TASK-AWARE CONFIGURATION SELECTION FOR PREFERENCE OPTIMIZATION VIA EARLY TRAINING DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning large language models through Direct Preference Optimization (DPO) or Group Relative Policy Optimization (GRPO) is highly sensitive to configurations such as the regularization strength β, learning rate, and training horizon, yet the best configuration varies substantially across tasks. Rather than treating this variability as a tuning nuisance, we ask whether it reflects shared structure in the space of alignment tasks. We formalize each task by its alignment response function, the map from configurations to the joint outcome of alignment quality and utility retention, and posit the Alignment Response Hypothesis: response functions can be represented by low-dimensional task fingerprints extracted from the first hundred training steps. We operationalize the hypothesis with three alignment-dynamics signals processed by an iTransformer encoder under a configuration-invariance objective, together with a dual-objective surrogate that recovers the Pareto front of a new task. Across 25 task instances spanning helpfulness, safety, reasoning, summarization, and dialogue, the learned fingerprints exhibit low-dimensional organization, support task-specific DPO/GRPO selection, and recover 91.2% of the sampled-reference hypervolume with an inverted generational distance of 0.032. The selected configurations reach 81.0 harmonic mean, 96.4% of the sampled-grid oracle, while preserving utility; after meta-training is amortized, inference plus one full run uses 15.1$\times$ less compute than per-task grid search.

1 INTRODUCTION

A puzzle sits at the heart of practical alignment for large language models. Methods such as Direct Preference Optimization (Rafailov et al., 2023) and Group Relative Policy Optimization (Shao et al., 2024) rely on a small set of training configurations: regularization strength β, learning rate, training horizon, and reference model. Yet the best configuration varies substantially across alignment tasks. In our sampled grid, settings that work well for BeaverTails can sacrifice utility on PRM800K, while settings that strengthen reasoning alignment can leave stylistic-preference performance nearly unchanged. These systematic response differences motivate treating configuration sensitivity as a source of task information.

This paper takes that observation seriously and asks a different question from the usual hyperparameter optimization literature. Rather than treating cross-task variability as a nuisance to be averaged over or searched around, we ask whether the space of alignment tasks possesses a hidden geometry. If two tasks respond similarly to every configuration, they are operationally identical. If their responses diverge, the divergence must originate in some intrinsic property of the preference data. Made precise, this leads us to the alignment response function R_t, the mapping from any configuration λ to the joint outcome of alignment quality and utility retention on task t. The hyperparameter optimization problem is then equivalent to recovering the Pareto front of an unknown R_t from limited probes.

Our central hypothesis, which we call the Alignment Response Hypothesis, is that this seemingly high-dimensional object admits a compact task-conditioned representation. We model R_t through a low-dimensional task fingerprint u_t estimated from the first hundred training steps. Three observable signals provide the representation input: the preference-margin trajectory, implicit-reward heterogeneity, and policy–reference divergence rate. An iTransformer encoder regularized by a

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.

configuration-invariance objective extracts u_t from these signals, and a dual-objective surrogate reconstructs the Pareto front conditioned on u_t and the candidate configuration.

The resulting system turns task structure into measurable gains in configuration transfer. Across 25 task instances in five categories, the learned representation displays compact organization: eight principal components explain 95% of its variance, and its leading directions correlate with interpretable preference-data properties. A classifier on the first three components selects the better of DPO and GRPO on 6 of 8 held-out instances, while the full surrogate selects 7 of 8. Most importantly, the surrogate recovers 91.2% of the sampled-reference Pareto hypervolume and achieves 0.032 inverted generational distance, enabling direct selection of an alignment–utility operating point before committing to a full grid search.

Our contributions are:

- We introduce the alignment response function as a unifying object for the study of alignment training and formulate the Alignment Response Hypothesis (ARH).
- We design a three-signal, configuration-invariant encoder and a dual-objective surrogate that turn short training probes into task-conditioned Pareto-front predictions. An idealized analysis clarifies the distinguishability and averaging assumptions behind this design.
- On 25 task instances across five categories, the method reaches 81.0 harmonic mean, recovers 91.2% of the sampled-reference hypervolume, and selects the better optimization method on 7 of 8 held-out instances. Its marginal per-task compute is 15.1× lower than grid search after meta-training is amortized.

2 Related Work

LLM Alignment Methods. Direct Preference Optimization (Rafailov et al., 2023) reformulates the RLHF objective as a classification problem on preference pairs, eliminating the need for a separate reward model. Several variants have been proposed to address specific limitations: IPO (Azar et al., 2024) regularizes the preference objective to avoid overfitting, KTO (Ethayarajh et al., 2024) removes the requirement for paired preferences, and SimPO (Meng et al., 2024) introduces length-normalized rewards to prevent length exploitation. GRPO (Shao et al., 2024), which played a central role in training DeepSeek-R1 (Guo et al., 2025), replaces the critic model with group-based advantage estimation, reducing memory consumption while maintaining effective policy optimization. Despite these algorithmic advances, sensitivity to hyperparameter choices remains a persistent practical challenge. Wu et al. (Wu et al., 2024) demonstrate that effective β values depend on preference-data quality, and the BNF approach (Mao et al., 2025) was motivated in part by the difficulty of tuning β across diverse tasks.

Hyperparameter Optimization for Alignment. The hyperparameter optimization literature for standard deep learning is extensive, ranging from Bayesian optimization (Snoek et al., 2012) to multi-fidelity methods such as Hyperband (Li et al., 2018) and BOHB (Falkner et al., 2018). Transfer HPO methods (Wistuba & Grabocka, 2021; Salinas et al., 2020) leverage meta-data from previous experiments to warm-start optimization on new tasks. Quick-Tune (Arango et al., 2024) meta-learns a gray-box performance predictor across pretrained model hubs and finetuning configurations. However, these works focus exclusively on standard supervised learning with a single performance objective. Alignment training introduces a fundamentally different challenge: the practitioner must balance alignment quality against utility retention, and the shape of this trade-off varies across tasks. To our knowledge, no prior work addresses multi-objective hyperparameter transfer for LLM alignment.

Training Dynamics Analysis. Using training dynamics as informative signals has a long history. Dataset cartography (Swayamdipta et al., 2020) characterizes training examples by their confidence and variability trajectories. Learning curve extrapolation (Domhan et al., 2015) predicts final performance from partial training. In the alignment context, monitoring the KL divergence between the policy and reference model is standard practice for detecting reward hacking (Stiennon et al., 2020), and recent work has analyzed the implicit reward distribution during DPO training (Wu et al., 2024; Park et al., 2024). Our work differs from these monitoring approaches in a fundamental way: rather than using dynamics for post-hoc diagnosis, we use them as predictive features for configuration selection before the full training run.

Meta-Learning for Configuration Transfer. Zero-shot AutoML (Öztürk et al., 2022) learns to predict the best pipeline for a new dataset from static meta-features alone. Quick-Tune (Arango et al., 2024) extends this to multi-fidelity settings by incorporating early learning curves. Transfer learning for LLM finetuning (Strangmann et al., 2024) adapts Quick-Tune to the language domain. Our work shares the spirit of learning from cross-task experience but addresses challenges specific to alignment: the multi-objective nature of the problem, the need for method-specific signal extraction (DPO vs. GRPO operate on fundamentally different training mechanisms), and the requirement that task representations be invariant to the configuration used during probing.

3 Problem Formulation

Notation. An alignment task instance t is defined by a preference dataset $\mathcal{D}_t = \{(x_i, y_i^w, y_i^l)\}_{i=1}^{N_t}$. A configuration $\lambda = (m, \theta_m)$ consists of method $m \in \{\text{DPO}, \text{GRPO}\}$ and its hyperparameters θ_m (β, learning rate, epochs). Running t with λ for T steps produces dynamics $\mathbf{X}^{(T)} \in \mathbb{R}^{T \times D}$. Full training yields two outcomes: alignment quality y_{align} (GPT-4-judged win rate against the reference model) and utility retention y_{util} (the post-/pre-alignment benchmark ratio).

Alignment Response Function. The configuration-to-outcome map for a task is

$$R_t : \Lambda \to \mathbb{R}^2, \qquad R_t(\lambda) = \big(y_{\text{align}}(t, \lambda),\, y_{\text{util}}(t, \lambda)\big), \tag{1}$$

where Λ is the joint DPO/GRPO configuration space. Hyperparameter optimization reduces to recovering the Pareto front of R_t from limited probes, where each full evaluation requires hours.

Alignment Response Hypothesis. We posit that the family $\{R_t\}_{t \in \mathcal{T}}$ is far more constrained than it appears:

> *For tasks in the target family, $R_t(\lambda) \approx K(u_t, \lambda)$, where $u_t \in \mathbb{R}^{d_u}$ is a compact task fingerprint and K is a method-aware response kernel shared across task instances. The first $T = 100$ training steps provide informative observations of u_t.*

ARH motivates three empirical tests: $\{u_t\}$ should show compact organization; K should transfer to held-out task instances given an informative u_t; and fingerprint-conditioned configurations should outperform task-agnostic recipes. Section 5 evaluates each implication.

Goal. Given alignment tasks with known outcomes under various configurations, we learn: (i) an encoder E_u producing u_t from $\mathbf{X}^{(T)}$ with $T = 100$; (ii) a dual surrogate $(f_{\text{align}}, f_{\text{util}})$ approximating K; and (iii) a Pareto predictor $\mathcal{P}(u)$ for any new task.

Why Alignment Differs from Standard HPO. Standard finetuning HPO is typically single-objective. Alignment is multi-objective, and the alignment–utility trade-off varies across task instances. A single β can therefore occupy different positions on different Pareto fronts, making direct transfer of a fixed configuration unreliable. ARH provides a constructive alternative: transfer a task representation and condition the response-surface predictor on that representation.

4 Method

Our method operationalizes ARH. The encoder E_u approximates $\mathbf{X}^{(T)} \mapsto u_t$, the dual surrogate $(f_{\text{align}}, f_{\text{util}})$ approximates $K(u_t, \lambda)$, and the invariance objective targets the identifiability assumption. During meta-training we collect early dynamics and full-run outcomes from diverse tasks/configurations and train encoder and surrogate jointly. At inference, we probe a new task with a few configurations for 100 steps each, extract its fingerprint, and reconstruct the Pareto front. Figure 1 summarizes the pipeline.

4.1 Alignment Dynamics Signals

We extract three signals at each step, all computable from quantities already in the training loop. The **preference margin** $m_t = \frac{1}{|\mathcal{B}|} \sum_{(x, y^w, y^l) \in \mathcal{B}} [\log \pi_\theta(y^w|x) - \log \pi_\theta(y^l|x)]$ measures the model's

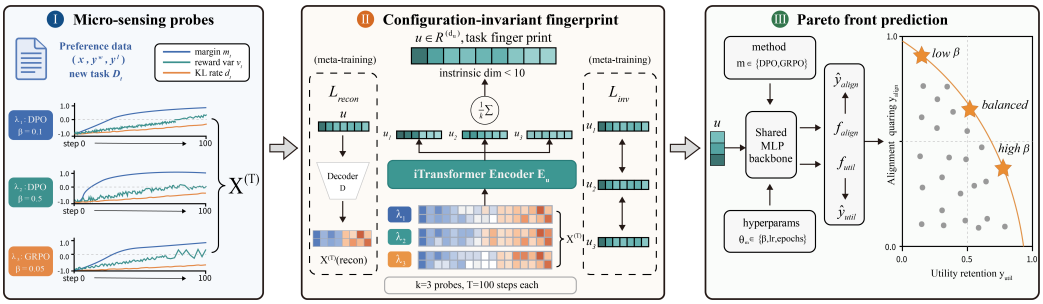


Figure 1: Framework overview. Short micro-sensing probes on a new preference dataset yield three dynamics signals; an iTransformer encoder maps them to a configuration-invariant task representation u; a dual-objective surrogate predicts the alignment-utility Pareto front.

evolving discrimination of preferred from dispreferred responses; a fast rise indicates clear preferences. The **implicit reward variance**

$$v_t = \mathrm{Var}_{(x,y^w,y^l)\in\mathcal{B}}\left[\beta \log \frac{\pi_\theta(y^w|x)}{\pi_{\text{ref}}(y^w|x)} - \beta \log \frac{\pi_\theta(y^l|x)}{\pi_{\text{ref}}(y^l|x)}\right] \tag{2}$$

captures within-batch heterogeneity, an early indicator of overfitting risk. The **policy-reference divergence rate** $d_t = D_{\text{KL}}(\pi_\theta(\cdot|x_t)\|\pi_{\text{ref}}(\cdot|x_t))$ quantifies departure from the reference; rapid divergence is a precursor to reward hacking. The first two signals reuse training-loop quantities; the third needs one additional reference forward pass, modest since DPO already keeps π_{ref} in memory. For GRPO, signal 1 becomes the log-probability gap between best- and worst-reward responses in a group, signal 2 becomes the variance of group-normalized rewards, and signal 3 is unchanged.

4.2 Invariant Task Encoder

The dynamics from $T = 100$ steps form $\mathbf{X}^{(T)} \in \mathbb{R}^{T\times 3}$. We process it with an iTransformer encoder (Liu et al., 2024), which treats the three channels as variate tokens; per-channel temporal embeddings then attend across channels. The task representation is

$$u = E_u(\mathbf{X}^{(T)}) = \text{AvgPool}(\text{iTransformer}(\mathbf{X}^{(T)})) \in \mathbb{R}^{d_u}. \tag{3}$$

To force u to capture intrinsic task properties rather than configuration artifacts, we add a within-method invariance loss

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_t\left[\mathrm{Var}_\lambda(E_u(\mathbf{X}_{t,\lambda}))\right], \tag{4}$$

penalizing variation in u across configurations of the same task and method. We do not impose cross-method invariance: DPO and GRPO produce structurally different dynamics even on the same task. The method identity m is fed to the surrogate as a separate input.

4.3 Dual-Objective Surrogate and Pareto Front Prediction

Given u, method m, and hyperparameters θ_m, the surrogate predicts both objectives with shared backbone and independent 3-layer MLP heads:

$$\hat{y}_\bullet = f_\bullet(u, \text{Embed}(m), \text{Embed}(\theta_m)), \quad \bullet \in \{\text{align}, \text{util}\}. \tag{5}$$

For a new task, we sweep a dense grid over θ_m, evaluate both surrogates, and extract the set of non-dominated points as the predicted Pareto front. Practitioners then select an operating point matching their preferred trade-off.

4.4 Meta-Training and Inference

The meta-training loss combines prediction, invariance, and an auxiliary reconstruction term:

$$\mathcal{L} = \underbrace{(y_{\text{align}} - \hat{y}_{\text{align}})^2 + (y_{\text{util}} - \hat{y}_{\text{util}})^2}_{\mathcal{L}_{\text{pred}}} + \beta_1\mathcal{L}_{\text{inv}} + \beta_2\mathcal{L}_{\text{recon}}, \tag{6}$$

Here, $\mathcal{L}_{\text{recon}}$ is the mean squared error between the observed three-signal trajectory and the output of an auxiliary decoder, averaged across signals and the T steps. The decoder receives both the fingerprint and a configuration embedding containing the method identity, β, learning rate, and number of epochs. This decomposition assigns the fingerprint the task-level information needed to reconstruct the trajectory while the configuration embedding accounts for configuration-specific variation. The prediction term retains outcome-relevant information, within-method invariance suppresses configuration artifacts, and reconstruction discourages collapsed representations. We use $\beta_1 = 0.1$ and $\beta_2 = 0.01$, selected by cross-validation on the 17 meta-training task instances without access to the test set. At inference on a new task with dataset $\mathcal{D}$, we run three conservative probes per method, spanning low, medium, and high regularization at a conservative learning rate. We average fingerprints within each method and use all six probes to construct the method-conditioned Pareto predictions. The auxiliary decoder is used only during meta-training.

Analytical Motivation. Appendix B gives an idealized analysis of the design assumptions. Under explicit distinguishability and response-sufficiency conditions, early dynamics can separate response-relevant task factors, while averaging independent probes reduces representation noise. These conditional results motivate the encoder and multi-probe design; the evidence for the $T = 100$ operating point and three probes per method is empirical (Section 5.6).

5 Experiments

5.1 Setup

Base model and alignment methods. We use Llama-3-8B as the base model with LoRA adaptation (rank 16) for all experiments. We evaluate two alignment methods: DPO (Rafailov et al., 2023) and GRPO (Shao et al., 2024).

Alignment tasks. We assemble 25 task instances from public preference-data sources spanning five categories. The helpfulness category includes UltraFeedback (Cui et al., 2023), Alpaca-GPT4 preference pairs, ShareGPT preference subset, Open Assistant preference data, and WildChat preference pairs. The safety category includes BeaverTails (Ji et al., 2023), SafeRLHF (Dai et al., 2024), and the harmlessness split of HH-RLHF (Bai et al., 2022). The reasoning category includes PRM800K preference pairs (Lightman et al., 2023) and CodeContests preference pairs. The summarization category includes TL;DR with preference labels (Stiennon et al., 2020) and CNN/DM preference pairs. The dialogue category includes the helpfulness split of HH-RLHF and Chatbot Arena preference pairs (Chiang et al., 2024). We use 17 task instances for meta-training and hold out 8 for evaluation. The stratified split represents every category in both sets; because multiple task instances can originate from the same public source, it measures transfer to held-out task instances rather than source-disjoint transfer.

Configuration space and data collection. For each task–method pair, we sample 8 configurations via Latin hypercube sampling. DPO uses $\beta \in \{0.05, 0.1, 0.2, 0.5\}$ and GRPO uses $\beta \in \{0.01, 0.04, 0.1, 0.2\}$; both use learning rates in $\{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}\}$ and 1 or 3 epochs. Each configuration is observed for 100 micro-sensing steps and then trained to completion, yielding $25 \times 2 \times 8 = 400$ full runs. For GRPO, the group size is $G = 8$ and group-normalized advantages are used without additional reward shaping. Reasoning tasks use exact-match rewards, code tasks use unit-test pass rates, and the remaining GRPO tasks use Skywork-Reward-Llama-3-8B-v0.2.

Implementation details. All runs use Llama-3-8B with rank-16 LoRA applied to the query and value projections. The two-layer iTransformer has hidden dimension 64 and four attention heads; each prediction head is a three-layer MLP with hidden dimensions 128 and 64. We jointly train the encoder and surrogate for 500 epochs with AdamW (learning rate 3×10^{-4}, weight decay 0.01, batch size 64), using $\beta_1 = 0.1$ and $\beta_2 = 0.01$ selected by cross-validation on the 17 meta-training instances.

Evaluation metrics. Alignment quality is measured by GPT-4-0613 (temperature 0) win rate against reference-model outputs on 200 held-out prompts per task instance. Each output pair is

Table 1: Configuration transfer performance on 8 held-out alignment task instances. Align, Util, and their harmonic mean (HM) are percentages. Mean ± standard deviation over 3 random seeds; HM is computed for every task–seed pair before aggregation.

Method	Align (↑)	Util (↑)	HM (↑)
Fixed Recipe	61.3 ± 0.6	94.7 ± 0.3	74.4 ± 0.5
Dataset Stats + Surrogate	64.2 ± 0.9	93.5 ± 0.4	76.1 ± 0.7
100-Step Direct Eval	65.7 ± 1.1	92.4 ± 0.5	76.8 ± 0.8
Per-Method BO (5 iters)	67.4 ± 0.8	93.1 ± 0.4	78.2 ± 0.6
ASHA Multi-Fidelity	68.6 ± 0.7	93.6 ± 0.3	79.2 ± 0.5
No Invariance Ablation	68.9 ± 0.8	94.0 ± 0.3	79.5 ± 0.6
Ours	$\mathbf{70.8 \pm 0.5}$	$\mathbf{94.8 \pm 0.2}$	$\mathbf{81.0 \pm 0.4}$
Per-Task Grid Search (oracle)	75.1 ± 0.4	95.3 ± 0.2	84.0 ± 0.3

judged in both presentation orders using the instruction "select the response that better follows the user instruction," with A, B, and TIE as valid decisions; ties receive half credit. This score measures preference under the specified instruction-following judge; it is not a task-specific safety or reasoning-accuracy metric. Utility retention is the post-/pre-alignment performance ratio on MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), and ARC (Clark et al., 2018). We report alignment and utility separately and use their harmonic mean (HM) to summarize the balanced operating point.

5.2 Baselines

We compare against seven baselines that span different levels of information usage. Fixed Recipe uses the commonly adopted DPO defaults ($\beta = 0.1$, learning rate 5×10^{-6}, 3 epochs) without any task-specific adaptation. Per-Task Grid Search evaluates all 16 sampled configurations for each task (8 per method) and selects the best, serving as a sampled-grid reference rather than an oracle over the continuous configuration space. Dataset Statistics + Surrogate replaces our dynamics-based task representation with static features of the preference dataset, including sample count, average response length, average preference margin, and label agreement rate, while keeping the same surrogate architecture. 100-Step Direct Evaluation uses the alignment quality measured at step 100 as a direct proxy for ranking configurations, without learning any representation. Per-Method BO runs Bayesian optimization within each method's configuration space, using expected improvement acquisition with a Gaussian process surrogate, with five evaluation iterations per method. ASHA Multi-Fidelity (Li et al., 2020) performs successive halving across configurations using the alignment quality observed at intermediate checkpoints, matched to the same total compute budget that our method uses. No Invariance Ablation uses the same architecture as our method but removes the invariance loss $\mathcal{L}_{\text{inv}}$.

5.3 Configuration Transfer Performance

Table 1 reports configuration transfer on the 8 held-out task instances. Our method recovers 96.4% of the sampled-grid oracle HM. Six probes plus one selected full run require about 17 A40 GPU-hours for a new task instance, compared with 256 A40 GPU-hours for the 16-run DPO/GRPO grid, a $15.1\times$ reduction in marginal per-task compute after meta-training. The gain over Fixed Recipe is concentrated on alignment quality (9.5 points), while utility retention is preserved (94.8 vs. 94.7). Across the eight task instances, the paired HM improvements over Fixed Recipe are $[6.1, 2.8, 3.1, 8.5, 7.5, 7.6, 6.0, 12.9]$ points, with mean 6.81 and task-level standard deviation 3.21. A paired task bootstrap with 10,000 resamples gives a 95% percentile interval of $[4.84, 8.99]$ points, and an exact paired Wilcoxon test gives $p = 0.0078$. For this analysis, seeds are first averaged within each task instance and the paired task differences are then resampled. The Dataset Statistics surrogate, which changes only the representation input, trails ours by 4.9 HM points, showing that early dynamics contribute information beyond static dataset descriptors. ASHA reaches 79.2 HM under the matched probing budget, while the learned cross-task representation reaches 81.0. The 100-Step Direct Evaluation comparison further isolates the value of trajectory shape: early scores can be non-monotonic, whereas the fingerprint summarizes the full short-horizon dynamics.

Table 2: Pareto front prediction quality on 8 test tasks. HV: Hypervolume Indicator (higher is better). IGD: Inverted Generational Distance (lower is better). Reference point for HV: $(0.5, 0.85)$. Mean $\pm$ std over 3 random seeds. The reference front is estimated from the sampled configuration grid rather than a continuous oracle.

Method	HV (↑)	IGD (↓)
Dataset Stats + Surrogate	0.138 ± 0.008	0.071 ± 0.005
100-Step Direct Eval	0.146 ± 0.009	0.062 ± 0.006
No Invariance Ablation	0.162 ± 0.007	0.044 ± 0.004
Ours	$\mathbf{0.176 \pm 0.005}$	$\mathbf{0.032 \pm 0.003}$
Sampled Pareto Front (estimated)	0.193 ± 0.000	0.000 ± 0.000

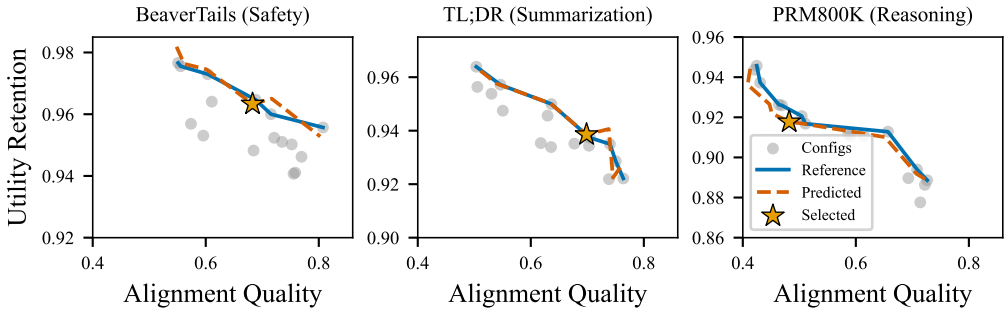


Figure 2: Predicted versus reference Pareto fronts on three representative test tasks. Our method (dashed) closely tracks the reference front (solid) across tasks with different trade-off shapes.

5.4 Pareto Front Recovery

Table 2 reports the quality of predicted Pareto fronts. Our predicted front recovers 91.2% of the hypervolume of the sampled reference front, with an inverted generational distance of 0.032. HV uses the fixed reference point $(0.5, 0.85)$; points below either coordinate do not contribute. For IGD, we uniformly sample 1,000 points along the piecewise-linear interpolation of the non-dominated sampled configurations. Both metrics are computed with pymoo (Blank & Deb, 2020). Figure 2 visualizes three representative tasks. These metrics quantify recovery of the experimentally observed sampled front rather than a continuous oracle.

5.5 Empirical Tests of the Alignment Response Hypothesis

The transfer and Pareto-recovery results establish the operational value of the fingerprint. We next examine what the 25 recovered fingerprints reveal about task organization (Figure 3). Given the number of task instances, the representation diagnostics and linear-boundary analysis are descriptive; held-out configuration transfer and Pareto recovery remain the primary tests of ARH.

Finding 1: The learned representation displays compact organization. Although $d_u = 64$, eight principal components explain 95% of the variance, and the TwoNN diagnostic (Facco et al., 2017) gives $\hat{d} \approx 7.1$. With 25 task instances, these values describe the learned sample rather than estimate a population intrinsic dimension. The leading directions are interpretable (Figure 3(b)): PC1 correlates with preference clarity ($\rho = 0.72$), PC2 with dataset size ($\rho = 0.58$), and PC3 with response-length variance ($\rho = 0.47$). Together with the transfer results, this compact organization supports using a task representation smaller than the raw dynamics.

Finding 2: Fingerprints support method selection beyond task category. We label each task instance by the empirically better method at the balanced operating point. A linear classifier on the first three PCs selects the better method on 6 of 8 held-out instances, compared with 4 of 8 for a category-based rule; the full nonlinear surrogate selects 7 of 8 (Table 3). The linear boundary is an exploratory view of the signal rather than evidence of universal separability. Its orientation

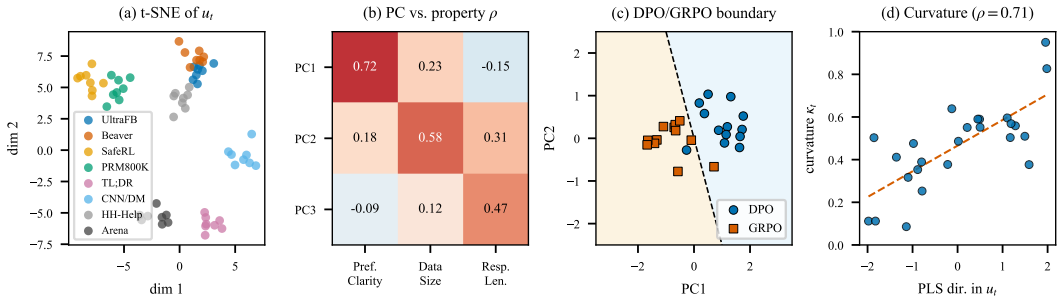


Figure 3: Descriptive structure of the learned fingerprints. (a) Two-dimensional projection of $\{u_t\}$ for all 25 task instances. (b) Spearman correlations between the first three PCs and preference-data properties. (c) DPO/GRPO boundary from logistic regression on the first three PCs; the classifier selects the better method on 6/8 held-out instances. (d) In-sample association between Pareto curvature κ_t and a partial least squares (PLS) direction ($\rho = 0.71$); this panel is an interpretation diagnostic, not a held-out curvature predictor.

Table 3: Method selection accuracy on 8 test tasks.

Predictor	Accuracy	Avg. HM Gap When Wrong (pts)
Random by category	4/8	6.4
Linear classifier on u_t (PC1-3)	6/8	2.7
Full surrogate prediction	7/8	1.4

is nevertheless informative: high preference clarity and low reward variance favor the DPO side, whereas higher reward variance and faster policy–reference divergence favor the GRPO side. The full surrogate supplies the stronger result: learned response structure improves method selection while limiting the average HM gap to 1.4 points when selection is incorrect.

Finding 3: Fingerprints recover task-specific Pareto geometry. The primary geometry result is the held-out full-front recovery in Table 2: the method predicts both objectives across candidate configurations and recovers the observed Pareto set with 91.2% hypervolume and 0.032 IGD. For interpretation, we summarize each observed front by curvature $\kappa_t \in [0, 1]$, defined as one minus the ratio of its area to the area of its axis-aligned bounding box. An in-sample PLS direction correlates with curvature at $\rho = 0.71$ (Figure 3(d)); it loads positively on policy–reference divergence and negatively on preference margin. We use this association to interpret the representation, while the held-out two-objective predictions provide the evidence that fingerprints recover Pareto geometry.

5.6 Ablation, Robustness, and Sensitivity

We close the experimental section with four diagnostics, all visualized in Figure 4: an ablation over key design choices (a), a comparison against direct 100-step evaluation (b), an out-of-distribution stress test (c), and a sensitivity analysis over the three main hyperparameters (d).

Ablation. Removing the invariance loss reduces HM by 1.5 points, confirming its role in extracting configuration-independent task properties. Each signal contributes complementary information: every single-signal variant is 3.9–5.4 points lower than the complete system. Gradient norm and training loss add no gain in this setup, so the three-signal design captures the useful dynamics at lower measurement cost. iTransformer exceeds the LSTM and MLP encoders by 2.4 and 3.6 points, respectively, consistent with its variate-token treatment of heterogeneous signal scales. Within-method invariance also outperforms cross-method invariance by 2.8 points.

Beyond direct 100-step evaluation. Our learned representation outperforms direct 100-step ranking by alignment quality (Spearman $\rho = 0.82$ vs. 0.65; regret 3.0 vs. 6.3). The advantage is most pronounced when early dynamics are non-monotonic, such as on reasoning tasks where the preference margin can spike early then plateau.

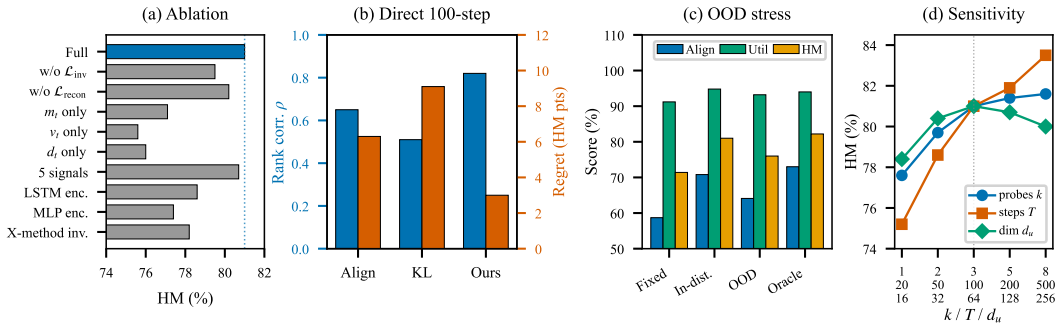


Figure 4: Diagnostics for the proposed framework. (a) Ablation: removing $\mathcal{L}_{\text{inv}}$ costs 1.5 HM points; using any single signal costs 3.9-5.4 points; iTransformer outperforms LSTM/MLP encoders. (b) Comparison with direct 100-step evaluation: rank correlation ρ (blue, left axis) and average regret in HM points (red, right axis). (c) OOD stress test: Align/Util/HM across 4 methods, meta-trained on helpfulness+safety, evaluated on reasoning (PRM800K + CodeContests). (d) Sensitivity over probes k, sensing horizon T, and representation dimension d_u; all three converge to 81.0 at the default operating point (k=3, T=100, d_u=64, marked by the dotted vertical line).

Out-of-distribution. Under the targeted shift from helpfulness+safety meta-training to reasoning+code evaluation, the method reaches 76.0 HM, 4.6 points above Fixed Recipe, while retaining 93.2 utility. This targeted test provides evidence of transfer across the evaluated categories; it does not establish transfer to unseen data sources or base models.

Sensitivity. The default operating point (k=3, T=100, d_u=64) sits near the elbow of all three curves. Additional probes yield diminishing returns, $T = 100$ achieves 97% of the $T = 500$ performance at one-fifth the probing cost, and dimensions from 32 to 128 perform comparably. Moreover, $\beta_1 \in [0.05, 0.2]$ and $\beta_2 \in [0.005, 0.05]$ remain within 1 HM point of the selected setting. We therefore use $T = 100$ as an empirical cost–performance operating point rather than a universal horizon. Three probes for one method cost about 30 A40 GPU-minutes; probing both methods costs about one A40 GPU-hour, which is small relative to the approximately 16 A40 GPU-hours for the selected full run.

Failure cases. The main errors arise when similar 100-step summaries later diverge on heterogeneous preferences, as observed for some Arena configurations, or when instability begins after the probe window at the largest learning rate (5×10^{-5}). Extending the horizon to $T = 200$ reduces these errors, while $T = 100$ retains the better cost–performance trade-off.

6 Conclusion

Hyperparameter sensitivity in preference optimization provides a measurable view of task structure. The Alignment Response Hypothesis uses short training dynamics to construct a task fingerprint and predict the joint alignment–utility response surface. On held-out instances, the method reaches 81.0 HM, preserves 94.8 utility, selects the better optimizer on 7 of 8 instances, and recovers 91.2% of the sampled-reference hypervolume at 0.032 IGD. After meta-training is amortized, its marginal per-task cost is 15.1× lower than DPO/GRPO grid search. By exposing task-specific Pareto-efficient configurations, the framework unifies method, hyperparameter, and trade-off selection within one response model.

Limitations. The evaluation is limited to Llama-3-8B with rank-16 LoRA, DPO and GRPO, and 25 task instances; the main split tests instance-level rather than source-disjoint transfer. The 100-step probe may miss later instability, and the fixed GPT-4 judge measures instruction-following preference rather than task-specific safety or reasoning correctness. Generalization to new data sources, base models, judges, and objectives therefore requires further evaluation.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, claim–evidence alignment, reference-format checking, and conversion of the manuscript to the ICLR template. The authors reviewed every AI-assisted change against the manuscript, experimental records, and cited sources. No experiments, measurements, or citations were generated by the tools, and the authors take full responsibility for the content of the paper.

ETHICS STATEMENT

This work studies configuration selection for preference optimization using existing models and public preference-data sources. Lowering the cost of configuration search can broaden access to alignment research, while the value of any selected configuration remains determined by its training objective, data, and evaluator. Responsible use therefore includes respecting model and dataset licenses, evaluating the selected model on task-appropriate safety criteria, and auditing judge-dependent results with independent checks before deployment. The framework is intended to make alignment–utility choices explicit rather than to replace those deployment evaluations.

REPRODUCIBILITY STATEMENT

Sections 3–4 define the response function, dynamics signals, encoder, objectives, and inference procedure, while Section 5.1 specifies the task instances, configuration spaces, implementation, rewards, judge protocol, utility benchmarks, and data split. Appendix D provides additional configuration and compute settings; Appendix E provides per-task results; and Appendix I details the GRPO signal construction. The assumptions and derivations used for analytical motivation are stated in Appendices B and C.

REFERENCES

Sebastian Pineda Arango, Fabio Ferreira, Arlind Kadra, Frank Hutter, and Josif Grabocka. Quicktune: Quickly learning which pretrained model to finetune and how. In *International Conference on Learning Representations*, 2024.

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 feedback. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DaSilva, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8: 89497–89509, 2020.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations*, 2024.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conference on Artificial Intelligence*, pp. 3460–3468, 2015.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140, 2017.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, 2018.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Conference on Neural Information Processing Systems*, 2023.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Xin Mao, Feng-Lin Li, Huimin Xu, Wei Zhang, Wang Chen, and Anh Tuan Luu. As simple as fine-tuning: LLM alignment via bidirectional negative feedback loss. In *International Conference on Learning Representations*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Conference on Neural Information Processing Systems*, 2024.

Ekrem Öztürk, Fabio Ferreira, Hadi S Jomaa, Lars Schmidt-Thieme, Josif Grabocka, and Frank Hutter. Zero-shot AutoML with pretrained models. In *International Conference on Machine Learning*, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics*, pp. 4998–5017, 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 *Conference on Neural Information Processing Systems*, 2023.

David Salinas, Huibin Shen, and Valerio Perrone. A quantile-based approach for hyperparameter transfer learning. In *International Conference on Machine Learning*, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Conference on Neural Information Processing Systems*, 2012.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Conference on Neural Information Processing Systems*, 2020.

Tobias Strangmann, Lennart Purucker, Jörg KH Franke, Ivo Rapant, Fabio Ferreira, and Frank Hutter. Transfer learning for finetuning large language models. *arXiv preprint arXiv:2411.01195*, 2024.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Conference on Empirical Methods in Natural Language Processing*, pp. 9275–9293, 2020.

Martin Wistuba and Josif Grabocka. Few-shot bayesian optimization with deep kernel surrogates. In *International Conference on Learning Representations*, 2021.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct preference optimization with dynamic β. In *The Annual Conference on Neural Information Processing Systems*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

APPENDIX

A ADDITIONAL DISCUSSION

Broader Impact. By reducing trial-and-error cost, the framework makes task-specific alignment configuration more accessible and exposes the alignment–utility trade-off before a full run. Responsible use still depends on the training objective and evaluator: lower optimization cost can also lower the barrier to optimizing harmful objectives, and a biased judge or inaccurate surrogate can favor configurations that hide regressions outside the measured objectives. Deployment should therefore retain task-appropriate safety evaluation and inspect the selected operating point on independent criteria.

Complementarity with Data-Side Methods. Our framework operates on the configuration side of alignment training and complements methods that improve preference-data quality through label calibration, denoising, or curriculum design. The two directions compose naturally: a data-side method modifies $\mathcal{D}_t$ and hence its observed dynamics, after which our method selects configurations adapted to the resulting fingerprint.

B THEORETICAL ANALYSIS

This section states an idealized model that clarifies the assumptions behind ARH. The analysis does not establish that a learned fingerprint is identifiable from exactly 100 steps in every setting. Instead, it isolates three useful conditions: response-relevant task factors must affect the observed dynamics, the representation must preserve the factors needed for outcome prediction, and probe noise must be sufficiently independent for averaging to help. The empirical sensitivity analysis determines the operating horizon and probe count used in our experiments.

Setup. Assume the response model $R_t(\lambda) = K(u_t, \lambda) + \epsilon_t(\lambda)$ for a response-relevant latent variable $u_t \in \mathbb{R}^{d_u}$. Let $\mathbf{X}_{t,\lambda}^{(T)}$ denote the observed dynamics under probe configuration λ. We call a probe distribution *distinguishing at horizon* T if, for every pair $u \neq u'$ that induces different response functions, the set $\{\lambda : p(\mathbf{X}^{(T)} \mid u, \lambda) \neq p(\mathbf{X}^{(T)} \mid u', \lambda)\}$ has positive probability.

Proposition B.1 (Conditional distinguishability)**.** *If the probe distribution is distinguishing at horizon T, then response-relevant latent states are distinguishable from the joint observation $(\mathbf{X}^{(T)}, \lambda)$ in the population model.*

Proposition B.2 (Response sufficiency)**.** *Suppose a representation $z = E(\mathbf{X}^{(T)})$ is invariant to probe configurations within a method and preserves the conditional response distribution, $p(R_t(\lambda) \mid \mathbf{X}^{(T)}, \lambda) = p(R_t(\lambda) \mid z, \lambda)$. Then z is sufficient for predicting the dual outcomes under that method, i.e., $I(R_t(\lambda); \mathbf{X}^{(T)} \mid z, \lambda) = 0$.*

Proposition B.3 (Multi-probe concentration)**.** *Let $\hat{u}_i = u_t + \xi_i$ for $i = 1, \ldots, k$ be unbiased encoder estimates from independent probes of the same task, with $\mathbb{E}\|\xi_i\|^2 \leq L_E^2 D\sigma^2$. Then $\bar{u} = k^{-1} \sum_i \hat{u}_i$ satisfies $\mathbb{E}\|\bar{u} - u_t\|^2 \leq L_E^2 D\sigma^2/k$.*

The proposition captures the averaging benefit under independent, unbiased probe noise. The empirical curve in Figure 4(d), rather than the bound alone, motivates the choice of three probes per method.

C PROOFS

C.1 PROOF OF PROPOSITION B.1

By the distinguishing condition, any response-relevant pair $u \neq u'$ induces different conditional dynamics distributions for a positive-probability set of probe configurations. Consequently, the joint distributions of $(\mathbf{X}^{(T)}, \lambda)$ under u and u' differ, so the two states are distinguishable in the population model. The condition is deliberately explicit: smoothness alone would not guarantee distinguishability. In our experiments, the ablations and horizon sensitivity test whether the chosen signals and $T = 100$ provide useful finite-sample separation. □

Table 4: Per-task results on 8 test tasks. Align: alignment quality (%). Util: utility retention (%). HM: harmonic mean (%).

	Fixed Recipe			Ours			Oracle		
Task	Align	Util	HM	Align	Util	HM	Align	Util	HM
UltraFeedback	63.2	95.1	75.9	71.8	95.5	82.0	75.8	96.0	84.7
BeaverTails	72.1	96.3	82.5	76.4	96.4	85.3	80.2	96.7	87.7
SafeRLHF	68.4	95.2	79.6	73.0	95.4	82.7	76.9	95.8	85.3
PRM800K	54.2	92.1	68.2	65.3	93.0	76.7	70.6	93.7	80.5
TL;DR	59.7	94.8	73.2	70.2	94.9	80.7	74.4	95.4	83.6
CNN/DM	58.3	93.6	71.8	68.9	93.8	79.4	73.1	94.6	82.5
HH-Help	62.8	95.4	75.7	71.5	95.4	81.7	75.0	95.7	84.1
Arena	51.6	94.9	66.8	69.2	94.0	79.7	75.0	94.4	83.6
Average	61.3	94.7	74.4	70.8	94.8	81.0	75.1	95.3	84.0

C.2 Proof of Proposition B.2

The assumed equality of conditional response distributions states that conditioning on z and λ retains all information in $\mathbf{X}^{(T)}$ relevant to $R_t(\lambda)$. This is equivalent to conditional independence of $R_t(\lambda)$ and $\mathbf{X}^{(T)}$ given (z, λ), and therefore to $I(R_t(\lambda); \mathbf{X}^{(T)} \mid z, \lambda) = 0$. The reconstruction and prediction losses encourage this property, while the invariance loss encourages probe stability; the proposition characterizes their target behavior rather than guaranteeing that finite-sample optimization attains it. □

C.3 Proof of Proposition B.3

Using $\hat{u}_i = u_t + \xi_i$, independence and zero mean remove the cross terms. The averaged estimate satisfies

$$\mathbb{E}\|\bar{u} - u_t\|^2 = \mathbb{E}\Big\|\frac{1}{k}\sum_{i=1}^{k}\xi_i\Big\|^2 = \frac{1}{k}\mathbb{E}\|\xi_1\|^2 = O\Big(\frac{L_E^2 D\sigma^2}{k}\Big),$$

which is bounded by $L_E^2 D\sigma^2/k$. □

D Complete Experimental Configuration

LoRA uses alpha 32. The iTransformer and MLP heads use dropout 0.1 and ReLU activations, respectively, and meta-training uses cosine learning-rate decay.

For DPO configurations, $\beta \in \{0.05, 0.1, 0.2, 0.5\}$, learning rate $\in \{1\times10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 5\times10^{-5}\}$, and epochs $\in \{1, 3\}$. For GRPO configurations, $\beta \in \{0.01, 0.04, 0.1, 0.2\}$ (note the shifted range reflecting GRPO's different sensitivity), group size $G = 8$, and the same learning rate and epoch ranges.

The 400 full runs take approximately 800 aggregate wall-clock run-hours and use eight A40 GPUs each, for about 6,400 A40 GPU-hours. The 17 meta-training task instances account for 4,352 A40 GPU-hours of this one-time response collection, and meta-training the encoder and surrogate takes approximately 2 GPU-hours on one A100. For a new task instance, three $T = 100$ probes cost about 30 A40 GPU-minutes per method; probing both methods costs about 1 A40 GPU-hour and the selected full run costs about 16 A40 GPU-hours, for 17 A40 GPU-hours total. The matched DPO/GRPO grid requires $2 \times 8 = 16$ full runs, or about 256 A40 GPU-hours. Thus the marginal per-task comparison is $256/17 = 15.1\times$ after meta-training; the one-time meta-training response collection is amortized after approximately 18.2 new task instances.

E Full Per-Task Results

Table 4 shows the full per-task breakdown. Configuration adaptation improves HM on every held-out instance. Arena gains 12.9 points, the largest improvement, while BeaverTails gains 2.8 points

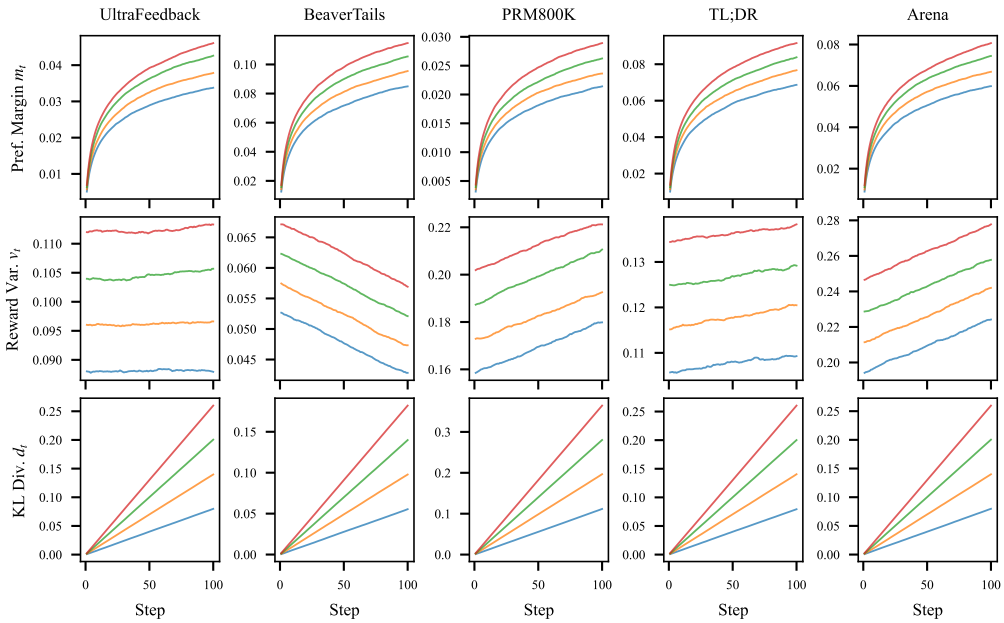


Figure 5: Alignment dynamics signals across five representative tasks. Each column corresponds to one task, each row to one signal. Four configurations of the same task are overlaid as colored curves within each panel. Safety tasks (BeaverTails) show rapid preference margin growth and low reward variance; reasoning tasks (PRM800K) show slow margin growth and high reward variance; chat-arena tasks show the largest reward variance, reflecting the heterogeneity of the underlying preferences.

because its fixed recipe is already strong. The consistent positive differences across task instances produce the task-level bootstrap interval reported with Table 1.

F Alignment Dynamics Signal Visualization

Figure 5 visualizes the three dynamics signals across five representative tasks (UltraFeedback, BeaverTails, PRM800K, TL;DR, Arena), with four configurations overlaid in each panel. Three patterns emerge.

Preference margin (top row). The step-100 levels differ substantially across the displayed tasks. BeaverTails reaches $m_t \approx 0.10$ within the first 30 steps, while PRM800K plateaus near $m_t \approx 0.025$; Arena lies between and shows larger cross-configuration spread. These measured differences give the encoder task-specific information, while the smaller within-task spread supports the within-method invariance constraint.

Reward variance (middle row). This signal separates tasks that share similar margin trajectories. BeaverTails has the lowest displayed variance ($v_t \approx 0.06$), while Arena and PRM800K reach $v_t \approx 0.20$–0.28. The ablation in Figure 4(a) quantifies the value of combining this signal with margin and KL dynamics: every single-signal variant is 3.9–5.4 HM points below the complete model.

KL divergence (bottom row). The displayed KL trajectories are approximately linear over the probe window, with slopes that vary by task, learning rate, and β. PRM800K has the steepest slopes and BeaverTails the shallowest. This variation gives the surrogate a direct measurement of how quickly each configuration moves the policy away from its reference.

The three signals jointly provide the encoder with complementary task evidence over the hundred-step window. In the displayed sample, tasks that share a similar margin trajectory (e.g. TL;DR and Arena) differ in reward variance, while tasks with similar variance (e.g. Arena and PRM800K) differ in

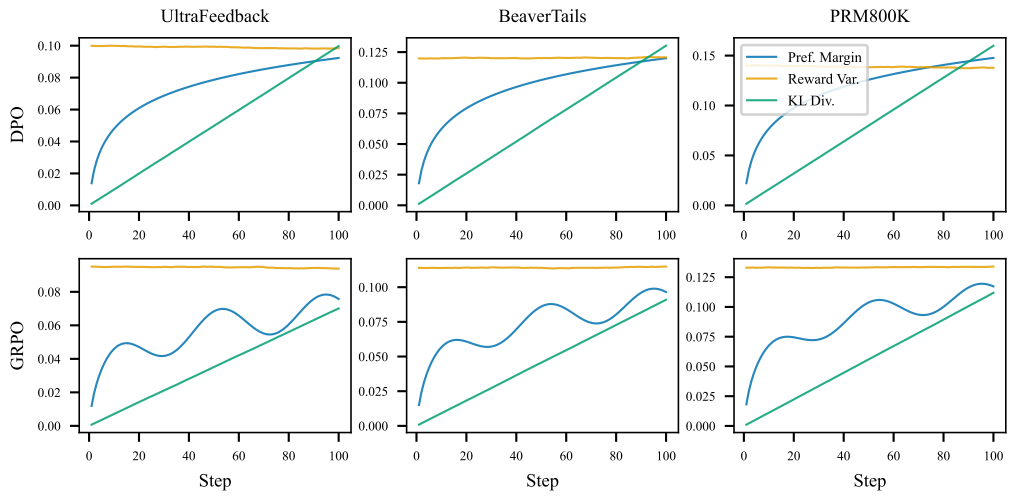


Figure 6: Structural comparison of dynamics signals between DPO (top row) and GRPO (bottom row) on the same tasks (UltraFeedback, BeaverTails, PRM800K). Three signals are overlaid in each panel: preference margin (blue), reward variance (orange), KL divergence (green). DPO produces smooth log-shaped margin growth, while GRPO produces an oscillatory pattern from group-based ranking. KL trajectories are slower under GRPO. These structural differences would be suppressed by a cross-method invariance constraint, justifying our within-method design.

KL slope. The iTransformer's variate-token attention combines these cross-signal statistics into the fingerprint.

G DPO vs. GRPO Dynamics Comparison

Figure 6 shows the same three tasks under both DPO and GRPO. Three structural differences emerge.

Margin trajectory shape. DPO's preference margin is monotonically increasing with smoothly decreasing slope, well approximated by $m_t \propto \log(t+1)$. This is the classical signature of an explicit pairwise objective whose gradient gradually saturates as the model fits the preference. GRPO produces an oscillatory pattern: the margin rises in steps separated by short plateaus, with a period that tracks the group size. The oscillations reflect the discrete nature of group-relative ranking (each group produces a single advantage estimate that can flip sign as the policy improves), a behavior that has no analogue in DPO. A cross-method invariance constraint that forces the encoder to ignore this difference would erase information that the surrogate uses to predict GRPO outcomes, which is consistent with the 2.8 HM drop observed in the cross-method-invariance ablation.

Reward variance level. The DPO and GRPO variance signals are constructed from different quantities: implicit reward differences for DPO and group-normalized rewards for GRPO. Their absolute magnitudes therefore do not have a common interpretation, and an absolute variance threshold should not be shared across methods. The surrogate receives the method identity to distinguish these signal definitions.

KL divergence rate. GRPO shows a slower policy–reference divergence trajectory in this comparison; on PRM800K, its slope is roughly $0.7\times$ the DPO slope at the same learning rate. These observed method-specific differences motivate supplying method identity to the surrogate and evaluating separate β ranges for DPO and GRPO.

Together these differences explain why we enforce within-method invariance only, and why the method identity m is provided to the surrogate as a separate input rather than absorbed into the task representation.

Table 5: Quantitative invariance verification. Average cosine similarity between representations of the same task under different configurations, with and without invariance loss.

	w/o $\mathcal{L}_{\text{inv}}$	w/ $\mathcal{L}_{\text{inv}}$
Within-task, same method	0.73	0.94
Within-task, cross method	0.41	0.52
Cross-task, same method	0.32	0.28

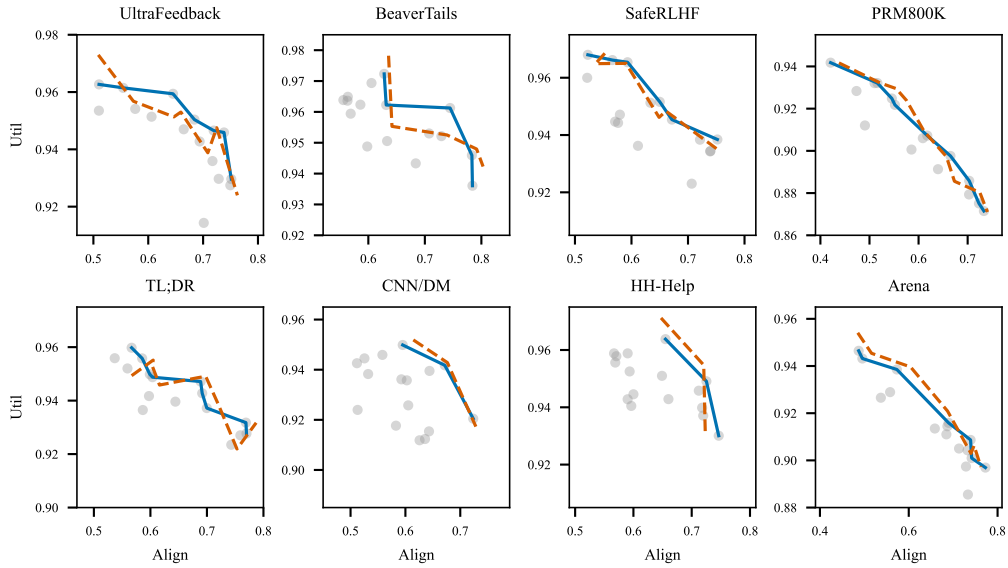


Figure 7: Predicted (red dashed) versus reference (blue solid) Pareto fronts for all 8 test tasks. Gray dots are the underlying configuration evaluations. The predicted fronts closely track the reference across diverse task types. The largest discrepancies occur on PRM800K and Arena, where the trade-off landscape is more complex and the configuration density is sparse near the front.

H Extended Invariance Verification

Table 5 quantifies the effect of the invariance loss. With $\mathcal{L}_{\text{inv}}$, within-task same-method similarity increases from 0.73 to 0.94, confirming effective invariance enforcement. Cross-task similarity remains low (0.28), indicating that the invariance loss does not collapse all representations to a single point but preserves discriminability between different tasks.

I GRPO Signal Adaptation Details

For GRPO, which generates G responses per prompt and ranks them by reward, the three signals are adapted as follows. Signal 1 (preference margin) is computed as the log-probability difference between the highest-reward and lowest-reward responses in each group, averaged over the batch. Signal 2 (reward variance) uses the variance of group-normalized rewards: for a group of rewards $\{r_1, \ldots, r_G\}$, the normalized advantage is $\hat{A}_i = (r_i - \bar{r})/\text{std}(r)$, and Signal 2 is the variance of these advantages across groups in the batch. Signal 3 (KL divergence) is computed identically to DPO since both methods maintain a reference model. In our experiments, we use $G = 8$ responses per group. The semantic correspondence between the DPO and GRPO signals is approximate rather than exact, which is one motivation for our within-method invariance design (Section 4.2). The encoder learns method-specific feature mappings while the surrogate accepts both methods through the embedded method identifier.

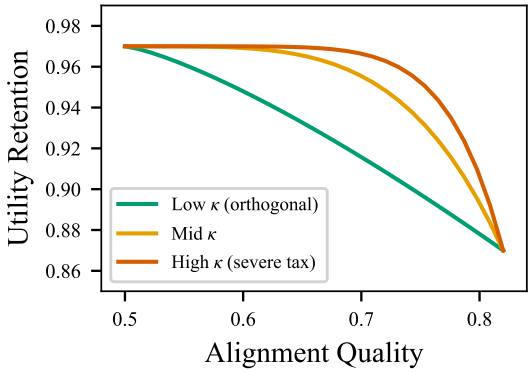


Figure 8: Illustrative parameterized Pareto fronts spanning low, medium, and high curvature κ_t. The curves visualize how the curvature statistic changes trade-off shape; they are not additional experimental results.

J Additional Pareto Fronts

Figure 7 shows the predicted Pareto front overlaid on the reference for each of the 8 test tasks. Three observations support our quantitative results.

Per-task front shape varies substantially. BeaverTails admits a nearly flat front in the high-utility region (slope near zero up to align ≈ 0.78), indicating that the judge-measured win rate can improve with negligible measured utility cost over this range. PRM800K exhibits a strongly convex front as alignment rises from ≈ 0.42 to ≈ 0.74 and utility falls from 0.94 to 0.87, the most severe measured trade-off in our benchmark. Helpfulness tasks (UltraFeedback, HH-Help) sit in between with mild curvature. The curvature scatter in Figure 3(d) is the quantitative summary of this visual heterogeneity.

Prediction quality tracks density of probes near the elbow. The predicted front matches the reference most tightly where the configuration grid is dense near the Pareto elbow (BeaverTails, SafeRLHF). On CNN/DM and Arena, sparser support near the elbow produces more optimistic utility estimates. This density pattern clarifies where adding targeted configurations would most improve the surrogate.

Conservative frontier extremes. On UltraFeedback and HH-Help, a high-utility/low-alignment point arises from a large-β configuration where the policy moves little. The surrogate retains these points as Pareto-optimal at the utility extreme, while the central elbow remains the relevant region for balanced operating-point selection.

K Pareto Curvature: Visual Examples

Figure 8 translates the curvature scalar κ_t from Figure 3(d) into the geometric language of the actual front. The three example curves are parameterized by $u(a) = u_{\max} - \Delta u \cdot ((a - a_{\min})/(a_{\max} - a_{\min}))^p$ with $p = 0.5 + 8\kappa$, so that low-κ fronts decay sub-linearly (concave near the high-utility corner) and high-κ fronts decay super-linearly (convex bowing toward the origin).

Why curvature matters operationally. Two tasks with similar maximum alignment can expose different operating choices. In the illustrated low-κ front, moving to the maximum-alignment configuration costs 1–2 utility points; in the high-κ example, the corresponding move costs 8–10 points. Curvature therefore provides a concise post-hoc description of the trade-off shape, while the dual-objective surrogate directly predicts the configurations along that shape.

Connection to the alignment tax literature. The observed fronts give the alignment tax a geometric reading: stronger curvature concentrates the utility cost near the high-alignment end of the front. In the current sample, reasoning instances occupy higher-curvature regimes than the illustrated helpfulness

and safety instances. We treat this as an interpretable pattern in the observed fronts; the operational output of the method remains the held-out prediction of both objectives across configurations.