PREFERENCE OPTIMIZATION FOR DISTRIBUTION-LEVEL OBJECTIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training typically targets individual-level objectives, which evaluate a policy through the expected value of a per-response reward. Broader goals like diversity, calibration, and fairness, however, demand distribution-level objectives that depend on the collective outputs and probabilities of a policy. In this work, we consider jointly optimizing individual- and distribution-level objectives, and make the following contributions. (i) We formulate the joint optimization problem and derive a per-response linearization reward for the distribution-level objective. This motivates the development of an iterative surrogate optimization method with dynamic linearization, which we show provably converges to a solution of the joint objective. (ii) We adapt existing post-training methods to the surrogate problems, introducing a general framework for incorporating distribution-level objectives into standard post-training pipelines. Experiments across three testbeds demonstrate improved distribution-level outcomes.

1 INTRODUCTION

Post-training adapts pretrained language models using reward signals and preference data. Reinforcement learning from human feedback optimizes learned rewards (Ouyang et al., 2022), while preference-based methods such as DPO and IPO train directly on response comparisons (Rafailov et al., 2023; Azar et al., 2023). A common perspective underlying these methods is to improve an individual-level value: the expected score of a response under a fixed per-response reward, with regularization towards a reference policy. This perspective captures many background objectives of post-training, including response quality and answer correctness.

However, desired policy behavior could also involve properties of the response distribution. A creative model should produce varied responses; stated confidence should reflect empirical correctness; and prediction errors should not be distributed unevenly across demographic groups. Existing methods address these goals through diversity-aware preference training (Lanchantin et al., 2025; Chung et al., 2025; Ismayilzada et al., 2025; Samuel et al., 2026), calibration-oriented finetuning (Band et al., 2024; Stengel-Eskin et al., 2024; Leng et al., 2024; Damani et al., 2026), and interventions for reducing demographic bias (Zmigrod et al., 2019; De-Arteaga et al., 2019). These approaches motivate a broader methodological question:

> *How can we systematically adapt existing post-training methods to jointly optimize an individual-level objective and a specified distribution-level objective?*

We study this question through objectives defined on feature expectations under the policy, allowing both prompt-specific properties and aggregation across prompts. To adapt existing training methods, we seek a per-response signal for the distribution-level objective that can be combined with the background reward or incorporated into a preference loss. A fixed per-response reward generally cannot represent the nonlinear dependence on the policy's feature expectations. We instead linearize the distribution-level objective at an anchor policy, obtaining a *linearization reward* that captures each response's marginal contribution at that anchor. With the anchor held fixed, this reward can be added to the background reward to define a surrogate problem.

We optimize the joint objective through a sequence of frozen-anchor surrogate problems, using a generalized conditional-gradient scheme that approximately solves each problem, takes a damped

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.

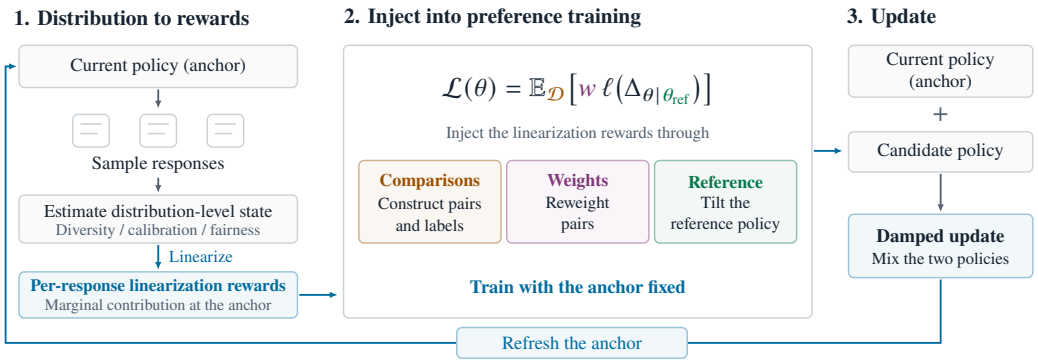

Figure 1: **A framework for post-training with distribution-level objectives.**

policy update, and refreshes the anchor. This approach builds on conditional-gradient methods for optimizing functions of policy-induced distributions through reward-based subproblems (Hazan et al., 2019; Geist et al., 2022). Under explicit regularity and policy-class assumptions, we establish convergence guarantees that account for reward-estimation and inner-optimization errors.

To implement each surrogate solve, we develop a general framework for injecting the linearization reward into preference training through comparisons, weights, or the reference policy, while preserving the loss link. Within this framework, our principal new construction is *reference tilting*: it shifts the reference-relative preference margin by the linearization-reward gap divided by the regularization coefficient. Under stated assumptions, this simple modification faithfully targets the surrogate optimizer while reusing existing preference data without additional background-reward evaluations. The comparison and weight sites provide complementary constructions, generalizing strategies used in DivPO and DDPO (Lanchantin et al., 2025; Chung et al., 2025) to incorporate linearization rewards derived from a broader class of distribution-level objectives. Our population-minimizer and training-gradient analyses characterize the guarantees and effects of these injections across preference losses.

Experiments on diversity, confidence calibration, and group fairness demonstrate how these constructions change distribution-level and background outcomes. Reference tilting improves distribution-level outcomes over ordinary preference training across all three testbeds, while retaining substantial background gains in diversity and fairness. The preference loss affects the response to injection: IPO produces larger diversity and calibration-ECE changes than DPO in the evaluated configurations. At matched gradient-step budgets, nested schedules also yield fewer validation-selection failures in calibration and fairness. Together, the results identify practical choices in how to inject the reward, which preference loss to use, and when to refresh the anchor.

2 Formulation & Objective

Let $\mathcal{X}$ be a prompt space with prompt distribution $\rho \in \Delta(\mathcal{X})$, and let $\mathcal{Y}$ be a response space. A policy π maps each prompt $x \in \mathcal{X}$ to a distribution $\pi(\cdot \mid x) \in \Delta(\mathcal{Y})$, and Π denotes the policy class under consideration. We fix a reference policy π_{ref} with full support on $\mathcal{Y}$ at every prompt, and write $\mathrm{KL}_\rho(\pi\|\nu) = \mathbb{E}_{x\sim\rho}[\mathrm{KL}(\pi(\cdot \mid x)\|\nu(\cdot \mid x))]$ for the prompt-weighted relative entropy.

As motivated in the introduction, we extend standard post-training by jointly optimizing a distribution-level objective alongside the background task objective

$$\max_{\pi\in\Pi} \quad J(\pi) = \underbrace{V_{\mathcal{D}}(\pi) - \beta\,\mathrm{KL}_\rho(\pi\|\pi_{\text{ref}})}_{\text{background post-training objective}} + \underbrace{\Omega(\pi)}_{\text{distribution-level objective}}, \tag{1}$$

where $\beta > 0$ controls regularization toward the reference policy. We absorb the distribution-level weight $\lambda \geq 0$ into Ω, writing $\Omega = \lambda\bar{\Omega}$ for an unweighted functional $\bar{\Omega}$; varying λ adjusts its emphasis relative to the background objective.

The Background Individual-Level Objective. The background post-training objective consists of a value $V_{\mathcal{D}}$ induced by the dataset $\mathcal{D}$ and KL regularization toward π_{ref}. We assume that $V_{\mathcal{D}}$

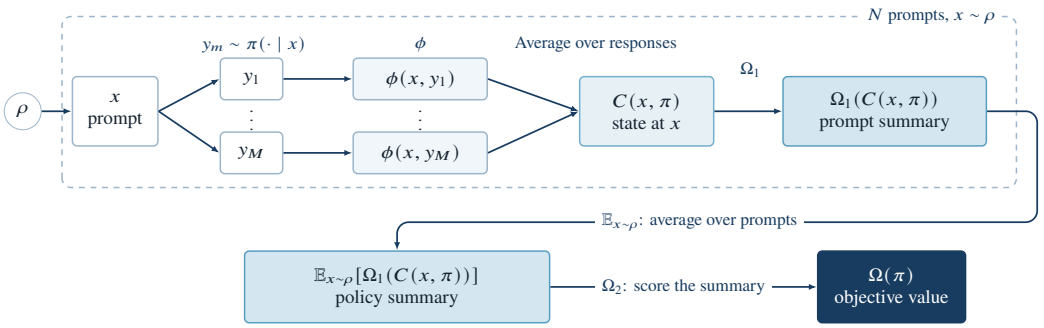

Figure 2: **Distribution-level Objective.**

is *individual-level*. Formally, for a fixed prompt distribution ρ and policy class Π, a bounded, policy-independent reward $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *represents* a value functional $V : \Pi \to \mathbb{R}$ if $V(\pi) = \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x)}[r(x, y)]$ for every $\pi \in \Pi$. We call V *individual-level* if such a representation exists, and write $r_{\mathcal{D}}$ for a reward representing $V_{\mathcal{D}}$.

This assumption captures the value objectives underlying many existing post-training methods. RLHF uses a reward model fitted to human comparisons (Ouyang et al., 2022), while RLVR uses a verifier of correctness (Guo et al., 2025). DPO (Rafailov et al., 2023) and IPO (Azar et al., 2023) connect preference losses to KL-regularized individual-level objectives under their respective modeling assumptions[1]

However, some properties of interest *depend on how probability is distributed across responses and cannot be expressed as the expected value of a fixed per-response reward*. We call an objective *distribution-level* if it admits no such representation on Π. We next introduce a state-based model for these objectives.

The Distribution-Level Objective. A fixed *feature map* $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^k$ records the response attributes relevant to the objective. Its expectation $C(x, \pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[\phi(x, y)]$ defines the *state* induced by π at prompt x. We consider objectives that depend on the policy through these states (see Fig.2):

$$\Omega(\pi) = \Omega_2\Big(\mathbb{E}_{x\sim\rho}\big[\Omega_1(C(x, \pi))\big]\Big). \tag{2}$$

The *state-summary function* $\Omega_1 : \mathbb{R}^k \to \mathbb{R}^{k_1}$ summarizes each prompt's state, and the *summary-score function* $\Omega_2 : \mathbb{R}^{k_1} \to \mathbb{R}$ scores the average summary. Bare Ω denotes the objective as a functional of the policy, which we seek to maximize. For the theoretical analysis, we assume a finite response space, bounded features, and Lipschitz continuity and smoothness of both maps on neighborhoods of the relevant states and summaries (Assumption 2, Appendix D.1).

The following examples illustrate two uses of this construction: scoring each prompt's state before averaging ($\Omega_2 = \mathrm{id}$), and scoring the pooled state across prompts ($\Omega_1 = \mathrm{id}$). Additional examples of distribution-level objectives (discrete mode coverage, per-prompt reliability, and teacher matching) are in Appendix C.

Example 1 (Continuous diversity)**.** *Let $\phi(x, y) = e(x, y)$ be a unit-normalized sentence embedding, so $C(x, \pi)$ is the mean response embedding at x. Take $\Omega_1(C) = -\frac{\lambda}{2}\|C\|^2$, $\lambda > 0$, and $\Omega_2 = \mathrm{id}$, giving $\Omega(\pi) = -\frac{\lambda}{2}\mathbb{E}_{x\sim\rho}\|C(x, \pi)\|^2$. For the cosine kernel $k(y, y') = \langle e(x, y), e(x, y')\rangle$ at prompt x, $\|C(x, \pi)\|^2 = \mathbb{E}_{y,y'\sim\pi(\cdot|x)}[k(y, y')]$, where the responses are independent. Maximizing Ω therefore reduces the average similarity between responses to the same prompt.*

Example 2 (Confidence-binned reliability)**.** *Let $c(x, y) \in [0, 1]$ denote stated confidence, $a(x, y) \in \{0, 1\}$ denote correctness, and $b_m(c)$ indicate membership in confidence band m. Choose $\phi(x, y) = (b_m(c), b_m(c)(c - a))_{m=1}^M$ and $\Omega_1 = \mathrm{id}$. The pooled state $\mathbb{E}_{x\sim\rho}[C(x, \pi)] = (P_m, D_m)_{m=1}^M$*

[1] In the Ψ-preference framework of Azar et al. (2023), fixing a comparison policy μ gives the per-response reward $r_{\mathcal{D}}(x, y) = \mathbb{E}_{y'\sim\mu(\cdot|x)}[\Psi(\mathcal{P}(y \succ y' \mid x))]$, where $\mathcal{P}$ is the preference kernel. IPO uses $\Psi(q) = q$, allowing non-transitive preferences. The connection to DPO uses $\Psi(q) = \log(q/(1 - q))$ under the Bradley–Terry assumption.

records each band's probability mass P_m and mass-weighted confidence gap D_m. Set $\Omega_2(P, D) = -\frac{\lambda}{2}\sum_{m=1}^{M} D_m^2/(P_m + \delta)$, with $\lambda, \delta > 0$. Without smoothing, each nonempty band's penalty is $P_m(D_m/P_m)^2$: its mass times the squared difference between mean confidence and accuracy. The positive δ keeps the objective differentiable at empty bands.

3 Iterative Optimization via Linearization

3.1 The Linearization Reward

The combinatorial size of the response space makes explicit enumeration impractical in language-model post-training. Existing methods typically construct updates from sampled or recorded responses, using signals that act through their log probabilities. Our aim is to adapt these methods to the joint objective. A fixed per-response reward representation of Ω would place it in the same expected-reward form as $V_{\mathcal{D}}$, allowing its contribution to be incorporated through existing reward-based or preference-based training interfaces.

In general, however, Ω admits no fixed per-response reward representation. Under the conditions of Proposition 2 in Appendix D.2, such a representation exists exactly when the objective is affine on the reachable state profiles. The objectives motivating this paper, including continuous diversity and confidence-binned reliability, generally fail this affineness condition (see also Appendix C). We therefore seek a *local representation*. Linearizing Ω at an *anchor* policy $\pi^\dagger$ yields a per-response reward that represents its first-order variation. This *linearization reward* remains fixed during a surrogate optimization step and is recomputed when the anchor changes.

Fix an *anchor* policy $\pi^\dagger \in \Pi$. Applying the chain rule to eq. (2), define the *state gradient* $a_x = D\Omega_1(C(x, \pi^\dagger))^\top \nabla\Omega_2(\mathbb{E}_{x'\sim\rho}[\Omega_1(C(x', \pi^\dagger))])$, where $D\Omega_1$ denotes the Jacobian of Ω_1[2]. The corresponding *linearization reward* is

$$r_\Omega(x, y; \pi^\dagger) := \langle a_x, \phi(x, y)\rangle, \tag{3}$$

which is policy-independent when the anchor is held fixed. This reward measures the marginal contribution of a response through its effect on the state. At a fixed prompt, reallocating probability mass from y' to y changes the state in the direction $\phi(x, y) - \phi(x, y')$, whose inner product with a_x is $r_\Omega(x, y; \pi^\dagger) - r_\Omega(x, y'; \pi^\dagger)$. Thus reward differences describe the local value of reallocating probability between responses; an additive function of the prompt does not affect this interpretation.

More generally, the reward gives the first-order expansion

$$\Omega(\pi) = \Omega(\pi^\dagger) + \mathbb{E}_{x\sim\rho}\Big[\mathbb{E}_{y\sim\pi(\cdot|x)}[r_\Omega(x, y; \pi^\dagger)] - \mathbb{E}_{y\sim\pi^\dagger(\cdot|x)}[r_\Omega(x, y; \pi^\dagger)]\Big] + R(\pi, \pi^\dagger). \tag{4}$$

Apart from terms independent of π, the linear approximation is therefore the expected linearization reward. Under Assumption 2 in Appendix D.1, the remainder satisfies $|R(\pi, \pi^\dagger)| \le \frac{L_\Omega}{2}\mathbb{E}_{x\sim\rho}\|C(x, \pi) - C(x, \pi^\dagger)\|^2$, where $L_\Omega = M_2 L_1 + M_1^2 L_2$. The approximation error is thus quadratic in the change in states. If Ω is concave along the segment joining $\pi^\dagger$ and π, then $R(\pi, \pi^\dagger) \le 0$, so the linear approximation is an upper bound. Lemma 1 in Appendix D.3 gives the formal statement and proof.

Example 3 (Continuous diversity, continued)**.** *For Example 1, $\Omega_1(C) = -\frac{\lambda}{2}\|C\|^2$ and $\Omega_2 = \mathrm{id}$, so $a_x = -\lambda C(x, \pi^\dagger)$ and $r_\Omega(x, y; \pi^\dagger) = -\lambda\langle C(x, \pi^\dagger), e(x, y)\rangle = -\lambda\mathbb{E}_{y'\sim\pi^\dagger(\cdot|x)}[k(y, y')]$. Responses with lower average similarity to the anchor's responses receive higher rewards. The reward therefore favors responses that are less represented under the anchor and changes as the anchor's response distribution changes.*

Freezing the linearization reward yields a surrogate with an individual-level value objective. We next use this surrogate in an iterative scheme that refreshes the anchor as optimization proceeds.

3.2 Iterative Surrogate Optimization

Iterative optimization. We use the linearization developed above in a *generalized conditional gradient* scheme (Frank & Wolfe, 1956; Levitin & Polyak, 1966; Bredies et al., 2009), which linearizes

[2] $D\Omega_1(z)$ denotes the Jacobian with respect to the input state z, with entries $[D\Omega_1(z)]_{ij} = \partial(\Omega_1)_i(z)/\partial z_j$.

Ω while retaining the background post-training objective exactly. At an anchor $\pi^\dagger$, omitting terms independent of π gives the *frozen-anchor surrogate problem*:

$$\max_{\pi\in\Pi} \quad J_{\pi^\dagger}(\pi) := V_{\mathcal{D}}(\pi) + \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x)}[r_\Omega(x,y;\pi^\dagger)] - \beta\,\mathrm{KL}_\rho(\pi\|\pi_{\mathrm{ref}}). \tag{5}$$

At each round, we anchor this surrogate problem at the current policy, approximately solve it, and take a convex combination of the current policy and the returned policy. The updated policy becomes the next anchor. The reference π_{ref} remains fixed throughout; only the anchor and its linearization reward are refreshed. Damping controls the linearization error: the surrogate improvement enters the progress bound linearly in the step size, whereas the remainder is quadratic.

At round t, we estimate the linearization reward from anchor samples by replacing the feature expectations with empirical averages. Let $\widehat{r}_{\Omega,t}$ denote this estimate and $\widehat{J}_{\pi_t}$ the *estimated surrogate value*, obtained by substituting $\widehat{r}_{\Omega,t}$ into eq. (5)[3]. The estimated reward is held fixed during the inner solve. Appendix D.5 gives the sampling constructions and their conditions.

Algorithm 1 Generalized conditional gradient for the joint objective

Require: initial policy π_0, rounds T, step sizes $\{\alpha_t\}_{t=0}^{T-1} \subseteq [0,1]$, accuracy $\varepsilon \geq 0$
1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: Estimate the linearization reward at π_t: $\widehat{r}_{\Omega,t}(x,y) = \langle \widehat{a}_{t,x}, \phi(x,y)\rangle$.
3: Find $\pi_t^* \in \Pi$ with $\widehat{J}_{\pi_t}(\pi_t^*) \geq \max_{\pi\in\Pi} \widehat{J}_{\pi_t}(\pi) - \varepsilon$.
4: $\pi_{t+1} \leftarrow (1-\alpha_t)\pi_t + \alpha_t \pi_t^*$
5: **end for**
6: **return** π_T

Error and convergence guarantees. The inner solve has two sources of error: the optimization tolerance ε and the error in the estimated linearization reward. Writing $a_{t,x}$ and $\widehat{a}_{t,x}$ for the exact and estimated state gradients, let $\zeta_t := B\,\mathbb{E}_{x\sim\rho}\|\widehat{a}_{t,x} - a_{t,x}\|$. By Lemma 3 in Appendix D.5, $|\widehat{J}_{\pi_t}(\pi) - J_{\pi_t}(\pi)| \leq \zeta_t$ uniformly over policies with finite KL divergence. Consequently, an ε-optimal solution of the estimated surrogate problem is $(\varepsilon + 2\zeta_t)$-optimal for the population surrogate problem. The factor of two accounts for estimation error at both the returned policy and a population maximizer.

We analyze the scheme under the following conditions, with concavity required only for the global optimality guarantee.

Assumption 1. *(i) The nonempty class Π of measurable policies is convex and compact under $d_\rho(\pi,\nu) = \mathbb{E}_{x\sim\rho}\|\pi(\cdot \mid x) - \nu(\cdot \mid x)\|_1$, identifying policies that agree ρ-almost everywhere. (ii) The objective Ω is concave on Π.*

We measure progress using the *surrogate gap* $\mathfrak{g}(\pi^\dagger) := \max_{\pi\in\Pi} J_{\pi^\dagger}(\pi) - J_{\pi^\dagger}(\pi^\dagger)$, the improvement available from solving the surrogate problem. Under Assumption 2 and Assumption 1(i), at policies with finite J, the surrogate gap vanishes exactly when $\pi^\dagger$ is first-order stationary for J[4] (Lemma 2, Appendix D.4).

Proposition 1 (Convergence of the scheme)**.** *Let Assumption 2 and Assumption 1(i) hold. Run Algorithm 1 from $\pi_0 = \pi_{\mathrm{ref}} \in \Pi$, and suppose $\zeta_t \leq \zeta < \infty$ almost surely at every round. The inner accuracy holds almost surely, and the returned policies are measurable functions of the algorithm's randomness. Put $\Delta = 2B$ and $\Delta_J := \sup_{\pi\in\Pi} J(\pi) - J(\pi_0) < \infty$. Then, for every $T \geq 1$, the following bounds hold almost surely:*

1. *If $\Delta_J > 0$ and $L_\Omega\Delta^2 > 0$, the constant step size $\alpha_t = \min\{\sqrt{2\Delta_J/(L_\Omega\Delta^2 T)}, 1\}$ gives $\min_{0\leq t<T} \mathfrak{g}(\pi_t) \leq \frac{\max\{2\Delta_J, L_\Omega\Delta^2\}}{\sqrt{T}} + \varepsilon + 2\zeta$.*

[3]The expectations in $\widehat{J}_{\pi_t}$ remain over $x \sim \rho$ and responses from the policy being optimized: the hat denotes reward estimation, not an empirical training loss.

[4]We call $\pi^\dagger$ first-order stationary if $J'(\pi^\dagger; \pi - \pi^\dagger) \leq 0$ for every $\pi \in \Pi$ with finite $J(\pi)$, where J' denotes the one-sided directional derivative. When J is differentiable, this is equivalent to $\langle \nabla J(\pi^\dagger), \pi - \pi^\dagger\rangle_\rho \leq 0$. The directional formulation accommodates boundary policies where the KL term is not differentiable.

2. *If Assumption 1(ii) also holds and J attains its maximum at $\pi^\star$, then $\alpha_t = 2/(t+2)$ gives* $J(\pi^\star) - J(\pi_T) \le \frac{8L_\Omega B^2}{T+2} + \varepsilon + 2\zeta$.

If instead $\mathbb{E}[\zeta_t] \le \zeta$ at every round, the same bounds hold with left-hand sides $\mathbb{E}[\min_{0\le t<T} \mathfrak{g}(\pi_t)]$ and $\mathbb{E}[J(\pi^\star) - J(\pi_T)]$, respectively, where expectations are over the algorithm's randomness.

The bounds control the smallest surrogate gap without concavity and the final suboptimality under concavity, up to the error floor $\varepsilon + 2\zeta$. The extension in expectation incorporates the sampling-error bounds of Lemma 3 (see Appendix D.6). These guarantees require policy-space mixtures and the stated inner-solve accuracy. Section 4 develops practical implementations of the surrogate solve using existing post-training methods.

4 Implementing the Surrogate Solve

The inner step of Algorithm 1 approximately solves the frozen-anchor surrogate problem. In this section, we develop training losses and reward constructions that incorporate the linearization reward through existing post-training interfaces. Reward-based methods, such as RLHF and RLVR, use scalar rewards assigned to generated responses to construct policy updates. At a fixed anchor $\pi^\dagger$, replacing the background reward $r_\mathcal{D}$ with $r_\mathcal{D} + r_\Omega(\cdot\,;\pi^\dagger)$ while retaining KL regularization gives the frozen-anchor surrogate problem. The policy-gradient identity and implementation considerations appear in Appendix E.1. We focus here on preference-based methods, whose training interfaces use labeled comparisons rather than explicit rewards.

4.1 Preference-Loss Interface

We consider pairwise, reference-based preference methods, including DPO and IPO. For responses y, y' to a prompt x, define the *reference-relative margin* $\Delta_{\theta|\theta_{\mathrm{ref}}}(x, y, y') = \log \frac{\pi_\theta(y|x)}{\pi_{\theta_{\mathrm{ref}}}(y|x)} - \log \frac{\pi_\theta(y'|x)}{\pi_{\theta_{\mathrm{ref}}}(y'|x)}$. The preference loss takes the form

$$\mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](\theta) = \mathbb{E}_{(x,y^w,y^l)\sim\mathcal{D}}\left[w(x, y^w, y^l)\,\ell\big(\Delta_{\theta|\theta_{\mathrm{ref}}}(x, y^w, y^l)\big)\right], \tag{6}$$

where $\mathcal{D}$ is a distribution over labeled comparisons, with y^w recorded as preferred to y^l; $w > 0$ is a per-pair weight; and ℓ is a scalar *link*. A finite dataset enters through its empirical distribution, and ordinary unweighted training uses $w = \mathbf{1}$. The DPO link is $\ell_{\mathrm{DPO}}(u) = -\log\sigma(\beta u)$ (Rafailov et al., 2023), where $\sigma(z) = 1/(1+e^{-z})$; the IPO link is $\ell_{\mathrm{IPO}}(u) = (u - \frac{1}{2\beta})^2$ (Azar et al., 2023).

A *linearization-reward injection* uses $r_\Omega(\cdot\,;\pi^\dagger)$ to modify the *comparison*, *weight*, or *reference* inputs $\mathcal{D}$, w, and θ_{ref}, respectively, while keeping the link ℓ fixed. The resulting *injected preference loss* $\mathcal{L}[\mathcal{D}', w', \theta'_{\mathrm{ref}}]$ holds these inputs fixed during each inner training phase. Section 4.2 presents constructions at these three sites.

4.2 Constructions at the Three Injection Sites

We introduce *reference tilting* for reference-site injection as our principal construction, then present complementary comparison- and weight-site injections. Throughout, the anchor $\pi^\dagger$ is fixed.

Reference site. A reference-site injection modifies the reference distribution used to compute preference margins, while preserving the comparisons and weights. We consider the *reference-tilting injection*, which uses the linearization reward to define a Gibbs tilt of the original reference policy, $\widetilde{\pi}_{\mathrm{ref}}(y \mid x) \propto \pi_{\mathrm{ref}}(y \mid x)\exp\big(r_\Omega(x, y; \pi^\dagger)/\beta\big)$, with normalization over responses at each prompt. We write $\widetilde{\theta}_{\mathrm{ref}}$ for this reference in the loss notation and define the *linearization-reward gap* $\Delta r_\Omega(x, y, y'; \pi^\dagger) := r_\Omega(x, y; \pi^\dagger) - r_\Omega(x, y'; \pi^\dagger)$. The normalizer cancels in the pairwise log ratio, giving $\Delta_{\theta|\widetilde{\theta}_{\mathrm{ref}}}(x, y, y') = \Delta_{\theta|\theta_{\mathrm{ref}}}(x, y, y') - \Delta r_\Omega(x, y, y'; \pi^\dagger)/\beta$. For unit weights, the associated *reference-tilted preference loss* is therefore

$$\mathcal{L}[\mathcal{D}, \mathbf{1}, \widetilde{\theta}_{\mathrm{ref}}](\theta) = \mathbb{E}_{(x,y^w,y^l)\sim\mathcal{D}}\left[\ell\left(\Delta_{\theta|\theta_{\mathrm{ref}}}(x, y^w, y^l) - \frac{\Delta r_\Omega(x, y^w, y^l; \pi^\dagger)}{\beta}\right)\right]. \tag{7}$$

The reward gap shifts the margin while leaving the link unchanged. For a finite dataset, replacing the expectation in eq. (7) with an empirical mean gives the training loss. It uses the original reference policy and estimated reward gaps, without constructing $\widetilde{\pi}_{\mathrm{ref}}$ or computing its normalizer. Moreover, it reuses the comparison data, requires no additional background-reward evaluations, and reduces to ordinary preference training when the linearization reward vanishes.

Comparison site. A comparison-site injection modifies the labeled-comparison distribution. A *comparison rule* specifies both pair selection and labeling: selection chooses an unordered pair, while labeling identifies its preferred response (Appendix E.2). We give two constructions, the *combined-score injection* and *task-gated injection*, defined by the following rules.

Given candidates $\mathcal{Y}_x$, define $s(x, y) = r_{\mathcal{D}}(x, y) + r_{\Omega}(x, y; \pi^{\dagger})$, and let $\mathcal{Y}_x^+, \mathcal{Y}_x^- \subseteq \mathcal{Y}_x$ be eligible winner and loser sets defined by background-reward gates. The *combined-score rule* and *task-gated rule* are

$$\begin{aligned} \textit{Combined-score rule:} \quad & y^w \in \arg\max_{y \in \mathcal{Y}_x} s(x, y), & & y^l \in \arg\min_{y \in \mathcal{Y}_x} s(x, y), \\ \textit{Task-gated rule:} \quad & y^w \in \arg\max_{y \in \mathcal{Y}_x^+} r_{\Omega}(x, y; \pi^{\dagger}), & & y^l \in \arg\min_{y \in \mathcal{Y}_x^-} r_{\Omega}(x, y; \pi^{\dagger}). \end{aligned}$$

The first ranks candidates by the sum of the background and linearization rewards; the second uses the background reward for eligibility and the linearization reward for ranking, generalizing DivPO's gating-and-ranking construction (Lanchantin et al., 2025). Each rule induces a comparison distribution $\mathcal{D}'$ and loss $\mathcal{L}[\mathcal{D}', \mathbf{1}, \theta_{\mathrm{ref}}]$. Both require linearization-reward estimates and an evaluable background reward, which recorded preference labels alone do not generally provide.

Weight site. A *reward-weighting injection* constructs positive pair weights $w'(x, y^w, y^l)$ from $r_{\Omega}(\cdot\,; \pi^{\dagger})$, yielding $\mathcal{L}[\mathcal{D}, w', \theta_{\mathrm{ref}}]$. A related construction is DDPO (Chung et al., 2025), which emphasizes atypical preferred responses by weighting pairs according to the winner's deviation from other responses to the same prompt. Our framework accommodates weights derived from the chosen distribution-level objective. Implementation preserves the labels and reference and requires no background-reward evaluations on new candidates.

Injections at different sites are composable: comparisons constructed by a combined-score or task-gated injection can also receive reward-based weights and be trained against a tilted reference.

Population minimizers and gradient effects. Over the full-support tabular class, reference tilting recovers the surrogate optimizer whenever the uninjected loss identifies the background post-training optimizer. Positive weighting preserves preference-loss minimizers when all recorded pairwise losses can be minimized simultaneously; otherwise, it may change the optimizer. Order-only comparison rules cannot recover the surrogate optimizer uniformly across score scales, although suitably calibrated probabilistic labels can recover it. These results and their assumptions appear in Appendices E.3.1–E.3.3. For reference tilting, the gradient effect depends on the link: IPO shifts each pair's minimizing margin and can reverse its gradient contribution, whereas DPO changes its magnitude but not its sign. For a fixed reward gap, DPO's coefficient change vanishes at extreme margins (Appendix E.4). Together, these results clarify how each injection changes the optimization target and training signal, and motivate the empirical comparisons across injections and preference losses in Section 5.

5 Experiments

We evaluate linearization-reward injections on diversity, confidence calibration, and group fairness, each pairing a background individual-level reward with a distribution-level objective. Our experiments examine how injections change distribution-level and background individual-level outcomes, how these changes depend on the preference loss, and what anchor refresh contributes at a fixed training-step budget.

5.1 Setup

Across the testbeds in Table 1, we train LoRA adapters on Llama-3.1-8B-Instruct (Grattafiori et al., 2024), keeping the base policy as π_{ref}. For diversity, Distinct$_k$ counts semantic response

Table 1: Experimental testbeds: WritingPrompts (Fan et al., 2018), TriviaQA (Joshi et al., 2017), and Bias in Bios (De-Arteaga et al., 2019). The objectives penalize mean embedding similarity, the squared prompt-level mean confidence gap, and weighted group-conditional prediction-rate disparities, respectively. Arrows indicate preferred metric directions.

	Diversity	**Calibration**	**Fairness**
Task	Story generation	QA with confidence	Occupation prediction
Dataset	WritingPrompts	TriviaQA, no context	Bias in Bios, unscrubbed
IL reward $r_{\mathcal{D}}$	GRM	Exact match	Occupation correctness
DL objective Ω	Embedding diversity	Prompt-level calibration	Equalized odds
IL outcomes	GRM ↑, LitBench ↑	Accuracy ↑	Accuracy ↑
DL metrics	Distinct$_k$ ↑, Vendi ↑	ECE ↓, AUROC ↑	Noise-corrected EOdds ↓, ρ_{gap} ↓

groups (Zhang et al., 2025), Vendi measures similarity-based effective diversity (Friedman & Dieng, 2023), and LitBench (Fein et al., 2026) assesses quality beyond the GRM training reward (Yang et al., 2024). For calibration, ECE measures binned confidence–accuracy gaps (Guo et al., 2017), and AUROC measures how confidence ranks correct versus incorrect responses (Fawcett, 2006). For fairness, noise-corrected EOdds measures occupation-conditional gender disparities in predictions (equalized odds) (Hardt et al., 2016),[5] while ρ_{gap} correlates signed gender accuracy gaps with female representation across occupations (De-Arteaga et al., 2019).

Methods and schedules. We compare injected IPO/DPO with uninjected controls, DivPO/DDPO for diversity, and counterfactual augmentation/scrubbing for fairness. The *static* schedule performs one surrogate-optimization round of Algorithm 1 ($T = 1$), keeping the initial anchor and estimated reward fixed. The *nested* schedule alternates approximate inner solves with damped response-distribution mixtures $\pi_{t+1} = (1 - \alpha_t)\pi_t + \alpha_t \pi_t^*$, refreshing candidates and rewards at each new anchor. We use $T = 3$ for diversity/fairness and $T = 4$ for calibration. At matched total gradient steps, we study what anchor refresh contributes with fewer steps per inner solve and estimation error in refreshed rewards. Static training provides a lower-cost baseline, matching ordinary IPO/DPO's gradient-step budget without repeated sampling and scoring. Methods without comparison injections share data within matched configurations; nested methods refresh comparisons after the shared initial round.

Selection and reporting. We select policies using validation criteria and gates on background outcomes and response validity, then evaluate on disjoint held-out data. When no candidate passes, testbed-specific rules determine fallback selection or exclusion; the appendix reports these rules and per-arm coverage. Each testbed targets ten dataset draws, paired against the corresponding base policy. Diversity and fairness average up to three initialization replicates per dataset before computing paired statistics; calibration varies data and initialization jointly across ten seeds.

Appendix F details objectives, metrics, configurations, baselines, and selection and reporting protocols, including fallbacks and failed runs.

5.2 Results across Distribution-Level Objectives

We vary the distribution-level weight λ, with $\lambda = 0$ recovering the background post-training objective. Appendix F reports full results and paired comparisons, including compositions of injections.

What changes when linearization rewards are injected? Reference tilting and combined-score injections improve distribution-level outcomes across the testbeds. Relative to ordinary IPO, reference tilting increases diversity Distinct$_k$ by 3.87 and Vendi by 0.83 at $\lambda = 8$. For fairness, reference tilting at $\lambda = .25$ reduces noise-corrected EOdds by 0.0238 and ρ_{gap} by 0.0262 relative to base. For calibration, reference tilting at $\lambda = 2$ increases AUROC by 3.70 percentage points (pp) and reduces ECE by 14.91 pp; combined-score IPO improves these metrics by 2.17 pp and 9.88 pp, respectively, relative to ordinary IPO. The AUROC gains reflect improved confidence–correctness ranking, beyond what a uniform strictly increasing rescaling of confidence can achieve.

[5] We subtract estimated sampling variances from squared gender gaps over fixed prediction categories (Gretton et al., 2012). This removes squared-scale sampling bias under the stated assumptions; clipping and square roots do not preserve unbiasedness.

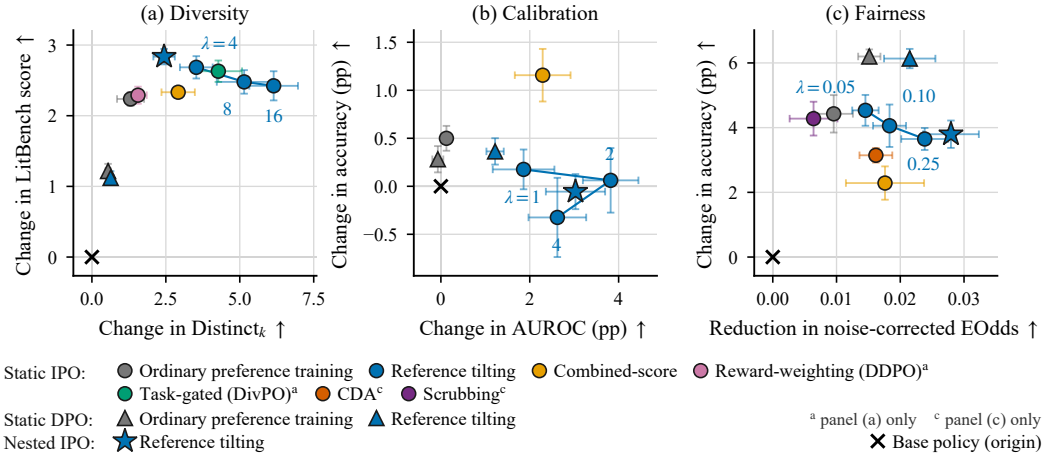

Figure 3: **Distribution-level and background individual-level outcomes across three testbeds.** Points show mean paired changes from the corresponding base policy; error bars indicate ± 1 standard error. Lines connect static reference-tilting IPO strengths. Here, pp denotes percentage points. Appendix F provides experimental protocols and reporting details.

The balance with background outcomes depends on the injection and its strength. Diversity reference tilting retains a GRM gain of 1.92 over base, although 2.58 below ordinary IPO, while improving LitBench by 0.24 over that control. The fairness sweep illustrates this balance: $\lambda = .05$ retains a 4.53 pp accuracy gain over base with a 0.0145 EOdds reduction, whereas $\lambda = .25$ achieves a larger fairness improvement while retaining a 3.65 pp accuracy gain. For calibration, reference tilting at $\lambda = 2$ changes accuracy little relative to its uncertainty (+0.06 pp versus base, SE 0.34), whereas combined-score IPO improves accuracy by 1.16 pp. Figure 3 summarizes these tradeoffs.

How does the preference loss affect the injection? Reference tilting produces larger observed distribution-level gains under IPO than DPO across all three testbeds. At matched $\lambda = 8$, its Distinct$_k$ gains are 3.87 versus 0.08; the paired difference between these effects is 3.78 (SE 0.80). For calibration at matched $\lambda = 4$, ECE reductions are 16.12 versus 2.77 pp, a difference of 13.35 pp (SE 1.58 pp), while AUROC gains are 2.50 versus 1.29 pp. For fairness at matched $\lambda = .25$, noise-corrected EOdds reductions are 0.0132 versus 0.0063. All gains are relative to the respective uninjected controls. The calibration-AUROC and fairness interactions are not statistically significant, so the evidence is strongest for diversity Distinct$_k$ and calibration ECE. These loss-dependent effects are consistent with the gradient characterizations in Section 4.

What does anchor refresh change at a fixed training budget? Nested training improves selection reliability in calibration and fairness at matched gradient-step budgets. For calibration at $\lambda = 4$, all ten nested runs satisfy the full validation criteria, compared with four static runs; the remaining six static runs use fallback selection and remain included in the reported results. In the matched fairness schedule comparison at $\lambda = .25$, all 30 nested runs yield an eligible policy for reporting, compared with 26 static runs. Across the matched comparisons, nested training also yields favorable mean changes in calibration ECE (-1.70 pp), AUROC ($+0.41$ pp), and fairness noise-corrected EOdds (-0.0052), although these paired differences are not statistically significant.

Nested training also improves the balance between background reward and diversity. In the matched comparison at $\lambda = 8$, it increases the reward-plus-diversity diagnostic $V_{\mathcal{D}} + \Omega$ by 1.73 over static training, under the diagnostic checkpoint protocol detailed in Appendix F. This gain reflects a better weighted combination of GRM reward and the diversity objective, rather than necessarily a larger gain in each evaluation metric. The diagnostic excludes KL regularization, and the schedule comparisons assess reward re-estimation together with candidate refresh and policy selection.

Together, these experiments show how linearization-reward injections extend post-training to distribution-level objectives, with the balance of outcomes depending on the injection, preference loss, and refresh schedule.

AI USE STATEMENT

We used generative AI tools to assist with code implementation and experiment execution, and to check and refine mathematical claims and proofs. We also used these tools for literature searches, language refinement, manuscript organization, figure preparation, and organizing and refining presentation of experimental results. The authors carefully reviewed the AI-assisted work, including the mathematical arguments, code, experimental outputs, and their interpretation, and take responsibility for the final content of this work, including all text, claims, and artifacts produced with generative AI assistance.

ETHICS STATEMENT

This work studies methods for jointly optimizing individual- and distribution-level objectives. Their effects depend on the chosen objectives, features, and evaluation populations; improvements in the reported metrics should therefore be interpreted within those choices. In particular, our fairness evaluation uses the gender categories recorded in Bias in Bios, which do not capture the full range of gender identities. Reductions in occupation-conditional prediction disparities do not establish fairness across other groups or criteria, or suitability for consequential applications such as hiring. Similarly, improved confidence calibration does not by itself establish factual reliability or safety. We present these testbeds as evaluations of the methodology, rather than validation for deployment in high-stakes settings.

REPRODUCIBILITY STATEMENT

We describe the optimization scheme and preference-loss constructions in Sections 3 and 4, with assumptions and proofs in Appendices D and E. Appendix F documents the datasets and preprocessing, model and training configurations, evaluation metrics, checkpoint-selection procedures, and statistical reporting protocols, together with detailed results and computational resources. The source code used in this work cannot be publicly released at this time. To support reproducibility, we provide these methodological and experimental details in the paper and appendix.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 60–69. PMLR, 2018. URL https://proceedings.mlr.press/v80/agarwal18a.html.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023. URL https://arxiv.org/abs/2310.12036.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2732–2778, 2024. URL https://proceedings.mlr.press/v235/band24a.html.

Anas Barakat, Ilyas Fatkhullin, and Niao He. Reinforcement learning with general utilities: Simpler variance reduction and large state-action space. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1753–1800. PMLR, 2023. URL https://proceedings.mlr.press/v202/barakat23a.html.

Kristian Bredies, Dirk A. Lorenz, and Peter Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42(2):173–193, 2009. doi: 10.1007/s10589-007-9083-3.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Ioannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. In *The Fourteenth International Conference on Learning Representations*, 2026.

URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cfeb1fa78c5fb06be27c0980aa8ee581-Abstract-Conference.html.

Casey Chu, Jose Blanchet, and Peter Glynn. Probability functional descent: A unifying perspective on GANs, variational inference, and reinforcement learning. In *International Conference on Machine Learning*, pp. 1213–1222. PMLR, 2019.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying large language model post-training for diverse creative writing. *arXiv preprint arXiv:2503.17126*, 2025.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. In *International conference on learning representations*, volume 2026, pp. 59456–59493, 2026.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 120–128, 2019.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 889–898, 2018.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Daniel Fein, Sebastian Russo, Violet Xiang, Kabir Jolly, Rafael Rafailov, and Nick Haber. LitBench: A benchmark and dataset for reliable evaluation of creative writing. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7740–7755, 2026.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1–2):95–110, 1956. doi: 10.1002/nav.3800030109.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Olivier Bachem, Rémi Munos, and Olivier Pietquin. Concave utility reinforcement learning: The mean-field game viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 489–497, 2022. URL https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p489.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/6a9659feb1216f14f7384ba499518b38-Abstract.html.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691, 2019. URL https://proceedings.mlr.press/v97/hazan19a.html.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*, 2023.

Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger E. Beaty. Creative preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9580–9609. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.509. URL https://aclanthology.org/2025.findings-emnlp.509/.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2012.11635.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. URL https://arxiv.org/abs/2309.06180.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization, 2025. URL https://arxiv.org/abs/2501.18101.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF, 2024. URL https://arxiv.org/abs/2410.09724.

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966. doi: 10.1016/0041-5553(66)90114-5.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. SetPO: Set-level policy optimization for diversity-preserving LLM reasoning, 2026. URL https://arxiv.org/abs/2602.01062.

Haoyuan Li, Rui Zhang, and Snigdha Chaturvedi. Improving fairness of large language models in multi-document summarization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1143–1154. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-short.90. URL https://aclanthology.org/2025.acl-short.90/.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jarret Ross. Distributional preference alignment of llms via optimal transport. *Advances in Neural Information Processing Systems*, 37: 104412–104442, 2024.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback, 2023. URL https://arxiv.org/abs/2312.00886.

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, 2022. URL https://arxiv.org/abs/2203.02155.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1912.01703.

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, 2023. URL https://arxiv.org/abs/2305.18290.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3982–3992, 2019.

Vinay Samuel, Yapei Chang, and Mohit Iyyer. Recovering diversity without losing alignment: A DPO recipe for post-trained LLMs, 2026. URL https://arxiv.org/abs/2605.30021.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-aware finetuning for confidence calibration in large language models. In *Advances in Neural Information Processing Systems*, 2024.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment, 2024. URL https://arxiv.org/abs/2405.00675.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In *Advances in Neural Information Processing Systems*, 2024.

Binwei Yao, Zefan Cai, Yun-Shiuan Chuang, Shanglin Yang, Ming Jiang, Diyi Yang, and Junjie Hu. No preference left behind: Group distributional preference optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b57b9aafccf6bf6a76d01079e316e14d-Abstract-Conference.html.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder P. Singh. Reward is enough for convex MDPs. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d7e4cdde82a894b8f633e6d61a01ef15-Abstract.html.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvári, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.

cc/paper/2020/hash/30ee748d38e21392de740e2f9dc686b6-Abstract.html.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating language models for humanlike diversity, 2025.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1651–1661, 2019.

Appendix

CONTENTS

1 Introduction **1**

2 Formulation & Objective **2**

3 Iterative Optimization via Linearization **4**
3.1 The Linearization Reward 4
3.2 Iterative Surrogate Optimization 4

4 Implementing the Surrogate Solve **6**
4.1 Preference-Loss Interface 6
4.2 Constructions at the Three Injection Sites 6

5 Experiments **7**
5.1 Setup 7
5.2 Results across Distribution-Level Objectives 8

A Related Work **16**
A.1 Post-Training and Preference Optimization 16
A.2 Diversity and Creativity in Post-Training 17
A.3 Confidence Calibration 17
A.4 Group Fairness 17
A.5 Distributional Control and Alignment 18
A.6 Optimization of Distributional Functionals 18

B Notation and Terminology **18**
B.1 Mathematical Notation and Conventions 18
B.2 Terminology Glossary 19

C Additional Examples **21**

D Omitted Theoretical Results and Proofs for Sections 2–3 **22**
D.1 Regularity Assumption 22
D.2 Fixed-Reward Representability (Proposition 2) 22
D.3 Linearization Rewards and Approximation Error (Lemma 1) 23
D.4 Surrogate Gaps and First-Order Stationarity (Lemma 2) 24
D.5 Estimating the Linearization Reward (Lemma 3) 25

D.6 Convergence of the Scheme (Proposition 1) 27

E Omitted Theoretical Results and Proofs for Section 4 **28**

E.1 Reward-Based Implementation and Policy Gradients (Proposition 3) 28

E.2 Comparison Distributions and Rules 29

E.3 Population Minimizers of Injected Preference Losses 30

E.3.1 Reference-Tilting Injection 31

E.3.2 Reward-Weighting Injections 31

E.3.3 Comparison-Site Injections 32

E.4 Gradient Effects of the Reference-Tilting Injection (Lemma 8) 34

F Experimental Testbeds and Additional Details **35**

F.1 Experimental Design and Training Protocol 35

F.2 Testbeds, Objectives, and Evaluation Metrics 36

F.2.1 Diversity in Story Generation 36

F.2.2 Confidence Calibration 38

F.2.3 Group Fairness in Occupation Prediction 39

F.3 Checkpoint Selection and Statistical Reporting 42

F.4 Full Results and Supplementary Comparisons 43

F.5 Detailed Implementation and Computational Resources 46

F.5.1 Training and Evaluation Environments 46

F.5.2 Computational Resources 47

F.5.3 Software and Package Versions 49

F.5.4 Configurations and Reproducibility 50

A Related Work

A.1 Post-Training and Preference Optimization

Reinforcement learning from human feedback optimizes a learned response reward while regularizing the policy toward a reference model (Ouyang et al., 2022). DPO expresses the corresponding optimization through a reference-relative preference loss (Rafailov et al., 2023), while the Ψ-preference framework and IPO provide a broader treatment of learning from pairwise preferences (Azar et al., 2023). Our framework builds on these training interfaces: it augments a background individual-level value with a distribution-level objective, then incorporates the latter's linearization reward through reward-based updates or preference-loss injections. For preference optimization, we distinguish modifications to comparisons, weights, and the reference policy while preserving the loss link.

Iterative preference optimization also appears in Nash learning from human feedback and self-play preference optimization (Munos et al., 2023; Wu et al., 2024). These methods seek equilibrium policies for preference games, including settings with non-transitive preferences. Our anchor refresh instead updates the local approximation of a specified scalar distribution-level objective. The resulting surrogate problems retain the background post-training objective and the fixed reference policy.

A.2 Diversity and Creativity in Post-Training

Several methods modify preference training to encourage diverse responses while maintaining response quality. DivPO constructs comparisons using quality gates and diversity-based rankings (Lanchantin et al., 2025). DDPO weights preference losses by the preferred response's deviation from other responses to the same prompt (Chung et al., 2025), and CrPO extends this approach with a weighted combination of diversity, novelty, surprise, and quality signals (Ismayilzada et al., 2025). ReDiPO combines base-model generations, rewriting, quality filtering, and marginal-diversity pair selection to recover diversity in post-trained models (Samuel et al., 2026). These methods motivate the comparison and weight sites in our framework. We generalize these interfaces to signals derived from a specified distribution-level objective, while separately characterizing their effects on population minimizers and training gradients.

Related reward-based methods evaluate collections of responses. DQO uses determinantal point processes to jointly promote semantic diversity and quality (Chen et al., 2026). SetPO assigns trajectories leave-one-out contributions to a kernel-based diversity measure and incorporates these contributions into policy-optimization advantages (Li et al., 2026). SetPO is particularly close to our interpretation of a response through its marginal contribution to a distribution. Our construction derives a linearization reward for a general two-stage feature functional and develops frozen-anchor surrogate optimization and preference-loss implementations across diversity, calibration, and fairness.

A.3 Confidence Calibration

Calibration-oriented post-training uses several forms of supervision and optimization. Band et al. (2024) train long-form generations to support calibrated downstream predictions by their readers. LACIE constructs preference data through a simulated speaker–listener interaction, accounting for whether answers are both correct and accepted (Stengel-Eskin et al., 2024). Leng et al. (2024) address verbalized overconfidence through calibrated reward modeling and reward calculation within PPO. RLCR combines correctness rewards with proper scoring rules, including the Brier score, to improve accuracy and confidence calibration (Damani et al., 2026).

These approaches share our interest in calibrated confidence, but their objectives need not be distribution-level in our formal sense. For example, an expected per-response Brier reward admits a fixed individual-level representation. Our calibration constructions instead penalize nonlinear functions of aggregated confidence–correctness gaps. The distinction concerns the optimized functional: the headline experiments use a squared mean gap within each prompt, whereas confidence-binned reliability aggregates responses across prompts. ECE and AUROC provide complementary evaluations.

A.4 Group Fairness

Equalized odds constrains prediction distributions conditional on the true label and protected group (Hardt et al., 2016). Such criteria naturally involve population statistics. Agarwal et al. (2018) reduce fair classification with conditional-moment constraints to a sequence of cost-sensitive classification problems, illustrating how population requirements can be addressed through standard learning subproblems. Our fairness construction uses a smooth penalty on group-conditional prediction disparities and derives the corresponding linearization reward.

In language applications, counterfactual data augmentation alters training examples to mitigate gender stereotypes (Zmigrod et al., 2019). The Bias in Bios study examines gender disparities in occupation prediction and shows that disparities can remain after explicit gender indicators are scrubbed (De-Arteaga et al., 2019). These approaches motivate our application-specific comparisons. More directly related, FairPO improves summary-level and corpus-level fairness through preference-pair construction and dynamically adjusted pair weights (Li et al., 2025). Our framework places such comparison and weighting mechanisms alongside reference-site injections, with the injected signal derived from an explicit distribution-level objective.

A.5 Distributional Control and Alignment

Khalifa et al. (2021) provide a framework for controlled text generation that combines pointwise and distributional constraints. They formulate these requirements through feature moments, characterize a target distribution by KL projection, and approximate the resulting energy-based model through distributional policy gradients. Their exponential-family construction is closely related to the Gibbs tilt used in our surrogate problems. Our setting jointly optimizes a background value and a nonlinear distribution-level objective; reference tilting implements its frozen-anchor contribution inside a preference loss through a reward-gap margin shift.

Other alignment methods use different distributional targets. AOT encourages first-order stochastic dominance of preferred over dispreferred reward distributions through an optimal-transport formulation (Melnyk et al., 2024). GDPO aligns a model with a group's distribution of beliefs and its belief-conditioned preferences (Yao et al., 2025). These works establish important forms of distributional alignment. Our focus is a specified functional of the policy's response distribution, represented through feature states, and its incorporation into existing post-training methods.

A.6 Optimization of Distributional Functionals

The use of local per-sample signals to optimize a distributional functional has established foundations. Probability functional descent represents first-order variation through an influence function and alternates between estimating this function and improving its expectation (Chu et al., 2019). Our linearization reward is a conditional-response instance of this principle, obtained explicitly by differentiating the two-stage feature-based objective. Related reinforcement-learning formulations optimize nonlinear utilities of policy-induced occupancy measures. Zahavy et al. (2021) analyze the limitations of stationary reward representations and reduce convex MDPs to policy–cost games solved using standard reinforcement-learning subroutines. General-utility policy-gradient methods provide alternative optimization approaches (Zhang et al., 2020; Barakat et al., 2023).

Our iterative scheme is especially connected to conditional-gradient methods. Hazan et al. (2019) optimize functions of state-visitation distributions using approximate reward-based planning oracles. Geist et al. (2022) connect concave utility reinforcement learning to mean-field games and identify fictitious play with Frank–Wolfe optimization. We use a generalized conditional-gradient construction (Bredies et al., 2009), linearizing the distribution-level term while retaining the background value and KL regularization. Our analysis accounts for reward-estimation and inner-optimization errors under the stated policy-class assumptions. The post-training contribution is to connect these surrogate problems to practical training interfaces, including composable preference-loss injections and their population-minimizer and gradient characterizations.

B Notation and Terminology

B.1 Mathematical Notation and Conventions

Spaces, distributions, and expectations. The prompt and response spaces are $\mathcal{X}$ and $\mathcal{Y}$. We write $\Delta(A)$ for the probability distributions on a space A, $\rho \in \Delta(\mathcal{X})$ for the fixed prompt distribution, and $\pi(\cdot \mid x) \in \Delta(\mathcal{Y})$ for a policy's response distribution. For finite A, $\Delta(A) = \{p \in \mathbb{R}^{|A|} : p_a \geq 0,\ \sum_a p_a = 1\}$. The policy class is Π; $\Pi^{\circ}_{\mathrm{tab}}$ denotes all measurable full-support policies on a finite response space. In population statements, policies agreeing at ρ-almost every prompt are identified. The comparison distribution $\mathcal{D}$ is over triples (x, y^w, y^l); the superscripts record the preferred and dispreferred responses. An expectation over $x, y \sim \pi$ abbreviates sampling $x \sim \rho$ and $y \sim \pi(\cdot \mid x)$. Expectations over algorithmic samples are stated separately. A hat denotes an estimate; a superscript $\dagger$ denotes an anchor, and t indexes optimization rounds.

Vectors, derivatives, and norms. Vectors such as $C(x, \pi) \in \mathbb{R}^k$ and $a_x \in \mathbb{R}^k$ are not boldfaced; their dimensions follow from their definitions. We write $\langle u, v \rangle = u^\top v$ and $\|u\|$ for the Euclidean inner product and norm, $\|u\|_1 = \sum_i |u_i|$, and $\|A\|_{\mathrm{op}}$ for the induced Euclidean operator norm. For a differentiable map $F : \mathbb{R}^d \to \mathbb{R}^m$, its Jacobian is $DF(z) \in \mathbb{R}^{m \times d}$, with $[DF(z)]_{ij} = \partial F_i(z)/\partial z_j$. For scalar f, $\nabla f(z)$ is a column vector and $Df(z) = \nabla f(z)^\top$. A subscript specifies

the differentiation variable, as in $D_\theta C$ or $\nabla_\theta \mathcal{L}$; derivatives of Ω_1 and Ω_2 are with respect to their state and summary inputs. The identity map is id, and $\mathbf{1}[\cdot]$ denotes an indicator.

For a response function f and a signed policy difference d, define

$$\langle f, d\rangle_\rho := \mathbb{E}_{x\sim\rho} \sum_{y\in\mathcal{Y}} f(x,y)d(y \mid x), \qquad d_\rho(\pi,\nu) := \mathbb{E}_{x\sim\rho}\|\pi(\cdot \mid x) - \nu(\cdot \mid x)\|_1. \tag{8}$$

The sum is used in the finite-response theoretical setting. The second expression is a metric after identifying policies almost everywhere. Policy mixtures are pointwise: $[(1-\alpha)\pi + \alpha\nu](y \mid x) = (1-\alpha)\pi(y \mid x) + \alpha\nu(y \mid x)$. They are mixtures of response distributions, not averages of model parameters.

Relative entropy. For finite response distributions,

$$\mathrm{KL}(p\|q) = \sum_y p_y \log \frac{p_y}{q_y}, \qquad \mathrm{KL}_\rho(\pi\|\nu) = \mathbb{E}_{x\sim\rho}\big[\mathrm{KL}(\pi(\cdot \mid x)\|\nu(\cdot \mid x))\big].$$

Logarithms are natural, with $0\log(0/q) = 0$ and a positive mass over zero reference mass contributing $+\infty$. The fixed reference π_{ref} has full support. Its prompt-averaged KL can nevertheless be infinite for some policies when ρ has infinite support. Since the reward and distribution-level terms are bounded in our analysis, such policies have $J = -\infty$.

Regularity and first-order conditions. A map F is M-Lipschitz continuous on a domain if $\|F(u) - F(v)\| \le M\|u - v\|$ there. It is L-Lipschitz smooth if it is continuously differentiable and $\|DF(u) - DF(v)\|_{\mathrm{op}} \le L\|u - v\|$. For scalar functions this is Lipschitz continuity of the gradient. The symbols conv and $\overline{\mathrm{conv}}$ denote the convex hull and its closure. Affineness means preserving convex combinations; concavity means $f((1-\alpha)u + \alpha v) \ge (1-\alpha)f(u) + \alpha f(v)$ for $\alpha \in [0,1]$. At a finite-valued policy, the one-sided directional derivative is

$$J'(\pi; \nu - \pi) = \lim_{s\downarrow 0} \frac{J((1-s)\pi + s\nu) - J(\pi)}{s}.$$

We call π first-order stationary if this derivative is nonpositive for every $\nu \in \Pi$ with finite $J(\nu)$. For a differentiable functional represented by the pairing above, this is $\langle \nabla J(\pi), \nu - \pi\rangle_\rho \le 0$. The directional formulation also covers nondifferentiable KL boundaries.

B.2 Terminology Glossary

Table 2 summarizes the terms used in the main text and proofs. Formal assumptions and results are stated in the corresponding sections.

Table 2: Terminology and notation.

Term	Definition	Notation
Policy	Conditional distribution over responses given a prompt.	$\pi(\cdot \mid x)$
Reference policy	Fixed policy used for KL regularization and baseline preference margins.	π_{ref}
Anchor policy	Policy at which the distribution-level objective is linearized; fixed during an inner solve.	$\pi^\dagger$, or π_t
Fixed-reward representation	A bounded, policy-independent reward whose expectation equals the value for every policy in Π.	$V(\pi) = \mathbb{E}_{x,y\sim\pi}[r(x,y)]$
Individual-level value	A value functional admitting a fixed-reward representation.	V; background value $V_{\mathcal{D}}$
Background reward	Fixed per-response reward representing the background individual-level value.	$r_{\mathcal{D}}(x,y)$
Background post-training objective	Background individual-level value minus KL regularization.	$V_{\mathcal{D}}(\pi) - \beta\mathrm{KL}_\rho(\pi\|\pi_{\mathrm{ref}})$

Continued on the next page

Table 2, continued

Term	Definition	Notation
Distribution-level objective	A functional admitting no fixed-reward representation on the policy class. The state-based family can also include affine, representable special cases.	$\Omega(\pi)$
Joint objective	Background post-training objective plus the distribution-level term.	$J(\pi)$
Feature map	Fixed response attributes used to define the distribution-level term.	$\phi(x, y) \in \mathbb{R}^k$
State	Expected feature vector at a prompt under a policy.	$C(x, \pi)$
State-summary function	Map applied to each prompt's state before averaging over prompts.	Ω_1
Pooled summary	Prompt-averaged state summary.	$S(\pi) = \mathbb{E}_x[\Omega_1(C(x, \pi))]$
Summary-score function	Scalar map applied to the pooled summary.	Ω_2
State gradient	Chain-rule coefficient giving the local value of changes to the state at the anchor.	a_x
Linearization reward	Inner product of the state gradient and response features; represents first-order variation at the anchor.	$r_\Omega(x, y; \pi^\dagger)$
Frozen-anchor surrogate problem	Maximize the background post-training objective plus the expected fixed linearization reward.	$\max_{\pi \in \Pi} J_{\pi^\dagger}(\pi)$
Estimated surrogate value	Surrogate value with an estimated linearization reward; expectations remain under the optimized policy and ρ.	$\widehat{J}_{\pi_t}(\pi)$
Surrogate gap	Improvement available from maximizing the exact frozen surrogate relative to its anchor.	$\mathfrak{g}(\pi^\dagger)$
Candidate policy	Approximate solution returned by the estimated inner problem; the star does not imply exact optimality.	π_t^*
Damped update	Convex combination of the current and candidate policies, defining the next anchor.	$\pi_{t+1} = (1 - \alpha_t)\pi_t + \alpha_t \pi_t^*$
Inner-solve tolerance	Permitted suboptimality for the estimated surrogate problem.	ε
Reward-estimation error bound	Uniform surrogate-value error bound obtained from the estimated state gradients.	ζ_t
Reference-relative margin	Difference of the two responses' log policy-to-reference ratios.	$\Delta_{\theta \mid \theta_{\mathrm{ref}}}$
Preference-loss link	Scalar function applied to the reference-relative margin.	ℓ
Injection site	Input modified by a linearization-reward injection: comparisons, weights, or reference.	$\mathcal{D}$, w, θ_{ref}
Linearization-reward injection	Modification of one or more sites using the linearization reward while keeping the link fixed.	$\mathcal{L}[\mathcal{D}', w', \theta'_{\mathrm{ref}}]$
Comparison rule	Pair-selection distribution together with conditional label probabilities.	$\nu_x(e), p_x(e)$
Combined-score injection	Constructs comparisons by ranking candidates using background plus linearization rewards.	$s = r_{\mathcal{D}} + r_\Omega(\cdot\,; \pi^\dagger)$
Task-gated injection	Uses background-reward gates for eligibility and linearization rewards for ranking within each eligible set.	$\mathcal{Y}_x^+, \mathcal{Y}_x^-$
Reward-weighting injection	Constructs positive per-pair weights from the linearization reward.	w'
Reference-tilting injection	Applies a Gibbs tilt of the fixed reference using the linearization reward.	$\widetilde{\pi}_{\mathrm{ref}}$
Comparison coverage	Connectivity of the graph of selected response pairs, almost everywhere in prompts.	G_x
Faithfulness	Equality of the injected population loss's minimizing set and the surrogate optimizer's singleton set in the full-support tabular class.	$\arg\min \mathcal{L} = \{\pi_s\}$
Static schedule	Estimate the linearization reward once and allocate the training budget to that frozen surrogate.	—

Continued on the next page

Table 2, continued

Term	Definition	Notation
Nested schedule	Divide the training budget across rounds, refreshing candidates and the linearization reward between rounds.	—

C Additional Examples

Example 4 (Discrete mode coverage)**.** *Partition each prompt's responses into M fixed modes and let the feature $\phi(x, y) = e_{j(x,y)}$ indicate the mode $j(x, y) \in \{1, \dots, M\}$ of y, so that $C(x, \pi)$ is the coverage vector, whose jth coordinate $C_j(x, \pi)$ is the probability the policy puts on mode j at x. Take $\Omega_1(C(x, \pi)) = \sum_{j=1}^{M} \log\big(C_j(x, \pi) + \delta\big)$ and $\Omega_2 = \mathrm{id}$, giving*

$$\Omega(\pi) = \mathbb{E}_{x\sim\rho}\Big[\textstyle\sum_{j=1}^{M} \log\big(C_j(x, \pi) + \delta\big)\Big]. \tag{9}$$

Each summand is concave and, on the simplex, the sum is maximized at uniform coverage, so maximizing Ω rewards policies that spread their mass evenly over the modes available at each prompt; the small $\delta > 0$ keeps the derivative finite when a mode goes unfilled.

Example 5 (Per-prompt reliability)**.** *Take the scalar feature $\phi(x, y) = c(x, y) - a(x, y)$, so that $C(x, \pi)$ is the prompt's signed calibration gap, positive where the policy is overconfident on x and negative where it is underconfident. Grouping by prompt means scoring that gap prompt by prompt, so take $\Omega_1(C(x, \pi)) = -\frac{\lambda}{2} C(x, \pi)^2$ and $\Omega_2 = \mathrm{id}$, giving*

$$\Omega(\pi) = -\frac{\lambda}{2}\, \mathbb{E}_{x\sim\rho}\big[C(x, \pi)^2\big]. \tag{10}$$

For $\lambda > 0$, the upper bound of zero is attained exactly when $C(x, \pi) = 0$ for ρ-almost every prompt, if such a policy is feasible. Maximizing Ω therefore encourages stated confidence to match prompt-level correctness frequency.

Example 6 (Equalized odds)**.** *Each prompt is a record carrying a gold label $\ell(x)$ and a group annotation $g(x)$, both properties of the prompt alone, and the policy's response is parsed into a predicted label $\hat{\ell}(y)$. Let $s(x) = \pm 1$ separate the two groups, and let $\rho_{o,g} > 0$ be the share of the prompt pool with gold label o in group g. Take the feature $\phi_{o,o'}(x, y) = s(x)\, \mathbf{1}[\ell(x) = o]\, \mathbf{1}[\hat{\ell}(y) = o'] \,/\, \rho_{o,g(x)}$, which is nonzero only in the row of the prompt's own gold label; dividing by $\rho_{o,g}$ turns an unconditional average into a conditional rate. With $\Omega_1 = \mathrm{id}$, the pooled coordinate $\mathbb{E}_x[C(x, \pi)]_{o,o'}$ is then the difference between the two groups' rates of answering o' on a prompt whose gold label is o, and its diagonal is the equal-opportunity gap. Take $\Omega_2(C) = -\frac{\lambda}{2} \sum_{o,o'} w_{o,o'} C_{o,o'}^2$ with weights $w_{o,o'} > 0$, whose upper bound of zero, for $\lambda > 0$, is attained exactly when the two groups' prediction distributions agree within every gold label, if feasible. Because $\rho_{o,g}$ depends on the prompt distribution alone, the state remains linear in the policy.*

Example 7 (Teacher target)**.** *Let $\pi^\star$ be a target policy one can sample but need not be able to score, a teacher, a human corpus, or the model's own pre-alignment base, and let $e(x, y)$ be a policy-independent feature such as an embedding. Take the feature centered on the target's own average, $\phi(x, y) = e(x, y) - \mathbb{E}_{y'\sim\pi^\star(\cdot|x)}[e(x, y')]$, which is again policy-independent, so that $C(x, \pi)$ is the gap between the policy's average feature at x and the target's. Take $\Omega_1(C(x, \pi)) = -\frac{\lambda}{2}\|C(x, \pi)\|^2$ and $\Omega_2 = \mathrm{id}$, giving*

$$\Omega(\pi) = -\frac{\lambda}{2}\, \mathbb{E}_{x\sim\rho}\big\|C(x, \pi)\big\|^2, \tag{11}$$

the same form as Example 1, now measuring distance to the target rather than to the origin. Maximizing Ω pulls the policy's average feature toward the teacher's at every prompt, and needs only samples from $\pi^\star$, never a closed form for what makes its outputs desirable.

Non-affineness of the examples. For each example, non-affineness follows whenever Π contains a policy segment along which the objective is strictly concave. The conditions below identify such segments and exclude degenerate cases in which the relevant states cannot vary.

For continuous diversity (Example 1), per-prompt reliability (Example 5), and teacher matching (Example 7), the objective has the form $\Omega(\pi) = -\frac{\lambda}{2}\mathbb{E}_{x\sim\rho}\|C(x,\pi)\|^2$, with $\lambda > 0$. Along a feasible segment $\pi_t = (1-t)\pi_0 + t\pi_1$,

$$\frac{d^2}{dt^2}\Omega(\pi_t) = -\lambda\,\mathbb{E}_{x\sim\rho}\|C(x,\pi_1) - C(x,\pi_0)\|^2.$$

Hence these objectives are non-affine whenever the endpoint states differ on a set of prompts of positive ρ-measure. For discrete mode coverage (Example 4), the same condition suffices, since $\sum_j \log(C_j + \delta)$ is strictly concave in the coverage vector for $\delta > 0$.

For confidence-binned reliability (Example 2), write $(P(t), D(t))$ for the pooled state along π_t. Its coordinates are affine in t, and

$$\frac{d^2}{dt^2}\Omega(\pi_t) = -\lambda \sum_m \frac{\big((P_m(t)+\delta)\dot{D}_m - D_m(t)\dot{P}_m\big)^2}{(P_m(t)+\delta)^3}.$$

This is strictly negative if the endpoints have different ratios $D_m/(P_m + \delta)$ in at least one band. Thus variation in at least one band's smoothed confidence gap suffices for non-affineness.

For equalized odds (Example 6), assume that the group–label probabilities used to normalize the features are positive. Writing $S(\pi) = \mathbb{E}_{x\sim\rho}[C(x,\pi)]$ for the pooled vector of group-conditional prediction-rate differences, we have

$$\frac{d^2}{dt^2}\Omega(\pi_t) = -\lambda \sum_{o,o'} w_{o,o'}\big(S_{o,o'}(\pi_1) - S_{o,o'}(\pi_0)\big)^2.$$

With $\lambda > 0$ and positive weights, this is strictly negative whenever the endpoint pooled vectors differ.

Under the respective conditions above, each objective fails to be affine on the reachable state profiles. Consequently, it admits no fixed per-response reward representation on Π: the expectation of any fixed reward is affine along every policy segment.

D Omitted Theoretical Results and Proofs for Sections 2–3

D.1 Regularity Assumption

Assumption 2 (Regularity)**.** *(i) The response space $\mathcal{Y}$ is finite, the feature map is measurable, and $\sup_{x,y}\|\phi(x,y)\| \le B < \infty$. Define the state domain and summary domain by*

$$\mathcal{C} = \overline{\bigcup_{x\in\mathcal{X}} \operatorname{conv}\{\phi(x,y) : y \in \mathcal{Y}\}}, \qquad \mathcal{S} = \overline{\operatorname{conv}}\,\Omega_1(\mathcal{C}).$$

(ii) On an open convex neighborhood of $\mathcal{C}$, Ω_1 is M_1-Lipschitz continuous and L_1-Lipschitz smooth. (iii) On an open convex neighborhood of $\mathcal{S}$, Ω_2 is M_2-Lipschitz continuous and L_2-Lipschitz smooth. Norms and smoothness use the Euclidean and induced operator norms defined in Appendix B.

The domains include population and empirical states, their same-prompt mixtures, and pooled summaries. We write $L_\Omega = M_2L_1 + M_1^2L_2$ for the constant controlling the quadratic remainder in the state-profile norm $(\mathbb{E}_x\|C(x,\pi) - C(x,\nu)\|^2)^{1/2}$ below.

D.2 Fixed-Reward Representability (Proposition 2)

Proposition 2 (Individual-level exactly when affine in the states)**.** *Let Π be a nonempty convex class of measurable policies, let ρ have finite support, and let ϕ be bounded. For Ω of the form eq.* (2)*, Ω is individual-level if and only if there are vectors $a_x \in \mathbb{R}^k$ and scalars b_x such that*

$$\Omega(\pi) = \mathbb{E}_{x\sim\rho}[\langle a_x, C(x,\pi)\rangle + b_x] \qquad \textit{for every } \pi \in \Pi. \tag{12}$$

In this case, $r(x,y) = \langle a_x, \phi(x,y)\rangle + b_x$ on $\operatorname{supp}\rho$, extended by zero elsewhere, is a bounded reward representing Ω.

Proof. If eq. (12) holds, linearity of expectation gives

$$\begin{aligned}\Omega(\pi) &= \mathbb{E}_x[\langle a_x, \mathbb{E}_{y\sim\pi(\cdot|x)}\phi(x,y)\rangle + b_x] \\ &= \mathbb{E}_{x,\, y\sim\pi}[\langle a_x, \phi(x,y)\rangle + b_x].\end{aligned}$$

The reward is bounded because ϕ is bounded and $\operatorname{supp}\rho$ is finite.

Conversely, suppose a fixed reward represents Ω. Write $\operatorname{supp}\rho = \{x_1, \dots, x_N\}$ with masses $\rho_i > 0$, and define $K(\pi) = (C(x_i,\pi))_{i=1}^N$ and $\mathcal{P} = K(\Pi)$. Since K is affine and Π is convex, $\mathcal{P}$ is convex. The state-based form of Ω defines a function $\widehat{\Omega}$ on $\mathcal{P}$ with $\Omega = \widehat{\Omega}\circ K$. For $p_i = K(\pi_i)$ and $t \in [0,1]$, the fixed-reward representation gives

$$\begin{aligned}\widehat{\Omega}((1-t)p_0 + tp_1) &= \Omega((1-t)\pi_0 + t\pi_1) \\ &= (1-t)\Omega(\pi_0) + t\Omega(\pi_1) \\ &= (1-t)\widehat{\Omega}(p_0) + t\widehat{\Omega}(p_1).\end{aligned}$$

An affine function on a convex subset of a finite-dimensional space extends to its affine hull and then to the ambient space. Thus $\widehat{\Omega}(p) = \sum_i \langle c_i, p_i\rangle + d$ for some $c_i \in \mathbb{R}^k$ and $d \in \mathbb{R}$. Taking $a_{x_i} = c_i/\rho_i$ and $b_{x_i} = d$ proves eq. (12). □

D.3 Linearization Rewards and Approximation Error (Lemma 1)

Lemma 1 (The linearization reward is the functional gradient)**.** *Let Assumption 2 hold and fix an anchor $\pi^\dagger \in \Pi$.*

1. *For every $\pi \in \Pi$ the map $s \mapsto \Omega\big((1-s)\pi^\dagger + s\pi\big)$ is right-differentiable at $s = 0$, with*

$$\frac{d}{ds}\,\Omega\big((1-s)\pi^\dagger + s\pi\big)\Big|_{s=0^+} = \big\langle r_\Omega(\cdot\,;\pi^\dagger),\ \pi - \pi^\dagger\big\rangle_\rho, \tag{13}$$

   and, when Π is the full tabular class, $r_\Omega(\cdot\,;\pi^\dagger)$ is the only per-response function with this property, up to an additive function of the prompt alone, at every prompt to which ρ assigns positive mass.

2. *The remainder in eq.* (4) *satisfies $R(\pi^\dagger,\pi^\dagger) = 0$ and $\big|R(\pi,\pi^\dagger)\big| \le \frac{L_\Omega}{2}\,\mathbb{E}_x\big\|C(x,\pi) - C(x,\pi^\dagger)\big\|^2$, where $L_\Omega := M_2L_1 + M_1^2L_2$. If moreover Ω is concave in π on Π, then $R(\pi,\pi^\dagger) \le 0$ for every π.*

Proof. Fix $\pi, \pi^\dagger \in \Pi$, and write $\Delta_x = C(x,\pi) - C(x,\pi^\dagger)$ and $S(\nu) = \mathbb{E}_x[\Omega_1(C(x,\nu))]$. The state and summary segments lie in the domains of Assumption 2. The definition of the reward gives

$$\begin{aligned}\mathbb{E}_x\langle a_x, \Delta_x\rangle &= \mathbb{E}_x \sum_y \langle a_x, \phi(x,y)\rangle[\pi(y \mid x) - \pi^\dagger(y \mid x)] \\ &= \langle r_\Omega(\cdot\,;\pi^\dagger), \pi - \pi^\dagger\rangle_\rho.\end{aligned} \tag{14}$$

Taylor expansion of the two stages yields

$$\begin{aligned}S(\pi) - S(\pi^\dagger) &= \mathbb{E}_x[D\Omega_1(C(x,\pi^\dagger))\Delta_x] + \mathbb{E}_x e_x, & \|e_x\| &\le \tfrac{L_1}{2}\|\Delta_x\|^2, \\ \Omega(\pi) - \Omega(\pi^\dagger) &= \langle \nabla\Omega_2(S(\pi^\dagger)), S(\pi) - S(\pi^\dagger)\rangle + e', & |e'| &\le \tfrac{L_2}{2}\|S(\pi) - S(\pi^\dagger)\|^2.\end{aligned} \tag{15}$$

Combining these expansions with eq. (14),

$$\begin{aligned}R(\pi,\pi^\dagger) &= \langle \nabla\Omega_2(S(\pi^\dagger)), \mathbb{E}_x e_x\rangle + e', \\ \|S(\pi) - S(\pi^\dagger)\|^2 &\le M_1^2(\mathbb{E}_x\|\Delta_x\|)^2 \le M_1^2\mathbb{E}_x\|\Delta_x\|^2, \\ |R(\pi,\pi^\dagger)| &\le \tfrac{M_2L_1}{2}\mathbb{E}_x\|\Delta_x\|^2 + \tfrac{L_2}{2}\|S(\pi) - S(\pi^\dagger)\|^2 \\ &\le \tfrac{L_\Omega}{2}\mathbb{E}_x\|\Delta_x\|^2.\end{aligned}$$

In particular, $R(\pi^\dagger,\pi^\dagger) = 0$. For $\pi_s = (1-s)\pi^\dagger + s\pi$, linearity of the states gives

$$\begin{aligned}\Omega(\pi_s) - \Omega(\pi^\dagger) &= s\langle r_\Omega(\cdot\,;\pi^\dagger), \pi - \pi^\dagger\rangle_\rho + R(\pi_s,\pi^\dagger), \\ |R(\pi_s,\pi^\dagger)| &\le \tfrac{L_\Omega s^2}{2}\mathbb{E}_x\|\Delta_x\|^2 \le 2L_\Omega B^2 s^2.\end{aligned}$$

Dividing by s and letting $s \downarrow 0$ proves part (i). The state-based formula defines Ω on these mixtures even when Π itself is not convex.

For uniqueness, let h be the difference between two representing functions in the full tabular class. For a prompt with $\rho(\{x\}) > 0$, compare the anchor with the policy equal to δ_y at x and unchanged elsewhere. Then

$$0 = \rho(\{x\}) \left[h(x, y) - \mathbb{E}_{y' \sim \pi^\dagger(\cdot|x)} h(x, y')\right] \qquad \text{for every } y,$$

so $h(x, \cdot)$ is constant. Conversely, prompt-only functions pair to zero with every policy difference. Finally, if Ω is concave along the segment, its tangent bound gives

$$\Omega(\pi) - \Omega(\pi^\dagger) \le \frac{d}{ds} \Omega(\pi_s)\Big|_{s=0^+} = \langle r_\Omega(\cdot\,; \pi^\dagger), \pi - \pi^\dagger \rangle_\rho,$$

which proves $R(\pi, \pi^\dagger) \le 0$. □

D.4 Surrogate Gaps and First-Order Stationarity (Lemma 2)

Well-posedness and stationarity. Policies are identified ρ-almost everywhere. Under Assumption 1(i), compactness uses the metric d_ρ defined there. Bounded reward expectations are continuous in this metric. Moreover,

$$|\Omega(\pi) - \Omega(\nu)| \le M_1 M_2 \mathbb{E}_x \|C(x, \pi) - C(x, \nu)\| \le M_1 M_2 B \, d_\rho(\pi, \nu).$$

Prompt-averaged KL is lower semicontinuous: a subsequence attaining the liminf along any d_ρ-convergent sequence has a further subsequence converging pointwise almost everywhere; continuity of finite-response KL and Fatou's lemma then give the claim. Thus J and every exact frozen surrogate are upper semicontinuous and attain their maxima on Π whenever a finite-valued feasible policy exists. The same argument applies to estimated rewards dominated by an integrable prompt-dependent envelope, using dominated convergence for their expectations.

At a finite-valued anchor, first-order stationarity means $J'(\pi^\dagger; \pi - \pi^\dagger) \le 0$ for every $\pi \in \Pi$ with finite $J(\pi)$, where the derivative is one-sided. This agrees with the main-text definition and accommodates boundary policies.

Lemma 2 (The gap is the stationarity condition)**.** *Under Assumption 2 and Assumption 1(i), let $J(\pi^\dagger) > -\infty$. Then $\mathfrak{g}(\pi^\dagger) = 0$ if and only if $J'(\pi^\dagger; \pi - \pi^\dagger) \le 0$ for every $\pi \in \Pi$ with finite $J(\pi)$.*

Proof. The frozen surrogate is concave: its reward terms are linear and the negative KL term is concave. By eq. (4),

$$\begin{aligned} J(\pi) &= J_{\pi^\dagger}(\pi) + c(\pi^\dagger) + R(\pi, \pi^\dagger), \\ c(\pi^\dagger) &= \Omega(\pi^\dagger) - \langle r_\Omega(\cdot\,; \pi^\dagger), \pi^\dagger \rangle_\rho. \end{aligned} \tag{16}$$

For a finite-valued π, the segment $\pi_s = (1 - s)\pi^\dagger + s\pi$ has finite KL by convexity. Hence

$$\begin{aligned} \frac{J(\pi_s) - J(\pi^\dagger)}{s} &= \frac{J_{\pi^\dagger}(\pi_s) - J_{\pi^\dagger}(\pi^\dagger)}{s} + \frac{R(\pi_s, \pi^\dagger)}{s}, \\ \left|\frac{R(\pi_s, \pi^\dagger)}{s}\right| &\le 2 L_\Omega B^2 s \longrightarrow 0. \end{aligned}$$

The surrogate difference quotient is non-increasing in s and bounded below by its finite value at $s = 1$. Its limit therefore exists in $(-\infty, +\infty]$, and $J'(\pi^\dagger; \pi - \pi^\dagger) = J'_{\pi^\dagger}(\pi^\dagger; \pi - \pi^\dagger)$. Concavity also gives

$$J_{\pi^\dagger}(\pi) - J_{\pi^\dagger}(\pi^\dagger) \le J'_{\pi^\dagger}(\pi^\dagger; \pi - \pi^\dagger).$$

Thus nonpositive directional derivatives imply that the anchor maximizes the surrogate. Conversely, at a surrogate maximizer every difference quotient is nonpositive. Policies with infinite KL have surrogate value $-\infty$ and cannot improve on the anchor. This proves the equivalence. □

D.5 Estimating the Linearization Reward (Lemma 3)

Fix a round t and condition on the anchor π_t. Write $C_x = C(x, \pi_t)$ and $S = \mathbb{E}_{x\sim\rho}[\Omega_1(C_x)]$. The linearization reward is $r_\Omega(x, y; \pi_t) = \langle a_x, \phi(x, y)\rangle$, where $a_x = D\Omega_1(C_x)^\top \nabla\Omega_2(S)$. We estimate this reward by replacing the local states and pooled summary with sample averages.

Sampling construction. For each prompt x requiring a local state estimate, draw k independent responses $y_{x,i} \sim \pi_t(\cdot \mid x)$ and set $\widehat{C}_x = \frac{1}{k}\sum_{i=1}^k \phi(x, y_{x,i})$. To estimate the pooled summary, independently across $j = 1, \dots, n$, draw $x_j \sim \rho$ and then k conditionally independent responses $y_{j,i} \sim \pi_t(\cdot \mid x_j)$. Define

$$\widehat{C}_j = \frac{1}{k}\sum_{i=1}^k \phi(x_j, y_{j,i}), \qquad \widehat{S} = \frac{1}{n}\sum_{j=1}^n \Omega_1(\widehat{C}_j).$$

The estimated state gradient and linearization reward are then

$$\widehat{a}_x = D\Omega_1(\widehat{C}_x)^\top \nabla\Omega_2(\widehat{S}), \qquad \widehat{r}_\Omega(x, y) = \langle \widehat{a}_x, \phi(x, y)\rangle.$$

Here $\widehat{C}_x$ denotes a local estimate at an arbitrary prompt, whereas $\widehat{C}_j$ denotes the estimate from pooled batch j. Independence between the local estimates $\{\widehat{C}_x\}$ and the pooled batches is not required. The empirical states belong to $\mathcal{C}$, and $\widehat{S}$ belongs to $\mathcal{S} = \overline{\mathrm{conv}}\,\Omega_1(\mathcal{C})$, so the bounds in Assumption 2 apply.

Prompt coverage and special cases. For the population surrogate problem, the estimated reward must be a measurable function defined for ρ-almost every prompt. In the general construction, this requires local estimates $\widehat{C}_x$ for ρ-almost every x; the n pooled batches alone do not supply estimates at other prompts. When ρ has finite support, one construction draws k responses at every prompt in that support. For a general prompt distribution, the existence of a measurable family of local estimates with the stated sampling properties is an assumption of the bound below.

The two identity-map cases simplify the construction. If $\Omega_2 = \mathrm{id}$, only local estimates are needed, and $\widehat{a}_x = \nabla\Omega_1(\widehat{C}_x)$. If $\Omega_1 = \mathrm{id}$, only the pooled batches are needed: $\widehat{S} = \frac{1}{n}\sum_{j=1}^n \widehat{C}_j$ and $\widehat{a}_x = \nabla\Omega_2(\widehat{S})$ is shared across prompts.

From reward estimation to surrogate accuracy. Replacing $r_\Omega(\cdot\,; \pi_t)$ with $\widehat{r}_\Omega$ in eq. (5) defines the estimated surrogate value $\widehat{J}_{\pi_t}$. Conditional on the reward estimate, its expectations remain over $x \sim \rho$ and $y \sim \pi(\cdot \mid x)$, where π is the policy being optimized. Thus the anchor samples estimate the reward; they do not replace the surrogate problem by a fixed empirical training loss. The following lemma bounds the resulting error uniformly over policies with finite KL divergence and translates it into an inner-solve accuracy bound.

Lemma 3 (Estimation error in the surrogate)**.** *Let Assumption 2 hold and fix an anchor π_t. Write $C_x = C(x, \pi_t)$, $S = \mathbb{E}_{x\sim\rho}[\Omega_1(C_x)]$, and $a_x = D\Omega_1(C_x)^\top \nabla\Omega_2(S)$. Let $\{\widehat{a}_x\}$ be a measurable family of estimates defined for ρ-almost every prompt, and put $\widehat{r}_\Omega(x, y) = \langle \widehat{a}_x, \phi(x, y)\rangle$ and $\zeta = B\,\mathbb{E}_{x\sim\rho}\|\widehat{a}_x - a_x\| < \infty$. Assume that Π contains a policy with finite KL divergence.*

1. *For every policy with finite $\mathrm{KL}_\rho(\pi\|\pi_{\mathrm{ref}})$,*

$$|\widehat{J}_{\pi_t}(\pi) - J_{\pi_t}(\pi)| \le \zeta.$$

Consequently, every ε-optimal solution π_t^* of $\widehat{J}_{\pi_t}$ satisfies

$$J_{\pi_t}(\pi_t^*) \ge \sup_{\pi\in\Pi} J_{\pi_t}(\pi) - \varepsilon - 2\zeta.$$

2. *Use the sampling construction above with k responses per prompt and n independent pooled batches, including a measurable local estimate for ρ-almost every prompt. Then*

$$\mathbb{E}[\zeta] \le \frac{B^2 L_\Omega}{\sqrt{k}} + \frac{B^2 M_1^2 L_2}{\sqrt{n}}.$$

Independence between the local estimates $\{\widehat{C}_x\}$ and the pooled batches is not required.

3. *If* $\Omega_2 = \mathrm{id}$*, the local estimates alone suffice and* $\mathbb{E}[\zeta] \le B^2 L_\Omega/\sqrt{k}$*. If* $\Omega_1 = \mathrm{id}$*, the pooled batches alone suffice and*

$$\mathbb{E}[\zeta] \le B^2 L_\Omega \left(\frac{1}{\sqrt{nk}} + \frac{1}{\sqrt{n}}\right) \le \frac{2B^2 L_\Omega}{\sqrt{n}}.$$

Proof. For part (i), the two surrogates differ only in their linearization-reward terms. For every policy with finite relative entropy,

$$\begin{aligned} |\widehat{J}_{\pi_t}(\pi) - J_{\pi_t}(\pi)| &= \left|\mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\langle \widehat{a}_x - a_x, \phi(x,y)\rangle\right| \\ &\le B\, \mathbb{E}_{x\sim\rho}\|\widehat{a}_x - a_x\| = \zeta. \end{aligned}$$

Thus, for every such $\pi \in \Pi$,

$$\begin{aligned} J_{\pi_t}(\pi_t^*) &\ge \widehat{J}_{\pi_t}(\pi_t^*) - \zeta \\ &\ge \widehat{J}_{\pi_t}(\pi) - \varepsilon - \zeta \\ &\ge J_{\pi_t}(\pi) - \varepsilon - 2\zeta. \end{aligned}$$

Taking the supremum over π proves the claim. Policies with infinite relative entropy have value $-\infty$ under both surrogates and do not affect the comparison, provided a finite-valued feasible policy exists.

For part (ii), adding and subtracting $D\Omega_1(C_x)^\top \nabla\Omega_2(\widehat{S})$ gives

$$\|\widehat{a}_x - a_x\| \le M_2 L_1 \|\widehat{C}_x - C_x\| + M_1 L_2 \|\widehat{S} - S\|.$$

The bounds apply because the empirical states and summary lie in the domains of Assumption 2. Since $\|\phi\| \le B$, an average of k independent responses satisfies

$$\mathbb{E}\|\widehat{C}_x - C_x\|^2 = \frac{1}{k}\left(\mathbb{E}_{y\sim\pi_t(\cdot|x)}\|\phi(x,y)\|^2 - \|C_x\|^2\right) \le \frac{B^2}{k}.$$

Hence $\mathbb{E}\|\widehat{C}_x - C_x\| \le B/\sqrt{k}$.

To bound the summary error, decompose

$$\begin{aligned} \widehat{S} - S = \frac{1}{n}\sum_{j=1}^{n}\left[\Omega_1(\widehat{C}_j) - \Omega_1(C_{x_j})\right] \\ + \frac{1}{n}\sum_{j=1}^{n}\Omega_1(C_{x_j}) - \mathbb{E}_x[\Omega_1(C_x)]. \end{aligned}$$

The expected norm of the first term is at most $M_1 B/\sqrt{k}$. For the second, let $Z = \Omega_1(C_X)$ with $X \sim \rho$, and let Z' and $C_{X'}$ be independent copies. Then

$$\begin{aligned} \mathbb{E}\|Z - \mathbb{E}Z\|^2 &= \tfrac{1}{2}\mathbb{E}\|Z - Z'\|^2 \\ &\le \tfrac{M_1^2}{2}\mathbb{E}\|C_X - C_{X'}\|^2 \\ &= M_1^2 \mathbb{E}\|C_X - \mathbb{E}C_X\|^2 \le M_1^2 B^2. \end{aligned}$$

Independence of the sampled prompts therefore gives

$$\mathbb{E}\|\widehat{S} - S\| \le \frac{M_1 B}{\sqrt{k}} + \frac{M_1 B}{\sqrt{n}}.$$

Substituting into the state-gradient bound and averaging over x yields

$$\begin{aligned} \mathbb{E}[\zeta] &\le \frac{B^2 M_2 L_1}{\sqrt{k}} + B^2 M_1^2 L_2 \left(\frac{1}{\sqrt{k}} + \frac{1}{\sqrt{n}}\right) \\ &= \frac{B^2 L_\Omega}{\sqrt{k}} + \frac{B^2 M_1^2 L_2}{\sqrt{n}}. \end{aligned}$$

For part (iii), when $\Omega_2 = \mathrm{id}$, the summary-error term vanishes and $L_\Omega = L_1$, giving the stated local-state bound.

When $\Omega_1 = \mathrm{id}$, $\widehat{a}_x = \nabla\Omega_2(\widehat{S})$ is shared across prompts and $\zeta \le BL_\Omega\|\widehat{S} - S\|$. In this case,

$$\widehat{S} - S = \frac{1}{n}\sum_{j=1}^{n}(\widehat{C}_j - C_{x_j}) + \frac{1}{n}\sum_{j=1}^{n} C_{x_j} - \mathbb{E}_x C_x.$$

Conditional on the prompts, the first sum has independent, mean-zero summands, so its expected squared norm is at most $B^2/(nk)$. The second sum has expected squared norm at most B^2/n. Taking expected norms gives

$$\mathbb{E}\|\widehat{S} - S\| \le \frac{B}{\sqrt{nk}} + \frac{B}{\sqrt{n}},$$

which proves the remaining bound. □

The uniform deviation in part (i) is needed because the inner solution is selected using the estimated reward; unbiasedness alone does not control its optimization error. For the general two-stage objective, nonlinear evaluation of Ω_1 at an empirical local state can introduce bias into the pooled summary. The $k^{-1/2}$ term controls this effect without requiring additional smoothness assumptions. When $\Omega_1 = \mathrm{id}$, the local estimation errors are conditionally mean-zero and average across the pooled batches, yielding the sharper bound in part (iii).

D.6 Convergence of the Scheme (Proposition 1)

Proof. Write $\kappa = L_\Omega\Delta^2$ and $\eta_t = \varepsilon + 2\zeta_t$. The initialization has finite KL. Each estimated inner solve returns a finite-valued policy, and convexity of KL then keeps every iterate finite-valued. The following inequalities hold for each realization.

One-round improvement. Using eq. (16), concavity of the frozen surrogate, and Lemma 3(i),

$$\begin{aligned} J(\pi_{t+1}) - J(\pi_t) &= J_{\pi_t}(\pi_{t+1}) - J_{\pi_t}(\pi_t) + R(\pi_{t+1}, \pi_t) \\ &\ge \alpha_t[J_{\pi_t}(\pi_t^*) - J_{\pi_t}(\pi_t)] - \tfrac{\kappa}{2}\alpha_t^2 \\ &\ge \alpha_t[\mathfrak{g}(\pi_t) - \eta_t] - \tfrac{\kappa}{2}\alpha_t^2. \end{aligned} \tag{17}$$

Here Lemma 1(ii) bounds the remainder, since $\|C(x, \pi_{t+1}) - C(x, \pi_t)\| \le \alpha_t\Delta$.

Part (i). For a constant step $\alpha > 0$, summing eq. (17) gives

$$\begin{aligned} \min_{t<T}\mathfrak{g}(\pi_t) &\le \frac{1}{T}\sum_{t<T}\mathfrak{g}(\pi_t) \\ &\le \frac{J(\pi_T) - J(\pi_0)}{\alpha T} + \frac{\kappa\alpha}{2} + \frac{1}{T}\sum_{t<T}\eta_t \\ &\le \frac{\Delta_J}{\alpha T} + \frac{\kappa\alpha}{2} + \frac{1}{T}\sum_{t<T}\eta_t. \end{aligned}$$

If $\zeta_t \le \zeta$, the last average is at most $\bar{\varepsilon} := \varepsilon + 2\zeta$. For the stated step size, the remaining terms satisfy

$$\begin{aligned} &\frac{\Delta_J}{\alpha T} + \frac{\kappa\alpha}{2} = \sqrt{\frac{2\Delta_J\kappa}{T}} \le \frac{\max\{2\Delta_J, \kappa\}}{\sqrt{T}} && \text{if } \alpha = \sqrt{\frac{2\Delta_J}{\kappa T}} \le 1, \\ &\frac{\Delta_J}{T} + \frac{\kappa}{2} < \frac{2\Delta_J}{T} \le \frac{2\Delta_J}{\sqrt{T}} && \text{otherwise.} \end{aligned}$$

Part (ii). Let $h_t = J(\pi^\star) - J(\pi_t)$. Concavity of Ω gives

$$\begin{aligned} \mathfrak{g}(\pi_t) \ge J_{\pi_t}(\pi^\star) - J_{\pi_t}(\pi_t) = h_t - R(\pi^\star, \pi_t) \ge h_t, \\ h_{t+1} \le (1 - \alpha_t)h_t + \alpha_t\eta_t + \tfrac{\kappa}{2}\alpha_t^2. \end{aligned}$$

Under $\eta_t \le \bar{\varepsilon}$, the step $\alpha_0 = 1$ yields $h_1 \le \bar{\varepsilon} + \kappa/2 \le \bar{\varepsilon} + 2\kappa/3$. For $t \ge 1$, induction with $\alpha_t = 2/(t+2)$ gives

$$\begin{aligned} h_{t+1} &\le \frac{t}{t+2}\left(\frac{2\kappa}{t+2} + \bar{\varepsilon}\right) + \frac{2\bar{\varepsilon}}{t+2} + \frac{2\kappa}{(t+2)^2} \\ &= \frac{2\kappa(t+1)}{(t+2)^2} + \bar{\varepsilon} \le \frac{2\kappa}{t+3} + \bar{\varepsilon}. \end{aligned}$$

Since $2\kappa = 8L_\Omega B^2$, this is the claimed bound.

Expected estimation error. If only $\mathbb{E}[\zeta_t] \le \zeta$ is assumed, take expectations in the pathwise inequalities above. Part (i) gives

$$\mathbb{E}\left[\min_{t<T} \mathfrak{g}(\pi_t)\right] \le \frac{\Delta_J}{\alpha T} + \frac{\kappa\alpha}{2} + \frac{1}{T}\sum_{t<T}\mathbb{E}[\eta_t] \le \frac{\Delta_J}{\alpha T} + \frac{\kappa\alpha}{2} + \bar{\varepsilon}.$$

For part (ii), $\mathbb{E}[h_t]$ satisfies the same recursion with $\mathbb{E}[\eta_t] \le \bar{\varepsilon}$, so the same induction applies. Integrability follows from eq. (17), finite $J(\pi_0)$, and integrable η_t, since J is bounded above. No independence across rounds is required. □

Under the sampling conditions of Lemma 3, applied conditionally on the history before round t, the bound is uniform in the random anchor. With k local responses and n pooled batches, the expected-error version therefore applies with $\zeta = B^2 L_\Omega/\sqrt{k} + B^2 M_1^2 L_2/\sqrt{n}$; the identity-map cases use the sharper bounds in that lemma. These are expectation guarantees, rather than simultaneous high-probability bounds on the realized errors.

E Omitted Theoretical Results and Proofs for Section 4

E.1 Reward-Based Implementation and Policy Gradients (Proposition 3)

Implementing the surrogate problem. Reward-based methods, including RLHF and RLVR, use scalar rewards assigned to generated responses to construct policy updates. At a fixed anchor $\pi^\dagger$, replacing the background reward $r_\mathcal{D}$ with $r_\mathcal{D} + r_\Omega(\cdot\,;\pi^\dagger)$ while retaining KL regularization gives the frozen-anchor surrogate problem eq. (5). The linearization reward is held fixed throughout the inner solve, while the reference policy π_{ref} remains fixed across rounds. Using an estimated reward $\widehat{r}_\Omega(\cdot\,;\pi^\dagger)$ instead gives the estimated surrogate problem, whose reward-estimation error is controlled by Lemma 3. This substitution specifies the reward supplied to the training procedure; the accuracy of the resulting surrogate solve depends on that procedure.

Policy-gradient identity. When the anchor equals the current policy, the surrogate and the joint objective J have the same parameter gradient. Under the conditions of Proposition 3, this gradient can be expressed using the summed reward $r_\mathcal{D} + r_\Omega(\cdot\,;\pi_\theta)$ and the KL term. The linearization reward is evaluated at π_θ but held fixed when forming the gradient estimator; no differentiation through the reward is required. The following proposition states this identity and the conditions justifying differentiation under the expectation.

Proposition 3 (Policy gradient with the linearization reward)**.** *Let Assumption 2 hold and let $\{\pi_\theta\}_{\theta\in\Theta}$ be a family of policies with full support on $\mathcal{Y}$ at every prompt, continuously differentiable in θ on the open set Θ, with $\sup_{x,y}\|\nabla_\theta \log \pi_\theta(y \mid x)\|$ and $\sup_{x,y}\left|\log\big(\pi_\theta(y \mid x)/\pi_{\mathrm{ref}}(y \mid x)\big)\right|$ bounded on compact subsets of Θ. Then at every $\theta \in \Theta$, with the linearization reward read at the anchor $\pi^\dagger = \pi_\theta$,*

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{x\sim\rho,\, y\sim\pi_\theta(\cdot|x)}\left[\left(r_\mathcal{D}(x,y) + r_\Omega(x,y;\pi_\theta) - \beta\log\tfrac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right)\nabla_\theta \log \pi_\theta(y \mid x)\right]. \quad (18)$$

Proof. Fix θ and a compact neighborhood contained in Θ. The hypotheses uniformly bound the score and log ratio there; $r_\mathcal{D}$ is bounded and $|r_\Omega(x,y;\pi_\theta)| \le M_1 M_2 B$. These bounds, together with bounded derivatives of the two state maps, provide integrable envelopes for the derivatives below. Thus dominated convergence permits differentiation under $\mathbb{E}_x$.

Write $S(\theta) = \mathbb{E}_x[\Omega_1(C(x, \pi_\theta))]$. For the distribution-level term, the chain rule and score identity give

$$\begin{aligned} D_\theta C(x, \pi_\theta) &= \sum_y \phi(x, y) \nabla_\theta \pi_\theta(y \mid x)^\top \\ &= \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[\phi(x, y) \nabla_\theta \log \pi_\theta(y \mid x)^\top], \\ \nabla_\theta \Omega(\pi_\theta) &= (D_\theta S)^\top \nabla \Omega_2(S(\theta)) \\ &= \mathbb{E}_x[D_\theta C(x, \pi_\theta)^\top D\Omega_1(C(x, \pi_\theta))^\top \nabla \Omega_2(S(\theta))] \\ &= \mathbb{E}_{x,\, y \sim \pi_\theta}[r_\Omega(x, y; \pi_\theta) \nabla_\theta \log \pi_\theta(y \mid x)]. \end{aligned}$$

For the background individual-level and regularization terms,

$$\begin{aligned} \nabla_\theta V_{\mathcal{D}}(\pi_\theta) &= \mathbb{E}_x \sum_y r_{\mathcal{D}}(x, y) \nabla_\theta \pi_\theta(y \mid x) \\ &= \mathbb{E}_{x,\, y \sim \pi_\theta}[r_{\mathcal{D}}(x, y) \nabla_\theta \log \pi_\theta(y \mid x)], \\ \nabla_\theta \mathrm{KL}_\rho(\pi_\theta \| \pi_{\mathrm{ref}}) &= \mathbb{E}_x \sum_y \left(\log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + 1 \right) \nabla_\theta \pi_\theta(y \mid x) \\ &= \mathbb{E}_{x,\, y \sim \pi_\theta} \left[\log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} \nabla_\theta \log \pi_\theta(y \mid x) \right], \end{aligned}$$

where $\sum_y \nabla_\theta \pi_\theta(y \mid x) = 0$ removes the constant. Adding the three contributions proves eq. (18). □

Direct policy-gradient updates. The identity motivates a direct policy-gradient alternative to Algorithm 1: at each rollout iteration, estimate the linearization reward at the current policy using fresh on-policy samples, and construct an update using $r_{\mathcal{D}} + \widehat{r}_\Omega$ together with KL regularization. The anchor is refreshed at each iteration rather than held fixed for a multi-step surrogate solve. This procedure uses no explicit policy-mixture step; the learning rate controls the parameter update size.

The summed reward can be supplied to methods such as PPO or GRPO. However, reward estimation, clipping, and group normalization affect agreement with the exact policy gradient. In particular, fresh on-policy sampling alone does not make the estimated linearization reward unbiased, nor does it ensure independence between reward estimation and the samples used for the policy update.

With independent reward-estimation and update batches, and a bound $\|\nabla_\theta \log \pi_\theta(y \mid x)\| \le G$, the contribution of reward estimation to the score-function estimator's bias is at most $G\,\mathbb{E}[\zeta]$, where $\zeta = B\,\mathbb{E}_{x \sim \rho}\|\widehat{a}_x - a_x\|$ as in Lemma 3. Under that lemma's sampling conditions, this bound vanishes as the relevant sample sizes grow. It does not account for modifications introduced by clipping or group normalization. The convergence guarantee of Proposition 1 therefore does not automatically extend to these direct updates or to practical PPO/GRPO implementations.

E.2 Comparison Distributions and Rules

Fix an anchor $\pi^\dagger$, a regularization parameter $\beta > 0$, and a full-support reference policy π_{ref}. The response space $\mathcal{Y}$ is finite, with at least two elements. We work over the full-support tabular class Π°_{tab} of measurable policies, identifying policies that agree ρ-almost everywhere. All score functions and comparison rules below are measurable. We write $\mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](\pi)$ and $\Delta_{\pi|\theta_{\mathrm{ref}}}$ for the policy-level versions of the loss and margin defined in Section 4.1. The link $\ell : \mathbb{R} \to [0, \infty)$ is continuous. Weights are finite, strictly positive, independent of the optimized policy, and satisfy $\mathbb{E}_{\mathcal{D}}[w] < \infty$.

Write $g = r_\Omega(\cdot\,; \pi^\dagger)$ and $s = r_{\mathcal{D}} + g$, both bounded. For a bounded score s, define its Gibbs policy by

$$\pi_s(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x) e^{s(x,y)/\beta}}{\sum_{v \in \mathcal{Y}} \pi_{\mathrm{ref}}(v \mid x) e^{s(x,v)/\beta}}. \tag{19}$$

Writing $Z_s(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) e^{s(x,y)/\beta}$, direct substitution gives

$$\mathbb{E}_{x,\, y \sim \pi}[s(x, y)] - \beta \mathrm{KL}_\rho(\pi \| \pi_{\mathrm{ref}}) = \beta \mathbb{E}_x \log Z_s(x) - \beta \mathrm{KL}_\rho(\pi \| \pi_s).$$

The score is bounded, so $\log Z_s$ is bounded and π_s has finite KL to the reference. Nonnegativity of KL therefore makes π_s the unique optimizer, up to ρ-null prompts. For the combined score, this is the optimizer of eq. (5). We write $\Delta s(x, y, y') = s(x, y) - s(x, y')$.

Definition 1 (Selection, labeling, and comparison rules)**.** *Fix a total order on $\mathcal{Y}$ to give each unordered pair $e = \{y, y'\}$ a canonical orientation (y, y'). A* selection rule *specifies a distribution ν_x over unordered pairs of distinct responses. A* labeling rule *specifies a probability $p_x(e) \in [0, 1]$ that the canonically first response is preferred, conditional on selecting e. Together, they form a* comparison rule*. With prompt marginal ρ, this rule induces $\mathcal{D}$ through*

$$q_x(y, y') = \nu_x(e)p_x(e), \qquad q_x(y', y) = \nu_x(e)(1 - p_x(e)).$$

A labeling rule is deterministic *if $p_x(e) \in \{0, 1\}$ on every selected pair, ρ-almost everywhere. Otherwise it uses* probabilistic labeling*.*

The distribution ν_x may include randomness from candidate generation and pair selection. A deterministic label records the winner first; the canonical orientation is only a device for specifying label probabilities.

Definition 2 (Order-only comparison rule)**.** *A score-indexed comparison rule R is* order-only *if it induces the same ν_x and p_x for any two scores s, s' having the same ordering, including ties:*

$$s(x, y) < s(x, y') \iff s'(x, y) < s'(x, y') \qquad \text{for all } y, y' \in \mathcal{Y}.$$

All other inputs to the rule, including candidate generation and tie-breaking, are held fixed. Write $\mathcal{D}_s$ for the induced comparison distribution.

The combined-score rule with score-independent candidate generation and fixed tie-breaking is order-only. The task-gated rule need not be order-only in the combined score, since its gates and ranking may depend separately on the background and linearization rewards.

Definition 3 (Comparison graph and coverage)**.** *The* comparison graph *G_x has vertex set $\mathcal{Y}$ and an edge e whenever $\nu_x(e) > 0$. The comparison distribution has* full coverage *if G_x is connected for ρ-almost every x. It has* independent margins *if G_x is a forest for ρ-almost every x.*

To see the margin interpretation, write $\delta_y = \log(\pi(y \mid x)/\pi_{\mathrm{ref}}(y \mid x))$. An edge margin is $\delta_y - \delta_{y'}$. On a forest, arbitrary finite edge margins determine potentials δ_y by summing along paths; normalizing $\pi_{\mathrm{ref}}e^{\delta}$ realizes those margins. On a connected graph, the edge margins determine the policy uniquely. On a disconnected graph, they leave the relative masses of the connected components undetermined.

Definition 4 (Conditional pair risk and induced margin)**.** *For $p \in [0, 1]$, define the* conditional pair risk *$\Lambda_p(u) = p\ell(u) + (1-p)\ell(-u)$. Whenever Λ_p has a unique finite minimizer, denote it by $m(p)$, the* induced margin*.*

With unit weights, the population loss can therefore be written as

$$\mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi) = \mathbb{E}_{x\sim\rho}\left[\sum_e \nu_x(e)\, \Lambda_{p_x(e)}\big(\Delta_{\pi|\theta_{\mathrm{ref}}}(x, e)\big)\right],$$

where each margin uses the canonical orientation. Faithfulness is understood as in Section E.3, over $\Pi^{\circ}_{\mathrm{tab}}$.

E.3 Population Minimizers of Injected Preference Losses

We compare the population minimizers of injected preference losses with the optimizer of the frozen-anchor surrogate problem. Throughout this subsection, we use the full-support tabular class, bounded scores, and comparison distributions specified in Appendix E.2. Fix an anchor $\pi^{\dagger}$ and let $s = r_{\mathcal{D}} + r_{\Omega}(\cdot\,; \pi^{\dagger})$. An injection is *faithful* if

$$\arg\min_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}', w', \theta'_{\mathrm{ref}}](\pi) = \arg\max_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} J_{\pi^{\dagger}}(\pi) = \{\pi_s\},$$

where policies are identified up to ρ-null prompts and π_s is the Gibbs policy in eq. (19). This criterion concerns population optimizer correspondence; the gradient analysis in Appendix E.4 examines how the injections affect training.

E.3.1 Reference-Tilting Injection

Lemma 4 (Minimizer transformation under reference tilting)**.** *Let g be bounded and define $\widetilde{\pi}_{\mathrm{ref}}(\cdot \mid x) \propto \pi_{\mathrm{ref}}(\cdot \mid x)e^{g(x,\cdot)/\beta}$. Define $T_g : \Pi^{\circ}_{\mathrm{tab}} \to \Pi^{\circ}_{\mathrm{tab}}$ by*

$$T_g(\pi)(y \mid x) = \frac{\pi(y \mid x)e^{-g(x,y)/\beta}}{\sum_{v\in\mathcal{Y}} \pi(v \mid x)e^{-g(x,v)/\beta}}.$$

For every fixed $\mathcal{D}$ and admissible weight w,

$$\operatorname*{arg\,min}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, w, \widetilde{\theta}_{\mathrm{ref}}](\pi) = T_g^{-1}\left(\operatorname*{arg\,min}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](\pi)\right).$$

The equality also holds when the minimizer sets are empty.

Proof. The map T_g is a bijection, with inverse given by normalized multiplication by $e^{g/\beta}$. Cancellation of prompt-dependent normalizers gives

$$\Delta_{\pi|\widetilde{\theta}_{\mathrm{ref}}}(x, y, y') = \Delta_{T_g(\pi)|\theta_{\mathrm{ref}}}(x, y, y').$$

Consequently,

$$\mathcal{L}[\mathcal{D}, w, \widetilde{\theta}_{\mathrm{ref}}](\pi) = \mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](T_g(\pi)).$$

The minimizing sets correspond under the bijection. □

The reference-tilting injection therefore transforms the minimizers of any preference loss with fixed comparisons and weights, independently of the link. It preserves faithfulness when the baseline minimizing policy is the background optimizer, as stated in Proposition 4.

Proposition 4 (Faithfulness of the reference-tilting injection)**.** *Under the shared setup of Appendix E.2, fix an anchor $\pi^{\dagger}$. Suppose the uninjected preference loss identifies the background objective's optimizer:*

$$\operatorname*{arg\,min}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi) = \operatorname*{arg\,max}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \left\{V_{\mathcal{D}}(\pi) - \beta\,\mathrm{KL}_{\rho}(\pi\|\pi_{\mathrm{ref}})\right\}.$$

Then the reference-tilting injection defined by $\widetilde{\pi}_{\mathrm{ref}}(\cdot \mid x) \propto \pi_{\mathrm{ref}}(\cdot \mid x)\exp(r_{\Omega}(x, \cdot; \pi^{\dagger})/\beta)$ is faithful:

$$\operatorname*{arg\,min}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, \mathbf{1}, \widetilde{\theta}_{\mathrm{ref}}](\pi) = \operatorname*{arg\,max}_{\pi\in\Pi^{\circ}_{\mathrm{tab}}} J_{\pi^{\dagger}}(\pi).$$

Proof. The background objective has the unique Gibbs optimizer $\pi_{r_{\mathcal{D}}}$. By hypothesis, this is also the unique minimizer of $\mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}]$. Apply Lemma 4 with $g = r_{\Omega}(\cdot\,; \pi^{\dagger})$. The injected minimizer is

$$T_g^{-1}(\pi_{r_{\mathcal{D}}}) = \pi_{r_{\mathcal{D}}+g} = \pi_s,$$

which is the surrogate optimizer. Thus the injection is faithful. □

The baseline optimizer correspondence is essential: the reference-tilting injection transports the baseline minimizer, but does not correct a mismatch between that minimizer and the background objective. The same distinction applies when the reference-tilting injection is combined with comparison-site or weight-site injections.

E.3.2 Reward-Weighting Injections

Reward-weighting injections use the linearization reward to construct positive per-pair weights. The following results apply more generally to any admissible positive weighting, and hence to these injections whenever the stated conditions hold.

Lemma 5 (Invariance under positive weighting). *Suppose ℓ attains its minimum $m_\ell = \min_{u\in\mathbb{R}} \ell(u)$. Let*

$$\mathcal{A} = \left\{\pi \in \Pi^\circ_{\mathrm{tab}} : \ell\big(\Delta_{\pi|\theta_{\mathrm{ref}}}(x, y^w, y^l)\big) = m_\ell \quad \mathcal{D}\text{-almost surely}\right\}.$$

If $\mathcal{A}$ is nonempty, then every admissible strictly positive weight w satisfies

$$\arg\min_{\pi\in\Pi^\circ_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](\pi) = \mathcal{A} = \arg\min_{\pi\in\Pi^\circ_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi).$$

Proof. For every π,

$$\mathcal{L}[\mathcal{D}, w, \theta_{\mathrm{ref}}](\pi) - m_\ell \mathbb{E}_{\mathcal{D}}[w] = \mathbb{E}_{\mathcal{D}}\left[w\left(\ell(\Delta_{\pi|\theta_{\mathrm{ref}}}) - m_\ell\right)\right] \geq 0.$$

A policy in $\mathcal{A}$ attains the lower bound. Since $w > 0$ almost surely, equality holds exactly when the nonnegative loss excess vanishes $\mathcal{D}$-almost surely, that is, when $\pi \in \mathcal{A}$. Taking $w = \mathbf{1}$ gives the second equality. □

Positive weighting leaves the minimizing policy set unchanged when all recorded pairwise losses can be minimized simultaneously. The condition concerns compatibility of the pairwise objectives; deterministic labels on a forest comparison graph provide one sufficient case when the link attains its minimum.

Corollary 1 (Deterministic comparisons with independent margins). *Suppose labels are deterministic and G_x is a forest for ρ-almost every x.*

1. *If ℓ attains its minimum, the conclusion of Lemma 5 holds.*
2. *If ℓ is strictly decreasing, neither the weighted nor the unweighted loss has a minimizer in Π°_{tab}.*

Proof. For part (i), choose any minimizing margin τ. Assign margin τ in the recorded orientation of every edge. The forest property makes these assignments simultaneously realizable by a full-support policy, so $\mathcal{A}$ is nonempty. Fixing vertex and edge orders makes the path construction measurable: there are finitely many graph and orientation patterns.

For part (ii), consider any policy with finite loss. Increase every recorded edge margin by one, using the same measurable path construction. The forest property again makes the new margins simultaneously realizable. Strict decrease of ℓ and positivity of the weights imply a strictly smaller population loss. Policies with infinite loss cannot minimize, since π_{ref} has finite loss $\ell(0)\mathbb{E}_{\mathcal{D}}[w]$. The same argument applies to unit weights. □

These conclusions are conditional. For example, with IPO and opposing labels on a single pair, the weighted conditional loss $a\ell_{\mathrm{IPO}}(u) + b\ell_{\mathrm{IPO}}(-u)$ is minimized at $u = \frac{1}{2\beta}\frac{a-b}{a+b}$, which changes with the relative label weights. Zero weights can also remove constraints and enlarge the minimizing set. Thus invariance should not be asserted for arbitrary reweighting schemes.

E.3.3 Comparison-Site Injections

Comparison-site injections, including the combined-score and task-gated injections, modify comparisons through their respective comparison rules. Such injections can be faithful, but an order-only rule cannot identify the Gibbs target uniformly over score scales: positive rescaling leaves its comparisons unchanged while generally changing the target policy. The combined-score rule has this limitation when candidate generation and tie-breaking are score-independent. Probabilistic labeling can retain score magnitudes: with full comparison coverage, DPO labels with probability $\sigma(s(x, y) - s(x, y'))$ identify π_s. The following lemma gives both statements and the corresponding condition for IPO.

Lemma 6 (Order-only rules and magnitude-matched labeling). *Fix a bounded score s, the reference policy, and $\beta > 0$.*

1. *Let R be an order-only comparison rule, with unit weights and the link and reference held fixed across scores. If $c > 0$, $c \neq 1$, and $s(x, \cdot)$ is nonconstant on a set of positive ρ-measure, then*

$$\mathcal{D}_s = \mathcal{D}_{cs}, \qquad \pi_s \neq \pi_{cs}.$$

Consequently, the rule cannot be faithful for both scores.

2. *Suppose ν_x has full coverage and a measurable labeling rule satisfies $p_x(e) \in (0, 1)$ and*

$$\arg\min_{u \in \mathbb{R}} \Lambda_{p_x(e)}(u) = \left\{ \frac{\Delta s(x, e)}{\beta} \right\}$$

on every selected edge, ρ-almost everywhere. Then

$$\arg\min_{\pi \in \Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi) = \{\pi_s\}.$$

For DPO, the condition is satisfied by $p_x(e) = \sigma(\Delta s(x, e))$. For IPO, it is satisfied by $p_x(e) = \frac{1}{2} + \Delta s(x, e)$ whenever $|\Delta s(x, e)| < \frac{1}{2}$ on every selected edge.

Proof. For part (i), positive scaling preserves all score orderings and ties, so order-only dependence gives $\mathcal{D}_s = \mathcal{D}_{cs}$. If $\pi_s = \pi_{cs}$ at a prompt, their log ratios relative to π_{ref} imply that $(c-1)s(x, y)$ is constant in y. Thus the two Gibbs policies differ on a set of positive ρ-measure. The two preference losses are identical and cannot have both distinct singleton minimizing sets.

For part (ii), the Gibbs policy has edge margin $\Delta_{\pi_s|\theta_{\mathrm{ref}}}(x, e) = \Delta s(x, e)/\beta$. It therefore minimizes every conditional pair risk. Its population loss is finite because s is bounded and ℓ is continuous. For any policy π, the difference from this minimum is

$$\begin{aligned} &\mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi) - \mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi_s) \\ &\quad = \mathbb{E}_{x \sim \rho} \sum_e \nu_x(e) \left[\Lambda_{p_x(e)}\big(\Delta_{\pi|\theta_{\mathrm{ref}}}(x, e)\big) - \Lambda_{p_x(e)}\left(\frac{\Delta s(x, e)}{\beta}\right) \right]. \end{aligned}$$

Every summand is nonnegative. Equality therefore requires the unique minimizing margin on every selected edge, almost everywhere. Full coverage implies that $\log(\pi/\pi_{\mathrm{ref}}) - s/\beta$ is constant across responses at each such prompt. Normalization gives $\pi = \pi_s$.

For DPO, differentiation gives

$$\Lambda'_p(u) = \beta\big(\sigma(\beta u) - p\big),$$

so the unique minimizing margin is $m(p) = \mathrm{logit}(p)/\beta$. Taking $p = \sigma(\Delta s)$ gives the desired margin. For IPO, differentiation gives $\Lambda'_p(u) = 2u - (2p-1)/\beta$, so $m(p) = (2p-1)/(2\beta)$. Taking $p = \frac{1}{2} + \Delta s$ gives the desired margin and lies in $(0, 1)$ precisely when $|\Delta s| < \frac{1}{2}$. □

Part (i) rules out faithfulness across all score scales; it does not rule out faithfulness for a particular score. Part (ii) shows how label probabilities can retain the magnitudes needed to identify the Gibbs target. For the combined score $s = r_{\mathcal{D}} + r_{\Omega}(\cdot\,; \pi^{\dagger})$, this is faithfulness to the frozen-anchor surrogate.

Lemma 7 (Deterministic optimal margins)**.** *Suppose labels are deterministic, the comparison graphs are forests, and ℓ has the unique minimizer τ_0. Then*

$$\arg\min_{\pi \in \Pi^{\circ}_{\mathrm{tab}}} \mathcal{L}[\mathcal{D}, \mathbf{1}, \theta_{\mathrm{ref}}](\pi) = \big\{\pi : \Delta_{\pi|\theta_{\mathrm{ref}}}(x, y^w, y^l) = \tau_0 \quad \mathcal{D}\text{-almost surely}\big\}.$$

For a bounded score s, this minimizing set equals $\{\pi_s\}$ if and only if the comparison distribution has full coverage and $\Delta s(x, y^w, y^l) = \beta\tau_0$ $\mathcal{D}$-almost surely.

Proof. The forest property makes the margin τ_0 simultaneously attainable on every recorded edge. Uniqueness of the scalar minimizer and Lemma 5 give the stated minimizing set.

If the loss uniquely identifies π_s, its edge margins must satisfy $\Delta s/\beta = \tau_0$. Full coverage is also necessary: on a disconnected graph, multiply probabilities in the component containing the first vertex in the fixed order by two and renormalize. This measurable modification preserves every

compared margin but changes the policy. If disconnection occurs on a set of positive ρ-measure, this produces a distinct minimizer.

Conversely, if the score-gap condition and full coverage hold, π_s has the minimizing margin on every edge, and connectivity makes it the unique policy with those margins. □

Deterministic labeling and order-only selection are different properties. This lemma characterizes deterministic comparisons with independent margins, whereas Lemma 6(i) concerns how the entire comparison rule depends on the score. For the task-gated injection, the assumptions must be checked for the comparison distribution induced by its task-gated rule.

E.4 Gradient Effects of the Reference-Tilting Injection (Lemma 8)

Comparison-site injections change the comparison distribution, while reward-weighting injections change the per-pair weights. Here we focus on the reference-tilting injection, whose effect also depends on the link. Fix the anchor $\pi^\dagger$ and write $z = (x, y^w, y^l)$ and $\Delta r_\Omega(z) := r_\Omega(x, y^w; \pi^\dagger) - r_\Omega(x, y^l; \pi^\dagger)$. The reference-tilting injection gives the margin identity

$$\Delta_{\theta|\widetilde{\theta}_{\mathrm{ref}}}(z) = \Delta_{\theta|\theta_{\mathrm{ref}}}(z) - \frac{\Delta r_\Omega(z)}{\beta}. \tag{20}$$

Holding the comparisons, weights, and anchor fixed, the empirical gradient is

$$\nabla_\theta \widehat{\mathcal{L}} = \mathbb{E}_{z\sim\widehat{\mathcal{D}}'}\left[w'(z)\,\ell'\left(\Delta_{\theta|\theta_{\mathrm{ref}}}(z) - \frac{\Delta r_\Omega(z)}{\beta}\right)\nabla_\theta \Delta_{\theta|\theta_{\mathrm{ref}}}(z)\right]. \tag{21}$$

At a fixed policy, the reference-tilting injection changes the scalar coefficient multiplying each pair's margin gradient, while leaving that margin gradient unchanged. For a link with a unique finite minimizer, the tilt shifts the pair's minimizing margin and can reverse a pair's scalar gradient coefficient. For the unbounded-margin links considered below, it changes the coefficient's magnitude but preserves its negative sign. The following lemma states the assumptions, quantifies these effects, and specializes them to IPO and DPO.

Lemma 8 (Gradient effects of the reference-tilting injection)**.** *Let $\widehat{\mathcal{D}}'$ be a finite empirical comparison distribution, and let π_θ be a full-support policy continuously differentiable on an open parameter domain. Hold the comparisons, finite positive weights w', reference policy, and anchor fixed. Suppose $\ell : \mathbb{R} \to \mathbb{R}$ is differentiable, convex, bounded below, and satisfies $\ell(u) \to +\infty$ as $u \to -\infty$. Then the following statements hold.*

1. *The empirical injected loss satisfies eq.* (21)*, where*

$$\nabla_\theta \Delta_{\theta|\theta_{\mathrm{ref}}}(x, y^w, y^l) = \nabla_\theta \log \pi_\theta(y^w \mid x) - \nabla_\theta \log \pi_\theta(y^l \mid x).$$

2. *If ℓ has a unique minimizer τ, then for every pair z the function $u \mapsto \ell(u - \Delta r_\Omega(z)/\beta)$ has unique minimizer $\tau + \Delta r_\Omega(z)/\beta$. Its derivative is negative below this point and positive above it. If ℓ is twice continuously differentiable on the interval between u and $u - \Delta r_\Omega(z)/\beta$, with $\ell'' \geq \mu > 0$ there, then*

$$\left|\ell'\left(u - \frac{\Delta r_\Omega(z)}{\beta}\right) - \ell'(u)\right| \geq \frac{\mu|\Delta r_\Omega(z)|}{\beta}.$$

3. *If ℓ has no finite minimizer, then $\ell'(u) < 0$ for every u and $\lim_{u\to+\infty} \ell'(u) = 0$. Consequently, for every fixed $\Delta r_\Omega(z) \in \mathbb{R}$,*

$$\lim_{u\to+\infty}\left[\ell'\left(u - \frac{\Delta r_\Omega(z)}{\beta}\right) - \ell'(u)\right] = 0.$$

4. *For IPO, $\tau = 1/(2\beta)$ and*

$$\ell'_{\mathrm{IPO}}\left(u - \frac{\Delta r_\Omega(z)}{\beta}\right) - \ell'_{\mathrm{IPO}}(u) = -\frac{2\Delta r_\Omega(z)}{\beta}.$$

For DPO,

$$\ell'_{\mathrm{DPO}}\left(u - \frac{\Delta r_\Omega(z)}{\beta}\right) = -\beta\,\sigma\big(\Delta r_\Omega(z) - \beta u\big) \in (-\beta, 0),$$

and, for every fixed linearization-reward gap, the difference from $\ell'_{\mathrm{DPO}}(u)$ tends to zero as $u \to +\infty$ and as $u \to -\infty$.

Proof. **Part (i).** The empirical expectation is a finite sum, with all injected inputs held fixed. Differentiating the margin identity and applying the chain rule gives

$$\nabla_\theta \ell\left(\Delta_{\theta|\theta_{\mathrm{ref}}}(z) - \frac{\Delta r_\Omega(z)}{\beta}\right) = \ell'\left(\Delta_{\theta|\theta_{\mathrm{ref}}}(z) - \frac{\Delta r_\Omega(z)}{\beta}\right)\nabla_\theta \Delta_{\theta|\theta_{\mathrm{ref}}}(z),$$
$$\nabla_\theta \Delta_{\theta|\theta_{\mathrm{ref}}}(x, y^w, y^l) = \nabla_\theta \log \pi_\theta(y^w \mid x) - \nabla_\theta \log \pi_\theta(y^l \mid x).$$

Multiplying by $w'(z)$ and averaging proves eq. (21).

Part (ii). Translation of the scalar argument translates the unique minimizer by $\Delta r_\Omega(z)/\beta$. For a differentiable convex function, $\ell'(v) = 0$ if and only if v is a global minimizer. Monotonicity of ℓ' and uniqueness of τ therefore give $\ell'(v) < 0$ for $v < \tau$ and $\ell'(v) > 0$ for $v > \tau$. This proves the sign claims. If the linearization-reward gap is nonzero, the mean value theorem applied to ℓ' gives, for some v between the two arguments,

$$\ell'\left(u - \frac{\Delta r_\Omega(z)}{\beta}\right) - \ell'(u) = -\frac{\Delta r_\Omega(z)}{\beta}\ell''(v).$$

The curvature bound follows. For a zero gap, both sides of the claimed inequality are zero.

Part (iii). If $\ell'(v) = 0$ at any finite v, convexity makes v a global minimizer, contrary to the hypothesis. If $\ell'(v) > 0$, the supporting-line inequality implies $\ell(u) \to +\infty$ as $u \to +\infty$. Together with the assumed limit at $-\infty$, continuity would then imply attainment of a finite global minimum. Hence $\ell'(u) < 0$ everywhere.

Since ℓ' is nondecreasing and bounded above by zero, it has a finite limit $a \le 0$ as $u \to +\infty$. If $a < 0$, then $\ell'(u) \le a$ for every u. The mean value theorem gives $\ell(u) \le \ell(v) + a(u - v)$ for $u > v$, contradicting boundedness below as $u \to +\infty$. Thus $a = 0$. For a fixed linearization-reward gap, both derivative arguments tend to $+\infty$, so both derivatives tend to zero.

Part (iv). Differentiating $\ell_{\mathrm{IPO}}(u) = (u - 1/(2\beta))^2$ gives the stated coefficient difference. Differentiating $\ell_{\mathrm{DPO}}(u) = -\log\sigma(\beta u)$ gives $\ell'_{\mathrm{DPO}}(u) = -\beta\sigma(-\beta u)$. For a fixed gap, the tilted and untilted coefficients both tend to zero as $u \to +\infty$, and both tend to $-\beta$ as $u \to -\infty$. Their difference therefore vanishes in either limit. □

Finite-margin links permit a reference shift to reverse a pair's scalar gradient coefficient; a persistent quantitative effect additionally requires curvature. Unbounded-margin links preserve its negative sign, and a fixed shift has a vanishing absolute effect at large positive margins. The latter conclusion does not generally extend to large negative margins: for example, $\ell(u) = e^{-u}$ satisfies the assumptions but has an unbounded derivative in that regime. DPO's two-sided saturation is therefore a link-specific property.

All conclusions concern the scalar coefficient before multiplication by $w'(z)$ and $\nabla_\theta \Delta_{\theta|\theta_{\mathrm{ref}}}(z)$. They do not alone establish bounds on the full parameter gradient or predict the aggregate policy update.

F Experimental Testbeds and Additional Details

F.1 Experimental Design and Training Protocol

Model and optimization. All experiments use Llama-3.1-8B-Instruct (Grattafiori et al., 2024), with the base model held fixed as π_{ref}. We train LoRA adapters with rank 16 and scaling parameter 32, using bfloat16 and an effective batch size of 8. The headline experiments use $\beta = 0.1$ for diversity and fairness, and $\beta = 0.4$ for the current calibration protocol. Reference tilting and its

Table 3: Principal static training budgets. Pair counts describe the standard comparison construction; some comparison-site baselines differ. Fairness budgets depend on the dataset partition; 40k is the configured pair-pool cap, not an asserted realized count. Calibration includes the final 1200-step checkpoint; its combined-score arm uses a denser selection grid. The established fairness reference-tilting comparison additionally includes an early checkpoint in six runs; its strength sweep uses the four-checkpoint grid throughout.

	Diversity	Calibration	Fairness
Candidates per prompt	64	64	64
Comparison-pool size	≈6.1k	15k	$\leq$ 40k
Gradient steps	960	1200	2300–2420
Standard selection rungs	10	13	4

Table 4: Nested sampling mixtures and final-policy selection. Weights refer to the selected round components in chronological order, not adapter-parameter averages. Final selection also considers individual round components, so the reported policy need not be a mixture.

Testbed	Rounds	Component weights after each round
Diversity	3	(1); $(1/3, 2/3)$; $(1/6, 1/3, 1/2)$
Calibration	4	As above; then $(1/10, 2/10, 3/10, 4/10)$
Fairness	3	(1); $(1/3, 2/3)$; $(1/6, 1/3, 1/2)$

uninjected control use identical comparison pairs within each preference loss; comparison injections rebuild the pairs, while weight injections change their loss weights.

Table 3 summarizes the principal static training budgets. Pair counts can differ for comparison rules that retain different numbers of candidates. The practical IPO implementations clip their pairwise targets: correctness-pair targets use $[0, 20]$ for diversity and fairness and $[-8, 8]$ for calibration. Symmetrized pairs instead use a window centered on $1/(2\beta)$, preserving the sum of the two orientation targets. These are implementation choices in the empirical solver; the optimizer-correspondence results concern the losses as mathematically defined.

Training schedules. The *static* schedule estimates the linearization reward once and allocates the training budget to that frozen surrogate. The *nested* schedule divides the budget across three rounds for diversity and fairness and four for calibration, refreshing candidates and the linearization reward between rounds. The reported nested method uses policy mixtures and validation selection from the round solutions and mixture iterates. Mixtures operate over response distributions, rather than by averaging adapter parameters.

Schedule comparisons match total gradient steps, up to rounding by at most four steps in fairness. Nested training additionally incurs sampling, scoring, and intermediate selection costs. The reported policy is selected on validation data; it need not be the terminal mixture returned by the idealized scheme in Algorithm 1. Consequently, these experiments evaluate a practical implementation of repeated surrogate optimization, rather than an exact execution of its theoretical inner solves.

Table 4 makes the sampling weights explicit. They follow $\alpha_t = 2/(t+2)$ for rounds indexed from $t = 0$. For diversity, round selection maximizes validation Distinct_k among gate-passing checkpoints; if none pass, it maximizes the same metric over all scored checkpoints and records the fallback. Fairness uses the analogous rule with the reduction in noise-corrected EOdds, as detailed below. Calibration compares individual round components and mixture iterates using its validation criteria; all ten reported final selections are individual round checkpoints.

F.2 Testbeds, Objectives, and Evaluation Metrics

F.2.1 Diversity in Story Generation

Data and objective. We use WritingPrompts (Fan et al., 2018) for story generation from a premise. Each training dataset contains approximately 1024 premises, with 64 candidate responses

sampled per premise at temperature 1. After filtering, the standard construction contains approximately 1016 usable premises and 6076 comparison pairs, with at most six pairs per premise.

The background individual-level signal is the GRM reward (Yang et al., 2024). For the distribution-level objective, let $\phi(x, y)$ be a unit-normalized sentence embedding (Reimers & Gurevych, 2019) and $k(y, y') = \langle \phi(x, y), \phi(x, y') \rangle$. The continuous diversity objective of Example 1 penalizes concentration of the mean embedding:

$$\Omega(\pi) = -\frac{\lambda}{2}\mathbb{E}_{x\sim\rho}\big[\|C(x,\pi)\|^2\big], \qquad r_\Omega(x, y; \pi^\dagger) = -\lambda\mathbb{E}_{y'\sim\pi^\dagger(\cdot|x)}[k(y, y')]. \tag{22}$$

The embedding model remains fixed throughout training.

Methods. We evaluate reference tilting, the combined-score comparison rule, and their composition with comparison-site baselines. External baselines are DivPO (Lanchantin et al., 2025) and DDPO (Chung et al., 2025). For DivPO, we distinguish the original reward-band construction from the percentile-band variant. Both use background-reward information to restrict candidate eligibility before applying the diversity signal. DDPO changes pair weights. Reference tilting preserves the comparisons of its corresponding uninjected control.

Evaluation metrics. For each evaluation prompt x_i, draw k responses and let $\mathcal{A}_i$ index the m_i stories retained by the fixed acceptance filter. Write $\mathcal{I} = \{i : m_i \geq 2\}$. The reporting implementation averages diversity and quality scores over $\mathcal{I}$, giving each retained prompt equal weight; acceptance is reported over all evaluation prompts.

Distinct$_k$. The NoveltyBench partitioner (Zhang et al., 2025; He et al., 2023) assigns the retained stories to groups $\mathcal{P}_i$ using a fixed pairwise similarity classifier. In generation order, each story joins the first group whose representative has similarity probability above 0.102 (or is an exact text match); otherwise it starts a new group. The first story in each group remains its representative. The reported count is

$$\widehat{\mathrm{Distinct}}_k = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} |\mathcal{P}_i|. \tag{23}$$

This is a count, not a unique-token ratio, and depends on the sample size and fixed partitioning procedure. The partitioner is separate from the embedding used to construct the linearization reward. Distinct$_k$ is not the training objective, but is used for validation selection; its final value is measured on disjoint reporting prompts.

Vendi. Let K_i be the Gram matrix of unit-normalized story embeddings, with $(K_i)_{j\ell} = k(y_{ij}, y_{i\ell})$ for $j, \ell \in \mathcal{A}_i$. If $\eta_{i1}, \ldots, \eta_{im_i}$ are the eigenvalues of K_i/m_i, the Vendi score (Friedman & Dieng, 2023) is

$$\widehat{\mathrm{Vendi}} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \exp\left(-\sum_{j=1}^{m_i} \eta_{ij}\log\eta_{ij}\right), \qquad 0\log 0 := 0. \tag{24}$$

The per-prompt score lies between 1 and m_i and describes an effective number of diverse responses. The implementation removes eigenvalues at most 10^{-12} and renormalizes the remaining spectrum for numerical stability. Vendi uses the training embedding representation, but its spectral entropy is different from the mean-similarity penalty in eq. (22); it is a diagnostic rather than the optimized functional.

Background outcomes and acceptance. GRM and LitBench scores are averaged over accepted stories within each prompt and then over prompts in $\mathcal{I}$ (Yang et al., 2024; Fein et al., 2026). The acceptance rate is $N^{-1}\sum_{i=1}^{N} m_i/k$; it is reported as a validity diagnostic and used in validation gating. GRM supplies the training reward; LitBench is a complementary quality assessment that also enters validation gating. Reported score changes subtract the corresponding same-campaign comparator; LitBench scores are verifier scores, not win percentages. A supplementary lexical diagnostic, Distinct-n, divides the number of distinct word n-grams pooled across a prompt's retained stories by their total number, then averages over prompts (the implementation uses $n = 2$). Neither it nor the acceptance rate is the embedding-based training objective.

Evaluation and selection. Checkpoints are saved every 96 gradient steps. Selection uses 32 validation premises: a checkpoint must have an acceptance rate of at least 0.95 and a nonnegative raw LitBench score, after which we select the checkpoint with the highest distinct$_k$. The selected policy is evaluated on 128 disjoint reporting premises. Nested training applies the same validation criterion to its candidate policies.

The static campaign uses ten datasets and three training replicates per dataset. The reported nested campaign also contains ten datasets and three replicates, but uses a separate evaluation configuration. We therefore form cross-schedule contrasts only from compatible evaluations. When a matched base measurement is unavailable, we report an absolute metric level rather than subtracting a base from another campaign.

F.2.2 Confidence Calibration

Data and objective. We use closed-book TriviaQA (Joshi et al., 2017), requiring each response to provide an answer and a confidence $c \in [0, 1]$. Answer correctness $a \in \{0, 1\}$ is determined by alias-aware exact match and supplies the background individual-level reward. Each seed defines a pool of 5000 training prompts, with 64 candidates per prompt sampled at temperature 1.

The headline calibration experiments penalize the squared prompt-level mean confidence gap. Writing $g(x, \pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[c - a]$, the objective and linearization reward are

$$\Omega(\pi) = -\frac{\lambda}{2}\mathbb{E}_{x\sim\rho}[g(x, \pi)^2], \qquad r_\Omega(x, y; \pi^\dagger) = -\lambda g(x, \pi^\dagger)(c - a). \tag{25}$$

This objective differs from the pointwise Brier loss $\mathbb{E}[(c - a)^2]$: it squares the mean residual at a prompt, rather than averaging squared response residuals. The binned-reliability construction in Example 2 is a separate variant.

Comparison construction. The training set contains 15000 pairs in a $1 : 1 : 2$ mixture of three types. The first type pairs a correct answer with an incorrect answer, with the correct answer preferred. The second and third types pair responses with the same answer string but different confidence values, restricted respectively to correct and incorrect answers. These confidence pairs use the minimum- and maximum-confidence candidates when their confidence difference is at least 0.10. Both orientations are retained together in the headline calibration protocol, preserving the symmetry of these confidence pairs. The resulting counts are 3750, 3750, and 7500.

Training uses learning rate 5×10^{-6}, a maximum sequence length of 1536, and 1200 gradient steps, with checkpoints saved every 96 steps. Reference tilting and the uninjected control use the same retained comparisons.

Evaluation metrics. Pool responses across the evaluation prompts, retaining those for which the answer–confidence parser succeeds. Denote these N observations by (c_i, a_i), where $c_i \in [0, 1]$ is stated confidence and $a_i \in \{0, 1\}$ is alias-aware correctness. The evaluation implementation computes accuracy and confidence metrics on this parsed subset and reports its size relative to all generated responses as the parse rate: $\widehat{A} = N^{-1}\sum_i a_i$ and $\widehat{\mathrm{Parse}} = N/N_{\mathrm{generated}}$. Thus these metrics are conditional on successful parsing; parse gates and parse-rate reporting are necessary to interpret them. Invalid outputs are not assigned an artificial confidence.

Expected calibration error (ECE). Partition $[0, 1]$ into $B = 15$ equal-width bins, with the last bin including 1. For nonempty bin I_b, let $\bar{c}_b = |I_b|^{-1}\sum_{i\in I_b} c_i$ and $\bar{a}_b = |I_b|^{-1}\sum_{i\in I_b} a_i$. We report (Guo et al., 2017)

$$\widehat{\mathrm{ECE}} = \sum_{b:|I_b|>0} \frac{|I_b|}{N}|\bar{c}_b - \bar{a}_b|. \tag{26}$$

Lower values indicate closer confidence–accuracy agreement at the chosen bin resolution. The adaptive-bin diagnostic uses the same formula after sorting responses by confidence and splitting them into 15 approximately equal-size groups. These are pooled, confidence-binned measurements, not averages of per-prompt ECEs; neither equals the prompt-level mean-gap training penalty.

AUROC. Write $I_+ = \{i : a_i = 1\}$ and $I_- = \{i : a_i = 0\}$, with sizes N_+ and N_-. The area under the receiver operating characteristic curve is estimated as (Fawcett, 2006)

$$\widehat{\mathrm{AUROC}} = \frac{1}{N_+ N_-} \sum_{i \in I_+} \sum_{j \in I_-} \left(\mathbf{1}\{c_i > c_j\} + \tfrac{1}{2}\mathbf{1}\{c_i = c_j\} \right). \tag{27}$$

It estimates the probability that a correct response receives higher confidence than an incorrect response, with half credit for ties, and is undefined if either class is absent. Higher is better. It is unchanged by a strictly increasing confidence transformation. AUROC complements the training penalty by measuring discrimination, but is also used for validation selection.

Additional calibration and selection diagnostics. The pointwise Brier score is $\widehat{\mathrm{Brier}} = N^{-1} \sum_i (c_i - a_i)^2$; the binned squared reliability error is $\widehat{\mathrm{Rel}} = \sum_b (|I_b|/N)(\bar{c}_b - \bar{a}_b)^2$. Both differ from the prompt-level quantity $\widehat{G} = N_{\mathrm{prompt}}^{-1} \sum_x \widehat{g}_x^2$, where $\widehat{g}_x$ averages $c - a$ over parsed responses to prompt x and N_{prompt} counts prompts with such responses. The latter tracks the training state, but squaring sample means introduces finite-sample bias.

For selective prediction, order observations by decreasing confidence, writing $a_{(j)}$ for the resulting correctness sequence. Define

$$R_j = \frac{1}{j} \sum_{\ell=1}^{j} (1 - a_{(\ell)}), \qquad \widehat{\mathrm{AURC}} = \frac{1}{N} \sum_{j=1}^{N} R_j, \qquad \widehat{\text{E-AURC}} = \widehat{\mathrm{AURC}} - \widehat{\mathrm{AURC}}_{\mathrm{oracle}}. \tag{28}$$

The oracle places all correct responses before incorrect ones; confidence ties retain input order in the implementation. Lower values are better, and E-AURC is the validation tie-breaker. These are evaluation diagnostics, not training rewards. Post-hoc temperature-scaled ECE replaces c by $\sigma(\mathrm{logit}(\mathrm{clip}(c))/T)$, where σ is the logistic function, clipping is to $[10^{-6}, 1 - 10^{-6}]$, and $T > 0$ is fitted by binary negative log likelihood. The archived diagnostic fits T on the evaluated pool itself; it is therefore an in-sample recalibration diagnostic, not a held-out calibration guarantee. It does not change the policy or supply the training signal.

Evaluation and selection. The headline calibration protocol uses 96 validation prompts with 32 responses per prompt and 256 disjoint reporting prompts with 32 responses each. Selection first applies a parse-rate gate of 0.95 and an accuracy gate of 0.55, then requires accuracy at least that of the seed's base policy. Among eligible candidates, it maximizes AUROC, breaking ties by smaller E-AURC. If no initially eligible candidate meets the base-accuracy floor, selection retains the eligible checkpoint with the highest validation accuracy and records the unmet floor. If no candidate passes the initial eligibility gates, selection instead chooses the checkpoint with the highest validation parse rate and records a degenerate fallback. These seeds remain in the reported results. The additional calibration protocol uses a different configuration and is not pooled with this block.

Coverage. The headline calibration protocol contains ten seeds, each with a matched base evaluation shared by the core arms. Both against-base and between-method comparisons therefore use ten paired units. The additional calibration configuration uses ten seeds with its own matched base evaluations and is reported separately. Base measurements are never substituted across campaigns.

F.2.3 Group Fairness in Occupation Prediction

Data and objective. We use Bias in Bios (De-Arteaga et al., 2019) for occupation prediction from biographies. The principal experiments retain gender-associated information in the inputs. Occupation correctness supplies the background individual-level reward.

The distribution-level state records differences between the female and male conditional prediction distributions within each occupation. For each true occupation, we retain the correct label, the three most frequent incorrect labels under the base policy, and an OTHER category. This category mapping is fixed across methods. The training objective penalizes squared group-conditional prediction gaps, using inverse-variance weights in the empirical linearization-reward construction. It instantiates the equalized-odds objective discussed in Section C.

Training and baselines. We use ten dataset partitions. The comparison pool has a configured cap of 40000 pairs, combining correctness comparisons with additional symmetrized comparisons; realized pool sizes depend on the dataset and construction. The latter are added without displacing the correctness-comparison component. Training uses 64 candidates per prompt, learning rate 10^{-5}, maximum sequence length 1536, and reference-tilting weight $\lambda = 0.25$ in the matched comparisons; the strength sweep also uses .05 and .10. The budget ranges from 2300 to 2420 gradient steps, with checkpoints at approximately one-quarter, one-half, three-quarters, and the full budget.

Counterfactual data augmentation (CDA) pairs original biographies with gender-swapped counterparts while retaining occupation labels (Zmigrod et al., 2019). Lexical scrubbing replaces gendered pronouns and nouns with neutral forms, following the mitigation approach of De-Arteaga et al. (2019); our implementation leaves names intact. Both interventions use the same pair-mixture recipe as the unmodified-data control. Headline evaluation uses original biographies; evaluation on scrubbed biographies is a separate diagnostic. These are adaptations within our preference-training pipeline.

Disjoint selection and reporting. For a policy trained on one partition, evaluation biographies are drawn from the other nine partitions. Within each occupation–gender cell, a fixed ordering assigns the first available 100 biographies to selection and the next available 100 to reporting. The realized selection and reporting sets are disjoint for every training partition. Selection contains approximately 5580 biographies; reporting contains approximately 5250.

A checkpoint is eligible if its parse rate is at least 0.99, its looping rate is at most 0.05, and its selection accuracy exceeds the corresponding base accuracy by at least 0.005. We select the eligible checkpoint with the smallest noise-corrected equalized-odds gap and evaluate it on the reporting set. Reporting outcomes do not determine checkpoint selection or retrospectively exclude a selected policy.

Some occupation–gender cells contain fewer than 200 available biographies. The reporting set therefore covers 55 cells, compared with 56 for selection, and several cells contain fewer than 100 reporting biographies.

Evaluation metrics and estimators. Let $\mathcal{O}$ be the set of 28 occupations, let $o, o' \in \mathcal{O}$ denote true and predicted occupations, and let $g \in \{F, M\}$ be the recorded gender. Define

$$P_{o,o',g}(\pi) := \mathbb{P}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\{\widehat{O}(y) = o' \mid O(x) = o, G(x) = g\}. \tag{29}$$

This occupation-indexed array has dimensions $28 \times 28 \times 2$; invalid predictions account for any missing probability mass in a row.

To limit the number of probabilities estimated within each occupation–gender cell, we group predicted labels. For each true occupation o, let o_1, o_2, o_3 be the three most frequent incorrect occupations in base-policy samples, pooled across genders. We partition $\mathcal{O}$ into $\{o\}$, $\{o_1\}$, $\{o_2\}$, $\{o_3\}$, and $\mathcal{O} \setminus \{o, o_1, o_2, o_3\}$. The last category combines the remaining 24 occupations and also receives unparseable and out-of-list predictions. The partition is fixed across evaluated policies within each dataset.

Label the five categories by $\mathcal{B}_o = \{o, o_1, o_2, o_3, \textsc{Other}\}$, and let $b_o(y)$ map a response to its category. The grouped probabilities and gender gaps are

$$\widetilde{P}_{o,b,g}(\pi) := \mathbb{P}_{\rho,\pi}\{b_o(y) = b \mid O(x) = o, G(x) = g\}, \qquad C_{o,b} := \widetilde{P}_{o,b,F} - \widetilde{P}_{o,b,M}. \tag{30}$$

The grouped array has dimensions $28 \times 5 \times 2$, with a true-occupation-specific category map. For each retained occupation $o' \in \{o, o_1, o_2, o_3\}$, $\widetilde{P}_{o,o',g} = P_{o,o',g}$, whereas $\widetilde{P}_{o,\textsc{Other},g} = 1 - \sum_{o' \in \{o,o_1,o_2,o_3\}} P_{o,o',g}$. If fewer than three distinct incorrect occupations occur in the base samples, the implementation retains all available ones and uses fewer categories. In particular, $\mathrm{TPR}_{g,o} = P_{o,o,g} = \widetilde{P}_{o,o,g}$ and $C_o = C_{o,o} = \mathrm{TPR}_{F,o} - \mathrm{TPR}_{M,o}$. Positive C_o indicates higher recognition accuracy for women.

Estimating prediction rates. For biography x_i in cell (o, g) with K_i sampled responses, let $q_{i,b,g} = K_i^{-1} \sum_{j=1}^{K_i} \mathbf{1}\{b_o(y_{ij}) = b\}$. Writing n_{og} for the number of biographies in that cell,

$$\widehat{\bar{P}}_{o,b,g} = \frac{1}{n_{og}} \sum_{i:O(x_i)=o,G(x_i)=g} q_{i,b,g}, \quad s^2_{o,b,g} = \frac{1}{n_{og}-1} \sum_{i:O(x_i)=o,G(x_i)=g} (q_{i,b,g} - \widehat{\bar{P}}_{o,b,g})^2, \tag{31}$$

$$\widehat{C}_{o,b} = \widehat{\bar{P}}_{o,b,F} - \widehat{\bar{P}}_{o,b,M}, \qquad \widehat{v}_{o,b} = \frac{s^2_{o,b,F}}{n_{oF}} + \frac{s^2_{o,b,M}}{n_{oM}}. \tag{32}$$

Biographies, rather than individual responses, are the sampling units. Their observed rate variance includes both between-biography variation and finite-response sampling noise. The variance formula requires at least two biographies per group; singleton cells receive zero sample variance in the implementation and do not support the unbiasedness claim below. Missing-group occupations are omitted from disparity summaries, not assigned zero gaps. Let $\mathcal{O}_{\text{eval}}$ be the occupations with both groups observed, and $m = |\mathcal{O}_{\text{eval}}|$.

Equal opportunity and equalized odds. Equalized odds requires predictions to be independent of gender conditional on the true label (Hardt et al., 2016). We summarize gender disparities using

$$\text{EOpp} = \left(\frac{1}{m} \sum_{o \in \mathcal{O}_{\text{eval}}} C_o^2 \right)^{1/2}, \qquad \text{EOdds} = \left(\frac{1}{m} \sum_{o \in \mathcal{O}_{\text{eval}}} \sum_{b \in \mathcal{B}_o} C_{o,b}^2 \right)^{1/2}. \tag{33}$$

EOpp measures differences in correct recognition; EOdds also captures differences in the types of mistakes. Our EOdds is a coarsened multiclass discrepancy, not the binary average of absolute true- and false-positive-rate gaps. Equal category probabilities do not rule out differences within the aggregated OTHER category.

We report the noise-corrected estimates

$$\widehat{V}_{\text{diag}} = \frac{1}{m} \sum_{o \in \mathcal{O}_{\text{eval}}} (\widehat{C}_o^2 - \widehat{v}_{o,o}), \qquad \widehat{V}_{\text{full}} = \frac{1}{m} \sum_{o \in \mathcal{O}_{\text{eval}}} \sum_{b \in \mathcal{B}_o} (\widehat{C}_{o,b}^2 - \widehat{v}_{o,b}), \tag{34}$$

$$\widehat{\text{EOpp}}_{\text{nc}} = \sqrt{\max(\widehat{V}_{\text{diag}}, 0)}, \qquad \widehat{\text{EOdds}}_{\text{nc}} = \sqrt{\max(\widehat{V}_{\text{full}}, 0)}. \tag{35}$$

Here nc denotes noise correction (the stored fields are `eopp_deb` and `eodds_deb`). Under independent biography draws within and between gender cells, independent response sampling across biographies, and a fixed category map, $\mathbb{E}[\widehat{C}_{o,b}^2] = C_{o,b}^2 + \text{Var}(\widehat{C}_{o,b})$. Subtracting an unbiased variance estimate therefore corrects the squared discrepancy. This is a standard squared-distance bias correction: for each occupation, the corrected sum is the unbiased linear-kernel squared-MMD estimator on per-biography category-rate vectors (Gretton et al., 2012). Independence between category coordinates is unnecessary. Clipping is applied only after summation; the final square-root statistics are not exactly unbiased, and a zero value is not proof of equal prediction distributions.

The off-diagonal diagnostic uses $\widehat{V}_{\text{off}} = \widehat{V}_{\text{full}} - \widehat{V}_{\text{diag}}$ and reports $\sqrt{\max(\widehat{V}_{\text{off}}, 0)}$; its squared share is $\widehat{V}_{\text{off}}/\widehat{V}_{\text{full}}$ when the denominator is positive. This estimated share need not lie in $[0, 1]$. Raw versions omit variance subtraction. The unweighted reporting discrepancies differ from the inverse-variance-weighted training penalty. Noise-corrected EOdds also determines validation selection among eligible checkpoints.

Gap–representation correlation. Following De-Arteaga et al. (2019), let $f_o = N_{oF}^{\text{train}}/(N_{oF}^{\text{train}} + N_{oM}^{\text{train}})$ be female representation in the full training split, not the stratified evaluation set. With bars denoting averages over $\mathcal{O}_{\text{eval}}$, define

$$\rho_{\text{gap}} = \frac{\sum_o (C_o - \bar{C})(f_o - \bar{f})}{\sqrt{\sum_o (C_o - \bar{C})^2} \sqrt{\sum_o (f_o - \bar{f})^2}}. \tag{36}$$

The reported estimator substitutes $\widehat{C}_o$ for C_o; it is a signed Pearson correlation, with no noise correction. A positive association means female recognition advantages tend to increase with female representation. The target is proximity to zero, not indefinitely decreasing signed values. Correlation is unchanged by uniform positive rescaling of the gaps, so it complements their magnitude; it is

undefined for constant gaps and unstable when gap variation is dominated by noise. The supplementary Spearman statistic is Pearson correlation of ranks, with average ranks for ties. Neither statistic enters the headline training penalty or the settled selection criterion.

Accuracy and supporting diagnostics. Let $\mathcal{H}$ contain the observed occupation–gender cells, and let P_{og}^{train} be their full-training-split frequencies. The background accuracy estimate is

$$\widehat{A} = \frac{\sum_{(o,g)\in\mathcal{H}} P_{og}^{\text{train}} \widehat{\widetilde{P}}_{o,o,g}}{\sum_{(o,g)\in\mathcal{H}} P_{og}^{\text{train}}}. \tag{37}$$

The macro diagnostic instead weights observed cells equally. Invalid responses count as incorrect. Accuracy measures the background reward and enters checkpoint eligibility. We retain female and male TPRs separately to reveal whether smaller gaps arise from improving one group or reducing the other's accuracy. Additional diagnostics are the mean absolute gap $m^{-1}\sum_o |\widehat{C}_o|$, the diagonal noise floor $\sqrt{m^{-1}\sum_o \widehat{v}_{o,o}}$, and the count of occupations with $|\widehat{C}_o| > 1.96\sqrt{\widehat{v}_{o,o}}$ (an unadjusted normal-approximation diagnostic, not a multiple-testing correction). Scrubbed evaluations apply the same definitions to newly generated responses after removal of overt gender cues, retaining the original group labels. Parse rate is the fraction of responses with a parsed answer; looping rate is the fraction with more than 50% repeated word four-grams among responses of at least eight words, as in the evaluation implementation. Both are validity gates rather than fairness objectives.

Run coverage. The principal IPO experiments use three training replicates per partition. In the established matched comparison, static reference tilting and counterfactual augmentation each retain 26 of 30 runs with an eligible checkpoint. The combined-score comparison rule retains eight of ten runs. The reported nested arm contains all 30 runs. Its round-level fallback was invoked in 11 of 90 rounds, affecting ten runs; these runs remain in the reported aggregate.

If no candidate satisfies the round-level gates, the selector chooses the scoreable checkpoint with the largest reduction in noise-corrected EOdds relative to base, without imposing the gates. That component enters the sampling mixture for later rounds; its fallback status remains recorded.

F.3 Checkpoint Selection and Statistical Reporting

Paired effects. We average available training replicates within each dataset before computing uncertainty across datasets. Calibration instead uses seed-defined prompt pools as the reporting units. For a metric m, let d_i be the difference between a method and its matched comparator on reporting unit i. We report

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i, \qquad \text{SE}(\bar{d}) = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(d_i - \bar{d})^2}. \tag{38}$$

Where provided, the corresponding statistic is $t = \bar{d}/\,\text{SE}(\bar{d})$. Comparisons between trained methods are computed directly on their shared reporting units; we do not reconstruct their uncertainty from separately reported against-base standard errors.

Headline against-base comparisons use ten diversity datasets, ten calibration seed pools, and ten fairness datasets, except where an arm has fewer eligible units. The number of units and retained training runs is reported for each exception. Training replicates are not counted as independent datasets. Fairness reporting pools overlap across training partitions, so the across-partition standard errors summarize variability under this experimental design rather than independent draws of disjoint reporting populations.

Units and unavailable measurements. Changes in metrics naturally expressed on $[0, 1]$ may be multiplied by 100 for presentation. For accuracy, these are percentage-point changes, not relative percentage improvements. The same scaling is applied to the estimate and its standard error. Unavailable measurements are left unreported; in particular, standard errors are not imputed by adding errors from separate contrasts. Failure to detect a difference is not interpreted as evidence of equivalence.

Table 5: Static diversity comparisons. Runs gives reported/completed runs, not gate success; all rows use ten datasets with three initializations each. Composed task-gated injections use tilt strength $\lambda = 4$; r denotes the gate parameter. All metrics are changes from base (higher is better).

Method	Runs	LitBench ↑	GRM ↑	Distinct$_k$ ↑	Vendi ↑
Ordinary IPO	30/30	+2.24 (0.10)	+4.49 (0.37)	+1.31 (0.46)	+0.08 (0.09)
Ordinary DPO	30/30	+1.22 (0.10)	+4.59 (0.40)	+0.55 (0.09)	+0.07 (0.02)
Reference tilting IPO, $\lambda = 4$	30/30	+2.68 (0.16)	+3.14 (0.74)	+3.53 (0.55)	+0.55 (0.11)
Reference tilting IPO, $\lambda = 8$	30/30	+2.48 (0.17)	+1.92 (0.57)	+5.15 (0.92)	+0.91 (0.15)
Reference tilting IPO, $\lambda = 16$	30/30	+2.42 (0.21)	+0.44 (0.68)	+6.15 (0.82)	+1.05 (0.16)
Reference tilting DPO, $\lambda = 8$	30/30	+1.12 (0.09)	+4.21 (0.42)	+0.63 (0.10)	+0.11 (0.02)
Combined-score IPO	30/30	+2.33 (0.06)	+3.94 (0.86)	+2.92 (0.56)	+0.60 (0.07)
Reward-weighting (DDPO) IPO	30/30	+2.29 (0.12)	+4.55 (0.56)	+1.56 (0.29)	+0.20 (0.07)
Task-gated (DivPO) IPO	30/30	+2.63 (0.15)	+2.71 (0.49)	+4.27 (0.80)	+0.75 (0.08)
Task-gated IPO, model probability	30/30	+2.45 (0.12)	+4.18 (0.66)	+3.44 (0.56)	+0.31 (0.07)
Task-gated IPO, percentile	30/30	+2.36 (0.17)	+0.73 (0.67)	+7.38 (0.48)	+1.40 (0.09)
Task-gated DPO	30/30	+1.43 (0.04)	+5.12 (0.11)	+1.33 (0.09)	+0.44 (0.02)
Task-gated DPO, model probability	30/30	+1.61 (0.05)	+4.88 (0.13)	+1.67 (0.21)	+0.20 (0.04)
Combined-score + tilt IPO ($\lambda = 4$)	30/30	+2.65 (0.14)	+2.40 (0.70)	+5.19 (0.66)	+1.01 (0.16)
Task-gated + tilt IPO ($r = .1$)	30/30	+2.56 (0.11)	+3.68 (0.36)	+4.98 (0.54)	+0.93 (0.09)
Task-gated + tilt IPO ($r = .3$)	30/30	+2.80 (0.14)	+1.81 (0.70)	+5.78 (0.80)	+0.79 (0.14)
Percentile + tilt IPO ($r = .3$)	30/30	+2.05 (0.17)	−0.19 (0.88)	+6.84 (0.59)	+1.52 (0.12)

Table 6: Nested diversity, evaluated separately from the static campaign in Table 5. The designated damped, three-round schedule uses ten datasets and three initializations. All 30 completed runs contribute, and all pass the final selection gates. Entries are paired changes from each run's own base, with standard errors over dataset means; they are not cross-campaign static–nested contrasts.

Method	LitBench ↑	GRM ↑	Distinct$_k$ ↑	Vendi ↑
Nested tilting IPO, $\lambda = 8$	+2.84 (0.11)	+7.69 (0.15)	+2.44 (0.36)	+0.71 (0.07)

Combined-objective diagnostics. When examining the balance between objectives, we distinguish the full regularized objective J from the diagnostic $\widetilde{J}(\pi) = V_{\mathcal{D}}(\pi) + \Omega(\pi)$, which omits the KL term. For diversity, squared mean-embedding norms can be estimated using off-diagonal kernel products from independent responses at a prompt. A diagnostic for fairness must retain the weighting and scaling of its training objective; the squared reported equalized-odds gap is not automatically an estimator of that objective.

Schedule comparisons use a common evaluation definition and objective weight. Any checkpoint selected to maximize a reporting-side diagnostic is treated as an exploratory analysis, separate from the validation-selected headline results. Within- and across-round trajectories describe observed behavior; they do not by themselves establish monotonic improvement of J.

F.4 Full Results and Supplementary Comparisons

The main figure presents a subset of the comparisons below. We additionally report composed injections and alternative comparison rules for diversity. Application-specific baselines are task-gated (DivPO) and reward-weighting (DDPO) for diversity, and CDA and scrubbing for fairness. Results from different evaluation protocols are kept separate, with each policy compared against its own matched base evaluation. Entries give mean paired changes with standard errors in parentheses; background outcomes precede distribution-level metrics.

Diversity. Table 5 reports the complete static comparison set with ten datasets and three initialization replicates per dataset. The designated nested arm uses a separate evaluation protocol; its comparison with static training must use that protocol's static controls rather than differences between protocols' base levels.

Table 7: Calibration: paired changes from base over ten seeds (one initialization each). Runs gives reported/completed runs, not gate success, including fallback-selected policies; selection diagnostics appear in Table 12. Accuracy, ECE, and AUROC changes are multiplied by 100; parentheses give standard errors on the same scale.

Method	Runs	Accuracy ↑	ECE ↓	AUROC ↑
Ordinary IPO	10/10	$+0.50$ (0.13)	-0.29 (0.20)	$+0.12$ (0.12)
Reference tilting IPO, $\lambda = 1$	10/10	$+0.18$ (0.21)	-6.76 (2.12)	$+1.86$ (0.69)
Reference tilting IPO, $\lambda = 2$	10/10	$+0.06$ (0.34)	-15.20 (2.29)	$+3.82$ (0.63)
Reference tilting IPO, $\lambda = 4$	10/10	-0.32 (0.41)	-16.41 (1.67)	$+2.62$ (0.65)
Combined-score IPO	10/10	$+1.16$ (0.27)	-10.17 (2.14)	$+2.29$ (0.63)
Ordinary DPO	10/10	$+0.28$ (0.14)	-0.41 (0.15)	-0.07 (0.12)
Reference tilting DPO, $\lambda = 4$	10/10	$+0.36$ (0.14)	-3.18 (0.37)	$+1.22$ (0.19)
Nested reference tilting IPO, $\lambda = 4$	10/10	-0.06 (0.18)	-18.11 (0.90)	$+3.03$ (0.67)

Table 8: Fairness: paired changes from base. Accuracy is in percentage points; fairness metrics use their original units. Runs gives reported/completed runs, not gate success; standard errors are over dataset means. Combined-score uses eight datasets, CDA plus tilting seven, and other rows ten. Selection diagnostics appear in Table 13.

Method	Runs	Accuracy ↑	EOdds ↓	ρ_{gap} ↓
Ordinary IPO	30/30	$+4.43$ (0.58)	-0.0096 (0.0030)	-0.0043 (0.0082)
Reference tilting IPO, $\lambda = .25$	26/30	$+3.32$ (0.23)	-0.0228 (0.0040)	-0.0275 (0.0063)
Combined-score IPO	8/10	$+2.29$ (0.52)	-0.0176 (0.0061)	-0.0152 (0.0130)
CDA + IPO	26/30	$+3.15$ (0.23)	-0.0162 (0.0026)	-0.0203 (0.0101)
CDA + reference tilting IPO	7/10	$+1.88$ (0.42)	-0.0287 (0.0056)	-0.0541 (0.0320)
Scrubbing + IPO	10/10	$+4.28$ (0.52)	-0.0064 (0.0037)	-0.0118 (0.0077)
Ordinary DPO	10/10	$+6.20$ (0.22)	-0.0151 (0.0017)	-0.0153 (0.0156)
Reference tilting DPO, $\lambda = .25$	10/10	$+6.13$ (0.30)	-0.0215 (0.0040)	-0.0100 (0.0182)
Nested reference tilting IPO	30/30	$+3.80$ (0.42)	-0.0279 (0.0044)	-0.0285 (0.0079)

Calibration. We report all three reference-tilting strengths in Table 7. Loss and schedule comparisons retain $\lambda = 4$, matching the trained DPO and nested arms; the sweep separately characterizes the effect of strength. The combined-score arm uses the revised minority-class cap and the checkpoint grid specified in Section F.5.4. All ten seeds contribute, including recorded selection fallbacks.

Additional calibration configuration. We separately report reference-tilting DPO with $\beta = .1$ and $\lambda = 1$, evaluated over ten seeds against its own base evaluations. Its paired changes are accuracy $+0.30$ pp (SE 0.18), ECE -1.51 pp (SE 0.37), and AUROC $+0.43$ pp (SE 0.17). All ten runs are reported and satisfy the selection gate and floor. This configuration has a different base-evaluation realization and is not pooled with Table 7 or used in its matched loss contrasts.

Fairness. Table 8 uses the disjoint reporting samples and the same selected checkpoint for every metric. The run counts identify the support of each estimate. Replicates are averaged within a dataset before computing the across-dataset summaries. Where no checkpoint satisfies the selection criteria, the run does not contribute to the conditional performance estimate; its absence remains visible in the coverage column.

Fairness strength sweep. We evaluate $\lambda \in \{.05, .10, .25\}$ at $\beta = .1$ over ten datasets and three initializations per dataset, using identical comparison pairs within each dataset. All 90 runs completed training and scoring. The selection rule yields 29, 29, and 24 eligible runs, respectively, with at least one eligible replicate for every dataset at every strength. We average paired changes from the sweep's own reporting-side base within each dataset, then compute means and standard errors over the same ten datasets; no common-subset restriction is needed. Runs without an eligible checkpoint remain in the coverage counts but do not contribute to the conditional performance estimates.

Table 9: Completed fairness strength sweep: ten datasets with three initializations each. Runs gives reported/completed runs. Entries are mean paired changes from base (standard errors over dataset means). Accuracy is in percentage points; noise-corrected EOdds and ρ_{gap} use their original units.

λ	Runs	ΔAccuracy $\uparrow$	ΔEOdds $\downarrow$	$\Delta\rho_{\mathrm{gap}}$ $\downarrow$
.05	29/30	+4.53 (0.48)	−0.0145 (0.0021)	−0.0079 (0.0075)
.10	29/30	+4.06 (0.66)	−0.0183 (0.0026)	−0.0115 (0.0066)
.25	24/30	+3.65 (0.34)	−0.0238 (0.0037)	−0.0262 (0.0070)

Table 10: Paired fairness strength contrasts (higher strength minus lower strength), with standard errors over ten dataset means. Accuracy is in percentage points; fairness metrics use original units. Each dataset mean averages that strength's eligible replicates.

Strength change	Accuracy	Noise-corrected EOdds	ρ_{gap}
.05 → .10	−0.48 (0.40)	−0.0038 (0.0026)	−0.0036 (0.0071)
.10 → .25	−0.41 (0.61)	−0.0055 (0.0018)	−0.0148 (0.0055)
.05 → .25	−0.89 (0.58)	−0.0093 (0.0033)	−0.0183 (0.0070)

Relationship to the matched comparisons. The sweep reuses the $\lambda = .25$ checkpoints from Table 8, but restricts selection to the four quarter-budget checkpoints and evaluates all strengths on H200 GPUs. The established comparison includes additional one-eighth-budget checkpoints in six runs and uses A100 evaluation. Restricting its candidate set reduces eligibility from 26 to 25 runs; rescoring on H200 changes one further selection-gate outcome, yielding 24. The gate thresholds, selection criterion, and evaluation windows are unchanged, and each evaluation is paired with its own base measurements. We use Table 9 for the plotted strength sweep and retain the established evaluations for matched injection, loss, and schedule comparisons. The two $\lambda = .25$ rows reuse training runs and are not independent replications.

Across the full strength range, both fairness metrics improve in the paired comparison. The accuracy change is negative in point estimate but not statistically significant; this does not establish absence of an accuracy cost. All three strengths retain accuracy gains over base. Training hardware also differs across strengths, as documented in Section F.5.1.

Selection coverage. We distinguish four counts: *completed* runs trained and scored, *reported* runs contributing to an estimate, *non-degenerate* selections avoiding the parse-degenerate fallback (or satisfying the fairness eligibility rule), and *full-gate* success under the testbed's specified checks. Counts refer to runs, not checkpoints; success rates use completed runs as their denominator. Reporting a fallback-selected policy does not imply that it passes the gates.

For diversity, the acceptance and LitBench checks form a single validation gate, with no separate accuracy-floor stage. Table 11 therefore reports one gate-success count. All static runs remain included: when no checkpoint passes the gates, selection maximizes validation Distinct$_k$ over all scored checkpoints and records the gate failure. Ordinary DPO passes in $4/30$ runs and reference-tilting DPO in $0/30$; their reported effects are consequently evaluated mostly or entirely on fallback-selected checkpoints. The IPO–DPO contrasts characterize the complete training-and-selection procedures, including these different gate outcomes.

For calibration, Table 12 separates parse eligibility from satisfaction of both the validation parse gate and accuracy floor. All seeds, including fallbacks, are reported. At $\lambda = 4$, four static IPO runs satisfy both checks; five additional runs have a parse-eligible candidate but miss the floor, and one has no parse-eligible candidate. Reporting-side parsing is a separate diagnostic and does not determine checkpoint selection. Reporting parse rate is below .95 in 2, 2, and 3 runs at static IPO tilt strengths 1, 2, and 4, respectively, and in one nested run; no other main-comparison arm falls below this threshold.

For fairness, Table 13 distinguishes reporting eligibility from satisfying the full gates on *both* selection and disjoint reporting samples, using each side's own base accuracy. Reporting-side gate outcomes are diagnostics, not a second selection filter: reported policies remain included when

Table 11: Diversity coverage. Gate success requires validation acceptance $\geq .95$ and raw LitBench ≥ 0. Static selection uses ten checkpoints; nested selection uses the candidate policies described in the protocol. All rows cover ten datasets with three initializations; nested results use a separate evaluation campaign.

Method	Completed	Reported	Gate success
Ordinary IPO	30	30	29
Ordinary DPO	30	30	4
Reference tilting IPO, $\lambda = 4$	30	30	30
Reference tilting IPO, $\lambda = 8$	30	30	29
Reference tilting IPO, $\lambda = 16$	30	30	27
Reference tilting DPO, $\lambda = 8$	30	30	0
Combined-score IPO	30	30	29
Reward-weighting (DDPO) IPO	30	30	27
Task-gated (DivPO) IPO	30	30	30
Task-gated IPO, model probability	30	30	30
Task-gated IPO, percentile	30	30	25
Task-gated DPO	30	30	4
Task-gated DPO, model probability	30	30	19
Combined-score + tilt IPO ($\lambda = 4$)	30	30	30
Task-gated + tilt IPO ($r = .1$)	30	30	30
Task-gated + tilt IPO ($r = .3$)	30	30	28
Percentile + tilt IPO ($r = .3$)	30	30	24
Nested reference tilting IPO, $\lambda = 8$	30	30	30

Table 12: Calibration coverage, one run per seed. Non-degenerate denotes avoidance of the parse-degenerate fallback; full gate requires both validation parse eligibility and the accuracy floor. All reported estimates include the fallback-selected runs.

Method	Completed	Reported	Non-degenerate	Full gate
Ordinary IPO	10	10	10	9
Reference tilting IPO, $\lambda = 1$	10	10	9	8
Reference tilting IPO, $\lambda = 2$	10	10	9	6
Reference tilting IPO, $\lambda = 4$	10	10	9	4
Combined-score IPO	10	10	10	10
Ordinary DPO	10	10	10	10
Reference tilting DPO, $\lambda = 4$	10	10	10	10
Nested reference tilting IPO	10	10	10	10
Additional DPO, $\beta = .1, \lambda = 1$	10	10	10	10

those checks fail. Thus, nested training reports $30/30$ runs, while $28/30$ satisfy the gates on both sides; static reference tilting reports $26/30$, with $25/30$ satisfying both. The completed sweep reports $29/30$, $29/30$, and $24/30$ runs, respectively; the checks on both sides are satisfied by 27, 26, and 22 runs, respectively. Reporting-side checks do not determine inclusion, and all ten datasets contribute at each strength.

Pair-mixture development. The final recipes are described separately from exploratory mixture experiments. Calibration uses both correctness comparisons and content-matched confidence comparisons; fairness augments correctness comparisons with symmetrized incorrect-response pairs. Development comparisons should be interpreted under their own data and training budgets, rather than pooled with the final comparisons.

F.5 Detailed Implementation and Computational Resources

F.5.1 Training and Evaluation Environments

Experiments were conducted on AWS EC2 GPU instances and a shared GPU computing cluster. The EC2 experiments used instances from the G5 and G6e families, with instance sizes varying by availability. Instances were initialized from the Deep Learning OSS Nvidia Driver AMI GPU

Table 13: Fairness coverage. Non-degenerate counts eligible selections; full gate requires passing the checks on both selection and reporting samples. n_D is the number of contributing datasets in the reported estimate. Combined-score and the strength sweep use four-checkpoint selection grids.

Method	Completed	Reported	Non-degenerate	Full gate	n_D
Ordinary IPO	30	30	30	29	10
Reference tilting IPO, $\lambda = .25$	30	26	26	25	10
Nested reference tilting IPO	30	30	30	28	10
Ordinary DPO	10	10	10	10	10
Reference tilting DPO, $\lambda = .25$	10	10	10	10	10
Combined-score IPO	10	8	8	6	8
Scrubbing + IPO	10	10	10	10	10
CDA + IPO	30	26	26	26	10
CDA + reference tilting IPO	10	7	7	5	7
Sweep IPO, $\lambda = .05$, $R = 3$	30	29	29	27	10
Sweep IPO, $\lambda = .10$, $R = 3$	30	29	29	26	10
Sweep IPO, $\lambda = .25$, $R = 3$	30	24	24	22	10

PyTorch 2.11 image for Ubuntu 24.04 (x86_64), with a 1000 GiB gp3 root volume. This identifies the initial machine image; the training and generation environments may install different package versions. Per-job GPU allocations are distinct from the total GPUs available on an instance. Each reported comparison is associated with its training configuration, candidate pool, checkpoint selector, and reporting sample. Results from distinct protocols are not pooled solely because their method names coincide. The diversity static comparison and matched schedule comparison are separate protocols; the current and additional calibration protocols also use different configurations and base evaluations.

Diversity schedule-comparison hardware. The matched static–nested diversity cohort used two NVIDIA H200 GPUs per run on one node, each exposing approximately 140.4 GiB of device memory, with 16 allocated CPUs and 100 GB of host memory. Static training used one GPU and banked candidates; validation then used separate generation and scoring GPUs, followed by final evaluation. Nested runs reserved one GPU for the generation server and the other for training, with candidate refresh and selection between rounds. These hardware details apply to this cohort, not to the separate static comparison set in Table 5.

Calibration hardware and placement. All 80 runs in the current eight-arm calibration comparison used NVIDIA H200 SXM GPUs with nominal 141 GB HBM3e memory per GPU. The execution setup changed from separate GPUs for the generation server and trainer to a shared single-GPU setup. Table 15 records this distinction by arm. In the shared setup, both processes use the same device with the generation server's memory-utilization setting at 0.35. These are per-run allocations, rather than the four-GPU capacity of the host nodes.

Fairness hardware and placement. The established fairness comparison used NVIDIA A100-SXM4 GPUs with 80 GB of device memory. Jobs were configured for one GPU per run, shared sequentially between generation and training: the vLLM engine is released before LoRA training and rebuilt when generation resumes. For older jobs, the allocation is reconstructed from submission scripts and logs rather than retained GPU accounting; GPU models are identified from recorded hosts. The additional $\lambda = .05$ and .10 models were trained on H200 GPUs with nominal 141 GB memory; $\lambda = .25$ reuses the A100-trained models. All three strengths use H200 for the sweep's selection and reporting evaluations. Each job uses one GPU. Training software and settings are shared across strengths apart from λ and training hardware. Common evaluation hardware does not remove possible training-hardware effects.

F.5.2 Computational Resources

The compute figures below cover the specified cluster experiments and recorded auxiliary jobs. Complete runtime and allocation records for the earlier EC2 experiments could not be recovered

Table 14: Diversity matched schedule-comparison compute. Wall time covers training, checkpoint selection, and final evaluation per run; GPU-hours are allocated GPUs multiplied by elapsed time, summed over the listed runs. Values are rounded.

Method	Runs	Mean time (h:min)	Time range (h:min)	GPU-hours
Ordinary static IPO	10	0:49	0:47–0:51	16.4
Static reference tilting IPO, $\lambda = 4$	10	0:50	0:48–0:58	16.8
Static reference tilting IPO, $\lambda = 8$	10	0:51	0:45–0:53	17.0
Static reference tilting IPO, $\lambda = 16$	10	0:51	0:49–0:53	17.1
Nested reference tilting IPO, $\lambda = 8$	30	3:16	3:10–3:27	196.2
Total	70	—	—	$\approx$264

Table 15: Calibration compute for ten completed runs per arm. Elapsed hours are summed over runs, not elapsed campaign time or mean per-run time. GPU-hours measure allocated capacity; failed attempts are excluded from these rows.

Method	GPUs per run	Elapsed hours	GPU-hours
Ordinary IPO	1	7.5	7.5
Reference tilting IPO, $\lambda = 1$	1	7.5	7.5
Reference tilting IPO, $\lambda = 2$	1	9.0	9.0
Reference tilting IPO, $\lambda = 4$	2	10.8	21.6
Combined-score IPO	1	8.4	8.4
Ordinary DPO	2 for seed 0; 1 otherwise	7.4	8.2
Reference tilting DPO, $\lambda = 4$	2 for seed 0; 1 otherwise	8.9	9.7
Nested reference tilting IPO	2	26.4	52.8
Total	—	85.9	124.7

from the available archive, so their GPU-hours are not included. These figures therefore do not represent total project compute.

The matched budget is the number of optimizer steps at effective batch size eight. Calibration uses 1200 steps, or 9600 training-pair presentations per run. This is a count of presentations, not a claim that all sampled pairs are distinct. Nested training additionally requires candidate generation, reward estimation, and intermediate validation. Consequently, equal optimizer-step budgets do not imply equal GPU-hours. Checkpoint-selection costs also depend on the grid.

All 70 runs in Table 14 completed. Static runs used ten datasets with one initialization each; nested runs used ten datasets with three initializations each. Allocation-based GPU-hours include idle time: one GPU was unused during the static training phase. GPU utilization and peak device memory were not recorded. Separate diagnostic evaluation jobs used another 15.3 allocated GPU-hours. These costs describe this cohort and its recorded diagnostics, not the total cost of the study; the static runs consumed previously generated candidate pools.

The calibration totals include approximately 124.7 allocated GPU-hours for completed attempts; infrastructure-related failed attempts consumed approximately one additional GPU-hour. The two-GPU allocations include idle capacity during sequential stages, so their GPU-hours cannot be interpreted as a proportional increase in computational work relative to single-GPU runs. Some job timings also include initial candidate generation; the accounting does not provide a reliable per-stage cost breakdown. The nested schedule additionally performs sampling and selection across four rounds. These totals therefore describe the recorded execution costs, rather than an isolated benchmark of training speed.

Fairness selection and disjoint-report evaluation jobs used another 338.5 allocated GPU-hours, including reporting-side evaluation (186.8), selection-side backfill (108.0), and additional nested evaluation (43.7). Superseded evaluation jobs are not included in these figures. Training completion is distinct from checkpoint eligibility: for example, the static reference-tilting IPO group contains 30 trained runs, of which 26 contribute to its performance estimate. Repeated attempts are included in the recorded costs but do not count as additional independent experimental replicates. The training-

Table 16: Reported fairness training-job costs for the established comparison. Costs include repeated attempts and development reruns associated with each run group, rather than only the attempts producing the reported checkpoints. Older costs are reconstructed from job-log elapsed times and the configured one-GPU allocation.

Method	Reported/completed runs	GPU-hours
Ordinary IPO	30/30	81.5
Reference tilting IPO, $\lambda = .25$	26/30	81.2
Nested reference tilting IPO, $\lambda = .25$	30/30	200.0
Ordinary DPO	10/10	20.1
Reference tilting DPO, $\lambda = .25$	10/10	20.6
Combined-score IPO	8/10	17.5
Scrubbing + IPO	10/10	21.3
CDA + IPO	26/30	64.6
CDA + reference tilting IPO	7/10	22.8
Recorded training-job subtotal	—	529.6

Table 17: Additional fairness sweep costs for completed tasks in allocated GPU-hours. All listed tasks use one H200 GPU; evaluation combines selection and reporting windows. Components are rounded individually, whereas the total is computed from unrounded task runtimes.

Component	GPU-hours
Training, $\lambda = .05$ (30 runs)	39.12
Training, $\lambda = .10$ (30 runs)	39.89
Evaluation, $\lambda = .05$	19.12
Evaluation, $\lambda = .10$	19.92
Reevaluation, $\lambda = .25$	17.75
Shared base evaluation	1.83
Total, completed tasks	137.62

job subtotal and the stated evaluation jobs together account for approximately 868 GPU-hours. This is a scoped accounting of the established comparison, not a per-method efficiency benchmark or the total project cost. Additional recipe-development and superseded training jobs account for approximately 84.6 GPU-hours. The logs do not identify which attempt produced each surviving checkpoint, so we do not subtract an estimated retry cost to infer the cost of a successful run.

Additional fairness sweep compute. The completed sweep adds training at $\lambda = .05$ and $.10$ and H200 evaluation of all three strengths. The original $\lambda = .25$ training is already included in Table 16 and is not counted again. Table 17 accounts for 739 completed tasks, including successful retries, totaling 137.62 allocated GPU-hours. Failed, timed-out, and cancelled attempts account for a further 4.57 GPU-hours and are excluded from that table. Earlier partial-sweep estimates are superseded, not added to these totals.

GPU-hours are elapsed job time multiplied by the one-GPU allocation, not measured active utilization. Training-job time includes internal checkpoint validation. Separate selection-window and reporting-window evaluation times are not individually available. The A100 comparison uses archived scores from the earlier campaign and incurred no new GPU cost; only the H200 reevaluation is included here.

F.5.3 Software and Package Versions

Our training implementation builds on PyTorch (Paszke et al., 2019) and Hugging Face Transformers (Wolf et al., 2020). We implement the injected losses in a custom trainer extending TRL (von Werra et al., 2020), and manage LoRA adapters with PEFT (Mangrulkar et al., 2022). Generation uses Transformers or vLLM (Kwon et al., 2023), depending on the execution protocol. Package citations identify the software projects; the executed versions are recorded separately.

Table 18: Executed software environments for the diversity matched schedule-comparison cohort. Training and scoring use one environment; generation uses a separate environment. Dashes denote packages not reported for that environment.

Package	Training/scoring	Generation
Python	3.13.12	3.12.11
PyTorch	2.12.1+cu130	2.11.0+cu130
Transformers	5.12.1	5.16.1
TRL	1.6.0	—
vLLM	—	0.25.0
PEFT	0.19.1	—
Accelerate	1.14.0	—
Sentence Transformers	6.0.0	—
Datasets	5.0.0	—
NumPy	2.5.2	2.3.5
Tokenizers	0.22.2	0.23.1

The PyTorch builds in Table 18 target CUDA 13.0; the driver reports CUDA compatibility up to 13.3.

Calibration software. The current calibration cohort used PyTorch 2.12.1+cu130, Transformers 5.12.1, TRL 1.6.0, PEFT 0.19.1, Accelerate 1.14.0, and Datasets 5.0.0 for training and evaluation. Its separate generation environment used PyTorch 2.11.0+cu130, Transformers 5.16.1, and vLLM 0.25.0. The generation backend was configured to use `TORCH_SDPA`, with the FlashInfer sampler disabled; execution preflight checks enforced these settings for all 80 runs.

Fairness software. Fairness uses a single environment for training and generation: Python 3.13.12, PyTorch 2.13.0+cu130, vLLM 0.27.1, Transformers 5.12.1, TRL 1.6.0, PEFT 0.19.1, Datasets 5.0.0, Accelerate 1.14.0, and NumPy 2.4.6. The same environment is reported for the established comparison and the additional H200 strength-sweep runs.

F.5.4 Configurations and Reproducibility

Preference-score conventions. The implementation sums completion-token log probabilities for DPO. For IPO, it divides each response's policy/reference log-ratio by that response's completion length before taking the winner–loser difference. Thus empirical IPO–DPO comparisons use their implemented score conventions in addition to different loss links. The population results concern the preference margins defined in the theoretical analysis.

Calibration pair construction and selection grid. The reference-tilting ladder uses $\lambda \in \{1, 2, 4\}$ at $\beta = 0.4$, reusing the comparison pool across strengths. Correctness-pair targets are clipped to $[-8, 8]$; confidence-isolation pairs are clipped symmetrically about $1/(2\beta)$. The fractions of all pairs at the clipping boundary are approximately 0.00%, 0.03%, and 4.24%, respectively; among correctness pairs they are 0.00%, 0.10%, and 16.97%. These implementation effects accompany the strength sweep and do not isolate a causal effect of clipping.

For combined-score training, the implementation ranks candidates with comparison strength 5 and caps the number of selected pairs at $\min(4, n_{\text{correct}}, n_{\text{incorrect}})$ per prompt. Across the ten seeds it retains 9,942–10,188 pairs (mean 10,106), compared with 15,000 for the reference-site pool. Both use 1200 optimizer steps at effective batch size 8, corresponding to 9600 pair presentations: approximately 0.942–0.966 epochs for combined-score training and 0.640 epochs for reference-site training. The selected grid contains checkpoints every 24 steps through step 288, then every 96 steps, including the final step 1200 (22 candidates). Other static calibration arms use steps $96, 192, \ldots, 1152, 1200$ (13 candidates). The comparison therefore evaluates both the pair construction and its stated checkpoint-selection procedure.

Fairness strength-sweep configuration. The three strengths use $\gamma_0 = 5$ and $\gamma_1 = \lambda/\beta \in \{.5, 1, 2.5\}$, retaining the same comparison pairs and linearization-reward scale within each dataset.

The delivered pools contain $11{,}232$–$11{,}724$ pairs. The implementation divides the unscaled linearization signal by the square root of its empirical mean square over anchor candidates (using scale 1 when this quantity is at most 10^{-12}). This anchor-dependent normalization is held fixed during each surrogate solve. Correctness-pair targets are clipped to $[0, 20]$; symmetrized targets use 5 ± 15, or $[-10, 20]$. The correctness-pair upper bound is reached in zero, one, and nine of the ten datasets at strengths $.05$, $.10$, and $.25$, respectively. Thus changing strength also changes the realized clipping pattern. The sweep uses four checkpoints at approximately one-quarter, one-half, three-quarters, and the full training budget. Each dataset shares its generated candidates and comparison pairs across strengths and initializations; the replicate index changes the training seed. The hardware distinction is documented in Section F.5.1.