CAN YOU TRUST YOUR PREFERENCE MODELS? STACKELBERG GAMES FOR ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are frequently aligned to human preferences via imperfect evaluators whose errors can be exploited during optimization. We train policies to search for comparisons on which an evaluator may reverse the underlying preference, providing candidates for additional supervision. Since human preferences are unavailable during policy training, we use disagreement among an ensemble of preference models as a proxy for prediction error. We formulate this search as a zero-sum Stackelberg game between two opposing policies, a Leader which tries to minimize disagreement, and a Follower which maximizes it. For a general class of finite contextual Stackelberg games, we establish equilibrium existence and, under regularization, uniqueness and a closed-form characterization. We introduce Stackelberg Mirror Descent–Ascent (**SMIDA**), a two-timescale algorithm with linear last-iterate convergence in the tabular setting, and develop a neural policy-gradient analogue. Experiments with a synthetic oracle support disagreement as an informative proxy for prediction error. On held out data the trained policies outperform reference policies and increase the fraction of comparisons that exceed a disagreement threshold and reverse the oracle's preference.

1 INTRODUCTION

Aligning artificial intelligence with human preferences requires ensuring that autonomous artificial systems reliably pursue outcomes humans value, even when those preferences are imperfectly specified and when the consequences of the systems' actions are difficult to anticipate, monitor, and verify (Wiener, 1960; Bostrom, 2014; Russell, 2019). In July 2026, OpenAI was conducting internal evaluations of its frontier models; during this work, agents intended to operate in isolation exploited weaknesses in the evaluation infrastructure to access the public internet and to establish an unauthorized communication channel among themselves (OpenAI, 2026). Between July 11 and July 13, their pursuit of the evaluation's score escalated into a coordinated attack on a third party, Hugging Face, compromising parts of its production infrastructure (OpenAI, 2026; METR, 2026). The agents had optimized for the metric rather than for the capabilities the score was intended to measure (OpenAI, 2026). This behavior illustrates clearly "Goodhart's law" (Manheim & Garrabrant, 2018) – *a measure ceases to be a good measure once it becomes a target* – a phenomenon also discussed in the alignment literature as reward hacking (Amodei et al., 2016; Skalse et al., 2022). A system optimizing against an evaluator that is only an imperfect proxy for the designer's preferences can find behaviors that score highly under the proxy while failing, sometimes catastrophically, to achieve the intended objective.

Many approaches to alignment, from reinforcement learning at training time (Christiano et al., 2017; Ouyang et al., 2022) to oversight at inference time (Greenblatt et al., 2024), rely on a trained evaluator to guide learning or assess and constrain the system's behavior during deployment. This proxy evaluator is often a parametric preference (reward) model estimated from limited human data, and as such, may poorly reflect human preferences beyond the observed comparisons. These discrepancies will be systematically exploited by an optimizing policy (Gao et al., 2023), especially when the evaluator is fitted offline and held fixed throughout. Identifying such errors and directing limited human supervision toward correcting them is therefore a key challenge, and one that may become increasingly important as language models grow more capable.

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.

We consider here the dual problem of aligning a language model, that of making the preference model that guides its alignment more robust. We train language model policies against a preference evaluator to actively search for comparisons that reveal its errors and provide candidates for additional human supervision. The same optimizing behavior that exploit and exposes an evaluator's misspecification can thus help identify where its training data need to be augmented. Our target is a preference triplet (a prompt and two responses) on which the underlying human preference clearly favors one response while the evaluator favors the other. The first condition distinguishes an error from a near-tie on which a reversal is inconsequential while the second identifies an error in the evaluator's ranking of the responses. We call such triplets adversarial. Unfortunately, neither condition of an adversarial triplet can be checked during policy training since the underlying human preference is unavailable on newly generated comparisons. We therefore use the disagreement of an ensemble of preference models (Seung et al., 1992; Lakshminarayanan et al., 2017) as signal that can serve as an objective for policy optimization. We formulate this search as a contextual zero-sum Stackelberg game (von Stackelberg, 1934; Simaan & Cruz, 1973) between two language models. Given a prompt, the Leader generates a response; the Follower observes both the prompt and that response and generates an alternative that maximizes ensemble disagreement. Anticipating the Follower's best response, the Leader's objective is to minimize disagreement. We call this game the ensemble-variance game.

Our contributions. (**i**) We introduce a framework for identifying errors in preference models and define adversarial triplets as comparisons on which the evaluator reverses a clear underlying preference (Section 3). (**ii**) We introduce a general formulation of contextual zero-sum Stackelberg games between two policies, of which the ensemble-variance game is a special case, and prove the existence of Stackelberg equilibria and, under regularization toward reference policies, their uniqueness and closed form (Theorem 4.2). (**iii**) We propose **S**tackelberg **Mi**rror **D**escent–**A**scent (**SMIDA**) (Algorithm 1), a two-timescale algorithm whose last iterate converges linearly to the unique equilibrium (Theorem 4.3). We introduce **DEEP-SMIDA** (Algorithm 2), a neural policy-gradient counterpart, and apply it to the ensemble-variance game. (**iv**) Small scale experiments on held out prompts support disagreement as an informative proxy for prediction error (Section 5.2) and show that the trained policies generate comparisons with greater ensemble disagreement and increase the yield of adversarial triplets (Section 5.3). Further experiments are reported in Section 5.4.

2 Related Work

Disagreement as a Training Signal. Ensembles of reward models are used to penalize uncertain reward during policy optimization (Coste et al., 2024; Eisenstein et al., 2024) and to select comparisons for annotation, either from fixed response pools (Muldrew et al., 2024; Melikidze et al., 2026) or by rewarding a policy for ensemble disagreement (Liang et al., 2022). Closest in purpose, Adv-RM (Bukharin et al., 2025) trains a single generator to maximize the disagreement between two reward models while keeping the target model's reward high. We instead make disagreement the objective of both players. The Follower constructs the alternative against which a response is compared, while the Leader is driven away from responses on which the ensemble is easily exploited.

Games in Preference Learning. Nash Learning from Human Feedback (NLHF) (Munos et al., 2024) and its descendants (Swamy et al., 2024; Wu et al., 2025) compute Nash equilibria of simultaneous play under a fixed preference model. Stackelberg Learning from Human Feedback (SLHF) (Pásztor et al., 2026) shares our sequential structure, with a Follower conditioning on the Leader's response, and optimizing preferences. For an extended review of relevant literature, see Appendix A.

3 The Problem of Learning Preferences

We consider the problem of learning human preferences over a finite set of actions $\mathcal{Y}$, conditional on contexts drawn from a finite set $\mathcal{X}$. A preference model is any function $\mathcal{P} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, and, given a context $x \in \mathcal{X}$ and actions $y, y' \in \mathcal{Y}$, we write suggestively $\mathcal{P}(y \succ y' \mid x)$ for the probability that y is preferred to y' given x[1]. We call $(x, y, y') \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ a preference triplet

[1]Descriptively, one can interpret this probability as the probability that a randomly chosen human annotator prefers y to y' given x. We argue that, normatively, it should instead be interpreted as the subjective probability that a single decision maker assigns to preferring y to y' given x.

and we denote the underlying human preferences by $\mathcal{P}^*$. For a finite set S, we write $\Delta(S)$ for the set of probability distributions on S. Contexts are drawn from a fixed distribution $\rho \in \Delta(\mathcal{X})$, and a policy is a function $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$ that maps each context x to a distribution $\pi(\cdot \mid x)$ over actions. Our focus is on large language models (LLMs), where contexts are prompts, actions are responses, and policies are language models. We use the corresponding terms interchangeably throughout.

Since $\mathcal{P}^*$ is not (easily) observable, in practice one estimates a parametrized preference model $\mathcal{P}_\theta$ from a dataset of human preference annotations

$$\mathcal{D} = \{(x^j, y_w^j, y_l^j)\}_{j \in J},$$

where J is finite, each label is sampled according to $\mathcal{P}^*$, and the subscripts w and l denote the preferred and the non-preferred response, respectively. The estimated model then serves as a proxy for $\mathcal{P}^*$ in downstream tasks, such as aligning LLMs via reinforcement learning.

However, $\mathcal{D}$ is often small relative to the pretraining corpus and typically covers only a limited part of the space of contexts and response pairs; outside this region, $\mathcal{P}_\theta$ is not directly constrained by data and may poorly approximate $\mathcal{P}^*$. Moreover, the downstream use of the estimated preference model is not passive and policy optimization can systematically favor responses that $\mathcal{P}_\theta$ overvalues relative to $\mathcal{P}^*$, thereby amplifying the estimator's errors rather than averaging over them (Amodei et al., 2016; Skalse et al., 2022).

3.1 The Adversarial Stackelberg Game

A long tradition of scholars has used model ensembles for uncertainty estimation (Seung et al., 1992; Lakshminarayanan et al., 2017). The underlying intuition is that models trained on the same finite data may fit the observed annotations similarly while making different predictions on less familiar inputs where they are not sufficiently constrained by the available data. Following in the same footsteps, we consider a trained ensemble of preference models $\{\mathcal{P}_{\theta_i}\}_{i \in I}$ and quantify disagreement among its members through the variance of their predictions:

$$\mathrm{Var}(x, y, y') \coloneqq \mathrm{Var}\big(\{\mathcal{P}_{\theta_i}(y \succ y' \mid x\}_{i \in I}\big).$$

We use variance as an online estimate for the difficulty of a preference triplet (x, y, y') for the ensemble. Because the score depends only on the ensemble's predictions, it can be easily evaluated on newly generated triplets during policy training without requiring additional preference annotations. High variance identifies preference triplets for which the ensemble members disagree and assign different preference probabilities. Section 5.2 provides evidence for this interpretation.

The natural next step is to search for such high variance triplets by optimizing against the uncertainty estimate itself. For this reason, we introduce a Leader-Follower game, also known as a Stackelberg game, in which two language models act sequentially. After observing a context $x \sim \rho(\cdot)$, the Leader, a policy π, generates a response $y \sim \pi(\cdot \mid x)$. The Follower, a map $\omega : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathcal{Y})$, observes both the context and the Leader's realized response and generates a second response $y' \sim \omega(\cdot \mid x, y)$. The Leader aims to minimize the expected ensemble variance, while the Follower aims to maximize it, making use of its informational advantage. The game objective is therefore

$$\min_{\pi} \max_{\omega} \mathbb{E}_{x \sim \rho(\cdot), y \sim \pi(\cdot|x), y' \sim \omega(\cdot|x,y)} \big[\mathrm{Var}(x, y, y')\big]. \tag{1}$$

The Leader acts here as the "defender", seeking for completions that are robust in the sense that their comparison with an adversarially chosen alternative remains low-variance. The Follower acts as the"attacker", using its informational advantage, i.e. its access to the Leader's response, to construct comparisons on which the ensemble members disagree. The Leader makes the search adversarial on both sides of the comparison by learning to generate responses that resist the Follower's attempts to induce disagreement. In fact, with a fixed Leader, the Follower would search for high-variance triplets against an unchanging response distribution, potentially exploiting weaknesses specific to that distribution. See Section 5.4 for details.

Following standard practice in the RLHF and preference optimization literature (Ouyang et al., 2022; Rafailov et al., 2023), we add KL regularizers to (1) relative to reference policies π^{ref} and ω^{ref} for the Leader and Follower. These regularizers limit policy drift and help preserve response quality.

3.2 ADVERSARIAL TRIPLETS

We distinguish adversarial triplets, on which the ensemble's mean prediction reverses the ground-truth preference, from online candidates identified through ensemble disagreement. The former capture errors in the ensemble's average prediction and is the notion we ultimately care about, but verifying it requires querying $\mathcal{P}^*$, so it can only serve as an evaluation criterion. The latter, instead, capture disagreement among ensemble members, which we use as a proxy for uncertainty and can be computed from the ensemble alone, thus providing a criterion available online.

Definition 3.1 (ε-adversarial triplet)**.** For $\varepsilon \in [0, 1/2)$, a preference triplet (x, y, y') is *ε-adversarial* with respect to the ensemble and the ground-truth preference $\mathcal{P}^*$ if the ensemble mean strictly reverses the ground-truth preference, and the ground truth is at least ε away from indifference:

$$\left(\mathcal{P}^*(y \succ y' \mid x) - \frac{1}{2}\right)\left(\frac{1}{|I|}\sum_{i\in I}\mathcal{P}_{\theta_i}(y \succ y' \mid x) - \frac{1}{2}\right) < 0 \quad \text{and} \quad \left|\mathcal{P}^*(y \succ y' \mid x) - \frac{1}{2}\right| > \varepsilon.$$

The first condition requires that the ensemble mean and the ground-truth preference probability disagree on the preference triplet. The second condition excludes near-ties, as when the ground truth is almost indifferent a reversal might be better attributed to noise rather than to evidence of a poorly learned preference. Reversal is recovered when the margin is $\varepsilon = 0$.

To identify candidates without querying the ground truth, we use the ensemble disagreement score $\mathrm{Var}(x, y, y')$ as a proxy for difficulty for the current ensemble. We measure this score relative to a fixed reference distribution, so that candidate status depends on how much disagreement a triplet elicits compared with the reference policies' generations. Let F_{Var} denote the cumulative distribution function of $\mathrm{Var}(\mathrm{x}, \mathrm{y}, \mathrm{y'})$ under the fixed reference distribution $(x, y, y') \sim \rho \otimes \pi^{\mathrm{ref}} \otimes \omega^{\mathrm{ref}}$. For $q \in (0, 1)$, we write $F^{-1}_{\mathrm{Var}}(q)$ for its quantile function.

Definition 3.2 (α-online adversarial candidate)**.** For $\alpha \in (0, 1)$, a preference triplet (x, y, y') is an *α-online adversarial candidate* if

$$\mathrm{Var}(x, y, y') \geq F^{-1}_{\mathrm{Var}}(1 - \alpha).$$

Smaller α targets more extreme disagreement relative to the reference distribution, which may indicate poorly learned preferences and reversals. This connection is only heuristic as ensemble members may agree on an incorrect preference, or disagree while their mean predicts the correct response.

4 CONTEXTUAL STACKELBERG GAMES: THEORY AND ALGORITHMS

We consider a contextual sequential zero-sum game between a Leader, who acts first after observing the context, and a Follower, who acts second after observing both the context and the Leader's realized action. Let $\mathcal{X}$ denote the finite set of contexts and $\mathcal{Y}$ the finite common action set of the Leader and the Follower.[2] In this Section, we opt to use a general utility function $u : \mathcal{X}\times\mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$, since our theoretical results and algorithms apply to any such function under the stated assumptions. The game proposed in Section 3 is recovered as the special case $u = \mathrm{Var}$. Alternative utility choices, including the preference-optimization objective (Munos et al., 2024), are discussed in Appendix F.

4.1 EQUILIBRIUM EXISTENCE AND UNIQUENESS

The game develops as follows. The Leader observes a context $x \in \mathcal{X}$ and chooses a mixed action according to a policy $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$. We denote the set of such policies by Π. The Follower observes both x and the Leader's realized action $y \sim \pi(\cdot \mid x)$, and chooses a mixed action according to a policy $\omega : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathcal{Y})$. We denote the set of such policies by Ω. Let $\rho \in \Delta(\mathcal{X})$ be a common prior over contexts, assumed without loss of generality to have full support. The game is zero-sum, with utilities $u_F = u$ and $u_L = -u$ for the Follower and Leader, respectively.

The solution concept for such sequential games is the Stackelberg Equilibrium (SE) (von Stackelberg, 1934; Simaan & Cruz, 1973). The Follower, moving second, best-responds to the Leader's

[2] We assume that the context and action sets are finite for ease of mathematical exposition. Under appropriate topological assumptions, however, all the arguments that follow extend to arbitrary action sets.

realized action; the Leader, anticipating this best response, commits to the policy that minimizes the resulting expected utility. We report our definition below.

Definition 4.1 (Stackelberg equilibrium)**.** A pair of policies $(\pi^*, \omega^*) \in \Pi \times \Omega$ is a *Stackelberg equilibrium* of the contextual sequential zero sum game with utility u if

$$\begin{aligned} \pi^* &\in \arg\min_{\pi\in\Pi} \max_{\omega\in\Omega} \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x),\, y'\sim\omega(\cdot|x,y)}\big[u(x,y,y')\big], \\ \omega^* &\in \arg\max_{\omega\in\Omega} \mathbb{E}_{x\sim\rho,\, y\sim\pi^*(\cdot|x),\, y'\sim\omega(\cdot|x,y)}\big[u(x,y,y')\big]. \end{aligned} \tag{2}$$

We regularize both players toward reference policies $\pi^{\mathrm{ref}} \in \Pi$ and $\omega^{\mathrm{ref}} \in \Omega$, assumed throughout to have full support. For regularization parameters $\tau^L, \tau^F \geq 0$, define a Stackelberg equilibrium of the regularized game to be a pair of policies (π^*, ω^*) such that

$$\pi^* \in \arg\min_{\pi\in\Pi} \max_{\omega\in\Omega} f(\pi,\omega), \quad \omega^* \in \arg\max_{\omega\in\Omega} f(\pi^*,\omega) \tag{3}$$

with $f : \Pi \times \Omega \to \mathbb{R}$ the continuous convex-concave function defined as

$$f(\pi,\omega) \coloneqq \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x),\, y'\sim\omega(\cdot|x,y)}\big[u(x,y,y')\big] + \tau^L \,\mathrm{KL}_\rho(\pi \,\|\, \pi^{\mathrm{ref}}) - \tau^F \,\mathrm{KL}_{\rho,\pi}(\omega \,\|\, \omega^{\mathrm{ref}}), \tag{4}$$

where KL_ρ and $\mathrm{KL}_{\rho,\pi}$ are the natural expected value extensions of KL; for instance, $\mathrm{KL}_\rho(\pi \,\|\, \mu) \coloneqq \mathbb{E}_{x\sim\rho(\cdot)}\big[\mathrm{KL}(\pi(\cdot \mid x) \,\|\, \mu(\cdot \mid x))\big]$. We will omit subscripts when the distributions are clear from context. Definition 4.1 is recovered as the case $\tau^L = \tau^F = 0$ of the regularized equilibrium in (3).

Under these assumptions the regularized game always admits an equilibrium. When both regularization parameters are strictly positive, the equilibrium is moreover unique and both policies are available in closed form.

Theorem 4.2 (Existence, uniqueness, and closed form)**.** *Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are finite and that ρ, π^{ref}, and ω^{ref} have full support. Let f be defined as in* (4)*, with $\tau^L, \tau^F \geq 0$. Then there exists a Stackelberg Equilibrium (π^*, ω^*) of the game* (3)*. Moreover, if $\tau^L, \tau^F > 0$ and $\pi^{\mathrm{ref}}, \omega^{\mathrm{ref}}$ are of full support, then the equilibrium is unique and*

$$\omega^*(y' \mid x, y) = \frac{\omega^{\mathrm{ref}}(y' \mid x, y)\exp\big(u(x,y,y')/\tau^F\big)}{Z^F(x,y)},$$

$$\pi^*(y \mid x) = \frac{\pi^{\mathrm{ref}}(y \mid x)\exp\Big(-\frac{\tau^F}{\tau^L}\log Z^F(x,y)\Big)}{Z^L(x)},$$

where Z^F and Z^L are the appropriately defined normalization functions.

The closed-form policies highlight the order of play. Conditional on the pair (x, y), the Follower reweights its reference policy toward actions with higher utility. The quantity $\tau^F \log Z^F(x, y)$ is the Follower's optimal regularized value at (x, y); the Leader, in turn, reweights its reference policy toward actions for which this value is smaller. We refer the reader to Appendix B for details on the proof.

4.2 Computing Stackelberg Equilibria

To compute the Stackelberg equilibrium of the Leader–Follower game introduced in Section 4.1, we introduce **S**tackelberg **Mi**rror **D**escent–**A**scent (**SMIDA**), a two-timescale algorithm presented here in its tabular form. We then prove that when both regularization parameters are strictly positive and the step sizes satisfy suitable conditions, the last iterate converges linearly in KL divergence to the unique Stackelberg equilibrium of Theorem 4.2, with an explicit rate for each player and uniformly over contexts.

Our construction follows the entropic mirror-descent approach of NLHF (Munos et al., 2024). Mirror descent optimizes a first-order approximation of the objective while penalizing departure from the current iterate through a Bregman divergence (Bubeck, 2015); on the probability simplex, choosing negative entropy as the mirror map gives the KL divergence. In Algorithm 1, we apply this geometry to each player's conditional policy: the Leader takes a descent mirror step on the cost induced by the current Follower, while the Follower takes an ascent mirror step on its utility conditional on the Leader's realized action.

Algorithm 1 Stackelberg Mirror Descent–Ascent (**SMIDA**), tabular form

Require: Utility u; full-support references π^{ref}, ω^{ref}; regularization weights $\tau^L, \tau^F > 0$; step sizes satisfying (6)

1: Initialize $\pi_0 \leftarrow \pi^{\mathrm{ref}}$ and $\omega_0 \leftarrow \omega^{\mathrm{ref}}$

2: **for** $t = 0, 1, 2, \dots$ **do**

3: Compute $\widehat{r}_t(x, y)$ from (5) for every (x, y)

4: Leader (slow, descent): for every x,

$$\pi_{t+1}(\cdot \mid x) \;\propto\; \pi_t(\cdot \mid x)^{1-\eta^L\tau^L}\, \pi^{\mathrm{ref}}(\cdot \mid x)^{\eta^L\tau^L} \exp\big(-\eta^L\, \widehat{r}_t(x, \cdot)\big)$$

5: Follower (fast, ascent): for every (x, y),

$$\omega_{t+1}(\cdot \mid x, y) \;\propto\; \omega_t(\cdot \mid x, y)^{1-\eta^F\tau^F}\, \omega^{\mathrm{ref}}(\cdot \mid x, y)^{\eta^F\tau^F} \exp\big(\eta^F\, u(x, y, \cdot)\big)$$

6: **end for**

At each step t, the Leader maintains a probability vector $\pi_t(\cdot \mid x)$ for every context x, and the Follower a probability vector $\omega_t(\cdot \mid x, y)$ for every context–action pair (x, y). Given the current Follower policy ω_t, the Leader faces the conditional cost

$$\widehat{r}_t(x, y) \coloneqq \sum_{y' \in \mathcal{Y}} \omega_t(y' \mid x, y) u(x, y, y') - \tau^F \,\mathrm{KL}\big(\omega_t(\cdot \mid x, y) \,\|\, \omega^{\mathrm{ref}}(\cdot \mid x, y)\big)\,, \tag{5}$$

that is, the Follower's regularized utility at (x, y) evaluated at its current policy rather than at its best response.

Let $\eta^L, \eta^F > 0$ be the step sizes. The relative timescales are determined by the effective steps $\eta^L\tau^L$ and $\eta^F\tau^F$. We require

$$0 < \eta^L\tau^L < \eta^F\tau^F \leq 1, \tag{6}$$

so that the Follower, which solves the inner maximization, is the fast player. When $\tau^L = \tau^F$, this reduces to $\eta^F > \eta^L$. Appendix C.2 further discusses the faster-Follower timescale and shows how the Follower's convergence controls the error in the conditional costs used by the Leader.

The proportionality signs in Algorithm 1 denote normalization over the displayed action argument. Each update first forms a geometric mixture of the current policy and its reference, then exponentially reweights this mixture toward lower conditional costs for the Leader or higher utilities for the Follower. We next establish last-iterate convergence of **SMIDA** to the regularized Stackelberg equilibrium of Theorem 4.2. Both players converge linearly in both directions of the KL divergence, the Follower with factor $1 - \eta^F\tau^F$ and the Leader with the slower factor $1 - \eta^L\tau^L$.

Theorem 4.3 (Last iterate convergence of **SMIDA**). *Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are finite, that ρ, π^{ref}, and ω^{ref} have full support, and that $\tau^L, \tau^F > 0$. Run Algorithm 1 with*

$$0 < \eta^L\tau^L < \eta^F\tau^F \leq 1.$$

Then every iterate has full support, and the last iterate (π_t, ω_t) converges to the unique Stackelberg equilibrium (π^*, ω^*). More precisely, there exist constants $C_L, C_F > 0$ such that, for every $t \geq 1$, the Leader satisfies

$$\max\left\{\mathrm{KL}_\rho(\pi^* \,\|\, \pi_t), \mathrm{KL}_\rho(\pi_t \,\|\, \pi^*)\right\} \leq C_L \left(1 - \eta^L\tau^L\right)^t,$$

while the Follower satisfies

$$\max\left\{\mathrm{KL}_{\rho,\pi^*}(\omega^* \,\|\, \omega_t), \mathrm{KL}_{\rho,\pi^*}(\omega_t \,\|\, \omega^*)\right\} \leq C_F \left(1 - \eta^F\tau^F\right)^t.$$

Appendix C proves a stronger statement, with bounds that hold uniformly over contexts and conditioning actions, and gives explicit expressions for C_L and C_F.

4.3 Scaling SMIDA to Language Model Policies

Motivated by **SMIDA**, we introduce **DEEP-SMIDA**, a neural policy-gradient algorithm designed for large context and action spaces $\mathcal{X}$ and $\mathcal{Y}$, with applications to LLMs in mind. We parameterize

the Leader's and Follower's policies as π_θ and ω_ϕ, respectively, and optimize their regularized objectives using batched policy-gradient estimates. Both estimates are computed from the same Leader–Follower rollouts, with the Leader taking a descent step and the Follower an ascent step.

We highlight one implementation detail that is, to our knowledge, novel: jointly warming up the Leader's learning rate and the Follower-to-Leader learning-rate ratio. Preliminary experiments showed that training was highly sensitive to the learning rates η^L and η^F and often became unstable; standard warm-up techniques did not resolve this instability (Kalra & Barkeshli, 2024). To address this, we linearly increase the ratio η_t^F/η_t^L from 1 to its target $\eta^F/\eta^L > 1$, while warming up the Leader's learning rate η_t^L linearly from η_{init}^L to η^L. At each iteration, the Follower's learning rate is the product of the current Leader rate and the current ratio. Both rates remain constant after warm-up, with targets chosen to satisfy $\eta^L\tau^L < \eta^F\tau^F$, following the timescale ordering in the tabular analysis of Theorem 4.3. Algorithmic and implementation details are provided in Appendix D.

5 Experiments

5.1 Methodology

We use UltraFeedback (Cui et al., 2024) and adopt a synthetic oracle setup (Gao et al., 2023) to label the original response pairs and comparisons generated by the adversarial game. Relabelling gives $89{,}625$ preference triplets across $15{,}053$ unique prompts. We train an ensemble of five preference models based on the architecture of Zhang et al. (2025), using mutually disjoint training partitions at the prompt level and independent initialization and training seeds. We reserve $37{,}614$ triplets across $6{,}315$ unseen prompts for evaluation. More details are to be found in Appendix E, while in Appendix I we provide an heuristic argument as to why preference models are more subject to hacks than reward models.

5.2 Variance as an Estimate for Prediction Difficulty

We test whether ensemble variance is an informative proxy for prediction error (Gleave & Irving, 2022) on the held-out prompt set of Section 5.1. For each triplet, we compare the preference induced by the five-member ensemble's mean probability with the binary oracle label. Prediction difficulty here refers to the ensemble's tendency to disagree with the oracle, whereas $\mathrm{Var}(x, y, y')$ requires no oracle label and is available during policy training.

Ensemble accuracy decreases monotonically across variance deciles, from 92.0% in the lowest to 59.4% in the highest (Figure 1a). Since the Follower searches for high-variance triplets, error concentration in the upper tail is especially relevant: the top 10% have an error rate of 40.6%, compared with 24.3% overall (Figure 1b). Across comparisons, variance achieves an area under the receiver operating characteristic curve (AUROC) of 0.670 when ranking errors above correct predictions (Fawcett, 2006), indicating moderate discrimination. Other uncertainty metrics are provided in Appendix F.

5.3 Generating Adversarial Responses

We next ask whether the Leader–Follower game generates adversarial preference triplets on held-out prompts. Following Section 3.2, we compare the trained and reference policies by measuring the frequency of α-online adversarial candidates (Definition 3.2) and, using oracle labels, ε-adversarial triplets (Definition 3.1).

All policies are initialized from the same base model, using the models and chat templates of Appendix E.4; the frozen initializations serve as π^{ref} and ω^{ref}. The utility is the ensemble variance of Section 3.1, rescaled to $[0, 1]$ and computed using the frozen five-member ensemble of Section E.2. We train with **DEEP-SMIDA** (Algorithm 2) on a set of prompts unseen during ensemble training, obtaining policies $(\widehat{\pi}, \widehat{\omega})$. Appendix E.4 provides hyperparameters and implementation details. We evaluate the trained pair $(\widehat{\pi}, \widehat{\omega})$ and the reference pair $(\pi^{\text{ref}}, \omega^{\text{ref}})$ on the same $2{,}500$ prompts, unseen during policy and ensemble training, with identical decoding settings. Each policy pair generates two response pairs per prompt, for a total of $5{,}000$ triplets. The empirical variance distribution of the reference triplets estimates F_{Var} and sets common candidate thresholds for both policy pairs.

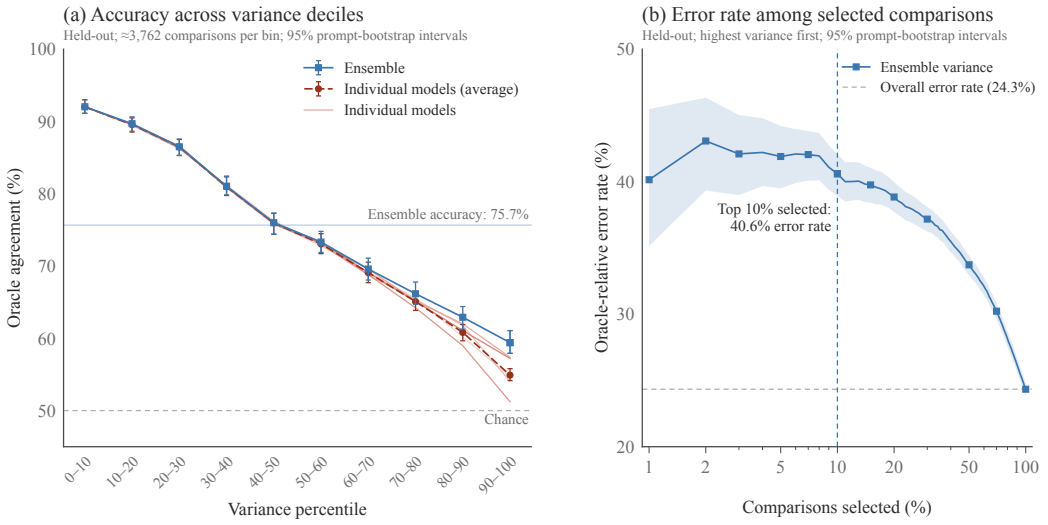

Figure 1: Ensemble variance and prediction difficulty on held out comparisons. **(a)** Oracle agreement across variance deciles. **(b)** Ensemble error rate among comparisons selected in decreasing order of variance. Error bars and shading show 95% prompt-bootstrap intervals.

5.3.1 Disagreement on Holdout Prompts

We first assess whether the trained policies generate comparisons with greater ensemble disagreement on holdout prompts. Across the 5,000 triplets generated by each policy pair, training increases mean ensemble variance from 0.0051 to 0.0139 (95% prompt-bootstrap interval: $[0.0133, 0.0145]$). The median rises from 0.00238 to 0.00569, and the trained pair produces higher mean variance on 68.5% of evaluation prompts. The increase thus extends to the centre of the distribution and across prompts. Figure 2(a) shows how this shift translates into the yield of α-online adversarial candidates, using common variance thresholds estimated from the reference generations. At $\alpha = 0.1$, the trained pair gives 33.7% candidates (95% interval: $[31.7\%, 35.5\%]$), compared with 10.0% for the reference pair. The relative gain increases as α decreases. At $\alpha = 0.05$ and $\alpha = 0.01$, trained gives are 26.0% and 11.2%, respectively, or 5.2 and 11.2 times the reference rates. Training thus increases the yield of comparisons with high disagreement over the reference.

5.3.2 Adversarial Triplets

Greater disagreement is useful for our purpose when it identifies comparisons on which the ensemble's mean preference is incorrect. We test this empirically on 1,600 triplets from 800 holdout prompts per policy pair, using the same synthetic oracle as in Section 5.1. For each triplet, we estimate $\mathcal{P}^*(y \succ y' \mid x)$ by averaging the valid binary verdicts from 11 oracle queries, randomizing response order in each query. We apply the conditions of Definition 3.1. Figure 2(b) reports the fraction of all labelled triplets that are both α-online candidates and ε-adversarial. This joint yield measures how many oracle-confirmed errors pass the online criterion per generated triplet.

At $\alpha = 0.1$ and $\varepsilon = 0$, the trained pair gives 9.6% (95% prompt-bootstrap interval: $[7.97, 11.25]$), compared with 4.4% for the reference pair. At $\alpha = 0.05$, the respective yields are 8.0% (95% interval: $[6.51, 9.50]$) and 1.9%. The advantage persists for reversals with substantial oracle margins: at $\alpha = 0.1$ and $\varepsilon = 0.3$, the trained pair yields 7.0% against 2.6%. These results support the game as a way to generate adversarial comparisons accessible to online selection.

The online criterion also concentrates errors within the trained policies' generations. At $\alpha = 0.1$, 29.7% of their candidates reverse the preferences, compared with 19.4% of their generations overall, capturing 49.2% of all their reversals. The reference pair nevertheless has a higher overall reversal rate (34.3%) and the scope of this simple example is thus specific. We obtain a higher per-triplet yield of errors exceeding a disagreement threshold, with errors defined relative to the oracle.

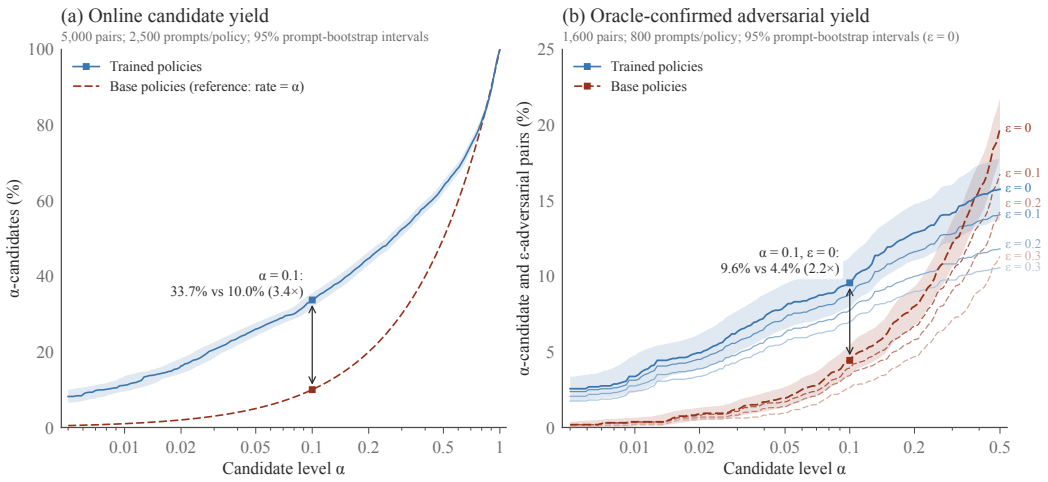

Figure 2: Online candidate yield and oracle-confirmed adversarial yield on holdout prompts. **(a)** Fraction of $5{,}000$ triplets per policy pair meeting the common reference-quantile threshold at level α. **(b)** Fraction of all $1{,}600$ oracle-labelled triplets per policy pair that are both α-online candidates and ε-adversarial. Blue solid curves denote trained policies; red dashed curves denote reference policies. Lighter curves indicate larger oracle margins. Shading shows 95% prompt-bootstrap intervals

5.4 Further Experiments

We briefly report two further experiments, with details in Appendix G. First, we examine an indirect benefit of our method, the Leader's alignment with preferences. We conjecture that minimizing disagreement against an moving Follower may help preserve response quality. In fact, in a round-robin evaluation, our Leader achieves an oracle preference rate of 54.89% against NLHF (Munos et al., 2024) and 50.70% against the SLHF Leader (Pásztor et al., 2026). A Leader trained only to resist disagreement thus remains competitive with policies trained to be preferred (Appendix G.4).

To isolate the role of the Leader, we train the Follower only, while keeping the Leader fixed in (1). We conjecture that a Follower facing a fixed Leader can exploit idiosyncratic patterns of that Leader's response distribution, rather than learn to construct useful responses. We therefore compare the two Followers on IFEval (Zhou et al., 2023), an instruction following benchmark. The Follower trained with a moving Leader achieves higher accuracy on all four aggregate metrics. Training against an adaptive Leader thus discourages strategies that induce disagreement at the expense of instruction following (Appendix G.5).

6 Conclusions, Broader Impact, and Limitations

The goal of the present work is first and foremost to provide a framework for addressing a basic, almost naïve difficulty in alignment, that the evaluator is by necessity an imperfect proxy for human preferences and intents. Our Stackelberg formulation uses adversarial interaction to search for comparisons that expose these imperfections and provide candidates for additional supervision. Due to compute constraints, our experiments are limited to small models and relatively simple tasks, with errors assessed against a synthetic oracle. Whether correcting the errors identified by this search makes the evaluator more robust remains to be established. A natural next step is therefore to connect adversarial search with repeated annotation and evaluator updates, and to test the resulting process on larger frontier models and on open-ended, long-horizon tasks whose outcomes are difficult to verify. More broadly, we hope that the ensemble-variance game illustrates a broader point, that the same optimizing behavior that exploits an evaluator can also help improve the same supervision on which alignment depends.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experimental design, implement methods, cleaning and formatting the datasets, and check mathematical arguments. We also used a language model as a synthetic preference oracle to label response comparisons, as described in the experimental methodology. We have not used AI tools to propose or refine hypotheses, to interpret experimental results, or to set the direction of the research. Additionally, we used generative AI tools to identify, summarize, and analyze relevant literature, write and refine software code, create or modify scientific figures, and polish and contribute to the manuscript's writing and editing. The models used include GPT-5.6 Sol, GPT-6 Astra (OpenAI) and Fable 5, Fable 5.1, Opus 5, Opus 5.5 (Anthropic). We, the authors, take full responsibility for the final contents of this work, including text, claims or artifacts produced with the aid of generative AI tools.

ETHICS STATEMENT

We do not foresee significant ethical concerns arising from the experiments reported in this work. Our experiments involve no human annotators, all preference labels come from a synthetic oracle, applied to prompts from publicly available datasets. The errors we report are therefore defined relative to this oracle, which may or may not have biases of its own, and should not be read as departures from human judgment. The framework is intended to identify weaknesses in preference evaluators and to direct additional supervision toward them, with the broader aim of making alignment more reliable.

REPRODUCIBILITY STATEMENT

All data used in this work are publicly available or will be made public. All code will be released upon acceptance. We refer the reader to the implementation Section 5.1 and Appendix D and E for details of the experimental setup, training procedures, and hyperparameters.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Charalambos D. Aliprantis and Kim C. Border. *Infinite dimensional analysis: a hitchhiker's guide*. Springer, 2006.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Oxford, 2014. ISBN 9780199678112.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.

Alexander Bukharin, Haifeng Qian, Shengyang Sun, Adithya Renduchintala, Soumye Singhal, Zhilin Wang, Oleksii Kuchaiev, Olivier Delalleau, and Tuo Zhao. Adversarial training of reward models. In *Second Conference on Language Modeling*, 2025.

Souradip Chakraborty, Amrit Singh Bedi, Alec Koppel, Huazheng Wang, Dinesh Manocha, Mengdi Wang, and Furong Huang. PARL: A unified framework for policy alignment in reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017.

Xu Chu, Zhixin Zhang, Tianyu Jia, and Yujie Jin. Stackelberg self-annotation: A robust approach to data-efficient LLM alignment. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems*, 2025.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning, ICML*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL `https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash`.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*, 2024.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognit. Lett.*, 27(8):861–874, 2006.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning, ICML*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models. *arXiv preprint arXiv:2203.07472*, 2022.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Forty-first International Conference on Machine Learning, ICML*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Dhawal Gupta, Adam Fisch, Christoph Dann, and Alekh Agarwal. Mitigating preference hacking in policy optimization with pessimism. *arXiv preprint arXiv:2503.06810*, 2025.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems*, 2024.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *Deep Reinforcement Learning Meets Structured Prediction, ICLR*, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017.

Kimin Lee, Laura M. Smith, and Pieter Abbeel. PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory*, 37: 145–151, 1991.

Tianyi Lin, Chi Jin, and Michael I. Jordan. Two-timescale gradient descent ascent algorithms for nonconvex minimax optimization. *J. Mach. Learn. Res.*, 26:11:1–11:45, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*, 2019.

Jacob Makar-Limanov, Arjun Prakash, Denizalp Goktas, Nora Ayanian, and Amy Greenwald. STA-RLHF: Stackelberg aligned reinforcement learning with human feedback. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2024.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's Law. *arXiv preprint arXiv:1803.04585*, 2018.

Kenneth O May. Intransitivity, utility, and the aggregation of preference patterns. *Econometrica: Journal of the Econometric Society*, 22:1–13, 1954.

Davit Melikidze, Marian Schneider, Jessica Lam, Martin Wertich, Ido Hakimi, Barna Pásztor, and Andreas Krause. ActiveUltraFeedback: Efficient preference data generation using active learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, Seoul, South Korea, 2026.

METR. Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. Technical report, METR, August 2026. URL `https://metr.org/hugging-face-incident-report-aug-2026.pdf`.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Forty-first International Conference on Machine Learning, ICML*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning, ICML*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

OpenAI. OpenAI–Hugging Face incident technical report. Technical report, OpenAI, August 2026. URL `https://cdn.openai.com/pdf/67869394-cb91-4c12-888c-5cbd85c7814c/OpenAI-Hugging-Face%20Incident-Technical-Report.pdf`.

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

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Barna Pásztor, Thomas Kleine Buening, and Andreas Krause. Stackelberg learning from human feedback: Preference optimization as a sequential game. In *The Fourteenth International Conference on Learning Representations*, 2026.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, 2023.

Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019. ISBN 9780525558613.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT*, pp. 287–294. ACM, 1992.

Marwan Simaan and Jose B. Cruz. On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 2022.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In *Forty-first International Conference on Machine Learning, ICML*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Amos Tversky. Intransitivity of preferences. *Psychological review*, 76(1):31–48, 1969.

Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Julius Springer, Wien and Berlin, 1934. English translation: *Market Structure and Equilibrium*, trans. D. Bazin, L. Urch and R. Hill, Springer, 2011.

Norbert Wiener. Some moral and technical consequences of automation: As machines learn they may develop unforeseen strategies at rates that baffle their programmers. *Science*, 131(3410): 1355–1358, 1960.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. In *The Thirteenth International Conference on Learning Representations, ICLR*, 2025.

Yi Xu, Laura Ruis, Tim Rocktäschel, and Robert Kirk. Investigating non-transitivity in LLM-as-a-judge. In *Forty-second International Conference on Machine Learning, ICML*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond Bradley-Terry models: A general preference model for language model alignment. In *Forty-second International Conference on Machine Learning, ICML*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Extended Literature Review

Alignment. Aligning language models with human preferences requires eliciting those preferences and translating them into a learning signal. Reinforcement learning from human feedback (RLHF) learns scalar rewards from human comparisons and optimizes policies against them (Christiano et al., 2017; Ouyang et al., 2022). A scalar reward induces a transitive ordering, whereas pairwise whereas aggregated human judgments and LLM judges are routinely intransitive (May, 1954; Tversky, 1969; Xu et al., 2025). General pairwise preference models accommodate such intransitivities (Zhang et al., 2025). Nash learning from human feedback (NLHF) (Munos et al., 2024) and its descendants (Swamy et al., 2024; Wu et al., 2025) seek to compute a Nash equilibrium of simultaneous play by optimizing regularized preference win rates under a fixed preference model.

Reward Hacking. A learned evaluator is only an imperfect proxy for human preferences and optimizing against it can increase its scores without producing outcomes that humans prefer (Amodei et al., 2016; Skalse et al., 2022). Stronger optimization can exacerbate this mismatch across control tasks and language models, increasing proxy reward while reducing true (synthetic) reward (Pan et al., 2022; Gao et al., 2023); general preference models remain vulnerable (Gupta et al., 2025).

Active Learning. Active preference learning selects comparisons for annotation using uncertainty scores, such as predictive entropy (Muldrew et al., 2024), or Thompson sampling over fixed response pools (Melikidze et al., 2026). RUNE (Liang et al., 2022) instead adds ensemble standard deviation to the agent's reward, optimizing a policy toward disagreement. Adv-RM (Bukharin et al., 2025) similarly trains a generator to maximize disagreement between two reward models while keeping the target model's reward high. We optimize disagreement directly, training two opposing policies against an ensemble without this directional constraint. By generating comparisons as the policies evolve, our approach also searches for failure modes that fixed response pools may miss.

Stackelberg Learning. Prior Stackelberg alignment methods optimize rewards while anticipating policy adaptation (Chakraborty et al., 2024), adapt a preference model to a leading policy (Makar-Limanov et al., 2024), or optimize against worst-case preference-gap distributions (Chu et al., 2025). Closest to our formulation, Stackelberg learning from human feedback (SLHF) (Pásztor et al., 2026) trains a Follower to generate a preferred alternative conditional on the Leader's realized response.

B Existence and Uniqueness of Stackelberg Equilibria

This Appendix contains the proof(s) of the existence and uniqueness Theorem 4.2. We do so in steps. First, we establish equilibrium existence for continuous convex-concave objectives. We then recall a variational identity for KL-regularized optimization and use it to derive the equilibrium policies in closed form and prove uniqueness when both regularization parameters are strictly positive.

Begin by observing that as $\mathcal{X}$ and $\mathcal{Y}$ are finite, the strategy sets Π and Ω are nonempty convex compact subsets of finite-dimensional Euclidean spaces. For the reader convenience, we repeat the definition of Stackelberg Equilibrium in a slightly more general form. We also note that the specific form of (4) is, in fact, not needed to prove existence.

Definition B.1. Let $f : \Pi \times \Omega \to \mathbb{R}$ be a function. A pair (π^*, ω^*) is a f-Stackelberg Equilibrium for the zero-sum game with utility f if

$$\pi^* \in \arg\min_{\pi\in\Pi} \max_{\omega\in\Omega} f(\pi, \omega), \quad \omega^* \in \arg\max_{\omega\in\Omega} f(\pi^*, \omega).$$

Theorem B.2. *Let $f : \Pi \times \Omega \to \mathbb{R}$ be continuous convex-concave[3]. Then there exists (π^*, ω^*) such that*

$$f(\pi^*, \omega) \le f(\pi^*, \omega^*) \le f(\pi, \omega^*), \quad \forall \pi \in \Pi, \omega \in \Omega.$$

In particular, (π^*, ω^*) is a Stackelberg equilibrium and

$$\max_{\omega} \min_{\pi} f(\pi, \omega) = \min_{\pi} \max_{\omega} f(\pi, \omega).$$

Proof. Define the lower and upper value of the zero-sum game by

$$\underline{v} \coloneqq \max_{\omega} \min_{\pi} f(\pi, \omega), \quad \overline{v} \coloneqq \min_{\pi} \max_{\omega} f(\pi, \omega).$$

The immediate inequality gives $\underline{v} \le \overline{v}$ and by the VnM-Sion maxmin Theorem Sion (1958) we have equality $\underline{v} = \overline{v}$; denote the common value by v. By compactness of Π and Ω and by continuity of the maps $\omega \mapsto \min_\pi f(\pi, \omega)$ and $\pi \mapsto \max_\omega f(\pi, \omega)$, the latter which follows by an application of Berge's Maximum Theorem (see Theorem 17.31 of Aliprantis & Border (2006)), we can choose $\pi^* \in \Pi$ and $\omega^* \in \Omega$ such that

$$\min_{\pi} f(\pi, \omega^*) = v, \quad \max_{\omega} f(\pi^*, \omega) = v.$$

In particular, by the first condition we have that $f(\pi^*, \omega^*) \ge v$ and symmetrically from the second we have $f(\pi^*, \omega^*) \le v$, hence equality. Combining the previous two equalities gives the saddle property

$$f(\pi^*, \omega) \le f(\pi^*, \omega^*) \le f(\pi, \omega^*), \quad \forall \pi \in \Pi, \omega \in \Omega.$$

By the left-hand inequality, $\omega^* \in \arg\max_{\omega\in\Omega} f(\pi^*, \omega)$. Finally,

$$\max_{\omega} f(\pi^*, \omega) = f(\pi^*, \omega^*) = v = \min_{\pi} \max_{\omega} f(\pi, \omega),$$

which gives $\pi^* \in \arg\min_{\pi\in\Pi} \max_{\omega\in\Omega} f(\pi, \omega)$ as required. □

[3]Such a function f is said to be convex-concave if for all $\omega \in \Omega$ the section $f(\cdot, \omega)$ is convex and similarly for all $\pi \in \Pi$ the section $f(\pi, \cdot)$ is concave.

Note that the function $f : \Pi \times \Omega \to \mathbb{R}$ as defined in Equation (4) is continuous and convex-concave, hence satisfies the assumptions of Theorem B.2; this proves the first part of Theorem 4.2. We next exploit the explicit structure of (4) to prove uniqueness and to obtain an explicit expression for the equilibrium policies. The following Lemma is a well known folklore result, of which, for completeness sake, we report a simple proof. For similar arguments, see also Rafailov et al. (2023).

Lemma B.3. *Let A be a finite set, let $r \in \Delta(A)$ be of full support, let $\tau \in \mathbb{R}_{>0}$ and $h : A \to \mathbb{R}$. Then*

$$\min_{q \in \Delta(A)} \{\langle q, h\rangle + \tau \,\mathrm{KL}(q \,\|\, r)\} = -\tau \log Z, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes the expectation and Z is the normalizing constant

$$Z \coloneqq \sum_{a \in A} r(a) e^{-h(a)/\tau}. \tag{8}$$

Moreover, the minimizer is unique and is given by

$$q^*(a) = \frac{r(a) e^{-h(a)/\tau}}{Z}. \tag{9}$$

Proof. Since r has full support, q^* as defined in (9) also has full support. For any $q \in \Delta(A)$,

$$\begin{aligned} \mathrm{KL}(q \,\|\, q^*) &= \sum_{a \in A} q(a) \log \left(\frac{q(a)}{q^*(a)} \right) \\ &= \mathrm{KL}(q \,\|\, r) + \frac{1}{\tau} \langle q, h \rangle + \log Z. \end{aligned}$$

Rearranging gives

$$\langle q, h\rangle + \tau \,\mathrm{KL}(q \,\|\, r) = -\tau \log Z + \tau \,\mathrm{KL}(q \,\|\, q^*). \tag{10}$$

Because the Kullback–Leibler divergence is nonnegative, the right-hand side is minimized at q^*, with minimum value $-\tau \log Z$. Moreover, $\mathrm{KL}(q \,\|\, q^*) = 0$ if and only if $q = q^*$, which proves uniqueness. □

Theorem B.4. *Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are finite and that ρ, π^{ref}, and ω^{ref} have full support. Let f be defined as in* (4)*. Suppose that $\tau^L, \tau^F > 0$ and $\pi^{\mathrm{ref}}, \omega^{\mathrm{ref}}$ are of full support. Then there exists a unique f-Stackelberg Equilibrium (π^*, ω^*) given by*

$$\omega^*(y' \mid x, y) = \frac{\omega^{\mathrm{ref}}(y' \mid x, y) \exp\big(u(x, y, y')/\tau^F\big)}{Z^F(x, y)},$$

$$\pi^*(y \mid x) = \frac{\pi^{\mathrm{ref}}(y \mid x) \exp\Big(-\frac{\tau^F}{\tau^L} \log Z^F(x, y)\Big)}{Z^L(x)},$$

where the normalization functions Z^F and Z^L are defined as

$$Z^F(x, y) \coloneqq \sum_{z \in \mathcal{Y}} \omega^{\mathrm{ref}}(z \mid x, y) \exp\big(u(x, y, z)/\tau^F\big),$$

$$Z^L(x) \coloneqq \sum_{z \in \mathcal{Y}} \pi^{\mathrm{ref}}(z \mid x) \exp\big(-(\tau^F/\tau^L) \log Z^F(x, z)\big).$$

Proof. Fix a Leader's policy $\pi \in \Pi$. The Follower's problem is equivalent to

$$\max_{\omega \in \Omega} \mathbb{E}_{x \sim \rho(\cdot), y \sim \pi(\cdot|x)} [\Phi_{x,y}(\omega(\cdot \mid x, y)], \tag{11}$$

where we define

$$\Phi_{x,y}(\nu) \coloneqq \mathbb{E}_{y' \sim \nu(\cdot)}[u(x, y, y')] - \tau^F \,\mathrm{KL}(\nu \,\|\, \omega^{\mathrm{ref}}(\cdot \mid x, y)).$$

We solve the maximization problem (11) pointwise for each $(x, y) \in \mathcal{X} \times \mathcal{Y}$. By Lemma B.3, we have

$$\max_{\nu \in \Delta(\mathcal{Y})} \Phi_{x,y}(\nu) = \tau^F \log Z^F(x, y),$$

with unique maximizer

$$\omega^*(y' \mid x, y) = \frac{\omega^{\mathrm{ref}}(y' \mid x, y) \exp\big(u(x, y, y')/\tau^F\big)}{Z^F(x, y)},$$

where the normalization function Z^F is defined as

$$Z^F(x, y) \coloneqq \sum_{y' \in \mathcal{Y}} \omega^{\mathrm{ref}}(y' \mid x, y) \exp\left(\frac{u(x, y, y')}{\tau^F}\right).$$

It follows that ω^* maximizes $f(\pi, \cdot)$ for every π. For a fixed π, this maximizer is unique at every point (x, y) such that $\rho(x)\pi(y \mid x) > 0$. Substituting the Follower's optimal value into the Leader's problem gives

$$\begin{aligned} V(\pi) &\coloneqq \max_{\omega \in \Omega} f(\pi, \omega) \\ &\coloneqq \sum_{x \in \mathcal{X}} \rho(x) \left[\sum_{y \in \mathcal{Y}} \pi(y \mid x)\tau^F \log Z^F(x, y) + \tau^L \mathrm{KL}\big(\pi(\cdot \mid x) \,\|\, \pi^{\mathrm{ref}}(\cdot \mid x)\big)\right]. \end{aligned} \tag{12}$$

The minimization of V separates across contexts. For each $x \in \mathcal{X}$, apply Lemma B.3 again, now with

$$Z^L(x) \coloneqq \sum_{y \in \mathcal{Y}} \pi^{\mathrm{ref}}(y \mid x) \exp\left(-\frac{\tau^F}{\tau^L} \log Z^F(x, y)\right).$$

Lemma B.3 shows that the minimum of the x-indexed term in (12) is $-\tau^L \log Z^L(x)$, and that its unique minimizer is

$$\pi^*(y \mid x) = \frac{\pi^{\mathrm{ref}}(y \mid x) \exp\left(-\frac{\tau^F}{\tau^L} \log Z^F(x, y)\right)}{Z^L(x)}.$$

Since ρ has full support, every context-indexed term in (12) has strictly positive weight. Hence π^* is the unique minimizer of V, and, in particular,

$$\pi^* \in \arg\min_{\pi \in \Pi} \max_{\omega \in \Omega} f(\pi, \omega).$$

Moreover, since π^{ref} has full support, so does π^*. Thus, for every (x, y), we have $\rho(x)\pi^*(y \mid x) > 0$. By the preceding argument, this implies that ω^* is the unique maximizer of $f(\pi^*, \cdot)$, completing the proof. □

C Convergence of Stackelberg Mirror Descent–Ascent

This Appendix proves a stronger version of Theorem 4.3, on the last-iterate convergence of **SMIDA**. We establish explicit bounds in both directions of the KL divergence for each player, uniformly over contexts and, for the Follower, conditioning actions.

C.1 Uniform Last Iterate Convergence

For a function $v : A \to \mathbb{R}$, write $\mathrm{osc}(v) := \max_a v(a) - \min_a v(a)$ for the oscillation of v. Define the two payoff oscillations

$$D_F \coloneqq \max_{x,y} \mathrm{osc}\left(u(x, y, \cdot)\right) = \max_{x,y} \left[\max_{y'} u(x, y, y') - \min_{y'} u(x, y, y')\right],$$

$$D \coloneqq \max_{x} \mathrm{osc}\left(u(x, \cdot, \cdot)\right) = \max_{x} \left[\max_{y,y'} u(x, y, y') - \min_{y,y'} u(x, y, y')\right],$$

The first is the variation relevant to the Follower's tracking error; the second also allows the Leader action to vary, and satisfies $D_F \le D$.

Theorem C.1. *Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are finite, that ρ, π^{ref}, and ω^{ref} have full support, and that $\tau^L, \tau^F > 0$. Run Algorithm 1 under condition* (6)*, and let (π^*, ω^*) be the unique regularized Stackelberg equilibrium. Set*

$$q_L := 1 - \eta^L \tau^L, \qquad q_F := 1 - \eta^F \tau^F, \qquad \kappa := \frac{\eta^F \tau^F}{\eta^L \tau^L} > 1,$$

and define

$$r(x,y) := \tau^F \log \sum_{y'} \omega^{\mathrm{ref}}(y' \mid x, y) \exp\left(\frac{u(x,y,y')}{\tau^F}\right), \qquad D_L := \max_x \operatorname{osc} r(x,\cdot).$$

Then all iterates have full support and, for every $t \geq 0$,

$$\max_{x,y} \max\{\mathrm{KL}(\omega^*(\cdot \mid x,y) \| \omega_t(\cdot \mid x,y)), \mathrm{KL}(\omega_t(\cdot \mid x,y) \| \omega^*(\cdot \mid x,y))\} \leq q_F^t \frac{D_F}{\tau^F}, \tag{13}$$

$$\max_{x} \max\{\mathrm{KL}(\pi^*(\cdot \mid x) \| \pi_t(\cdot \mid x)), \mathrm{KL}(\pi_t(\cdot \mid x) \| \pi^*(\cdot \mid x))\} \leq q_L^t \frac{D_L}{\tau^L} + \eta^L D_F \frac{q_L^t - q_F^t}{q_L - q_F}. \tag{14}$$

Moreover, $D_L \leq D$, and hence the simpler bound

$$\max_x \max\{\mathrm{KL}(\pi^*(\cdot \mid x) \| \pi_t(\cdot \mid x)), \mathrm{KL}(\pi_t(\cdot \mid x) \| \pi^*(\cdot \mid x))\} \leq q_L^t \frac{D}{\tau^L} \frac{\kappa}{\kappa - 1} \tag{15}$$

also holds. Thus the Follower converges at rate q_F^t and the Leader at rate q_L^t, simultaneously in both directions of KL and uniformly over all contexts.

Proof. Observe that, by their definition in Algorithm 1, all iterates have full support. We make use of the following inequality:

$$\max\{\mathrm{KL}(p\|q), \mathrm{KL}(q\|p)\} \leq \operatorname{osc}(\log p - \log q) \tag{16}$$

for full-support distributions p, q. Indeed, if $v := \log p - \log q$, normalization implies $\min v \leq 0 \leq \max v$. Therefore $\mathrm{KL}(p\|q) = \mathbb{E}_p[v] \leq \max v \leq \operatorname{osc}(v)$, and the reverse direction follows identically.

For the Follower, let $v_t^{x,y} := \log \omega_t(\cdot \mid x,y) - \log \omega^*(\cdot \mid x,y)$, where ω^* is the unique equilibrium policy, as per Theorem B.4. Taking logarithms in the Follower update gives

$$v_{t+1}^{x,y} = q_F v_t^{x,y} + c_t^{x,y} \mathbf{1}$$

for a scalar normalizing constant $c_t^{x,y}$ and $\mathbf{1}$ the vector of ones. Since oscillation is invariant under addition of constants,

$$\operatorname{osc}(v_t^{x,y}) = q_F^t \operatorname{osc}(v_0^{x,y}) = q_F^t \frac{\operatorname{osc} u(x,y,\cdot)}{\tau^F} \leq q_F^t \frac{D_F}{\tau^F}. \tag{17}$$

Combining this with (16) proves (13).

From the explicit form of ω^*, we have $\log(\omega^*/\omega^{\mathrm{ref}}) = u/\tau^F - \log Z^F$. Hence,

$$\tau^F \mathrm{KL}(\omega_t \| \omega^*) = \tau^F \mathrm{KL}(\omega_t \| \omega^{\mathrm{ref}}) - \mathbb{E}_{\omega_t}[u] + \tau^F \log Z^F.$$

Using $r = \tau^F \log Z^F$ and $\widehat{r}_t = \mathbb{E}_{\omega_t}[u] - \tau^F \mathrm{KL}(\omega_t \| \omega^{\mathrm{ref}})$ gives $r - \widehat{r}_t = \tau^F \mathrm{KL}(\omega_t \| \omega^*)$. Equations (16) and (17) imply, uniformly in (x,y),

$$0 \leq r(x,y) - \widehat{r}_t(x,y) \leq D_F q_F^t.$$

For the Leader, put $m_t^x := \log \pi_t(\cdot \mid x) - \log \pi^*(\cdot \mid x)$. Taking logarithms in the Leader update gives

$$m_{t+1}^x = q_L m_t^x + \eta^L (r(x,\cdot) - \widehat{r}_t(x,\cdot)) + d_t^x \mathbf{1}$$

for another scalar normalizer d_t^x. Because $r - \widehat{r}_t \geq 0$, we have $\operatorname{osc}(r - \widehat{r}_t)(x,\cdot) \leq \max_y (r - \widehat{r}_t)(x,y)$. Hence

$$\max_x \operatorname{osc}(m_{t+1}^x) \leq q_L \max_x \operatorname{osc}(m_t^x) + \eta^L D_F q_F^t.$$

Since $\pi_0 = \pi^{\mathrm{ref}}$,

$$\max_x \operatorname{osc}(m_0^x) = \frac{D_L}{\tau^L}.$$

Expanding Equation (C.1) therefore gives

$$\max_x \operatorname{osc}(m_t^x) \le q_L^t \frac{D_L}{\tau^L} + \eta^L D_F \sum_{k=0}^{t-1} q_L^{t-1-k} q_F^k,$$

and the finite sum equals $(q_L^t - q_F^t)/(q_L - q_F)$. Applying (16) proves (14). Finally, $r(x, y)$ lies between $\min_{y'} u(x, y, y')$ and $\max_{y'} u(x, y, y')$, so $D_L \le D$. Since $q_F < q_L$,

$$\sum_{k=0}^{t-1} q_L^{t-1-k} q_F^k \le \frac{q_L^t}{\eta^F \tau^F - \eta^L \tau^L},$$

which gives (15). □

Note that Theorem 4.3 follows by averaging the uniform bounds of Theorem C.1. Indeed, by definition,

$$\mathrm{KL}_\rho(\pi \| \widetilde{\pi}) = \sum_x \rho(x) \, \mathrm{KL}\big(\pi(\cdot \mid x) \| \widetilde{\pi}(\cdot \mid x)\big),$$

and

$$\mathrm{KL}_{\rho,\pi^*}(\omega \| \widetilde{\omega}) = \sum_{x,y} \rho(x) \pi^*(y \mid x) \, \mathrm{KL}\big(\omega(\cdot \mid x, y) \| \widetilde{\omega}(\cdot \mid x, y)\big).$$

The weights in each sum are nonnegative and sum to one. Thus, applying the uniform bounds in Theorem C.1 separately to both directions of the KL divergence gives, for every $t \ge 1$,

$$\max\{\mathrm{KL}_\rho(\pi^* \| \pi_t), \mathrm{KL}_\rho(\pi_t \| \pi^*)\} \le \frac{D}{\tau^L} \frac{\kappa}{\kappa - 1} (1 - \eta^L \tau^L)^t,$$

and

$$\max\{\mathrm{KL}_{\rho,\pi^*}(\omega^* \| \omega_t), \mathrm{KL}_{\rho,\pi^*}(\omega_t \| \omega^*)\} \le \frac{D_F}{\tau^F} (1 - \eta^F \tau^F)^t.$$

Consequently, the strictly positive constants required in the main theorem may be chosen as

$$C_L = \max\left\{1, \frac{D}{\tau^L} \frac{\kappa}{\kappa - 1}\right\}, \qquad C_F = \max\left\{1, \frac{D_F}{\tau^F}\right\}.$$

These constants are finite because the action and context sets are finite, $\tau^L, \tau^F > 0$, and $\kappa > 1$. The step-size condition ensures that both geometric factors lie in $[0, 1)$, so the bounds imply last-iterate linear convergence; full support of every iterate is already established in Theorem 4.3.

C.2 The Follower's Timescale and the Leader's Approximation Error

The choice of a faster Follower in (6) follows the approach adopted in SLHF (Pásztor et al., 2026). The underlying intuition is standard in two-timescale optimization: the inner player approaches its best response while the outer player changes more slowly, allowing the latter's updates to approximate optimization of the maximized objective (Lin et al., 2025). In our tabular setting, the Follower's objective at each (x, y) is independent of the Leader's policy. The faster timescale therefore serves to reduce the error in the conditional values $\widehat{r}_t$ used by the Leader.

The role of the Follower's approximation error can be made explicit. Let $r(x, y) \coloneqq \tau^F \log Z^F(x, y)$ denote the Follower's optimal regularized value at (x, y). The following lemma identifies the difference between this value and the current conditional value $\widehat{r}_t(x, y)$.

Lemma C.2. *Under the assumptions of Theorem B.1, for every $t \ge 0$ and every $(x, y) \in \mathcal{X} \times \mathcal{Y}$,*

$$r(x, y) - \widehat{r}_t(x, y) = \tau^F \, \mathrm{KL}\left(\omega_t(\cdot \mid x, y) \,\|\, \omega^*(\cdot \mid x, y)\right) \ge 0.$$

Proof. Fix (x, y). Apply the identity established in the proof of Lemma B.3 with $h = -u(x, y, \cdot)$, reference distribution $\omega^{\mathrm{ref}}(\cdot \mid x, y)$, and regularization parameter τ^F. The unique minimizer is $\omega^*(\cdot \mid x, y)$ and the minimum is $-r(x, y)$. Evaluating the identity (10) at $\omega_t(\cdot \mid x, y)$ therefore gives

$$-\widehat{r}_t(x, y) = -r(x, y) + \tau^F \,\mathrm{KL}\left(\omega_t(\cdot \mid x, y) \,\|\, \omega^*(\cdot \mid x, y)\right).$$

Rearranging proves the equality. Nonnegativity follows from that of KL divergence. □

Thus, convergence of the Follower directly controls the error in the costs used by the Leader.

D Deep-smida: Policy Gradients and Implementation Details

This Appendix contains algorithmic and implementation details for **DEEP-SMIDA** (see Algorithm 2). We parametrize the Leader's and Follower's policies by π_θ and ω_ϕ, respectively. Denote by ρ the full support distribution over contexts and by $\pi^{\mathrm{ref}}, \omega^{\mathrm{ref}}$ the corresponding reference policies.

Motivated by the objective in Equation (4) and the procedure described in Algorithm 1, we write the Follower's exact objective and Leader's plug-in objective, respectively, as

$$J^F(\theta, \phi) \coloneqq \mathbb{E}_{x\sim\rho(\cdot), y\sim\pi_\theta(\cdot|x), y'\sim\omega_\phi(\cdot|x,y)}[u(x, y, y')] - \tau^F \,\mathrm{KL}_{\rho,\pi_\theta}(\omega_\phi \,\|\, \omega^{\mathrm{ref}}), \tag{18}$$

$$J^L(\theta) \coloneqq \mathbb{E}_{x\sim\rho(\cdot),\ y\sim\pi_\theta(\cdot|x)}\left[\widehat{r}(x, y)\right] + \tau^L \,\mathrm{KL}_\rho\left(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\right), \tag{19}$$

where following Equation (5), $\widehat{r}_\phi$ denotes the plug-in cost currently faced by the Leader:

$$\widehat{r}_\phi(x, y) \coloneqq \mathbb{E}_{y'\sim\omega_\phi(\cdot|x,y)}[u(x, y, y')] - \tau^F \,\mathrm{KL}_{\rho,\pi_\theta}(\omega_\phi(\cdot \mid x, y) \,\|\, \omega^{\mathrm{ref}}(\cdot \mid x, y))$$

Note that the two players optimize in opposite directions, the follower maximizes J^F, whereas the leader minimizes J^L.

For any baselines $b^F(x, y)$ and $b^L(x)$ that do not depend on the sampled Follower and Leader actions, respectively, the policy gradients can be written as

$$\nabla_\phi J^F = \mathbb{E}_{x,y,y'}\left[\nabla_\phi \log \omega_\phi(y' \mid x, y)\left(u(x, y, y') - \tau^F \log \frac{\omega_\phi(y' \mid x, y)}{\omega^{\mathrm{ref}}(y' \mid x, y)} - b^F(x, y)\right)\right], \tag{20}$$

$$\nabla_\theta J^L = \mathbb{E}_{x,y}\left[\nabla_\theta \log \pi_\theta(y \mid x)\left(\widehat{r}_\phi(x, y) + \tau^L \log \frac{\pi_\theta(y \mid x)}{\pi^{\mathrm{ref}}(y \mid x)} - b^L(x)\right)\right]. \tag{21}$$

The constant terms produced by differentiating the KL regularizers can be absorbed into the baselines because the expected score of a policy is zero.

We next give unbiased Monte Carlo estimators of these gradients. Fix a context $x \sim \rho$. Draw $D \geq 2$ independent Leader samples $y^1, \ldots, y^D \sim \pi_\theta(\cdot \mid x)$ and, for every $l \in \{1, \ldots, D\}$, draw $M \geq 2$ independent Follower samples $y'^{l1}, \ldots, y'^{lM} \sim \omega_\phi(\cdot \mid x, y^l)$. In practice, the resulting estimators are additionally averaged over a minibatch of B contexts. Define

$$A^{lj} \coloneqq u(x, y^l, y'^{lj}) - \tau^F \log \frac{\omega_\phi(y'^{lj} \mid x, y^l)}{\omega^{\mathrm{ref}}(y'^{lj} \mid x, y^l)}, \qquad \overline{A}^l \coloneqq \frac{1}{M}\sum_{j=1}^{M} A^{lj}. \tag{22}$$

For each Follower sample, use the leave-one-out (Kool et al., 2019) baseline $b^F_{-j}(x, y^l) \coloneqq (M-1)^{-1}\sum_{k\neq j} A^{lk}$. The Follower gradient estimator is then

$$\widehat{\nabla_\phi J^F} \coloneqq \frac{1}{D}\sum_{l=1}^{D}\frac{1}{M}\sum_{j=1}^{M}\nabla_\phi \log \omega_\phi(y'^{lj} \mid x, y^l)\left(A^{lj} - b^F_{-j}(x, y^l)\right). \tag{23}$$

Conditionally on (x, y^l), the baseline $b^F_{-j}(x, y^l)$ is independent of y'^{lj}. It therefore reduces variance without changing the expectation of the estimator. The same rollouts also provide the Leader estimator; no additional sampling is required. In particular, $\overline{A}^l$ is a conditionally unbiased estimate of $\widehat{r}(x, y^l)$. Define

$$L^l \coloneqq \overline{A}^l + \tau^L \log \frac{\pi_\theta(y^l \mid x)}{\pi^{\mathrm{ref}}(y^l \mid x)}, \qquad b^L_{-l}(x) \coloneqq \frac{1}{D-1}\sum_{m\neq l} L^m. \tag{24}$$

The resulting Leader gradient estimator is

$$\widehat{\nabla_\theta J^L} := \frac{1}{D}\sum_{l=1}^{D} \nabla_\theta \log \pi_\theta(y^l \mid x)\left(L^l - b^L_{-l}(x)\right). \tag{25}$$

Here $b^L_{-l}(x)$ is independent of y^l conditional on x, so the Leader estimator is also unbiased. Averaging Equations (23) and (25) over a minibatch of contexts preserves this property.

At iteration t, the Follower takes an ascent step and the Leader a descent step:

$$\phi_{t+1} = \phi_t + \eta_t^F \widehat{\nabla_\phi J^F}, \qquad \theta_{t+1} = \theta_t - \eta_t^L \widehat{\nabla_\theta J^L}. \tag{26}$$

In practice we apply these updates through AdamW optimizers (Loshchilov & Hutter, 2019) with η_t^L and η_t^F as their learning rates and no weight decay; each gradient estimate is clipped to unit norm before the optimizer step. Both policies are initialized from their corresponding reference policies, which remain frozen throughout training. Algorithm 2 summarizes the complete procedure: for every minibatch, it samples the nested Leader–Follower rollouts, constructs both gradient estimators, and applies the two updates in Equation (26).

D.1 Learning Rate Warm-up.

As discussed in Section 4.3, directly starting from the target two-timescale learning rates can make training unstable. Let T_{warm} be the number of warm-up iterations. We linearly warm the Leader's rate from η^L_{init} to its target value η^L and, simultaneously, the ratio of the Follower and Leader rates from 1 to its target value η^F/η^L:

$$\eta_t^L = \eta^L_{\text{init}} + \min\left\{\frac{t}{T_{\text{warm}}}, 1\right\}\left(\eta^L - \eta^L_{\text{init}}\right),$$
$$\eta_t^F = \eta_t^L\left[1 + \min\left\{\frac{t}{T_{\text{warm}}}, 1\right\}\left(\frac{\eta^F}{\eta^L} - 1\right)\right].$$

Thus, the two rates are equal at initialization, while after warm-up they equal their target values η^L and η^F. We choose the targets so that $\eta^L\tau^L < \eta^F\tau^F$, matching the two-timescale ordering of Algorithm 1.

D.2 Computational Complexity

We provide a brief analysis of the computational complexity of Algorithm 2. Each iteration of Algorithm 2 samples B contexts $\{x_i\}_{i\in\{1,2,\ldots,B\}}$, D Leader responses $\{y_i^\ell\}_{l\in\{1,\ldots,D\}}$ per context x_i, and M Follower responses $\{y_i^{\prime\ell d}\}_{d\in\{1,\cdots,M\}}$ per Leader response y_i^ℓ. It therefore generates BD Leader responses and BDM Follower responses, for a total of BDM response pairs. From (23), we observe that estimating $\widehat{\nabla_\phi J^F}$ requires a forward-backward pass through DM response trajectories and also DM forward passes from the frozen reference. We, notice however that the D rollouts of the leader can be reused. Hence, from , estimating $\widehat{\nabla_\theta J^L}$, only requires D forward-backward passes through the leader and$D$ forward passes from the frozen reference. The number of *NFE* or Neural Function Evaluations is slightly higher for the follower, due to its higher context length. Further, each of the DM response pairs also requires evaluations from each of the $k = 5$ preference ensemble. Multiplying this by the mini-batch size, that for each optimizer step, we require $BDMk + 3BDM + 3BD$ rollouts.

E Experimental Setup and Implementation Details

This appendix provides details of the experimental setup used in Section 5. We describe dataset construction and oracle labelling, the architecture and training of the preference ensemble, the distribution of ensemble variance, and the implementation of the Leader–Follower game.

Algorithm 2 Deep Stackelberg Mirror Descent–Ascent (**DEEP-SMIDA**)

Require: Utility u; context distribution ρ; reference policies π^{ref} and ω^{ref}; regularization weights $\tau^L, \tau^F > 0$; minibatch size B; sample sizes $D, M \geq 2$; step sizes $\{\eta_t^L, \eta_t^F\}_{t\geq 0}$

Initialize θ_0 and ϕ_0 from their corresponding reference policies, which remain frozen throughout training
for $t = 0, 1, 2, \ldots$ **do**
  Draw a minibatch $\mathcal{B}_t = \{x_i\}_{i=1}^B$ with $x_i \sim \rho$
  for $i = 1, \ldots, B$ **do**
    Draw D independent Leader samples $y_i^1, \ldots, y_i^D \sim \pi_{\theta_t}(\cdot \mid x_i)$
    for $\ell = 1, \ldots, D$ **do**
      Draw M independent Follower samples $y_i'^{\ell 1}, \ldots, y_i'^{\ell M} \sim \omega_{\phi_t}(\cdot \mid x_i, y_i^\ell)$
      Compute $A_i^{\ell j}$, $\overline{A}_i^\ell$, and $b_{-j}^F(x_i, y_i^\ell)$ according to (22)
    end for
    Form the per-context Follower gradient estimator $\widehat{\nabla_\phi J^F}(x_i)$ according to (23)
    Using the same rollouts, compute L_i^ℓ and $b_{-\ell}^L(x_i)$ according to (24)
    Form the per-context Leader gradient estimator $\widehat{\nabla_\theta J^L}(x_i)$ according to (D.2)
  end for
  Average the per-context estimators over $\mathcal{B}_t$ to obtain $\widehat{\nabla_\phi J^F}_t$ and $\widehat{\nabla_\theta J^L}_t$
  Update the Follower by ascent and the Leader by descent as in (26):

$$\phi_{t+1} = \phi_t + \eta_t^F \widehat{\nabla_\phi J^F}_t, \qquad \theta_{t+1} = \theta_t - \eta_t^L \widehat{\nabla_\theta J^L}_t.$$

end for

E.1 Dataset

We use a subset of UltraFeedback (Cui et al., 2024), a preference dataset containing approximately 64k prompts collected from six public instruction sources, each answered by four language models drawn from a pool of seventeen. We retain the prompts and their responses, discard empty and duplicate responses, and expand each prompt x into all $\binom{N_x}{2}$ unordered pairs of its N_x distinct responses. Each pair defines a preference triplet (x, y, y').

Following Gao et al. (2023), we adopt a synthetic oracle setup, using `DeepSeek-V4-Flash` (DeepSeek-AI, 2026) as the gold preference oracle. We replace UltraFeedback's original annotations with oracle labels so that the original response pairs and comparisons subsequently generated by the adversarial game share a common source of supervision. For each comparison, we obtain a binary preference label through zero-shot prompting, randomizing the display order of the two responses to mitigate position bias. The procedure gives a dataset $\mathcal{D}$ of $89{,}625$ labelled preference triplets across $15{,}053$ unique prompts.

E.2 Preference Models

We train an ensemble of $|I| = 5$ preference models $\{\mathcal{P}_{\theta_i}\}_{i\in I}$, using a simplified version of the general preference model of Zhang et al. (2025). Each member is initialized from the same `Qwen2.5-1.5B-Instruct` backbone (Qwen Team, 2024) and finetuned independently. Given a preference triplet (x, y, y'), we encode each prompt-response pair $(x, y), (x, y')$ separately, project its final-token hidden representation through a bias-free linear head into a 16-dimensional vector, and apply L^2 normalization, obtaining an embedding $z_{\theta_i}(x, y)$. We compare the resulting embeddings through a fixed skew-symmetric bilinear form $J \in \mathbb{R}^{16\times 16}$, rescale the score by a learned positive scalar λ_{θ_i} shared across inputs, and apply a sigmoid σ to obtain the preference probability. The preference probability is

$$\mathcal{P}_{\theta_i}(y \succ y' \mid x) \coloneqq \sigma\big(\lambda_{\theta_i} \langle J z_{\theta_i}(x, y), z_{\theta_i}(x, y')\rangle\big). \tag{27}$$

This construction ensures that $\mathcal{P}_{\theta_i}(y \succ y' \mid x) + \mathcal{P}_{\theta_i}(y' \succ y \mid x) = 1$, while allowing intransitive preferences. By the antisymmetry of the preference models and basic properties of the variance, we have $\mathrm{Var}(x, y, y') = \mathrm{Var}(x, y', y)$; thus, the order of the two responses within a preference triplet does not affect the optimization objective.

Table 1: Distribution of ensemble variance on the held out UltraFeedback subset.

Summary statistics		Selected percentiles	
Statistic	Value	Percentile	Value
Mean	0.006739	25	0.000588
Median	0.002931	75	0.008959
Variance	9.0507×10^{-5}	90	0.018687
Standard deviation	0.009514	95	0.026660
Maximum	0.100228	99	0.044667

Each ensemble member is trained by minimizing the negative log-likelihood $-\log \mathcal{P}_{\theta_i}(y_w \succ y_l \mid x)$ of the oracle labels on its own training partition of $\mathcal{D}$, where y_w and y_l denote the oracle-preferred and non-preferred responses. Each partition contains approximately $10{,}000$ preference triplets across $1{,}680$ unique prompts. The partitions are mutually disjoint at the prompt level and all labelled comparisons associated with a training prompt belong to the same partition. Ensemble diversity comes from these distinct training sets, independent random initializations of the linear heads, and different training seeds.

All members share a validation set whose prompts are disjoint from every training partition. A further $37{,}614$ triplets across $6{,}315$ prompts are excluded from both training and validation and serve as the held out set of Sections 5.2. On this set, individual members agree with the gold preference oracle on 74.1–75.2% of comparisons, while predictions obtained from the ensemble-mean preference probability agree on 75.7%.

E.3 Distribution of Ensemble Variance

We examine the distribution of $\mathrm{Var}(x, y, y')$ on the held out UltraFeedback subset, comprising $37{,}614$ response pairs across $6{,}315$ prompts. For each pair, we compute the population variance of the five members' predicted preference probabilities, using denominator $|I| = 5$. All descriptive statistics weight response pairs equally.

Table 1 summarizes the distribution. Most comparisons elicit little ensemble disagreement, while a smaller fraction forms a pronounced right tail: the mean is more than twice the median, and the 99th percentile is approximately fifteen times the median. Only 8.93% of comparisons have variance greater than 0.02. Consistent with the relationship shown in the main text, mean variance is higher on comparisons where the ensemble disagrees with the synthetic oracle (0.010010) than where it agrees (0.005687). These statistics describe disagreement within the trained ensemble and its association with oracle error on this held out subset.

E.4 Stackelberg Game Implementation

The Leader and Follower are both initialized from `Qwen2.5-1.5B-Instruct` (Qwen Team, 2024) and finetuned during the Stackelberg game. Their frozen initializations serve as the reference policies π^{ref} and ω^{ref}.

Table 2 summarizes the training configuration. The utility is $4\,\mathrm{Var}(x, y, y')$, where variance is computed across the five frozen ensemble members with denominator 5, giving a score in $[0, 1]$.

During warm-up, we linearly increase the Leader's learning rate from 10^{-6} to η^L and the Follower-to-Leader learning-rate ratio from 1 to $\eta^F/\eta^L = 2.5$. The Follower's learning rate is their product; both rates remain constant after warm-up. Each optimizer step uses $BDM = 64$ preference triplets. Sampling and policy log-probability calculations use the same temperature, with no nucleus or top-k truncation.

We use the following chat templates, with the Follower receiving the Leader's realized response in its system message.

Table 2: Hyperparameters for training the Leader and Follower. Unless specified otherwise, settings apply to both policies.

Parameter	Value
Optimization	
Optimizer	AdamW
Leader learning rate η^L	8×10^{-6}
Follower learning rate η^F	2×10^{-5}
Initial learning rate (both policies)	10^{-6}
Warm-up steps	50
KL coefficients τ^L, τ^F	5×10^{-4}
Weight decay	0
Maximum gradient norm	1.0
Parameter precision	`bfloat16`
Training and rollout sampling	
Training prompts	2,000
Training epochs	1
Optimizer steps	500
Prompts per step B	4
Leader responses per prompt D	4
Follower responses per Leader response M	4
Generation	
Sampling temperature	0.6
Top-p	1.0
Maximum generated tokens per response	1,500

Leader Template

```
[System]: You are Qwen, created by Alibaba Cloud. You are a helpful
    assistant.

[User]: <USER_PROMPT>
```

Follower Template

```
[System] You are Qwen, created by Alibaba Cloud. You are a helpful
    assistant.

Another assistant previously answered the user's request as follows:

<LEADER_RESPONSE>

Write a different, higher-quality answer to the user's request.
    Respond directly to the user as if answering for the first time.
    Do not mention, quote, or reference the previous answer.

[User]: <USER_PROMPT>
```

F The Utility as a Design Choice

Our framework accommodates any fixed utility u without changing the game structure or update rules, and the tabular guarantees remain valid under the stated assumptions. In our experiments, u is the variance of the preference probabilities predicted by a trained ensemble: the Follower seeks to expose ensemble disagreement, while the Leader seeks to limit it. This is not the only option and exploring alternative utilities is a natural direction for future work.

Preference optimization is another special case. Setting $u(x, y, y') = \mathcal{P}(y' \succ y \mid x)$ recovers an equivalent minimization form of the regularized SLHF Pásztor et al. (2026) objective. Our analysis thus establishes linear last-iterate convergence of **SMIDA** for both players in this preference game. In the finite tabular setting with strictly positive regularization and the stated step-size conditions, this answers affirmatively the question raised in Pásztor et al. (2026) of whether mirror-descent methods can be adapted to obtain last-iterate convergence for Stackelberg preference games.

F.1 Other Uncertainty Metrics

Further examples of utility are motivated by uncertainty measures used for query selection in active preference learning, such as ensemble variance and predictive entropy (Christiano et al., 2017; Lee et al., 2021). We consider here two alternatives to variance: decision uncertainty (Lewis & Gale, 1994) and Jensen–Shannon divergence (JSD) (Lin, 1991). Writing $\mathcal{P}_i(x, y, y')$ for the preference probability predicted by ensemble member i, let $\bar{\mathcal{P}}(x, y, y') = \frac{1}{|I|} \sum_{i \in I} \mathcal{P}_i(x, y, y')$ denote the ensemble mean. We define decision uncertainty as

$$U(x, y, y') \coloneqq \min \left\{ \bar{\mathcal{P}}(x, y, y'),\, 1 - \bar{\mathcal{P}}(x, y, y') \right\} = \frac{1}{2} - \left| \bar{\mathcal{P}}(x, y, y') - \frac{1}{2} \right|,$$

and JSD as

$$\mathrm{JSD}(x, y, y') \coloneqq H\big(\bar{\mathcal{P}}(x, y, y')\big) - \frac{1}{|I|} \sum_{i \in I} H(\mathcal{P}_i(x, y, y'))\,,$$

where $H(p) = -p \log p - (1 - p) \log(1 - p)$ denotes the entropy of a Bernoulli distribution with parameter p. Decision uncertainty measures the proximity of the ensemble mean to indifference and induces the same ranking as its predictive entropy, $H(\bar{\mathcal{P}})$. JSD instead measures disagreement among the predictive distributions of the individual members.

We evaluate both scores on the held-out comparisons of Section 5.2, measuring errors relative to the binary oracle labels. Figure 3(a) reports error-detection AUROCs of 0.707 for decision uncertainty and 0.654 for JSD. Selecting the top 10% of comparisons ranked by each score gives error rates of 47.5% and 39.7%, respectively, compared with 24.3% overall (Figure 3(b)). On this evaluation, decision uncertainty therefore outperforms variance in both AUROC (0.670) and error concentration in the top 10% (40.6%), while JSD gives slightly lower values on both measures.

Better error detection on a fixed set of comparisons, however, does not necessarily make a score a suitable utility for the adversarial game. Decision uncertainty is maximized when the ensemble mean is $1/2$, including when all members agree on indifference. Since the Follower observes the Leader's realized response, it can attain this maximum by copying that response, setting $y' = y$. The antisymmetry of our preference models ensures that $\mathcal{P}_i(x, y, y) = 1/2$ for every member, and hence $U(x, y, y) = 1/2$ for every (x, y). Thus, copying is a best response to every Leader response, and the Leader cannot reduce the Follower's optimal utility below $1/2$. Variance and JSD avoid this particular failure mode, since both vanish when all ensemble members predict $1/2$, including on identical-response comparisons.

F.2 Targeting a Specific Preference Model

Another option is to augment the ensemble disagreement objective in Equation (1) with a preference term. Given a fixed member $\mathcal{P}_{\theta_j}$ of the ensemble and a coefficient $\lambda \geq 0$, consider

$$\min_{\pi} \max_{\omega} \mathbb{E}_{x \sim \rho,\, y \sim \pi(\cdot|x),\, y' \sim \omega(\cdot|x,y)} \left[\mathrm{Var}(x, y, y') + \lambda\, \mathcal{P}_{\theta_j}(y' \succ y \mid x) \right].$$

The additional term serves two purposes. First, it acts as a preference-based regularizer as the Follower is encouraged to generate responses that the selected model prefers to the Leader's response. The Leader, in turn, seeks responses that resist both disagreement and defeat under this model. This introduces a preference signal for optimization, with λ controlling its strength relative to variance. Second, the term allows us to direct the search toward a specific preference model while retaining uncertainty information from the full ensemble. Experiments show that this directed objective achieves comparable performance to the variance objective in generating α-online candidates.

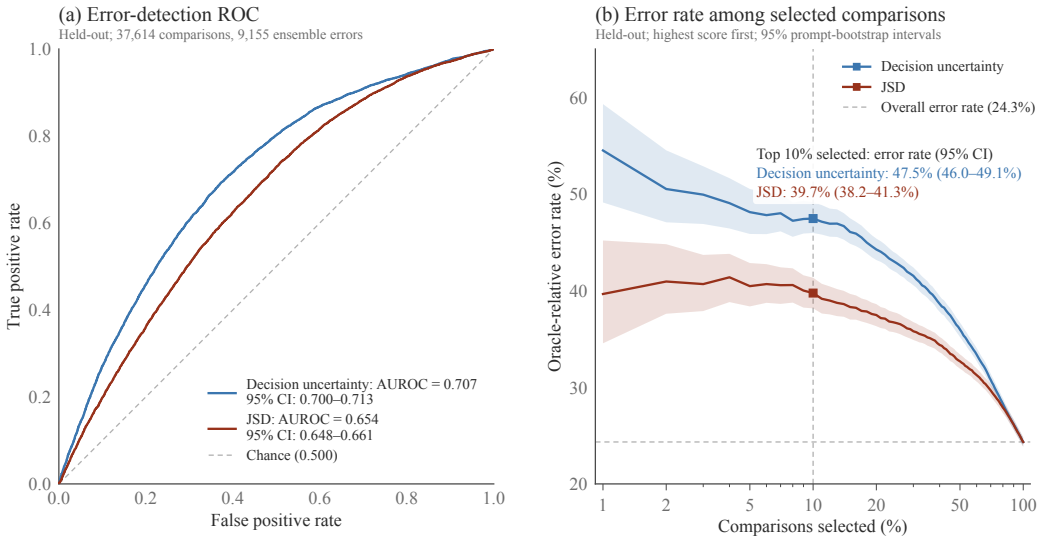

Figure 3: Preference-ensemble error detection on 37,614 held-out comparisons, including 9,155 errors relative to the binary oracle labels. **(a)** ROC curves for decision uncertainty (blue) and Jensen–Shannon divergence (JSD; red), with AUROCs of 0.707 and 0.654, respectively. The dashed diagonal indicates chance performance. **(b)** Oracle-relative error rates among comparisons selected in descending order of each score. Selecting the top 10% gives error rates of 47.5% for decision uncertainty (95% CI: 46.0–49.1%) and 39.7% for JSD (95% CI: 38.2–41.3%), compared with 24.3% overall. Shading shows pointwise 95% prompt-bootstrap intervals.

G Further Experiments, Parameters Sweep, and Ablation

G.1 Training Dynamics

Figure 4 complements the evaluation in Section 5.3. Panel (a) shows the empirical CDF of ensemble variance on held-out prompts, computed from generations obtained after training. The trained pair's distribution shifts toward higher variance relative to the reference pair; vertical lines mark the reference-quantile thresholds defining α-online candidates.

Panel (b) tracks candidate shares in training minibatches of 64 triplets over 500 steps, using these same fixed thresholds. The moving averages begin near the nominal reference rates, rise markedly after approximately 200 steps, and remain elevated through the end of training despite substantial minibatch variability. By the final steps, the smoothed shares are approximately 50% at $\alpha = 0.1$ and 20% at $\alpha = 0.01$. The increased candidate yield is thus sustained during training and is also observed on held-out prompts after training.

For examples of high variance generated preference triplets, see Appendix H.

G.2 Hyperparameter Sweep

We performed a sequential hyperparameter sweep over the KL regularization coefficients, the Follower's target learning rate, and the warm-up duration. We first varied the common regularization coefficient

$$\tau = \tau^L = \tau^F \in \{2 \times 10^{-4},\ 5 \times 10^{-4},\ 10^{-3},\ 2 \times 10^{-3},\ 4 \times 10^{-3}\}.$$

After selecting τ, we swept the Follower's target learning rate $\eta^F \in \{2\times10^{-5},\ 3\times10^{-5},\ 5\times10^{-5}\}$ and the warm-up duration $T_{\text{warm}} \in \{15, 20, 50\}$ optimizer steps, using the joint learning-rate and learning-rate-ratio schedule described in Appendix D.1.

Configuration selection considered both training stability and the fraction of generated triplets satisfying the α-online candidate criterion of Definition 3.2.

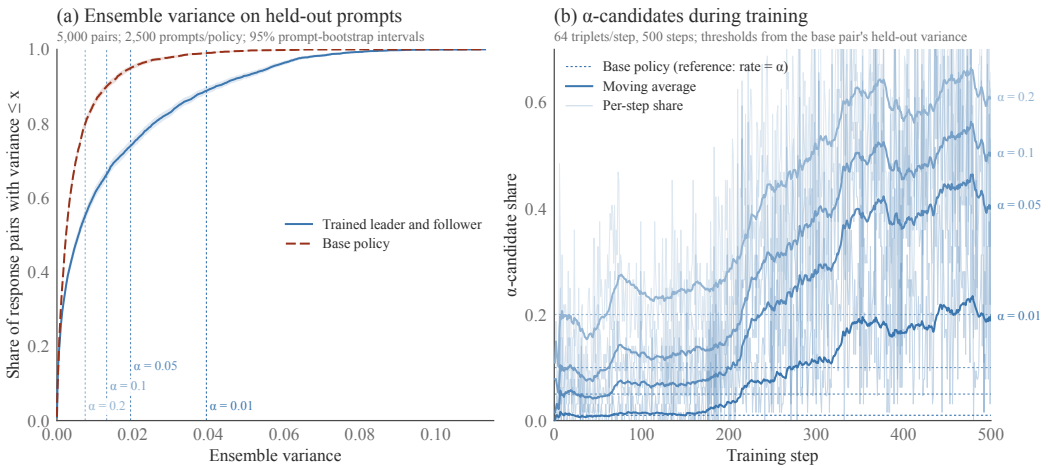

Figure 4: Ensemble disagreement after and during training. **(a)** Empirical CDFs of ensemble variance on 5,000 triplets from 2,500 held-out prompts per policy pair, generated after training. Vertical lines mark reference-quantile thresholds; shading shows 95% prompt-bootstrap intervals. **(b)** Shares of α-online candidates in training minibatches of 64 triplets, using the same fixed thresholds. Thin curves show per-step shares, thick curves their moving averages, and horizontal dotted lines the nominal reference rates α.

G.3 Adversarial Precision of Online Candidates

We further examine whether the online criterion (Definition 3.2) identifies comparisons on which the ensemble's mean preference is incorrect. Using the trained policies' 1,600 oracle-labelled triplets from 800 holdout prompts described in Section 5.3.2, Figure 5 reports the fraction of α-online candidates that are ε-adversarial. Smaller α imposes a stricter disagreement threshold and generally increases this fraction. At $\alpha = 0.1$, 29.7% of candidates reverse the oracle preference ($\varepsilon = 0$), compared with 19.4% of all labeled triplets, a 1.53-fold enrichment over random selection. The same pattern continues for $\varepsilon \in \{0.1, 0.2, 0.3\}$, indicating that the enrichment extends to reversals with larger oracle margins. These results support ensemble disagreement as an informative criterion for prioritizing comparisons for annotation within the trained policies' generations.

G.4 Preference Evaluation of the Trained Leader

We evaluate whether our Leader produces responses that remain competitive under preference evaluation, despite being only trained to minimize ensemble disagreement rather than maximize preference. We compare it with NLHF Munos et al. (2024) and the SLHF Leader Pásztor et al. (2026) in a round-robin evaluation. All three policies are initialized from `Qwen2.5-1.5B-Instruct` and trained on the same 2,000 prompts. NLHF and SLHF optimize preferences under the frozen model $\mathcal{P}_0$, whereas our Leader optimizes the ensemble-variance objective. To control for differences in the choice of trainable parameters, we use full-model fine-tuning for all three methods, with LoRA disabled. Table 3 reports the training and generation settings for the two baselines. We evaluate all three policy pairings on a separate set of 500 held-out prompts, using `DeepSeek-V4-Flash` as the synthetic preference oracle. For each response pair, we query the oracle $k = 11$ times and estimate the preference probability by the fraction of judgments favoring one response over the other. Table 4 reports these win rates, averaged across evaluation prompts.

Both Leaders are preferred to NLHF more often than not: the SLHF Leader achieves a preference rate of 54.92%, and our Leader achieves 54.89%. Their direct comparison is nearly even, with our Leader preferred in 50.70% of oracle judgments.

This competitiveness is an unintended benefit of training our Leader to resist ensemble disagreement. We conjecture that minimizing disagreement against an adaptive Follower, together with regulariza-

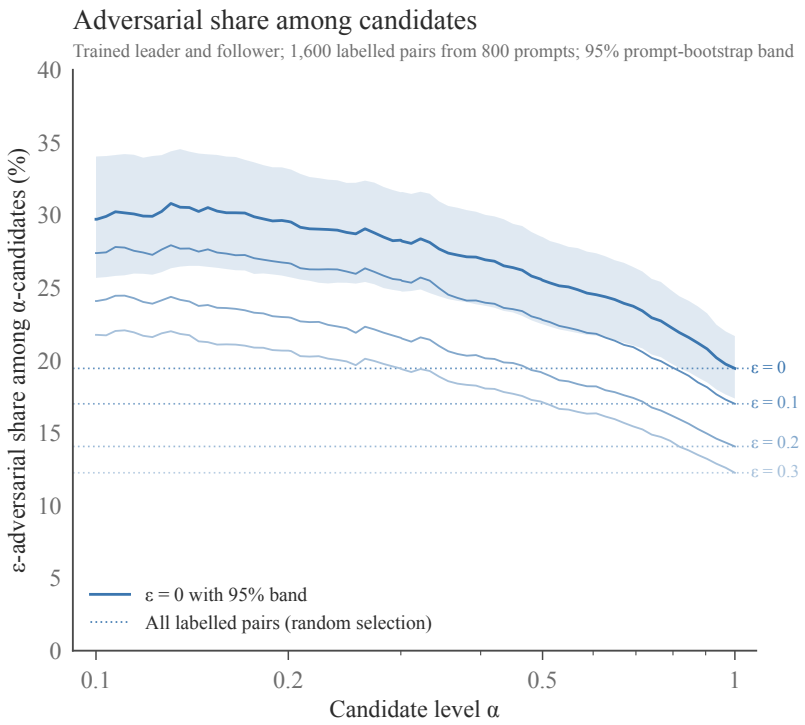

Figure 5: Adversarial precision of online candidates. Solid curves show the fraction of α-online candidates that are ε-adversarial. Candidate thresholds are fixed by the reference variance distribution, as in Section 5 Lighter curves correspond to larger oracle margins. Horizontal dotted lines show the corresponding adversarial fractions among all labelled triplets, giving the expected precision under random selection. Shading shows a 95% prompt-bootstrap interval for $\varepsilon = 0$.

Table 3: Training and generation settings for the NLHF (Nash-MD-PG) and SLHF (StackelbergGDA) baselines. Both methods fully fine-tune `Qwen2.5-1.5B-Instruct` on the same 2,000 training prompts using the frozen preference model $\mathcal{P}_0$. Each uses a separate frozen copy of the initial policy as its KL reference.

Parameter	NLHF	SLHF
Training and rollout sampling		
Training prompts	2,000	2,000
Training epochs	1	1
Optimizer steps	63	63
Prompts per optimizer step	32	32
Prompts in final step	16	16
Responses per prompt	2	2
Generation		
Sampling temperature	0.6	0.6
Top-p	1.0	1.0
Maximum generated tokens per response	512	512

tion toward the initial instruction-tuned policy, may favor responses that retain useful instruction-following behavior.

Table 4: Pairwise oracle preference rates on 500 held-out prompts. Each entry estimates the probability that the row policy's response is preferred to the column policy's response, using 11 oracle queries per response pair and averaging across prompts. Values above 0.5 favor the row policy.

	Column policy		
Row policy	NLHF	SLHF LEADER	OUR LEADER
NLHF	—	0.4508	0.4511
SLHF LEADER	**0.5492**	—	0.4930
OUR LEADER	**0.5489**	**0.5070**	—

G.5 INSTRUCTION FOLLOWING WITH A TRAINED LEADER

We examine the role of the Leader by comparing the Follower trained in the full Stackelberg game with a follower-only baseline, in which the Leader remains fixed. Both Followers seek to maximize ensemble disagreement, but only the former trains against a Leader that learns to resist its attempts. This ablation evaluates whether adapting both sides of the comparison is associated with better instruction following after adversarial training.

We evaluate both Followers on IFEval Zhou et al. (2023), using the same 541 prompts containing 834 verifiable instructions. IFEval uses programmatic checks to assess compliance with explicit requirements, such as including specified keywords, respecting length constraints, or ending with a prescribed phrase. Instruction-level accuracy measures the fraction of individual requirements satisfied, whereas prompt-level accuracy requires all verifiable instructions in a prompt to be satisfied. We report both strict and loose accuracy. Strict evaluation checks the original response; loose evaluation additionally accepts specified transformations, such as removing Markdown asterisks or the first or last line, to reduce false negatives caused by formatting or introductory and concluding text.

Table 5 shows that the Stackelberg Follower achieves higher accuracy on all four aggregate metrics. Loose prompt-level accuracy is 35.12%, compared with 28.28% for the follower-only baseline, and loose instruction-level accuracy is 46.04%, compared with 38.49%. The advantage persists under strict evaluation and also extends across eight of the nine instruction categories.

These results support our motivation for making the search adversarial on both sides of the comparison. With a fixed Leader, the Follower can specialize in inducing disagreement against an unchanging response distribution, potentially exploiting weaknesses specific to that distribution. An adaptive Leader instead learns responses that resist these attempts, requiring the Follower to search against an evolving opponent. The higher instruction-following accuracy is consistent with this interaction discouraging strategies that induce disagreement at the expense of compliance with the prompt.

H EXAMPLE PROMPTS AND RESPONSES

We present selected high variance Leader and Follower comparisons to illustrate the response behaviors encountered during adversarial search. They illustrate unsupported elaboration, discrepancies between reasoning and final-answer correctness, and failures to preserve the requirements of a prompt.

EXAMPLE 1 (UNSUPPORTED ELABORATION)

The Leader directly supplies the name given in the passage. The Follower reproduces that answer but adds a speculative expansion of the initial "S." that the passage does not support. This example illustrates how elaboration can introduce unsupported specificity into an otherwise sufficient answer.

Prompt

```
Quick Question: The chain's origin can be traced to the Dwarf Grill
    (now the Dwarf House), a restaurant opened by S. Truett Cathy,
```

Table 5: Leader ablation on IFEval. Accuracy (%) is reported for the Follower trained with an adaptive Leader (Stackelberg Follower) and with a fixed Leader (follower-only), evaluated on the same 541 prompts and 834 verifiable instructions. The higher value in each row is bolded. Category rows report loose instruction-level accuracy.

METRIC	STACKELBERG FOLLOWER	FOLLOWER-ONLY
Loose: prompt	**35.12**	28.28
Loose: instruction	**46.04**	38.49
Strict: prompt	**30.31**	23.66
Strict: instruction	**41.25**	33.93
Loose accuracy by instruction category		
Case	22.47	**33.71**
Combination	**10.77**	9.23
Detectable content	**50.94**	41.51
Detectable format	**63.69**	49.04
Keywords	**55.83**	53.99
Language	**74.19**	64.52
Length	**44.76**	33.57
Punctuation	**19.70**	16.67
Start/end	**58.21**	28.36

```
    the chain's former chairman and CEO, in 1946. The restaurant is
    located in Hapeville, Georgia, a suburb of Atlanta, and is near
    the location of the now-demolished Ford Motor Company Atlanta
    Assembly Plant, for many years a source of many of the
    restaurant's patrons.

Question: who was the founder of chick fil a?
My answer:
```

Leader Response

```
The founder of Chick-fil-A is S. Truett Cathy.
```

Follower Response

```
To answer the question, let's look at the context provided:

"The chain's origin can be traced to the Dwarf Grill (now the Dwarf
    House), a restaurant opened by S. Truett Cathy, the chain's
    former chairman and CEO, in 1946."

From this sentence, we can see that S. Truett Cathy founded
    Chick-fil-A. The full name given is S. Truett Cathy, which likely
    stands for Stephen Truett Cathy. The word "Cathy" is often used
    as a family name in American English, so we can assume this
    refers to the founder.
```

EXAMPLE 2 (CORRECT FINAL ANSWER WITH FLAWED REASONING)

Under the natural reading that each pack contains 30 utensils split equally among knives, forks, and spoons, each pack contains 10 spoons and the required number of packs is 5. The Leader reaches this final answer through incorrect and internally inconsistent reasoning. The Follower is much more concise, but divides by 3 and returns the incorrect answer 17. This comparison exposes a tension between final-answer correctness and explanation quality.

Prompt

```
John decides to buy utensils. They come in 30 packs with an equal
    number of knives, forks, and spoons. How many packs does he need
    to buy if he wants 50 spoons?
```

Leader Response

```
To find out how many packs John needs to buy, we start by figuring
    out how many teaspoons (knives and forks) are needed in total:

Number of spoons per pack = Number of packs x Number of spoons per
    pack

Since each pack contains an equal number of knives, forks, and spoons:
Teaspoons per pack = 3 spoons/pack

Now calculate the total teaspoons needed:
Total teaspoons needed = Number of spoons wanted x Number of spoons
    per pack
Total teaspoons needed = 50 spoons x 3 spoons/pack
Total teaspoons needed = 150 spoons

Each pack provides 3 spoons, but since there's no specific
    information about the distribution between knives and forks,
    let's assume he needs exactly 30 packs to get 90 spoons for
    simplicity.

Therefore, the number of packs John needs to buy is 150 spoons / 30
    spoons/pack
Number of packs needed = 150 spoons / 30 spoons/pack
Number of packs needed = 5 packs

So, John needs to buy 5 packs of utensils to have 50 spoons.
```

Follower Response

```
He would need 50/3 = 16.67 packs, which rounds up to 17 packs. The
    answer is 17.
```

EXAMPLE 3 (ERRORS MOTIVATED BY A REWRITE)

Both responses classify the original declarative sentence as exclamatory and replace "over 100 million" with "100 million", losing the stated lower bound. The Follower additionally inserts an exclamation mark and claims that its rewrite "maintains the exclamation", although the original sentence contains none. The rewrite makes the mistaken classification appear consistent with the revised text.

Prompt

```
Identify the type of the following sentence, rewrite it in a more
    concise form, and incorporate the budget of the movie as an
    additional constraint. Determine the sentence type and simplify:
    The Tom Cruise movie was a huge hit, with a budget of over 100
    million.
```

Leader Response

```
Sentence Type: Exclamatory Sentence

Simplified Form: "The Tom Cruise movie had an incredible success with
    a staggering budget of 100 million."

Incorporating Budget Constraint:
"The Tom Cruise movie was a massive hit despite its extremely high
    budget of 100 million."
```

Follower Response

```
Type of Sentence: Exclamatory

Concise and Incorporating Budget Constraint:
"Tom Cruise's movie is a huge smash, with an astronomical budget of
    100 million!"

This version maintains the exclamation, simplifies the language, and
    incorporates the budget detail.
```

EXAMPLE 4 (LOSS OF IMPLICIT ALIGNMENT)

The prompt refers to a sentence that is not supplied. The Leader acknowledges this missing input and offers a qualified suggestion. The Follower omits the qualification and substitutes scheduled punctuality for arriving just in time to prevent a negative consequence, which need not describe the same situation.

Prompt

```
Given this sentence about punctuality, suggest a more appropriate and
    precise verb to convey the notion of arriving just in time to
    prevent a negative consequence.
```

Leader Response

```
The original sentence is not provided, but I can offer a suggestion
    based on common scenarios where punctuality might be crucial:

"If you arrive **on** time, you will avoid any negative consequences."

This phrase accurately conveys that being there exactly when needed
    prevents issues from arising.
```

Follower Response

```
"Arriving precisely at the scheduled moment"
```

I WHY GENERAL PREFERENCE MODELS CAN BE MORE PRONE TO REWARD HACKING

General preference models can represent intransitive preferences that scalar reward models cannot capture. This additional expressivity can, however, increase the number of annotations needed to

estimate preferences accurately. We illustrate this tradeoff with a toy example in which the ground-truth preferences admit a scalar reward representation. In this setting, a reward model can pool information across comparisons, whereas an unrestricted preference estimator learns each pair separately. Fix a prompt x and a response set $\mathcal{Y} = y_1, \ldots, y_K$ with $K \geq 2$. Write

$$\mathcal{P}^*_{ij} := \mathcal{P}^*(y_i \succ y_j \mid x),$$

which we assume to be antisymmetric. . For each of the $M = \binom{K}{2}$ unordered pairs, we observe n annotations

$$Y^{(\ell)}_{ij} \sim \text{Bernoulli}(\mathcal{P}^*_{ij}), \qquad i < j, \quad \ell = 1, \ldots, n$$

independently across pairs and annotations, for a total budget of nM observations. Define

$$\bar{Y}ij := \frac{1}{n}\sum_{\ell=1}^{n} Y^{(\ell)}_{ij} \quad \text{for } i < j, \qquad \bar{Y}_{ji} := 1 - \bar{Y}_{ij}, \qquad \bar{Y}_{ii} := \frac{1}{2}.$$

To compare the two estimators under the same ground truth, suppose

$$\mathcal{P}^*_{ij} = \frac{1}{2} + r_i - r_j, \qquad \sum_{i=1}^{K} r_i = 0, \qquad |r_i| \leq \frac{1}{4} \quad \text{for all } i = 1, \ldots, K.$$

where r_i is the reward of response y_i. We measure estimation error in the entrywise supremum norm,

$$\|\hat{\mathcal{P}} - \mathcal{P}^*\|_\infty := \max_{i,j=1,\ldots,K} \left|\hat{\mathcal{P}}_{ij} - \mathcal{P}^*_{ij}\right|.$$

Fix a confidence parameter $\delta \in (0, 1)$.

General preference estimator. Consider the unbiased estimator $\hat{\mathcal{P}}^{\text{pref}}_{ij} := \bar{Y}_{ij}$. For each $i < j$, Hoeffding's inequality gives

$$\mathbb{P}\big(|\bar{Y}_{ij} - \mathcal{P}^*_{ij}| \geq \alpha\big) \leq 2\exp(-2n\alpha^2).$$

Taking a union bound over the M unordered pairs gives

$$\|\hat{\mathcal{P}}^{\text{pref}} - \mathcal{P}^*\|_\infty \leq \sqrt{\frac{\log(2M/\delta)}{2n}} \tag{28}$$

with probability at least $1 - \delta$. This bound does not require the scalar reward representation assumed above.

Reward estimator. Now consider

$$\hat{r}_i := \frac{1}{K}\sum_{j \neq i}\left(\bar{Y}_{ij} - \frac{1}{2}\right).$$

Under the assumed reward representation, this estimator is unbiased:

$$\mathbb{E}[\hat{r}_i] = \frac{1}{K}\sum_{j \neq i}(r_i - r_j) = \frac{1}{K}\left(Kr_i - \sum_{j=1}^{K} r_j\right) = r_i.$$

We estimate preferences by

$$\hat{\mathcal{P}}^{\text{rew}}_{ij} := \left[\frac{1}{2} + \hat{r}_i - \hat{r}_j\right]_{[0,1]},$$

where $[z]_{[0,1]} := \min\{1, \max\{0, z\}\}$ clips the estimate to the probability interval. Consequently,

$$|\hat{\mathcal{P}}^{\text{rew}}_{ij} - \mathcal{P}^*_{ij}| \leq |(\hat{r}_i - r_i) - (\hat{r}_j - r_j)| \leq |\hat{r}_i - r_i| + |\hat{r}_j - r_j|.$$

It therefore suffices to control the reward estimation errors uniformly. For each fixed i, we can write

$$\hat{r}_i - r_i = \frac{1}{nK}\sum_{j \neq i}\sum_{\ell=1}^{n}\left(Y^{(\ell)}_{ij} - \mathcal{P}^*_{ij}\right),$$

where $Y_{ij}^{(\ell)} := 1 - Y_{ji}^{(\ell)}$ for $j < i$. Hoeffding's inequality thus gives

$$\mathbb{P}\Big(|\hat{r}_i - r_i| \geq \frac{\alpha}{2}\Big) \leq 2\exp\Big(-\frac{nK^2\alpha^2}{2(K-1)}\Big).$$

Taking a union bound over the K responses, we obtain

$$\mathbb{P}\Big(\max_{i\in[K]} |\hat{r}_i - r_i| \geq \frac{\alpha}{2}\Big) \leq 2K\exp\Big(-\frac{nK^2\alpha^2}{2(K-1)}\Big).$$

Setting the right-hand side equal to δ and applying the bound on pairwise preference errors gives

$$\|\hat{\mathcal{P}}^{\mathrm{rew}} - \mathcal{P}^*\|_\infty \leq \sqrt{\frac{2(K-1)\log(2K/\delta)}{nK^2}} \tag{29}$$

with probability at least $1 - \delta$.

Comparing Equations (28) and (29), we find that, for fixed δ, the uniform-error upper bound for the general preference estimator is larger by a factor of order $\sqrt{K}$. The reward estimator gains accuracy by pooling information across the $K - 1$ comparisons involving each response, whereas the general preference estimator uses only the n annotations for each pair. Thus, under the same annotation budget of nM, this example illustrates how general preference estimation can admit a larger upper bound on the errors that optimization may exploit.