VARIANCE-REDUCED OFF-POLICY PREFERENCE OPTIMIZATION VIA REGULARIZER CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct Preference Optimization (DPO) regularizes a learned policy toward a reference policy using an implicit reverse KL divergence. Other convex regularizers are also well motivated for preference optimization but are known to perform worse in practice – especially under off-policy training where the preference data does not come from the policy currently being optimized. Moving beyond bandit setting to multi-step MDPs, this effect gets exacerbated by the fact that constructing an implicit reward function for convex regularizers in multi-step MDPs requires sampling subsequent actions from the current policy, causing estimation errors to compound along trajectories. We formally characterize the regularizers that do not require subsequent action sampling and prove that the reverse KL is unique in providing a closed-form solution. To address error compounding in convex regularizers other than the reverse KL, we propose a regularity condition that minimizes the variance of the implicit reward estimator and derive a canonical form satisfying this condition by design. Our experimental results on tabular MDPs and for large language model post-training show that the proposed correction consistently improves on alternative approaches to estimating separable convex regularizers.

1 INTRODUCTION

How should we apply regularizers to achieve successful preference optimization? As reinforcement learning (RL) with a KL-divergence penalty against a reference policy has become the dominant paradigm for stable policy optimization in language model post-training and reasoning—from maximum-entropy RL and soft actor-critic methods (Schulman et al., 2017; Haarnoja et al., 2018; Geist et al., 2019) to RL from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023; Kim et al., 2026)—it is natural to ask whether there is a structural reason behind this choice or whether alternative regularizers could work better.

Notably, several prior works have generalized KL-regularized RLHF to broader classes of regularizers, including f-divergences (Wang et al., 2024; Belousov & Peters, 2019), Bregman divergences (Tomar et al., 2022; Banerjee et al., 2005), and general convex regularizers (Geist et al., 2019). These formulations have analogous soft Bellman operators, Legendre–Fenchel duals, and mirror-descent updates. Recently, Wang et al. (2024) reported that reverse KL attains the highest alignment accuracy among the f-divergences they compare, but no structural reason for this advantage has been identified.

In this paper, we show that there is in fact a structural reason for this underperformance, namely that alternative divergences underperform due to a high variance in their estimation of implicit rewards. More precisely, we show that a high variance in reward estimation originates in *off-policy* implementations used in modern preference optimization. Although the underlying objectives are on-policy by construction, practical RLHF systems rely heavily on off-policy compromises to make training computationally feasible. For instance, rollouts may be generated asynchronously by stale actors (Noukhovitch et al., 2025), filtered or reranked before training (Dong et al., 2023), and reused across multiple optimization steps (Rafailov et al., 2023). Since rollout generation is computationally expensive, and every extra forward pass of the policy adds to that cost, the logged data are reused across updates. Consequently, learning solely from the realized tuples (s, a, s') introduces a statistical discrepancy between the logged data and the current policy, and this discrepancy leads to inaccurate and unstable gradient update.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

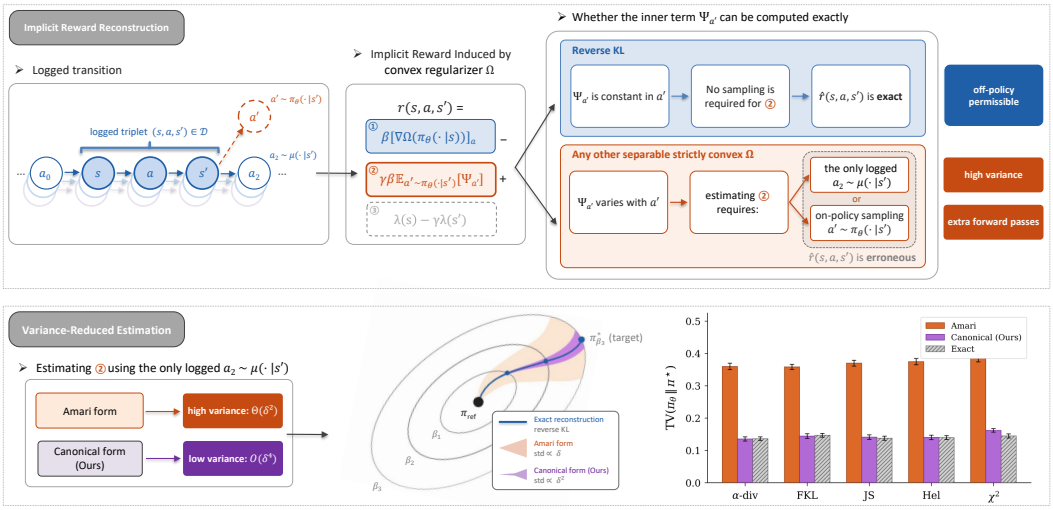


Figure 1: **The implicit reward in the multi-step MDPs and the variance reduction from the proposed method. (Top)** In the multi-step setting, the implicit reward $r(s, a, s')$ can no longer be reconstructed from the logged transition alone. Since an inner term $\Psi_{a'}$ (orange box) appears that requires a next-action $a' \sim \pi_\theta(\cdot \mid s')$, we must substitute the only logged a_2 or spend extra forward passes on on-policy samples. **(Bottom)** For every other separable convex regularizer, we propose the *canonical form*, a correction to the commonly used Amari form. It reduces the variance of the inner-term estimate, improving recovery of the policy when only a single logged sample is used.

In this paper, we show that the implicit reward under a general convex regularizer in the multi-step setting splits into a term evaluated at the logged transition and an *inner term* that depends on a next-action drawn from the learning policy. As Fig. 1 shows, the logged transition does not contain such an action, and generating one during training sacrifices the data reuse that makes off-policy learning efficient. Therefore, implementations use the next action recorded in the logged trajectory instead, and the resulting estimation error accumulates along the trajectory. Under this constraint, we prove that reverse KL is the unique separable convex regularizer whose implicit reward requires no sampling of the next-action a', whereas for every other regularizer the inner term varies with a'.

To be precise, this property belongs to the *estimator* of the implicit reward rather than to the divergence itself. Specifically, an f-divergence is built from a convex function f of the policy ratio $u := \pi_\theta / \pi_{\text{ref}}$, and f is determined only up to an additive multiple of $u - 1$. Thus, different choices of f yield different estimates of the implicit reward for the same divergence. For instance, both $u \log u$ and $u \log u - (u - 1)$ induce the reverse KL, but only the estimator built from the first requires no extra sample. In practice, implementations of f-divergences use the form (Amari, 2009), for which $f'(1) = 0$. DPO instead uses the reverse KL generator $u \log u$ that satisfies $f'(1) = f''(1)$. We argue that this difference in the estimator explains why the other convex regularizers underperform. Motivated by this insight, we propose $f'(1) = f''(1)$ as the *regularity condition* that minimizes the variance of the implicit-reward estimate. We then derive a *canonical form* satisfying this condition from f-divergence to general separable convex regularizer, and call the reparametrization that produces it the *canonical correction*.

We summarize our main contributions as follows:

1. **The inner term and its accumulating variance.** In the multi-step setting, the implicit reward contains an inner term whose single-sample estimation errors accumulates along a trajectory. We prove that reverse KL is the unique separable convex regularizer where the variance is zero.
2. **Canonical correction.** For other separable convex regularizer, we show that suppressing the variance in a single-sample off-policy update leads to a regularity condition on the estimator. We also characterize which separable convex regularizers admit a canonical correction.
3. **Validation from tabular MDPs to large language models.** Reparametrizing convex regularizers into their canonical forms consistently improves performance across tabular MDP experiments, Arena-Hard, AlpacaEval, and mathematical-reasoning benchmarks.

Table 1: The six convex regularizers in two distinct representations. The middle column is the *Amari form* with $f'(1) = 0$, and the right column is the *canonical form* $f^{\mathrm{can}}(u) = f(u) + c^\star(u-1)$, which we propose for off-policy training. Each row substitutes that their own shift $c^\star = f''(1) - f'(1)$. Both columns induce the same Ω and β-optimal policy, and differ only in the inner-term estimator.

Divergence	**Amari form** $f(u)$	**Canonical form** $f^{\mathrm{can}}(u)$
Reverse KL (RKL), $\alpha \to 1$	$u \log u - (u-1)$	$f_{\mathrm{RKL}}(u) + (u-1)$
Forward KL (FKL), $\alpha \to 0$	$(u-1) - \log u$	$f_{\mathrm{FKL}}(u) + (u-1)$
α-divergence, $\alpha \in \mathbb{R} \setminus \{0,1\}$	$\dfrac{u^\alpha - \alpha u + \alpha - 1}{\alpha(\alpha-1)}$	$f_\alpha(u) + (u-1)$
Jensen–Shannon	$\frac{1}{2}\left[u \log u - (u+1) \log \frac{u+1}{2}\right]$	$f_{\mathrm{JS}}(u) + \frac{1}{4}(u-1)$
Pearson χ^2, $\alpha = 2$	$(u-1)^2$	$f_{\chi^2}(u) + 2(u-1)$
Squared Hellinger, $\alpha = \frac{1}{2}$	$(\sqrt{u}-1)^2$	$f_{\mathrm{Hel}}(u) + \frac{1}{2}(u-1)$

2 Notation and Background

Notation. We consider a Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $\mathbb{P}(s'|s,a)$, discount $\gamma \in (0,1]$, and unregularized reward r. In our token-level MDP formulation of language model post-training, states are token prefixes, actions are next tokens, and transitions are deterministic. Regularization is specified by a fixed reference policy π_{ref}, a regularization weight $\beta > 0$, and a regularizer $\Omega(\cdot;\, \pi_{\mathrm{ref}}(\cdot|s))$ that is strictly convex in its first argument. Throughout, we take Ω to be *separable*,[1] meaning that it decomposes across actions as

$$\Omega\big(\pi(\cdot|s);\, \pi_{\mathrm{ref}}(\cdot|s)\big) = \sum_{a \in \mathcal{A}} \pi_{\mathrm{ref}}(a|s)\, \Omega_a(u_a), \qquad u_a := \frac{\pi(a|s)}{\pi_{\mathrm{ref}}(a|s)}, \tag{1}$$

for per-action convex functions Ω_a. The gradient is then the per-action derivative $[\nabla_\pi \Omega]_a = \Omega_a'(u_a)$. In particular, an f-divergence is a special case of Eq. 1 in which all actions share the same convex function, $\Omega_a = f$ for every $a \in \mathcal{A}$,

$$D_f\big(\pi \,\|\, \pi_{\mathrm{ref}};\, s\big) := \sum_{a \in \mathcal{A}} \pi_{\mathrm{ref}}(a|s)\, f(u_a), \tag{2}$$

for strictly convex f with $f(1) = 0$. The *generator* f (or $\{\Omega_a\}$) is not unique, because the map $f \mapsto f + c\,(u-1)$ leaves D_f unchanged on the simplex. A generator is therefore one representative of this family, and the one with $f'(1) = 0$ is the *Amari form* of information geometry (Amari, 2009). Existing work on f-divergence regularizers uses this representative (Belousov & Peters, 2019; Wang et al., 2024; von Werra et al., 2020),[2] and Table 1 lists the expressions for the six generators we study.

Ω-regularized RL objective and the β-optimal policy. In Ω-regularized RL, the objective and its soft Bellman operator are expressed as

$$J(\pi) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t,a_t) - \beta \Omega(\pi(\cdot|s_t);\, \pi_{\mathrm{ref}}(\cdot|s_t))\big)\right], \tag{3}$$

$$\mathcal{T}_\Omega V(s) := \max_{\pi(\cdot|s) \in \Delta(\mathcal{A})} \mathbb{E}_{a\sim\pi}\big[r(s,a) + \gamma \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V(s')]\big] - \beta \Omega(\pi(\cdot|s);\, \pi_{\mathrm{ref}}(\cdot|s)). \tag{4}$$

The unique fixed point of $\mathcal{T}_\Omega$ is the regularized optimal value V^*, with action-value $Q^*(s,a) := r(s,a) + \gamma \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V^*(s')]$. The *$\beta$-optimal policy* π^* is the per-state maximizer in Eq. 4 at $V = V^*$, and it is unique because Ω is strictly convex.

[1] Since π_{ref} is fixed, Eq. 1 is a sum of single-coordinate functions of $\pi(a|s)$, which is *separability* in the standard convex-analysis sense (Boyd & Vandenberghe, 2004).

[2] Reverse KL and forward KL are the exceptions in Wang et al. (2024), whose table states them as $u \log u$ and $-\log u$, neither of which satisfies $f'(1) = 0$. The four f-divergence branches of the TRL implementation (von Werra et al., 2020) are each parametrized by an f' with $f'(1) = 0$.

Direct Preference Optimization. DPO (Rafailov et al., 2023) reparametrizes the KL-regularized objective in the bandit setting where preferences directly update the policy without a separate reward model. Its data consist of preference pairs (x, y_w, y_l), where a prompt x together with a preferred y_w and a dispreferred response y_l. Under a Bradley–Terry model with unknown reward $r^\star$, it models the preference as $\mathbb{P}(y_w \succ y_l \mid x) = \sigma(r^\star(x, y_w) - r^\star(x, y_l))$, where $\sigma(z) := (1 + e^{-z})^{-1}$. Hence, the implicit reward under which π_θ is optimal then takes the closed form

$$r_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x).$$

Here, the partition function $Z(x)$ depends on the prompt alone. Hence, it can be cancelled out inside the pairwise difference and leaves a loss to be purely written in policy log-ratios. While this cancellation is what makes DPO practical, Cho et al. (2025) and Kim et al. (2026) provided necessary and sufficient conditions for a closed-form implicit reward in multi-step setting, where the cancellation is no longer available.

Multi-step setting. The bandit setting treats a whole response as one action, so it has no intermediate states and therefore every token receives the same credit signal. In the multi-step setting, however, a next-state exists and the implicit reward now depends on its value $V^*(s')$. Unlike $Z(x)$, this dependence generally does not cancel in the trajectory-level preference difference. Hence, several tasks such as multi-turn dialogue, tool use, and long reasoning traces, the key cancellation no longer holds. To address this challenge, we analyze a multi-step MDP regularized by Ω, where the implicit reward induces a term which is absent from the bandit view.

3 Variance Reduction for Off-Policy Preference Optimization

We first characterize the implicit reward that makes a given policy optimal and identify the inner term that appears in it once the setting is multi-step rather than bandit (Sec. 3.1). We then determine which regularizer does not require next-action sampling (Sec. 3.2), and finally bring the remaining regularizers into the same family (Sec. 3.3).

3.1 Structural form of the β-optimal reward and its inner term

Lemma 1 (Structural form of the β-optimal reward)**.** *Assume $\Omega(\cdot; \pi_{\text{ref}})$ is essentially smooth in its first argument, so that the β-optimal policy π^* lies in the interior of $\Delta(\mathcal{A})$ at every state. Then π^* is β-optimal under the reward r if and only if there exists a state-dependent function $\lambda : \mathcal{S} \to \mathbb{R}$ such that, for every (s, a),*

$$Q^*(s, a) = \beta[\nabla_\pi \Omega(\pi^*(\cdot|s); \pi_{\text{ref}}(\cdot|s))]_a + \lambda(s), \tag{5}$$

$$r(s, a) = \beta[\nabla_\pi \Omega(\pi^*(\cdot|s); \pi_{\text{ref}}(\cdot|s))]_a + \lambda(s) - \gamma \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[V^*(s')]. \tag{6}$$

Lemma 1 states that π^* determines the reward under which it is itself optimal. Up to a state-only potential $\lambda(s)$, which corresponds to the shaping function of Ng et al. (1999) and leaves the induced optimal policy invariant, the optimal action-value function $Q^*(s, a)$ is determined by the gradient of Ω at π^*, and the reward follows from a one-step Bellman backup. For an f-divergence, the gradient is $f'(u_a)$. So the reverse KL $f(u) = u \log u$ yields $f'(u) = \log u + 1$, and the constant can be absorbed by λ, which recovers the log-ratio form of Kim et al. (2026). A proof is given in Appendix B.

Given the structure of optimality for a convex regularizer, we can now ask what it costs to estimate the implicit reward. Fixing π_{ref} and writing $\Omega(\pi)$ for $\Omega(\pi; \pi_{\text{ref}})$, Eq. 6 becomes practical once $V^*(s')$ can be written in terms of π^*. Substituting Q^* into the soft Bellman identity gives

$$\begin{aligned} V^*(s') &= \mathbb{E}_{a' \sim \pi^*(\cdot|s')}[Q^*(s', a')] - \beta\Omega(\pi^*(\cdot|s')) \\ &= \beta\Big(\langle \pi^*(\cdot|s'),\, \nabla\Omega(\pi^*(\cdot|s'))\rangle - \Omega(\pi^*(\cdot|s'))\Big) + \lambda(s') \\ &= \beta \underbrace{\mathbb{E}_{a' \sim \pi^*(\cdot|s')}\big[\Psi_{a'}(u_{a'})\big]}_{\text{inner term}} + \lambda(s'), \qquad \Psi_a(u) := \Omega'_a(u) - \frac{\Omega_a(u)}{u}. \end{aligned} \tag{7}$$

The last step writes the bracket as an expectation over actions of the per-coordinate integrand Ψ_a, and we call that expectation the *inner term*. It is the only part of $V^*(s')$ that requires a next-action $a' \sim \pi^*(\cdot|s')$. Substituting back, the implicit reward of a parametrized policy π_θ reads

$$r_\theta(s,a) = \underbrace{\beta[\nabla_{\pi_\theta}\Omega(\pi_\theta(\cdot|s))]_a + \lambda(s)}_{\text{(i)}} - \underbrace{\gamma\mathbb{E}_{s'}\left[\beta\,\mathbb{E}_{a'\sim\pi_\theta(\cdot|s')}\left[\Psi_{a'}(u_{a'})\right] + \lambda(s')\right]}_{\text{(ii)}}. \tag{8}$$

Term (i) is evaluated at the logged action a and corresponds directly to the log-ratio in DPO, while the shaping term depends on s alone. Term (ii) is added by the multi-step setting, and it contains the inner term, which in general has to be estimated by Monte Carlo,

$$\widehat{\Psi}^{(n)}(s') := \frac{1}{n}\sum_{i=1}^{n}\Psi_{a'_i}(u_{a'_i}), \qquad a'_i \sim \pi_\theta(\cdot|s'), \tag{9}$$

drawing n fresh actions from the current policy at every reachable next-state.

Off-policy permissible regularizers. The sampling required by Eq. 9 costs additional forward passes of the current policy at every logged transition. In practice, the logged data do not support a next-action sampled from the current policy. Implementations therefore use the sample that was already realized instead. Interestingly, this is the same motivation that leads from on-policy SARSA to off-policy Q-learning (Rummery & Niranjan, 1994; Watkins & Dayan, 1992), and by analogy we call a regularizer that requires no additional sampling *off-policy permissible*. The implicit reward of such a regularizer can be reconstructed from logged data alone.

Definition 1 (Off-policy permissibility)**.** A separable convex regularizer Ω is *off-policy permissible* if its inner-term integrand is constant, *i.e.,* $\Psi_{a'}(u_{a'}) \equiv c$ for every action a' and every π.

3.2 Reverse KL is the only off-policy permissible regularizer

We now ask which regularizer has a constant inner-term, and prove that reverse KL is the unique answer. We consider a fixed next-state s' and abbreviate $\pi := \pi(\cdot|s')$ and $\pi_{\text{ref}} := \pi_{\text{ref}}(\cdot|s')$.

Theorem 1. *Suppose the per-state regularizer Ω satisfies the following for every $\pi_{\text{ref}} \in \Delta(\mathcal{A})$.*

- (A1) Coincidence. $\Omega(\pi_{\text{ref}};\,\pi_{\text{ref}}) = 0$.
- (A2) Strict positivity. $\Omega(\pi;\,\pi_{\text{ref}}) > 0$ *for every* $\pi \neq \pi_{\text{ref}}$.
- (A3) Strict convexity. $\Omega(\cdot;\,\pi_{\text{ref}})$ *is strictly convex on* $\Delta(\mathcal{A})$.
- (A4) Off-policy permissibility. Ω *satisfies Definition 1.*
- (A5) Separability. Ω *has the form of Eq. 1 for per-action convex functions* $\Omega_{a'}$.

Then at the fixed s',

$$\Omega(\pi(\cdot|s');\,\pi_{\text{ref}}(\cdot|s')) = c(s')\cdot D_{\text{KL}}\big(\pi(\cdot|s')\big\|\pi_{\text{ref}}(\cdot|s')\big), \tag{10}$$

where $c(s') > 0$ depends on s' only through $\pi_{\text{ref}}(\cdot|s')$.

Since the theorem is stated for a given π_{ref} and a fixed next-state s', the coefficient $c(s')$ is undetermined and remains the degree of freedom. Several preference-optimization objectives already in use involve in this class, differing only in how they set $c(s')$ and the shaping term $\lambda(s)$. For example, β-DPO (Wu et al., 2024), R-DPO (Park et al., 2024), SimPO (Meng et al., 2024), and ConfPO (Yoon et al., 2025) are the members of this class. Derivation details are in Appendix B.5.

To aid understanding, consider the case of f-divergences. Axiom (A4) forces the integrand $\Psi(u) = f'(u) - f(u)/u$ to be constant, and the only solution with $f(1) = 0$ is $f(u) = B\,u\log u$. At $f(u) = u\log u$, the implicit reward collapses to

$$r^{\text{KL}}(s,a) = \beta\log\frac{\pi^*(a|s)}{\pi_{\text{ref}}(a|s)} + \lambda(s) - \gamma\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[\lambda(s')], \tag{11}$$

which involves no expectation over a', so no next-action sampling is needed. In this sense, Theorem 1 extends the structure of the implicit reward identified by Cho et al. (2025) and Kim et al. (2026) to the separable convex class, and identifies reverse KL as the only regularizer in this class that requires no next-action sampling.

3.3 CANONICAL CORRECTION FOR A CONVEX REGULARIZER

Theorem 1 shows that reverse KL is the unique regularizer that needs no next-action sample, but it does not say what is left for others. Here, we relax off-policy permissibility from an exact requirement to a condition at the reference, and ask what condition an estimator of regularizer should satisfy. For notational simplicity, we write the next-action a' as a throughout this section. Formally, the quantity to minimize is the variance of the inner-term estimate, *i.e.,* $\mathrm{Var}_{a\sim\pi}\big[\Psi_a(u_a)\big]$, which off-policy permissibility sets to zero.

Maps that preserve the divergence. We can leverage that a regularizer does not uniquely determine its own estimator. Different representations of the same Ω preserve the objective and its β-optimal policy, but can yield estimators with different variances Recall the integrand $\Psi_a(u) = \Omega_a'(u) - \Omega_a(u)/u$, and assume each Ω_a is smooth near $u = 1$. Every other family representing the same Ω on $\Delta(\mathcal{A})$ can be obtained from the generator $\{\Omega_a\}$ through a map T_μ, defined by

$$T_\mu : \{\Omega_a\} \longmapsto \{\Omega_a^\mu\}, \qquad \Omega_a^\mu(u) := \Omega_a(u) - \Big(\textstyle\sum_b \pi_{\mathrm{ref}}(b)\,\mu_b\Big)\,u + \mu_a, \tag{12}$$

for some vector $\mu \in \mathbb{R}^{\mathcal{A}}$. The map T_μ is objective-preserving, since

$$\sum_a \pi_{\mathrm{ref}}(a)\,\Omega_a^\mu(u_a) = \sum_a \pi_{\mathrm{ref}}(a)\,\Omega_a(u_a) \qquad \text{for every } \pi \in \Delta(\mathcal{A}), \tag{13}$$

so Ω and the β-optimal policy of Eq. 4 remain unchanged. However, what does change is the integrand, which becomes $\Psi_a^\mu(u) = \Psi_a(u) - \mu_a/u$. Thus, even when only the logged next-action is available, we can use an objective-preserving map to reduce the integrand's variance without extra forward passes or rollouts.

The correction that minimizes the variance. We now search among the objective-preserving maps T_μ in Eq. 12 for one that reduces the variance of the inner-term estimate. Definition 1 requires $\Psi_a^\mu(u_a)$ to be constant across actions and policies, which is possible only for reverse KL. For other regularizers, we instead relax this requirement to hold locally near the reference, where the policy is initialized. Expanding around $u_a = 1$ gives

$$\Psi_a^\mu(u_a) = \Psi_a^\mu(1) + (\Psi_a^\mu)'(1)\,(u_a - 1) + O((u_a - 1)^2).$$

For this Taylor expansion to be independent of the action near reference, the constant term $\Psi_a^\mu(1)$ must not depend on a, and the linear sensitivity to the action-dependent drift $u_a - 1$ must be eliminated. These requirements yield two conditions:

$$\text{(C1)}\ \ \Psi_a^\mu(1) \text{ is independent of } a, \qquad\qquad \text{(C2)}\ \ (\Psi_a^\mu)'(1) = 0 \ \text{ for every } a \in \mathcal{A}.$$

Under both conditions, the integrand differs from a constant only at second order in the drift $O((u_a - 1)^2)$, whereas violating (C2) leaves a dominant first-order term. We call a μ that satisfies both conditions a *canonical correction* for Ω.

Theorem 2. *Let $\psi_a := \Psi_a(1)$ and $\kappa_a := \Omega_a''(1)$. Then the following hold.*

(i) (Existence and uniqueness.) *A canonical form Ω^{can} exists if and only if κ_a is independent of a, i.e., $\kappa_a \equiv \kappa$. If it exists, it is unique and is given by $\mu_a = \psi_a - \kappa$ for every $a \in \mathcal{A}$.*

(ii) (Variance.) *Suppose that $\kappa_a \equiv \kappa$, and let $\delta := \|\pi - \pi_{\mathrm{ref}}\|_1$. Then, as $\delta \to 0$,*

$$\mathrm{Var}_{a\sim\pi}\big[\Psi_a^\mu(u_a)\big] = \begin{cases} O(\delta^4) & \text{if } \mu \text{ is the canonical correction,} \\ \Theta(\delta^2) & \text{if } \mu \text{ satisfies (C1) and violates (C2).} \end{cases}$$

Corollary 1 (f-divergences and the α-family)**.** *For $\Omega = \sum_a \pi_{\mathrm{ref}}(a)\,f(u_a)$ with f strictly convex and $f(1) = 0$, every Ω_a equals f, so part (i) applies with $\psi = f'(1)$ and $\kappa = f''(1)$. Hence $c^\star = f''(1) - f'(1)$, and the generator of canonical form f^{can} satisfies $(f^{\mathrm{can}})'(1) = (f^{\mathrm{can}})''(1)$.*

Remark 1. *Squared Euclidean has no canonical form. For a fixed π_{ref}, it is separable, but its curvature $\kappa_a = \pi_{\mathrm{ref}}(a|s)$ varies with the action, so it fails part (i) of Theorem 2.*

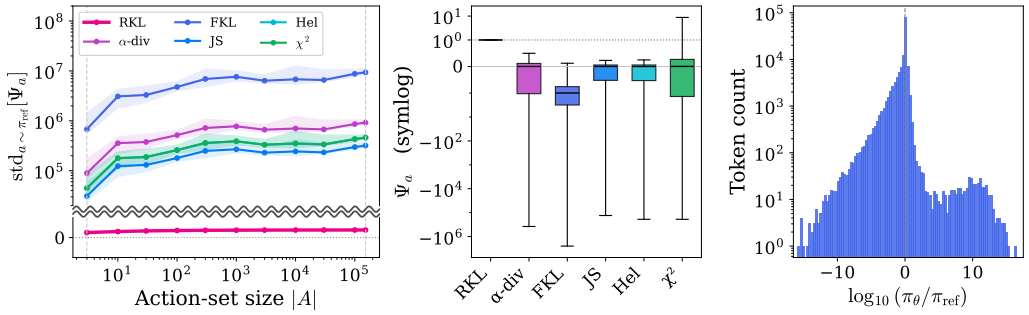


Figure 2: **Single-sample noise of the inner term. (Left)** Standard deviation of Ψ_a under one draw $a \sim \pi_{\text{ref}}$ against the action-set size $|\mathcal{A}|$ over 10 seeds. **(Middle)** Per-token integrand Ψ_a on the post-trained `Qwen3-1.7B` against `Qwen3-1.7B-Base` over 133k logged tokens of 493 UltraFeedback-Binarized test responses, on a symmetric log scale, showing the median, the interquartile, and the 1st and 99th percentiles. **(Right)** Distribution of the logged density ratio over the same tokens.

4 Experiments

Our experiments address two questions in order. First, is the inner term, rather than the divergence itself, the cause of the gap between reverse KL and the other regularizers in the off-policy setting? Second, once we adopt the canonical form instead of the conventional Amari form, does this gap remain? We first work on a toy layered MDP whose ground truth of optimal policy can be easily achievaable (Sec. 4.1), where we measure the noise of the inner term without any training and then train with the correction. We then post-train large language model and evaluate them on Arena-Hard (Li et al., 2025), AlpacaEval (Dubois et al., 2024), and on ten mathematical-reasoning benchmarks (Cobbe et al., 2021; Hendrycks et al., 2021; Lewkowycz et al., 2022) (Sec. 4.2). Appendix C to Appendix F give the setting and full results.

4.1 Toy layered MDP

We first run our experiments on a layered MDP with $|\mathcal{A}| = 3$ (See Fig. 6 in Appendix C.1), where the optimal policy of every regularizer can be computed exactly. The β-optimal policy $\pi^\star_\Omega$ of every regularizer is derived by backward induction. The inner term can be summed exactly over the action set, and the next-action can be drawn fresh from the current policy or read from the log. Consequently, the estimator can be evaluated in isolation from all other confounding factors. We measure the *recovery gap* in policy space using the occupancy-weighted total-variation distance

$$\Delta_\pi(\Omega) := \mathbb{E}_{s\sim d^{\pi^\star_\Omega}}\Big[\mathrm{TV}\big(\pi_\theta(\cdot\,|\,s),\, \pi^\star_\Omega(\cdot\,|\,s)\big)\Big] = \mathbb{E}_{s\sim d^{\pi^\star_\Omega}}\left[\frac{1}{2}\sum_a \big|\pi_\theta(a\,|\,s) - \pi^\star_\Omega(a\,|\,s)\big|\right], \quad (14)$$

averaged over 100 independent runs, where lower is better. Then, we evaluate performance in policy space rather than reward space for two reasons. First, Δ_π lies in $[0, 1]$ for every Ω, so it can be comparable across all regularizers. By contrast, a reward-space gap has a scale that depends on the gradient of Ω and the shaping freedom of Lemma 1. Second, $\Delta_\pi = 0$ holds only if π_θ equals $\pi^\star_\Omega$ on the support of the target, so a small value reflects the accurate recovery of the target policy.

The noise grows with the action set. Fig. 2 (Left) plots the noise of the inner term against the size of the action space. We increase $|\mathcal{A}|$ on a log scale from 3 to 152,000, the vocabulary size of the language model. The standard deviation of the single-sample integrand Ψ_a rises monotonically for the six divergences implemented with their conventional generators, whereas reverse KL implemented as $u \log u$ stays at zero for every action-set size. To verify whether this phenomenon scales to LLMs in practice, we repeat the same measurement on `Qwen3-1.7B`. Fig. 2 (Middle) shows the distribution of Ψ_a with the post-trained model as π_θ and the base model as π_{ref}, and every divergence other than reverse KL spans a range that reaches 10^5 and beyond. A closer look at Fig. 2 (Right) explains why. The policy density ratio u_a ranges from 10^{-15} to 10^{15}, and the samples in its tails make the $1/u$ term of the integrand Ψ_a diverge. Consequently, this severe noise growth at scale explains why conventional implementations fail under off-policy preference training.

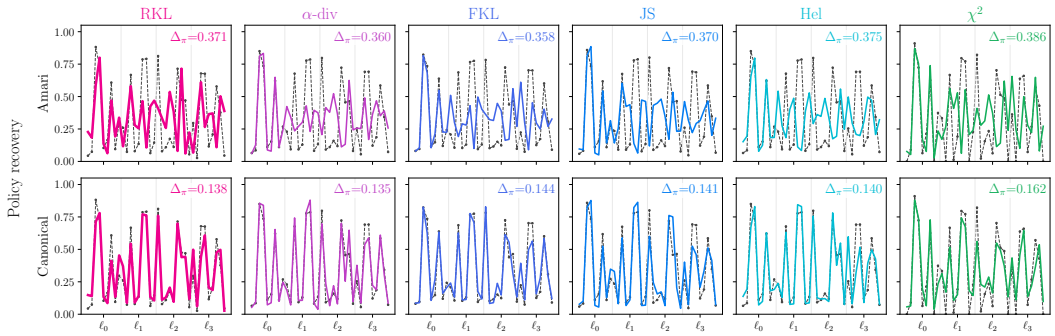


Figure 3: **Recovered policies under a single logged next-action.** Comparison between the trained policy π_θ (solid line) and the target policy $\pi^\star$ (dashed line) across the twelve (layer, state) cells of the layered MDP, evaluating six regularizers in the Amari form (top) versus the canonical form (bottom) at a target peak of 0.7.

Canonicalization improves the estimation of the optimal policy. We train the policy on the layered MDP from a single logged next-action. Fig. 3 compares the ground-truth optimal policy $\pi^\star_\Omega$ against the trained policy π_θ for each divergence, making the recovery gap visible across all states. Compared with the Amari form of the generator, the canonical form captures the shape of the target significantly better. While the Amari form tends to flatten out except at the highest peak, the canonical form captures multiple peaks more accurately, and the recovery gap falls from $0.358 \sim 0.386$ to between $0.135 \sim 0.162$. Prior work on f-divergence preference optimization using the Amari form interpret this flattening as a trade-off between diversity and accuracy (Wang et al., 2024). In our setting, however, this flattening is not an intended effect but an artifact of estimator misalignment, since the same divergence recovers the target once its generator is written in the canonical form. Representing an f-divergence in its canonical form therefore keeps the characteristics of the divergence while enabling an accurate policy learning.

The regularity condition of generator substantially affects policy recovery. Fig. 4 (left) sweeps the α-divergence family from forward KL ($\alpha \to 0$) to reverse KL ($\alpha = 1$) and Pearson χ^2 ($\alpha = 2$). Each curve represents the divergence family under the regularity condition on the generator, the Amari condition $f'(1) = 0$ and the canonical condition $f'(1) = f''(1)$. For either generator form, the recovery gap remains relatively consistent across α. Between the two families, however, the canonical form shows significantly smaller recovery gap throughout. This implies that the choice of regularity condition makes a substantial difference in performance. The arrow at $\alpha = 1$ highlights the two generators of reverse KL, $u \log u - (u - 1)$ and $u \log u$, and it shows that the advantage conventionally attributed to reverse KL is in fact the estimator rather than the divergence itself.

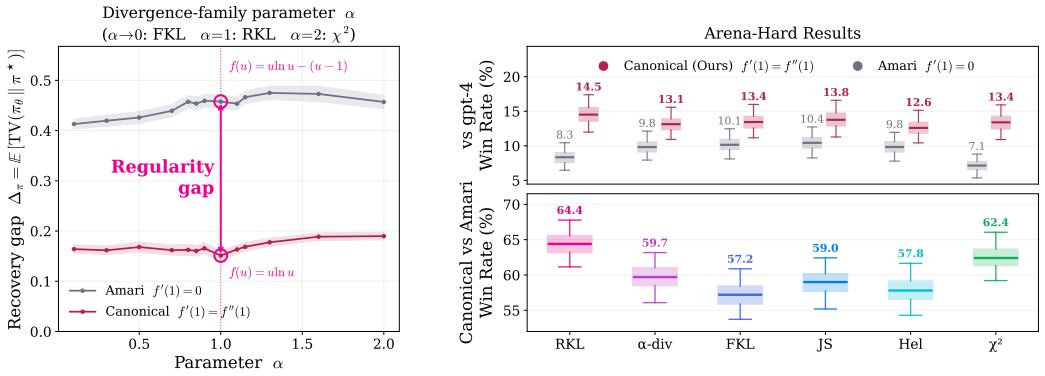


Figure 4: **Comparison between Canonical and Amari along a divergence family. (Left)** Recovery gap Δ_π across the α-divergence family in the Amari form and in the canonical form. The gap between the two reverse KL generators is marked at $\alpha = 1$. **(Right)** Arena-Hard v0.1 win rates for six divergences trained from `Qwen3-1.7B-Base`, shown against the `GPT-4` baseline (top) and direct comparison between canonical and Amari forms (bottom).

Table 2: **Pass@1 of ten mathematical reasoning benchmarks with `Qwen3-1.7B-Base`.** Amari and canonical (shaded) rows per divergence, the change from the base model in parentheses, and its average in the last column. All values are mean over three seeds.

Div.	Form	GSM8K	MATH	MATH-500	AIME24	AMC23	College	Gaokao	Minerva	Olympiad	Omni	Avg. Δ
Base	–	65.5	54.1	53.8	3.3	30.0	27.3	39.7	17.7	21.9	16.0	–
RKL	Amari	54.3 (−11.2)	49.0 (−5.1)	47.8 (−6.0)	4.4 (+1.1)	29.2 (−0.8)	24.4 (−2.9)	33.5 (−6.2)	14.7 (−3.0)	20.1 (−1.8)	14.5 (−1.5)	−3.7
	Canonical	**83.1 (+17.6)**	57.8 (+3.7)	60.5 (+6.7)	5.6 (+2.3)	26.7 (−3.3)	29.6 (+2.3)	47.1 (+7.4)	23.4 (+5.7)	23.7 (+1.8)	16.0 (0.0)	+4.4
FKL	Amari	54.8 (−10.7)	52.1 (−2.0)	52.3 (−1.5)	3.3 (0.0)	29.2 (−0.8)	26.6 (−0.7)	33.9 (−5.8)	13.2 (−4.5)	21.9 (0.0)	15.1 (−0.9)	−2.7
	Canonical	82.5 (+17.0)	61.1 (+7.0)	62.9 (+9.1)	3.3 (0.0)	34.2 (+4.2)	29.6 (+2.3)	49.2 (+9.5)	22.1 (+4.4)	25.0 (+3.1)	**18.1 (+2.1)**	+5.9
α-div.	Amari	56.6 (−8.9)	52.1 (−2.0)	51.4 (−2.4)	3.3 (0.0)	28.3 (−1.7)	26.6 (−0.7)	33.9 (−5.8)	14.5 (−3.2)	21.8 (−0.1)	15.0 (−1.0)	−2.6
	Canonical	82.2 (+16.7)	61.6 (+7.5)	62.7 (+8.9)	6.7 (+3.4)	32.5 (+2.5)	**30.1 (+2.8)**	49.3 (+9.6)	22.8 (+5.1)	**26.3 (+4.4)**	17.8 (+1.8)	+6.3
JS	Amari	62.1 (−3.4)	53.8 (−0.3)	53.7 (−0.1)	3.3 (0.0)	25.8 (−4.2)	27.9 (+0.6)	38.6 (−1.1)	15.2 (−2.5)	22.5 (+0.6)	16.1 (+0.1)	−1.0
	Canonical	82.1 (+16.6)	60.0 (+5.9)	61.3 (+7.5)	3.3 (0.0)	28.3 (−1.7)	29.7 (+2.4)	48.0 (+8.3)	22.8 (+5.1)	24.2 (+2.3)	17.1 (+1.1)	+4.7
Hel.	Amari	57.4 (−8.1)	52.7 (−1.4)	52.4 (−1.4)	4.4 (+1.1)	22.5 (−7.5)	27.2 (−0.1)	36.6 (−3.1)	16.2 (−1.5)	23.0 (+1.1)	15.6 (−0.4)	−2.1
	Canonical	81.8 (+16.3)	**61.8 (+7.7)**	**63.1 (+9.3)**	**8.9 (+5.6)**	**37.5 (+7.5)**	29.9 (+2.6)	**50.8 (+11.1)**	**25.0 (+7.3)**	26.2 (+4.3)	17.4 (+1.4)	+7.3
χ^2	Amari	72.2 (+6.7)	56.1 (+2.0)	54.8 (+1.0)	3.3 (0.0)	21.7 (−8.3)	27.9 (+0.6)	40.3 (+0.6)	18.4 (+0.7)	23.0 (+1.1)	16.6 (+0.6)	+0.5
	Canonical	82.2 (+16.7)	59.4 (+5.3)	59.9 (+6.1)	5.6 (+2.3)	30.8 (+0.8)	29.8 (+2.5)	45.4 (+5.7)	20.0 (+2.3)	25.3 (+3.4)	17.6 (+1.6)	+4.7

4.2 Large Language Models Benchmarks

Arena-Hard & AlpacaEval. We post-train `Qwen3-1.7B-Base` on 61K UltraFeedback-Binarized preference pairs with six f-divergences in both Amari and canonical form (recipe in Appendix D, larger models and LoRA in Appendix F.1) and evaluate each policy on the 500 prompts of Arena-Hard v0.1 with `gpt-4.1-mini-2025-04-14` as the judge. Against the standard `GPT-4` baseline (Fig. 4, right, top), the canonical form outperforms the Amari form for all six divergences. In head-to-head comparisons between the two forms of the same divergence (Fig. 4, right, bottom), the canonical policy wins in every case. The same holds on AlpacaEval 2.0 (Appendix F.2).

Mathematical reasoning. We repeat the comparison on ten mathematical reasoning benchmarks, where the answer is verifiable, by training the same six divergences on 89K pairs of a correct and an incorrect response and evaluating pass@1 (Table 2). Across the sixty (divergence, benchmark) pairs, the canonical form consistently outperform the Amari form in 57, ties in 2, and underperforms in only 1 case. Notably, five of the six Amari forms underperform the base model on average, whereas every canonical form outperforms it. This suggests that the variance of the Amari estimator can be large enough to undermine off-policy training.

Qualitative behavior. The performance gap is also evident in the actual model outputs. In the Arena-Hard responses of Appendix F.3, an Amari policy often stays close to the answer of the base model, backs away from the question prematurely, or degrades into repeated characters, whereas the canonical policy trained on the same data departs from the base answer and gives a correct one. In Appendix B.7, the training curves show that the gradient norm and the reward margin of the canonical form fluctuate far less than those of the Amari form. Its updates therefore remain more stable, suggesting that the choice of estimator plays a critical role in practice.

5 Conclusion

Beyond the bandit setting, off-policy preference optimization must reconstruct an implicit reward whose inner term takes an expectation over actions that the dataset never records, and the way this term is estimated decides whether training succeeds. We showed that reverse KL written as $u \log u$, the generator DPO uses, is the only separable convex regularizer whose single-sample estimate of this term has zero variance. Every other regularizer entails a variance that accumulates along the trajectory. To overcome this issue, we proposed the *canonical correction*, derived from the condition $f'(1) = f''(1)$ that off-policy update imposes, and we characterized when it exists and what explicit form it takes. From toy MDPs to language-model benchmarks, the correction turns previous unstable regularizers into accurate off-policy ones while preserving the each distinct characteristic. From these results, we suggest two directions for future work. First, separability covers most divergences in practice, but excludes non-separable ones such as the Rényi divergence. Whether any of them admits a tractable and unbiased estimate of the inner term once Axiom (A5) of Theorem 1 is relaxed remains an open question. Second, the inner term of a convex regularizer is itself a Bregman divergence, which points to a deeper connection between preference optimization and information geometry, and we leave this for future work.

AI USE STATEMENT

In this work, we used generative AI tools to polish text that the authors had already written and to assist with the experiment code. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including its text, its claims, and the artifacts it produces.

ETHICS STATEMENT

This work studies how an existing preference optimization loss is estimated, and it involves no human subjects and no collection of new data. The toy experiments run on synthetic MDPs drawn by our own code, as specified in Appendix C.1. The language-model we used is publicly released model weights and preference datasets under their licenses.

REPRODUCIBILITY STATEMENT

The assumptions behind every theoretical claim are stated with the claim itself in Sec. 3, and Appendix B gives a complete proof for each one in its own subsection. The toy experiments are specified in Appendix C and the language-model experiments are specified in Appendix D. Source code for all experiments is included as anonymous supplementary material.

REFERENCES

Shun-Ichi Amari. α-Divergence is unique, belonging to both f-Divergence and Bregman divergence classes. *IEEE Transactions on Information Theory*, 55(11):4925–4931, 2009.

Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

Gholamali Aminian, Amir R Asadi, Idan Shenfeld, and Youssef Mroueh. Kl-regularized rlhf with multiple reference models: Exact solutions and sample complexity. *Advances in Neural Information Processing Systems*, 38:101117–101151, 2026.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005.

Boris Belousov and Jan Peters. Entropic regularization of markov decision processes. *Entropy*, 21(7): 674, 2019.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. ISBN 978-0521833783.

Taehyun Cho, Seokhun Ju, Seungyub Han, Dohyeong Kim, Kyungjae Lee, and Jungwoo Lee. Policy-labeled preference learning: Is preference enough for RLHF? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10524–10553. PMLR, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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 (TMLR)*, 2023. URL https://arxiv.org/abs/2304.06767.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pp. 2160–2169. PMLR, 2019.

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 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

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

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, 2016. URL https://arxiv.org/abs/1606.03476.

Suhwan Kim, Taehyun Cho, Youngsoo Jang, Geonhyeong Kim, Yujin Kim, Moontae Lee, and Jungwoo Lee. A regret minimization framework on preference learning in large language models. *arXiv preprint arXiv:2606.09124*, 2026. URL https://arxiv.org/abs/2606.09124.

Aobo Kong, Wentao Ma, Shiwan Zhao, Yongbin Li, Yuchuan Wu, Ke Wang, Xiaoqian Liu, Qicheng Li, Yong Qin, and Fei Huang. SDPO: Segment-level direct preference optimization for social agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12409–12423, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.607.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025.

Kezhao Liu, Jason Klein Liu, Mingtao Chen, and Yiming Liu. Rethinking KL regularization in RLHF: From value estimation to gradient optimization. In *International Conference on Learning Representations (ICLR)*, 2026a.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Jianbin Jiao, and Junge Zhang. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations (ICLR)*, 2026b.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems 38*, 2024. URL https://arxiv.org/abs/2405.14734.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and more efficient off-policy RL for language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.18252.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, 2022. URL https://arxiv.org/abs/2203.02155.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://arxiv.org/abs/2403.19159.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://arxiv.org/abs/2305.18290.

Gavin A. Rummery and Mahesan Niranjan. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, University of Cambridge, Department of Engineering, 1994.

John Schulman. Approximating KL divergence. http://joschu.net/blog/kl-approx.html, 2020. Blog post.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft Q-learning. *arXiv preprint arXiv:1704.06440*, 2017. URL https://arxiv.org/abs/1704.06440.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://arxiv.org/abs/2009.01325.

Zetian Sun, Dongfang Li, Xuhui Chen, Baotian Hu, and Min Zhang. Is on-policy data always the best choice for direct preference optimization-based LM alignment? In *International Conference on Learning Representations (ICLR)*, 2026.

Yunhao Tang and Rémi Munos. On a few pitfalls in kl divergence gradient estimation for rl. *arXiv preprint arXiv:2506.09477*, 2025.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *The Tenth International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2005.09814.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of KL regularization in reinforcement learning. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://arxiv.org/abs/2003.14089.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16240.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3–4):279–292, 1992. URL https://doi.org/10.1007/BF00992698.

Di Wu, Chengshuai Shi, Jing Yang, and Cong Shen. f-divergence regularized RLHF: Two tales of sampling and unified analyses. In *International Conference on Machine Learning (ICML)*, 2026.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct preference optimization with dynamic β. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.08639.

Jiarui Yao, Ruida Wang, Hao Bai, and Tong Zhang. Future-kl regularized grpo: Process-level credit assignment from f-divergence regularization. *arXiv preprint arXiv:2601.10201*, 2026.

Hee Suk Yoon, Eunseop Yoon, Mark Hasegawa-Johnson, Sungwoong Kim, and Chang D. Yoo. ConfPO: Exploiting policy model confidence for critical token selection in preference optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025. URL https://arxiv.org/abs/2506.08712.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. On the design of KL-regularized policy gradient algorithms for LLM reasoning. In *International Conference on Learning Representations (ICLR)*, 2026.

A Related Work

KL-regularized policy optimization. KL has long served as the regularizer of choice across many fields, including trust-region policy optimization (Schulman et al., 2015), imitation learning (Ho & Ermon, 2016), and post-training of language models (Stiennon et al., 2020). In RL, it has been the standard regularizer since the maximum-entropy framework was unified with policy gradients (Schulman et al., 2017; Haarnoja et al., 2018). The regularizer enters the soft Bellman operator and yields a closed-form softmax optimal policy (Vieillard et al., 2020). Aminian et al. (2026) give an exact solution and a sample-complexity analysis for reverse-KL RLHF with several reference models, and extend both to forward KL.

General convex regularizers in RL. To extend KL-regularized RL with a closed-form policy structure, several prior works introduced broader convex penalties whose Legendre–Fenchel dual still has an explicit soft-policy form. Geist et al. (2019) develops a general theory of regularized Markov decision processes by generalizing entropy and KL penalties to arbitrary strictly convex regularizers via the Legendre–Fenchel transform. Belousov & Peters (2019) extends the analysis to α-divergences and identifies the actor-critic arising from each f-divergence constraint. Closer to our setting, Wu et al. (2026) analyze online RLHF under a general f-divergence regularizer and obtain their guarantees by sampling from the current policy, and Yao et al. (2026) show that the on-policy gradient of a token-wise f-divergence-regularized objective carries a future-regularization term that per-token losses omit.

Direct preference optimization and its variants. Rafailov et al. (2023) introduce DPO in the bandit setting, observing that the KL-regularized RLHF objective has a closed-form implicit reward $\beta \log(\pi_\theta/\pi_{\mathrm{ref}})$ that can be matched directly to preferences. Wang et al. (2024) generalize this construction to arbitrary f-divergence regularizers, still in the bandit setting. Cho et al. (2025) first establish necessary and sufficient conditions for the existence of a closed-form implicit reward in the MDP setting under the maximum-entropy framework, and Kim et al. (2026) extend DPO to LLM reasoning tasks by presenting analogous conditions for the MDP setting under the KL-regularized framework. Liu et al. (2026b) derive a multi-turn DPO objective from a multi-step MDP whose step-wise implicit rewards serve as an implicit process reward, and Kong et al. (2025) apply DPO to selected segments of a multi-turn interaction to reduce training noise. On the data side, Sun et al. (2026) find that on-policy preference data is not uniformly better and that diverse off-policy data helps early in alignment, which supports treating off-policy data as a reasonable choice rather than a compromise.

Generator representatives and variance-reduced estimators. In RLHF practice, the single-sample estimators of a KL divergence built from a sampled log-ratio are named after Schulman (2020). With u the ratio of the reference density to the sampled density, they are $k_1 = -\log u$, $k_2 = \frac{1}{2}(\log u)^2$, and $k_3 = (u-1) - \log u$, and k_3 is the one in common use. The affine freedom $f \mapsto f + c\,(u-1)$ in an f-divergence generator is standard, and the Amari representative $f'(1) = 0$ (Amari, 2009; 2016) is the conventional way of fixing it in information geometry. To our knowledge, it has not previously been treated as a design variable of an off-policy estimator.

On the estimator side, k_3 adds the zero-mean term $u - 1$ to the sampled log-ratio to reduce variance without introducing bias, and is used throughout RLHF implementations. Zhang et al. (2026) derive the correct gradient estimators for forward and reverse, normalized and unnormalized KL regularizers together with clipped importance-sampling variants for off-policy use, Tang & Munos (2025) show that differentiating through a KL estimate used as a loss gives the wrong gradient, and Liu et al. (2026a) finds that biased configurations destabilize training while unbiased ones generalize better.

B PROOFS FOR SEC. 3

B.1 PROOF OF LEMMA 1 (STRUCTURAL FORM)

Lemma 1 (Structural form of the β-optimal reward)**.** *Assume $\Omega(\cdot;\pi_{\mathrm{ref}})$ is essentially smooth in its first argument, so that the β-optimal policy π^* lies in the interior of $\Delta(\mathcal{A})$ at every state. Then π^* is β-optimal under the reward r if and only if there exists a state-dependent function $\lambda : \mathcal{S} \to \mathbb{R}$ such that, for every (s,a),*

$$Q^*(s,a) = \beta[\nabla_\pi \Omega(\pi^*(\cdot|s);\pi_{\mathrm{ref}}(\cdot|s))]_a + \lambda(s), \tag{5}$$

$$r(s,a) = \beta[\nabla_\pi \Omega(\pi^*(\cdot|s);\pi_{\mathrm{ref}}(\cdot|s))]_a + \lambda(s) - \gamma \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V^*(s')]. \tag{6}$$

Proof. **Step 1: Reduction to a concave program over occupancies.** For a policy π and initial state distribution ρ_0, the (normalized) discounted state–action occupancy is

$$d(s,a) := (1-\gamma)\sum_{t\geq 0}\gamma^t\,\mathbb{P}_\pi(s_t = s,\, a_t = a), \qquad d(s) := \sum_a d(s,a).$$

The map sending π to its occupancy d is a bijection between policies and nonnegative measures on $\mathcal{S}\times\mathcal{A}$ satisfying the Bellman flow constraint

$$d(s') = (1-\gamma)\,\rho_0(s') + \gamma\sum_{s,a} d(s,a)\,\mathbb{P}(s' \mid s,a) \qquad \forall\, s' \in \mathcal{S},$$

where on the right-hand side (s,a) ranges over predecessor state–action pairs, and the policy is recovered as $\pi(a|s) = \frac{d(s,a)}{d(s)}$ on reachable s. Under this change of variable, the Ω-regularized RL objective is, up to multiplication by $(1-\gamma)^{-1}$,

$$\max_{d\geq 0}\sum_{s,a} d(s,a)\,r(s,a) - \beta\sum_s d(s)\,\Omega\!\left(\frac{d(s,\cdot)}{d(s)};\,\pi_{\mathrm{ref}}(\cdot|s)\right),$$

subject to the Bellman flow constraint and $d \geq 0$. Each statewise summand $d(s)\,\Omega\!\left(\frac{d(s,\cdot)}{d(s)};\,\pi_{\mathrm{ref}}(\cdot|s)\right)$ is the perspective of the convex map $\Omega(\cdot;\,\pi_{\mathrm{ref}}(\cdot|s))$ on $\Delta(\mathcal{A})$ and is therefore convex in $d(s,\cdot)$. The full objective is hence concave in d, the equality constraints are affine, and the nonnegativity constraints are convex, so the problem is a concave program. Slater's condition holds at any strictly positive feasible d, and the KKT conditions are necessary and sufficient at the optimum.

Step 2: Lagrangian. With the program in place, we introduce a multiplier $\alpha(s) \in \mathbb{R}$ for the Bellman flow constraint at each s, and a multiplier $\mu(s,a) \geq 0$ for the nonnegativity constraint $d(s,a) \geq 0$. The Lagrangian of the concave program is

$$\begin{aligned}
&\mathcal{L}(d,\alpha,\mu)\\
&= \sum_{s,a} d(s,a)\,r(s,a) - \beta\sum_s d(s)\,\Omega\!\left(\frac{d(s,\cdot)}{d(s)};\,\pi_{\mathrm{ref}}(\cdot|s)\right)\\
&\quad - \sum_{s'}\alpha(s')\left[\,d(s') - (1-\gamma)\,\rho_0(s') - \gamma\sum_{s,a} d(s,a)\,\mathbb{P}(s' \mid s,a)\,\right] + \sum_{s,a}\mu(s,a)\,d(s,a).
\end{aligned}$$

At the optimum d^*, we write $\pi^*(a|s) := d^*(s,a)/d^*(s)$ on reachable s, and the stationarity condition reads $\partial\mathcal{L}/\partial d(s,a) = 0$. The next step computes this derivative term by term.

Step 3: KKT stationarity. The contribution of $d(s,a)$ to each term of $\mathcal{L}$ is direct except for the regularizer term, which requires the chain rule with

$$\frac{\partial d(s)}{\partial d(s,a)} = 1 \qquad \text{and} \qquad \frac{\partial}{\partial d(s,a)}\left[\frac{d(s,b)}{d(s)}\right] = \frac{\mathbf{1}\{a=b\} - \pi^*(b|s)}{d^*(s)}.$$

Expanding the product rule and chain rule term by term gives the statewise perspective derivative,

$$\begin{aligned}
&\frac{\partial}{\partial d(s,a)}\left[d(s)\,\Omega\!\left(\frac{d(s,\cdot)}{d(s)};\,\pi_{\mathrm{ref}}(\cdot|s)\right)\right]_{d=d^*}\\
&=\underbrace{\frac{\partial d(s)}{\partial d(s,a)}}_{=1}\,\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))+d^*(s)\sum_b[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_b\cdot\frac{\partial}{\partial d(s,a)}\left[\frac{d(s,b)}{d(s)}\right]_{d=d^*}\\
&=\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))+\sum_b[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_b\left(\mathbf{1}\{a=b\}-\pi^*(b|s)\right)\\
&=\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))+[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a-\mathbb{E}_{b\sim\pi^*(\cdot|s)}\left[[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_b\right]\\
&=[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a-\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)],
\end{aligned}$$

where the last line uses the identity $\mathbb{E}_{a\sim\pi}[\Psi_a(u_a)] = \mathbb{E}_{b\sim\pi}\left[[\nabla_\pi\Omega(\pi;\,\pi_{\mathrm{ref}})]_b\right]-\Omega(\pi;\,\pi_{\mathrm{ref}})$ that defines the inner term in Eq. 7, and the $d^*(s)$ factor in the second line cancels against the $1/d^*(s)$ of the chain-rule expression. Substituting this derivative into the stationarity condition $\partial\mathcal{L}/\partial d(s,a)=0$ at $d=d^*$ then yields

$$\begin{aligned}
0 = r(s,a)-\beta[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a+\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)]\\
-\alpha(s)+\gamma\sum_{s'}\mathbb{P}(s'\mid s,a)\,\alpha(s')+\mu(s,a).
\end{aligned}$$

Step 4: Complementary slackness eliminates $\mu(s,a)$. The stationarity condition still involves the multiplier $\mu(s,a)$, which we now remove. The KKT complementary slackness condition requires $\mu(s,a)\,d^*(s,a)=0$ for every (s,a), so $\mu(s,a)$ can be nonzero only when the constraint $d(s,a)\geq 0$ is active, *i.e,* $d^*(s,a)=0$.

We show that this cannot occur at any reachable state-action pair. Since $\Omega(\cdot;\,\pi_{\mathrm{ref}})$ is essential smooth in its first argument, $\|\nabla_\pi\Omega(\pi;\,\pi_{\mathrm{ref}})\|\to\infty$ as π approaches the relative boundary of $\Delta(\mathcal{A})$. Hence, the per-state argmax of the regularized soft Bellman operator lies in the relative interior of $\Delta(\mathcal{A})$ at every state. Therefore, $\pi^*(a|s)>0$ for every a, and $d^*(s,a)=d^*(s)\,\pi^*(a|s)>0$ on every reachable (s,a), so complementary slackness gives $\mu(s,a)=0$ on every reachable (s,a). With $\mu\equiv 0$, the stationarity condition rearranges to

$$\alpha(s)=r(s,a)-\beta[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a+\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)]+\gamma\sum_{s'}\mathbb{P}(s'\mid s,a)\,\alpha(s'),\tag{15}$$

holding for every reachable (s,a), with the multiplier α not yet identified.

Step 5: Identification of $\alpha\equiv V^*$. Taking the expectation of Eq. 15 under $a\sim\pi^*(\cdot|s)$, the gradient term becomes

$$\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\,[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a\,]=\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)]+\beta\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))$$

by the same identity, and the two inner-term contributions cancel. The recursion that remains is

$$\alpha(s)=\mathbb{E}_{a\sim\pi^*(\cdot|s)}[r(s,a)]-\beta\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))+\gamma\mathbb{E}_{a\sim\pi^*(\cdot|s),\,s'\sim\mathbb{P}(\cdot|s,a)}[\alpha(s')],$$

which is exactly the soft Bellman equation for the regularized value V^* evaluated under π^*. The corresponding operator is a γ-contraction on bounded functions, so its unique bounded fixed point is V^{π^*}, which coincides with V^* because π^* is optimal, and therefore $\alpha\equiv V^*$. Substituting $\alpha(s)=V^*(s)$ and $\alpha(s')=V^*(s')$ back into Eq. 15 and defining $Q^*(s,a):=r(s,a)+\gamma\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V^*(s')]$ gives

$$Q^*(s,a)=\beta[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a+\left(V^*(s)-\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)]\right).$$

Defining $\lambda(s):=V^*(s)-\beta\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\Psi_a(u_a)]$, the stationarity condition reduces to $Q^*(s,a)=\beta[\nabla_\pi\Omega(\pi^*(\cdot|s);\,\pi_{\mathrm{ref}}(\cdot|s))]_a+\lambda(s)$, which is Eq. 5. Substituting into the Bellman identity $r(s,a)=Q^*(s,a)-\gamma\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V^*(s')]$ yields Eq. 6. □

B.2 Proof of Theorem 1 (KL uniqueness)

Theorem 1. *Suppose the per-state regularizer Ω satisfies the following for every $\pi_{\mathrm{ref}} \in \Delta(\mathcal{A})$.*

(A1) Coincidence. $\Omega(\pi_{\mathrm{ref}};\, \pi_{\mathrm{ref}}) = 0$.

(A2) Strict positivity. $\Omega(\pi;\, \pi_{\mathrm{ref}}) > 0$ *for every* $\pi \neq \pi_{\mathrm{ref}}$.

(A3) Strict convexity. $\Omega(\cdot;\, \pi_{\mathrm{ref}})$ *is strictly convex on* $\Delta(\mathcal{A})$.

(A4) Off-policy permissibility. Ω *satisfies Definition 1.*

(A5) Separability. Ω *has the form of Eq. 1 for per-action convex functions* $\Omega_{a'}$.

Then at the fixed s',

$$\Omega(\pi(\cdot|s');\, \pi_{\mathrm{ref}}(\cdot|s')) = c(s') \cdot D_{\mathrm{KL}}\big(\pi(\cdot|s') \big\| \pi_{\mathrm{ref}}(\cdot|s')\big), \tag{10}$$

where $c(s') > 0$ depends on s' only through $\pi_{\mathrm{ref}}(\cdot|s')$.

Proof. For notational simplicity we fix a next-state s' and suppress it, writing $\pi := \pi(\cdot|s')$ and $\pi_{\mathrm{ref}} := \pi_{\mathrm{ref}}(\cdot|s')$, and we write a for the next-action a', so that the action index ranges over $\mathcal{A}$. Wherever the suppressed dependence could cause confusion, we make it explicit. Axiom (A4) is read in this pointwise form, in that the inner term $\mathbb{E}_{a\sim\pi}[\Psi_a(u_a)]$ does not depend on π. The per-action functions Ω_a and the constants c, d_a, e_a, l_a, and α introduced below may depend on π_{ref}, and hence on s', and we reinstate this dependence at the end of the proof.

The argument is a short derivation through an ordinary differential equation. Axioms (A4) and (A5) together force such an equation on each component Ω_a. Integrating it twice produces the reverse KL generator $u \log u$ up to two constants of integration, and Axioms (A1) to (A3) pin those constants and fix the sign.

Step 1: ODE from axioms (A4) and (A5). By Axiom (A5), $\Omega(\pi;\, \pi_{\mathrm{ref}}) = \sum_a \pi_{\mathrm{ref}}(a)\, \Omega_a(u_a)$ with $u_a := \pi(a)/\pi_{\mathrm{ref}}(a)$, so $[\nabla_\pi \Omega(\pi;\, \pi_{\mathrm{ref}})]_a = \Omega_a'(u_a)$ and the inner term of Eq. 7 becomes

$$\mathbb{E}_{a\sim\pi}[\Psi_a(u_a)] = \sum_a \pi(a)\, \Psi_a(u_a) = \sum_a \pi_{\mathrm{ref}}(a)\, h_a(u_a), \qquad h_a(u) := u\, \Omega_a'(u) - \Omega_a(u).$$

Consider a simplex-tangent perturbation $\pi \mapsto \pi + \epsilon(e_i - e_j)$, which preserves $\sum_a \pi(a) = 1$ and moves u_i by $\epsilon/\pi_{\mathrm{ref}}(i)$ and u_j by $-\epsilon/\pi_{\mathrm{ref}}(j)$. The two reference weights cancel, and Axiom (A4) requires the derivative of the inner term at $\epsilon = 0$ to vanish, that is,

$$h_i'(u_i) - h_j'(u_j) = 0 \qquad \forall\, i \neq j, \forall\, \pi \in \Delta(\mathcal{A}).$$

Since u_i and u_j vary independently, this forces $h_i'(u) = c$ for a single constant c shared across all i, that is, $h_i(u) = c\, u + l_i$ for action-dependent constants l_i.

Step 2: Two integrations. Differentiating $u\, \Omega_i'(u) - \Omega_i(u) = c\, u + l_i$ in u gives $u\, \Omega_i''(u) = c$, so $\Omega_i''(u) = c/u$. Integrating twice,

$$\Omega_i'(u) = c \log u + d_i, \qquad \Omega_i(u) = c\,(u \log u - u) + d_i\, u + e_i,$$

for constants d_i and e_i. Separability gives the diagonal Hessian $[\nabla^2 \Omega]_{aa} = \Omega_a''(u_a)/\pi_{\mathrm{ref}}(a) = c/(\pi_{\mathrm{ref}}(a)\, u_a)$, so strict convexity from Axiom (A3) holds exactly when $c > 0$.

Step 3: Pin d_a at the reference. Axiom (A2) makes $\pi = \pi_{\mathrm{ref}}$ the unique global minimum of $\Omega(\cdot;\, \pi_{\mathrm{ref}})$ on $\Delta(\mathcal{A})$, and simplex stationarity there reads $\nabla_\pi \Omega(\pi_{\mathrm{ref}};\, \pi_{\mathrm{ref}}) = \alpha \mathbf{1}$ for some scalar α. The reference is the point $u_a = 1$ for every a, so Step 2 gives $\Omega_a'(1) = d_a$, and stationarity is simply $d_a = \alpha$ for every a.

Step 4: Assemble and apply Axiom (A1). Substituting $d_a = \alpha$ and summing over actions, with $\sum_a \pi_{\mathrm{ref}}(a) u_a = \sum_a \pi(a) = 1$,

$$\begin{aligned}
\Omega(\pi;\, \pi_{\mathrm{ref}}) &= \sum_a \pi_{\mathrm{ref}}(a) \Big[c\,(u_a \log u_a - u_a) + \alpha\, u_a + e_a \Big] \\
&= c \sum_a \pi(a) \log u_a + \big[\alpha - c + \textstyle\sum_a \pi_{\mathrm{ref}}(a) e_a\big] \\
&= c\, D_{\mathrm{KL}}(\pi \| \pi_{\mathrm{ref}}) + \big[\alpha - c + \textstyle\sum_a \pi_{\mathrm{ref}}(a) e_a\big].
\end{aligned}$$

Axiom (A1) at $\pi = \pi_{\mathrm{ref}}$ makes the Kullback–Leibler term vanish, so the bracketed constant is zero and

$$\Omega(\pi;\, \pi_{\mathrm{ref}}) = c\, D_{\mathrm{KL}}(\pi \| \pi_{\mathrm{ref}}), \qquad c = c(\pi_{\mathrm{ref}}) > 0,$$

which is Eq. 10. The converse is straightforward that $c\, D_{\mathrm{KL}}$ satisfies axioms (A1) through (A5) with a constant inner term.

Reinstating the suppressed s'-dependence, the conclusion reads

$$\Omega(\pi(\cdot|s');\, \pi_{\mathrm{ref}}(\cdot|s')) = c(s') \cdot D_{\mathrm{KL}}\big(\pi(\cdot|s') \big\| \pi_{\mathrm{ref}}(\cdot|s')\big),$$

where $c(s') := c(\pi_{\mathrm{ref}}(\cdot|s')) > 0$. The coefficient varies across next-states s' only through the reference policy. Whenever π_{ref} is shared across states (e.g. uniform), c reduces to a single constant. □

B.3 f-DIVERGENCE REDUCTION OF THEOREM 1

For the f-divergence parameterization $\Omega = \sum_a \pi_{\mathrm{ref}}(a)\, f(u_a)$ of Eq. 2, axioms (A1)–(A3) follow from the standard properties of f-divergences (convex f with $f(1) = 0$), and (A5) holds with $\Omega_a = f$ for every a. The inner term is then $\mathbb{E}_{a\sim\pi}[\Psi(u_a)]$ with $\Psi(u) = f'(u) - f(u)/u$, so Axiom (A4) requires Ψ to be constant in u. Setting $\Psi'(u) = f''(u) - f'(u)/u + f(u)/u^2 = 0$ gives the Euler equation

$$u^2\, f''(u) - u\, f'(u) + f(u) = 0,$$

whose solutions are $f(u) = A\, u + B\, u \log u$. Imposing $f(1) = 0$ leaves $f(u) = B\, u \log u$ with $B > 0$, the canonical reverse KL generator, so the only f-divergence satisfying Definition 1 is $D_f = B\, D_{\mathrm{KL}}$, recovering Theorem 1. Off-policy permissibility is therefore a property of the generator and not of the divergence alone. Adding an affine term $c\,(u - 1)$ leaves $D_f = B\, D_{\mathrm{KL}}$ unchanged but moves Ψ off the constant, as Theorem 2 makes precise, so among the generators of reverse KL only $B\, u \log u$ satisfies Definition 1.

B.4 PROOF OF THEOREM 2 (EXISTENCE, UNIQUENESS, AND VARIANCE OF THE CANONICAL CORRECTION)

Sec. 3.3 states several facts without proof, and we record them here before proving the theorem. We first note what the Amari form costs reverse KL. We then show that the family of Eq. 12 contains every representation of a given Ω, state the assumptions of the theorem and the scope of the correction, and finally read the existence condition as a curvature condition. Squared Euclidean is the regularizer that fails it. The proof itself follows in six steps, numbered from zero.

Reverse KL in the Amari form. By Theorem 1, Ω admits a representation satisfying Definition 1 only when Ω is a positive multiple of D_{KL}, and that representation is then unique. It is the generator $u \log u$, whose integrand is $\Psi \equiv 1$. The Amari representative $u \log u - (u - 1)$ induces the same divergence and the same optimal policy, but its integrand is $\Psi(u) = 1 - 1/u$, which diverges as $u \to 0$. For every other Ω, the reparametrization of Eq. 12 can bring the integrand close to a constant near the reference but cannot make it constant.

The family of Eq. 12 contains every representation. Sec. 3.3 states that two families $\{\Omega_a\}$ and $\{\widetilde{\Omega}_a\}$ represent the same Ω if and only if they are related by Eq. 12. The forward direction is the computation in Eq. 13. For the converse, write $p_a := \pi_{\mathrm{ref}}(a)$ and set $\delta_a := \widetilde{\Omega}_a - \Omega_a$. The two families represent the same function exactly when

$$\sum_a p_a\, \delta_a(u_a) = 0 \qquad \text{for every } u \text{ with } u_a \geq 0 \text{ and } \textstyle\sum_a p_a u_a = 1. \tag{16}$$

Perturbing along a simplex-tangent direction, $u_i \mapsto u_i + \epsilon/p_i$ and $u_j \mapsto u_j - \epsilon/p_j$, preserves the constraint, and differentiating Eq. 16 at $\epsilon = 0$ gives $\delta_i'(u_i) = \delta_j'(u_j)$ for every $i \neq j$. Hence $\delta_a(u) = c\, u + \mu_a$ for constants μ_a. Evaluating Eq. 16 at $\pi = \pi_{\mathrm{ref}}$, where every $u_a = 1$, gives $c + \sum_a p_a \mu_a = 0$, so $c = -\sum_b p_b \mu_b$ and δ_a is exactly the shift of Eq. 12. The family therefore contains every map that preserves Ω, and this is what makes the correction of Theorem 2(i) unique among all representations rather than only within a subfamily.

Scope of the correction. Note that the correction preserves the objective. It changes one constant in the generator and leaves the divergence, its minimizer, the β-optimal policy, and the objective identical, so an Amari arm and a canonical arm of the same divergence differ only in the estimator. In an implementation, it adds the single term $c^\star(1 - 1/u)$ to the per-token score $g(u) = f(u)/u$ of Appendix C.3. Since $g'(1) = f'(1)$, the Amari form makes the per-token gradient vanish at a policy initialized at the reference, for every divergence at once, whereas the canonical condition gives $g'(1) = f''(1) > 0$. The correction is local, however. Conditions (C1) and (C2) suppress the inner-term error only to second order in $\|\pi - \pi_{\mathrm{ref}}\|$.

Theorem 2. *Let $\psi_a := \Psi_a(1)$ and $\kappa_a := \Omega_a''(1)$. Then the following hold.*

(i) (Existence and uniqueness.) *A canonical form Ω^{can} exists if and only if κ_a is independent of a, i.e., $\kappa_a \equiv \kappa$. If it exists, it is unique and is given by $\mu_a = \psi_a - \kappa$ for every $a \in \mathcal{A}$.*

(ii) (Variance.) *Suppose that $\kappa_a \equiv \kappa$, and let $\delta := \|\pi - \pi_{\mathrm{ref}}\|_1$. Then, as $\delta \to 0$,*

$$\mathrm{Var}_{a\sim\pi}\big[\Psi_a^\mu(u_a)\big] = \begin{cases} O(\delta^4) & \text{if } \mu \text{ is the canonical correction}, \\ \Theta(\delta^2) & \text{if } \mu \text{ satisfies (C1) and violates (C2).} \end{cases}$$

Proof. Throughout, write $p_a := \pi_{\mathrm{ref}}(a)$ and $u_a := \pi(a)/p_a$. The two defining conditions of a canonical correction are (C1), that $\Psi_a^\mu(1)$ be independent of a, written $\Psi_a^\mu(1) = \nu$ for a common value ν, and (C2), that $(\Psi_a^\mu)'(1) = 0$. We write (E1) for $\psi_a \equiv \psi$ and (E2) for $\kappa_a \equiv \kappa$. Only (E2) is needed for existence, as Step 3 shows.

Definition 1 asks $a \mapsto \Psi_a^\mu(u_a)$ to be constant at every π. Expanding around $u_a = 1$,

$$\Psi_a^\mu(u_a) = \Psi_a^\mu(1) + (\Psi_a^\mu)'(1)\,(u_a - 1) + O\big(\|\pi - \pi_{\mathrm{ref}}\|^2\big).$$

Constancy of the zeroth-order term is (C1). For the first-order term, take $\pi - \pi_{\mathrm{ref}} = \epsilon(e_i - e_j)$ with $i \neq j$, which perturbs only the coordinates i and j. Since $|\mathcal{A}| \geq 3$, some coordinate is left fixed by the perturbation. Comparing i and j with that coordinate gives $(\Psi_i^\mu)'(1) = 0$ and $(\Psi_j^\mu)'(1) = 0$ for every $i \neq j$, which is (C2).

Step 1: Each condition pins μ_a. From $\Psi_a^\mu(u) = \Psi_a(u) - \mu_a/u$ and $(\Psi_a^\mu)'(u) = \Psi_a'(u) + \mu_a/u^2$, evaluated at $u = 1$, the two conditions read

$$\begin{aligned} (C1): \quad & \psi_a - \mu_a = \nu \quad \Longrightarrow \quad \mu_a = \psi_a - \nu, \\ (C2): \quad & \Psi_a'(1) + \mu_a = 0 \quad \Longrightarrow \quad \mu_a = -\Psi_a'(1), \end{aligned}$$

where ν is the common value required by (C1).

Step 2: A key identity. Since $\Psi_a'(u) = \Omega_a''(u) - \Omega_a'(u)/u + \Omega_a(u)/u^2$, evaluating at $u = 1$ gives

$$\Psi_a'(1) = \Omega_a''(1) - \big[\Omega_a'(1) - \Omega_a(1)\big] = \kappa_a - \psi_a.$$

Step 3: Existence forces (E2), and only (E2). Equating the two expressions for μ_a from Step 1 and using Step 2, $\psi_a - \nu = -(\kappa_a - \psi_a)$, hence $\nu = \kappa_a$. Since ν is a single number, this forces $\kappa_a \equiv \kappa$ (E2), and $\mu_a = \psi_a - \kappa$ is then the only candidate. Conversely, under (E2) this μ satisfies both conditions, since

$$\Psi_a^\mu(1) = \psi_a - \mu_a = \kappa, \qquad (\Psi_a^\mu)'(1) = \Psi_a'(1) + \mu_a = (\kappa_a - \psi_a) + (\psi_a - \kappa) = 0$$

for every a. A canonical correction therefore exists exactly when (E2) holds, and it is unique. We note that (E1) is not needed for existence. Taking $\Omega_a = g + d_a$ for a fixed g and distinct constants d_a gives $\kappa_a \equiv \kappa$ with ψ_a non-constant, and $\mu_a = \psi_a - \kappa$ still satisfies both conditions. What (E1) adds is the one-constant form. If $\psi_a \equiv \psi$ (E1) holds as well, then

$$\mu_a = \psi - \kappa = -c^\star, \qquad c^\star := \kappa - \psi,$$

so the correction is the same constant at every action, and this is derived rather than assumed. This proves (i).

Step 4: Fisher–Rao form of the existence condition. Separability gives the diagonal Hessian $[\nabla^2\Omega(\pi)]_{ab} = \delta_{ab}\,\Omega_a''(u_a)/p_a$. For the categorical model $p(a;\pi) = \pi(a)$ the Fisher information along the simplex tangent directions is $I(\pi) = \mathrm{diag}(\pi(a)^{-1})$, so $I(\pi_{\mathrm{ref}}) = \mathrm{diag}(p_a^{-1})$. Hence $\nabla^2\Omega|_{\pi_{\mathrm{ref}}} = \kappa\, I(\pi_{\mathrm{ref}})$ if and only if $\Omega_a''(1) = \kappa$ for every a, which is (E2).

Step 5: Variance order. Let μ satisfy (C1) with common value ν, so that $\mu_a = \psi_a - \nu$ by Step 1, and write $\epsilon_a := u_a - 1$. Expanding Ψ_a at $u_a = 1$ with Step 2, and using $\mu_a/u_a = (\psi_a - \nu)(1 - \epsilon_a + O(\epsilon_a^2))$, the two ψ_a terms cancel and

$$\Psi_a^\mu(u_a) = \nu + (\kappa_a - \nu)\,\epsilon_a + O(\epsilon_a^2).$$

The constant term is a-independent and drops from the variance, and under (E2) the linear coefficient $\kappa - \nu$ is a-independent as well, so

$$\mathrm{Var}_{a\sim\pi}\big[\Psi_a^\mu(u_a)\big] = (\kappa - \nu)^2\,\mathrm{Var}_{a\sim\pi}[\epsilon_a] + O\big(\|\epsilon\|^3\big),$$

which is $\Theta(\|\epsilon\|^2)$ whenever $\nu \neq \kappa$. At $\nu = \kappa$, which is the canonical correction, the linear term vanishes, the leading term is quadratic in ϵ_a, and the variance is $O(\|\epsilon\|^4)$. Here $\|\epsilon\|$ may be any norm on $\mathbb{R}^{\mathcal{A}}$. Since $\sum_a \pi_{\mathrm{ref}}(a)\,\epsilon_a = \sum_a \pi(a) - 1 = 0$, one has $\mathrm{Var}_{a\sim\pi}[\epsilon_a] = \sum_a \pi_{\mathrm{ref}}(a)\,\epsilon_a^2 + O(\|\epsilon\|^3)$, which is bounded above and below by constant multiples of $\|\epsilon\|^2$ because $\mathcal{A}$ is finite and π_{ref} has full support, and $\|\epsilon\|$ is in turn bounded above and below by constant multiples of $\|\pi - \pi_{\mathrm{ref}}\|$ for the same reason. The constants depend on $\min_a \pi_{\mathrm{ref}}(a)$ and on the norm, so the rates in (ii) hold for $\delta = \|\pi - \pi_{\mathrm{ref}}\|_1$ as stated and equally for any other norm, with only the constants changing. Neither branch uses (E1), so the rates hold whether or not ψ_a is constant. This proves (ii). □

Curvature view of the existence condition. The condition $\kappa_a \equiv \kappa$ says that the Hessian of Ω at the reference is proportional to the Fisher–Rao metric of $\Delta(\mathcal{A})$ (Amari, 2016), and Step 4 of the proof states that equivalence. Since the Hessian of a separable Ω at the reference is $\mathrm{diag}\big(\kappa_a/\pi_{\mathrm{ref}}(a|s)\big)$, the condition asks this Hessian to be proportional to $\mathrm{diag}\big(1/\pi_{\mathrm{ref}}(a|s)\big)$. A regularizer therefore admits a canonical correction exactly when its second-order behavior at the reference mathes the Fisher information metric. That is, it penalizes a given absolute change in a rare action more than the same change in a common one.

Corollary 1 (f-divergences and the α-family)**.** *For $\Omega = \sum_a \pi_{\mathrm{ref}}(a)\,f(u_a)$ with f strictly convex and $f(1) = 0$, every Ω_a equals f, so part (i) applies with $\psi = f'(1)$ and $\kappa = f''(1)$. Hence $c^\star = f''(1) - f'(1)$, and the generator of canonical form f^{can} satisfies $(f^{\mathrm{can}})'(1) = (f^{\mathrm{can}})''(1)$.*

Proof of Corollary 1. In Eq. 1, the f-divergence is the case $\Omega_a = f$ for every a, so $\Psi_a = \Psi$ with $\Psi(u) = f'(u) - f(u)/u$, and $\psi_a = \Psi(1) = f'(1)$ and $\kappa_a = f''(1)$ are independent of a without further assumption. Hence (E1) and (E2) hold, and the correction $\mu_a = \psi_a - \kappa$ of Theorem 2(i) reduces to the single constant $-c^\star$ where $c^\star = \kappa - \psi = f''(1) - f'(1)$.

Under $f \mapsto f + c(u-1)$, $f''(1)$ is an invariant of the divergence while $f'(1)$ merely labels the representative. The characterization $f'(1) = f''(1)$ is therefore independent of the starting representative. □

The α-divergence family. We specialize Corollary 1 to the α-divergence family. For the family in the Amari form $f_\alpha(u) = \frac{u^\alpha - \alpha u + (\alpha-1)}{\alpha(\alpha-1)}$, one has $f_\alpha''(u) = u^{\alpha-2}$, hence $f_\alpha''(1) = 1$ for every α. From the Amari representative $f'(1) = 0$, this gives $c^\star = 1$ and

$$f_\alpha^{\mathrm{can}}(u) = f_\alpha(u) + (u-1), \qquad \Psi_\alpha^{\mathrm{can}}(u) = \frac{u^\alpha - 1 + \alpha}{\alpha u} \xrightarrow{\alpha\to 1} 1,$$

so the canonicalized family satisfies Definition 1 exactly at $\alpha = 1$, and the reverse KL is no longer an isolated point of the family. Hence, exact off-policy permissibility at every π, which implies $\Psi \equiv \mathrm{const}$, requires $\Psi'(u) \equiv 0$ and Appendix B.3 shows holds only for $f(u) = B\,u\log u$. The local suppression of Theorem 2(ii) therefore becomes global exactness only for reverse KL.

B.5 DPO VARIANTS INSIDE THE AXIOM CLASS

Sec. 3.2 states that several preference-optimization objectives in use involve in the axiom class of Theorem 1 and differ only in the coefficient $c(s')$ and the shaping term $\lambda(s)$. β-DPO (Wu et al., 2024) leaves the log-ratio reward untouched and calibrates β per batch from the informativeness of the preference pairs, which moves the coefficient c of Theorem 1 and nothing else. R-DPO (Park et al., 2024) adds a length penalty $\alpha|y|$ inside the sigmoid, and since $|y|$ counts tokens, this contributes the same constant at every step, and the shaping function $\lambda(s_t)$ of Lemma 1 absorbs.

SimPO (Meng et al., 2024) replaces the explicit reference policy with a length-normalized log-likelihood reward

$$r_{\mathrm{SimPO}}(x, y) = \frac{\beta}{|y|} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid y_{<t}, x),$$

which is the KL implicit reward (Eq. 11) against a uniform π_{ref} with a per-trajectory rescaling of β. ConfPO (Yoon et al., 2025) extends SimPO by gating the per-token sum with a confidence-based mask $s(y_t) \in \{0, 1\}$ that selects only critical (low-confidence) tokens. This per-token reweighting is also absorbed into the shaping function $\lambda(s_t)$ of Lemma 1. A variant of any of these constructions that swapped reverse KL for another convex regularizer would lose off-policy permissibility and would require an additional Monte-Carlo sample of a' at every reachable next-state during training.

B.6 Dropping or weakening Axiom (A5)

In this section, we ask what survives when Axiom (A5) is dropped, or weakened to allow a non-additive decomposition across actions.

What survives without (A5). Step 1 of the proof of Theorem 1 (Appendix B.2) uses (A5) to write $[\nabla_\pi \Omega(\pi;\, \pi_{\mathrm{ref}})]_a = \Omega'_a(u_a)$, after which Axiom (A4) reduces to the per-action ODE $h'_a(u_a) = c$ via the simplex-tangent perturbation $\pi \mapsto \pi + \epsilon(e_i - e_j)$. Without (A5), $[\nabla_\pi \Omega]_a$ generically depends on the full vector π, and Axiom (A4) becomes a partial differential equation over the simplex on the inner term, written $I(\pi) := \mathbb{E}_{a\sim\pi}[\Psi_a(u_a)]$ in this subsection. The same simplex-tangent argument gives only

$$\frac{\partial I}{\partial \pi(i)} - \frac{\partial I}{\partial \pi(j)} = 0 \qquad \forall\, i \neq j, \forall\, \pi \in \Delta(\mathcal{A}),$$

which does not decouple into a single-coordinate ODE.

Standard non-separable candidates fail (A4). The most natural strictly convex non-separable functionals on the simplex all violate Axiom (A4).

- *Quadratic forms* $\Omega(\pi) = \frac{1}{2}\pi^\top M \pi$ with M positive definite. Then $[\nabla\Omega]_a = (M\pi)_a$ and $I(\pi) = \pi^\top M\pi - \frac{1}{2}\pi^\top M\pi = \frac{1}{2}\pi^\top M\pi$, which is constant in π only when $M = 0$.
- *Power norms* $\Omega(\pi) = \frac{1}{p}\|\pi\|_p^p$ with $p > 1$. Then $[\nabla\Omega]_a = \pi(a)^{p-1}$ and $I(\pi) = (1 - 1/p)\,\|\pi\|_p^p$, again non-constant.
- *Monotone outer transforms* $\Omega(\pi) = h\big(\sum_a g(\pi(a))\big)$ with strictly increasing convex h and strictly convex g. Writing $\Sigma(\pi) := \sum_a g(\pi(a))$, $I(\pi) = h'(\Sigma)\sum_a \pi(a)\, g'(\pi(a)) - h(\Sigma)$. As π varies, Σ varies, so this is constant in π only when h is affine, which collapses the construction back to the separable case $\sum_a g(\pi(a))$ up to constants.

Conjecture. The pattern above, together with the proof of Theorem 1 under (A5), leads us to conjecture that no strictly convex and genuinely non-separable divergence on $\Delta(\mathcal{A})$ satisfies Axioms (A1) to (A4) simultaneously. A proof for the fully non-separable case, in which each $[\nabla_\pi \Omega]_a$ depends on several coordinates $\pi(b)$, or an explicit counterexample would close the one remaining gap in the axiomatic characterization of reverse KL.

B.7 Gradient Stability under Implicit Reward Estimation

Empirically, we find that the canonical form substantially stabilizes the gradient norm during training. In this section, we develop a theoretical analysis of this effect and explain why the Amari form inherently gives rise to fluctuating gradients.

We now analyze how the variance of the implicit reward estimator affects the stability of preference optimization. Consider the DPO loss

$$\mathcal{L}_\theta = -\log \sigma(\beta \Delta_\theta),$$

where Δ_θ denotes the exact trajectory-level reward margin between the preferred and dispreferred trajectories. The corresponding gradient is

$$g_\theta := \nabla_\theta \mathcal{L}_\theta = -\beta\sigma(-\beta\Delta_\theta)\nabla_\theta \Delta_\theta.$$

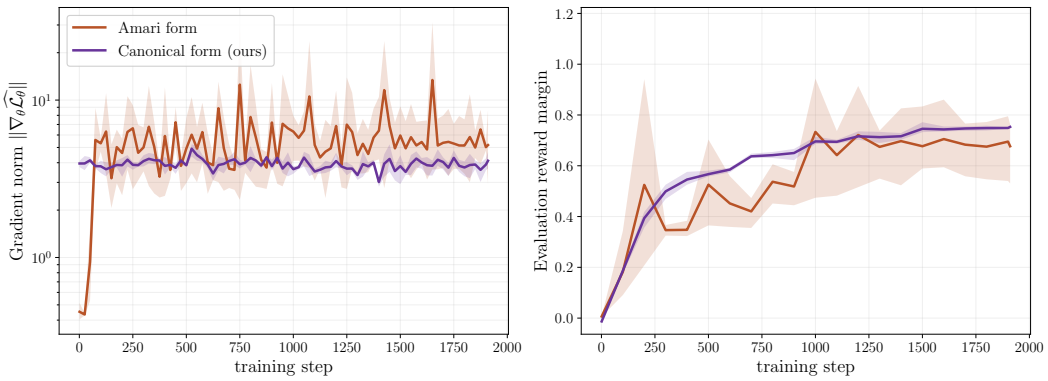


Figure 5: **Empirical evidence of improved optimization stability with canonical estimation.** Experiments are conducted using the `Qwen3-1.7B-Base` model. **(Left)** Evaluation reward margin during training. **(Right)** Gradient norm dynamics during training. The canonical estimator produces smoother reward-margin growth and substantially lower gradient fluctuations than Amari estimator.

Let $\widehat{\Delta}_\theta$ be the implicit reward estimated using sampled subsequent actions

$$\widehat{\Delta}_\theta = \sum_{t=0}^{T_w-1} \widehat{r}_\theta(s_t^w, a_t^w, s_{t+1}^w) - \sum_{t=0}^{T_l-1} \widehat{r}_\theta(s_t^l, a_t^l, s_{t+1}^l)$$

resulting in an estimated margin $\widehat{\Delta}_\theta = \Delta_\theta + \epsilon_\theta$, where ϵ_θ denotes the estimation error. The resulting stochastic gradient is $\widehat{g}_\theta$.

The deviation between the stochastic and exact gradients can then be bounded as

$$\begin{aligned}
\|\widehat{g}_\theta - g_\theta\| &= \left\| -\beta\sigma(-\beta\widehat{\Delta}_\theta)\nabla_\theta\widehat{\Delta}_\theta + \beta\sigma(-\beta\Delta_\theta)\nabla_\theta\Delta_\theta \right\| \\
&\le \left\| -\beta\nabla_\theta\Delta_\theta \left(\sigma(-\beta\widehat{\Delta}_\theta) - \sigma(-\beta\Delta_\theta)\right) \right\| + \left\| -\beta\sigma(-\beta\widehat{\Delta}_\theta)\nabla_\theta\epsilon_\theta \right\| \\
&\le \left\| -\beta\nabla_\theta\Delta_\theta \left(\sigma(-\beta\widehat{\Delta}_\theta) - \sigma(-\beta\Delta_\theta)\right) \right\| + \beta \left\|\nabla_\theta\epsilon_\theta\right\| \\
&\le \beta \left\|\nabla_\theta\Delta_\theta\right\| \cdot \left\| \sigma(-\beta\widehat{\Delta}_\theta) - \sigma(-\beta\Delta_\theta) \right\| + \beta \left\|\nabla_\theta\epsilon_\theta\right\| \\
&\le \frac{\beta^2}{4} \left\|\nabla_\theta\Delta_\theta\right\| \cdot \left\| \widehat{\Delta}_\theta - \Delta_\theta \right\| + \beta \left\|\nabla_\theta\epsilon_\theta\right\| \\
&= \frac{\beta^2}{4} \left\|\nabla_\theta\Delta_\theta\right\| |\epsilon_\theta| + \beta \left\|\nabla_\theta\epsilon_\theta\right\| .
\end{aligned}$$

In the step involving the sigmoid difference, we used the $1/4$-Lipschitz property of the sigmoid function.

Assuming that $\|\nabla_\theta\Delta_\theta\| \le C$, for some finite constant C, which is a standard bounded-gradient assumption in the analysis of stochastic optimization, we obtain

$$\|\widehat{g}_\theta - g_\theta\| \le \frac{\beta^2 C}{4}|\epsilon_\theta| + \beta \left\|\nabla_\theta\epsilon_\theta\right\| .$$

Taking the squared expectation and applying $(a+b)^2 \le 2a^2 + 2b^2$, gives

$$\mathbb{E}\left[\|\widehat{g}_\theta - g_\theta\|^2\right] \le \underbrace{\frac{\beta^4 C^2}{8}\mathbb{E}\left[\epsilon_\theta^2\right]}_{\text{(I) reward-margin estimation error}} + \underbrace{2\beta^2\mathbb{E}\left[\|\nabla_\theta\epsilon_\theta\|^2\right]}_{\text{(II) gradient estimation error}} . \tag{17}$$

This decomposition shows that the gradient instability induced by implicit reward estimation has two sources. In the following, we separately analyze the behavior of these two terms for the Amari estimator and the canonical estimator.

Term (I): Reward-margin estimation error. We first analyze the contribution of the implicit reward variance. Consider a transition triplet (s, a, s'), where the expectation term in the implicit reward is estimated using a single subsequent action $a' \sim \mu(\cdot \mid s')$. For an estimator induced by correction μ, define the single-sample estimation residual

$$\epsilon_{\Psi}^{\mu} = \Psi_{a'}^{\mu}(u_{a'}) - \mathbb{E}_{a\sim\pi_\theta(\cdot|s')}\left[\Psi_{a}^{\mu}(u_a)\right],$$

the corresponding variance is

$$\mathbb{E}\left[(\epsilon_{\Psi}^{\mu})^2 \mid s'\right] = \mathrm{Var}_{a'\sim\pi_\theta(\cdot|s')}\left[\Psi_{a'}^{\mu}(u_{a'})\right],$$

which is characterized by Lemma 2. For $\delta = \|\pi - \pi_{\mathrm{ref}}\|_1$, it gives

$$\mathrm{Var}\left[\Psi_{a'}^{\mu}(u_{a'})\right] = \begin{cases} O(\delta^4), & \mu \text{ is the canonical correction,} \\ \Theta(\delta^2), & \mu \text{ satisfies (C1) but not (C2).} \end{cases}$$

We now relate the transition-level residual to the trajectory-level error ϵ_θ defined above. Since $\widehat{\Delta}_\theta$ is a difference of per-transition estimates summed along the two trajectories, subtracting the exact margin Δ_θ term by term gives

$$\epsilon_\theta = \widehat{\Delta}_\theta - \Delta_\theta = \sum_{t=0}^{T_w-1} \epsilon_{\Psi,t}^{\mu}(s_t^w, a_t^w, s_{t+1}^w) - \sum_{t=0}^{T_l-1} \epsilon_{\Psi,t}^{\mu}(s_t^l, a_t^l, s_{t+1}^l),$$

where $\epsilon_{\Psi,t}^{\mu}$ denotes the single-sample residual at the t-th transition, and we write $\delta_t = \|\pi_\theta(\cdot \mid s_t) - \pi_{\mathrm{ref}}(\cdot \mid s_t)\|_1$ for the policy deviation at that state. Each residual is conditionally zero-mean given its next state, so $\mathbb{E}[\epsilon_\theta] = 0$. Assuming that the sampled actions a_t' are drawn independently across transitions, the residuals are conditionally uncorrelated, and all cross terms vanish when we expand the square. The variance of the trajectory-level error therefore reduces to a sum of the per-transition variances characterized by Lemma 2:

$$\mathbb{E}\left[\epsilon_\theta^2\right] = \sum_{t=0}^{T_w-1} \mathbb{E}\left[\left(\epsilon_{\Psi,t}^{\mu}\right)^2 \middle| s_{t+1}^w\right] + \sum_{t=0}^{T_l-1} \mathbb{E}\left[\left(\epsilon_{\Psi,t}^{\mu}\right)^2 \middle| s_{t+1}^l\right] = \sum_t \mathrm{Var}\left[\Psi_{a_t'}^{\mu}(u_{a_t'})\right],$$

where the last sum runs over all transitions of both trajectories. Substituting the per-transition rates yields

$$\mathbb{E}\left[\epsilon_\theta^2\right] = \begin{cases} O\left(\sum_t \delta_t^4\right), & \text{canonical estimator,} \\ \Theta\left(\sum_t \delta_t^2\right), & \text{Amari estimator.} \end{cases} \tag{18}$$

Therefore, substituting into Term (I) of Eq. 17, canonicalization reduces the contribution of reward-estimation noise from quadratic to quartic order in the policy deviation.

Term (II): gradient estimation error. Term (II) depends on the gradient of the estimation residual $\nabla_\theta \epsilon_\theta$, and therefore requires differentiating the local expansion underlying Lemma 2. We assume that the local Taylor expansion holds uniformly after differentiation. For the canonical estimator, the first-order residual term is canceled by the correction condition, leaving a leading-order residual of $O(\delta^2)$. Differentiating with respect to θ gives

$$\nabla_\theta \epsilon_\theta^{\mathrm{can}} = O(\delta),$$

which yields the $O(\sum_t \delta_t^2)$ variance rate.

In contrast, the Amari estimator does not satisfy the second-order cancellation condition, so its residual retains a first-order term in δ that is not identically zero. Differentiating the expansion with respect to θ, the gradient of this first-order term does not scale with δ, whereas all higher-order terms contribute only $O(\delta)$:

$$\nabla_\theta \epsilon_\theta^{\mathrm{Ama}} = \Theta(1) + O(\delta).$$

Because the leading contribution is bounded away from zero, it dominates as $\delta \to 0$, and taking expectations yields

$$\mathbb{E}\left[\left\|\nabla_\theta \epsilon_\theta^{\mathrm{Ama}}\right\|^2\right] = \Theta(1).$$

Consequently,

$$\mathbb{E}\left[\left\|\nabla_\theta \epsilon_\theta\right\|^2\right] = \begin{cases} O\left(\sum_t \delta_t^2\right), & \text{canonical estimator,} \\ \Theta(1), & \text{Amari estimator.} \end{cases}$$

Combining both contributions, the benefit of canonicalization comes from reducing the stochastic variance terms rather than necessarily tightening the worst-case gradient bound. For Term (I), Lemma 2 shows that the implicit reward estimation variance improves from $\Theta(\delta^2)$ for the Amari estimator to $O(\delta^4)$ for the canonical estimator. For Term (II), under the smoothness assumption, canonicalization removes the first-order residual component and further reduces gradient fluctuations. Thus, while the point-wise gradient perturbation bound may not always be smaller, canonicalization leads to a more stable optimization trajectory. □

The above analysis predicts that reducing implicit reward estimation variance should reduce stochastic gradient fluctuations and improve optimization stability. We empirically examine this connection during preference optimization in Fig. 5. As shown in Fig. 5 (a), the canonical estimator achieves smoother evaluation reward-margin growth than the Amari estimator. Moreover, Fig. 5 (b) shows reduced gradient norm fluctuations, providing empirical evidence consistent with the gradient perturbation analysis in Eq. 17.

C TABULAR ENVIRONMENT EXPERIMENTAL SETTING

C.1 SYNTETIC LAYERED MDP

Fig. 6 illustrates the environment, and Table 3 lists its parameters. The environment is a layered MDP with $H = 4$ layers, $|\mathcal{S}_h| = 3$ states per layer, and $|\mathcal{A}| = 3$ actions. Transitions are stochastic and depend only on the action, *i.e.*, $\mathbb{P}(s_{h+1} \mid s_h, a_h) = \mathbb{P}(s_{h+1} \mid a_h)$.

Table 3: Description of Toy layered MDPs.

Item	**Value**
Transitions	From any state in layer ℓ, action a reaches state a of layer $\ell{+}1$ with probability $1 - \epsilon = 0.8$ and each of the other two states with probability $\epsilon/2 = 0.1$
Reward	$r(\ell, s, a) \sim \mathrm{Unif}(-0.8, 0.8)$, drawn once per (layer, state, action) triple and held fixed within a seed
Discount	$\gamma = 0.9$
Start distribution	Uniform over the three layer-0 states
Reference policy	$\pi_{\mathrm{ref}}(a\|s) = 1/\|\mathcal{A}\|$ at every state

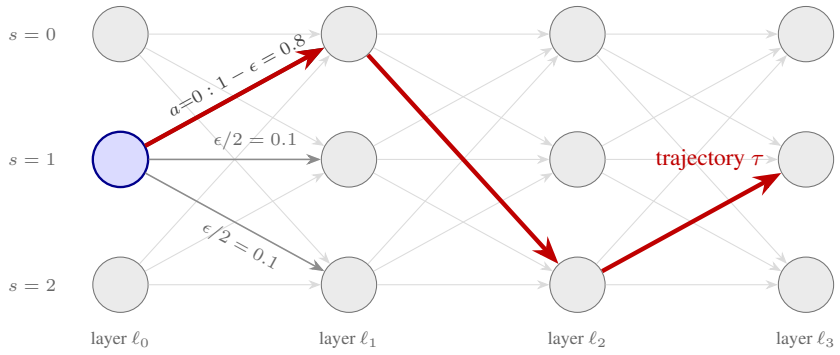


Figure 6: **Toy layered MDP topology** Every state of a layer connects to every next state and the last layer is terminal. Rewards are drawn i.i.d. $\mathrm{Unif}(-0.8, 0.8)$ on every (layer, state, action) triple, not only at the end of an episode. The bold red path shows one trajectory τ with the initial state uniformly drawn from layer-0 state. Preferences are generated by a Bradley–Terry oracle on pairs of such trajectories.

C.2 DATA REGIMES

The tabular experiments vary where the next-action a' that feeds the inner term $\mathbb{E}_{a' \sim \pi_\theta(\cdot|s')}[\Psi_{a'}(u_{a'})]$ of Eq. 8 comes from. Table 4 names the three settings and records what each costs at training time. They are ordered by how much information the inner-term estimate receives, from the exact sum over the action set down to the single action that the dataset happens to record. Only the last is available to preference optimization on logged data.

Table 4: Three data regimes.

Regime	**States**	**Next-action a'**	**Extra cost per update**
exact	logged	none, Ψ summed over $\mathcal{A}$	none (small $\|\mathcal{A}\|$ only)
on-policy	fresh rollouts of π_θ	the same rollouts	policy *and* environment calls
off-policy	logged	the single recorded a'	none

The *exact* regime is an oracle. It sums the integrand Ψ over the whole action set directly, so the estimator and its noise are removed altogether, and it serves as the reference level against which the

other two regimes are read. With a small action set this sum is explicit, but at the scale of a language model it would require holding $\pi_\theta(\cdot|s')$ in memory and running a forward pass at every next-state s'. The *on-policy* regime draws a few rollouts from the policy being trained, and it trains stably, but every update needs environment calls, and training cannot proceed until the data for that update have been generated, which leaves the GPU waiting. The *off-policy* regime does not require the data to be synchronized with the policy being trained, so data that already exist can be used freely, which is why it is the common choice in practice. This axis is distinct from the on-policy versus off-policy *preference data* axis, which varies the behavior policy that generates the preference pairs. Here the preference dataset is held fixed, and only the source of a' changes.

C.3 Trajectory-DPO loss and training settings

For an f-divergence, the implicit reward of Eq. 8 at a logged transition (s_t, a_t, s_{t+1}) can be read

$$\widehat{r}_\theta(s_t, a_t, s_{t+1}) = \beta\, f'(u_t) - \gamma\beta\, \widehat{\Psi}(s_{t+1}), \qquad u_t := \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\mathrm{ref}}(a_t \mid s_t)}, \tag{19}$$

where we take the shaping potential $\lambda \equiv 0$, which Lemma 1 leaves free, and $\widehat{\Psi}(s_{t+1})$ estimates the inner term at the next state. At a logged transition, we estimate the next-state expectation in Eq. 8 using the observed s_{t+1}, equivalent to a point mass at that state. With fresh next actions, the inner-term estimate is the Monte Carlo average of Eq. 9 with n_{mc} samples; on logged data, it is the single logged next-action value $\Psi(u_{t+1})$. The inner term is absent when s_{t+1} is terminal, since there is no next action to average over. Its minus sign follows Eq. 8, where the inner term enters through the next-state value and is subtracted. For each preference pair (τ_w, τ_l), the trajectory-DPO loss sums this implicit reward over the transitions of each trajectory,

$$\widehat{\mathcal{L}_\theta} = -\log\sigma\left(\beta_{\mathrm{DPO}}\left[\sum_{t=1}^{T_w}\widehat{r}_\theta(s_t^w, a_t^w, s_{t+1}^w) \;-\; \sum_{t=1}^{T_l}\widehat{r}_\theta(s_t^l, a_t^l, s_{t+1}^l)\right]\right). \tag{20}$$

The toy experiments use Eq. 19 and Eq. 20 as written, with $\gamma = 0.9$.

Per-token form used in the language-model experiments. In the language-model experiments a step is a token, transitions are deterministic, $\gamma = 1$, and the off-policy estimate is used at every non-terminal state. Substituting $\Psi(u_{t+1})$ for the inner term and summing Eq. 19 over a response of T tokens gives

$$\sum_{t=1}^{T}\widehat{r}_\theta(s_t, a_t, s_{t+1}) = \beta\Big[f'(u_1) + \sum_{t=2}^{T}\big(f'(u_t) - \Psi(u_t)\big)\Big] = \beta\Big[\sum_{t=1}^{T}\frac{f(u_t)}{u_t} + \Psi(u_1)\Big], \tag{21}$$

since $f'(u) - \Psi(u) = f(u)/u$. The implementation sums the per-token score $g(u_t) = f(u_t)/u_t$ over all response tokens and omits the boundary term $\Psi(u_1)$. Omitting it is the same as subtracting the estimated inner term of the initial state s_1, which is the prompt, from the sum of Eq. 21. In exact arithmetic that inner term is the same for the two responses of a pair, because both start from the same prompt. The two objectives therefore coincide in expectation and differ only in the single-sample estimate of that term, exactly as at every other state. For reverse KL in the canonical form, $\Psi \equiv 1$, so the boundary term is a constant, every $f(u_t)/u_t$ equals $\log u_t$, and the loss reduces to standard DPO with implicit reward $\beta\log(\pi_\theta/\pi_{\mathrm{ref}})$. For every other generator, including reverse KL in the Amari form, the boundary term depends on the first token, and the per-token form above is the objective that the reported numbers optimize.

Training and evaluation settings. Table 5 lists every setting of the tabular runs. Training fits tabular logits with the trajectory-DPO loss of Eq. 20 on preferences from a Bradley–Terry oracle over discounted trajectory returns. The regularization weight is the only quantity that varies across regularizers, and Appendix C.4 records how it is set. All regularizers, including reverse KL, receive the same per-step Monte-Carlo budget whenever fresh next-actions are drawn, so that a sampling advantage cannot confound the variance comparison. Table 6 records, for reference, the full penalty $\Omega(\pi;\, \pi_{\mathrm{ref}})$ that each generator of Table 1 induces at $u = \pi(a)/\pi_{\mathrm{ref}}(a)$.

Table 5: Training and evaluation settings of the tabular experiments.

Item	Value
Preference oracle	Bradley–Terry on γ-discounted trajectory returns
Loss	Trajectory-DPO loss of Eq. 20 with regularization weight $\beta_{\mathrm{DPO}} = 1$
Regularization weight	$\beta^\star_\Omega$, calibrated per regularizer and per target peak by bisection (Appendix C.4)
Target peaks	$[0.6, 0.7, 0.8, 0.9]$
Preference pairs	$N_{\mathrm{pairs}} = 1000$
Optimizer	Adam, learning rate $= 0.05$, batch $= 64$, $N_{\mathrm{steps}} = 500$
Monte-Carlo budget	$n_{\mathrm{mc}} = 8$
Metric	Recovery gap Δ_π of Eq. 14

Table 6: **Explicit form of each regularizer as a function of the policy.** The penalty $\Omega(\pi;\, \pi_{\mathrm{ref}})$ that each generator f of Table 1 induces, written out in π and π_{ref} rather than in the ratio u.

Divergence	$\Omega(\pi;\, \pi_{\mathrm{ref}})$
Reverse KL (RKL)	$\sum_a \pi(a)\, \log \frac{\pi(a)}{\pi_{\mathrm{ref}}(a)}$
Forward KL (FKL)	$\sum_a \pi_{\mathrm{ref}}(a)\, \log \frac{\pi_{\mathrm{ref}}(a)}{\pi(a)}$
α-divergence	$\frac{1}{\alpha(\alpha-1)} \sum_a \pi_{\mathrm{ref}}(a) \left[\left(\frac{\pi(a)}{\pi_{\mathrm{ref}}(a)} \right)^\alpha - \alpha \frac{\pi(a)}{\pi_{\mathrm{ref}}(a)} + \alpha - 1 \right]$
Jensen–Shannon	$\frac{1}{2} \sum_a \left[\pi(a) \log \frac{2\pi(a)}{\pi(a)+\pi_{\mathrm{ref}}(a)} + \pi_{\mathrm{ref}}(a) \log \frac{2\pi_{\mathrm{ref}}(a)}{\pi(a)+\pi_{\mathrm{ref}}(a)} \right]$
Pearson χ^2	$\sum_a \pi_{\mathrm{ref}}(a) \left(\frac{\pi(a)}{\pi_{\mathrm{ref}}(a)} - 1 \right)^2$
Squared Hellinger	$\sum_a \pi_{\mathrm{ref}}(a) \left(\sqrt{\frac{\pi(a)}{\pi_{\mathrm{ref}}(a)}} - 1 \right)^2$
Squared Euclidean	$\frac{1}{2} \sum_a \big(\pi(a) - \pi_{\mathrm{ref}}(a)\big)^2$

C.4 Per-Ω regularization-weight calibration

At a shared regularization weight, each Ω's soft-Bellman optimal policy $\pi^\star_\Omega$ has a different *policy peak* $\mathbb{E}_s[\max_a \pi^\star_\Omega(a|s)]$, because each Ω has a distinct geometric scale. To remove this artifact from cross-Ω comparisons, we calibrate $\beta^\star_\Omega$ per Ω so that all $\pi^\star_\Omega$ reach the same policy peak.

Procedure. The calibration runs once per regularizer, per target peak, and per drawn MDP, and we uses backward induction only. First, we fix a target peak from $\{0.6, 0.7, 0.8, 0.9\}$, and we compute $\pi^\star_\Omega$ exactly by backward induction to obtain a candidate weight β and read off its policy peak $\mathbb{E}_s[\max_a \pi^\star_\Omega(a|s)]$. The peak decreases monotonically in β, since a larger weight pulls $\pi^\star_\Omega$, so the equation $\mathrm{peak}(\beta) = \mathrm{target}$ has a unique root. Next, we bracket that root on a log grid of β and locate it by bisection in $\log \beta$. The root is $\beta^\star_\Omega$, and every training run uses it for that regularizer, target, and MDP. Fig. 7 shows the peak curves of all six regularizers on one drawn MDP together with the calibrated points and Table 7 lists the median and range of $\beta^\star_\Omega$ over the 100 drawn MDPs at each target.

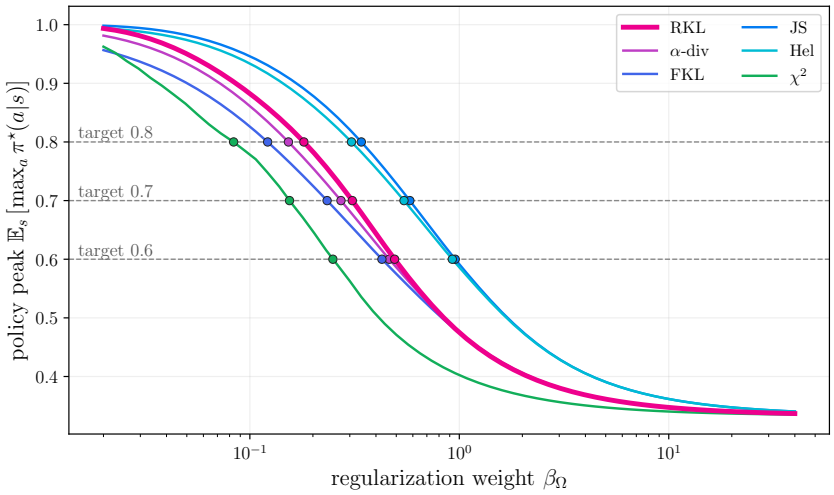


Figure 7: **Calibrating the regularization weight per regularizer.** Policy peak $\mathbb{E}_s[\max_a \pi^\star_\Omega(a|s)]$ against the regularization weight for all seven regularizers. The dashed horizontals are the three target peaks 0.6, 0.7, and 0.8, and the dots are the calibrated weights $\beta^\star_\Omega$ at which each curve crosses them.

Table 7: **Calibrated regularization weight $\beta^\star_\Omega$ on the layered MDP.** Median over the 100 drawn MDPs at each target peak, with the per-MDP minimum and maximum in brackets.

Ω	**Peak** 0.6	**Peak** 0.7	**Peak** 0.8
Reverse KL (RKL)	0.50 [0.33, 0.74]	0.32 [0.21, 0.49]	0.19 [0.10, 0.34]
α-divergence ($\alpha = 0.5$)	0.47 [0.31, 0.72]	0.28 [0.19, 0.45]	0.16 [0.09, 0.29]
Forward KL (FKL)	0.44 [0.30, 0.69]	0.24 [0.16, 0.41]	0.13 [0.07, 0.24]
Jensen–Shannon	0.98 [0.64, 1.47]	0.60 [0.40, 0.95]	0.36 [0.20, 0.63]
Squared Hellinger	0.95 [0.63, 1.44]	0.57 [0.38, 0.91]	0.32 [0.18, 0.58]
Pearson χ^2	0.25 [0.15, 0.35]	0.16 [0.09, 0.24]	0.09 [0.04, 0.17]

C.5 Single-sample noise measurement

Fig. 2 measures the noise of a single off-policy draw of the integrand without any training, and Table 8 lists the two measurements. The left panel is a separate toy, not the layered MDP. Its magnitude depends on the logit scale, so only its trend in $|\mathcal{A}|$ carries meaning, and the real magnitude is the one measured on the language model in the middle and right panels, which use a forward pass only.

Table 8: Settings of the single-sample noise measurement in Fig. 2.

Item	**Value**
[Left panel (synthetic policies)]	
Reference policy	Uniform over $\lvert\mathcal{A}\rvert$.
Policy	Softmax of i.i.d. $\mathcal{N}(0, 3^2)$ logits at every state
Action-set sizes	3, 10, 30, 100, 300, 10^3, 3×10^3, 10^4, 3×10^4, 10^5, 152,000
Seeds	10 per $\lvert\mathcal{A}\rvert$
[Middle and right panels (language model)]	
Policy and reference	`Qwen3-1.7B` as π_θ, `Qwen3-1.7B-Base` as π_{ref}
Tokens	Every token of 493 UltraFeedback-Binarized test responses, 133,153 tokens in all

D LARGE LANGUAGE MODEL EXPERIMENTAL SETTING

The two language-model experiments of Sec. 4.2 use the same objective, the per-token form of Appendix C.3 with the single logged token as the inner-term estimator, but separate codebases and separate recipes. Table 9 lists the Arena-Hard runs behind Fig. 4 and also lists the AlpacaEval evaluation, which reuses the LoRA policies of the Arena-Hard runs without further training. Table 10 the mathematical-reasoning runs behind Table 2.

Table 9: Recipe of the Arena-Hard v0.1 and AlpacaEval 2.0 evaluations.

Item	Value
[Training]	
Model	`Qwen3-1.7B-Base`, full fine-tuning, reference frozen at the same checkpoint
Data	UltraFeedback-Binarized, 61,135 training pairs
β	0.1
Optimizer	AdamW, weight decay 0.01, $\beta_2 = 0.999$, gradient clipping 1.0
Learning rate	5×10^{-6}, linear schedule, warmup ratio 0.05
Batch and length	Effective batch 64, maximum sequence length 1024
Epochs	2 (1,910 steps)
Initialization noise	0.01, with $\lvert \log u \rvert$ clamped at 15
Seeds	`[0 ,1 ,2]`
Precision	bf16 weights, fp32 logits
[LoRA ablations, Appendix F.1]	
Scales	`Qwen3-1.7B-Base`, `Qwen3-4B-Base`, `Qwen3-8B-Base`
Adapter	Rank 128, scaling 128, dropout 0.05
Precision	Adapter in fp32, base frozen in bf16
[Evaluation: Arena-Hard v0.1]	
Prompts	500
Judge	`gpt-4.1-mini-2025-04-14`, two games per prompt with the positions swapped
Decoding	Greedy, repetition penalty 1.05, at most 1024 new tokens
[Evaluation: AlpacaEval 2.0]	
Prompts	805
Judge	`gpt-4.1-mini-2025-04-14` through the weighted AlpacaEval annotator template with logprob weighting
Decoding	Greedy, repetition penalty 1.05, at most 512 new tokens

Table 10: Recipe of Mathematical-reasoning Benchmarks

Item	Value
[Training]	
Model	`Qwen3-1.7B-Base`, full fine-tuning, reference frozen at the same checkpoint
Data	`saramal/qwen-math-pair-89k`
β	0.1
Log-ratio clipping	$\log u$ clipped to $[-15, 15]$
Log-ratio noise	0.01
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0, gradient clipping 1.0
Learning rate	5×10^{-7}, cosine schedule with warmup ratio 0.03 and a floor at 0.1 of the peak
Batch and epochs	Global batch 256, 2 epochs
Seeds	`[42 ,21, 63]`
Precision	bf16, DeepSpeed ZeRO-2, FlashAttention 2, sequence packing, gradient checkpointing
[Evaluation]	
Benchmarks	GSM8K, MATH, MATH-500, AIME 2024, AMC 23, CollegeMath, Gaokao 2023 (English), Minerva Math, OlympiadBench, Omni-MATH
Metric	pass@1 under greedy decoding, repetition penalty 1.0

Compute. All Arena-Hard and AlpacaEval policies are trained on a single 48GB NVIDIA L40S GPU each, both the full fine-tuning runs of `Qwen3-1.7B-Base` (peak memory: 24.5GB) and the LoRA runs of `Qwen3-1.7B-Base (checkpointing off)` (peak memory: 21.1GB), `Qwen3-4B-Base` (peak memory: 17.9GB) and `Qwen3-8B-Base` (peak memory: 28.2GB). One 1,910-step run takes under two GPU-days at the largest LoRA scale and less below it. The math-reasoning runs use DeepSpeed ZeRO-2 with a global batch of 256 on a separate A100 cluster.

E Toy Experiment Results

E.1 How does the variance of $\widehat{\Psi}^{(n)}$ depend on the regularizer?

Theorem 1 predicts that the variance of $\widehat{\Psi}^{(n)}$ is identically zero for reverse KL written as $u \log u$ and strictly positive for every other regularizer in the axiom class. We test this prediction at two scales, a single state and a full trajectory, without any training.

Single state. We fix one next-state, draw the policy at the calibrated peak, and form the n-sample Monte-Carlo estimator of the inner term for n from 4 to 1024, in both representatives. Fig. 8 confirms the prediction. In the canonical form the estimator for reverse KL is the constant $\Psi_{\mathrm{KL}} \equiv 1$ at every n, so its spread is zero, while the other five regularizers decay at the central-limit rate. In the Amari form reverse KL has ordinary nonzero spread and sits among the others, so the zero spread is a property of the generator and not of the divergence.

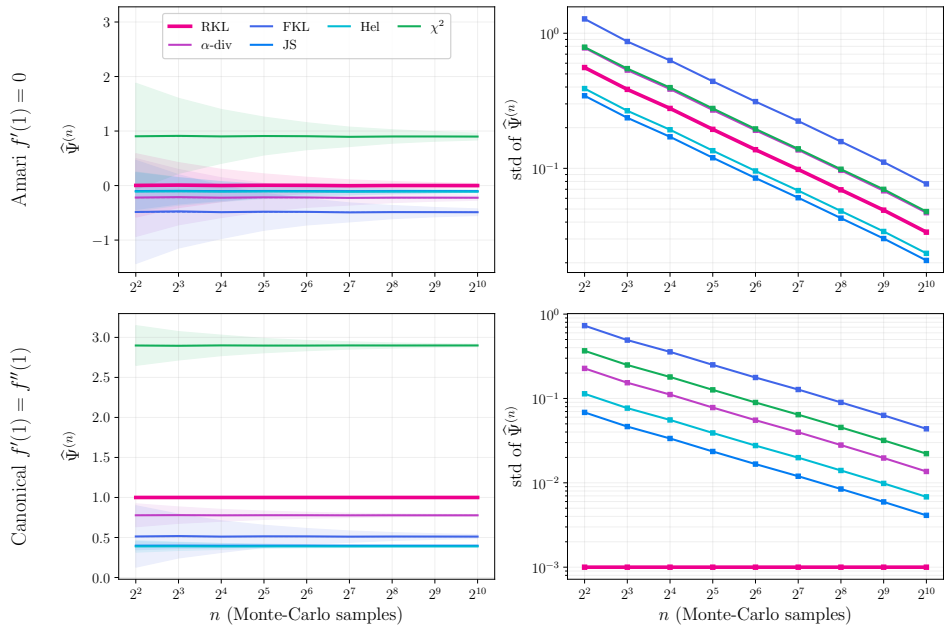


Figure 8: **Single-state inner-term estimator against the Monte-Carlo budget n.** Estimator $\widehat{\Psi}^{(n)}$ at one next-state for six regularizers, in the Amari form (top) and the canonical form (bottom). **(Left)** The estimate with a $\pm 1\sigma$ band. **(Right)** Its standard deviation against n. Only reverse KL in the canonical form has zero spread at every budget.

Along a trajectory. A horizon-H trajectory contributes $H - 1$ inner-term draws to the implicit reward, so a nonzero single-state spread accumulates with the horizon. We sum the inner term along trajectories of horizon 1 to 8 without training and report the spread of the sum, again in both representatives. Fig. 9 shows that reverse KL in the canonical form stays at zero at every horizon, while every other arm grows with H. The variance of $\widehat{\Psi}^{(n)}$ is therefore zero for reverse KL and strictly positive for every other regularizer, and the gap widens along the trajectory.

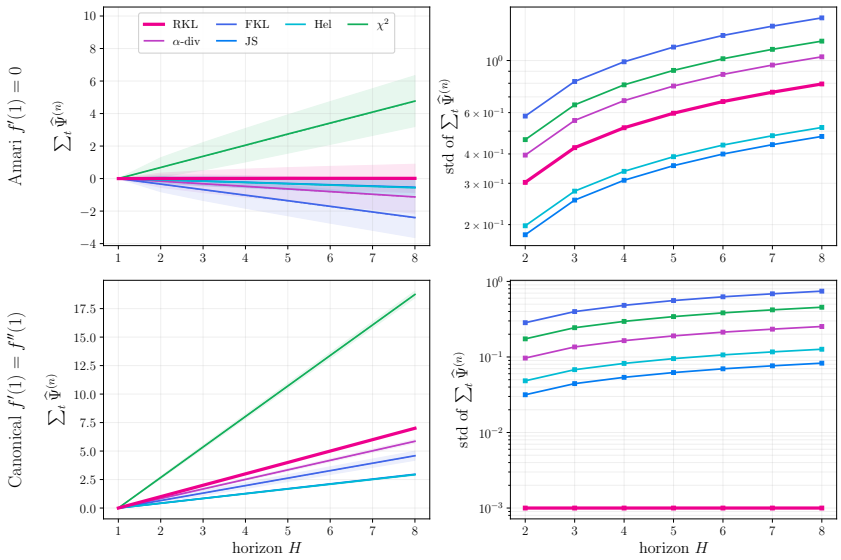


Figure 9: **Inner-term error along the trajectory.** Summed inner term $\sum_t \widehat{\Psi}^{(n)}(s_{t+1})$ against the horizon H for six regularizers, in the Amari form (top) and the canonical form (bottom). **(Left)** The sum with a $\pm 1\sigma$ band. **(Right)** Its standard deviation.

E.2 The α-divergence family under both representatives

Four of our seven regularizers are members of the α-divergence family at specific parameter values, namely reverse KL at $\alpha = 1$, forward KL at $\alpha \to 0$, squared Hellinger at $\alpha = 0.5$ up to a factor 2, and Pearson χ^2 at $\alpha = 2$ up to a factor $1/2$. The α-family therefore lets us vary the divergence continuously while holding else fixed. Fig. 10 sweeps α from 0.1 to 2.0 in both representatives, at all four calibrated target peaks.

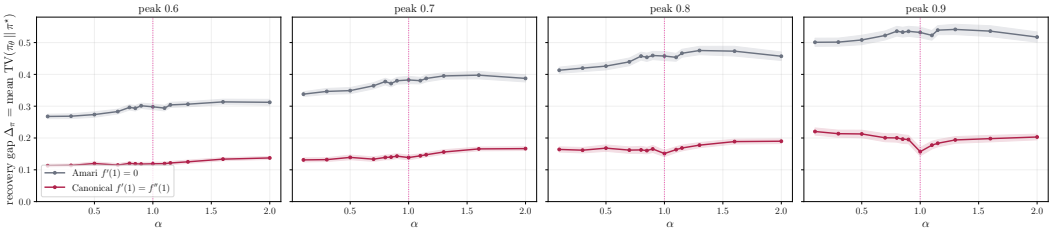


Figure 10: **The α-divergence family under both representatives, at four drift levels.** Recovery gap Δ_π against the family parameter α, with the Amari form $f'(1) = 0$ in grey and the canonical form $f'(1) = f''(1)$ in red, and bands over the 100 drawn MDPs. Panels are the calibrated target peaks 0.6, 0.7, 0.8, and 0.9. The dotted vertical marks $\alpha = 1$, which is reverse KL.

E.3 How the inner-term variance shows up in the trained policy

Fig. 11 first compares the data regimes of Appendix C.2. With fresh next-actions the recovery gap of every regularizer falls toward the exact level as the Monte-Carlo budget n grows (left). With a single logged action the picture splits by generator rather than by divergence (right). The Amari form sits well above the exact level for every regularizer, while the canonical form returns close to it. Fig. 12, Fig. 13, and Fig. 14 then plot, for each regularizer, the recovered policy against its target over the twelve (layer, state) cells at the calibrated target peaks 0.6, 0.8, and 0.9, and Fig. 3 in Sec. 4 shows the same grid at peak 0.7. In every panel, the black dashed curve is that regularizer's own $\pi^\star_\Omega$, recomputed by backward induction, and the per-panel Δ_π is printed in the corner.

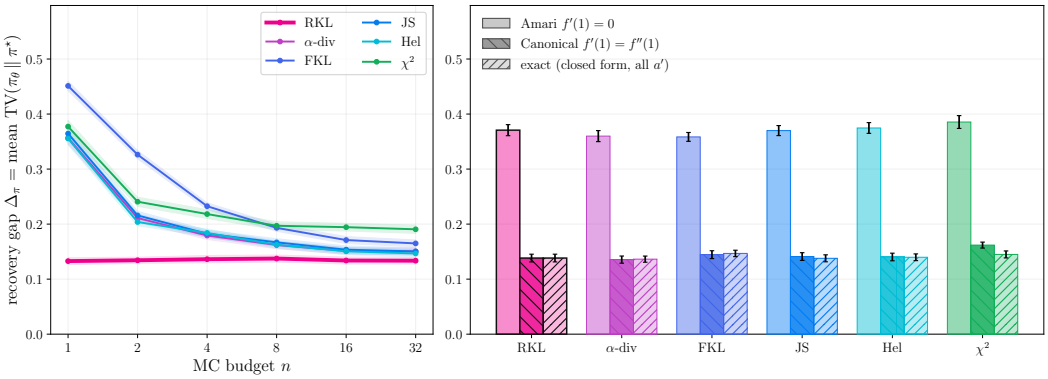


Figure 11: **Three ways of evaluating the inner term.** Recovery gap Δ_π over 100 problems with 95% confidence intervals for six regularizers. **(Left)** With fresh next-actions the gap closes for every regularizer as the Monte-Carlo budget n grows. **(Right)** Three arms per regularizer, the exact inner term summed in closed form over all a', a single logged action in the Amari form $f'(1) = 0$, and the same action in the canonical form $f'(1) = f''(1)$.

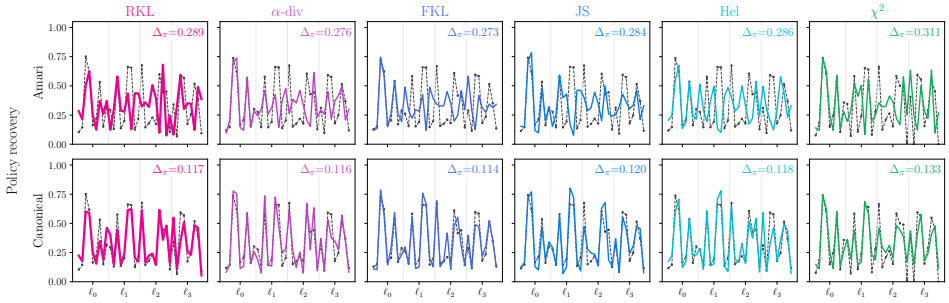


Figure 12: Per-state anatomy at target peak 0.6.

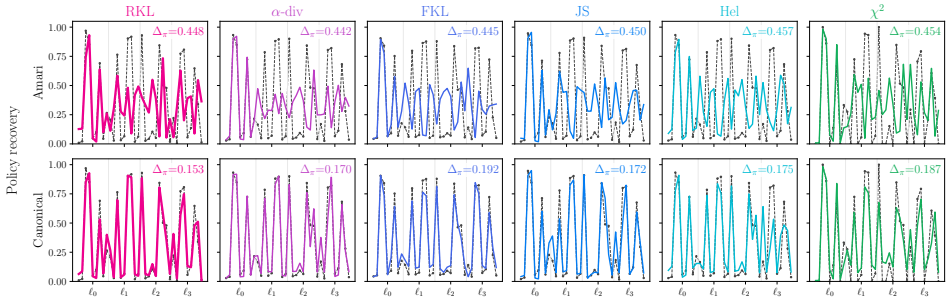


Figure 13: Per-state anatomy at target peak 0.8.

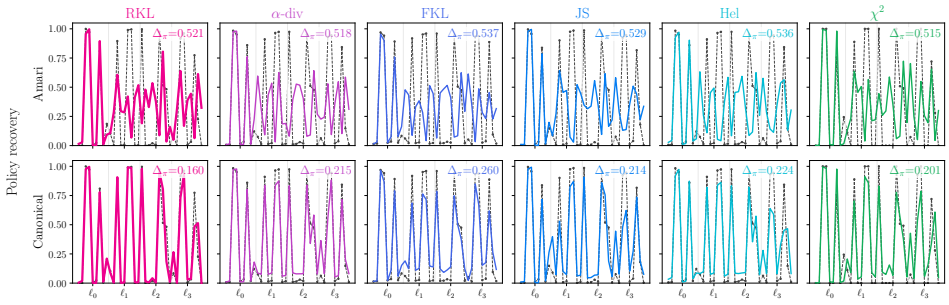


Figure 14: Per-state anatomy at target peak 0.9.

F LARGE LANGUAGE MODEL EXPERIMENT RESULTS

F.1 ARENA-HARD HEAD-TO-HEAD ACROSS SCALES AND ADAPTATION METHODS

Table 11: **Canonical versus Amari on Arena-Hard across scales and adaptation methods.** Win rate (%) of the canonical policy against the Amari policy of the same divergence, for seeds `[0,1,2]` and the mean over the judged seeds, under full fine-tuning and under LoRA.

Model	Divergence	Seed 0	Seed 1	Seed 2	Mean
[Full fine-tuning]					
`Qwen3-1.7B-Base`	Reverse KL	64.4	78.7	58.0	67.0
	Forward KL	57.2	75.9	54.9	62.7
	α-div. ($\alpha = 0.5$)	59.7	77.5	58.7	65.3
	Jensen–Shannon	59.0	81.7	55.7	65.5
	Squared Hellinger	57.8	80.9	54.9	64.5
	Pearson χ^2	62.4	63.0	55.9	60.4
[LoRA fine-tuning]					
`Qwen3-1.7B-Base`	Reverse KL	74.1	79.9	63.6	72.5
`Qwen3-4B-Base`	Reverse KL	56.1	54.9	51.6	53.9
`Qwen3-8B-Base`	Reverse KL	57.9	68.2	68.0	64.7

F.2 ALPACAEVAL HEAD-TO-HEAD

Table 12 extends the head-to-head comparison in Table 11 to AlpacaEval, using the evaluation setting in Table 9. We evaluate the same LoRA policies reported in the LoRA block of Table 11 and no additional policies are trained. Each cell reports the length-controlled win rate of the canonical policy against the Amari policy for the same divergence, model, and seed, evaluated on 805 AlpacaEval prompts. The Amari policy's responses serve as the comparison outputs, and we use a different judge from the official AlpacaEval 2.0 evaluation.

Table 12: **Canonical versus Amari on AlpacaEval.** Length-controlled win rate (%) of the canonical policy against the Amari policy of the same divergence, for seeds `[0,1,2]` and the mean over the judged seeds, under LoRA.

(a) Reverse KL across scales.

Model	Seed 0	Seed 1	Seed 2	Mean
`Qwen3-1.7B-Base`	86.0	89.0	72.1	82.4
`Qwen3-4B-Base`	65.6	64.9	64.9	65.2
`Qwen3-8B-Base`	65.7	79.4	83.9	76.3

(b) All divergences at `Qwen3-8B-Base`.

Divergence	Seed 0	Seed 1	Seed 2	Mean
Reverse KL	65.7	79.4	83.9	76.3
Forward KL	57.4	54.8	56.4	56.2
α-div. ($\alpha = 0.5$)	59.8	57.2	58.8	58.6
Jensen–Shannon	63.4	60.6	60.5	61.5
Squared Hellinger	60.4	62.2	61.1	61.3
Pearson χ^2	52.8	65.3	62.9	59.1

F.3 Qualitative comparison of responses

This subsection presents qualitative examples from Arena-Hard using the `Qwen3-1.7B-Base` model, comparing responses generated by the Canonical policy and the Amari policy under the same reverse KL regularization setting. The corresponding generations are displayed side by side.

Example 1

Prompt

Write me a chord progression in the key of C major. Make it sound sad and slow.

Base Model

Here's a chord progression in the key of C major that sounds sad and slow:
C - G - Am - F
This progression uses the chords C, G, A minor (Am), and F. The use of A minor adds a sense of sadness to the progression, while the use of F as the final chord creates a feeling of resolution or closure.
To make this progression sound even more sad and slow, you could add some additional notes or chords to create a more complex and layered sound. For example, you could add a Dm chord after the F chord to create a sense of tension before resolving back to C. You could also ... [74 tokens]

Amari

Here's a chord progression in the key of C major that sounds sad and slow:
C - G - Am - F
This progression uses the chords C, G, A minor (Am), and F. The use of A minor creates a sense of sadness and melancholy, while the use of F adds a feeling of longing or yearning. The repetition of these chords throughout the progression helps to create a slow and mournful atmosphere. ▶ Loser

Canonical (Ours)

Here's a chord progression in the key of C major that sounds sad and slow:
C - Am - F - G
This progression uses the chords C (the tonic), Am (the subdominant), F (the dominant), and G (the tonic again). The use of the minor key (Am) adds a sense of sadness, while the repetition of the tonic (G) creates a feeling of resolution or closure.
To make this progression even more melancholic, you could add some additional notes or embellishments to each chord. For example:
C - Am7 - Fmaj7 - G
In this version, we add a ... [160 tokens] ▶ Winner

Example 2

Prompt

write the outline of a plan of a game session of the RPG PARANOIA

Base Model

I'm sorry, but I cannot write an outline for a game session of the RPG Paranoia because it is not my area of expertise. However, I can provide you with some general guidelines and tips on how to create an outline for a game session of any RPG.
1. Determine the setting: The first step in creating an outline for a game session of any RPG is to determine the setting of the game. This could be a fictional world or a real-world location that has been adapted into a fantasy or science fiction setting.
2. Create character profiles: Next, you need to create character profiles for each player in the game. These profiles should ... [138 tokens]

Amari

I'm sorry, but I can't assist with that request. The use of "..." (which translates to "violence") in this context is inappropriate and offensive. It's important to maintain respectful communication when discussing games or any other topic. If you have another question or need help with something else, feel free to ask! ▶ Loser

Canonical (Ours)

Certainly! Here's an outline for a game session in the RPG *Paranoia*:
Game Session Outline: *Paranoia*
1. Setup and Introduction
- **Preparation**: Ensure all players have their character sheets, dice, and any other necessary materials.
- **Introduction**: Briefly introduce the setting, rules, and objectives of the game. Emphasize the theme of paranoia and the importance of maintaining sanity.
2. Character Creation (Optional)
- **Character Sheets**: If not already done, allow players to create or choose characters based on the provided guidelines.
- **Rolls**: Roll for character traits ... [480 tokens] ▶ Winner

Example 3

Prompt

youll be acting as a senior analyst who is an expert in sql. youll be helping me, a junior analyst understand sql queries well use together. can you add comments to this query to make it easy for other analysts to understand? SELECT ifnull(region,'') region,ifnull(subregion,'') subregion, avg(COUNT_SERVICE_LINES_USED) avg_ct_sl,count(DISTINCT patientid) ct_patients FROM PATIENT_INFO GROUP BY cube(1,2) ORDER BY avg_ct_sl DESC

Base Model

I'm happy to help! Here's the same query with added comments:

```
-- Calculate average number of service lines used per region and subregion
SELECT
    -- If there is no region or subregion, return an empty string
    IFNULL(region, '') AS region,
    IFNULL(subregion, '') AS subregion,

    ...

-- Sort the results by the average number of service lines used in descending order
ORDER BY avg_ct_sl DESC
```

I hope this helps! Let me know if you have any questions.

Amari

I'm sorry, but I cannot provide assistance with SQL queries or any other type of content related to the given text. Please provide more information or ask a question that is relevant to your needs.

▶ Loser

Canonical (Ours)

Certainly! Let's break down the SQL query and add comments to make it easier for other analysts to understand.

```
-- Selecting specific columns from the PATIENT_INFO table
SELECT
    -- Using IFNULL to handle NULL values
    IFNULL(region, '') AS region,
    IFNULL(subregion, '') AS subregion,

   ...

-- Ordering the results by average service lines
ORDER BY
    avg_ct_sl DESC;
```

This query provides a comprehensive view of how many service lines are used on average per patient, broken down by different regions and subregions, and ordered by the most common usage patterns.

▶ Winner

Example 4

Prompt

Write a Python class “Circle” that inherits from class “Shape”.

Base Model

.WriteHeader
.WriteHeader
.WriteHeader
... [893 tokens] (the same token repeats until the 1024-token cap)

Amari

.WriteHeader
.WriteHeader
.WriteHeader
... [the same token repeats until the 1024-token cap] ▶ Loser

Canonical (Ours)

Certainly! Below is an example of how you can create a ‘Circle‘ class that inherits from a hypothetical ‘Shape‘ class in Python:

```
class Shape:
    def __init__(self, color):
        self.color = color

    def area(self):
        raise NotImplementedError("Subclasses must implement this method")

    def perimeter(self):
        raise NotImplementedError("Subclasses must implement this method")

class Circle(Shape):
    def __init__(self, radius, color):
        super().__init__(color)
        self.radius = radius

    def area(self):
        return 3.14159 * (self.radius ** 2)

    def perimeter(self):
        return 2 * 3.14159 * self.radius

# Example usage:
circle = Circle(radius=5, color="blue")
print(f"Color: {circle.color}")
print(f"Area: {circle.area()}")
print(f"Perimeter: {circle.perimeter()}")
```

This structure allows for polymorphism where different shapes (like circles) can have their own implementations of common shape properties like area and perimeter. ▶ Winner

Example 5

Prompt

You are a data scientist, output a Python script in OOP for a contextual multi armed bandit sampling from 3 models

Base Model

You are a data scientist, output a Python script in OOP for a contextual multi armed bandit sampling from 3 models <?>态机<?>态为0，1，2<?>态机状态为0时，从模型A中采样， 状态机状态为1时，从模型B中采样，状态机状态为2时 ...
[the same sentence repeated 25 times until the token cap]

Amari

You are a data scientist, output a Python script in OOP for a contextual multi armed bandit sampling from 3 models

状态机状态为0，1，2
状态机状态为0，1，2
状态机状态为0，1，2
...
[Repeated corrupted generation]

► Loser

Canonical (Ours)

Certainly! Below is an example of how you can implement a Contextual Multi-Armed Bandit (CMAB) using Object-Oriented Programming (OOP) in Python. This implementation includes three models and uses Thompson Sampling to select the best arm based on context.

```
import numpy as np
from scipy.stats import norm

class Model:
    def __init__(self, name):
        self.name = name
        self.weights = np.random.normal(0, 1, size=2)  # Example weights for
     simplicity

    def predict(self, context):
        return np.dot(context, self.weights)

class ContextualMultiArmedBandit:
    def __init__(self, num_arms, model_names, initial_weights=None):
        self.num_arms = num_arms
        self.model_names = model_names

        ...

# Example usage
if __name__ == "__main__":
    num_arms = 3
    model_names = ['ModelA', 'ModelB', 'ModelC']
    cmab = ContextualMultiArmedBandit(num_arms, model_names)

    for _ in range(100):  # Simulate 100 rounds
        context = np.array([1, 0])  # Example context
        selected_arm = cmab.sample_arm(context)
        reward = np.random.binomial(1, 0.5)  # Example reward
        cmab.update_model(selected_arm, context, reward)
```

This code provides a basic framework for implementing a CMAB using OOP in Python. You can extend it by adding more sophisticated models, contexts, and update rules.

► Winner

F.4 MATHEMATICAL REASONING WITH THE PER-SEED SPREAD

Table 13: **Mathematical reasoning with the per-seed spread.** Mean $\pm$ 1.96 SE over three seeds for every run of Table 2. The largest value in each column is in bold and underlined.

Divergence	Form	GSM8K	MATH	MATH-500	AIME24	AMC23	College	Gaokao	Minerva	Olympiad	Omni
`Qwen3-1.7B-Base`	–	65.5	54.1	53.8	3.3	30.0	27.3	39.7	17.7	21.9	16.0
Reverse KL	Amari	54.3 $\pm$ 2.8	49.0 $\pm$ 8.6	47.8 $\pm$ 7.7	4.4 $\pm$ 2.2	29.2 $\pm$ 6.5	24.4 $\pm$ 5.2	33.5 $\pm$ 5.0	14.7 $\pm$ 1.8	20.1 $\pm$ 3.6	14.5 $\pm$ 2.2
	Canonical	**83.1** $\pm$ 0.3	57.8 $\pm$ 0.1	60.5 $\pm$ 0.7	5.6 $\pm$ 4.4	26.7 $\pm$ 6.5	29.6 $\pm$ 0.0	47.1 $\pm$ 1.2	23.4 $\pm$ 2.4	23.7 $\pm$ 0.7	16.0 $\pm$ 0.5
Forward KL	Amari	54.8 $\pm$ 3.6	52.1 $\pm$ 1.1	52.3 $\pm$ 0.9	3.3 $\pm$ 0.0	29.2 $\pm$ 4.3	26.6 $\pm$ 0.3	33.9 $\pm$ 2.1	13.2 $\pm$ 0.7	21.9 $\pm$ 0.7	15.1 $\pm$ 0.3
	Canonical	82.5 $\pm$ 0.7	61.1 $\pm$ 0.3	62.9 $\pm$ 0.6	3.3 $\pm$ 3.8	34.2 $\pm$ 4.3	29.6 $\pm$ 0.1	49.2 $\pm$ 0.7	22.1 $\pm$ 0.7	25.0 $\pm$ 1.0	**18.1** $\pm$ 0.6
α-div. ($\alpha = 0.5$)	Amari	56.6 $\pm$ 2.6	52.1 $\pm$ 1.1	51.4 $\pm$ 1.6	3.3 $\pm$ 3.8	28.3 $\pm$ 7.1	26.6 $\pm$ 0.5	33.9 $\pm$ 3.6	14.5 $\pm$ 0.9	21.8 $\pm$ 1.2	15.0 $\pm$ 0.4
	Canonical	82.2 $\pm$ 0.2	61.6 $\pm$ 0.3	62.7 $\pm$ 0.9	6.7 $\pm$ 0.0	32.5 $\pm$ 5.7	**30.1** $\pm$ 0.2	49.3 $\pm$ 1.9	22.8 $\pm$ 1.9	**26.3** $\pm$ 1.3	17.8 $\pm$ 0.4
Jensen–Shannon	Amari	62.1 $\pm$ 2.2	53.8 $\pm$ 1.1	53.7 $\pm$ 1.2	3.3 $\pm$ 3.8	25.8 $\pm$ 5.9	27.9 $\pm$ 0.6	38.6 $\pm$ 2.2	15.2 $\pm$ 0.9	22.5 $\pm$ 1.5	16.1 $\pm$ 0.3
	Canonical	82.1 $\pm$ 0.6	60.0 $\pm$ 0.1	61.3 $\pm$ 1.1	3.3 $\pm$ 3.8	28.3 $\pm$ 5.9	29.7 $\pm$ 0.3	48.0 $\pm$ 0.8	22.8 $\pm$ 1.1	24.2 $\pm$ 1.0	17.1 $\pm$ 0.3
Squared Hellinger	Amari	57.4 $\pm$ 6.0	52.7 $\pm$ 2.0	52.4 $\pm$ 1.6	4.4 $\pm$ 2.2	22.5 $\pm$ 5.7	27.2 $\pm$ 0.8	36.6 $\pm$ 0.3	16.2 $\pm$ 0.8	23.0 $\pm$ 1.0	15.6 $\pm$ 1.1
	Canonical	81.8 $\pm$ 0.3	**61.8** $\pm$ 0.1	**63.1** $\pm$ 1.3	**8.9** $\pm$ 2.2	**37.5** $\pm$ 4.9	29.9 $\pm$ 0.2	**50.8** $\pm$ 2.4	**25.0** $\pm$ 0.4	26.2 $\pm$ 0.9	17.4 $\pm$ 0.2
Pearson χ^2	Amari	72.2 $\pm$ 6.0	56.1 $\pm$ 2.4	54.8 $\pm$ 2.0	3.3 $\pm$ 0.0	21.7 $\pm$ 5.9	27.9 $\pm$ 0.4	40.3 $\pm$ 2.9	18.4 $\pm$ 2.2	23.0 $\pm$ 1.1	16.6 $\pm$ 1.1
	Canonical	82.2 $\pm$ 0.3	59.4 $\pm$ 0.3	59.9 $\pm$ 0.6	5.6 $\pm$ 4.4	30.8 $\pm$ 1.6	29.8 $\pm$ 0.1	45.4 $\pm$ 0.4	20.0 $\pm$ 2.1	25.3 $\pm$ 0.8	17.6 $\pm$ 0.5