\|\, q^*\big) \tag{49}$$

$$\implies \max_{q(\mathcal{B}_N|W)} \mathcal{L}(q_W, \pi_W; x_t) = \ln Z_N(W), \quad \text{at } q(\mathcal{B}_N \mid W) = q^*. \tag{50}$$

where the first line again uses the equivalence of Eq. 3. This equivalence of ELBO maximization and analytical rewriting in terms of the posterior makes the last step seem somewhat trivial, but the important part is to write the optimization as a function of W_t which we need to perform ascent for the variational belief. This function is equal to the policy's partition function.

The rewriting is valid whenever $0 < Z_N(W) < \infty$, which holds even for potentials that assign zero to some batch-items (e.g., the tournament potential of Eq. (5) for some choices of C and K). Such items receive $q^*_{\text{non-elite}} = 0$, which is fine as long as the per batch partition gives $Z_N(W) > 0$.

Ascent in $q(W)$. With the E-step solved, Eq. (45) reduces to a belief-optimization: maximize $\mathbb{E}_{q(W_t|H_{<t})}[\ln Z_N(W_t)] - KL(q(W_t \mid H_{<t}) \,\|\, p(W_t \mid H_{<t}))$ over the belief. This maximization recovers the underlying objective,

$$\mathbb{E}_{q(W_t|H_{<t})}\big[\ln Z_N(W_t)\big] - KL\big(q(W_t \mid H_{<t}) \,\|\, p(W_t \mid H_{<t})\big)$$
$$= \ln p_\pi(\mathcal{O}_t \mid H_{<t}) - KL\big(q(W_t \mid H_{<t}) \,\|\, p(W_t \mid \mathcal{O}_t, H_{<t})\big). \tag{51}$$

In other words, the belief-optimization minimizes divergence to the tilted Bayesian posterior. Therefore, if we look at the role of the posterior $p(W_t \mid \mathcal{O}_t, H_{<t})$ we can conclude that coordinate ascent in the belief $q(W_t)$ amounts to variational Bayesian filtering. To make this optimization tractable, we start with a simple uninformative prior $q(W_0) = p(W_0)$ and perform iterative coordinate ascent (expectation-maximization). We iterate the ascent step for the belief q_{t+1} by setting the reference belief as the previous belief q_t and writing the loss as $\ell_t(W) \triangleq -\ln Z_{N,t}(W)$. This turns each ascent step into a proximal optimization step,

$$q_{t+1} = \arg\max_q \Big\{\mathbb{E}_q\big[\ln Z_{N,t}\big] - KL(q \,\|\, q_t)\Big\} = \arg\min_q \Big\{\mathbb{E}_q\big[\ell_t\big] + KL(q \,\|\, q_t)\Big\}. \tag{52}$$

stated equivalently as a minimization to match mirror-descent framing (which we need later for the implementation). When q is unrestricted, Eq. (52) has a closed-form solution, $q_{t+1} \propto q_t\, Z_{N,t}$, which uses Bayes' rule to update the previous belief with the likelihood of $\mathcal{O}_t$. This unrestricted closed-form solution is what motivates the variational coordinate-ascent update rule.

Solving Eq. (52) for a Gaussian. We will instead use a Gaussian belief for the variational posterior $q(W_t \mid H_{<t}) = \prod_a \mathcal{N}(w_a; \mu_{a,t}, \Lambda^{-1}_{a,t})$ as in the Setup paragraph, for this we first write the Gaussian in terms of its exponential family form with sufficient statistics $T(w) = (w, ww^\top)$. Expanding the log-density shows what this form is,

$$\ln \mathcal{N}(w; \mu, \Lambda^{-1}) = w^\top \Lambda \mu - \tfrac{1}{2} w^\top \Lambda w + \text{const} = \big\langle \Lambda\mu,\, w \big\rangle + \big\langle -\tfrac{1}{2}\Lambda,\, ww^\top \big\rangle + \text{const}, \tag{53}$$

a linear function of the two statistics. The coefficients multiplying the statistics w are called the natural parameters ν, and the means of w under q are called the expectation parameters m,

$$\nu = \big(\Lambda\mu,\, -\tfrac{1}{2}\Lambda\big), \qquad m = \big(\mathbb{E}_q[w],\, \mathbb{E}_q[ww^\top]\big) = \big(\mu,\, \Lambda^{-1} + \mu\mu^\top\big), \tag{54}$$

Either pair identifies the same Gaussian and their normalizer links them. If we write $A(\nu)$ for the log of the normalizer as a function of ν (i.e., the log-partition function of the Gaussian under the natural parametrization), then we can use a standard exponential-family identity to obtain $m = \nabla_\nu A(\nu)$.

This coordinate-system makes the proximal optimization step of Eq. (52) simple to perform for two reasons. First, between two Gaussians the KL penalty equals the Bregman divergence of A evaluated at their natural parameters. Eq. (52) thus takes the standard form of a mirror-descent step, which is gradient descent with the squared Euclidean penalty to some reference replaced by the Bregman divergence, here that of A (Khan and Rue, 2023, Sec. 2.2) to ν_t. Second, gradients with respect to ν reduce to gradients with respect to m, which we can easily compute. Since all terms in Eq. (52) depend on the belief only through ν, we can differentiate the objective in ν.

Completing this gives $\nabla_\nu KL(q_\nu \,\|\, q_{\nu_t}) = \nabla^2 A(\nu)\,(\nu - \nu_t)$ for the penalty, and the chain rule through $m = \nabla A(\nu)$ gives $\nabla_\nu \mathbb{E}_{q_\nu}[\ell_t] = \nabla^2 A(\nu)\,\nabla_m \mathbb{E}_{q_\nu}[\ell_t]$ for the loss (negative log-likelihood). Setting the sum of the two gradients to zero gives $\nabla^2_\nu A(\nu)\,\big(\nu - \nu_t + \nabla_m \mathbb{E}_{q_\nu}[\ell_t]\big) = 0$. We can then use another exponential-family identity that gives the second-order derivative of A as the covariance of the sufficient statistics, $\nabla^2_\nu A = \mathrm{Cov}_{q_\nu}[T(w)]$, which equals the Fisher information in natural coordinates. Since a covariance is positive definite it is invertible, so multiplying the stationarity condition by $(\nabla^2_\nu A)^{-1}$ leaves $\nu - \nu_t + \nabla_m \mathbb{E}_{q_\nu}[\ell_t] = 0$, and thus,

$$\begin{aligned} \nu_{t+1} &= \nu_t - \nabla_m \mathbb{E}_q[\ell_t], \quad \text{i.e.,} \\ (\Lambda\mu)_{t+1} &= (\Lambda\mu)_t - \nabla_{m_1} \mathbb{E}_q[\ell_t], \qquad \big(-\tfrac{1}{2}\Lambda\big)_{t+1} = \big(-\tfrac{1}{2}\Lambda\big)_t - \nabla_{m_2} \mathbb{E}_q[\ell_t]. \end{aligned} \tag{55}$$

Since $\nabla_m \mathbb{E}_{q_\nu}[\ell_t]$ depends on ν, stationarity puts ν_{t+1} on both sides (cross-dependence/ stationary-point). The implementation instead evaluates the gradient at the previous belief q_{ν_t}.

Finally, we need to express $\nabla_m \mathbb{E}_q[\ell_t]$ in terms of (μ, Λ) to unpack the learning rule. Inverting Eq. (54) gives $\mu = m_1$ and $\Lambda^{-1} = m_2 - m_1 m_1^\top$, so the chain rule yields $\nabla_{m_1} \mathbb{E}_q[\ell_t] = \nabla_\mu \mathbb{E}_q[\ell_t] - 2(\nabla_{\Lambda^{-1}} \mathbb{E}_q[\ell_t])\,\mu$ and $\nabla_{m_2} \mathbb{E}_q[\ell_t] = \nabla_{\Lambda^{-1}} \mathbb{E}_q[\ell_t]$. Substituting into Eq. (55), the second component gives $\Lambda_{t+1} = \Lambda_t + 2\nabla_{\Lambda^{-1}} \mathbb{E}_q[\ell_t]$. The first component gives $(\Lambda\mu)_{t+1} = \Lambda_t \mu_t - \nabla_\mu \mathbb{E}_q[\ell_t] + 2(\nabla_{\Lambda^{-1}} \mathbb{E}_q[\ell_t])\,\mu_t = \Lambda_{t+1}\mu_t - \nabla_\mu \mathbb{E}_q[\ell_t]$, where the last equality collects the two terms in μ_t into the new precision. Then, the theorems of Bonnet (1964) and Price (1958) convert the parameter gradients into expectations of the derivatives of the loss with respect to the arm's row w, $\nabla_\mu \mathbb{E}_q[\ell_t] = \mathbb{E}_q[\nabla_w \ell_t]$ and $\nabla_{\Lambda^{-1}} \mathbb{E}_q[\ell_t] = \frac{1}{2}\mathbb{E}_q[\nabla^2_w \ell_t]$, which yields

$$\boxed{\Lambda_{a,t+1} = \Lambda_{a,t} + \mathbb{E}_q\big[\nabla^2_{w_a} \ell_t\big], \qquad \mu_{a,t+1} = \mu_{a,t} - \Lambda^{-1}_{a,t+1}\, \mathbb{E}_q\big[\nabla_{w_a} \ell_t\big],} \quad \forall a \in \mathcal{A}, \tag{56}$$

the Bayesian learning rule of Khan and Rue (2023, Sec. 1.3.2) derived through our bound, matching their online Newton update with unit learning rate and, as in their conjugate-Bayes setting (their Sec. 5.1), the previous belief as the prior. The increments to the precision are positive semi-definite (Eq. (60)), so the precision only accumulates and the effective step Λ^{-1}_{t+1} anneals as evidence arrives.

Note that the natural-parameter derivation placed the updated precision Λ_{t+1} in front of the mean update by itself, so the mean step accounts for the most recent evidence. This would not naturally occur if we derived the update through (μ, Λ) directly. It is also the correct choice, since with Λ_{t+1} the update reproduces the exact Gaussian posterior whenever the loss is quadratic in W, and with Λ_t it does not. The two derivatives of ℓ_t we need for computing Eq. (56) are derived next.

Cross-entropy loss. Differentiating the log-partition with respect to W, we obtain,

$$\nabla_W \ell_t = -\nabla_W \ln Z_{N,t}(W) = -\frac{\mathbb{E}_{\pi_W}\big[\Phi\, \nabla_W \ln \pi_W(\mathcal{B}_N)\big]}{Z_{N,t}(W)} = -\,\mathbb{E}_{q^*}\big[\nabla_W \ln \pi_W(\mathcal{B}_N)\big], \tag{57}$$

with $\ln \pi_W(\mathcal{B}_N) = \sum_i \ln \pi_W(a_i \mid x_i)$ and using the definitions of q^* and $Z_N(W)$ from Eq. (50). This is the gradient of the batch cross-entropy $-\mathbb{E}_{q^*}[\ln \pi_W(\mathcal{B}_N)]$ against the fixed target q^*. The tilt Φ also doesn't need numerical evaluation and sampling is sufficient. The per arm gradients w_a split up from W.

Then we derive the Hessian needed for the precision update of Eq. (56),

$$\nabla^2_W \ell_t = \mathbb{E}_{q^*}\big[\nabla^2_W\big(-\ln \pi_W(\mathcal{B}_N)\big)\big] \;-\; \mathbb{E}_{q^*}\big[(\nabla_W \ln q^*)\,(\nabla_W \ln \pi_W(\mathcal{B}_N))^\top\big]. \tag{58}$$

We use the generalized Gauss-Newton (GGN) approximation by keeping the first term only and dropping the second, so we essentially drop the derivative of q^*. This enforces that the second-order

term is positive semi-definite, which cannot be guaranteed with the dropped term included. In other words, we differentiate as if q^* were fixed even though it technically varies with W.

The GGN term is available per arm in closed form for the softmax policy. Writing $-\ln \pi_W(a_i \mid x_i) = -z_{i,a_i} + \mathrm{logsumexp}(z_i)$ with logits $z_i = W\bar{\phi}(x_i)/\tau$ linear in W, the chain rule gives,

$$\nabla^2_{w_a w_b}\big(-\ln \pi_W(a_i \mid x_i)\big) \;=\; \big(\delta_{ab}\, \pi_W(a \mid x_i) - \pi_W(a \mid x_i)\, \pi_W(b \mid x_i)\big)\, \bar{\phi}(x_i)\bar{\phi}(x_i)^\top/\tau^2, \quad (59)$$

independent of the sampled action a_i. The factorized belief holds one precision per arm, so Eq. (56) uses only the $a = b$ blocks, summed over the batch,

$$\nabla^2_{w_a} \ell_t \;=\; \textstyle\sum_{i=1}^N \pi_W(a \mid x_i)\big(1 - \pi_W(a \mid x_i)\big)\, \bar{\phi}(x_i)\bar{\phi}(x_i)^\top/\tau^2, \quad (60)$$

which with non-negative coefficients guarantees precision increments of Eq. (56) that are positive semi-definite.

FTW's variational optimization cannot condition on hypotheses. The derivation above solves the inner optimization for q^* separately at each hypothesis W_t, which is what makes $\ln Z_N(W)$ a function of W. Our implementation for FTW however uses a replay buffer with (x, a, r) tuples whose generating hypotheses are implicitly marginalized out, and tournament selection ranks by observed reward. The tilt the implementation samples is therefore

$$\tilde{q}^*(\mathcal{B}_N) \;\propto\; \Phi(\mathcal{B}_N)\, \mathbb{E}_{q(W)}\big[\pi_W(\mathcal{B}_N)\big], \quad (61)$$

the potential applied to the belief-mixture of policies: each step pushes actions rolled out from a fresh posterior draw $W \sim q_t$ into the buffer, so uniform draws from the buffer sample the mixture over recent beliefs, and keeping tournament winners tilts those samples by Φ without evaluating it. This tilt coincides with the pointwise $q^*(\cdot \mid W)$ as the belief concentrates, and the update of Eq. (56) is unchanged. The restriction only affects the variational gradient on the policy and the belief step itself is still solved exactly as in Eq. (56), just with a different tilt. It also strengthens the Gauss-Newton choice in Eq. (58), since $\tilde{q}^*$ carries no W at all, the GGN is the exact Hessian.

How GRPO, DPO, and FTW instantiate the Hessian. The three algorithms of main consideration in this manuscript correspond to different choices of the potential Φ through our theoretical framework. Within this construction, only GRPO cannot be interpreted as a probabilistic graphical model, due to the Jensen upper bound of Proposition 1. This paragraph briefly summarizes how each potential produces its Hessian, and what decides it is not the potential itself but the route by which it enters the belief update: through the sampling of q^* alone, through a closed-form loss that itself depends on W, or through a bound that admits no likelihood. Throughout we apply the Gauss-Newton approximation, which guarantees positive semi-definite precision increments.

Filtered cross-entropy. The potential Φ is the elite-filtering of Eq. (5), and FTW follows the derivation above literally: the potential enters only through the sampling of q^*. The loss is therefore the cross-entropy against q^* regardless of which Φ is in force, its exact Hessian is Eq. (60), and the Gauss-Newton step is FTW's only approximation.

Bradley-Terry. The potential compares one $|\mathcal{B}_N| = 2$ pair through the Plackett-Luce potential of Corollary 1, but DPO does not take the route above: its potential is available in closed form and depends on W itself, so DPO descends it directly rather than sampling q^*. That loss is a log-sigmoid of the delta, $-\ln \sigma(\beta\Delta)$ with $\Delta = \ln \pi_W(a_w \mid x_w) - \ln \pi_W(a_l \mid x_l)$ the log-probability gap between winner and loser (the uniform reference policy cancels). Since Δ is nonlinear in W, its exact Hessian is indefinite, and the Gauss-Newton step keeps only the positive semi-definite part, making the belief the posterior of the Bradley-Terry model linearized about the current mean (Immer et al., 2021).

Centred advantage. Proposition 1 explains why this case is different. The exact Rao-Blackwellized coefficients $\alpha_n = \ln g_n - \ln \sum_k g_k$ are log-probabilities, hence $\alpha_n \le 0$, and the resulting loss is a non-negative combination of cross-entropies with a positive semi-definite Hessian. The Jensen bound of Proposition 1 replaces them by centred advantages $A_n = G_n - \frac{1}{N}\sum_k G_k$ of mixed sign, and as a consequence the Hessian directly becomes indefinite even for the GGN case. Thus, we hold $\Lambda \equiv \Lambda_0$ and update only the mean, $\mu_{t+1} = \mu_t - \eta\Lambda_0^{-1}\mathbb{E}_q[\nabla_W \ell_t]$. This mean-only update with a swept step η is plain GRPO, and we additionally implement two controls.

These two controls are: GRPO-NPG which replaces the typical gradient-descent step by a natural-gradient preconditioner from the current batch, $\mu_{t+1} = \mu_t - (F_t + \Lambda_0)^{-1}\mathbb{E}_q[\nabla_W \ell_t]$ with F_t the per-arm softmax Fisher of Eq. (60). Second, GRPO-ONG accumulates this same Fisher into Λ exactly as Eq. (56) would, i.e., we heuristically insert the GRPO loss into FTW's belief learning rule.

G Experimental Setup

G.1 Hardware

Experiments were conducted across two cloud providers. Bandit, Brax, and Search-R1 experiments ran on Google Cloud Platform (GCP) using `a2-highgpu-1g` (1× NVIDIA A100-40GB, 12 vCPUs, 85 GB RAM, Intel Xeon Cascade Lake), `a2-ultragpu-1g` (1× NVIDIA A100-80GB, 12 vCPUs, 170 GB RAM, Intel Xeon Cascade Lake), and `a3-highgpu-1g` (1× NVIDIA H100-80GB, 26 vCPUs, 234 GB RAM, Intel Xeon Sapphire Rapids), each with 500 GB SSD persistent disk. The 40GB A100s were sufficient only to host the Search-R1 retrieval server (our own simplified faiss-index implementation to sidestep a bug in the base implementation that caused VRAM overheads), all other agents ran on the 80GB A100s or the H100. Sokoban experiments required a larger vision-language model and were run on Lambda Cloud using a 2× NVIDIA H100-80GB instance (52 vCPUs, 442 GB RAM, Intel Xeon Platinum 8480+, 5.5 TB SSD). The Search-R1 seed repetitions ran on a Lambda 8× NVIDIA A100-80GB instance, one run per GPU with a shared retrieval server.

Reproducing the experiments in this paper requires roughly 6,000 GPU-hours, of which Sokoban (4 algorithms, 5 seeds, about 3 days per run on 1 to 4 GPUs) accounts for about 3,400, Brax (3 algorithms, 10 seeds, about 2 days per run) for about 1,400, and Search-R1 for the remainder, corresponding to an estimated operating cost of $25,000 to $35,000 (USD) at on-demand rates. This excludes setup, algorithm development, and debugging, which vary substantially with practitioner experience and were not separately tracked.

G.2 Compute and Memory Footprint

We briefly compare the computational footprint of FTW against GRPO, PPO, and REINFORCE++ on three items, data generation, the learning update, and memory. For data generation, all four methods consume the same number of rollouts per step in our Sokoban runs, but GRPO must obtain them as n rollouts of the same prompt, which requires resetting the environment to identical initial states, whereas FTW, PPO, and REINFORCE++ draw one rollout per prompt. Prefix caching can offset part of the group overhead, but the duplication requirement remains. The learning updates differ negligibly in cost apart from the critic. FTW performs a cross-entropy loss on the filtered samples plus a KL penalty, GRPO and REINFORCE++ add PPO-style clipping and advantage normalization, and PPO additionally trains a second model of the same size as the policy, which incurs a further forward and backward pass per update. In our Sokoban runs this meant 4 GPUs for PPO against 2 for GRPO and REINFORCE++ and 1 to 2 for FTW. In terms of storage, FTW is the most expensive because of the replay buffer of past rollouts, 2,000 trajectories on Sokoban and 10,000 on Search-R1, held in CPU memory, whereas the other three methods are on-policy. The overhead depends on the data modality, text environments such as Search-R1 have a far smaller footprint per trajectory than image-based ones such as Sokoban. GRPO extended with a replay buffer, as done by Arnal et al. (2026), would match FTW's memory footprint, leaving the group duplication as the remaining difference. REINFORCE++ is the cheapest on every axis and is also the method that lags in Figure 4, which is what motivates paying for one of the three variance-reduction sources.

G.3 Software Stack

Our experiments span four distinct environments: Bandits, Brax continuous control, Search-R1, and Sokoban. Each environment required a separate software stack and varying setup, for three main reasons. First, *packaging conflicts*: the verl-agent fork we build on (Feng et al., 2025) lags upstream `verl` and pins `torch 2.8.0+cu128/vLLM 0.11.0`. Sokoban requires the legacy `gym 0.26.2` API (via `gym-sokoban`), which conflicts with the `gymnasium`-based stack used elsewhere. Second, *hardware heterogeneity across providers*: we ran on a mix of A100-40GB, A100-80GB, and H100-80GB instances (see Appendix G.1), and VRAM size materially changed configuration. Third, *model choices dictated by baseline parity and modality*: Search-R1 uses `Qwen2.5-3B-Instruct` to match the scale reported in prior work, and Sokoban uses the vision-language `Qwen2.5-VL-3B-Instruct` to consume the RGB puzzle rendering directly. The Brax environments were fortunately much simpler, as they only required a small MLP provided by the reference implementation of Freeman et al. (2021). All cloud runs share Ubuntu 24.04.4 LTS, Python 3.12.3, NVIDIA driver 570.211.01, and set `NCCL_NET=Socket`. This socket setting made our runs slower, but it was required due to GCP-related issues. Table 2 lists the key package versions per environment. For more specific details, we advise the

reader to look at the verl-agent documentation `https://github.com/langfengQ/verl-agent` which has extensive documentation for the environment setup.

Table 2: Key software versions per experimental environment. Top section: policy model. Bottom section: core framework (first row) and other critical pinned packages. All LLM-agent runs additionally use `transformers 4.57.6` and `ray 2.54.1`. †`flash-attn` built from source to resolve ABI mismatch with `torch 2.8.0+cu128`. SearchR1 requires a separate server for the wiki retrieval, making it the most involved environment to set up.

Bandit & Brax	**SearchR1**	**Sokoban**
Logit-Table / MLP	Qwen2.5-3B-Instruct	Qwen2.5-VL-3B-Instruct
JAX 0.9.2	verl-agent 0.3.1.dev	verl-agent 0.3.1.dev
brax 0.14.2	torch 2.8.0+cu128	torch 2.8.0+cu128
mujoco-mjx 3.6.0	vLLM 0.11.0	vLLM 0.11.0
flax 0.12.6	flash-attn 2.7.4.post1†	gym 0.26.2
optax 0.2.8	faiss-gpu-cu12	gym-sokoban 0.0.6

G.4 Main Environments

We evaluated across seven environments: four feature-isolating bandit problems (the three diagnostic bandits of Figure 2 and a linear contextual bandit), continuous control (HalfCheetah), a text-based multi-turn LLM-agent task (Search-R1), and a vision-language grid puzzle (Sokoban).

σ-trap bandit (Figure 2, left). A two-armed Gaussian bandit with an optimal arm ($\mu = 10$, $\sigma = 0.5$) and a zero-mean distractor whose standard deviation is swept over $\sigma_{\text{trap}} \in [0.5, 32768]$. A risk-neutral learner commits to the optimal arm at every noise level, while a risk-seeking learner is eventually drawn to the distractor as its variance grows. All methods draw $N = 8$ pulls per step for $T = 10^6$ steps with a shared learning rate of 10^{-4}, except for the GRPO lr-adjusted version which scales the learning rate by $\sqrt{\sigma_1/\max_a \sigma_a}$ to compensate for increased gradient noise but has to train for a $T' = 32 \cdot T$ step budget to actually converge. FTW uses C=2, K=1, J=8 per step over a FIFO buffer of max age h, our DPO ($\beta = 1$) version samples pairs from this same buffer mechanism. Replay-buffered curves are averaged over $h \in \{16, 64, 256, 1024, 4096\}$. We report the asymptotic $\pi(a^*)$ per noise level (10 seeds, bootstrap 95% CIs).

Truncated Pareto-α bandit (Figure 2, center). A ten-armed tabular bandit with multiplicative heavy-tailed returns $r = c_a(1-u)^{-1/\alpha}$, $u \sim \text{Unif}$, truncated to $[1, 10^6]$ so that all moments exist, with arm locations $c_a = 1 + 0.5a$. The tail index is swept over $\alpha \in \{0.05, \ldots, 1.7\}$, where smaller α produces heavier upper tails and the untruncated moment-generating function diverges at $\alpha \leq 1$. We report average (bold) and sample (transparent) learning curves of $\pi(a^*)$, see also Appendix H.3 for additional results on different reward distributions.

State-aliasing corridor (Figure 2, right). The short-corridor task of Sutton and Barto (2018) (Ex. 13.1): states $\{s_0, s_1, s_2\}$ with reward -1 per step, where the action effect in s_1 is reversed. The policy is a single global softmax over the two actions, which aliases all states and makes the optimal policy stochastic, $\pi(\text{right}) = 2 - \sqrt{2}$. We plot the L1 distance $|\pi(a) - \pi^*(a)|$ between the learned and optimal action distributions (represented as a 2D vector), averaged over a learning-rate population for 20k update steps (see also Appendix H.4 for additional results).

Linear contextual bandit. A contextual bandit with deterministic rewards and linear features that reports simple and cumulative regret to the optimal arm (all results report mean and 95% bootstrap CI over 30 seeds). We define the reward as $r(x, a) = 1/2 + 1/2\, \theta_a^\top \tilde{\phi}(x)$ with $\|\theta_a\| = 1$ and $\|\tilde{\phi}(x)\| \leq 1$, which bounds $r \in [0, 1]$ without clipping and fixes the marginal $\mathbb{E}_x[r(x, a)] = 1/2$. As a result, no action is better on average and only positional information allows exploitation. We achieve this by sampling for every task instance (i.e., one complete run for a given seed) the triple $\langle \omega, b, \theta \rangle$, such that $\phi_j(x) = \sqrt{2/d}\cos(\omega_j x + b_j)$ with $\omega_j \sim \mathcal{N}(0, \ell^{-2})$, $b_j \sim \text{Unif}[0, 2\pi)$, and lengthscale $\ell = 0.1$. The features are zero-meaned over the context grid and divided by the largest ℓ_2 norm $\max_{x'} \|\phi(x')\|_2$, so that $\|\tilde{\phi}(x)\|_2 \leq 1$ everywhere with feature dimensionality $d = 16$. Contexts x are drawn uniformly

over a linspace of 256 points in $[0, 1]$. Finally, we sample each arm's reward parameters uniformly from the unit-sphere: $\theta_a \sim \mathcal{N}(0, I_d)$ then l2 normalize to $\|\theta_a\| = 1$. We keep one special "safe" arm with $\theta = 0$ such that the reward is $1/2$ independent of context (safe but suboptimal).

All policy methods share the linear softmax policy $\pi(a \mid x) \propto \exp(W\bar{\phi}(x))_a$ over the bias-augmented features $\bar{\phi}(x) = (1, \tilde{\phi}(x))$, with $W \in \mathbb{R}^{|\mathcal{A}|\times(d+1)}$. Appendix F describes the complete parametrization and derivation of the learning rule. We used $|\mathcal{A}| = 20$ actions with a per-step batch-size (number of contexts) of 16. So, FTW and DPO draw $N = 16$ contexts and a single action, however, GRPO draws 4 contexts per step with an on-policy group of $G = 4$ actions for the group-mean baseline. As a baseline we use Thompson sampling (Thompson, 1933; Chapelle and Li, 2011) via conjugate Bayesian linear regression per arm.

Brax continuous control (Freeman et al., 2021). HalfCheetah with the Brax `spring` backend, episode length $1{,}000$, action repeat 1, observation normalization enabled. The policy is an MLP $[256, 256]$ with ReLU activations and a `NormalTanhDistribution` head (brax `SACNetworks`).

- Observations are continuous, with 18 dimensions.
- The action space is a bounded continuous vector in $[-1, 1]^D$, with $D = 6$.
- Rewards follow the standard Brax forward-velocity plus control-cost shaping (norm of the action).
- Episodes terminate when the agent reaches an unhealthy joint configuration.

The key metric is the evaluation episode return, computed with a fixed policy at lower temperature.

Search-R1 (Jin et al., 2025). Multi-turn open-domain QA with a retrieval tool, framed as a POMDP where the agent alternates between internal reasoning and tool invocation until it commits to an answer. At each turn the prompt contains the question and the interaction history, where prior search queries and the retrieved passages are interleaved as `<search>...</search>` and `<information>...</information>` blocks. The LLM emits chain-of-thought inside `<think>...</think>` and then chooses exactly one of two actions: a new query in `<search>...</search>` (control returns to the retriever) or a final answer in `<answer>...</answer>` (terminating the episode). Emitting both or neither consumes a turn. Retrieved passages are obtained by dense lookup against an E5 index of the Wikipedia-18 corpus (`wiki-18.jsonl`, 14 GB) compressed from a 61 GB flat index to an 815 MB IVF-PQ index (`e5_IVFPQ.index`) and served from a dedicated retrieval venv on port 8000, returning the top-3 passages per query. Reward is an exact-match indicator on the final answer against the gold label, with an additional -0.1 penalty for malformed emissions (missing or duplicate action tags). Train set: 169,615 questions from `PeterJinGo/nq_hotpotqa_train` (a mix of Natural Questions and HotpotQA). Test set, on which the main-text accuracies are computed, is the 51,713-question eval set bundled with the same release, drawn from NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultihopQA, Musique, and Bamboogle. Training-time curves use a fixed 1,280-question slice of it. Max turns: 4. Prompt/response budgets: 4,096 / 512 tokens.

Sokoban (Schrader, 2018). Vision-language box-pushing puzzle (`gym_sokoban`) on a 6×6 grid with one box and a generator search depth of 30 (the `num_gen_steps` parameter controlling puzzle complexity at instantiation, not the per-episode horizon). At each turn the environment's `rgb_array` rendering is supplied to the policy as an image via `data.image_key=images`, alongside the textual task description and a short history of prior actions. The vision-language model emits chain-of-thought inside `<think>...</think>` followed by a single direction in `<action>...</action>`, with admissible set {up, down, left, right}. Each direction is dispatched as the corresponding `Push` action in `gym_sokoban`, and unparsable or illegal emissions consume a turn. Rewards follow the `gym_sokoban` defaults: -0.1 per step, ±1 on pushing the box on/off its target, and $+10$ upon solving, with an additional -0.1 format penalty consistent with the main text. Train/validation split: 32 / 128 puzzles. Episode horizon: 15 turns. Prompt/response budgets: 1,024 / 512 tokens, with image tokens accounted for separately by the VLM processor and not counted against the prompt budget.

G.5 Hyperparameters

Hyperparameters are reported in two tables: Brax continuous control (Table 3) and the shared LLM-agent RL configuration (Table 4). For every baseline, SAC and the Brax network architecture from

Freeman et al. (2021), and the GRPO defaults from verl-agent (Feng et al., 2025), we retain the published values wherever possible to keep comparisons fair.

Brax continuous control. All methods share the same optimizer, replay buffer, observation normalization, and network architecture. Algorithm-specific differences (Table 3) concern the actor learning objective and the update-to-data (UTD) ratio.

Table 3: Brax HalfCheetah hyperparameters. All methods share $\mathrm{lr}{=}3{\times}10^{-4}$ (Adam), $\gamma{=}0.99$, replay size $2{\times}10^{6}$, 64 parallel envs, batch size 256, episode length 1000, 10^{9} total env steps, warmup 8192 transitions, and 10 seeds. Any parameters not mentioned are all kept at their default values from the Brax 0.14.2 (Freeman et al., 2021) implementation of SAC (Haarnoja et al., 2018).

Parameter	**FTW**	**SAC**
Actor objective	BC on top-K winners	$\alpha \log \pi - Q$
Elite filter C / J / K	64 / 256 / $\{1, 4\}$	n/a
Updates per env step	8 (BC)	1
Entropy	0.05 (fixed)	auto, target $\bar{H}{=}-\frac{1}{2}\lvert\mathcal{A}\rvert$
Critic	n/a	clipped double-Q, Polyak τ

LLM-agent RL. For both Search-R1 and Sokoban, all settings in Table 4 are shared across all tested algorithms. The replay buffer size is listed at the bottom of Table 4. GRPO replaces FTW's elite filtering with group-relative advantages over n rollouts per prompt, combined with a PPO-clip objective rather than the cross-entropy SFT loss. PPO replaces the group baseline with a learned critic, a second Qwen2.5-VL-3B-Instruct initialised from the base model and trained at lr 10^{-5} with value clipping 0.5, using GAE with $\lambda{=}1$ and $\gamma{=}1$ so the advantage is the Monte-Carlo return minus the value estimate. REINFORCE++ (Hu et al., 2025) keeps a single rollout per prompt and normalises returns by the global batch mean and standard deviation, with the same PPO-clip objective, KL penalty, and entropy coefficient as GRPO. Both baselines run directly through the verl-agent implementation (Feng et al., 2025) by switching its advantage estimator, with all other settings held fixed. The FTW configurations differ only in the minibatch size C and the number of elites per minibatch K. On Search-R1 we use $C{=}4$ with $K{=}1$, and on Sokoban $C{=}5$ with $K \in \{1, 2, 3\}$ retained trajectories, sweeping from strong to weak selection pressure. The limiting case $C{=}2$, $K{=}1$ is binary selection, which coincides with DPO's pairwise potential and is not run here. For all FTW configurations, the number of minibatches per gradient step J is set proportional to K such that each update receives $J \cdot K$ training targets at a fixed effective batch size. The cross-environment differences in Table 4 (e.g. KL coefficient, discount γ) are inherited from the published baseline settings for each benchmark rather than tuned for FTW. Default system prompts for the LLMs are kept at their default values as provided by the packages pinned in Table 2.

Table 4: LLM-agent hyperparameters. All runs use `bfloat16` rollouts and optimize with AdamW (Loshchilov and Hutter, 2019). Unless otherwise stated, we use the exact hyperparameter defaults as described in the respective baselines as implemented by verl-agent 0.3.1.dev (Feng et al., 2025). For FTW, the number of minibatches per update is $J = 128/K$, so every update uses $J \cdot K = 128$ training targets regardless of K.

Parameter	**SearchR1**	**Sokoban**
Learning rate	1×10^{-6}	1×10^{-6}
Weight decay	0.01	0.01
Gradient clip	1.0	1.0
Rollout batch (prompts)	256	32
Rollouts per prompt n (GRPO only)	5	8
Mini-batch size	512	64
Epochs per update	1	1
KL coefficient	10^{-3}	10^{-3}
Entropy coefficient	0.001	0.001
Invalid-action penalty	0.01	0.1
γ (multi-turn discount)	0.95	1.0
Temperature (train / val)	1.0 / 0.4	1.0 / 0.4
Top-p / top-k	1.0 / -1	1.0 / -1
Max prompt / response	4096 / 512	1024 / 512
Max model len (vLLM)	8192	1536
Replay buffer size N	10 000	2000

H Supplementary Results

This appendix collects results that complement the main paper. Section H.1 reports in-distribution learning curves for the LLM-agent experiments (test-time curves appear in the main text, Figure 4). Section H.2 dissects the FTW selection operator on the σ-trap, grading the polynomial filter by selection pressure and sweeping the temperature of an exponential-rank alternative. Section H.3 extends the Pareto-α sweep of Figure 2 to four additional return distributions, isolating which features of the return distribution drive the failure mode of cardinal-exponential potentials. Section H.4 expands the state-aliasing experiment with a learning-rate sweep, showing that the polynomial-vs-exponential concentration gap is not an artifact of a single learning rate (which dampens learning instability). Section H.5 identifies a solution bias of elite filtering over the replay buffer when returns saturate and comparisons are dominated by ties, complementing the robustness results above with the adversarial regime of the rank-based potential.

H.1 Training-time Success Rates for Sokoban and Search-R1

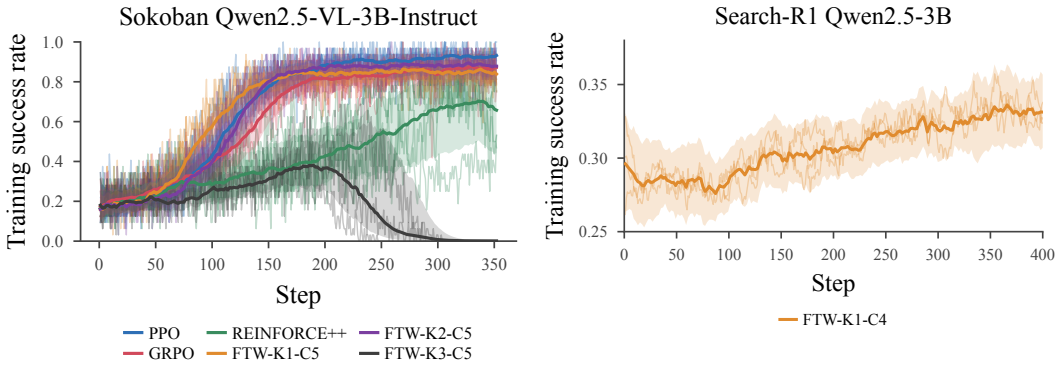


Figure 5: Online episode success rate on Sokoban (left) and Search-R1 (right) with EMA-smoothed means and $\pm$ 1 standard deviation over seeds, individual seeds are drawn faintly. Note that, the Search-R1 training split covers only NQ and HotpotQA, so the right panel is not on the same question distribution as the main-text reported test accuracy.

Figure 5 shows success rates on training episodes that complement the test-time evaluations in the main text. On Sokoban (left), our FTW-K1-C5 and FTW-K2-C5 and the GRPO and PPO baselines cluster almost indistinguishably together, whereas REINFORCE++ trains markedly slower and with larger seed variance. In contrast, our FTW-K3-C5 peaks on average near 0.4 and then collapses to zero. This collapse can partly be attributed to Sokoban's effectively binary returns. Whenever a group contains fewer than K successes the remaining elite slots are always filled with zeros, and such groups are far more common at $K{=}3$ than at $K{=}1$, where one success out of C already suffices. Although the imitated elite threshold still averages above the buffer mean in expectation (Appendix 3), the share of imitated zeros grows as the success rate falls and this drags each consecutive update further down. Ultimately this is a finite-sample effect rather than a flaw in the operator, a larger replay buffer would smooth the K3 curve out at the cost of slower learning. On Search-R1 (right), FTW-K1-C4 dips slightly during initial training, and then steadily improves. Note that the training success rate is not directly comparable to the test success rate, due to the validation set only covering NQ and HotpotQA questions.

H.2 Varying FTW Selection Operators for the noisy Bandit-trap

Extending upon Figure 2 (left) we provide more examples of the FTW filtering pressure on the σ-trap in Figure 6. All variants share the identical backbone of FTW, they only differ in how the filtering (winner selection) is applied to the replay buffer. As in the main text, results are averaged over buffer sizes $h \in \{16, 64, 256, 1024, 4096\}$ for FTW-polynomial and with $h \in \{16, 64\}$ for the exponential filtering on the ranking, as FTW-exp requires sorting the entire replay buffer, which gets expensive. Note that the FTW-exp-marg from the main text (Fig. 2) filtered using a softmax over the buffer returns, whereas this experiments uses a multinomial over their ordered rank (argsort index) for a

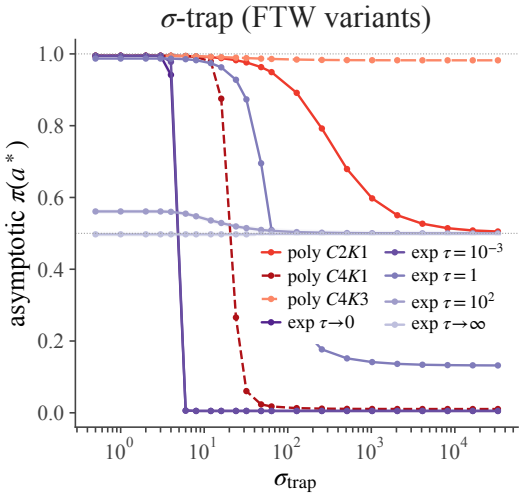


Figure 6: σ-trap solutions for FTW selection variants (10 seeds; shaded: bootstrap 95% CIs).

given temperature τ. This was done to eliminate the ordinal vs. marginal confounding argument of our FTW operator.

The polynomial selection (red lines, Figure 6) are determined by the selection pressure $^C/_K$. We see for a mild $C{=}4, K{=}3$ filter that the risk-profile stays minimal, even at larger σ. In contrast the more aggressive $C{=}4, K{=}1$ filter collapses onto the trap arm, as suspected. For the exponential-ordinal filter, we see that the greedy limit $\tau \to 0$ (buffer argmax) collapses onto the trap even swifter than the aggressive polynomial configuration. Unsurprisingly, the other extreme $\tau \to \infty$ stays constant to its 1/2 prior. The temperatures inbetween $\tau \in \{0.01, 1.0, 100.0\}$ form a type of U-curve that finds the optimal arm for moderate or low noise (temp = 1.0) before interpolating back to uniform for large τ or back to the overly aggressive $\tau = 0$ version.

H.3 Bandit Distribution Sweep

Figure 7 extends the Pareto-α sweep of Figure 2 (center) to four additional return distributions. The five rows isolate distinct features of the return distribution and check whether the theoretical risk profile of Proposition 2 predicts the empirical failure modes correctly.

Beta (κ shape). Bounded support, where varying κ moves between U-shaped, uniform, and bell-shaped returns. All three methods solve the bandit, and FTW shows slightly faster scale-regret reduction in the early phase. As expected, no method is destabilized when returns are bounded, regardless of shape.

Binomial. The clean sparsity sweep. All three methods identify the optimal arm. GRPO converges visibly slower than the rank-based methods at low p, consistent with its cardinal treatment of returns under sparse signals.

Binomial scaled ($\times 1000$). The same Bernoulli-rate sweep with rewards scaled by 1000, isolating magnitude robustness from sparsity. GRPO's curves remain bounded but the slow-convergence gap at low p widens substantially under the scale. FTW and DPO are scale-invariant by construction (rank-based), and their curves overlap their unscaled counterparts. This confirms that magnitude (not sparsity per se) is the destabilizing factor for the cardinal baseline.

Pareto-α. Reproduces the main-text result for the non-truncated version of the distribution. GRPO fails to identify the optimal arm at heavy-tail $\alpha \leq 1$, where the moment-generating function diverges. FTW and DPO remain stable, as predicted by the rank-based potentials' well-defined fixed point under any continuous return distribution.

Student-t (ν). Symmetric, polynomially-tailed distribution on $\mathbb{R}$. At $\nu \leq 2$ the variance of returns becomes undefined, and GRPO's group-mean baseline destabilizes accordingly, with scale-regret curves drifting noticeably. FTW and DPO remain unaffected, since the rank-based potential depends

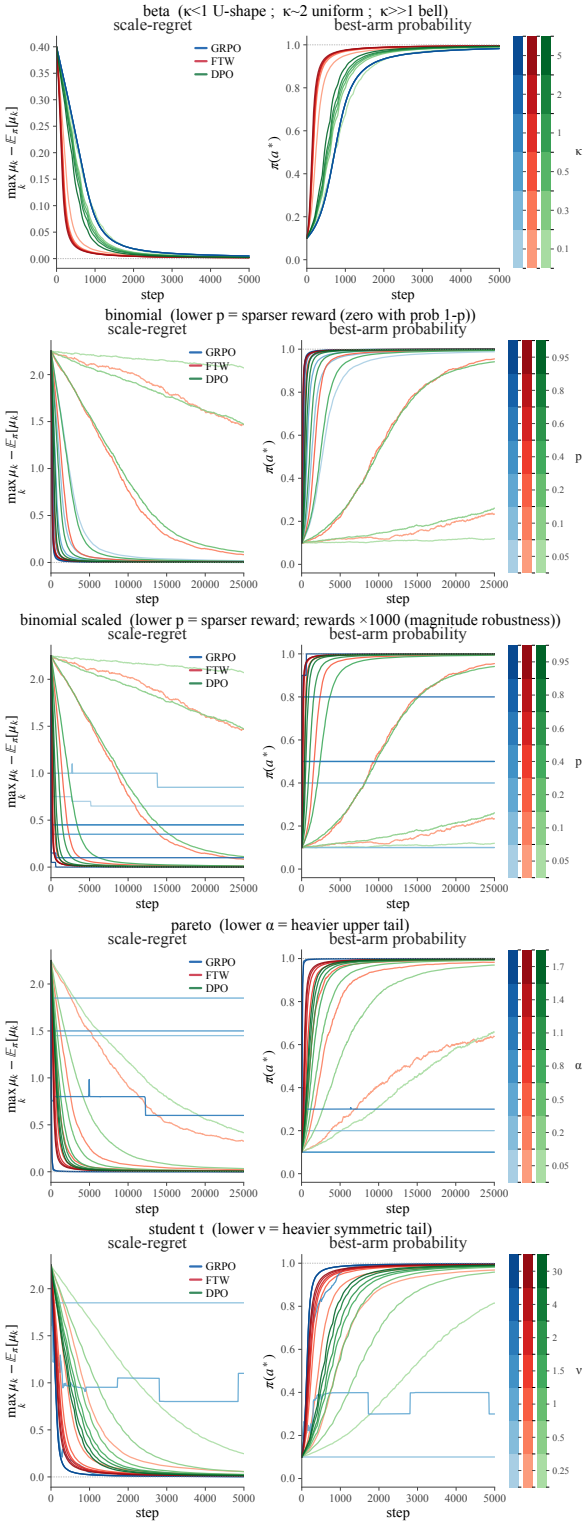


Figure 7: Bandit sweep across five return distributions (Beta, Binomial, scaled Binomial, Pareto, Student-t). Left column shows the scale-regret $\max_k \mu_k - \mathbb{E}_\pi[\mu_k]$, right column shows the probability of selecting the optimal arm. Color encodes the distribution-shape parameter, and we use the same legend as our Pareto-α result in the main text.

only on the order statistic. This confirms that the destabilization of GRPO under heavy tails is not specific to the one-sided support of Pareto.

Across all five rows, the rank-based potentials never destabilize, while GRPO destabilizes precisely in the regimes where the moment-generating function diverges. This matches the prediction of Proposition 2. The cardinal-exponential potential has an undefined fixed point under heavy tails, while rank-based potentials depend only on the order statistic and are insensitive to tail behavior.

H.4 State-Aliasing under Learning-Rate Variation

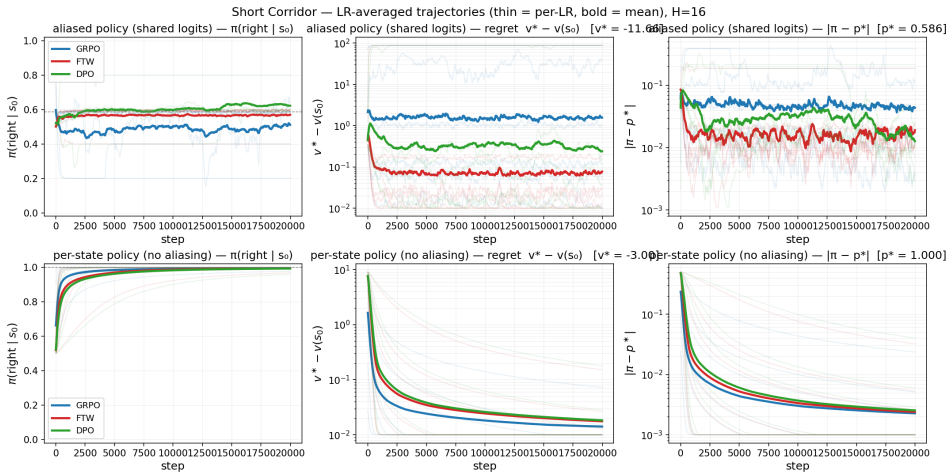


Figure 8: State-aliasing (corridor) task with a learning-rate sweep at horizon $H = 16$. We plot average L1 distance $|\pi(a) - \pi^*(a)|$ between the learned and optimal action distributions, averaged across learning rates within each method. Lower is better.

Figure 8 extends the state-aliasing result of Figure 2 (right) by sweeping the learning rate. The main text reports a single curve per method. Here we average across a learning-rate population to verify that FTW's lower L1 distance is not a property of a favorable learning-rate choice. The ordering FTW $<$ DPO, GRPO holds across the swept range, supporting the claim that polynomial concentration retains more useful stochasticity under partial observability than exponential concentration, independently of step-size tuning.

H.5 FTW under Tie-Saturated Rewards

This subsection briefly characterizes the effect of tie-breaking when returns frequently saturate around similar values, which is an important discussion item as it eliminates the discriminative signal that elite-filtering is supposed to provide. This is common under Binomial returns or sparse discrete reward settings, like our LLM post-training on SearchR1. We isolate the effect in a two-armed bandit with Bernoulli returns $G \in \{0, 1\}$ and success probabilities $p_1 = 0.999$ and $p_2 < p_1$, whose difference defines the gap $\delta \triangleq p_1 - p_2$, swept over $\delta \in \{0.099, 0.009, 0.004, 0.001, 10^{-4}\}$. FTW is configured as in the main text with $C = 2$, $K = 1$, $J = 32$, and we ablate the buffer's maximum sample age h, where $h = \infty$ never deletes data. On-policy GRPO is used as a reference.

Mechanism. Consider a single FTW update in this setting. When both arms succeed with probability close to one, the two sampled returns in a pairwise comparison are almost always 1 and 1, which makes the arms close to indistinguishable. To make this precise, let $\hat{q}(a)$ denote the buffer-occupancy of arm a. The probability that arm a wins from b under the sketched Bernoulli is $p_a(1 - p_b)$, and suppose it wins half of the tied cases giving $\Pr(a \text{ beats } b) = \frac{1}{2}(1 + p_a - p_b)$. This gives the general tilting distribution as a special case of Eq. (5),

$$\Pr(\text{elite} = a) \;\propto\; \hat{q}(a)\,\big(1 + p_a - \mathbb{E}_{b\sim\hat{q}}[p_b]\big). \tag{62}$$

Where $\mathbb{E}_{b\sim\hat{q}}[p_b] = \hat{q}(1)\,p_1 + \hat{q}(2)\,p_2$ is the average success probability of a random buffer sample. In expectation, the better arm receives the factor $1 + \hat{q}(2)\,\delta$ and the worse arm the factor $1 - \hat{q}(1)\,\delta$.

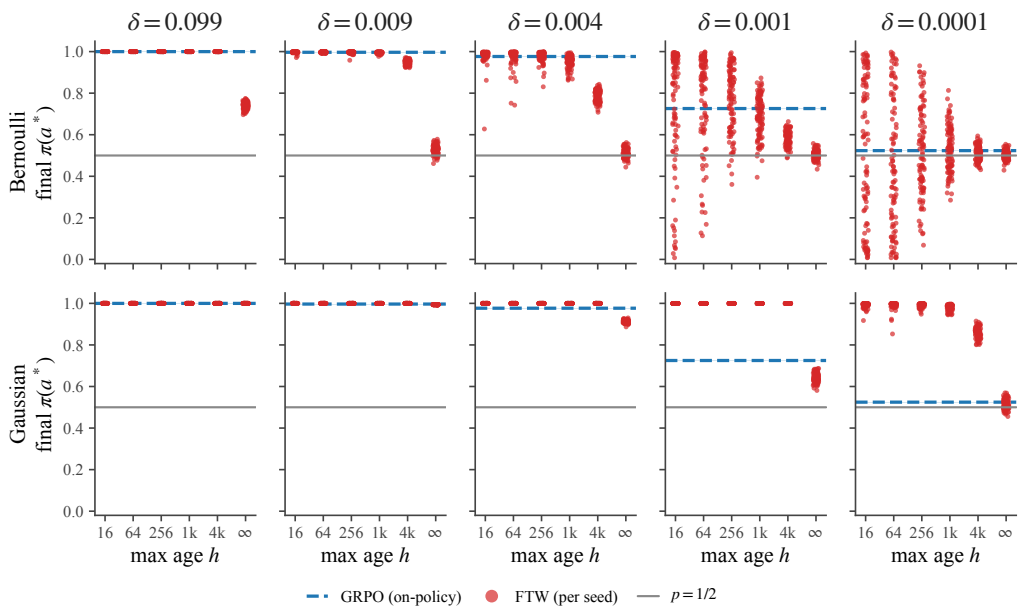


Figure 9: Population solutions after $T = 8 \times 10^5$ steps of training vs. maximum sample age h (100 seeds). Dashed line: on-policy GRPO mean and 95% CI. The top row shows results under Bernoulli returns and the bottom row under homoscedatic Gaussian noise $\mathcal{N}(\mu_a, 0.01)$ returns.

Selection therefore prefers the better arm, but only by a factor of at most $1 + \delta$ regardless of the buffer contents, so under saturation elite-filtering is almost indistinguishable from the buffer occupancy.

Under saturation, learning then entails imitating the buffer plus an $\mathcal{O}(\delta)$ nudge toward the better arm. Thus, the policy can only improve as fast as the buffer itself moves. With a finite sample age h, old data expires, the buffer tracks the recent and improving policy, and the nudge compounds over updates. With $h = \infty$ nothing expires. Each new batch is then a vanishing $1/t$ fraction of the buffer, so the policy is perpetually trained to imitate its own near-uniform early history.

GRPO does not inherit this property for a simple reason. Its advantage $G_i - \bar{G}$ is exactly zero on a tied group, so where elite filtering must imitate, GRPO simply performs no update and waits for an informative batch. Thus, FTW's learning dynamic is to oscillate around indistinguishable winners, whereas GRPO stays put.

Result. Figure 9 shows the population of solutions across buffer sizes, and Figure 10 shows the corresponding average learning curves. Under Bernoulli returns (top row), the never-delete buffer anchors FTW at chance for small δ, and only the large gap $\delta = 0.099$ produces a marginal shift toward the optimal arm, as the tilt in Eq. (62) scales with δ. Finite buffers avoid the anchor because dropping old samples lets the policy improvement compound, but smaller buffers turn tie-breaking into selection and update noise, whereas GRPO's tied advantages vanish and its learning is noticeably more stable. Under Gaussian returns (bottom row), every finite buffer for FTW finds the optimal arm whereas GRPO's learning signal seems to get stale for small δ.

The key takeaway is that tie-breaking is inherent to any RL algorithm on saturated rewards, but the ordinal tilting of methods like FTW induces different learning dynamics than typical marginal methods. The results present a buffer-size trade-off: larger buffers reduce selection noise and in expectation resolve ties through more accurate non-parametric estimation, at the cost of smoothing and slower policy movement.

H.6 Epistemic Contextual Bandit

This subsection is a case study in extracting the *maximum* policy improvement that a selection operator and replay buffer allow. The mirror-descent update of Eq. (56) sets the learning rate and

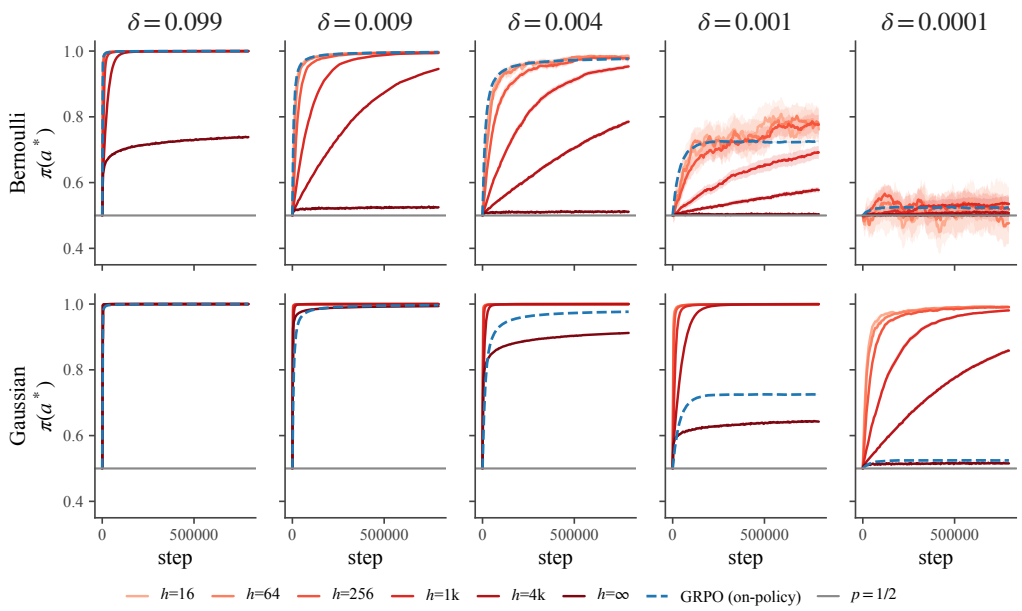


Figure 10: Mean learning curves for Figure 9 (shaded: 95% CI). Evidently, convergence takes proportionally longer as δ decreases.

the KL weight exactly as derived through our theory, eliminating the need for tuning. At least this goes for FTW and DPO, GRPO in contrast cannot be instantiated as a graphical model and its centred-advantage loss admits no posterior precision update rule that yields a positive-definite covariance (Appendix F). Thus, GRPO is run in three imperfect variants to approximate FTW's and DPO's learning rule, one version where normal GRPO is tuned for a learning rate, one where we assume that the covariance is learned through FTW's update rule (ONG), and one where we use the local Fisher information as a gradient preconditioner (NPG). The environment is the linear contextual bandit of Appendix G.4, rewards are deterministic and bounded, every arm has the same marginal $\mathbb{E}_x[r(x,a)] = 1/2$, so all uncertainty is epistemic and positional structure needs to be exploited for low regret.

The reason why we structured the learning rule and policy for this case study in this particular way is because the explicit Gaussian belief and the bootstrapping from the replay buffer form an analogous stand-in to how data is generated and resampled in typical LLM post-training. The LLM predictive (policy) behaves like an amortized posterior predictive that inserts its samples into the buffer. Resampling from that buffer exposes implicitly what the model does and does not know, at least structurally. This case-study then tests: *given* an agent whose uncertainty is represented explicitly and changing over time, does elite filtering act on that uncertainty and reduce it over time? An affirmative answer would hint at the idea that our framework and instantiation through FTW induces knowledge-seeking in LLM post-training. This question is also different to plain in-context learning, as data shouldn't only reduce posterior uncertainty, they should also concentrate on the optimal policy.

Results. Figure 11 shows the learning curves of the main configurations, we show regret-metrics alongside the belief and predictive diagnostics. Then Figure 12 sweeps each method's concentration parameters (top row) and the buffer depth (bottom row).

One somewhat obvious finding is that too little selection pressure never concentrates the policy, and too much prematurely converges on early winners before the belief has identified them, so regret decreases faster for intermediate selection pressure under each methodology (Figure 12, top row). The effect is of similar size for FTW and DPO, roughly a fourfold spread in final simple regret across their knobs (Figure 12, top row). In general, FTW and DPO are statistically indistinguishable at their best parameter-configurations.

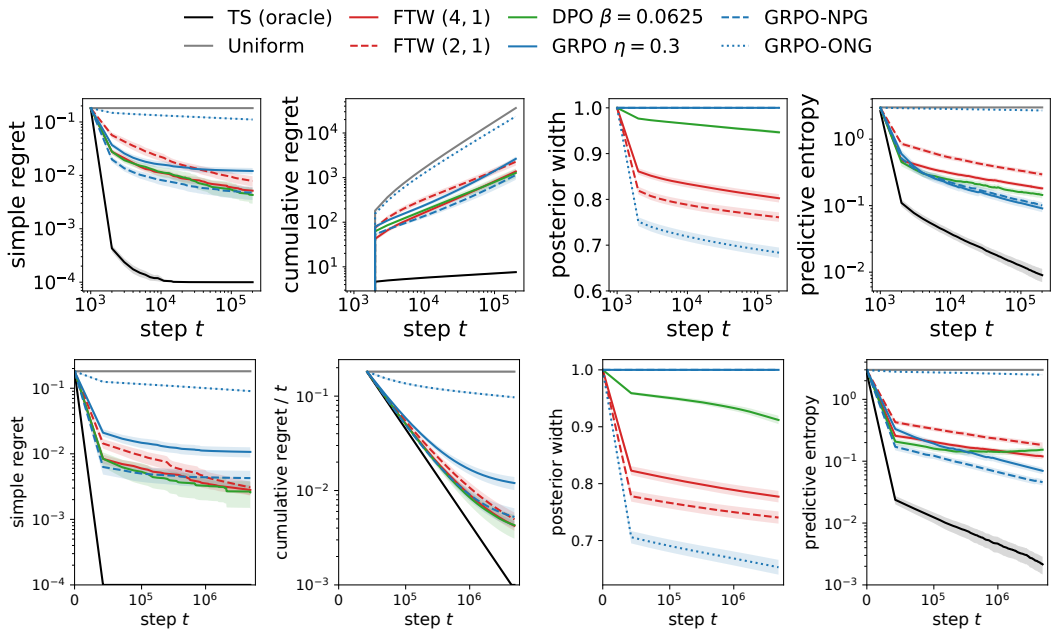


Figure 11: Learning curves at $h = 128$ (30 seeds, mean and Gaussian 95% bands). Left to right: offline simple regret of the posterior predictive, cumulative regret divided by t, posterior width, and predictive entropy. Colors denote the method family and line styles its variants. Thompson sampling is an oracle handed the true linear-Gaussian structure whereas the policy methods must learn it through the variational learning rule.

What is also interesting is how the belief precision differs between FTW and DPO, despite achieving similar regret. The posterior width for FTW decreases more sharply than for DPO, and its predictive entropy ends lower as well (right of Figure 11). At equal regret, DPO thus retains a wider belief and a more diffuse predictive than FTW. GRPO's belief metrics are either stale or overcontract due to the model misspecification or mismatch that prevents implementing a proper update rule in our framework.

Plain GRPO converges prematurely at every fixed step size, whereas the Fisher preconditioner of GRPO-NPG removes most of this gap in both simple and cumulative regret (Figure 11). The misspecified FTW precision update used by GRPO-ONG contracts the belief before the mean has moved, so its policy stays near uniform and it achieves the worst regret of all tested methods aside from the uniform upper bound. Finally, the buffer-depth ladder (Figure 12, bottom row) shows FTW insensitive to h over three orders of magnitude, whereas DPO degrades sharply at $h = 1024$ and GRPO is flat at its higher regret level.

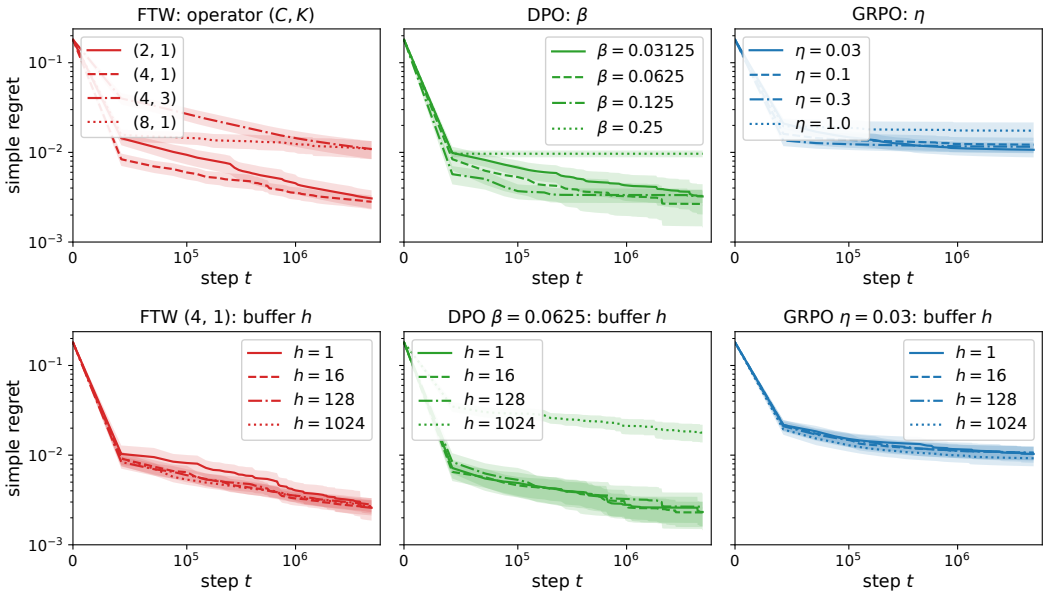


Figure 12: Offline simple regret (30 seeds, Gaussian 95% bands). Top row: each method's concentration knob at $h = 128$. Bottom row: the buffer-depth ladder at each method's best knob setting.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We specifically hypothesize desiderata for an efficient RFT algorithm and we base our derivation and experiments around this. Furthermore, all theoretical properties that we state are tested in small-scale Bandit setting to confirm algorithm behavior without the LLM complexity.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated Limitations paragraph in Section 8 covering the risk-seeking offset under aleatoric noise, the memory overhead of the replay-buffer proposal, and the total-ordering (IIA) assumption behind the elite filter.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All claims and theoretical results are supported by either a proof sketch in the main paper accompanied with a complete proof in the appendix, or a standard derivation and explicit statement of assumptions in the main paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide clear pseudocode and document the environments, datasets, hyperparameters, and setup needed to obtain our results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not release code, models, or data at this time. All environments and datasets are public and the setup needed to reproduce our results is documented well in Appendix G.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the general picture of the experiments in the main text and extensively document hyperparameters, dependencies, datasplit and sizes, and objectives in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Each figure with results explicitly states how confidence bars are constructed and the number of repetitions. We do not make any claims of statistical significance that are not supported by our data.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide an explicit account of all hardware needed (GPU, CPU, RAM, and SSD), along with approximate running costs of our provider in Appendix section G.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work uses public models, environments, and datasets, and involves no human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a broader impact discussion in Appendix A.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We release no data or models, the post-trained model weights are not released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all forked repositories (verl, verl-agent), environments and datasets used for evaluation (Brax, Sokoban, Search-R1), and pre-trained models used in our experiments (Qwen2.5 lineage). All assets are released under permissive research licenses (Apache 2.0 for Qwen2.5, verl, verl-agent, and Brax; MIT for gym-sokoban; CC-BY-SA for the Wikipedia-18 retrieval corpus and Natural Questions / HotpotQA derivatives used in Search-R1) and our use is consistent with their terms. Software versions in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe how we used LLMs in this manuscript in the Appendix section A.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.