TRIBE: TRUST-REGION POLICY IMPROVEMENT FROM BATCH EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training language models from scalar rewards is commonly approached with policy-optimization methods developed for more general sequential decision problems. In sequence-level post-training, treating a complete response as a single action allows policy improvement to be written directly in terms of how the candidate policy reweights sampled responses. This motivates **TRIBE**, a two-stage method that first solves for the desired policy improvement in importance-ratio space and then updates the language model to realize it. The resulting formulation provides a unified approach to on-policy, off-policy, and fully offline training, including settings where the policies that generated the offline data are unavailable, and it allows additional policy constraints to be incorporated directly into the improvement step. Across GSM8K and MATH, TRIBE is competitive with strong on-policy baselines and is the strongest behavior-policy-independent method in our controlled offline experiments. On UltraFeedback, TRIBE is the only evaluated method to improve over rejection fine-tuning on both win rate and reward advantage. We also show that an expected-cost constraint can be added without changing the model-fitting stage, yielding competitive safety-constrained alignment.

1 INTRODUCTION

Post-training for large language models (LLMs) increasingly uses reinforcement learning (RL) algorithms to improve a policy from scalar rewards assigned to sampled responses (Ouyang et al., 2022; Shao et al., 2024). Many widely used algorithms (Shao et al., 2024; Yu et al., 2025) for this purpose build on policy-optimization ideas developed for general sequential decision problems, including conservative policy iteration, TRPO, and PPO (Kakade & Langford, 2002; Schulman et al., 2015; 2017). In a sequential decision problem, changing the policy changes not only the action distribution but also the distribution of states visited under the new policy. Policy improvement therefore depends on the state distribution induced by the candidate policy. As is standard in sequence-level language-model post-training with outcome-level rewards, we treat the complete response as a single action conditioned on the prompt. Under this formulation, the problem is a contextual bandit rather than a Markov decision process (MDP) (Puterman, 2014). Methods such as DPO (Rafailov et al., 2023) already exploit this sequence-level structure through preference-based objectives. We ask whether policy improvement from scalar rewards can instead be solved nonparametrically over the sampled responses, before projecting the resulting update onto the parameterized policy, rather than through optimization machinery inherited from general sequential decision problems.

In the contextual-bandit setting, the improvement of a candidate policy π over the behavior policy π_b can be expressed as an expectation over responses from π_b. Each response is reweighted according to how much more or less likely π makes that response relative to π_b, as measured by the importance ratio $\rho(x, y) = \pi(y \mid x)/\pi_b(y \mid x)$ (Kakade & Langford, 2002; Schulman et al., 2015). Unlike in an MDP, there is no additional change in the state distribution that needs to be accounted for. The resulting reweighted objective therefore directly describes the change in expected reward between the two policies. More importantly, the candidate policy affects this objective only through these response-level importance ratios. This suggests optimizing these ratios directly. Rather than updating the policy parameters and then observing the new response probabilities, we solve for the desired ratios of the sampled responses and update the policy to match them.

Because the variance of the importance-sampling estimate grows as the candidate policy moves away from the behavior policy (Metelli et al., 2018), we constrain the ratio-space update to a trust

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.

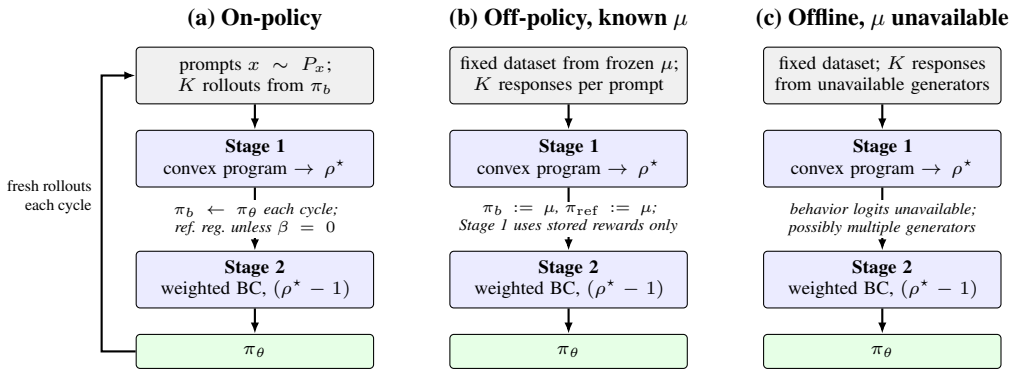

Figure 1: Overview of TRIBE across on-policy, off-policy (known μ), and fully offline settings. The two-stage design is shared; only the batch source and the choices of π_b and π_{ref} differ. In (b)–(c), taking $\pi_{\mathrm{ref}} = \pi_b$ lets Stage 1 use stored rewards without behavior log-probabilities. Panel (c) allows responses from several generators. Further constraints are added in Stage 1 only.

region around the behavior policy. This gives **TRIBE**, a two-stage method that separates policy improvement from policy parameterization by first optimizing in importance-ratio space and then updating the parameterized policy. In Stage 1, TRIBE optimizes directly over the importance ratios assigned to the sampled responses, subject to normalization constraints required for the ratios to define a valid policy and a trust region defined through an f-divergence that controls the size of the policy update. We show that the resulting empirical optimization is a finite-dimensional convex program over the likelihood ratios of the responses in the batch, so the desired policy improvement can be solved globally without optimizing over the language model parameters. In Stage 2, the resulting target ratios are used to update the parameterized language model policy, increasing the likelihood of responses assigned larger ratios and decreasing the likelihood of responses assigned smaller ratios. Thus, TRIBE solves for the policy improvement in ratio space, then projects this update onto the model parameters.

The two-stage formulation makes the same policy-improvement procedure useful across several practically different post-training settings, as illustrated in Figure 1. In standard on-policy training with verifiable rewards, reference regularization is often omitted (Yu et al., 2025; Hu et al., 2025; He et al., 2025; Mistral-AI et al., 2025). In this case, the Stage 1 solution for a broad class of f-divergences can be expressed directly in terms of the response advantages, with only the normalization and trust-region constraints remaining to be satisfied. For example, a χ^2 trust region produces an update proportional to the normalized group advantage used by GRPO (Shao et al., 2024), whereas reverse KL exponentially reweights responses according to their advantages. The same formulation also extends naturally to fixed offline datasets, where access to the generating policy may be limited or unavailable. In offline RL, policy updates are commonly constrained relative to the behavior distribution represented by the dataset (Levine et al., 2020; Wu et al., 2019; Kumar et al., 2019). TRIBE follows this view by defining the offline update relative to the behavior distribution represented by the sampled responses. Taking the behavior policy itself as the reference policy makes the corresponding reference term zero, so Stage 1 can be solved using only the stored responses and rewards, without requiring access to the behavior policy or its log-probabilities. This allows TRIBE to train from datasets for which the generating policy is unavailable, including datasets assembled from responses generated by multiple, potentially closed-source, policies. Finally, for applications with additional requirements beyond reward maximization, Stage 1 can incorporate them directly as constraints without changing Stage 2. For example, an expected-cost limit can be imposed as an additional constraint (Dai et al., 2023).

Contributions. First, we formulate single-turn language-model policy improvement directly in importance-ratio space and introduce TRIBE, a two-stage method that first solves for the target likelihood ratios of the sampled responses and then updates the parameterized policy toward realizing the resulting policy improvement. We show that Stage 1 is a finite-dimensional convex program with one variable per sampled response, and derive its solutions for commonly used trust regions. Second, we show that the same formulation supports on-policy, off-policy, and fully offline training. In the offline setting, Stage 1 uses only the stored responses and rewards and does not require access to

the behavior policy or its log-probabilities. Because Stage 1 is an explicit constrained optimization problem, additional requirements can also be incorporated as constraints without changing Stage 2. Third, we evaluate TRIBE across on-policy and offline settings. On GSM8K and MATH with Llama-3.2 models, TRIBE is competitive with the evaluated on-policy baselines and is the strongest evaluated behavior-independent method across the controlled offline settings. On fully offline UltraFeedback data, where the generating policies and their log-probabilities are unavailable, TRIBE is the only evaluated method that improves over rejection fine-tuning on both reported evaluation metrics. We also demonstrate safety-constrained alignment by incorporating an expected-cost limit directly into Stage 1, without modifying the Stage 2 policy-fitting procedure.

2 PRELIMINARIES

Setting. A policy π maps a prompt x to a distribution over complete responses y, where each response is a sequence of tokens. Prompts are drawn from a distribution P_x, and each response receives a scalar reward $r(x, y)$, supplied, for example, by a verifier (Yue et al., 2025) or a reward model trained from preferences (Ziegler et al., 2019; Ouyang et al., 2022). We omit $x \sim P_x$, unless noted otherwise. The objective is to maximize the expected reward $J(\pi) \coloneqq \mathbb{E}_{y\sim\pi(\cdot|x)}[\, r(x, y)\,]$. By treating the complete response y as the action conditioned on context x, this defines a contextual-bandit problem.

Data and notation. We are given a batch of rewarded responses. For each of N prompts $\{x_i\}_{i=1}^{N}$, we have K responses $\{y_{i,k}\}_{k=1}^{K}$ sampled from a *behavior* policy π_{b}, with reward $r_{i,k} \coloneqq r(x_i, y_{i,k})$. For each prompt, we use the mean reward of its sampled responses as a baseline and define the group-relative advantage estimate $A^{\pi_{\mathrm{b}}}(x_i, y_{i,k}) = r_{i,k} - \frac{1}{K}\sum_{j=1}^{K} r_{i,j}$ (Shao et al., 2024). We write π_θ for the parametric policy being optimized and π_{ref} for a frozen reference policy, typically the checkpoint from which training begins. Regularizing π_θ toward π_{ref} limits policy drift and can help mitigate reward hacking (Skalse et al., 2025; Gao et al., 2022; Rafailov et al., 2024). When training with verifiable rewards, recent work often omits this reference regularization because it can hurt performance (Yu et al., 2025; Hu et al., 2025; He et al., 2025; Mistral-AI et al., 2025). Omitting it also avoids maintaining and evaluating a separate reference model during training, reducing the associated memory and computation overhead. For a candidate policy π, we define its importance ratio relative to π_{b} as $\rho(x, y) \coloneqq \pi(y \mid x)/\pi_{\mathrm{b}}(y \mid x)$ (Precup et al., 2000). We omit the arguments (x, y) where they are clear from context.

Policy improvement. Define the population advantage of a response under π_{b} as $A^{\pi_{\mathrm{b}}}(x, y) \coloneqq r(x, y) - \mathbb{E}_{y'\sim\pi_{\mathrm{b}}(\cdot|x)}[r(x, y')]$. For any candidate policy π whose support is contained in that of π_{b}, the performance-difference lemma (Kakade & Langford, 2002) and importance sampling give

$$J(\pi) - J(\pi_{\mathrm{b}}) \;=\; \mathbb{E}_{y\sim\pi(\cdot|x)}[A^{\pi_{\mathrm{b}}}(x, y)] \;=\; \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho(x, y)\, A^{\pi_{\mathrm{b}}}(x, y)] \;\coloneqq\; L(\pi; \pi_{\mathrm{b}}). \qquad (1)$$

In the contextual-bandit setting, $L(\pi; \pi_{\mathrm{b}})$ is exactly the change in expected reward from π_{b} to π, rather than a surrogate approximation as in TRPO (Schulman et al., 2015).

3 TRIBE

TRIBE maximizes $L(\pi; \pi_{\mathrm{b}})$ from Eq. (1) in two stages. Stage 1 solves a convex program for the importance ratios $\rho^\star$ on the responses in the batch, under normalization and trust region constraints. Stage 2 uses these ratios to update the parameterized policy through the fitting objective described in Sec. 3.2.

3.1 STAGE 1: SOLVING FOR THE IMPORTANCE RATIOS

Policy improvement is normally posed over policy parameters. Eq. (1) lets us pose it over the importance ratios instead, since the improvement depends on π only through ρ. Not every nonnegative ρ is admissible, because $\rho\,\pi_{\mathrm{b}}$ has to be a policy.

Lemma 1 (Ratio coordinates)**.** *Fix π_{b}. $\pi \mapsto \pi/\pi_{\mathrm{b}}$ is a bijection from the policies supported inside π_{b} onto $\mathcal{P} \coloneqq \{\rho \geq 0 : \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho(x, y)] = 1\ \forall x\}$, with inverse $\rho \mapsto \rho\,\pi_{\mathrm{b}}$ (proof in Sec. D.1).*

We cannot maximize eq. (1) over $\mathcal{P}$ directly, because its population objective must be estimated from only K responses sampled from π_{b}. The variance of this importance-sampling estimate grows with the divergence between π and π_{b} (Metelli et al., 2018), so we restrict the step to a neighborhood of

π_{b}, measured by an f-divergence, $D_f(\pi \,\|\, \pi_{\mathrm{b}}) = \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)]$ for convex f with $f(1) = 0$, and bounded by δ. Appendix B provides an overview of f-divergences.

The budget δ keeps each update close to π_{b}, but π_{b} is updated as training proceeds, so a sequence of admissible updates can still carry the policy far from where it started. We therefore also keep π close to π_{ref}. Writing $\phi \coloneqq \pi_{\mathrm{b}}/\pi_{\mathrm{ref}}$ and $c \coloneqq \log\phi$, this second distance is again a function of ρ,

$$\mathbb{E}\big[D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))\big] \;=\; \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\log\rho] + \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\, c]\,. \tag{2}$$

and we add it to the objective with weight β. This is the reverse-KL penalty standard in RLHF (Ziegler et al., 2019; Ouyang et al., 2022), written in terms of ρ. Together these give the problem

$$\max_{\rho\,\geq\,0} \quad \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\, A^{\pi_{\mathrm{b}}} - \beta(\rho\log\rho + \rho\, c)] \tag{3a}$$

$$\text{s.t.} \quad \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)] \;\leq\; \delta, \tag{3b}$$

$$\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho(x,y)] \;=\; 1 \quad \forall x. \tag{3c}$$

The program is convex: the objective in Eq. 3a is concave, eq. (3b) is convex because f is, and eq. (3c) is affine. It is also finite-dimensional. We evaluate ρ only at the pairs in the batch. Under this empirical group measure, rather than under $x \sim P_x$, the decision variable is a vector in $\mathbb{R}^{NK}$, and eq. (3c) becomes $\frac{1}{K}\sum_k \rho_{i,k} = 1$ for each prompt. The improvement step therefore has one variable per stored response, independent of model size. We next characterize the solution of this finite-dimensional program.

Proposition 2 (Solution of the improvement problem)**.** *Let f be differentiable and strictly convex on $(0,\infty)$ with f' invertible on its range. Then eqs.* (3a) *to* (3c) *has a unique solution, and at $\beta = 0$*

$$\rho^\star(x,y) \;=\; (f')^{-1}\!\left(\frac{A^{\pi_{\mathrm{b}}}(x,y) - \nu(x)}{\lambda}\right) \;=\; \nabla f^*\!\left(\frac{A^{\pi_{\mathrm{b}}}(x,y) - \nu(x)}{\lambda}\right), \tag{4}$$

where $f^*(u) = \sup_{t>0}\{ut - f(t)\}$ is the convex conjugate of f, the multiplier $\nu(x)$ is fixed per prompt by eq. (3c)*, and the scalar $\lambda \geq 0$ is fixed globally by eq.* (3b)*. Proof in Sec. D.2.*

Equation (4) is stated at $\beta = 0$. With the anchor active, the same stationarity condition holds with $A^{\pi_{\mathrm{b}}}$ replaced by $A^{\pi_{\mathrm{b}}} - \beta c$ and an additional $\beta\log\rho$ term. For $f(\rho) = \rho\log\rho$, the two logarithmic terms combine and the solution is again explicit, yielding an exponential tilt with temperature $\beta + \lambda$ (eq. (6)). For other f, we solve the resulting convex program numerically.

We work out two instances, to show how the choice of f affects the solution. The first is the one we use in the offline data experiments.

Corollary 3 (GRPO's normalization)**.** *At $\beta = 0$ with a χ^2 trust region, $f(\rho) = (\rho-1)^2$, both multipliers are available in closed form, and for $\delta < 1/(K-1)$,*

$$\rho_k^\star - 1 \;=\; \sqrt{\delta}\;\frac{A_k^{\pi_{\mathrm{b}}} - \bar{A}^{\pi_{\mathrm{b}}}}{\sqrt{\mathrm{Var}_K(A^{\pi_{\mathrm{b}}})}} \tag{5}$$

(derived in Sec. D.3).

The right-hand side is the mean- and standard-deviation-normalized group advantage used by GRPO (Shao et al., 2024), scaled by $\sqrt{\delta}$. Thus, under a χ^2 trust region, the GRPO weighting arises as the exact solution of the constrained policy-improvement problem, while δ directly controls the magnitude of the update through the factor $\sqrt{\delta}$. We require $\delta < 1/(K-1)$ because proposition 4 then keeps the feasible interval strictly positive, so the constraint $\rho \geq 0$ cannot bind. A reverse-KL trust region instead yields an exponential tilt of the group advantages,

$$\rho^\star(x,y) \;=\; \frac{1}{Z(x)}\exp\!\left(\frac{A^{\pi_{\mathrm{b}}} - \beta\, c}{\beta+\lambda}\right), \qquad Z(x) \;=\; \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}\left[\exp\!\left(\frac{A^{\pi_{\mathrm{b}}} - \beta c}{\beta+\lambda}\right)\right], \tag{6}$$

and at $\beta = 0$ a forward-KL trust region gives a hyperbolic tilt. Corollary 7 derives all three.

Solving the program is cheap when the solution is explicit. The budget multiplier λ is a single scalar for the whole batch, found by a monotone one-dimensional root-find against eq. (3b), with $O(K)$ cost per evaluation. There is one normalization multiplier $\nu(x)$ per prompt, each determined by a monotone scalar equation on that prompt's group. For the two instances above, these have closed forms: $\nu = \bar{A}^{\pi_{\mathrm{b}}}$ under χ^2, while under reverse KL $\nu(x)$ is absorbed into $Z(x)$, making eq. (6) a softmax over the group. When $\beta = 0$, or when the anchor is the behavior policy itself, the anchor term vanishes and Stage 1 depends only on the sampled advantages, requiring no model evaluations. Sec. 3.3 exploits this in the offline setting (Levine et al., 2020).

3.2 STAGE 2: FITTING THE POLICY

Stage 1's solution $\pi^\star := \rho^\star \, \pi_\mathrm{b}$ is nonparametric, defined only on the sampled responses, so Stage 2 fits π_θ to it. We fit in the moment direction, minimizing $\mathbb{E}\big[D_\mathrm{KL}(\pi^\star \,\|\, \pi_\theta)\big]$. Up to a constant in θ, this objective is $-\mathbb{E}_{y\sim\pi_\mathrm{b}(\cdot|x)}[\rho^\star \log \pi_\theta]$: maximum likelihood over the sampled responses, weighted by $\rho^\star$. Since this is an expectation under π_b, it can be estimated directly from the batch. The reverse KL would instead require samples from π_θ and evaluating $\pi^\star$ on them, which is not possible because $\pi^\star$ is defined only on the sampled responses.

With these weights, every sampled response receives a nonnegative coefficient, so the objective can promote responses but provides no explicit signal to demote them. A prompt-dependent baseline $b(x)$ contributes $b(x)\,\mathbb{E}_{y\sim\pi_\mathrm{b}(\cdot|x)}[\nabla_\theta \log \pi_\theta]$ to the gradient, which is zero when $\pi_\theta = \pi_\mathrm{b}$ and remains small when the two policies are close. We therefore subtract $b(x) = \mathbb{E}_{y\sim\pi_\mathrm{b}(\cdot|x)}[\rho^\star] = 1$, yielding weights that are negative exactly where $\pi^\star$ assigns less probability than π_b. We found this baseline important for stable training. Stage 2 minimizes the average over the stored batch,

$$\mathcal{L}_\mathrm{proj}(\theta) = -\mathbb{E}\left[(\rho^\star(x,y) - 1)\log \pi_\theta(y \mid x)\right], \tag{7}$$

where the weight $\rho^\star(x,y) - 1$ specifies the relative change in probability prescribed by Stage 1. An autoregressive factorization $\log \pi_\theta(y \mid x) = \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t})$ means Eq. (7) gives longer responses a larger gradient even for the same ratio-space update.

In our implementation, we instead use a length-normalized sequence log-likelihood,

$$\mathcal{L}_\mathrm{proj}(\theta) = -\mathbb{E}\left[(\rho^\star(x,y) - 1)\frac{1}{|y|}\log \pi_\theta(y \mid x)\right]. \tag{8}$$

This preserves the direction of the update prescribed by $\rho^\star$ while preventing response length from independently scaling the magnitude of the Stage 2 update. This sample-level reduction follows the original GRPO convention, which first averages token losses within each response before averaging across responses (Shao et al., 2024).

3.3 THE ALGORITHM

Stage 2 never uses the policy that produced the batch, while Stage 1 requires it only through the anchor log-ratio c when $\beta > 0$. The same two stages therefore apply across different data-access settings, with the assignment of π_b and π_ref determining what information Stage 1 requires. In the offline case, taking $\pi_\mathrm{ref} = \pi_\mathrm{b}$ cancels the anchor term and leaves Stage 1 dependent only on the stored rewards, so TRIBE can be run on data whose generating distribution is unavailable. Algorithm 1 states the three settings together.

The three settings.

- *Offline.* The batch is a dataset of (x, y, r) triples assembled before training by some policy μ, and no further responses are collected. We set $\pi_\mathrm{b} := \mu$, since μ produced the responses. Offline RL commonly constrains updates to remain near the data-generating distribution, since the batch is informative only in that region (Levine et al., 2020). We therefore set $\pi_\mathrm{ref} := \mu$ as well, which makes $\phi \equiv 1$ and $c \equiv 0$, so Stage 1 uses the stored rewards alone. Neither stage evaluates μ, so TRIBE runs on data whose generating policy is unavailable, whether it was discarded, is proprietary, or was a mixture of several policies. The reduced program involves neither π_θ nor μ, so it can be solved either over the whole dataset or independently on each minibatch, with the latter enforcing the trust-region budget at the minibatch level.
- *Off-policy.* Responses are drawn during training, but from a policy μ that may not be π_θ: a checkpoint held fixed for several updates, a lagged actor, or another model. We again set $\pi_\mathrm{b} := \mu$ and $\pi_\mathrm{ref} := \mu$, so the update is constrained relative to the data-generating distribution and the anchor term vanishes. Stage 1 therefore uses only the sampled rewards, while the known behavior policy remains available to methods that explicitly require its log-probabilities. We interchangeably refer to this setting as offline training with a known behavior policy μ.
- *On-policy.* At the start of each cycle we set $\pi_\mathrm{b} \leftarrow \pi_\theta$ and hold it fixed while responses are drawn, as in PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). The anchor π_ref is never refreshed. After the first cycle π_b has moved and π_ref has not, so $\phi \neq 1$ and the anchor term of eq. (3a) stays active. Setting $\beta = 0$ removes it, as is common in RLVR (Sec. 2). Stage 1 is solved at each update, on the groups drawn at that step, so $\rho^\star$ is fresh at every update.

Stage 2 adjustments.

- *Clamping.* For responses with $\rho^\star < 1$, the Stage 2 objective is unbounded below as their probability approaches zero. In the offline setting, this can cause negatively weighted responses to continue exerting probability-decreasing pressure even after they have become very unlikely. A similar issue has been noted in prior off-policy REINFORCE work (Roux et al., 2025). We therefore clamp the token-level log-probabilities of negatively weighted responses at a small fixed floor. Let $C_{\min} > 0$ denote the corresponding token-probability floor. The offline Stage 2 objective is

$$\mathcal{L}_{\text{proj}}(\theta) = -\,\mathbb{E}\left[(\rho^\star - 1)\,\frac{1}{|y|}\sum_{t=1}^{|y|}\log\max\big(\pi_\theta(y_t \mid x, y_{<t}),\; C_{\min}\,\mathbb{I}[\rho^\star < 1]\big)\right]. \tag{9}$$

  Once an individual token probability in a negatively weighted response falls below $C_{\min}$, that token contributes no further gradient. We use $C_{\min} = 10^{-8}$ in our experiments. Clamping is applied only in the offline and off-policy settings; on-policy, fresh responses are sampled at each update.
- *Negative subsampling* (offline only). Offline minibatches can contain many more negatively than positively weighted responses, causing Stage 2 updates to be dominated by probability-decreasing terms, a form of dataset imbalance that can hurt offline learning (Hong et al., 2023). We therefore subsample responses with $\rho^\star < 1$ using retention probability f_{neg} while retaining all responses with $\rho^\star \geq 1$, reducing the aggregate contribution of negatively weighted examples. We use $f_{\text{neg}} \in [0.5, 0.6]$ in our experiments. Additional details are provided in Appendix C.

TRIBE is reference-free whenever the anchor is the behavior policy: throughout the offline setting, in the off-policy setting when $\pi_{\text{ref}} := \mu$, and in the on-policy setting at $\beta = 0$. TRIBE requires behavior-policy log-probabilities only when using a reference policy distinct from π_{b} with $\beta > 0$; otherwise, the anchor term vanishes. In the reference-free case, TRIBE avoids maintaining and evaluating a separate reference model.

3.4 Theoretical analysis

In this section we analyze the program of Sec. 3.1: the range it leaves for the Stage-2 weights, the geometry of the set it searches over, and the improvement the two stages give over π_{b}.

Feasible set and its geometry. Equation (3c) fixes $\sum_k \rho_k = K$ on each group, so an update can raise some ratios only by lowering others; it redistributes probability within a group rather than adding to it. The trust-region constraint in eq. (3b) controls the group as a whole but, by itself, need not bound any single ratio. For the Kullback–Leibler generators, one ρ_k can grow without limit while the others shrink to compensate; eq. (3c) prevents this by imposing $\rho_k \leq K$. Fig. 2 shows the two constraints for a group of two. Together they leave a bounded range of values for ρ_k. Its endpoints correspond to two extreme updates: a large ρ_k concentrates probability on one response, while $\rho_k = 0$ removes one from the support.

Proposition 4 (Range of the ratios)**.** *Fix a prompt with K responses. Normalization alone puts ρ_k in $[0, K]$, so the Stage-2 weights in eq. (7) lie in $[-1,\, K-1]$. Writing $g(\rho) := f(\rho) + (K-1)\, f\big(\frac{K-\rho}{K-1}\big)$, the budget tightens this to*

$$\{\rho \in [0, K] \;:\; g(\rho) \leq K\,\delta\}, \tag{10}$$

an interval containing 1 whose endpoints solve $g(\rho){=}K\delta$ if inside $[0, K]$ (proof in Sec. E).

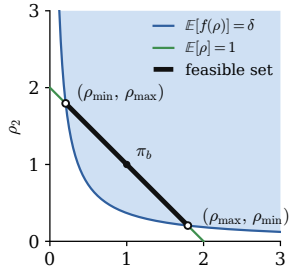

Figure 2: Forward KL, $K{=}2$, $\delta{=}0.5$. Shaded: the budget, which is unbounded. Bold: what normalization keeps (eq. (10)).

Normalization therefore gives every Stage 2 weight a lower bound of -1, independent of the rewards or group size. In contrast, reward-weighted regression (Peng et al., 2020) uses $\exp(A^{\pi_{\text{b}}}/\beta)$, which is unbounded above, while REINFORCE uses $A^{\pi_{\text{b}}}$, which is unbounded in both directions. In TRIBE, the feasible-set geometry additionally controls how large the positive weights can be through the choice of trust region and budget.

Under χ^2 the endpoints are $1 \pm \sqrt{\delta(K-1)}$. Sec. E works out the other two and gives the condition under which the lower endpoint reaches zero; Fig. 4 there draws all three.

Theorem 5 (Improvement). *Let $R_{\max} = \sup_{x,y} |r(x,y)|$ and let $\pi^\star$ be the Stage 1 solution. The projection error is $\epsilon_{\text{proj}} := \mathbb{E}\left[D_{\text{KL}}(\pi^\star(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x))\right]$. Then*

$$J(\pi_\theta) \geq J(\pi_b) + \mathbb{E}_{y\sim\pi_b(\cdot|x)}\left[\rho^\star(x,y) A^{\pi_b}(x,y)\right] - R_{\max}\sqrt{2\epsilon_{\text{proj}}}. \tag{11}$$

When $\phi \equiv 1$, the middle term is nonnegative, so an exact projection, $\epsilon_{\text{proj}} = 0$, gives $J(\pi_\theta) \geq J(\pi_b)$.

4 Results

We evaluate TRIBE through four research questions (RQs) spanning progressively more restrictive data-access settings and broader optimization requirements. **(RQ1)** Can TRIBE remain competitive with strong policy-optimization baselines in the standard on-policy setting? **(RQ2)** On fixed offline data from a known behavior policy μ, can TRIBE improve the learner without access to behavior-policy log-probabilities while remaining competitive with methods that use them? **(RQ3)** When the behavior policies that generated an offline dataset are unavailable, can TRIBE still improve the model using only stored responses and rewards? **(RQ4)** Can TRIBE incorporate an additional policy constraint through Stage 1 while preserving a competitive reward–constraint tradeoff?

Experimental Setup. We evaluate TRIBE using the Llama-3.2 (Grattafiori et al., 2024) family of models on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). We report Pass@1 on the GSM8K test split and Pass@4 on the more challenging MATH test split. For **RQ4**, we use PKU-SafeRLHF-10K (Dai et al., 2023) to evaluate safety-constrained alignment, reporting reward and cost win rates of the aligned models relative to the SFT initialization on a held-out evaluation set.

4.1 RQ1: On-Policy Alignment

We first evaluate TRIBE in the standard on-policy setting, where the current policy generates fresh responses throughout training. We compare against GRPO(Shao et al., 2024), DAPO(Yu et al., 2025), and RLOO (Ahmadian et al., 2024), as well as rejection fine-tuning (RFT), a broad family of self-training methods that generate candidate responses, retain successful or high-reward trajectories, and fine-tune the policy on the retained samples. Related instances of this paradigm include STaR (Zelikman et al., 2022), RAFT (Dong et al., 2023), ReST^{EM} (Singh et al., 2024), and rejection sampling fine-tuning for mathematical reasoning (Yuan et al., 2023). We call this family RFT (Appendix C).

As shown in Table 1, TRIBE matches the performance of strong on-policy policy-optimization baselines on both tasks, despite optimizing the policy improvement step directly in importance-ratio space rather than through a parameter-space surrogate. In contrast, rejection fine-tuning performs substantially worse: filtering to successful trajectories alone leaves considerable signal unused.

4.2 RQ2: Offline Alignment with a Known Behavior Policy

We next evaluate TRIBE in a controlled offline setting. We use Llama-3.2-3B-Instruct as the behavior policy μ to generate a fixed offline dataset, sampling 16 responses per prompt for GSM8K and 32 responses per prompt for MATH. All methods are then trained only on this frozen dataset. We align two learner models: Llama-3.2-3B-Instruct and Llama-3.2-1B-Instruct. The 3B setting measures offline policy improvement when the learner matches the model that generated the data, while the 1B setting represents a weaker learner trained entirely from responses generated by the stronger 3B model. Because the behavior policy is known in this setting, we can compare methods that require access to its log-probabilities against methods that do not. This setting additionally creates a mismatch between the policy being optimized and the policy that generated the training responses. This allows us to compare two classes of methods on exactly the same offline data. The first class consists of methods whose objectives require access to the behavior or a frozen reference policy, including DPO(Rafailov et al., 2023), GRPO(Shao et al., 2024), and TOPR(Roux et al., 2025). We additionally include truncated importance sampling (TIS) (Espeholt et al., 2018; Munos et al., 2016), a standard

Table 1: On-policy Llama-3.2-3B-Instruct (mean $\pm$ std over 3 seeds).

Method	**GSM8K** Pass@1	**MATH** Pass@4
TRIBE	$\mathbf{81.17 \pm 0.94}$	$\mathbf{64.23 \pm 0.50}$
DAPO	80.67 ± 1.09	62.91 ± 0.91
RLOO	80.52 ± 0.45	63.76 ± 0.42
GRPO	79.86 ± 0.68	64.09 ± 0.36
RFT	52.64 ± 4.74	22.20 ± 1.98

Table 2: Offline alignment with a known behavior policy μ. The offline dataset is generated once using Llama-3.2-3B-Instruct. Results are mean $\pm$ standard deviation over three seeds.

	GSM8K Pass@1		MATH Pass@4	
Method	**3B**	**1B**	**3B**	**1B**
Offline / behavior-independent				
TRIBE	$\mathbf{73.99 \pm 0.79}$	$\mathbf{46.15 \pm 0.89}$	$\mathbf{57.01 \pm 2.07}$	$\mathbf{40.17 \pm 0.64}$
SimPO	70.91 ± 1.18	42.51 ± 0.24	36.43 ± 3.02	40.08 ± 0.66
RFT	62.02 ± 0.33	44.93 ± 0.53	50.15 ± 0.46	33.69 ± 1.03
REINFORCE	36.27 ± 6.05	16.96 ± 4.28	32.91 ± 3.17	28.51 ± 0.79
REINFORCE + RLOO	0.48 ± 0.83	0.08 ± 0.13	26.93 ± 1.90	31.89 ± 5.57
Off-policy / behavior-dependent				
DPO	$\mathbf{79.18 \pm 0.56}$	41.67 ± 1.01	55.11 ± 0.92	33.73 ± 0.29
GRPO	69.02 ± 3.03	$\mathbf{47.38 \pm 0.98}$	$\mathbf{58.07 \pm 0.53}$	$\mathbf{40.11 \pm 1.03}$
TOPR	68.33 ± 0.76	45.24 ± 0.46	49.83 ± 1.44	32.33 ± 0.06
TIS	67.55 ± 1.15	45.79 ± 2.70	47.51 ± 0.94	35.87 ± 0.44

off-policy correction from reinforcement learning that applies REINFORCE with a truncated importance ratio. TIS is particularly relevant here because the asymmetric TOPR objective uses the same importance-sampling correction for negative generations. The second class contains methods that can train directly from the stored responses and rewards without evaluating μ, including TRIBE, SimPO (Meng et al., 2024), rejection fine-tuning (RFT), naive REINFORCE (Williams, 1992), and REINFORCE with an RLOO-style leave-one-out baseline (Ahmadian et al., 2024; Kool et al., 2019). In this offline setting, naive REINFORCE and RLOO yield biased estimates of the target-policy gradient because they apply policy-gradient updates to samples drawn from μ without correcting the behavior-policy mismatch through importance sampling. Overall, this controlled setting lets us compare off-policy methods that explicitly leverage the known behavior policy against offline methods that can train without access to behavior-policy log-probabilities.

As shown in Table 2, TRIBE is the strongest behavior-independent method across all four settings and remains highly competitive with methods that explicitly use behavior-policy log-probabilities. In three of the four settings, TRIBE is within $\sim$1 point of the strongest off-policy baseline, and slightly exceeds it on MATH with the 1B weak learner. The main exception is GSM8K with the 3B learner, where DPO obtains a larger advantage. These results suggest that, in most settings, TRIBE recovers much of the benefit of behavior-policy correction without the behavior-policy log-probabilities.

4.3 RQ3: Alignment from Fully Offline Data

Finally, we consider a fully offline setting in which the behavior policies that generated the training responses are unavailable. We train on UltraFeedback (Cui et al., 2024), which contains roughly 64k prompts, each paired with $K = 4$ completions generated by a diverse pool of open- and closed-source models and scored by GPT-4. We use only the prompts, stored completions, and GPT-4 overall scores as scalar rewards $r(x, y)$; no online rollouts or behavior-policy log-probabilities are available. All methods are initialized from Llama-3.2-3B-Base and trained for one epoch, avoiding the strong prior alignment already present in an instruction-tuned model. Consequently, methods whose objectives require access to the generating policy log-probabilities cannot be applied in this setting, and we compare TRIBE against behavior-independent offline baselines: RFT, RLOO, and SimPO.

We reserve 10% of UltraFeedback as a held-out test set and evaluate all trained models on this split using UltraRM (Cui et al., 2024) as a judge. We compare each model against two references: the RFT model and the highest-scoring completion among the four UltraFeedback candidates for each prompt in the test set. For each test prompt, we define the reward advantage over a reference as the difference between the UltraRM score of the evaluated model's generation and that of the reference response. We report the pairwise win rate and the median reward advantage.

As shown in Table 3, TRIBE is the only method to improve over RFT on both evaluation metrics, achieving a 57.5% win rate and a positive median reward advantage. It also performs best against the stronger Best-of-4 reference, attaining the highest win rate and the smallest median reward deficit. In contrast, RLOO and SimPO substantially underperform both RFT and the untrained base model.

Table 3: Fully offline alignment of Llama-3.2-3B-Base on UltraFeedback, evaluated using UltraRM as the judge. Win rate and median reward advantage are reported against RFT and the highest-scoring UltraFeedback completion (Best-of-4) on the held-out test set. Results use the 10% held-out split.

Method	**Win vs. Best-4**	**Win vs. RFT**	**Median Adv. vs. RFT**	**Median Adv. vs. Best-4**
Base (untrained)	0.0%	6.5%	−8.035	−13.633
RFT	9.0%	50.0%	0.000	−5.148
RLOO	0.0%	2.5%	−10.391	−15.836
SimPO	0.0%	0.0%	−12.498	−17.414
TRIBE	**14.5%**	**57.5%**	**+0.818**	**−4.531**

Table 4: Safety-constrained alignment on PKU-SafeRLHF-10K from the Llama-3.2-3B SFT initialization. Reward and cost win rates, together with mean reward and cost advantages, are measured relative to the SFT checkpoint. Lower cost is safer.

Method	Reward WR	Cost WR (safer)	Reward Adv.	Cost Adv.
TRIBE (`threshold` $= 0$)	**96.2%**	**93.7%**	**+2.646**	−3.739
Safe-RLHF (`threshold` $= 0$)	95.2%	**93.7%**	+2.434	−3.708

4.4 RQ4: Adding Constraints to TRIBE

We finally test whether TRIBE's explicit constrained optimization formulation can accommodate additional policy constraints without changing the overall training procedure. As a representative case, we consider safety-constrained alignment and augment the Stage 1 convex program with an expected-cost constraint, requiring the reweighted policy to satisfy a prescribed cost limit in addition to the existing trust-region constraint. TRIBE accommodates this extension easily, by adding an expected-cost constraint to the Stage 1 convex program. Like the reward objective, the expected cost, for a cost model $\psi(x, y)$, can be written directly in importance-ratio space, i.e., $\mathbb{E}_{y\sim\pi_b(\cdot|x)}\left[\rho(x, y)\psi(x, y)\right] \leq 0$. Thus the safety constraint adds only a linear constraint in Stage 1; Stage 2 is unchanged.

We compare against Safe-RLHF's PPO-Lagrangian baseline (Dai et al., 2023). Both methods are initialized from the same Llama-3.2-3B SFT checkpoint, obtained by supervised fine-tuning the Llama-3.2-3B-Instruct model, on the Alpaca dataset (Taori et al., 2023) to match the Safe-RLHF training pipeline, and are trained on PKU-SafeRLHF-10K (Dai et al., 2023) using trained reward and cost models. We evaluate on Safe-RLHF's official test-set consisting of $\sim$7k de-duplicated prompts, reporting reward and cost win rates together with mean reward and cost advantages relative to the SFT initialization. The goal of this experiment is to test whether TRIBE can absorb an additional constraint while retaining competitive reward–cost performance, rather than to introduce a new benchmark.

As shown in Table 4, TRIBE achieves a reward–cost tradeoff comparable to Safe-RLHF. The key result is that this behavior is obtained without modifying TRIBE's projection stage or introducing actor–critic or dual-optimization machinery: the safety requirement is handled entirely by adding a linear constraint to the existing Stage 1 convex program.

5 Conclusion

We introduced TRIBE, a framework for sequence-level LLM post-training that performs policy improvement directly in importance-ratio space and then projects the resulting update back into the parameterized model. This separation yields a finite-dimensional convex Stage 1 problem, simple solutions for important divergence choices, and a natural mechanism for adding further policy constraints without changing Stage 2. The same formulation applies across on-policy, off-policy, and fully offline training, and in the offline setting TRIBE can construct its update from stored rewards without requiring behavior-policy log-probabilities. Empirically, TRIBE is competitive with strong on-policy and off-policy baselines, remains effective when the policies that generated the training data are unavailable, and accommodates an expected-cost constraint while retaining competitive reward and cost performance. More broadly, our results suggest that for sequence-level post-training, it is useful to separate the problem of determining the desired policy improvement from the problem of fitting a parameterized language model to realize it.

AI USE STATEMENT

The authors used AI tools to assist with writing and polishing the manuscript. AI tools were also used to support the implementation of TRIBE and auxiliary experimental pipelines, including code development and debugging. In addition, AI tools were used to assist in checking and verifying mathematical derivations and proofs. The authors reviewed and validated all AI-assisted content, code, experimental results, and mathematical arguments, and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Sec. 3 states both stages of TRIBE in full, and algorithm 1 gives the pseudocode for all three training settings. Sec. C lists the hyperparameters for every experiment, including group size, learning rate, trust-region budget δ, divergence, anchor weight β and negative-retention fraction, together with the model, dataset and evaluation protocol used in each. All datasets and models are publicly available, and we give their exact identifiers in Sec. C. Results in Tabs. 1 and 2 are means and standard deviations over three seeds. The formal claims are proved in Secs. D and E, and each statement carries its own assumptions. Code will be made available upon publication.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL https://arxiv.org/abs/2402.14740.

Boris Belousov and Jan Peters. f-divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL https://arxiv.org/abs/2310.01377.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2310.12773.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures, 2018. URL https://arxiv.org/abs/1802.01561.

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

Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jujie He et al. Skywork open reasoner 1 technical report, 2025. URL https://arxiv.org/abs/2505.22312.

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. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Zhang-Wei Hong, Aviral Kumar, Sathwik Karnik, Abhishek Bhandwaldar, Akash Srivastava, Joni Pajarinen, Romain Laroche, Abhishek Gupta, and Pulkit Agrawal. Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets. *Advances in neural information processing systems*, 36:4985–5009, 2023.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

Sham M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14:1531–1538, 2002.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pp. 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Wouter Kool, Herke Van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction, 2019. URL https://arxiv.org/abs/1906.00949.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward, 2024. URL https://arxiv.org/abs/2405.14734.

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Mistral-AI et al. Magistral, 2025. URL https://arxiv.org/abs/2506.10910.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning, 2016. URL https://arxiv.org/abs/1606.02647.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Long Ouyang, Jeff 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 Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 2022.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 2000. URL https://scholarworks.umass.edu/cs_faculty_pubs/80/.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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. *arXiv preprint arXiv:2305.18290*, 2023. URL https://arxiv.org/abs/2305.18290.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms, 2024. URL https://arxiv.org/abs/2406.02900.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for llms, 2025. URL https://arxiv.org/abs/2503.14286.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. URL https://arxiv.org/abs/1502.05477.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300. Introduces Group Relative Policy Optimization (GRPO).

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking, 2025. URL https://arxiv.org/abs/2209.13085.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019. URL https://arxiv.org/abs/1911.11361.

Qiying Yu et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. URL https://arxiv.org/abs/2504.13837. RLVR biases the output distribution toward rewarded paths, improving sampling efficiency at small k while narrowing the pass@k reasoning boundary at large k.

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

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Related Work

TRIBE draws on trust-region policy optimization for its objective and constraints, on the weighted-regression family for its two-stage structure, on the post-training objectives of Sec. 4 for its empirical comparisons, and on offline RL for the behavior-regularization principle (Wu et al., 2019; Levine et al., 2020) underlying its reference-free setting.

Trust-region policy optimization. A long line of methods improves a policy by maximizing a per-iteration surrogate of J under a proximity constraint to the policy that produced the data. Conservative policy iteration (Kakade & Langford, 2002) established the template, bounding the improvement and taking a step short enough for the bound to hold. Natural policy gradient (Kakade, 2002), TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) instantiate it with different mechanisms: a Fisher metric on the parameter step, an explicit KL constraint, and a clip applied to ρ. POIS (Metelli et al., 2018) instead keeps the importance-sampling estimate as the optimization target and penalizes the Rényi divergence between the two policies, which bounds the estimator's variance. These methods optimize the parameterized policy directly; in the MDP setting, policy-improvement objectives must also account for the change in state distribution induced by the new policy. In TRIBE's contextual-bandit setting, eq. (1) is instead an exact identity, so there is no state-distribution surrogate error to control. Moreover, the trust region is imposed directly on ρ rather than enforced through a parameter-space update. The practical consequence is that clipping-based methods must evaluate ρ before applying the clip, whereas eq. (3b) constrains the ratios directly, enabling the offline instance of Sec. 3.3.

Policy improvement by weighted regression. A second family splits the update in two: solve a constrained program for a distribution over the sampled data, then fit a parametric policy to that distribution by weighted maximum likelihood. Relative entropy policy search (Peters et al., 2010) is the template, maximizing expected reward under a KL constraint and a normalization constraint and solving in closed form for $\pi^\star \propto \pi_\mathrm{b} \exp(A^{\pi_\mathrm{b}}/\eta)$. MPO (Abdolmaleki et al., 2018) carries the two steps to MDPs as an expectation-maximization procedure, and advantage-weighted regression (Peng et al., 2020; Nair et al., 2020) applies the fitting step to stored data. Belousov & Peters (2017) replaces the KL constraint with a general f-divergence, which is the same generalization eq. (3b) makes; we do not claim it. TRIBE's two stages are this family's two steps, and our program reduces to theirs: with $\pi_\mathrm{ref} = \pi_\mathrm{b}$ the anchor term drops, and at $\beta = 0$ a reverse-KL trust region leaves eq. (6), which is $\pi_\mathrm{b} \exp(A^{\pi_\mathrm{b}}/\lambda)$ renormalized over the group; this is their solution, with the budget multiplier λ in place of their temperature η. We differ in three respects. The improvement is exact rather than a surrogate, for the reason given above. Normalization is imposed on the K responses actually drawn rather than on a density, and on a finite group that constraint has two consequences: it bounds every weight, and it closes a trust region that the divergence alone leaves open (proposition 4). And the weights are signed, so the update can demote a response as well as promote one: $\exp(A^{\pi_\mathrm{b}}/\beta)$ is nonnegative, while $\rho^\star - 1$ lies in $[-1, K-1]$ and is negative exactly where $\pi^\star$ assigns less probability than π_b did.

Objectives for language-model post-training. The objectives used in practice differ in how they hold the update near the data and in what they must evaluate to do it. GRPO (Shao et al., 2024) standardizes the rewards within a group and uses them as weights, with a clip on ρ; DAPO (Yu et al., 2025) and RLOO (Ahmadian et al., 2024) modify the clipping and baseline choices. Equation (5) identifies the GRPO standardization as the exact solution of eqs. (3a) to (3c) under a χ^2 trust region, scaled by $\sqrt{\delta}$, giving that normalization a constrained policy-improvement interpretation. Methods that explicitly correct for behavior-policy mismatch instead control the ratio after it is formed: TOPR (Roux et al., 2025) tapers it and truncated importance sampling (Munos et al., 2016; Espeholt et al., 2018) truncates it, following the off-policy correction of Precup et al. (2000). These methods require behavior-policy log-probabilities, while objectives regularized to a distinct frozen reference

additionally require reference-policy log-probabilities (Ziegler et al., 2019; Ouyang et al., 2022). Other approaches avoid behavior-policy correction by changing the learning signal: DPO (Rafailov et al., 2023) and SimPO (Meng et al., 2024) optimize preferences between responses, while rejection fine-tuning (Zelikman et al., 2022; Dong et al., 2023; Singh et al., 2024; Yuan et al., 2023) trains only on selected high-reward responses and discards the rest. In its reference-free offline setting, TRIBE requires neither behavior-policy nor reference-policy log-probabilities while retaining the scalar rewards of all sampled responses.

Offline training without the generating policy. Offline RL is organized around controlling how far a learned policy may move from the distribution the data came from, whether by regularizing towards the behavior policy (Wu et al., 2019) or by penalizing values away from the data (Kumar et al., 2019); Levine et al. (2020) surveys the argument. TRIBE takes that prescription as its own: offline the trust region is measured against the policy that produced the responses, and the same policy serves as the anchor, so the anchor term drops out and Stage 1 reads the stored rewards alone (Sec. 3.3). This reaches a setting the methods above cannot. When the responses come from generators that can no longer be queried, or from several models at once, as in UltraFeedback (Cui et al., 2024), an off-policy correction has no π_{b} to correct against and a reference-anchored objective has no anchor to evaluate. What remains in that setting either trains on preference pairs or discards the unsuccessful responses, which is the comparison Sec. 4.3 reports.

B f-DIVERGENCES

Definition 6 (f-divergence)**.** Let p and q be distributions with $\operatorname{supp} p \subseteq \operatorname{supp} q$, and let f be convex with $f(1) = 0$. The f-divergence of p from q is

$$D_f(p \,\|\, q) \;=\; \mathbb{E}_{z\sim q}\left[f\left(\frac{p(z)}{q(z)}\right)\right]. \tag{12}$$

Nonnegativity. Jensen's inequality applied to eq. (12) gives $D_f(p \,\|\, q) \geq f(1) = 0$, with equality if and only if $p = q$ when f is strictly convex. The divergence is not symmetric in p and q, and is not a metric.

Ratio form. The argument of f is a density ratio. Setting $p = \pi$ and $q = \pi_{\mathrm{b}}$ expresses the divergence as a function of ρ alone,

$$D_f(\pi \,\|\, \pi_{\mathrm{b}}) \;=\; \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)]\,, \tag{13}$$

which is the trust region of eq. (3b).

Instances. Three generators are used in this paper. The final column gives $(f')^{-1}$, which appears in proposition 2; the corresponding solutions are derived in corollary 7.

	$f(t)$	$D_f(\pi \,\|\, \pi_{\mathrm{b}})$	$(f')^{-1}(u)$
Reverse KL	$t \log t$	$D_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{b}})$	$e^{\,u-1}$
Forward KL	$-\log t$	$D_{\mathrm{KL}}(\pi_{\mathrm{b}} \,\|\, \pi)$	$-1/u$
χ^2	$(t-1)^2$	$D_{\chi^2}(\pi \,\|\, \pi_{\mathrm{b}})$	$1 + u/2$

Domain of f'. Whether the constraint $\rho \geq 0$ can bind depends on where the generator is finite. If f is finite only on $(0, \infty)$ then f' has that domain, so $(f')^{-1}$ takes values in $(0, \infty)$ and the constraint is inactive; this is the case for both Kullback–Leibler generators. If instead f extends to $\mathbb{R}$, as $(t-1)^2$ does, then $(f')^{-1}$ takes values in $\mathbb{R}$ and the constraint has to be checked. Of the three instances above, only χ^2 is of this kind.

C IMPLEMENTATION DETAILS

C.1 TRAINING DETAILS

Unless otherwise stated, all experiments use Llama-3.2 models, bf16 precision, gradient checkpointing, one training epoch, and a constant learning-rate schedule. The MATH, UltraFeedback, and

Safe-RLHF experiments use DeepSpeed ZeRO-3 (Rasley et al., 2020). Experiments were run on two NVIDIA A100 GPUs, except for Safe-RLHF, which used four A100 GPUs due to the additional memory and compute overhead of the critic models. Table 5 summarizes the TRIBE-specific hyperparameters used across experimental settings.

On-policy experiments. For GSM8K[1] and MATH[2], all methods use Llama-3.2-3B-Instruct and generate $K = 8$ responses per prompt. We train with per-device batch size 8, gradient accumulation of 4, and learning rate 10^{-6}. The maximum completion length is 512 tokens for GSM8K and 1024 for MATH. TRIBE uses $\beta = 0$, a reverse-KL trust region with budget $\delta = 0.05$, no negative subsampling ($f_{\text{neg}} = 1$), and no log-probability floor. GRPO and RLOO also use $\beta = 0$. DAPO uses clipping parameters $\epsilon = 0.2$ and $\epsilon_{\text{high}} = 0.28$, with soft overlength cache lengths of 64 and 128 for GSM8K and MATH respectively. We evaluate all models on their respective unseen held-out test sets, using Pass@1 for GSM8K and Pass@4 for the more challenging MATH benchmark.

Controlled offline experiments. For the controlled offline setting, the behavior policy is Llama-3.2-3B-Instruct. We sample 16 responses per prompt for GSM8K and 32 for MATH, and train all methods for one epoch on the resulting frozen dataset. Unless otherwise noted, the learning rate is 5×10^{-7}. For TRIBE, we use a χ^2 trust region with budget $\delta = 0.5$ and negative-retention fraction $f_{\text{neg}} = 0.6$. The corresponding minibatch sizes are 8 for GSM8K and 16 for MATH.

DPO uses $\beta = 0.1$ with batch size 4 and gradient accumulation 16. SimPO uses $\beta = 2.5$, $\gamma = 1.4$, and a cosine learning-rate schedule, as recommended in their paper; for MATH we additionally use maximum sequence length 2048. TOPR, TIS, RLOO, and naive REINFORCE use the same 5×10^{-7} learning rate as TRIBE. Rejection fine-tuning uses the same offline data but retains only the successful responses. In this offline setting, we train both the Llama-3.2-3B-Instruct and Llama-3.2-1B-Instruct models exclusively on offline data generated by the Llama-3.2-3B-Instruct behavior policy. We evaluate all models on their respective unseen held-out test sets, using Pass@1 for GSM8K and Pass@4 for the more challenging MATH benchmark.

Fully offline UltraFeedback experiments. We initialize from Llama-3.2-3B-Base and train on UltraFeedback[3], which provides exactly four completions per prompt. We therefore use group size $K = 4$, per-device batch size 16, maximum completion length 1024, one training epoch, and DeepSpeed ZeRO-3. TRIBE uses a χ^2 trust region with $\delta = 0.5$ and learning rate 10^{-7}. We use $f_{\text{neg}} = 0.5$ and $\beta = 0$ variant, which uses the closed-form χ^2 solution. We evaluate all model generations using UltraRM[4] as the judge on an unseen held-out test set comprising 10% of the full UltraFeedback dataset.

RFT uses a learning rate of 2×10^{-5} with cosine decay and 0.03 warmup. RLOO uses 10^{-7} with a constant schedule. SimPO uses learning rate 5×10^{-7}, $\beta = 2.0$, $\gamma = 1.0$, as recommended in their paper, and a preference dataset generated from the offline UltraFeedback dataset.

Safety-constrained alignment. Both methods start from the same Llama-3.2-3B model supervised fine-tuned on the Alpaca dataset[5]. The reward and cost models are trained from the same SFT initialization, and training uses PKU-SafeRLHF-10K[6]. TRIBE uses $K = 8$ generations per prompt, batch size 8, gradient accumulation 4, maximum completion length 512, learning rate 10^{-6}, $\beta = 0.1$, reverse-KL trust-region budget $\delta = 0.05$, and cost limit 0.

The Safe-RLHF PPO-Lagrangian baseline uses actor and critic learning rates 5×10^{-6}, KL coefficient 0.1, policy clipping ratio 0.1, initial Lagrange multiplier 1.0, multiplier learning rate 0.01, and cost threshold 0. We otherwise retain the settings from the corresponding Safe-RLHF training configuration. We evaluate both models on Safe-RLHF's official test-set[7] consisting of ∼7k de-duplicated prompts

C.2 Negative Subsampling

As discussed in Sec. 3.3, offline Stage 2 updates can occasionally contain substantially more negatively than positively weighted responses, which can cause the update to be dominated by probability-

[1] `openai/gsm8k` [2] `DigitalLearningGmbH/MATH-lighteval`
[3] `openbmb/UltraFeedback` [4] `openbmb/UltraRM-13b` [5] `tatsu-lab/alpaca`
[6] `PKU-Alignment/PKU-SafeRLHF-10K` [7] `PKU-Alignment/PKU-SafeRLHF`

Table 5: TRIBE hyperparameters across experimental settings.

Setting	K	LR	δ	Divergence	β	f_{neg}
GSM8K on-policy	8	1×10^{-6}	0.05	Reverse KL	0	1.0
MATH on-policy	8	1×10^{-6}	0.05	Reverse KL	0	1.0
GSM8K offline	16	5×10^{-7}	0.5	χ^2	0	0.6
MATH offline	32	5×10^{-7}	0.5	χ^2	0	0.6
UltraFeedback fully offline	4	1×10^{-7}	0.5	χ^2	0	0.5
Safe TRIBE	8	1×10^{-6}	0.05	Reverse KL	0.1	1.0

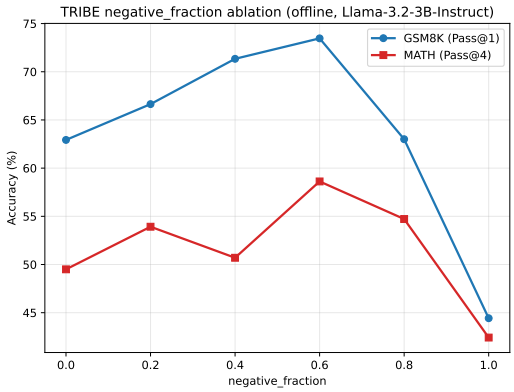

Figure 3: Effect of the negative-retention fraction f_{neg} on offline TRIBE performance.

decreasing terms. We therefore subsample responses with $\rho^\star < 1$ using the negative-retention fraction f_{neg}, while retaining all responses with $\rho^\star \geq 1$. Retained negative responses are not reweighted, so subsampling reduces their aggregate contribution while preserving the per-example gradient magnitude of those retained. In our experiments, $f_{\text{neg}} \in [0.5, 0.6]$ yields an average positive-to-negative ratio of approximately 2:1 per minibatch. No negative subsampling is used on-policy ($f_{\text{neg}} = 1$). Figure 3 studies the sensitivity of offline performance to this choice.

Figure 3 shows that retaining an intermediate fraction of negatively weighted responses performs substantially better than either extreme. On both GSM8K and MATH, using all negative responses ($f_{\text{neg}} = 1$) gives the weakest performance, while completely removing them ($f_{\text{neg}} = 0$) is also suboptimal. Performance is strongest for intermediate retention rates, with $f_{\text{neg}} = 0.6$ giving the best result on both tasks. This suggests that negative responses provide useful training signal, but allowing them to dominate the offline Stage 2 update can be harmful.

C.3 TRIBE ALGORITHM

Algorithm 1 TRIBE: Trust-Region Policy Improvement from Batch Experience.
The two stages are the same in every setting; the settings differ in how the batch is obtained and in what π_{b} and π_{ref} are taken to be.

Require: reward r, group size K, budget δ, divergence f, anchor weight β, projection steps M, learning rate η

initialise θ; **offline:** a dataset $\mathcal{D}$ of (x, y, r) triples from a frozen μ, and $\pi_{\mathrm{b}} \leftarrow \mu$, $\pi_{\mathrm{ref}} \leftarrow \mu$
for each iteration **do**
  if on-policy **then**
    $\pi_{\mathrm{b}} \leftarrow \pi_\theta$; draw K responses per prompt from π_{b} and score them with r
  else
    take the next minibatch of groups from $\mathcal{D}$ ▷ offline: π_{b} is neither sampled nor evaluated
  end if
  $A^{\pi_{\mathrm{b}}}(x, y_k) \leftarrow r(x, y_k) - \frac{1}{K}\sum_j r(x, y_j)$
  Stage 1 — solve
  $c \leftarrow \log(\pi_{\mathrm{b}}/\pi_{\mathrm{ref}})$ ▷ $\equiv 0$ when $\pi_{\mathrm{ref}} = \pi_{\mathrm{b}}$
  solve eqs. (3a) to (3c) for $\rho^\star$, giving $\nu(x)$ per prompt and λ for the batch ▷ proposition 2
  Stage 2 — project
  for $m = 1, \dots, M$ **do**
    $\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}(\theta; \rho^\star)$ ▷ eq. (8); offline, eq. (9)
  end for
end for

D PROOFS

Throughout, expectations are taken with respect to the measure $P_x(x)\,\pi_{\mathrm{b}}(y \mid x)$, which is positive on $\operatorname{supp}\pi_{\mathrm{b}}$, and all statements are understood on that support. Under the empirical group measure a prompt x_i carries K responses, the normalization constraint eq. (3c) reads $\frac{1}{K}\sum_k \rho_{i,k} = 1$, and we drop the prompt index where it is not needed.

D.1 PROOF OF LEMMA 1 (RATIO COORDINATES)

Proof. Let π be a policy with $\operatorname{supp}\pi(\cdot \mid x) \subseteq \operatorname{supp}\pi_{\mathrm{b}}(\cdot \mid x)$ for every x. Then $\rho = \pi/\pi_{\mathrm{b}}$ is well defined and nonnegative on $\operatorname{supp}\pi_{\mathrm{b}}$, and

$$\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho(x,y)] \;=\; \sum_{y\in\operatorname{supp}\pi_{\mathrm{b}}(\cdot|x)} \pi_{\mathrm{b}}(y \mid x)\,\frac{\pi(y \mid x)}{\pi_{\mathrm{b}}(y \mid x)} \;=\; \sum_{y\in\operatorname{supp}\pi_{\mathrm{b}}(\cdot|x)} \pi(y \mid x) \;=\; 1,$$

the last equality because π places no mass outside $\operatorname{supp}\pi_{\mathrm{b}}$. Hence $\rho \in \mathcal{P}$.

Conversely let $\rho \in \mathcal{P}$ and put $\pi \coloneqq \rho\,\pi_{\mathrm{b}}$. Then $\pi \geq 0$, and for every x, $\sum_y \pi(y \mid x) = \sum_y \rho(x,y)\,\pi_{\mathrm{b}}(y \mid x) = \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho(x,y)] = 1$, so π is a policy; it satisfies $\operatorname{supp}\pi(\cdot \mid x) \subseteq \operatorname{supp}\pi_{\mathrm{b}}(\cdot \mid x)$ by construction. The two maps are mutually inverse on $\operatorname{supp}\pi_{\mathrm{b}}$, since $(\rho\,\pi_{\mathrm{b}})/\pi_{\mathrm{b}} = \rho$ and $(\pi/\pi_{\mathrm{b}})\,\pi_{\mathrm{b}} = \pi$ there. □

Both maps are determined only on $\operatorname{supp}\pi_{\mathrm{b}}$, and the objective and constraints of eqs. (3a) to (3c) evaluate ρ nowhere else.

D.2 PROOF OF PROPOSITION 2 (SOLUTION OF THE IMPROVEMENT STEP)

Proof. The program is convex and Slater's condition holds. In eq. (3a) the term $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\,A^{\pi_{\mathrm{b}}}]$ is linear in ρ and $-\beta\,\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\log\rho + \rho\,c]$ is concave, since $\rho\log\rho$ is convex and $\rho\,c$ is linear; the objective is therefore concave. The feasible set is convex: $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)] \leq \delta$ is a sublevel set of a convex function because f is convex, eq. (3c) is affine, and $\rho \geq 0$ is a convex constraint. The point $\rho \equiv 1$ satisfies every equality constraint and gives $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(1)] = 0 < \delta$, so it is strictly

feasible for the inequality, and Slater's condition holds. The KKT conditions are therefore necessary and sufficient.

Stationarity. Introduce $\lambda \geq 0$ for eq. (3b) and $\nu(x)$ for the equality constraint at x, and write the Lagrangian

$$\begin{aligned}\mathcal{L}(\rho, \lambda, \nu) \;=\; & \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\, A^{\pi_{\mathrm{b}}}] \;-\; \beta\, \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho \log \rho + \rho\, c] \\ & -\; \lambda\big(\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)] - \delta\big) \;-\; \mathbb{E}\big[\nu(x)\big(\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho] - 1\big)\big].\end{aligned}$$

Every term is an expectation against the positive measure $P_x(x)\pi_{\mathrm{b}}(y \mid x)$, so stationarity may be taken pointwise in $\rho(x, y)$, giving

$$A^{\pi_{\mathrm{b}}}(x, y) \;-\; \beta\big(\log \rho(x, y) + 1\big) \;-\; \beta\, c(x, y) \;-\; \lambda\, f'(\rho(x, y)) \;-\; \nu(x) \;=\; 0. \tag{14}$$

Setting $\beta = 0$ and solving for ρ gives $f'(\rho) = (A^{\pi_{\mathrm{b}}} - \nu(x))/\lambda$, and since f is strictly convex and differentiable, f' is strictly increasing and therefore invertible on its range, yielding eq. (4). The second equality there is the Fenchel identity: for a closed, proper, strictly convex f the conjugate f^* satisfies $\nabla f^* = (f')^{-1}$ on the range of f'.

Uniqueness. If $\beta > 0$ the objective is strictly concave, because $-\beta\, \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho \log \rho]$ is, so the maximiser is unique. If $\beta = 0$ the objective is linear, and we argue from the feasible set. It is closed, and bounded by proposition 4, hence compact; and $\{\rho : \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)] \leq \delta\}$ is strictly convex because f is strictly convex. A linear functional attains its maximum over a compact strictly convex set at a unique point unless it is constant on that set. By eq. (3c) the functional $\rho \mapsto \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho\, A^{\pi_{\mathrm{b}}}]$ is constant on the feasible set exactly when $A^{\pi_{\mathrm{b}}}$ is constant within each prompt's group, that is, when every response for that prompt receives the same reward. Otherwise the solution is unique.

Determination of λ. Complementary slackness gives $\lambda\big(\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho^\star)] - \delta\big) = 0$, so either the constraint is slack and $\lambda = 0$, or it is active and λ solves $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho^\star(\lambda))] = \delta$. That equation has at most one solution because $\lambda \mapsto \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho^\star(\lambda))]$ is nonincreasing. To see this, write $g(\rho) = \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho A^{\pi_{\mathrm{b}}}]$ and $h(\rho) = \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)]$, and let ρ_1, ρ_2 maximise $g - \lambda_1 h$ and $g - \lambda_2 h$ respectively over the set $\mathcal{P}$, with $\lambda_1 < \lambda_2$. Optimality of each gives

$$g(\rho_1) - \lambda_1 h(\rho_1) \;\geq\; g(\rho_2) - \lambda_1 h(\rho_2), \qquad g(\rho_2) - \lambda_2 h(\rho_2) \;\geq\; g(\rho_1) - \lambda_2 h(\rho_1),$$

and adding the two and cancelling g leaves $(\lambda_2 - \lambda_1)\big(h(\rho_1) - h(\rho_2)\big) \geq 0$, so $h(\rho_1) \geq h(\rho_2)$. Monotonicity licenses the bisection used in Sec. 3.1.

The reverse-KL closed form, quoted after the proposition in Sec. 3.1. Take $f(\rho) = \rho \log \rho$, so $f'(\rho) = \log \rho + 1$, and substitute into eq. (14):

$$A^{\pi_{\mathrm{b}}} - \beta \log \rho - \beta - \beta c - \lambda \log \rho - \lambda - \nu(x) \;=\; 0 \quad\Longrightarrow\quad \log \rho \;=\; \frac{A^{\pi_{\mathrm{b}}} - \beta c - \nu(x)}{\beta + \lambda} - 1.$$

Exponentiating separates a factor depending on (x, y) from one depending on x alone,

$$\rho^\star(x, y) \;=\; \exp\!\left(\frac{A^{\pi_{\mathrm{b}}} - \beta c}{\beta + \lambda}\right) \cdot \underbrace{\exp\!\left(-\frac{\nu(x)}{\beta + \lambda} - 1\right)}_{:= 1/Z(x)},$$

and imposing $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho^\star] = 1$ identifies $Z(x) = \mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}\big[\exp\big((A^{\pi_{\mathrm{b}}} - \beta c)/(\beta + \lambda)\big)\big]$, which is eq. (6). Note that β and λ enter only through their sum, because the entropic term of the objective has the same functional form as a reverse-KL trust region, so the two multipliers combine additively at stationarity. □

D.3 Named instances of proposition 2

Sec. 3.1 states the χ^2 instance of eq. (4) and names the other two. We state all three and derive them here.

Corollary 7 (Named instances)**.** *Take $\beta = 0$ and a group of K responses with advantages $A_1^{\pi_{\mathrm{b}}}, \ldots, A_K^{\pi_{\mathrm{b}}}$, sample mean $\bar{A}^{\pi_{\mathrm{b}}}$ and sample variance $\mathrm{Var}_K(A^{\pi_{\mathrm{b}}})$, and suppose the nonnegativity constraint is inactive. Under the conditions of proposition 2:*

1. (χ^2, $f(\rho) = (\rho-1)^2$.) $\rho_k^\star - 1 = \sqrt{\delta}\,(A_k^{\pi_\mathrm{b}} - \bar{A^{\pi_\mathrm{b}}})/\sqrt{\mathrm{Var}_K(A^{\pi_\mathrm{b}})}$, *with both multipliers in closed form.*
2. (Reverse KL, $f(\rho) = \rho\log\rho$.) $\rho_k^\star = K\ \mathrm{softmax}_k(A_k^{\pi_\mathrm{b}}/\lambda)$, *an exponential tilt at temperature* λ.
3. (Forward KL, $f(\rho) = -\log\rho$.) $\rho_k^\star = \lambda/(\nu - A_k^{\pi_\mathrm{b}})$ *with* $\nu > \max_k A_k^{\pi_\mathrm{b}}$.

Proof. Throughout take $\beta = 0$ and work within one group, writing $A_k^{\pi_\mathrm{b}}$ for the advantage of its k-th response, $\bar{A^{\pi_\mathrm{b}}}$ for the group mean and $\mathrm{Var}_K(A^{\pi_\mathrm{b}})$ for the group variance.

(1) χ^2 trust region. With $f(\rho) = (\rho-1)^2$ we have $f'(\rho) = 2(\rho-1)$ and $(f')^{-1}(u) = 1 + u/2$, so eq. (4) gives $\rho_k^\star = 1 + (A_k^{\pi_\mathrm{b}} - \nu)/(2\lambda)$. Averaging over the group and imposing eq. (3c) yields $\frac{1}{K}\sum_k (A_k^{\pi_\mathrm{b}} - \nu)/(2\lambda) = 0$, hence $\nu = \bar{A^{\pi_\mathrm{b}}}$ and $\rho_k^\star - 1 = (A_k^{\pi_\mathrm{b}} - \bar{A^{\pi_\mathrm{b}}})/(2\lambda)$. Since the objective is linear in ρ the trust region is active, so $\frac{1}{K}\sum_k(\rho_k^\star - 1)^2 = \delta$; substituting the previous display gives $\mathrm{Var}_K(A^{\pi_\mathrm{b}})/(4\lambda^2) = \delta$, that is $2\lambda = \sqrt{\mathrm{Var}_K(A^{\pi_\mathrm{b}})/\delta}$. Eliminating λ gives eq. (5). Both multipliers are thus available in closed form and no root-find is needed.

This derivation assumed the nonnegativity constraint inactive. It is inactive precisely when $1 + \sqrt{\delta}\,\bar{A^{\pi_\mathrm{b}}}_k \geq 0$ for every k, that is when

$$\delta \;\leq\; \frac{\mathrm{Var}_K(A^{\pi_\mathrm{b}})}{\big(\bar{A^{\pi_\mathrm{b}}} - \min_k A_k^{\pi_\mathrm{b}}\big)^2}. \tag{15}$$

Above that threshold the responses with the most negative standardized advantage are driven to $\rho_k^\star = 0$, and eq. (5) continues to describe the remainder with ν and λ recomputed on the surviving set.

(2) Reverse-KL trust region. With $f(\rho) = \rho\log\rho$ we have $(f')^{-1}(u) = e^{\,u-1}$ and eq. (4) gives $\rho_k^\star = \exp\big((A_k^{\pi_\mathrm{b}} - \nu)/\lambda - 1\big)$. The factor $e^{-\nu/\lambda - 1}$ does not depend on k, so imposing $\sum_k \rho_k^\star = K$ determines it and

$$\rho_k^\star \;=\; K\,\frac{e^{A_k^{\pi_\mathrm{b}}/\lambda}}{\sum_j e^{A_j^{\pi_\mathrm{b}}/\lambda}} \;=\; K\ \mathrm{softmax}_k\big(A_k^{\pi_\mathrm{b}}/\lambda\big),$$

an exponential tilt of the group advantages at temperature λ. This is eq. (6) with $\beta = 0$, and nonnegativity is automatic.

(3) Forward-KL trust region. With $f(\rho) = -\log\rho$ we have $f'(\rho) = -1/\rho$ and $(f')^{-1}(u) = -1/u$, so $\rho_k^\star = \lambda/(\nu - A_k^{\pi_\mathrm{b}})$. Positivity requires $\nu > \max_k A_k^{\pi_\mathrm{b}}$, and ν is then the unique such value solving $\frac{1}{K}\sum_k \lambda/(\nu - A_k^{\pi_\mathrm{b}}) = 1$, the left-hand side being strictly decreasing in ν on $(\max_k A_k^{\pi_\mathrm{b}}, \infty)$ with limits $+\infty$ and 0 at the two ends. □

Each f above is the generator of the divergence it is named for, in the sense that $\mathbb{E}_{y\sim\pi_\mathrm{b}(\cdot|x)}[f(\rho)]$ equals $D_{\chi^2}(\pi\,\|\,\pi_\mathrm{b})$, $D_{\mathrm{KL}}(\pi\,\|\,\pi_\mathrm{b})$ and $D_{\mathrm{KL}}(\pi_\mathrm{b}\,\|\,\pi)$ respectively.

D.4 Proof of theorem 5 (improvement)

Proof. Write the bound as the sum of what Stage 1 achieves and what Stage 2 loses.

For the first part, A^{π_b} is centered under π_b, so eq. (1) applied to $\pi^\star = \rho^\star\pi_\mathrm{b}$ gives

$$\mathbb{E}_{y\sim\pi_\mathrm{b}(\cdot|x)}[\rho^\star\,A^{\pi_\mathrm{b}}] \;=\; \mathbb{E}_{y\sim\pi^\star(\cdot|x)}[A^{\pi_\mathrm{b}}(x,y)] \;=\; J(\pi^\star) - J(\pi_\mathrm{b}).$$

This is an identity, not an approximation, because eq. (1) is exact in the bandit setting, as Sec. 2 notes.

For the second part, $|J(\pi^\star) - J(\pi_\theta)|$ is a difference of expectations of the same bounded function under two policies sharing the prompt distribution, so Hölder gives

$$\begin{aligned}|J(\pi^\star) - J(\pi_\theta)| \;&\leq\; R_{\max}\,\mathbb{E}\big\|\pi^\star(\cdot\mid x) - \pi_\theta(\cdot\mid x)\big\|_1 \\ &\leq\; R_{\max}\,\mathbb{E}\sqrt{2\,D_{\mathrm{KL}}(\pi^\star\,\|\,\pi_\theta)} \;\leq\; R_{\max}\sqrt{2\,\varepsilon_{\mathrm{proj}}},\end{aligned}$$

by Pinsker's inequality and then Jensen's, the latter applied to the concave $\sqrt{\cdot}$. Adding the two displays gives eq. (11).

For the sign of the middle term when $\phi \equiv 1$: the constant ratio $\rho \equiv 1$ is feasible, since it satisfies eq. (3c) and gives $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[f(\rho)] = f(1) = 0 \le \delta$. With $\phi \equiv 1$ we have $c \equiv 0$, so the objective eq. (3a) at $\rho \equiv 1$ is $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[A^{\pi_{\mathrm{b}}}] = 0$, again because $A^{\pi_{\mathrm{b}}}$ is centered. Optimality of $\rho^\star$ therefore gives $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho^\star A^{\pi_{\mathrm{b}}}] - \beta\,\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho^\star \log \rho^\star] \ge 0$, and $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho^\star \log \rho^\star] = D_{\mathrm{KL}}(\pi^\star \,\|\, \pi_{\mathrm{b}}) \ge 0$, so $\mathbb{E}_{y\sim\pi_{\mathrm{b}}(\cdot|x)}[\rho^\star A^{\pi_{\mathrm{b}}}] \ge 0$. Taking $\varepsilon_{\mathrm{proj}} = 0$ in eq. (11) then leaves $J(\pi_\theta) \ge J(\pi_{\mathrm{b}})$. □

The bound holds for any θ Stage 2 returns, optimal or not; $\varepsilon_{\mathrm{proj}}$ is whatever objective value the fit reached. It does not cover the gap between the batch and the population: $\rho^\star$ solves the empirical program, and eq. (11) is stated for the population quantities.

E Geometry of the ratio-space trust region

This appendix proves proposition 4 and gives the three instances.

E.1 Proof of proposition 4 (extremes of the trust region)

Proof. Fix a prompt and its group of K responses. Under the empirical group measure the two constraints eqs. (3b) and (3c) read

$$\sum_{j=1}^{K} \rho_j = K, \qquad \rho_j \ge 0 \ \ \forall j, \qquad \frac{1}{K}\sum_{j=1}^{K} f(\rho_j) \le \delta.$$

The first claim uses normalization alone. Nonnegativity gives $\rho_j \ge 0$ for every j and the group sums to K, so $\rho_k = K - \sum_{j\neq k} \rho_j \le K$ because the omitted terms are nonnegative; hence $\rho_k \in [0, K]$ and the Stage-2 weight obeys $\rho_k - 1 \in [-1, K-1]$.

For the rest, by symmetry it is enough to characterise the values available to ρ_1. Fix $t \in [0, K]$ and ask when $\rho_1 = t$ can be completed to a feasible point. The remaining coordinates must be nonnegative and sum to $K - t$, so t is attainable exactly when

$$f(t) \ + \ \min\Big\{ \textstyle\sum_{j\ge 2} f(\rho_j) \ : \ \rho_j \ge 0,\ \sum_{j\ge 2} \rho_j = K - t \Big\} \ \le \ K\,\delta.$$

Since f is convex, Jensen gives $\sum_{j\ge 2} f(\rho_j) \ge (K-1)\, f\big(\frac{K-t}{K-1}\big)$ for any such $\rho_{2:K}$, with equality at the uniform completion $\rho_2 = \cdots = \rho_K = \frac{K-t}{K-1}$, which is itself feasible because $t \le K$ makes it nonnegative. The minimum is therefore attained and equals that value, so t is attainable if and only if $g(t) \le K\delta$ with $g(\rho) = f(\rho) + (K-1) f\big(\frac{K-\rho}{K-1}\big)$ as in the statement.

It remains to see that $\{g \le K\delta\}$ is an interval around 1. The first term of g is convex and the second is f composed with an affine map, hence convex, so g is convex on $[0, K]$; and $g(1) = f(1) + (K-1)f(1) = 0$ because $f(1) = 0$. A sublevel set of a convex function is an interval, and it contains 1 whenever $\delta \ge 0$. Where an endpoint falls in the interior of $[0, K]$ it satisfies $g = K\delta$ with equality, by continuity of g. □

The argument uses no property of f beyond convexity and $f(1) = 0$, and in particular does not need f differentiable. Tightness comes from the uniform completion: pushing one ratio to an extreme costs least in divergence when the rest of the group absorbs the change evenly.

E.2 The three instances

(χ^2.) With $f(\rho) = (\rho - 1)^2$, the inner argument shifts to $\frac{K-\rho}{K-1} - 1 = \frac{1-\rho}{K-1}$, so

$$g(\rho) \ = \ (\rho-1)^2 + (K-1)\frac{(1-\rho)^2}{(K-1)^2} \ = \ (\rho-1)^2\Big(1 + \frac{1}{K-1}\Big) \ = \ \frac{K}{K-1}\,(\rho-1)^2.$$

Setting $g(\rho) = K\delta$ gives $(\rho-1)^2 = \delta(K-1)$, hence the closed form

$$\rho \ \in \ \big[\,1 - \sqrt{\delta(K-1)},\ \ 1 + \sqrt{\delta(K-1)}\,\big]. \tag{16}$$

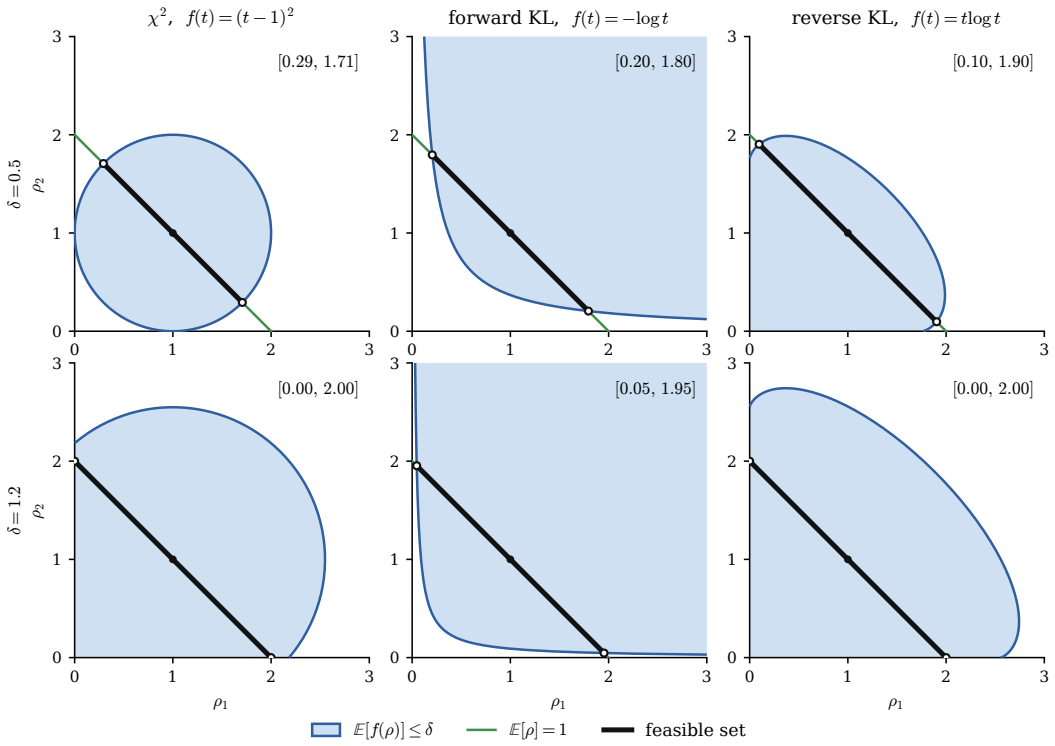

Figure 4: The trust region in ratio space for the three generators, on a group of $K = 2$ responses. Shaded is the divergence budget $\{\mathbb{E}_{y\sim\pi_b(\cdot|x)}[f(\rho)] \le \delta\}$, the straight line is eq. (3c), and the bold segment is what the two leave, with the endpoints of eq. (10) marked and printed. The three generators differ in shape: the χ^2 budget is a disc, the forward-KL budget is open towards the top right, and the reverse-KL budget is bounded but reaches the axes. At $\delta = 0.5$ (top) all three hold ρ off the axes, and the interval widens from χ^2 to reverse KL. At $\delta = 1.2$ (bottom) the χ^2 and reverse-KL segments reach the axes, so a response can be driven out of the support; forward KL does not, at any δ. The thresholds are those of corollary 8, here $1/(K-1) = 1$ and $\log 2 \approx 0.69$.

(Forward KL.) With $f(\rho) = -\log\rho$, the equation $g(\rho) = K\delta$ is $-\log\rho - (K-1)\log\frac{K-\rho}{K-1} = K\delta$, that is

$$\rho\,(K-\rho)^{K-1} \;=\; (K-1)^{K-1}e^{-K\delta}, \tag{17}$$

whose two positive roots are the endpoints. At $K = 2$ this is the quadratic $\rho(2-\rho) = e^{-2\delta}$, with roots $1 \pm \sqrt{1-e^{-2\delta}}$; Fig. 2 is drawn at $\delta = 0.5$, where they are 0.205 and 1.795.

(Reverse KL.) With $f(\rho) = \rho\log\rho$,

$$\rho\log\rho + (K-\rho)\log\frac{K-\rho}{K-1} \;=\; K\,\delta, \tag{18}$$

which has no closed form and is solved by bisection on each side of 1.

E.3 When the trust region preserves support

The main text reads the lower endpoint off $g(0)$.

Corollary 8 (Support)**.** *The smallest feasible ρ_k is strictly positive if and only if $g(0) = f(0^+) + (K-1)\,f\big(\frac{K}{K-1}\big) > K\,\delta$. For the three instances: forward KL satisfies this for every δ; χ^2 satisfies it iff $\delta < 1/(K-1)$; and reverse KL satisfies it iff $\delta < \log\frac{K}{K-1}$.*

Proof. The first claim is proposition 4 evaluated at $\rho = 0$, since g is convex with $g(1) = 0$ and so $\{g \le K\delta\}$ contains 0 exactly when $g(0) \le K\delta$. For forward KL, $f(0^+) = +\infty$, so

$g(0) = +\infty$ and the condition holds for every finite δ. For χ^2, $f(0) = 1$ and $f\left(\frac{K}{K-1}\right) = \frac{1}{(K-1)^2}$, so $g(0) = 1 + \frac{1}{K-1} = \frac{K}{K-1}$, and $g(0) > K\delta$ reads $\delta < 1/(K-1)$; the same threshold follows from eq. (16). For reverse KL, $f(0^+) = 0$ and $f\left(\frac{K}{K-1}\right) = \frac{K}{K-1} \log \frac{K}{K-1}$, so $g(0) = K \log \frac{K}{K-1}$ and the condition reads $\delta < \log \frac{K}{K-1}$. □

The upper end needs no such condition. Proposition 4 gives $\rho_k \leq K$ from normalization alone, and the budget improves on it strictly whenever $g(K) > K\delta$, where $g(K) = f(K) + (K-1)f(0^+)$. This is $+\infty$ for forward KL, $K(K-1)$ for χ^2 and $K \log K$ for reverse KL, so for any δ of a size one would use, the trust region is strictly interior to $[0, K]$ at its upper end under all three generators.