FAST & EFFICIENT ONLINE ALIGNMENT FOR MULTI-DIMENSIONAL REWARD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider online alignment of language model outputs to user-specific preferences. We assume each user's preferences are driven by an unknown function over a chosen multi-dimensional vector of measurable text qualities (e.g., helpfulness, conciseness, factuality), so that personalisation reduces to learning that function from live feedback. Online adaptation without updating the underlying LLM requires learning this function from sparse and heterogeneous feedback, with no guarantee that any structural assumption about how qualities combine fits every user. Under a KL-regularised objective, a standard identity gives the optimal policy as an exponentially tilted base distribution whose tilt is the user's reward; with it assumed to be a function of qualities, learning the user's preferences therefore reduces to learning that reward from feedback. We cast this as a bandit problem and solve it with Thompson sampling under two posterior families: a linear-$\boldsymbol{\lambda}$ Gaussian belief and a non-parametric Gaussian process (GP). In synthetic experiments, the linear variant converges within tens of interactions where its assumption holds, and the GP variant succeeds on non-linear structures where the linear model degrades. On both the synthetic worlds and a PRISM-derived persona benchmark, achieved reward depends jointly on what the posterior has learned and on how sharply the decoder acts on it. We characterise this dependence and show its role alongside the choice of quality basis in the gap to an in-context baseline.

1 INTRODUCTION

Aligning large language models (LLMs) to human preferences has been one of the central objectives in enabling their use as helpful instruction-following systems. In the standard alignment scenario, an aligned LM produces the same responses across every user, while real preferences are individual and multi-dimensional, and the trade-off among these dimensions shifts across contexts, and these trade-offs are relative and often implicit even to the user themselves. We assume that the LLM can already produce a set of useful candidate responses for a given task, and our goal is to quickly learn which of those a specific user would prefer via noisy feedback signals available at interaction time (scalar rating, thumbs-up, or pairwise preference).

Fine-tuning-based alignment methods such as RLHF (Ziegler et al., 2020; Ouyang et al., 2022), DPO (Rafailov et al., 2024) and KTO (Ethayarajh et al., 2024) are impractical for per-user adaptation at interaction time: per-step gradient updates add substantial latency even with LoRA (Hu et al., 2021); preference-tuning losses overfit on the small number of user interactions ($\sim$tens) (Azar et al., 2023); and for batched online-RLHF (Ziegler et al., 2020) maintaining a per-user fine-tuned model does not scale. Decoding-time methods tilt a frozen base LLM toward a fixed reward model (Mudgal et al., 2024; Khanov et al., 2024) or a fixed combination of reward dimensions (Wang et al., 2024b; Ramé et al., 2023), but commit to a chosen weighting a priori, effectively making an assumption about what the user wants.

In-context learning (Brown et al., 2020) enables conditioning on previous interactions but offers no principled way of incorporating feedback into a belief. Concurrent methods such as T-POP (Qu et al., 2025), AMULET (Zhang et al., 2025), MACO (Dai et al., 2025), PReF (Shenfeld et al., 2025) target the same interaction-time regime, but each commits to a fixed reward parametrisation or feedback channel (§2). None maintains an explicit per-user posterior over how a multi-dimensional quality basis maps to reward under heterogeneous feedback channels.

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.

Our online alignment framework keeps the base LLM frozen (never retrains) and incorporates live user feedback into an explicit posterior over a multi-dimensional quality basis. Under a KL-regularised reward-maximisation objective against the base LLM, the optimal policy has a closed form, an exponential tilt of the base distribution by the user's reward function (Lemma 1). Personalising is equivalent to learning that reward from feedback. We posit that the reward depends on the response only through a vector $\boldsymbol{q}$ of measurable text qualities, and place a Bayesian prior over how $\boldsymbol{q}$ maps to reward. Each round the agent samples a reward from the posterior, selects among K base-LLM samples by SIR (resampling or argmax), observes user's feedback, and updates.

We instantiate the belief in two ways. The linear-$\boldsymbol{\lambda}$ Gaussian belief is constant-size and sample-efficient when the reward is linear in $\boldsymbol{q}$; the Gaussian process captures non-linear shapes through its kernel and is robust when it is not, but its state stores every past observation and grows with the number of interactions.

We evaluate the framework on synthetic preference landscapes spanning linear, multiplicative, and threshold structures and on a PRISM-derived persona benchmark (Kirk et al., 2024). On synthetic worlds, linear-$\boldsymbol{\lambda}$ TS converges to the user-optimal tilt within tens of interactions under scalar, binary, and pairwise feedback, while GP TS extends this to non-linear reward landscapes where the linear model degrades or accumulates negative reward. On the persona benchmark, where each user is impersonated by an LLM scorer fitted to their rating history, both variants accumulate more reward than an ICL baseline on every quality basis. Reward depends on both the learned direction and how sharply the tilt acts on it: argmax widens the lead over ICL along a learned reward but not a random one, and for SIR ($\tau = 1$) the tilt flattens (more uniform) as the posterior concentrates.

On the PRISM benchmark, a frozen fitted model (on five feedback signals) matches the online posterior, and a reward shared by all users does better than both. Our analysis shows that the quality bases does not capture how users differ, thus, the basis rather than the learning procedure limits what can be learned from per-user feedback using PRISM (§5.4).

Contributions. The mathematical machinery we use (the KL-regularised tilt of Lemma 1, linear and GP Thompson sampling) is standard. Our contribution is to bring these pieces together for interaction-time personalisation and characterise it empirically:

- We cast online alignment as a sequential-decision (bandit) problem over a fixed multi-dimensional quality basis $\boldsymbol{q}$ (§3.1); personalisation reduces to inferring how $\boldsymbol{q}$ maps to a user-specific reward, in contrast to decoding-time methods that fix this mapping ahead of time.
- We use two posterior families that share the same Bayesian-bandit interface: a linear-$\boldsymbol{\lambda}$ Gaussian belief with closed-form or tractable updates for scalar, binary, and pairwise feedback (§3.2.1); and a non-parametric Gaussian process with regression updates for scalar feedback and a Laplace-approximation update for binary feedback (§3.2.2).
- We characterise achieved reward as a joint function of what the posterior has learned and of how sharply we act on it, with sharpness measured as the KL the tilt spends; measuring it separates the effect of the tilt from that of the quality basis in the gap to ICL (§5.3, §5.4).

2 Related Work

We organise prior work by when adaptation to user preferences occurs. Training-time methods adapt the model itself from a preference dataset before any user interaction. Test-time methods steer generation at inference toward a fixed reward or user-specific artefact. Interaction-time methods, our regime, update the steering object from feedback observed during the interaction, as in the bandit and online-learning settings (Russo et al., 2018), and often via decoding-time mechanisms.[1]. This is

[1]Our taxonomy is more nuanced than the common use of "test-time" in recent literature, where the term sometimes covers both one-shot decoders and methods that update from feedback at inference. T-POP (Qu et al., 2025) and AMULET (Zhang et al., 2025) identify as test-time, and PReF (Shenfeld et al., 2025) positions itself within "inference-time alignment"; we class all three as interaction-time because they update from per-step user feedback at inference, distinct from one-shot decoders such as MAP (Wang et al., 2024b) or PAD (Chen et al., 2024) that commit to a fixed steering object a priori.

distinct from "online RLHF" (Ziegler et al., 2020), where preference data is gathered during policy training the final learned policy is typically frozen before deployment.

MAP (Wang et al., 2024b) introduces the combined reward-tilting and SIR decoding scheme on which we build, with the reward defined as a linear combination of quality-specific reward models; the user specifies target expected reward levels per dimension and the trade-off weights are solved one-shot from this specification rather than learned from per-step feedback. PReF (Shenfeld et al., 2025) learns the user-specific weights over a linear combination of pretrained reward functions via active pairwise queries. We recover the per-user weights online from heterogeneous feedback (scalar, binary, pairwise) under an explicit Thompson-sampling posterior, and add a non-parametric (GP) instantiation for when the linear assumption breaks.

2.1 Training-time alignment

The dominant paradigm is RLHF (Christiano et al., 2023; Ziegler et al., 2020; Ouyang et al., 2022), which fine-tunes the full LLM against a learned reward model under a KL regulariser. Subsequent methods, such as DPO (Rafailov et al., 2024), IPO (Azar et al., 2023), and KTO (Ethayarajh et al., 2024), drop the explicit reward model. Multi-dimensional variants train the model against a basis of reward components rather than a single scalar: fine-grained RLHF (Wu et al., 2023) and multi-objective DPO (Zhou et al., 2024). SEA (Liu et al., 2024) interleaves training-time fine-tuning with Thompson-sampling queries against an oracle, learning the reward ensemble and updating the policy in the same loop. All produce a single fine-tuned policy that does not adapt after training.

2.2 Test-time alignment

Decoding-time methods tilt generation at inference toward a fixed scalar reward (Stiennon et al., 2022; Mudgal et al., 2024; Khanov et al., 2024; Verdun et al., 2025) or a fixed combination of reward dimensions (Wang et al., 2024b), with the weighting chosen a priori. Rewarded Soups (Ramé et al., 2023) fine-tunes one policy per reward dimension and linearly interpolates them at inference under user-supplied weights; Linear Alignment (Gao et al., 2024) derives a closed-form aligned policy without RLHF fine-tuning or feedback. A complementary thread takes a textual user description as input and uses it to condition the policy: Poddar et al. (2024) maps the description to a latent and conditions a latent-variable reward model on it, while PAD (Chen et al., 2024) conditions the policy directly on the text via a learned head. Drift (Kim et al., 2025) learns per-user weights over an attribute basis from a small offline preference set, but keeps these weights as a fixed estimate and applies them via token-level logit steering. All fix a personalisation artefact (a reward palette, weight vector, latent, or textual descriptor) before interaction, whether user supplied or estimated, and do not refine it from per-step user feedback.

2.3 Interaction-time alignment

In-context learning. ICL (Brown et al., 2020) is the standard non-parametric approach: it conditions on past prompts and feedback through the prompt but offers no principled mechanism for translating accumulated feedback into a calibrated belief about the user. We use it as a baseline (§4).

Bandits. Recovery of a latent user-specific reward can be viewed as a bandit problem: stochastic linear bandits (Chu et al., 2011) model the reward as $\boldsymbol{\lambda}_\star^\top \boldsymbol{\phi}$, and a Gaussian process prior gives a non-parametric generalisation (Srinivas et al., 2010). Thompson sampling (Thompson, 1933; Russo et al., 2018) is the canonical algorithmic recipe across these bandit settings, and the backbone of our method.

LLM-targeted methods. Several recent works apply this template at interaction time. Shin et al. (2025) treat each generated token as a bandit arm. MACO (Dai et al., 2025) runs a phase-elimination bandit over pre-generated completions with a single linear reward shared across users. AMULET (Zhang et al., 2025) drives token-level online learning from a user-supplied textual prompt. T-POP (Qu et al., 2025) learns a per-user black-box scalar reward from dueling-bandit pairwise queries and steers via controlled decoding. PReF (Shenfeld et al., 2025), the closest approach, learns the reward basis offline and then infers per-user weights via active learning on preferences. We differ from each by combining (i) a multi-dimensional quality basis $\boldsymbol{q}$ rather than a black-box scalar

or fixed pretrained-reward basis, (ii) an explicit Thompson-sampling posterior with closed-form or tractable updates, (iii) support for scalar, binary, and pairwise feedback under the same belief representation, and (iv) a non-parametric GP instantiation for non-linear preferences. Drift (Kim et al., 2025), PAD (Chen et al., 2024), MAP (Wang et al., 2024b), PReF (Shenfeld et al., 2025) and T-POP (Qu et al., 2025) configure the sharpness of their tilt by a sweep or a convention and do not measure the KL it spends against the base model.

3 METHODOLOGY

We assume preferences, while not explicitly defined, can be expressed through interpretable text attributes (e.g., conciseness, humour, helpfulness). We cast personalisation as recovering how an unknown reward combines such attributes, either as a linear trade-off or as a more general mapping, and learn it via Thompson sampling. §3.1 sets up the policy class and objective, and §3.2 describes the algorithm and the two structural assumptions on the reward.

3.1 PROBLEM FORMULATION

Let $\boldsymbol{q} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be a known multidimensional feature function whose dimensions quantify particular text qualities. $\boldsymbol{q}(x, y)$ measures the qualitative level of an LLM response $y \in \mathcal{Y}$ to a prompt $x \in \mathcal{X}$. After each generation we observe a feedback signal ρ from the user, governed by a latent unknown reward $r_\star : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that encodes the user's preferences over $\boldsymbol{q}(x, y)$. Modelling assumptions on $r_\star$ and the type of ρ are introduced in §3.2.

Policy class. We restrict the search over language policies to distributions of the form

$$p(y \mid x,\, r) \;=\; \frac{1}{Z(x,r)}\, p_{\text{LLM}}(y \mid x)\, e^{r(x,y)}, \qquad Z(x,r) \doteq \mathbb{E}_{y\sim p_{\text{LLM}}(\cdot|x)}\big[e^{r(x,y)}\big],$$

where $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a reward function drawn from a family $\mathcal{F}$. The induced policy class is $\mathcal{P} \doteq \{p(\cdot \mid x,\, r) : r \in \mathcal{F}\}$. Lemma 1 below justifies the restriction to this class. Once $\hat{r} \in \mathcal{F}$ is estimated, generating aligned completions amounts to sampling from $p(\cdot \mid x,\, \hat{r})$.

Objective. Interaction proceeds over T rounds. At round t, a prompt x_t is presented, the agent selects a scoring function $r_t \in \mathcal{F}$ and samples $y_t \sim p_t(\cdot) \doteq p(\cdot \mid x_t, r_t)$, and the user returns feedback ρ_t. We treat alignment as KL-regularised reward maximisation against the base LLM, with cumulative reward

$$G_T = \sum_{t=1}^{T} \Big(\mathbb{E}_{y_t \sim p_t}\big[r_\star(x_t, y_t)\big] - \mathrm{KL}\big[p_t \,\|\, p_{\text{LLM}}(\cdot \mid x_t)\big] \Big). \tag{1}$$

We equivalently seek to minimise the soft pseudo-regret $R_T = G_T^\star - G_T$, where $G_T^\star$ is the value achieved by the optimal policy $p_\star(y \mid x) \doteq p(y \mid x, r_\star)$. Computing $G_T^\star$ exactly under $r_\star$ is intractable; we approximate it on the same SIR candidate pool as the agent's selection, as detailed in Appendix A.6. The closed form of $p_\star$ is the standard KL-regularised tilt underlying RLHF (Ziegler et al., 2020), DPO (Rafailov et al., 2024), controlled decoding (Mudgal et al., 2024), and MAP (Wang et al., 2024b).

Lemma 1 (KL-regularised optimum) *The distribution maximising equation 1 at each prompt x is*

$$p_\star(y \mid x) \;\propto\; p_{LLM}(y \mid x)\, e^{r_\star(x,y)}.$$

The proof is a short KL manipulation (Appendix A.1).

3.2 PROPOSED SOLUTION

We instantiate the search over $\mathcal{F}$ as Thompson sampling (Thompson, 1933; Russo et al., 2018): the agent maintains a Bayesian belief over $r_\star$, samples a scoring function r_t from it at each round, generates y_t from the induced tilted policy, and updates the belief from the observed feedback ρ_t. The framework is summarised in Algorithm 1. The components that vary across settings are the representation of the belief and the resulting UPDATEBELIEF step; both depend on the structural assumption on $r_\star$ and the channel of ρ. We present two modelling settings: a linear reward (§3.2.1) and a Gaussian process reward (§3.2.2).

Algorithm 1 Thompson sampling for interaction-time alignment

Require: Prior belief over $r_\star$.
1: **for** $t = 1, \dots, T$ **do**
2: $r_t \sim \text{Belief}_{t-1}$ ▷ sample reward from current belief
3: $p_t(\cdot) \leftarrow p(\cdot \mid x_t, r_t)$ ▷ induce tilted policy
4: $y_t \sim p_t(\cdot)$ via SIR ▷ generate completion
5: Observe feedback ρ_t
6: $\text{Belief}_t \leftarrow \textsc{UpdateBelief}(\text{Belief}_{t-1},\, x_t,\, y_t,\, \rho_t)$
7: **end for**

Sampling from the tilted policy. Direct sampling from $p(y \mid x, r)$ is intractable, so we use sampling-importance resampling (SIR) (Smith & Gelfand, 1992; Wang et al., 2024b; Verdun et al., 2025): draw K samples $\{y_t^i\}_{i=1}^K$ from $p_{\text{LLM}}(\cdot \mid x_t)$ and select one according to weight $w_i \propto e^{r(x_t, y_t^i)/\tau}$. The temperature τ sets how sharply the tilt selects: $\tau = 1$ is the optimum of Lemma 1, and $\tau \to 0$ is argmax. In experiments, we measure sharpness as the KL between the tilt and the uniform pool: 0 when uniform, $\log K$ at argmax (Appendix A.13). Appendix A.2 gives background on SIR.

3.2.1 Linear-$\boldsymbol{\lambda}$ reward model

The simplest assumption is that $r_\star$ is a linear combination of qualities (Wang et al., 2024b; Ramé et al., 2023; Wu et al., 2023; Zhou et al., 2024; Shenfeld et al., 2025),[2]

$$r_\star(x, y) = \boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x, y).$$

The induced tilt reduces to

$$p(y \mid x,\, \boldsymbol{\lambda}) \;=\; \frac{1}{Z(x, \boldsymbol{\lambda})}\, p_{\text{LLM}}(y \mid x)\, e^{\boldsymbol{\lambda}^\top \boldsymbol{q}(x,y)}.$$

We place a Gaussian prior $\boldsymbol{\lambda}_\star \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and maintain a Gaussian posterior; at each step the agent samples $\boldsymbol{\lambda}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ and uses it to define the policy $p_t = p(\cdot \mid x_t, \boldsymbol{\lambda}_t)$.

Scalar feedback. With $\rho_t = \boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x_t, y_t) + \varepsilon_t$ and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, Bayes' rule gives the closed-form Gaussian conjugate update

$$\boldsymbol{\Sigma}_t^{-1} = \boldsymbol{\Sigma}_{t-1}^{-1} + \sigma^{-2}\, \boldsymbol{q}(x_t, y_t)\, \boldsymbol{q}(x_t, y_t)^\top,$$
$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t \big(\boldsymbol{\Sigma}_{t-1}^{-1} \boldsymbol{\mu}_{t-1} + \sigma^{-2}\, \boldsymbol{q}(x_t, y_t)\, \rho_t \big).$$

A numerically convenient Woodbury form is given in Appendix A.3.

Binary and pairwise feedback. For $\rho_t \in \{0, 1\}$ we use a logistic likelihood, $p(\rho_t = 1 \mid \boldsymbol{q}_t, \boldsymbol{\lambda}_\star) = \sigma(\boldsymbol{\lambda}_\star^\top \boldsymbol{q}_t)$ with $\boldsymbol{q}_t \doteq \boldsymbol{q}(x_t, y_t)$; for a pair $y_t^{(1)}, y_t^{(2)}$ the Bradley–Terry model (Bradley & Terry, 1952) gives the same likelihood in the quality difference $\hat{\boldsymbol{q}}_t \doteq \boldsymbol{q}(x_t, y_t^{(1)}) - \boldsymbol{q}(x_t, y_t^{(2)})$. Both posteriors are approximated as Gaussian by assumed-density filtering, with the same rank-one form as the scalar update (Appendix A.4).

3.2.2 Gaussian process reward model

A more general assumption is that $r_\star$ is an arbitrary function of the qualities,

$$r_\star(x, y) = f_\star[\boldsymbol{q}(x, y)], \qquad f_\star : \mathbb{R}^d \to \mathbb{R}.$$

The induced tilt is

$$p(y \mid x,\, f) \;=\; \frac{1}{Z(x, f)}\, p_{\text{LLM}}(y \mid x)\, e^{f[\boldsymbol{q}(x,y)]}.$$

We place a Gaussian process prior $f_\star \sim \mathcal{GP}(m_0, k_0)$ and maintain a GP posterior; at each step the agent samples $f_t \sim \mathcal{GP}(m_{t-1}, k_{t-1})$ and uses it to define the policy $p_t = p(\cdot \mid x_t, f_t)$. We write $\boldsymbol{q}_t \doteq \boldsymbol{q}(x_t, y_t)$ throughout this section.

[2] $\boldsymbol{\lambda}_\star$ is a modelling artefact rather than an observable quantity, the latent vector implied by the linear assumption.

Scalar feedback (GP regression). With $\rho_t = f_\star(\boldsymbol{q}_t) + \varepsilon_t$, $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, and a zero-mean prior, the posterior after t observations $\mathcal{D}_t = \{(\boldsymbol{q}_i, \rho_i)\}_{i=1}^t$ remains a GP in closed form. With Gram matrix $(\mathbf{K}_{1:t})_{ij} = k(\boldsymbol{q}_i, \boldsymbol{q}_j)$, kernel vector $\boldsymbol{k}_{1:t}(\boldsymbol{q}) = [k(\boldsymbol{q}_1, \boldsymbol{q}), \ldots, k(\boldsymbol{q}_t, \boldsymbol{q})]^\top$, and $\boldsymbol{\rho}_{1:t} = [\rho_1, \ldots, \rho_t]^\top$,

$$m_t(\boldsymbol{q}) = \boldsymbol{k}_{1:t}(\boldsymbol{q})^\top \left(\mathbf{K}_{1:t} + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{\rho}_{1:t}, \tag{2}$$
$$k_t(\boldsymbol{q}, \boldsymbol{q}') = k(\boldsymbol{q}, \boldsymbol{q}') - \boldsymbol{k}_{1:t}(\boldsymbol{q})^\top \left(\mathbf{K}_{1:t} + \sigma^2 \mathbf{I}\right)^{-1} \boldsymbol{k}_{1:t}(\boldsymbol{q}').$$

In practice $(\mathbf{K}_{1:t} + \sigma^2 \mathbf{I})^{-1}$ is computed via an incrementally updated Cholesky decomposition; see Rasmussen & Williams (2006, Chapter 2).

Binary feedback. The logistic likelihood breaks GP conjugacy, so we use a Laplace approximation around the posterior mode (Appendix A.5). We do not report a pairwise GP variant: under Bradley–Terry the object is a posterior over $f_\star(\boldsymbol{q}_1) - f_\star(\boldsymbol{q}_2)$, which changes the function class and the SIR step, and we leave it to future work.

4 Experimental Setup

We evaluate along two axes: synthetic environments where the latent reward is known, and a PRISM-derived persona benchmark with simulated real users.

Environments. A synthetic environment is parametrised by a quality-feature mapping $\boldsymbol{q} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ and a ground-truth reward $r_\star$; varying $r_\star$ across profiles simulates a population of users with different preferences. We use three families: a linear world $r_\star(\boldsymbol{q}) = \boldsymbol{\lambda}_\star^\top \boldsymbol{q}$ over $\boldsymbol{\lambda}_\star \in \mathbb{R}^d$; a non-linear family with quadratic, multiplicative, and gating terms across five canonical profiles (Appendix A.9.2); and a threshold-style family producing piecewise-constant rewards (Appendix A.9.3). Feedback ρ_t is scalar (Gaussian noise σ), binary, or pairwise.

To test on real user preferences, we replace synthetic user profiles with PRISM-derived user feedback (Kirk et al., 2024): each evaluator is impersonated by Claude-Opus 4.5 prompted with that evaluator's PRISM survey responses and fixed past rated interactions as in-context examples, then asked to assess a new generation as that evaluator would; the feedback is either a scalar score or a pairwise preference. Pipeline details and cohort selection are in Appendix A.8.

Quality bases. Throughout the body experiments the parametric agents see a quality vector $\boldsymbol{q}$ drawn from one of four bases: the canonical **HHH** ($D = 3$: helpfulness, harmlessness, humor), **HHH+HP** ($D = 5$: HHH plus honesty and politeness), **13q** ($D = 13$ HHH+HP plus coherence, factuality, relevance, emotional intelligence, readability, subjectivity, diversity, positiveness), and **ArmoRM-19** ($D = 19$: per-attribute reward heads of ArmoRM (Wang et al., 2024a)). Each dimension is a frozen scalar text-quality model; the model identifiers, head names, and what each one measures are listed in Appendix A.9.1. The ICL baseline (§4 below) is basis-invariant by construction: it does not read $\boldsymbol{q}$ at all, so a single ICL value applies across every basis we report.

Agents and protocol. We compare (i) **linear-$\boldsymbol{\lambda}$ TS** (§3.2.1), (ii) **GP TS** (§3.2.2) with an RBF kernel, and (iii) **ICL**, a non-parametric baseline that conditions on past prompt–generation–feedback tuples in-context (prompt construction in Appendix A.7). Prompts are drawn from Anthropic's HH-RLHF dataset (Bai et al., 2022); base completions come from Qwen2.5-7B-Instruct (Yang et al., 2024) via vLLM (Kwon et al., 2023), each candidate generated from the prompt wrapped as a single user turn in Qwen's chat template with an empty system message, with a content-addressed cache so all agents see identical SIR candidate pools per step. We select candidates by argmax: at $\tau = 1$ the SIR tilt is near-uniform on PRISM and flattens further as the agent learns (Appendix A.13); §5.3 compares the two rules on identical pools. We run $T = 100$ rounds and $K = 100$ candidates throughout, and report cumulative reward and soft pseudo-regret with ± 1 standard error.

5 RESULTS

5.1 SYNTHETIC WORLDS: BOTH BELIEFS LEARN THE REWARD THEY ARE GIVEN

Linear rewards. Both beliefs drive pseudo-regret to near zero within tens of interactions (Figure 1, top); GP TS pays a small constant cost for fitting a linear function with an RBF kernel. ICL's per-round reward does not improve, so its cumulative reward grows at a constant, lower rate (Figure 1, bottom). The gap narrows under binary and pairwise feedback (Appendix A.12).

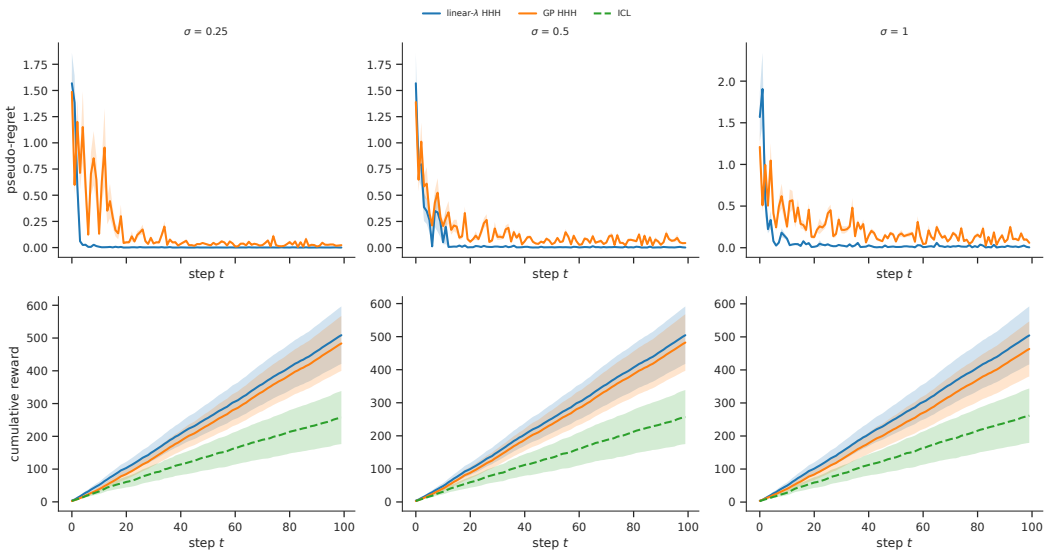


Figure 1: **Linear rewards: parametric Thompson sampling recovers $\boldsymbol{\lambda}_\star$ within tens of interactions.** HHH basis, scalar feedback, seven $\boldsymbol{\lambda}_\star$ profiles, $\sigma \in \{0.25, 0.5, 1.0\}$ (columns). Top: per-step pseudo-regret. Bottom: cumulative reward. Mean ± 1 SE. Per-profile table in Appendix A.10.

Non-linear and threshold rewards. Non-linear rewards are the case for the GP. When $r_\star$ is non-linear in $\boldsymbol{q}$, GP TS keeps pseudo-regret near zero on every profile, while linear-$\boldsymbol{\lambda}$ TS levels off at a profile-dependent residual (Figure 2A1–A2) and accumulates negative reward on Jackpot (Figure 2B). Threshold rewards give the same ordering (Appendix A.11).

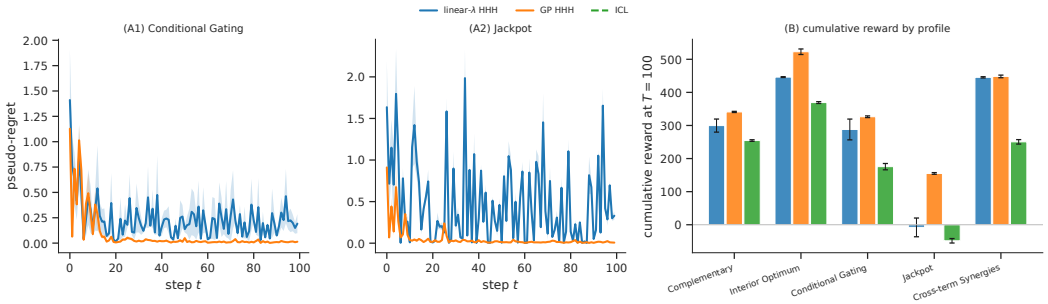


Figure 2: **Non-linear rewards: GP TS remains robust when the linear surrogate fails.** HHH, scalar feedback, five profiles (Appendix A.9.2). (A1, A2) per-step pseudo-regret on **Conditional Gating** and **Jackpot**. (B) cumulative reward at $T = 100$. Mean ± 1 SE. Per-profile table in Appendix A.10.

5.2 PERSONA BENCHMARK: BOTH BELIEFS LEAD ICL ON EVERY BASIS

On PRISM (Figure 3) both beliefs lead ICL on every quality basis under both decoders (Appendix A.14). In panel A both beliefs (ArmoRM-19) start below ICL and cross it around step 35.

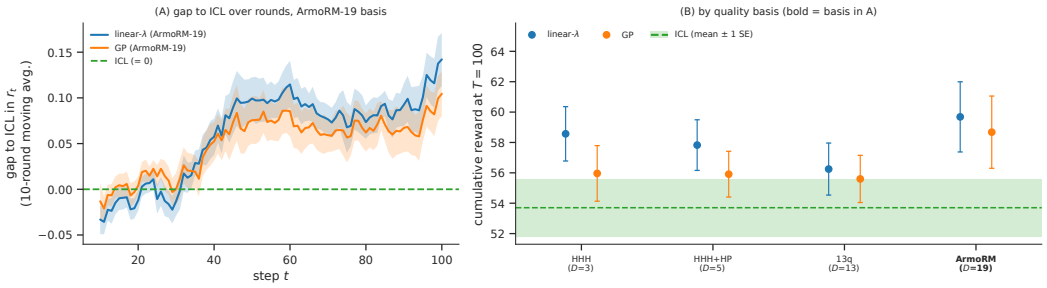


Figure 3: **Persona benchmark: with a sharp decoder, both beliefs lead ICL on every quality basis.** PRISM, 10 users, argmax. (A) per-round reward gap to ICL on ArmoRM-19, 10-round moving average. (B) cumulative reward at $T = 100$ on four bases (bold: basis shown in A), ICL as a band. Mean ± 1 SE. Per-basis gaps in Appendix A.14.

5.3 REWARD DEPENDS JOINTLY ON DIRECTION AND SHARPNESS

We cross the direction with the decode rule on identical pools (Figure 4) and observe that sharpening pays only along a learned direction on both environments. For $\tau = 1$ the tilt also loses sharpness as the agent learns: as the posterior concentrates the Thompson draw shrinks, so a fixed-τ tilt flattens (Appendix A.13). A frozen RM is the linear posterior fit once on n held-out ratings and never updated.

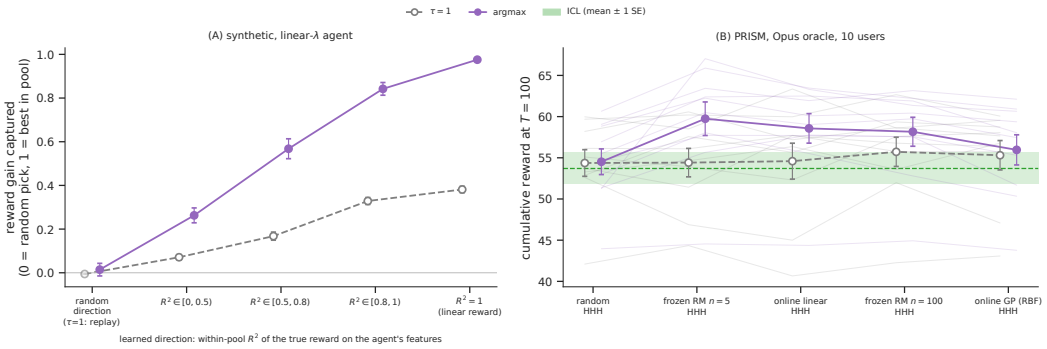


Figure 4: **Sharpening pays only along a learned direction.** SIR ($\tau = 1$) vs. argmax. (A) Synthetic, linear-$\boldsymbol{\lambda}$ agent: share of the achievable reward gain captured (0: uniform pick, 1: best candidate), for a random direction and for learned directions grouped by the within-pool R^2 of the true reward on the agent's features ($R^2 = 1$: linear rewards). (B) PRISM, 10 users: cumulative reward at $T = 100$ by direction; grey lines are users. Mean ± 1 SE.

5.4 THE DECODER, THE BASIS AND PER-USER LEARNING IN THE LEAD OVER ICL

Figure 5 shows two things. (i) The decoder saturates. Along a learned direction the lead over ICL grows with the KL spent, then flattens. The bases coincide until argmax, where ArmoRM-19 pulls ahead. (ii) Five ratings are as good as a hundred: the frozen RM fit on $n = 5$ matches the online posterior at argmax, and the generic head, with no user labels, leads both (Table 6). The reason is the basis: it captures what users agree on, and almost nothing of how they differ (Appendix A.17).

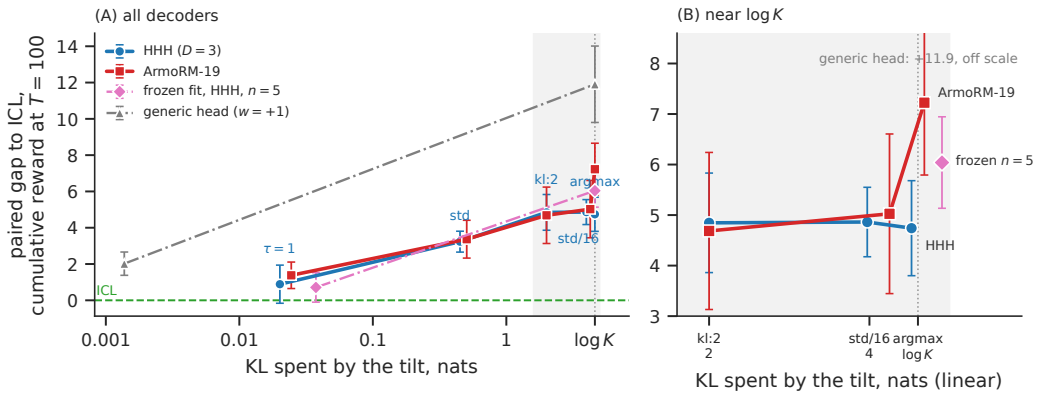


Figure 5: **The lead over ICL against the KL the tilt spends.** PRISM, 10 users, linear-$\boldsymbol{\lambda}$ belief. Paired gap to ICL in cumulative reward at $T = 100$ against the KL the tilt spends, for five decode rules (Appendix A.13) on two bases, plus the frozen $n = 5$ fit and the generic head (ArmoRM's overall score, shared by all users, no learning). (A) log axis; (B) the sharp end, linear axis, argmax points offset sideways. Mean ± 1 SE.

6 Conclusions

We introduce an online alignment framework that casts per-user adaptation of a frozen LLM as a sequential decision problem over exponential tilts of the base distribution. Per user, the KL-regularised optimum (Lemma 1) reduces personalisation to inferring how a fixed quality basis $\boldsymbol{q}$ maps to reward. We learn this mapping online by Thompson sampling (TS) under a linear-$\boldsymbol{\lambda}$ Gaussian belief (linear-$\boldsymbol{\lambda}$ TS) or a Gaussian process (GP TS), from scalar, binary or pairwise feedback.

On synthetic landscapes the framework converges as predicted: linear-$\boldsymbol{\lambda}$ TS is sample-efficient when its linearity assumption holds, and GP TS remains robust when it does not. On the PRISM persona benchmark, both linear-$\boldsymbol{\lambda}$ TS and GP TS obtain higher cumulative reward at $T = 100$ than the in-context learning (ICL) baseline on all four quality bases. Reward depends on both the learned direction and the decoding rule: argmax beats SIR along a learned direction, and does nothing along a random one. Under argmax, the choice of basis matters as much as the choice of the decoding rule (§5.4).

The framework provides an explicit and inexpensive per-user posterior over a quality basis. The posterior learns the user's reward, and the sharpness of the selection, measured in KL, decides how much of it reaches the user. The K candidates and the quality basis bound what can be achieved. Next steps should target these rather than the posterior: learning the quality basis from user data, generating candidates guided by the learned reward instead of selecting among independent samples, and evaluating with more users and with human raters.

AI use statement

In this work, we used generative AI tools (Claude Code by Anthropic and ChatGPT by OpenAI) as authoring aids. More exactly, we used these tool to assiss with the implementation of plotting functions for formatting and presentation (mostly Claude). We also used AI to aid in writing, specifically identifying alternative wording and synonyms, and fixing grammar (both Claude Code and ChatGPT).

Generating synthetic data is part of the evaluation methodology. We use Claude Opus 4.5 to impersonate PRISM users in order to produce user-specific feedback in our persona benchmark (Section 4, Appendix A.8). This is not authoring aid. The rest of the required disclosure tasks are not applicable to this work.

We have reviewed all AI-assisted work. The authors reviewed and ran all AI-assited code, and verified and edited AI-assisted text (proposed wording and synonyms) where necessary. We take

responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952. URL https://api.semanticscholar.org/CorpusID:125209808.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. Pad: Personalized alignment of llms at decoding-time, 2024. URL https://arxiv.org/abs/2410.04070. ICLR 2025.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023. URL https://arxiv.org/abs/1706.03741.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 208–214, 2011.

Xiangxiang Dai, Yuejin Xie, Maoli Liu, Xuchuang Wang, Zhuohua Li, Huanyu Wang, and John C.S. Lui. A multi-agent conversational bandit approach to online evaluation and selection of user-aligned llm responses, 2025. URL https://arxiv.org/abs/2501.01849.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024. URL https://arxiv.org/abs/2402.01306.

Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, Qi Zhang, and Dahua Lin. Linear alignment: A closed-form solution for aligning human preferences without tuning and feedback, 2024. URL https://arxiv.org/abs/2401.11458.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Advances in Neural Information Processing Systems*, 2018.

Roman Garnett. Lecture 13 notes. https://www.cse.wustl.edu/~garnett/cse515t/spring_2015/files/lecture_notes/13.pdf, 2015.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. CRC Press, third edition, 2013.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search, 2024. URL https://arxiv.org/abs/2402.01694.

Minbeom Kim, Kang il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. Drift: Decoding-time personalized alignments with implicit user preferences, 2025. URL https://arxiv.org/abs/2502.14289.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models, 2024. URL https://arxiv.org/abs/2404.16019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention, 2023. URL https://arxiv.org/abs/2309.06180.

Zichen Liu, Changyu Chen, Chao Du, Wee Sun Lee, and Min Lin. Sample-efficient alignment for llms, 2024. URL https://arxiv.org/abs/2411.01493.

Thomas Minka. A family of algorithms for approximate bayesian inference. *PhD Thesis*, 05 2014.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models, 2024. URL https://arxiv.org/abs/2310.17022.

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

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning, 2024. URL https://arxiv.org/abs/2408.10075.

Zikun Qu, Min Zhang, Mingze Kong, Xiang Li, Zhiwei Shang, Zhiyong Wang, Yikun Ban, Shuang Qiu, Yao Shu, and Zhongxiang Dai. T-pop: Test-time personalization with online preference feedback, 2025. URL https://arxiv.org/abs/2509.24696.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards, 2023. URL https://arxiv.org/abs/2306.04488.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. Language model personalization via reward factorization, 2025. URL https://arxiv.org/abs/2503.06358.

Suho Shin, Chenghao Yang, Haifeng Xu, and Mohammad T. Hajiaghayi. Tokenized bandit for llm decoding and alignment, 2025. URL https://arxiv.org/abs/2506.07276.

Adrian F.M. Smith and Alan E. Gelfand. Bayesian statistics without tears: A sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022, 2010.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio P. Calmon. Soft best-of-n sampling for model alignment, 2025. URL https://arxiv.org/abs/2505.03156.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts, 2024a. URL https://arxiv.org/abs/2406.12845.

Xinran Wang, Qi Le, Ammar Ahmed, Enmao Diao, Yi Zhou, Nathalie Baracaldo, Jie Ding, and Ali Anwar. Map: Multi-human-value alignment palette, 2024b. URL https://arxiv.org/abs/2410.19198.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training, 2023. URL https://arxiv.org/abs/2306.01693.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Zhaowei Zhang, Fengshuo Bai, Qizhi Chen, Chengdong Ma, Mingzhi Wang, Haoran Sun, Zilong Zheng, and Yaodong Yang. Amulet: Realignment during test time for personalized preference adaptation of llms, 2025. URL https://arxiv.org/abs/2502.19148.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization, 2024. URL https://arxiv.org/abs/2310.03708.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

A Appendix

A.1 Proof of Lemma 1

Proof 1 *We have that*

$$\begin{aligned}
&\mathbb{E}_{y\sim p(y|x,r)}\big[r_\star(x,y)\big] - \mathrm{KL}\big[p(y \mid x,r) \,\|\, p_{LLM}(y \mid x)\big] \\
&\quad = \mathbb{E}_{y\sim p(y|x,r)}\big[r_\star(x,y) + \log p_{LLM}(y \mid x)\big] + H[p(y \mid x,r)] \\
&\quad = \mathbb{E}_{y\sim p(y|x,r)}\big[r_\star(x,y) + \log p_{LLM}(y \mid x) - \log p(y \mid x,r)\big] \\
&\quad = -\mathrm{KL}\big[p(y \mid x,r) \,\|\, \tfrac{1}{Z(x,r_\star)}\, p_{LLM}(y \mid x)\, e^{r_\star(x,y)}\big] + \log Z(x,r_\star) \\
&\quad = -\mathrm{KL}\big[p(y \mid x,r) \,\|\, p(y \mid x,r_\star)\big] + \log Z(x,r_\star),
\end{aligned}$$

where

$$Z(x, r_\star) \doteq \mathbb{E}_{y\sim p_{LLM}(\cdot|x)}\big[e^{r_\star(x,y)}\big].$$

Since $\log Z(x, r_\star)$ *is constant with respect to the policy, maximising the objective is equivalent to minimising the KL divergence above, which vanishes when both arguments are identical, i.e.,* $p(y \mid x, r) = p(y \mid x, r_\star) \doteq p_\star(y \mid x)$.

A.2 Sampling-importance resampling details

For sample selection from the importance weights $w_i \propto e^{r(x_t, y_t^i)}$, Wang et al. (2024b, Algorithm 1, Step 3) propose normalising the weights via softmax followed by multinomial sampling, while their reference code computes the greedy argmax over the candidates; see implementation at MAP decoding (line 141); PReF (Shenfeld et al., 2025) also serves the argmax over the candidate pool, see PReF inference (lines 96–107). More background on SIR can be found in Gelman et al. (2013, Section 10.4) and Smith & Gelfand (1992, Section 3.2).

A.3 Woodbury form of the Gaussian scalar update

The closed-form Gaussian posterior update of §3.2.1 can equivalently be written via the Woodbury identity. Let

$$s_t = \boldsymbol{q}(x_t, y_t)^\top \boldsymbol{\Sigma}_{t-1} \boldsymbol{q}(x_t, y_t) + \sigma^2, \qquad \boldsymbol{k}_t = \boldsymbol{\Sigma}_{t-1}\boldsymbol{q}(x_t, y_t)/s_t.$$

Then

$$\begin{aligned}\boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_{t-1} - \boldsymbol{k}_t\, \boldsymbol{q}(x_t, y_t)^\top \boldsymbol{\Sigma}_{t-1},\\ \boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t-1} + \boldsymbol{k}_t\big(\rho_t - \boldsymbol{q}(x_t, y_t)^\top \boldsymbol{\mu}_{t-1}\big).\end{aligned}$$

A.4 ADF update for binary and pairwise feedback

For binary feedback $\rho_t \in \{0, 1\}$ with logistic likelihood

$$p(\rho_t = 1 \mid \boldsymbol{q}(x_t, y_t), \boldsymbol{\lambda}_\star) = \sigma\big(\boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x_t, y_t)\big), \qquad \sigma(z) \doteq \frac{1}{1 + e^{-z}},$$

the posterior is no longer Gaussian. Assumed-density filtering (ADF) approximates it by $\tilde{p}_t(\boldsymbol{\lambda}_\star) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ minimising $\mathrm{KL}[p_t(\boldsymbol{\lambda}_\star) \,\|\, \tilde{p}_t(\boldsymbol{\lambda}_\star)]$ via moment matching. The resulting one-step update is

$$\begin{aligned}\boldsymbol{\Sigma}_t^{-1} &= \boldsymbol{\Sigma}_{t-1}^{-1} + \tau\, \boldsymbol{q}(x_t, y_t)\boldsymbol{q}(x_t, y_t)^\top,\\ \boldsymbol{\mu}_t &= \boldsymbol{\Sigma}_t\big(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\mu}_{t-1} + \nu\, \boldsymbol{q}(x_t, y_t)\big),\end{aligned}$$

where

$$\tau = \frac{-\beta}{1 + \beta\, \boldsymbol{q}(x_t, y_t)^\top \boldsymbol{\Sigma}_{t-1} \boldsymbol{q}(x_t, y_t)}, \qquad \nu = \frac{\alpha - \beta\, \boldsymbol{\mu}_{t-1}^\top \boldsymbol{q}(x_t, y_t)}{1 + \beta\, \boldsymbol{q}(x_t, y_t)^\top \boldsymbol{\Sigma}_{t-1} \boldsymbol{q}(x_t, y_t)},$$

and α, β are the first and second derivatives of the log-likelihood

$$\log p(\rho_t \mid z) = \rho_t \log \sigma(z) + (1 - \rho_t) \log[1 - \sigma(z)],$$

evaluated at the predictive mean $z = \boldsymbol{\mu}_{t-1}^\top \boldsymbol{q}(x_t, y_t)$:

$$\alpha = \frac{\partial}{\partial z} \log p(\rho_t \mid z)\bigg|_{z=\boldsymbol{\mu}_{t-1}^\top \boldsymbol{q}(x_t, y_t)}, \qquad \beta = \frac{\partial^2}{\partial z^2} \log p(\rho_t \mid z)\bigg|_{z=\boldsymbol{\mu}_{t-1}^\top \boldsymbol{q}(x_t, y_t)}.$$

More on ADF and the derivation of these updates can be found in Garnett (2015) and Minka (2014, Section 3.1).

Pairwise reduction. In a pairwise reward environment, where two completions $y_t^{(1)}, y_t^{(2)}$ are generated for the same prompt x_t and $\rho_t \in \{0, 1\}$ indicates which is preferred, the Bradley–Terry model (Bradley & Terry, 1952) gives

$$p\big(\rho_t = 1 \mid \boldsymbol{q}(x_t, y_t^{(1)}), \boldsymbol{q}(x_t, y_t^{(2)}), \boldsymbol{\lambda}_\star\big) = \frac{e^{\boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x_t, y_t^{(1)})}}{e^{\boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x_t, y_t^{(1)})} + e^{\boldsymbol{\lambda}_\star^\top \boldsymbol{q}(x_t, y_t^{(2)})}} = \sigma\big(\boldsymbol{\lambda}_\star^\top \hat{\boldsymbol{q}}_t\big),$$

where $\hat{\boldsymbol{q}}_t \doteq \boldsymbol{q}(x_t, y_t^{(1)}) - \boldsymbol{q}(x_t, y_t^{(2)})$ is the comparison feature vector. The pairwise signal therefore reduces to a Bernoulli observation with logit $z = \boldsymbol{\lambda}_\star^\top \hat{\boldsymbol{q}}_t$, and the ADF update above applies verbatim with $\boldsymbol{q}(x_t, y_t)$ replaced by $\hat{\boldsymbol{q}}_t$.

A.5 Laplace approximation for GP binary feedback

For the GP variant under logistic-link binary feedback (§3.2.2), the posterior over $\boldsymbol{f}_{1:t} = [f(\boldsymbol{q}_1), \dots, f(\boldsymbol{q}_t)]^\top$ is non-Gaussian:

$$p(\boldsymbol{f}_{1:t} \mid \mathcal{D}_t) \;\propto\; \mathcal{N}(\boldsymbol{f}_{1:t} \mid \mathbf{0}, \mathbf{K}_{1:t}) \prod_{i=1}^{t} \sigma(f_i)^{\rho_i} [1 - \sigma(f_i)]^{1-\rho_i}.$$

We approximate it via a Laplace expansion around the MAP estimate $\hat{\boldsymbol{f}}_{1:t} = \arg\max_{\boldsymbol{f}_{1:t}} \log p(\boldsymbol{f}_{1:t} \mid \mathcal{D}_t)$:

$$p(\boldsymbol{f}_{1:t} \mid \mathcal{D}_t) \;\approx\; \mathcal{N}\big(\hat{\boldsymbol{f}}_{1:t},\; \boldsymbol{\Sigma}_t\big), \qquad \boldsymbol{\Sigma}_t = \big(\mathbf{K}_{1:t}^{-1} + \mathbf{W}\big)^{-1},$$

where

$$\mathbf{W} = -\nabla^2_{\boldsymbol{f}_{1:t}} \sum_{i=1}^{t} \log p(\rho_i \mid f_i)\Big|_{\boldsymbol{f}=\hat{\boldsymbol{f}}} = \operatorname{diag}\big(\hat{\pi}_i(1-\hat{\pi}_i)\big), \qquad \hat{\pi}_i = \sigma(\hat{f}_i).$$

A.6 Pseudo-regret estimation

The pseudo-regret curves in this paper are estimated from the agent's posterior samples and a per-prompt candidate pools drawn from $p_{\text{LLM}}(\cdot \mid x)$, both gathered throughout experiments. We give the full derivation here so the estimator is reproducible from the artefacts.

Given the tilted policy

$$p_t(y) \;=\; Z(f_t)^{-1}\, p_{\text{LLM}}(y \mid x_t)\, e^{f_t[\boldsymbol{q}(x_t,y)]} \;=\; p_{\text{LLM}}(y \mid x_t)\, e^{f_t[\boldsymbol{q}(x_t,y)] - \log Z(f_t)},$$

we write KL between p_t and the base LLM as:

$$\begin{aligned}\mathrm{KL}[p_t \,\|\, p_{\text{LLM}}(\cdot \mid x_t)] &= \mathbb{E}_{y\sim p_t}\Big[\log p_{\text{LLM}}(y \mid x_t) + f_t[\boldsymbol{q}(y)] - \log Z(f_t) - \log p_{\text{LLM}}(y \mid x_t)\Big] \\ &= \mathbb{E}_{y\sim p_t}\big[f_t[\boldsymbol{q}(y)]\big] \;-\; \log Z(f_t). \end{aligned} \tag{3}$$

The same identity applies to $p_\star(y) \propto p_{\text{LLM}}(y \mid x_t) e^{r_\star(x_t,y)}$ with $f_t \leftarrow r_\star$, simplifing the optimal cumulative reward:

$$\begin{aligned} G_T^\star &= \sum_{t=1}^{T} \Big(\mathbb{E}_{y\sim p_\star}[r_\star(y)] - \mathrm{KL}[p_\star \,\|\, p_{\text{LLM}}(\cdot \mid x_t)]\Big) \\ &= \sum_{t=1}^{T} \Big(\mathbb{E}_{y\sim p_\star}[r_\star(y)] - \mathbb{E}_{y\sim p_\star}[r_\star(y)] + \log Z_t(r_\star)\Big) \;=\; \sum_{t=1}^{T} \log Z_t(r_\star). \end{aligned}$$

Monte-Carlo / SNIS estimator. At initialisation we draw M samples per prompt x_t, $y_t^{(m)} \sim p_{\text{LLM}}(\cdot \mid x_t)$, store their feature vectors $\boldsymbol{q}_t^{(m)} = \boldsymbol{q}(x_t, y_t^{(m)})$, and reuse them throughout the run (so the same baseline batch is used across agents and seeds; this is the SIR pool of §3.2.1). At step t, with shorthand $s_t^{(m)} = f_t[\boldsymbol{q}_t^{(m)}]$ for the agent's posterior-sample score, self-normalised importance sampling (SNIS) gives

$$w_t^{(m)} \;=\; \frac{p_t(y_t^{(m)})}{p_{\text{LLM}}(y_t^{(m)} \mid x_t)} \;\propto\; e^{s_t^{(m)}}, \qquad w_t^{(m)} \;=\; \frac{e^{s_t^{(m)}}}{\sum_{j=1}^{M} e^{s_t^{(j)}}}.$$

We approximate the expectation in equation 3 as

$$\mathbb{E}_{y\sim p_t}\big[f_t[\boldsymbol{q}(y)]\big] \;\approx\; \sum_{m=1}^{M} w_t^{(m)}\, s_t^{(m)}, \tag{4}$$

and the log-partition function via the same baseline batch:

$$\log Z(f_t) \;=\; \log \mathbb{E}_{y\sim p_{\text{LLM}}(\cdot \mid x_t)}\big[e^{f_t[\boldsymbol{q}(y)]}\big] \;\approx\; \log \tfrac{1}{M} \sum_{m=1}^{M} e^{s_t^{(m)}}. \tag{5}$$

Substituting equation 4 and equation 5 into equation 3 yields

$$\mathrm{KL}[p_t \,\|\, p_{\mathrm{LLM}}(\cdot \mid x_t)] \;\approx\; \sum_{m=1}^{M} w_t^{(m)}\, s_t^{(m)} \;-\; \log \tfrac{1}{M} \sum_{m=1}^{M} e^{s_t^{(m)}}. \tag{6}$$

Applying SNIS once more for $\mathbb{E}_{y\sim p_t}[r_\star(y)]$ on the same baseline batch gives the cumulative-reward approximation

$$G_T \;\approx\; \sum_{t=1}^{T}\Bigg(\sum_{m=1}^{M} w_t^{(m)}\, r_\star\big(y_t^{(m)}\big) \;-\; \sum_{m=1}^{M} w_t^{(m)}\, s_t^{(m)} \;+\; \log \tfrac{1}{M} \sum_{m=1}^{M} e^{s_t^{(m)}} \Bigg). \tag{7}$$

The third term carries a $+$ sign because $G_t = \mathbb{E}_{p_t}[r_\star] - \mathrm{KL}$ and the KL estimator equation 6 contributes $-\log \frac{1}{M}\sum e^{s_t^{(m)}}$ which is then subtracted.

Soft pseudo-regret. Combining $G_T^\star = \sum_t \log Z_t(r_\star)$ with equation 7 and approximating $\log Z_t(r_\star) \approx \log \frac{1}{M}\sum_m e^{r_\star(y_t^{(m)})}$ on the same batch gives

$$\begin{aligned} R_T \;=\; G_T^\star - G_T \;&\approx\; \sum_{t=1}^{T}\Bigg[\log \tfrac{1}{M}\sum_m e^{r_\star(y_t^{(m)})} \;-\; \sum_m w_t^{(m)} r_\star\big(y_t^{(m)}\big) \;+\; \sum_m w_t^{(m)} s_t^{(m)} \;-\; \log \tfrac{1}{M}\sum_m e^{s_t^{(m)}} \Bigg] \\ &=\; \sum_{t=1}^{T}\Bigg[\log \frac{\sum_m e^{r_\star(y_t^{(m)})}}{\sum_m e^{s_t^{(m)}}} \;+\; \sum_m w_t^{(m)} \Big(s_t^{(m)} - r_\star\big(y_t^{(m)}\big) \Big) \Bigg]. \end{aligned}$$

Cumulative reward. The cumulative-reward curves use the feedback the agent actually received, $\rho_t \doteq r_\star(\boldsymbol{q}_t)$ for scalar / threshold scorers or $\rho_t \in \{0, 1\}$ in binary / pairwise channels, summed up to t. This is the unified reward axis across linear-$\boldsymbol{\lambda}$, GP, linear-$\boldsymbol{\lambda}\sigma$, and ICL because it does not need the agent's posterior.

A.7 ICL PROMPT CONSTRUCTION

The ICL baseline conditions a frozen LLM on past prompt–generation–feedback tuples through the system message, with no parameter updates. We use one fixed system-prompt template per feedback channel; user requests are passed through unchanged so the LLM sees the same prompt the parametric agents condition on. The placeholders `{icl_examples}`, `{prompt}`, `{completion}`, `{score}`, `{preferred_completion}`, `{rejected_completion}` are filled by the runtime; example formatters concatenate per-example renderings with blank-line separators before substituting them into the system prompt. The verbatim blocks below reproduce the prompts as they are sent to the model.

Initial template (no examples observed yet). All three channels fall back to this template at $t = 0$:

```
You are an assistant that should answer the user's question in the way
↪  the user is most likely to prefer.

No past interaction examples are available yet.

Do not mention preferences, examples, or scores.
Output only the answer to the new question.
```

Scored feedback. System prompt:

```
You are an assistant that has to respond to prompts in the preferred way
↪  by the user. In the following you will be given a list of example
↪  past interactions, including the user question,
the assistant answer, and the score given by the user. Higher scores
↪  mean higher preference for the answer.
```

```
You will then be given a new question and you'll have to respond in the
↪  preferred way by the user, as implied by the scores in the previous
↪  examples.

Example past interactions:
{icl_examples}
```

Per-example rendering:

```
User request:
{prompt}

Assistant answer:
{completion}

Score:
{score}
```

Binary feedback. System prompt:

```
You are an assistant that has to respond to prompts in the preferred way
↪  by the user. In the following you will be given a list of example
↪  past interactions, including the user question,
the assistant answer, and the binary score given by the user. A score of
↪  1 means the user preferred the answer and a score of 0 means the
↪  user did not prefer the answer.

You will then be given a new question and you'll have to respond in the
↪  preferred way by the user, as implied by the binary scores in the
↪  previous examples.

Example past interactions:
{icl_examples}
```

Per-example rendering:

```
User request:
{prompt}

Assistant answer:
{completion}

Score:
{score}/1
```

Pairwise feedback. System prompt:

```
You are an assistant that has to respond to prompts in the preferred way
↪  by the user. In the following you will be given a list of example
↪  past interactions, including the user question, the preferred
↪  answer, and the rejected answer.
You will then be given a new prompt and you have to respond to it in the
↪  preferred way.

Example past interactions:
{icl_examples}
```

Per-example rendering:

```
User request:
{prompt}
```

```
Preferred assistant answer:
{preferred_completion}

Rejected assistant answer:
{rejected_completion}
```

Conditioning window. Only the scored variant uses a bounded conditioning window. Past interactions are stored in a max-heap keyed by score, and at each step the top k scored examples (highest scores first) are inserted into `{icl_examples}`. The binary and pairwise variants use the full history at every step (no truncation).

A.8 PRISM IMPERSONATION PIPELINE

Dataset and tasks. We use the PRISM dataset (Kirk et al., 2024), which contains ∼8000 AI conversations annotated by ∼1500 different evaluators. We establish two prediction tasks to measure how well a LLM can impersonate an evaluator (user):absolute score, which predicts a 1–100 rating the user would assign a generation given a prompt, and preference, that predicts which of two candidates responses the user would prefer for a given prompt.

Strategies and metrics. We compare two impersonation strategies under leave-one-conversation-out cross-validation (LOOCV): an ICL strategy that embeds the evaluator's previous rating records into the prompt as few-shot examples, and a distillation method that summarises the history into a latent persona document and uses that as context. We use the Prediction Skill Score $\mathrm{PSS} = 1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$ to obtain absolute-score performance; a positive score signifies predictive skill above the baseline. Preference performance is assessed via chance-adjusted accuracy $\mathrm{Acc}_{\mathrm{adj}} = 2\,\mathrm{Acc} - 1$ ($0 =$ chance, $1 =$ perfect).

Model and prompt selection. Sweeping prompt designs across Claude Haiku and Claude Opus models on a 10-user set (Table 1) showed that Claude Opus outperforms in absolute scoring (PSS 0.454 compared to 0.289 for Haiku). A multi-step reasoning prompt makes the two comparable w.r.t. preference prediction. Consequently, we employ Claude Opus with an ICL strategy for user impersonation.

Table 1: LOOCV performance of the impersonation strategies on a 10-user reference set.

Model	Method & Task	PSS	Chance-Adj. Acc.
Claude Haiku	Best ICL (Score)	0.289	–
Claude Haiku	Best ICL (Preference)	–	0.463
Claude Haiku	Best Distillation (Score)	0.216	–
Claude Haiku	Best Distillation (Preference)	–	0.365
Claude Opus	Best ICL (Score)	**0.454**	–
Claude Opus	Best Distillation (Preference)	–	0.436

User cohort selection. An evaluator's internal consistency deteremines impersonation efficacy. Thus, we screen the evaluator pool down to a 10-user set via the following steps: (i) filter profiles with ≥ 5 conversations and rating-scale SD ≥ 15; (step outputs 250 candidates); (ii) score each evaluator via LOOCV with the best Haiku ICL prompt; (iii) filter again with $\mathrm{PSS} \geq 0.10$ or $\mathrm{Acc}_{\mathrm{adj}} \geq 0.10$, rank by $0.5\,\mathrm{Norm}(\mathrm{PSS}) + 0.5\,\mathrm{Norm}(\mathrm{Acc}_{\mathrm{adj}})$ and keep the top 10. The final cohort has median PSS ≈ 0.53 and median chance-adjusted accuracy ≈ 0.64.

A.9 IMPLEMENTATION

Base completions are generated by Qwen2.5-7B-Instruct (Yang et al., 2024) served via vLLM (Kwon et al., 2023). The Gaussian process belief is implemented in GPyTorch (Gardner et al., 2018) (RBF and Matérn kernels, exact inference for scalar feedback, Laplace approximation for binary feedback).

Sampling from the base LLM is mediated by a content-addressed cache keyed by SHA256 of the prompt and a backend / config / seed tuple, so that any two runs sharing prompt, seed and sampler configuration receive identical SIR candidate pools at each step. As such agent comparisons are over the same candidate sets.

A.9.1 Quality dimensions and scoring models

Each component of the quality vector $\boldsymbol{q}$ is produced by a frozen text-quality model. Helpfulness and harmlessness are assessed using GPT-2 reward models fine-tuned for these attributes (`Ray2333/gpt2-large-helpful-reward_model`, `Ray2333/gpt2-large-harmless-reward_model`). Humor is assessed with a humor classifier (`mohameddhiab/humor-no-humor`), while positiveness is scored using a sentiment classifier (`siebert/sentiment-roberta-large-english`). Honesty is derived from the contradiction signal of an NLI model (`typeform/distilbert-base-uncased-mnli`), and politeness is approximated using a formality ranker (`s-nlp/roberta-base-formality-ranker`). Coherence, factuality, and relevance are evaluated using a DeBERTa-v3 NLI model with dimension-specific templates (`MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli`). Emotional intelligence, readability, and subjectivity are scored using pretrained classifiers for emotion recognition (`Sathwik3/distilbert-emotion-classifier`), readability level (`kiddom/modernbert-readability-grade-predictor`), and subjective style(`cffl/bert-base-styleclassification-subjective-neutral`), respectively. Diversity is computed as in MAP (Wang et al., 2024b).

ArmoRM-19 We expose the available per-attribute rewards of `RLHFlow/ArmoRM-Llama3-8B-v0.1` (Wang et al., 2024a) as quality dimensions: 5 HelpSteer heads (helpfulness, correctness, coherence, complexity, verbosity), 5 UltraFeedback heads (overall score, instruction following, truthfulness, honesty, helpfulness), BeaverTails is-safe, Prometheus score, 2 Argilla heads (overall quality, judge-LM), and five code heads (complexity, style, explanation, instruction following, readability).

A.9.2 Non-linear reward profiles

The five non-linear profiles use the HHH basis (q_0=helpfulness, q_1=harmlessness, q_2=humour)

Profile 1 — Complementary. $r_\star = 4q_0 + 4q_1 - 8\,q_0q_1 + q_2$.

Profile 2 — Interior Optimum. $r_\star = 10\,q_0 - 10\,q_0^2 + 8\,q_1 - 8\,q_1^2 + 2\,q_2$.

Profile 3 — Conditional Gating. $r_\star = 2q_0 - 2q_2 + 6\,q_1q_2$.

Profile 4 — Jackpot. $r_\star = -q_0 - q_1 - q_2 + 10\,q_0q_1q_2$.

Profile 5 — Cross-term Synergies. $r_\star = q_0 + 0.5\,q_1 + 0.5\,q_2 + 4\,q_1q_2 + 1.5\,q_0q_1$.

A.9.3 Threshold-style reward profiles

The threshold family use a linear base in $\boldsymbol{q}$ plus a bonus that fires only when a subset of dimensions clear a specified cutoffs,

$$r_\star(\boldsymbol{q}) \;=\; \boldsymbol{w}^\top \boldsymbol{q} \;+\; b \;+\; \text{bonus} \cdot \mathbb{1}[\, q_{i_k} > \tau_k \;\; \forall k \,] \,.$$

Profile T1 (HHH). $\boldsymbol{q} = (q_0, q_1, q_2)$ for (helpfulness, harmlessness, humour).

$$r_\star \;=\; q_0 + 0.5\,q_1 + 0.5\,q_2 \;+\; 6 \cdot \mathbb{1}[\, q_0 > 0.5 \;\wedge\; q_1 > 0.85 \,] \,.$$

Profile T2 (HHH). $\boldsymbol{q} = (q_0, q_1, q_2)$ for (helpfulness, harmlessness, humour).

$$r_\star \;=\; 2\,q_0 + q_1 \;+\; 5 \cdot \mathbb{1}[\, q_1 > 0.75 \;\wedge\; q_2 > 0.3 \,] \,.$$

Profile T3 (HHH+HP). $\boldsymbol{q} = (q_0, q_1, q_2, q_3, q_4)$ for (helpfulness, harmlessness, humour, honesty, politeness).

$$r_\star \;=\; 0.5 \sum_{i=0}^{4} q_i \;+\; 8 \cdot \mathbb{1}[\, q_0 > 0.5 \;\wedge\; q_1 > 0.85 \;\wedge\; q_3 > 0.7 \,].$$

A.10 PER-PROFILE SYNTHETIC RESULTS

Table 2: End-of-horizon results per profile, HHH basis, scalar feedback, mean ± 1 SE across runs. Each panel has one block per decode rule: multinomial SIR, and argmax (the decode rule of Figures 1 and 2). ICL does not use the decoder, so its column is the same in both sections. (A) Linear world, Figure 1: last-10 pseudo-regret. (B) Non-linear world, Figure 2: last-10 pseudo-regret and cumulative reward at $T = 100$.

(A) Linear world: last-10 pseudo-regret						
	$\sigma = 0.25$		$\sigma = 0.5$		$\sigma = 1.0$	
$\boldsymbol{\lambda}_\star$	linear-$\boldsymbol{\lambda}$	GP	linear-$\boldsymbol{\lambda}$	GP	linear-$\boldsymbol{\lambda}$	GP
SIR, $\tau = 1$						
$(5, 0.5, 0.5)$	0.001 ± 0.000	0.011 ± 0.003	0.003 ± 0.001	0.019 ± 0.005	0.011 ± 0.003	0.075 ± 0.017
$(0.5, -5, 0.5)$	0.001 ± 0.000	0.009 ± 0.002	0.003 ± 0.002	0.026 ± 0.002	0.021 ± 0.012	0.045 ± 0.009
$(0.5, 5, 5)$	0.001 ± 0.000	0.013 ± 0.000	0.003 ± 0.001	0.042 ± 0.005	0.016 ± 0.007	0.133 ± 0.034
$(5, -5, 0.5)$	0.001 ± 0.000	0.011 ± 0.003	0.004 ± 0.002	0.021 ± 0.002	0.012 ± 0.004	0.062 ± 0.009
$(5, 5, 5)$	0.001 ± 0.000	0.023 ± 0.003	0.002 ± 0.000	0.079 ± 0.017	0.013 ± 0.006	0.175 ± 0.015
$(5, 0, 5)$	0.001 ± 0.000	0.010 ± 0.001	0.003 ± 0.001	0.035 ± 0.002	0.009 ± 0.004	0.097 ± 0.016
$(0, 5, -5)$	0.001 ± 0.000	0.013 ± 0.004	0.004 ± 0.001	0.020 ± 0.001	0.015 ± 0.003	0.046 ± 0.009
argmax						
$(5, 0.5, 0.5)$	0.001 ± 0.000	0.030 ± 0.015	0.003 ± 0.001	0.103 ± 0.046	0.010 ± 0.003	0.155 ± 0.056
$(0.5, -5, 0.5)$	0.001 ± 0.000	0.020 ± 0.003	0.002 ± 0.001	0.034 ± 0.005	0.011 ± 0.005	0.046 ± 0.005
$(0.5, 5, 5)$	0.001 ± 0.000	0.022 ± 0.001	0.003 ± 0.001	0.088 ± 0.014	0.013 ± 0.003	0.117 ± 0.008
$(5, -5, 0.5)$	0.001 ± 0.000	0.030 ± 0.014	0.003 ± 0.001	0.044 ± 0.011	0.015 ± 0.009	0.124 ± 0.006
$(5, 5, 5)$	0.002 ± 0.001	0.034 ± 0.003	0.006 ± 0.002	0.084 ± 0.002	0.017 ± 0.004	0.179 ± 0.016
$(5, 0, 5)$	0.001 ± 0.000	0.020 ± 0.003	0.003 ± 0.001	0.047 ± 0.007	0.012 ± 0.004	0.087 ± 0.007
$(0, 5, -5)$	0.002 ± 0.000	0.019 ± 0.005	0.005 ± 0.001	0.061 ± 0.015	0.018 ± 0.004	0.090 ± 0.012

(B) Non-linear world						
	last-10 pseudo-regret			cumulative reward at $T = 100$		
profile	linear-$\boldsymbol{\lambda}$	GP	ICL	linear-$\boldsymbol{\lambda}$	GP	ICL
SIR, $\tau = 1$						
Complementary	0.106 ± 0.012	0.011 ± 0.001	–	248.8 ± 3.9	256.7 ± 2.2	254.5 ± 2.4
Interior Optimum	0.248 ± 0.002	0.009 ± 0.001	–	394.2 ± 3.4	418.5 ± 3.6	369.2 ± 2.8
Conditional Gating	0.100 ± 0.003	0.009 ± 0.001	–	197.8 ± 4.1	213.3 ± 3.8	175.5 ± 9.5
Jackpot	0.336 ± 0.008	0.009 ± 0.001	–	-41.8 ± 5.8	15.8 ± 6.9	-48.5 ± 6.8
Cross-term Synergies	0.064 ± 0.004	0.012 ± 0.001	–	311.9 ± 6.6	326.4 ± 2.7	250.5 ± 6.8
argmax						
Complementary	0.405 ± 0.136	0.021 ± 0.006	–	299.6 ± 19.7	340.9 ± 1.8	254.5 ± 2.4
Interior Optimum	0.244 ± 0.004	0.028 ± 0.003	–	446.2 ± 1.3	522.8 ± 8.4	369.2 ± 2.8
Conditional Gating	0.222 ± 0.102	0.013 ± 0.002	–	287.9 ± 31.4	326.4 ± 2.6	175.5 ± 9.5
Jackpot	0.507 ± 0.061	0.012 ± 0.001	–	-8.0 ± 28.3	155.0 ± 2.5	-48.5 ± 6.8
Cross-term Synergies	0.081 ± 0.002	0.015 ± 0.002	–	445.5 ± 1.9	448.2 ± 4.0	250.5 ± 6.8

A.11 THRESHOLD-STYLE REWARDS

Table 3: Threshold-style sparse rewards on the HHH basis with scalar feedback, (profile definitions in Appendix A.9.3). Cumulative reward at $T = 100$ and last-10 pseudo-regret, mean ± 1 SE across runs. ICL does not use the decoder, and its pseudo-regret is omitted because computing it requires the agent's posterior.

	T1		T2	
Agent	Cum. reward at $T = 100$	Last-10 pseudo-regret	Cum. reward at $T = 100$	Last-10 pseudo-regret
linear-$\boldsymbol{\lambda}$ TS, SIR ($\tau = 1$)	175.8 ± 6.8	2.24 ± 0.02	385.8 ± 16.2	1.10 ± 0.03
linear-$\boldsymbol{\lambda}$ TS, argmax	464.4 ± 4.9	2.36 ± 0.06	510.4 ± 4.4	1.05 ± 0.01
GP TS, SIR ($\tau = 1$)	314.2 ± 29.7	1.10 ± 0.05	442.8 ± 7.5	0.41 ± 0.01
GP TS, argmax	532.6 ± 1.5	1.14 ± 0.05	462.3 ± 59.5	0.85 ± 0.32
ICL	198.3 ± 10.4	–	284.3 ± 9.1	–

A.12 FEEDBACK CHANNELS

Figure 6 and Table 4 compare the three feedback channels on the linear world of Figure 1: HHH basis, the seven $\boldsymbol{\lambda}_\star$ profiles, $K = 100$, $T = 100$, scalar feedback at $\sigma = 0.25$. On pairwise the linear agent is the linear-Gaussian belief with the logistic ADF update (Appendix A.4); no pairwise GP belief exists.

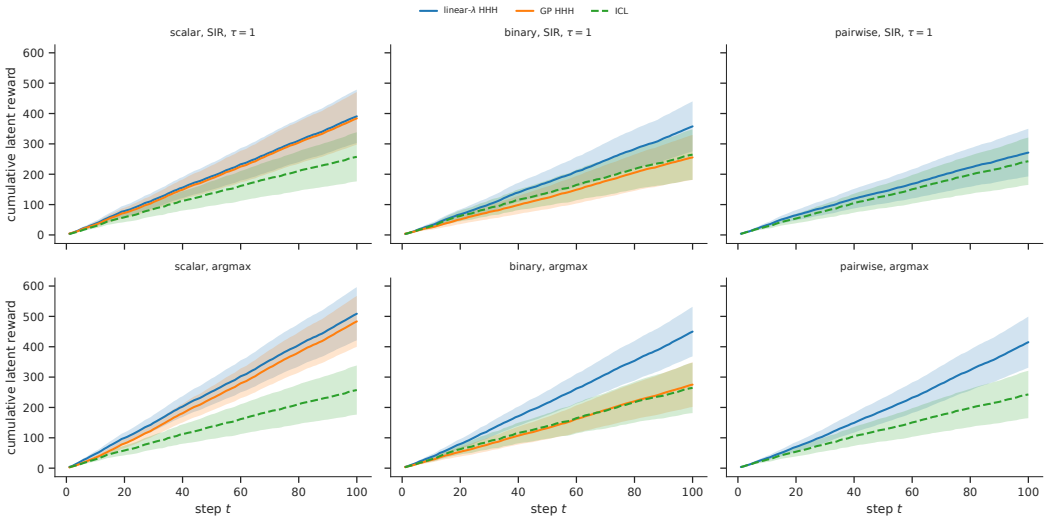


Figure 6: Linear world across feedback channels (columns) and decode rules (rows). Cumulative latent reward of the served responses for linear-$\boldsymbol{\lambda}$ TS, GP TS (scalar and binary only) and ICL. Mean ± 1 SE.

Table 4: Cumulative latent reward at $T = 100$ per feedback channel, linear world. ICL does not use the decoder. linear-$\boldsymbol{\lambda}$ $-$ ICL is the gap averaged over seeds within each $\boldsymbol{\lambda}_\star$ profile, then mean ± 1 SE over the seven profiles.

		SIR, $\tau = 1$			argmax		
channel	ICL	linear-$\boldsymbol{\lambda}$	GP	linear-$\boldsymbol{\lambda}$ $-$ ICL	linear-$\boldsymbol{\lambda}$	GP	linear-$\boldsymbol{\lambda}$ $-$ ICL
scalar	257.3 ± 81.0	390.7 ± 88.0	384.1 ± 86.8	133.4 ± 22.2	508.8 ± 87.7	483.5 ± 83.9	251.4 ± 25.7
binary	264.5 ± 83.1	357.6 ± 82.4	255.4 ± 74.2	93.1 ± 19.8	449.8 ± 81.7	275.2 ± 73.7	185.4 ± 23.9
pairwise	242.8 ± 78.1	271.3 ± 78.6	–	28.4 ± 4.6	415.1 ± 84.3	–	172.3 ± 28.3

A.13 SHARPNESS OF THE SIR TILT

With tilt logits $z_i = r(x_t, y_t^i)$ over the K candidates,

$$w_i \;\propto\; e^{z_i/\tau_t}, \qquad \mathrm{KL}_t \;=\; \sum_{i=1}^{K} w_i \log(K w_i) \;\in\; [0, \log K],$$

The rules in Figure 5:

$$\begin{aligned} \tau = 1: &\quad \tau_t = 1 \\ \text{std}: &\quad \tau_t = \mathrm{sd}_i(z_i) \\ \text{std}/16: &\quad \tau_t = \mathrm{sd}_i(z_i)/16 \\ \text{kl:2}: &\quad \tau_t \text{ such that } \mathrm{KL}_t = 2 \\ \text{argmax}: &\quad \tau_t \to 0 \end{aligned}$$

Only $\tau = 1$ depends on the scale of r, the other are scale-free.

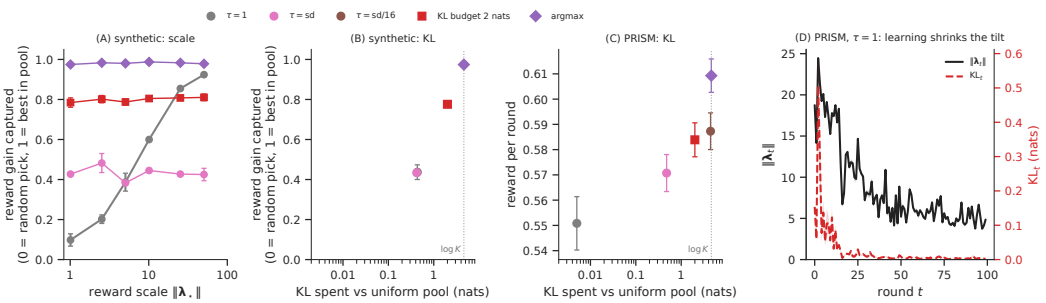


Figure 7: **At $\tau = 1$, sharpness depends on the reward scale; scale-free rules do not.** (A) Synthetic: capture against reward scale $\|\boldsymbol{\lambda}_\star\|$ at fixed SNR, by decode rule. (B) Synthetic: capture against KL spent. (C) PRISM (ArmoRM-19, linear-$\boldsymbol{\lambda}$, 10 users): reward per round against KL spent. Dotted line in (B, C) at $\log K$. (D) PRISM at $\tau = 1$: $\|\boldsymbol{\lambda}_t\|$ and KL_t over rounds. Mean ± 1 SE.

A.14 Paired differences to ICL on the persona benchmark

Table 5: Paired difference to ICL (agent minus ICL), cumulative reward at $T = 100$, PRISM, Opus oracle, 10 users; mean ± 1 SE over users; in parentheses, the number of users (of 10) for whom the difference is positive. $\tau = 1$: one run per user; argmax: mean of three.

belief	basis	gap, $\tau = 1$	gap, argmax	decoder effect
linear-$\boldsymbol{\lambda}$	HHH ($D = 3$)	$+0.9 \pm 1.1$ (7/10)	$+4.9 \pm 0.7$ (10/10)	$+4.0 \pm 1.3$ (9/10)
	HHH+HP ($D = 5$)	$+1.0 \pm 1.2$ (8/10)	$+4.1 \pm 0.5$ (10/10)	$+3.1 \pm 1.5$ (7/10)
	13q ($D = 13$)	$+2.0 \pm 0.7$ (8/10)	$+2.5 \pm 0.6$ (9/10)	$+0.5 \pm 0.9$ (6/10)
	ArmoRM-19	$+1.4 \pm 0.7$ (7/10)	$+6.0 \pm 1.5$ (10/10)	$+4.6 \pm 1.5$ (9/10)
GP	HHH ($D = 3$)	$+1.6 \pm 0.8$ (8/10)	$+2.3 \pm 0.6$ (9/10)	$+0.7 \pm 0.7$ (7/10)
	HHH+HP ($D = 5$)	$+2.2 \pm 0.7$ (9/10)	$+2.2 \pm 1.0$ (8/10)	-0.0 ± 1.4 (3/10)
	13q ($D = 13$)	$+2.4 \pm 1.1$ (9/10)	$+1.9 \pm 0.8$ (8/10)	-0.5 ± 0.9 (4/10)
	ArmoRM-19	$+2.7 \pm 0.8$ (9/10)	$+5.0 \pm 1.6$ (10/10)	$+2.2 \pm 1.9$ (5/10)
none	generic head ($w = +1$)	$+2.0 \pm 0.6$ (9/10)	$+11.9 \pm 2.1$ (10/10)	$+9.9 \pm 1.9$ (10/10)
linear-$\boldsymbol{\lambda}$	basis effect	$+1.1 \pm 1.4$ (6/10)	$+3.4 \pm 1.5$ (7/10)	–
GP	basis effect	$+1.1 \pm 1.1$ (8/10)	$+3.1 \pm 1.9$ (6/10)	–

basis effect = largest − smallest learned basis, paired: linear-$\boldsymbol{\lambda}$: 13q − HHH ($\tau = 1$), ArmoRM-19 − 13q (argmax); GP: ArmoRM-19 − HHH ($\tau = 1$), ArmoRM-19 − 13q (argmax).

A.15 What per-user learning buys on the persona benchmark

Table 6: Cumulative reward at $T = 100$, argmax, identical pools, Opus oracle, 10 users, mean ± 1 SE; paired difference to the online linear posterior on HHH; in parentheses, the number of users (of 10) for whom the difference is positive.

arm	user labels used	reward	vs. online linear (HHH)
random direction, HHH	none	54.5 ± 1.6	-4.0 ± 1.1 (0/10)
ICL	all, in context	53.7 ± 1.9	-4.9 ± 0.7 (0/10)
online GP TS, HHH	all, online	56.0 ± 1.8	-2.6 ± 0.7 (1/10)
online linear-$\boldsymbol{\lambda}$ TS, HHH	all, online	58.6 ± 1.8	–
online linear-$\boldsymbol{\lambda}$ TS, ArmoRM-19	all, online	59.7 ± 2.3	$+1.1 \pm 0.9$ (6/10)
frozen linear fit, HHH, $n = 5$	5, offline	59.7 ± 2.0	$+1.2 \pm 0.5$ (8/10)
frozen linear fit, HHH, $n = 100$	100, offline	58.2 ± 1.8	-0.4 ± 0.4 (4/10)
generic head ($w = +1$)	none	65.6 ± 2.7	$+7.0 \pm 1.5$ (10/10)

A.16 SELECTING A STYLE DIRECTIVE INSTEAD OF A CANDIDATE

The proposed directive bandit has 11 arms. Each arm is a system prompt: no directive, or one of 10 fixed style directives, such as "answer bluntly and directly" or "answer in a formal, professional register". Each round it picks one arm by Gaussian Thompson sampling over the arms' persona rewards. It generates a single completion under that system prompt. The user scores it, and only the chosen arm's posterior is updated. Everything else matches §5.2. The bandit uses no quality features and one generation per round (vs.$K = 100$ for SIR).

The bandit learns user-specific arms, but steering among a fixed set of text styles recovers less reward than choosing sharply from K i.i.d. samples, and it does not reach ICL.

Table 7: Directive bandit against the main arms, cumulative reward at $T = 100$, 10 users; paired difference is bandit minus arm, mean ± 1 SE over users; in parentheses, the number of users (of 10) for whom the bandit scores higher.

arm	cumulative reward	bandit minus arm
directive bandit	52.2 ± 2.8	–
ICL	53.7 ± 1.9	-1.5 ± 2.2 (1/10)
linear-$\boldsymbol{\lambda}$, HHH, $\tau = 1$	54.6 ± 2.2	-2.4 ± 2.6 (3/10)
linear-$\boldsymbol{\lambda}$, HHH, argmax	58.6 ± 1.8	-6.4 ± 1.8 (1/10)
generic head, argmax	65.6 ± 2.7	-13.4 ± 1.1 (0/10)

A.17 SHARED AND USER-SPECIFIC VARIATION IN PERSONA RATINGS

All arms on the PRISM benchmark select from the same cached candidate pools, and all ten users receive the same 100 prompts. Many candidates were rated by the simulated users. The same user also rated the same candidate in different runs, which measures impersonation noise. For the same prompt, we decompose the variation of feedbacks among candidates into a part shared by all users, a user-specific part, and rater noise. User-specific preferences are 16 to 33% of the non-noise variation.

Table 8: Variance of ratings among candidates for the same prompt, PRISM, Opus oracle, scores standardised per user.

noise estimate	shared	user-specific	rater noise
all repeated ratings	0.55	0.27	0.18
differing repeats only (upper bound)	0.55	0.10	0.34

We test if a quality basis can represent the users by regressing each user's standardised ratings on the basis, both within user and prompt, with 5-fold cross-validation over prompts, once with a single weight vector shared by all users and once with a separate weight vector per user (≈3k ratings per user).

Table 9: Cross-validated R^2 of a user's within-prompt ratings; the last row uses no features: it predicts a user's rating by the mean rating of the other users on the same candidate. Per-user weights add at most 0.03 R^2 even if fitted at 3 times the budget of the TS arms.

predictor	shared weights	per-user weights	gain from per-user weights
overall ArmoRM score ($D = 1$)	0.212	0.236	$+0.025$
HHH ($D = 3$)	0.068	0.077	$+0.009$
HHH+HP ($D = 5$)	0.080	0.086	$+0.006$
13q ($D = 13$)	0.076	0.083	$+0.008$
ArmoRM-19	0.229	0.255	$+0.027$
other users' mean rating	0.226	–	–

Table 10 compares the final posterior means of linear-$\boldsymbol{\lambda}$ TS (argmax, $T = 100$).

Table 10: Mean cosine similarity between final posterior means of linear-$\boldsymbol{\lambda}$ TS, PRISM, argmax.

basis	two users, same belief seed	same user, two belief seeds
HHH ($D = 3$)	0.91	0.89
ArmoRM-19	0.45	0.66

User-specific variation lies outside the span of the quality bases, and the shared variation is captured similarly by the single overall ArmoRM score as by the other users' feedback signals. Thus, a general-purpose score, applied identically to every user, outperforms per-user learning on this benchmark (§5.4).