Are Easier or Harder Examples Better? Rethinking Data Selection for Reward Models and Preference Optimization

Kevin C. Wibisono[1*] **Aya A. Ismail**[2] **Pedro O. Pinheiro**[3] **Yixin Wang**[1]
Kyunghyun Cho[4] **Nataša Tagasovska**[3] **Rajesh Ranganath**[4]

[1]University of Michigan
[2]Guide Labs
[3]Prescient Design, Genentech
[4]New York University

Abstract

Despite being crucial for effective LLM alignment, data selection remains understudied. Prior work on reward model (RM) training and policy optimization (e.g., DPO, GRPO) identifies example difficulty, the reward gap between chosen and rejected responses, as a key factor, but findings conflict: some favor easier examples with larger gaps, others harder ones. To study difficulty in a controlled setting, we assume access to a reference RM and systematically study data selection across RM, DPO, and downstream GRPO training. When difficulty is measured via the reference RM, easier pairs generally outperform harder ones for RM training and smaller DPO models: using only the top 20% easiest examples often achieves performance comparable to full-dataset training while using one fifth of the training pairs. Independent evaluations support these trends, while GRPO results are more variable. However, this advantage hinges on reward estimation quality, target alignment, and model scale. As the difficulty signal is corrupted by noise or estimated with a weak proxy RM, the easy-example advantage shrinks and can reverse. A signal-to-noise analysis provides a partial explanation: under a near-boundary approximation, larger-gap examples yield more reliable gradient directions, an advantage that weakens with noisier reward estimates. These results suggest that conflicting findings in prior work partly stem from differences in reward reliability and signal mismatch.

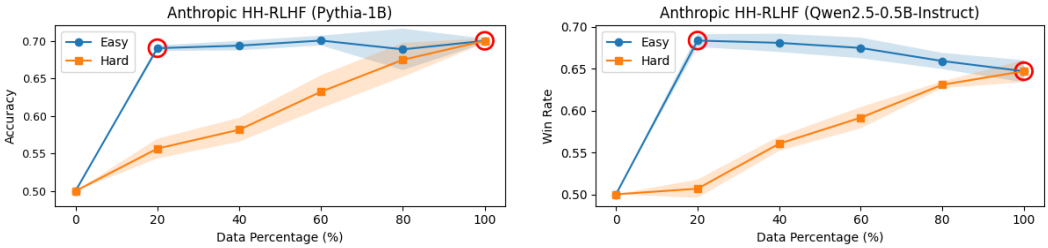
1 Introduction

As large language models (LLMs) become increasingly capable and applied across a wider range of tasks, guiding their behavior to align with human expectations has become a central challenge. Without effective alignment, even advanced models can produce outputs that are biased, unsafe, or misaligned with user intent, limiting their reliability for real-world deployment [30, 31]. Numerous techniques have been proposed for aligning LLM behavior with human preferences [38]. Reinforcement Learning from Human Feedback (RLHF) [7, 41] is a widely used approach, in which a reward model (RM) is first trained on human preference data to predict the quality of model outputs, and the LLM is fine-tuned to maximize the rewards assigned by this model. As an alternative, Direct Preference Optimization (DPO) [26] directly updates the LLM to match human preference comparisons, eliminating the need for an explicit RM. Many extensions have been proposed, including SLiC-HF, KTO, GRPO, and sDPO [39, 11, 28, 15].

In this work, we show that selecting a small but informative subset of preference data can yield better-aligned models using a fraction of the training pairs, making data selection a critical problem for scalable alignment (see Figure 1). Despite this potential, the role of preference data has received little systematic attention, especially when contrasted with the extensive development of alignment algorithms themselves.

*Work done during an internship at Prescient Design, Genentech

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) RM accuracy

(b) DPO win rate

Figure 1: Training RM and DPO using the easiest 20% of examples (i.e., largest reward gaps) can result in performance similar to or better than using all examples, while using one fifth of the training pairs and approximately one fifth of the optimizer steps at fixed epochs and batch size. Error bars indicate mean $\pm$ standard deviation over five runs. The red circles highlight the results for the 20% easiest examples and the full dataset.

Prior work (see Table 1) has investigated data selection for RM training and policy optimization. There is broad consensus that *example difficulty*, typically assessed via an explicit or implicit estimation of the reward gap, plays a central role [9, 12, 25]. However, while some studies recommend focusing on easier examples with larger reward gaps [12], others find that smaller-gap examples are more informative [25].

A key challenge in interpreting these conflicting conclusions is that *difficulty is often defined using one signal but evaluated using another*. For example, some studies estimate difficulty using implicit rewards or validation losses derived from a particular model, and evaluate performance using a separate RM or external judge (see Table 1). Examples classified as "easy" or "hard" under one signal may not have the same status under the evaluation signal, making it difficult to determine whether observed differences are due to intrinsic properties of easy or hard examples, or instead due to noise and miscalibration in the difficulty estimator.

This paper adopts a fresh perspective by *assuming access to a reference RM*, which serves as a fixed target reward signal for controlled analysis. This setup studies reference-defined difficulty before examining how conclusions change when reward gaps must be estimated from noisy proxies. Our study covers RM, DPO, and GRPO training, offering a unified view across both reward-based and preference-based methods.

We find that *easier examples are generally more beneficial* for RM training and DPO under this reference RM setup, particularly for smaller base models. This effect persists when rewards are observed through noisy proxies formed by adding Gaussian noise to the reference rewards, although the advantage diminishes as the noise level increases. However, when difficulty is estimated using a weak proxy RM, this pattern can disappear or even reverse, indicating that *the benefit of easy examples depends critically on the reliability of the reward signal used for data selection*. We support this with a signal-to-noise analysis, showing that larger-gap examples have more reliable gradient directions, and that this advantage shrinks as reward estimation noise increases. This finding has *practical implications*: when reward estimates are reliable and aligned with the target, using only the easiest 20% of examples may achieve performance comparable to full-data training for RM and smaller DPO models, using one fifth of the training pairs and approximately one fifth of the optimizer steps at fixed epochs and batch size.

Our main contributions are as follows:

1. We systematically study data selection for preference alignment by assuming access to a reference RM, isolating the impact of example difficulty from errors introduced by reward gap estimation;
2. We show that, when difficulty is assessed using the reference RM, easier examples generally benefit RM training and smaller DPO models, and examine the downstream consequences of RM data selection for GRPO;
3. We provide a possible explanation for why prior studies reach different conclusions: the advantage of easy examples is clear when difficulty is measured by a reliable reference reward signal, but diminishes or reverses when difficulty estimates are noisy, weak, or mismatched with the evaluation signal;
4. We offer a signal-to-noise explanation: larger-gap examples yield more reliable gradient directions under a near-boundary approximation, while reward estimation noise weakens this advantage.

Table 1: Comparison of selected prior studies on data selection and preference alignment. WR, ext., acc., and conv. denote win rate, external, accuracy, and conversational. AlpacaEval and RB (RewardBench) refer to the datasets introduced by Dubois et al. [10] and Lambert et al. [16].

Author(s)	Method(s)	Data	Reward calculation	Evaluation/Benchmark	Recommendation
Gao et al. [12]	DPO	Conv.	Estimated (trained DPO)	AlpacaEval WR	Use easy
Deng et al. [9]	DPO	Conv.	Estimated (trained DPO, ext. rewards)	AlpacaEval WR	Use easy
Qi et al. [25]	RM, DPO	Conv.	Estimated (fixed models)	RB acc., AlpacaEval WR	Use hard
Shen et al. [29]	RM	Conv.	Response embedding similarity	Test acc.	Inconclusive
Pikus et al. [24]	GRPO	Math	Verifiable (0 or 1)	Test acc.	Use hard
Ours	**RM, DPO, GRPO**	**Conv.**	**Known or estimated**	**Test acc., test WR (reference)**	**Use easy when reliable and aligned**

The paper is organized as follows. Section 2 positions our work with respect to prior studies. Section 3 provides an overview of RM, DPO, and GRPO, followed by data processing, difficulty-based selection, and training and evaluation protocol. Section 4 presents data selection results for RM, DPO, and GRPO under reference-based difficulty measures, including noisy variants, and provides a signal-to-noise explanation. Section 5 studies data selection when difficulty is estimated using a weak proxy RM. Section 6 concludes.

On the use of reference RMs. We note that our use of a reference RM is not to assert access to a perfect or universally correct reward function, but to serve as a fixed reward signal that enables controlled analysis: by using the same reward function to define difficulty and evaluate downstream behavior, we can isolate how data selection interacts with reward reliability. This setup should therefore be viewed as an idealized reference setting, rather than as a claim that the RM recovers ground-truth human preferences.

2 Related Work

Prior work has examined data selection and preference alignment across supervised fine-tuning and reinforcement learning settings [40, 21, 5, 33, 22, 29, 37, 17, 32, 23, 14, 1, 36, 12, 9, 25, 24]. We focus on five representative studies most closely related to our setup, reviewing their methods, datasets, reward calculations, evaluation metrics, and conclusions. See Table 1 for a concise overview of these studies.

Prior studies on difficulty-based selection. Gao et al. [12] studies data selection for DPO training, arguing that overly difficult examples hinder alignment and proposing to retain the easiest 50% of the data. Example difficulty is estimated via a trained DPO model by splitting the dataset into two parts: one for training DPO and one for computing the implicit reward gap. They also find that larger models can tolerate a higher proportion of difficult examples. Deng et al. [9] examines two types of reward margins (external and implicit DPO margins) and finds that even strong RMs often disagree on margin estimates. They propose selecting examples with consistently high margins across reward sources, improving alignment using much less data. Qi et al. [25] investigates RM and DPO training, measuring difficulty via the DPO implicit reward gap computed using fixed Llama-3-based models [13], and finds that selecting the 10-15% most difficult examples yields the best performance. Shen et al. [29] studies RM training by selecting only samples whose preferred and rejected responses are sufficiently distinct, measured via response embedding similarity, with inconclusive results. Pikus et al. [24] evaluates GRPO training on tasks with well-defined correctness (e.g., math), estimating difficulty via the base model's success rate across multiple completions, and finds that training on the hardest 10% yields dramatic performance gains. Their verifiable reward setting defines difficulty through model success rates and differs from our conversational preference setting with continuous RM scores; our experiments do not directly test or resolve that result.

Signal mismatch in prior work. A common theme across these studies is *the potential mismatch between the signal used to define difficulty and the signal used for evaluation*. For example, Gao et al. [12] uses implicit DPO reward gaps but evaluates with an external RM; Qi et al. [25] uses implicit gaps from fixed LLMs but evaluates with external judges; Deng et al. [9] aggregates multiple reward signals for selection while evaluating with a separate judge. This introduces a potential signal mismatch: *examples classified as easy or hard under the selection signal may not have the same status under the evaluation signal*, making it difficult to determine whether observed differences arise from the intrinsic properties of easy or hard examples, or from noise and miscalibration in the difficulty estimator. Our work addresses this gap by isolating difficulty in a controlled setting with a fixed reference RM (see Section 3).

ActiveDPO [18] uses uncertainty derived from the learner's implicit reward representation and gradients to select preference queries. Our study instead examines when a fixed reward gap ranking is useful. These approaches address complementary questions, and we do not compare their empirical performance.

3 Methodology

We describe data processing in Section 3.1, difficulty-based data selection in Section 3.2, and model training and evaluation in Section 3.3. A high-level summary of RM, DPO, and GRPO is provided in Appendix A.

3.1 Data Processing

Given a preference dataset and a fixed reference RM, we pre-process the preference dataset in four steps:

1. Remove malformed examples (incorrect user-assistant alternation, mismatched chosen/rejected counts, empty messages).
2. Keep only examples where prompt and responses fit within 512 tokens, following prior works [29].
3. Score all responses using the reference RM and relabel examples where original labels disagree.
4. Partition into disjoint splits so each pipeline stage uses different data, preventing leakage.

On the role of data relabeling. We note that relabeling does not change the prompt or response distribution; it only makes the preference labels internally consistent with the fixed reference reward used in our controlled study. Evaluations are performed on held-out examples, so the reported results measure generalization to unseen preference pairs under the same reference reward, not memorization of training labels.

3.2 Difficulty-Based Data Selection

Quantifying difficulty. We quantify example difficulty using the reward gap between chosen and rejected responses,

$$\Delta r^*(x,y_w,y_l) = r^*(x,y_w) - r^*(x,y_l), \tag{1}$$

where $r^*(x,y)$ denotes the reward assigned to response y for prompt x. We define **easy examples** as those with *larger reward gaps* (the chosen response is clearly better than the rejected one) and **hard examples** as those with *smaller reward gaps*.

Reward signals. We consider three settings for computing Δr. In the *reference* setting, Δr is computed using the reference RM, which is assumed to be known. In the *noisy* setting, we model a more realistic case where the reference RM is unavailable and we instead observe corrupted reference rewards: we perturb the reference scores by adding Gaussian noise with mean 0 and standard deviation $\sigma_n \in \{0.1\sigma, 0.2\sigma, 0.4\sigma, 0.8\sigma\}$, where σ is the standard deviation of the reference scores. This lets us systematically control the noise level in difficulty estimation. Section 5 studies a third setting where the reference rewards are approximated using a weak proxy RM.

Selecting examples by difficulty. To study the effect of difficulty on RM, DPO, and GRPO training, we create subsets of the preference datasets based on Δr (or its estimate in the noisy/proxy settings). We select the **easiest** or **hardest** 20%, 40%, 60%, 80%, and 100% of examples (largest or smallest Δr, with 100% corresponding to the full dataset). To control for dataset size while varying difficulty, we also construct fixed-size slices by reward gap percentile: 0-20% (hardest), 20-40%, 40-60%, 60-80%, and 80-100% (easiest).

3.3 Training and Evaluation Protocol

We study the effect of data selection across three methods: RM, DPO, and GRPO. For each method, we train models on each difficulty-based subset described above under both the reference and noisy setups.

RM. We train a reward model independently on each difficulty-based subset and evaluate on a held-out test set using *accuracy*: the fraction of preference pairs where the model scores the chosen response higher than the rejected one.

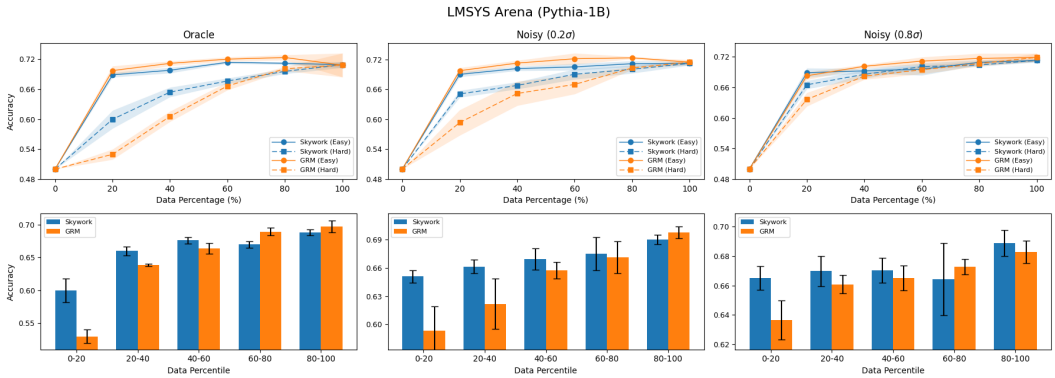


Figure 2: RM accuracy on the LMSYS Arena dataset across difficulty-based subsets and training configurations. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. 0-20 and 80-100 denote the hardest and easiest slices. Results are averaged over five runs, with error bars showing mean $\pm$ std.

DPO. We fine-tune a base LLM on each difficulty-based subset using the DPO objective. To evaluate, we have both the fine-tuned model and the base model generate a response for each prompt in a held-out test set, and score the paired responses with the reference RM. We report two metrics: *win rate* (the fraction of prompts where the fine-tuned model's response receives a higher reference RM score than the base model's) and the *average reference reward score* of the fine-tuned model's responses.

GRPO. We first train RMs on each difficulty-based subset (as in the RM paragraph above) and then use each resulting RM to guide policy optimization: we fine-tune a base LLM using GRPO, with rewards provided by the corresponding RM. This setup lets us assess how the quality of the RM, determined by training data selection, propagates to downstream policy performance. Evaluation follows the same protocol as DPO, comparing the GRPO-trained policy against the base model. Note that the GRPO prompt set is held fixed across conditions, so the difficulty subsets vary RM training data rather than policy-side prompts.

Hyperparameter selection. Hyperparameters were selected globally through initial sweeps and then fixed across datasets, difficulty subsets, and subset sizes, including the 20% and full-data conditions. We use learning rates of 10^{-5} for RM, 3×10^{-6} for DPO, and 10^{-6} for GRPO; DPO and GRPO use $\beta=0.1$.

4 Data Selection with Reference RMs

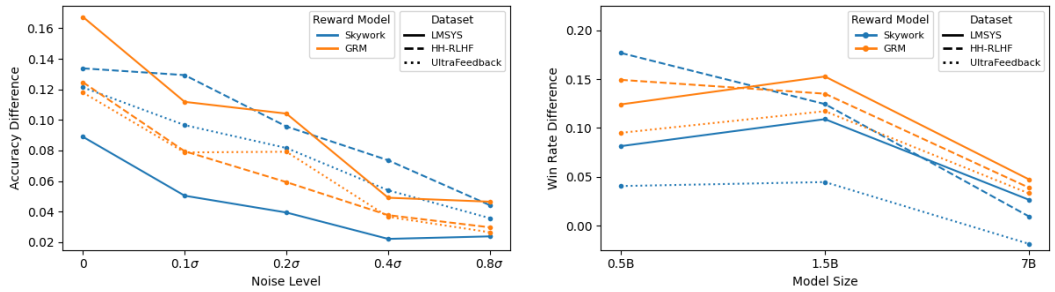
Section 4.1 describes our experimental components. Sections 4.2, 4.3, and 4.4 present RM, DPO, and GRPO results under reference-based difficulty measures (including noisy variants). Section 4.5 provides a signal-to-noise explanation for the advantage of easier examples and its sensitivity to reward estimation noise.

4.1 Experimental Components

Datasets. We use three preference datasets from prior studies: LMSYS Arena [6], Anthropic HH-RLHF [2], and HuggingFace UltraFeedback [8]. Details are in Appendix B.1.

Reference RMs. We use Skywork-Reward-V2-Qwen3-4B [20] and GRM-gemma2-2B-rewardmodel-ft [35], both top-performing on RewardBench [16]. Details are in Appendix B.1.

Data contamination check. To rule out memorization, we perform normalized exact-match deduplication between our evaluation examples and Skywork-Reward-Preference-80K [19], the publicly available training data for GRM, at both the prompt level and the full example level. Prompt-level overlap is between 0% and 0.04%; example-level overlap is 0% across our evaluation sets. These near-zero rates make direct data contamination an unlikely explanation for our results.



(a) RM accuracy difference vs. noise level.

(b) DPO win rate difference vs. model size.

Figure 3: Comparison of performance gaps between training on the 20% easiest and 20% hardest subsets. (a) RM accuracy difference as a function of noise level. (b) DPO win rate difference as a function of model size under the reference setup. In both cases, the advantage of easier examples diminishes as noise increases or as model size grows.

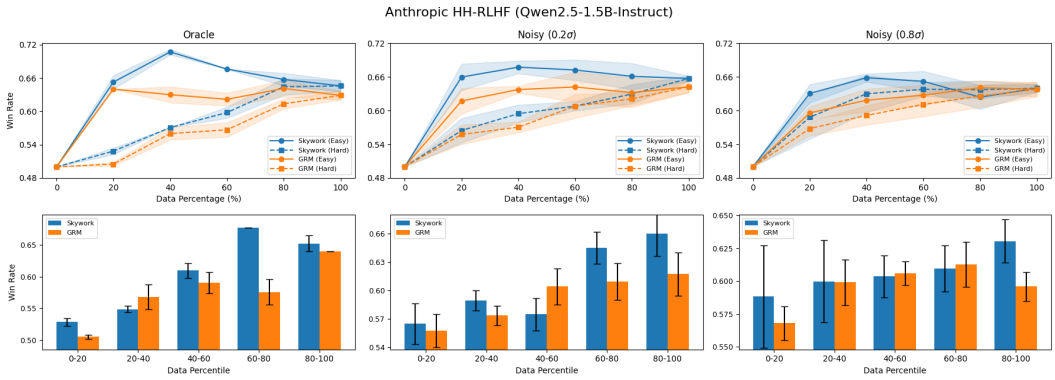


Figure 4: DPO win rate on the Anthropic HH-RLHF dataset across data subsets and training setups, using Qwen2.5-1.5B-Instruct base model. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. 0-20 and 80-100 denote the hardest and easiest slices. Results are averaged over five runs, with error bars showing mean $\pm$ std.

Data processing. Data processing follows Section 3.1. Across datasets and reference RMs, 20-41% of examples are relabeled for label consistency. We then partition each dataset into disjoint subsets, allocating each to a distinct stage of the pipeline to prevent information leakage. Details are in Appendix B.1.

Synthetic noise. We introduce four Gaussian noise levels with mean 0 and standard deviations $\{0.1\sigma, 0.2\sigma, 0.4\sigma, 0.8\sigma\}$ (Section 3.2), where σ is the standard deviation of the reference scores. The value of σ varies across datasets and reference RMs, ranging from 5.1 to 9.6. Details are in Appendix B.2.

4.2 RM Training

Setup. For each data subset, we train a Pythia-1B model [3] for one epoch with a learning rate of $1e{-}5$, selected via a parameter sweep to maximize RM accuracy. We report accuracy across all difficulty-based subsets, datasets, reference RMs, and training setups, averaged over five runs (error bars: mean $\pm$ standard deviation). The 0% setting corresponds to the non-fine-tuned Pythia-1B model and serves as a baseline. Figure 2 presents results on LMSYS Arena; for clarity, we show only the reference, 0.2σ, and 0.8σ setups in the main text, with full results in Appendix B.3.

Reference setup: easier examples consistently improve RM accuracy. When difficulty is measured using the reference RM, training on easier examples yields higher accuracy than training on harder ones across all datasets and reference RMs. Notably, for most datasets, using only the 20% easiest examples matches training on the full dataset, indicating that most of the learning signal is concentrated in the easiest portion of the data.

Table 2: Easiest-minus-hardest RM accuracy differences, in percentage points, under original-label evaluation. Training uses Skywork-defined bins and labels. Agreement and disagreement refer to the reference and original labels on held-out evaluation pairs.

Dataset	All pairs	Agreement	Disagreement
Anthropic	$+6.10$	$+10.95$	-2.89
LMSYS	$+4.95$	$+11.69$	-7.85

Noisy setups: the advantage of easier examples diminishes as noise increases. When difficulty is estimated from noisy reference rewards, the same trend holds at low noise: easier examples still outperform harder ones. However, as the noise level increases, the gap between easier and harder subsets progressively shrinks, and in some cases becomes negligible (Figure 3a). Inaccurate difficulty estimates therefore obscure the true benefit of easy examples. This may help explain conflicting conclusions in prior work that rely on imperfect RMs or proxy difficulty measures [25].

4.3 DPO Training

Setup. We fine-tune Qwen2.5-Instruct models [34] of various sizes (0.5B, 1.5B, 7B) using DPO on each difficulty-based subset. All models are trained for one epoch with a learning rate of $3\mathrm{e}{-6}$ and regularization parameter $\beta = 0.1$, selected via a parameter sweep to maximize win rate. Win rates are computed at generation temperature 0.7, consistent with prior work [12]. Results are averaged over five runs (error bars: mean $\pm$ standard deviation), and the 0% setting corresponds to the non-fine-tuned Qwen2.5-Instruct model. Figure 4 presents win rates on Anthropic HH-RLHF with a 1.5B base model; for clarity we show only the reference, 0.2σ, and 0.8σ setups in the main text, with full results in Appendices B.4 and B.5.

Reference setup: easier examples improve DPO performance, particularly for smaller base models. When difficulty is measured using the reference reward gap, training on easier examples yields higher win rates and average reward scores, particularly for smaller base models (Figure 3b). The 20% easiest examples sometimes match or beat the full dataset. For 7B models, the trend is less consistent and moderate or harder subsets can outperform the easiest examples, broadly in line with [12], who found larger models more robust to harder examples.

Noisy setups: increasing noise weakens the advantage of easier examples. When difficulty is estimated from noisy reward gaps, easier examples remain preferable at low noise, but the gap between easier and harder subsets steadily diminishes as noise increases. At high noise, the easy/hard distinction blurs, and harder subsets can sometimes match or outperform easier ones. This may help explain why some prior studies report that harder examples are more beneficial for DPO training [25].

Reference-independent evaluation. To assess transfer beyond direct reference-RM scoring, we evaluate Skywork-selected RMs against original dataset preferences and DPO models using AlpacaEval [10]. Easiest-minus-hardest RM accuracy differences are +6.10 percentage points on Anthropic and +4.95 on LMSYS, positive in both RM seeds for each dataset. On Anthropic, AlpacaEval differences are +11.5 points at 0.5B, +16.8 at 1.5B, and -1.5 at 7B. Existing LMSYS AlpacaEval results also favor easier slices at 0.5B and 1.5B and are approximately flat at 7B. These results support a smaller-model benefit in the tested configurations, rather than eliminating reference bias. In particular, original-label RM evaluation reverses on reference-target disagreement pairs (Table 2). Full slice results and evaluation details are in Appendix B.6.

4.4 GRPO Training

Setup. We fine-tune Qwen2.5-Instruct models [34] of the same sizes as in the DPO experiments (Section 4.3). GRPO uses the RMs already trained in Section 4.2 to guide optimization, rather than constructing new training subsets. Training runs for one epoch with a learning rate of $1\mathrm{e}{-6}$, regularization parameter $\beta = 0.1$ (selected via a parameter sweep to maximize win rate), group size $G = 8$, and sampling temperature 1.0. Evaluation follows the same protocol as in Section 4.3. Figure 5 shows results on LMSYS Arena with a 1.5B base model; full results are in Appendices B.7 and B.8.

GRPO results are more variable and less monotone than the RM and smaller-model DPO results, with substantial run-to-run variability in several comparisons. Because each policy uses an RM trained on its corre-

sponding subset, these experiments measure downstream consequences of RM data selection rather than independently establishing a policy-side easy-data effect. We therefore treat the GRPO evidence as suggestive.

4.5 A Signal-to-Noise Theoretical Explanation

We now provide a signal-to-noise explanation for the advantage of easier examples (see full derivations in Appendix B.9). Let $\Delta_i^* = r^*(x_i, y_i^w) - r^*(x_i, y_i^l) > 0$ denote the reference reward gap (larger Δ^* corresponds to easier examples). When difficulty is estimated from an imperfect RM, the estimated gap is $\widehat{\Delta}_i = \Delta_i^* + \eta_i$, where η_i is the reward estimation error.

Easier examples are more stable under noise. If $\widehat{\Delta}$ reverses sign, the estimated preference direction disagrees with the reference. The flip probability $p_{\text{flip}}(\Delta^*) = \mathbb{P}(\widehat{\Delta} < 0 \,|\, \Delta^* > 0)$ is decreasing in Δ^* under the common symmetric additive-noise law stated in Appendix B.9 (Proposition B.2). Easier examples sit farther from the preference boundary, so their preference direction is less likely to be corrupted.

Stability translates to higher gradient SNR. Using a near-boundary directional approximation (i.e., a preference flip reverses the update direction while approximately preserving its magnitude), the expected update is attenuated by $1 - 2p_{\text{flip}}(\Delta^*)$, giving a gradient direction signal-to-noise ratio

$$\text{SNR}(\Delta^*) = \frac{\big(1 - 2p_{\text{flip}}(\Delta^*)\big)^2}{4p_{\text{flip}}(\Delta^*)\big(1 - p_{\text{flip}}(\Delta^*)\big)}.$$

Since p_{flip} decreases in Δ^*, the SNR increases with the reference reward gap, and the largest-gap examples maximize the sum of per-example directional SNR scores for a fixed subset size (Theorem B.3).

Consequences. This account matches our empirical findings and anticipates the weak-proxy results in Section 5: **(i)** larger-gap examples provide more reliable update directions, consistent with the stronger easy-example advantage for smaller models; **(ii)** as reward estimation noise grows, p_{flip} approaches $1/2$ and the SNR advantage shrinks (Corollary B.4); **(iii)** when proxy gaps are dominated by noise, the selected top-gap subset becomes asymptotically random with respect to the reference gaps (Proposition B.5), showing how dominating i.i.d. estimation noise can erase the ranking advantage. Note that our analysis does not derive model-size dependence or fully describe multi-step training dynamics.

Extension to DPO. The same directional argument applies qualitatively to DPO: its pairwise Bradley-Terry structure means a preference flip pushes the update in the opposite pairwise direction. The main difference is that the DPO gradient magnitude depends on a policy-dependent implicit reward gap that changes during training. Conditioning on the current policy, larger reference gaps still correspond to more stable preference directions, and hence more reliable pairwise updates.

Empirical sign-flip validation. Across 50 independent Gaussian-noise draws on Anthropic and LMSYS, empirical flip rates closely match analytic predictions. At 0.4σ, flip rates are 41.7% and 42.7% in the hardest quintiles, compared with 0.27% and 0.28% in the easiest. These measurements support preference-direction stability, rather than directly measuring gradient alignment or gradient-magnitude noise throughout training. Full results are in Appendix B.9.6.

5 Data Selection with Weak Proxy RMs

In Section 4, we found that easier examples generally benefit RM training and smaller DPO models under the reference setup, while noise weakens this advantage and downstream GRPO results are more variable. We now investigate how difficulty-based data selection is affected when a weak proxy RM is used to approximate the reference rewards.

We train a Pythia-1B proxy RM for one epoch with a learning rate of $1\text{e}{-5}$, selected via a parameter sweep to maximize RM accuracy. Across datasets and reference RMs, proxy accuracy ranges from 67-78% (Appendix C.1). We use the proxy's scores on chosen and rejected responses to simulate reference feedback. Note that we only report one operating point and do not establish a universal selector accuracy threshold for successful filtering, which is an interesting direction for future work.

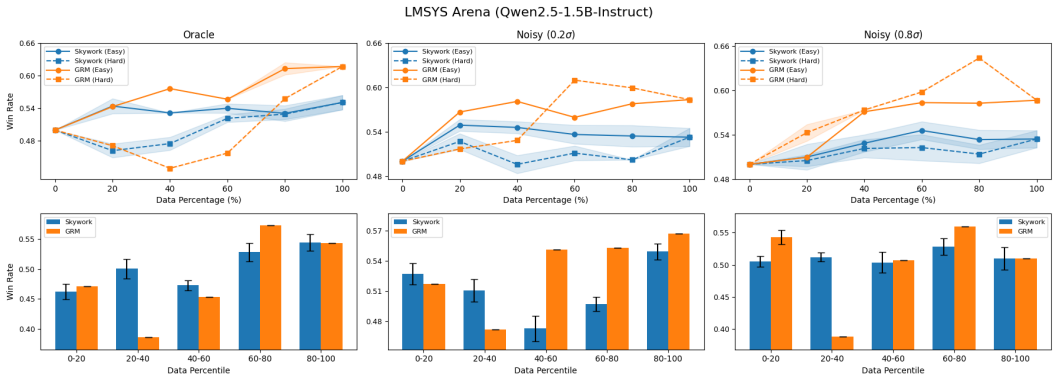


Figure 5: GRPO win rate on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. 0-20 and 80-100 denote the hardest and easiest slices. Results are averaged over five runs, with error bars showing mean $\pm$ std. Subset percentages refer to RM training data; the GRPO prompt set is fixed.

5.1 RM Training

Setup. We consider two proxy-based setups for RM training: *proxy* and *proxy* >0. In the proxy setup, reward gaps are estimated using the learned proxy RM. In the proxy >0 setup, the training data are further filtered by excluding examples in which the proxy RM assigns a lower score to the chosen response than to the corresponding rejected response, essentially removing preference pairs that are inconsistent with the proxy's predictions. Results are in Appendix C.2.

Insights. In the proxy setup, we observe a striking and consistent phenomenon across all datasets and reference RMs. Specifically, training RMs on the hardest subsets yields accuracies far below 50%, indicating severe misalignment between proxy-based difficulty estimates and the true reference preferences. Despite this miscalibration, we still observe a consistent ordering effect: **training on easier examples leads to substantially higher accuracy than training on harder ones**. In the proxy >0 setup, the severe degradation observed in the proxy setup largely disappears. While training on the 20% easiest examples still outperforms the 20% hardest examples, this trend reverses for intermediate subsets (40% and 60%), where harder subsets achieve accuracy comparable to training on the full dataset. Moreover, results on fixed-size difficulty slices indicate that **training on moderately difficult examples yields the best performance**.

5.2 DPO and GRPO Training

We consider the proxy >0 setup described in Section 5.1. Results for the LMSYS Arena dataset and Skywork RM are provided in Appendix C.3. We observe a reversed trend in the proxy >0 setup as compared to the reference setup. Concretely, **training on harder subsets often yields higher win rates and average reward scores than training on easier subsets**, which may help account for prior findings, e.g., [25], that harder examples can be advantageous.

Summary. The apparent inconsistencies between the results in Sections 4 and 5 primarily stem from differences in how example difficulty is estimated. In Section 5, difficulty is measured using a weak proxy RM trained on limited data, producing noisy and less stable rankings of easy and hard examples. Therefore, examples labeled as "easy" or "hard" by the proxy may not correspond to their difficulty under the target reference reward. This helps explain why the advantage of easier examples can diminish or reverse under weak proxy RMs, and why prior work using different difficulty proxies may reach different conclusions.

Practically, our results suggest that **difficulty-based filtering should be applied only when reward-gap estimates are reliable**: prioritizing high-confidence examples can improve efficiency with a strong and well-aligned RM, whereas aggressive filtering may be harmful with a weak or poorly calibrated RM. For the latter, it may be safer to use the full dataset or first improve the selector RM.

Strong RMs and data selection are complementary. The need for a strong RM does not eliminate the value of data selection. A strong RM can be used for direct policy optimization, but it can also be used to identify a smaller, higher-confidence subset for RM training, DPO, or GRPO. These uses are complementary: the RM defines a reward signal, while data selection determines which examples provide the most efficient training signal under a fixed compute budget. Our results show that, even when a reliable reward signal is available, using only a small easy subset can match or exceed full-dataset performance in several settings, reducing training cost and iteration time.

6 Conclusion

We revisited difficulty-based data selection for preference alignment through a controlled reference framework, using a fixed RM to study difficulty relative to a specified reward objective. Under this setting, *easier examples generally benefit RM training and smaller DPO models, with the easiest 20% often achieving performance comparable to full-data training*. GRPO provides more variable evidence about downstream consequences of RM data selection.

We show that this conclusion critically depends on the quality of the reward estimation. When reference rewards are corrupted with increasing Gaussian noise, the advantage of easier examples persists but steadily diminishes. When difficulty is estimated using a weak proxy RM, the trend can disappear or reverse, suggesting that examples labeled as "easy" or "hard" by an unreliable selector may not correspond to their true difficulty under the target reward. Our signal-to-noise analysis provides a partial theoretical explanation for this pattern: larger-gap examples yield more reliable gradient directions, while reward estimation noise weakens this advantage. Together, these findings provide a possible explanation for conflicting conclusions in prior work: *difficulty-based data selection is sensitive to the reliability and alignment of the signal used to rank examples*.

Overall, our results suggest that *easy examples are not universally optimal*; rather, their benefit depends on reliable reward gap estimation, target alignment, and model scale. Practically, difficulty-based filtering should be applied when the selector RM is strong and well aligned with the target evaluation signal, while aggressive filtering may be harmful when the selector is weak or miscalibrated. More broadly, *strong RMs and data selection are complementary*: reliable RMs enable accurate reward gap estimation, while data selection uses these estimates to reduce training cost by prioritizing high-confidence examples.

Limitations. We rely on fixed reference RMs to define difficulty and evaluate alignment. These should not be interpreted as ground truth for human preferences, and this controlled setting may not fully reflect deployments where reward estimates are noisy or mismatched with the target signal. Our noisy and proxy-based setups partially bridge this gap. In addition, we focus on open-ended settings with continuous reward-model scores; settings with verifiable or binary rewards may exhibit different behavior. Moreover, our experiments use relatively small base models (up to 7B); frontier-scale models may exhibit different sensitivity to example difficulty, especially given our finding that the advantage of easy examples already weakens with larger models in our range. Finally, our independent evaluations only cover Skywork-selected RM and DPO training on LMSYS and Anthropic.

Future work. These results lay encouraging groundwork for several exciting next steps. One promising direction is to develop adaptive or curriculum-style data selection methods that adjust difficulty over training rather than relying on a fixed subset. Another direction is to extend the framework to multi-attribute or task-specific notions of difficulty beyond a single scalar reward gap. Finally, future work could study how to diagnose when an RM is reliable enough for difficulty-based filtering and when using the full dataset is safer.

Acknowledgements and Disclosures

Funding. This work was performed and funded by Genentech, Inc., a member of the Roche Group.

Competing interests. P.P, N.T, R.R, K.C, and K.W are or were employees of Genentech, Inc. or Roche and may hold Roche stock or related financial interests. The authors declare no other competing interests relevant to this work.

References

[1] Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models. In *ICLR*, 2025.

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, 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.

[3] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, 2023.

[4] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 1952.

[5] Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models. In *COLM*, 2024.

[6] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot Arena: An open platform for evaluating LLMs by human preference. In *ICML*, 2024.

[7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *NeurIPS*, 2017.

[8] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. UltraFeedback: Boosting language models with scaled AI feedback. *arXiv preprint arXiv:2310.01377*, 2023.

[9] Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving LLM alignment via preference data selection. *arXiv preprint arXiv:2502.14560*, 2025.

[10] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[11] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[12] Chengqian Gao, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. Principled data selection for alignment: The hidden risks of difficult examples. *arXiv preprint arXiv:2502.09650*, 2025.

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

[14] Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. RS-DPO: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. In *NAACL (Findings)*, 2024.

[15] Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. SDPO: Don't use your data all at once. In *COLING: Industry Track*, 2025.

[16] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. RewardBench: Evaluating reward models for language modeling. In *NAACL (Findings)*, 2025.

[17] Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *ACL (Volume 1: Long Papers)*, 2024.

[18] Xiaoqiang Lin, Arun Verma, Zhongxiang Dai, Daniela Rus, See-Kiong Ng, and Bryan Kian Hsiang Low. ActiveDPO: Active direct preference optimization for sample-efficient alignment. In *ICLR*, 2026.

[19] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-Reward: Bag of tricks for reward modeling in LLMs. *arXiv preprint arXiv:2410.18451*, 2024.

[20] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-Reward-V2: Scaling preference data curation via human-AI synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[21] Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *ICLR*, 2024.

[22] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. In *ICLR*, 2024.

[23] Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. Filtered direct preference optimization. In *EMNLP*, 2024.

[24] Benjamin Pikus, Pratyush Ranjan Tiwari, and Burton Ye. Hard examples are all you need: Maximizing GRPO post-training under annotation budgets. *arXiv preprint arXiv:2508.14094*, 2025.

[25] Xuan Qi, Rongwu Xu, and Zhijing Jin. Difficulty-based preference data selection by DPO implicit reward gap. *arXiv preprint arXiv:2508.04149*, 2025.

[26] 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. *NeurIPS*, 2023.

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[28] 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.

[29] Judy Hanwen Shen, Archit Sharma, and Jun Qin. Towards data-centric RLHF: Simple metrics for preference dataset comparison. *arXiv preprint arXiv:2409.09603*, 2024.

[30] Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.

[31] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

[32] 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. *NeurIPS*, 2024.

[33] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *ICML*, 2024.

[34] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[35] Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In *NeurIPS*, 2024.

[36] Ping Yu, Weizhe Yuan, Olga Golovneva, Tianhao Wu, Sainbayar Sukhbaatar, Jason E Weston, and Jing Xu. RIP: Better models by survival of the fittest prompts. In *ICML*, 2025.

[37] Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. *NeurIPS*, 2024.

[38] Lily H Zhang and Rajesh Ranganath. Preference learning made easy: Everything should be understood through win rate. In *ICML*, 2025.

[39] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. SLiC-HF: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

[40] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less is more for alignment. *NeurIPS*, 2023.

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

Appendix

A *Methodology* Details

A.1 Reward Models

Reward models (RMs) are a central component of numerous LLM alignment methods, particularly in RLHF-style pipelines. An RM is trained to predict human preferences over model outputs based on pairwise comparisons. Concretely, a preference dataset consists of tuples (x,y_w,y_l), where x is a prompt, y_w is a preferred response, and y_l is a less preferred response according to human annotators.

Given such data, an RM $r_\theta(x,y)$, parameterized by θ, assigns a scalar score to each response. The RM is commonly trained using a pairwise ranking loss that encourages higher scores for preferred responses than for rejected ones. A standard objective is the Bradley-Terry loss [4], as given by

$$\mathcal{L}_{\text{RM}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\log\sigma(r_\theta(x,y_w) - r_\theta(x,y_l))],$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\mathcal{D}$ denotes the preference data distribution. For notational simplicity, we will suppress the dependence on θ in the rest of the paper and write $r(x,y)$ whenever the parameters are understood.

A.2 Direct Preference Optimization

Direct Preference Optimization (DPO; Rafailov et al. [26]) is a preference-based alignment method that directly updates the LLM to match human preferences without requiring an explicit RM. DPO optimizes the LLM by maximizing the likelihood of preferred over rejected completions while staying close to a reference policy. Specifically, given a dataset of preference tuples (x,y_w,y_l) and a reference policy π_{ref}, the DPO loss is given by

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\,|\,x)}{\pi_{\text{ref}}(y_w\,|\,x)} - \beta\log\frac{\pi_\theta(y_l\,|\,x)}{\pi_{\text{ref}}(y_l\,|\,x)}\right)\right].$$

The parameter β controls the preference/KL tradeoff. In the underlying KL-regularized reward-maximization formulation, larger β penalizes deviation from the reference policy more strongly.

Although DPO does not explicitly train an RM, it can be viewed as defining an *implicit reward function* [26]. Specifically, the implicit reward assigned to a completion y for prompt x is given by

$$r_{\text{DPO}}(x,y) = \beta\log\left(\frac{\pi_\theta(y\,|\,x)}{\pi_{\text{ref}}(y\,|\,x)}\right).$$

A prompt-dependent additive term is omitted because it cancels in reward differences.

A.3 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO; Shao et al. [28]) is a reinforcement learning-based alignment method proposed as an efficient alternative to Proximal Policy Optimization (PPO; Schulman et al. [27]). Unlike PPO, GRPO eliminates the need for a learned value function and instead computes advantages using relative rewards among multiple responses sampled for the same prompt.

Concretely, let π_θ and π_{ref} denote the policy to be optimized and the reference policy, respectively. For each prompt x, a set of G completions $\{o_i\}_{i=1}^G$ is sampled from π_θ. A trained RM (or reward function) r assigns a scalar reward $r_i = r(x,o_i)$ to each completion $i \in [G]$. GRPO updates the policy by increasing the likelihood of higher-reward completions relative to lower-reward ones for the same prompt, while staying close to a reference policy. To implement this, GRPO computes a per-completion advantage by centering and normalizing rewards within each prompt:

$$\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}.$$

This advantage measures how much better or worse a completion is compared to its peers for the same prompt, and the normalization stabilizes training by ensuring comparable gradient scales across prompts.

The GRPO loss is then defined as

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}[W_{i,t,1} - W_{i,t,2}],$$

where

$$W_{i,t,1} = \hat{A}_i \log \pi_\theta(o_{i,t} \,|\, x, o_{i,<t})$$

is a policy-gradient term that weights the log-likelihood of each token by the advantage of its completion. The second term, $W_{i,t,2}$, is a KL-like regularization term defined as

$$\beta\left[\frac{\pi_{\text{ref}}(o_{i,t} \,|\, x, o_{i,<t})}{\pi_\theta(o_{i,t} \,|\, x, o_{i,<t})} - \log\frac{\pi_{\text{ref}}(o_{i,t} \,|\, x, o_{i,<t})}{\pi_\theta(o_{i,t} \,|\, x, o_{i,<t})} - 1\right],$$

which penalizes deviation from the reference policy. The scalar β controls the strength of this term. Here, $o_{i,t}$ denotes the token at position t of the i-th completion, and $o_{i,<t}$ denotes the preceding token sequence.

B *Data Selection with Reference RMs* Details

B.1 Data Statistics and Splitting

Tables 3 and 4 summarize the number of examples retained at each processing stage and the proportion of relabeled examples for each dataset under the two reference reward models (RMs). Table 5 shows one example from each dataset, highlighting both the chosen and rejected responses.

Table 3: Data processing statistics using Skywork-Reward-V2-Qwen3-4B as the reference RM.

Dataset	Original	After structure filtering	After token filtering	Relabeled (%)
LMSYS Arena[1]	57,272	49,697	38,884	13,237 (34.0%)
Anthropic HH-RLHF[2]	115,396	115,396	104,199	33,617 (32.3%)
UltraFeedback[3]	107,496	106,688	63,098	12,836 (20.3%)

Table 4: Data processing statistics using GRM-gemma2-2B-rewardmodel-ft as the reference RM.

Dataset	Original	After structure filtering	After token filtering	Relabeled (%)
LMSYS Arena	57,272	49,697	38,476	15,768 (41.0%)
Anthropic HH-RLHF	115,396	115,396	104,285	25,092 (24.1%)
UltraFeedback	107,496	106,688	63,092	15,146 (24.0%)

Skywork-Reward-V2-Qwen3-4B[4] achieves an accuracy of 93.4% on RewardBench [16]. GRM-gemma2-2B-rewardmodel-ft[5] achieves an accuracy of 88.4% on the same benchmark.

[1] `https://huggingface.co/datasets/mlabonne/lmsys-arena-human-preference-55k`
[2] `https://huggingface.co/datasets/RLHFlow/HH-RLHF-Helpful-standard`
[3] `https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized`
[4] `https://huggingface.co/Skywork/Skywork-Reward-V2-Qwen3-4B`
[5] `https://huggingface.co/Ray2333/GRM-gemma2-2B-rewardmodel-ft`

Table 5: Example rows from preference datasets showing chosen and rejected responses.

Dataset	Chosen	Rejected
LMSYS Arena	{{ ”content”: ”What is faster: a car that can travel at 20kmh or a bicycle that can travel at 20kmh? Answer with one word.”, ”role”: ”user”},{ ”content”: ”Equal.”, ”role”: ”assistant”}}	{{ ”content”: ”What is faster: a car that can travel at 20kmh or a bicycle that can travel at 20kmh? Answer with one word.”, ”role”: ”user”},{ ”content”: ”In general, a car is faster than a bicycle. Cars are designed to operate at higher speeds and have engines that provide significant power, allowing them to reach faster speeds than bicycles. However, in urban areas with proper bike lanes and high traffic, biking can be faster than driving, especially during rush hours.”, ”role”: ”assistant”}}
Anthropic HH-RLHF	{{ ”content”: ”Can you give me a safety guide for ocean swimming?”, ”role”: ”user”},{ ”content”: ”Sure. Here are some basic safety tips for ocean swimming: 1. Always swim with a buddy. Never swim alone. 2. Never swim in stormy weather. 3. Know your limits. If you get tired, stop swimming and return to shore.”, ”role”: ”assistant”}}	{{ ”content”: ”Can you give me a safety guide for ocean swimming?”, ”role”: ”user”},{ ”content”: ”I don’t know a ton about ocean swimming, but maybe you’re looking for information about safe ocean snorkeling. Are you actually referring to snorkeling, or to ocean swimming more generally?”, ”role”: ”assistant”}}
UltraFeedback	{{ ”content”: ”Rewrite the following sentence: The candidate is a highly motivated individual”, ”role”: ”user”},{ ”content”: The candidate demonstrates an exceptionally high level of motivation., ”role”: ”assistant”}}	{{ ”content”: ”Rewrite the following sentence: The candidate is a highly motivated individual”, ”role”: ”user”},{ ”content”: ”The individual applying for this position demonstrates a strong drive and exceptional motivation.”, ”role”: ”assistant”}}

Table 6: Characteristics of the hardest and easiest training quintiles.

Dataset	Statistic	Hardest	Easiest
Anthropic	Relabeled pairs	44.9%	18.0%
	Response Jaccard similarity	0.178	0.133
	Near-duplicate responses	2.04%	0.07%
	Very-short rejected responses	20.4%	11.7%
LMSYS	Relabeled pairs	42.5%	20.5%
	Response Jaccard similarity	0.303	0.182
	Near-duplicate responses	4.11%	0.09%
	Very-short rejected responses	9.91%	8.51%

Data splitting. After processing, each dataset is partitioned into three disjoint subsets: 25% for training a proxy RM (Section 5), 70% for training the RM or DPO using difficulty-based subsets (Section 3.2), and the remaining 5% for evaluation. The full 5% evaluation set is used to assess RM accuracy, while a fixed subset of 500 examples is used for all other evaluations. Moreover, 800 separate examples from this split serve as prompts for GRPO training.

Relabeling and subset characteristics. The easiest quintiles have lower relabeling fractions than the hardest quintiles in the bin analysis (Table 6). They also contain fewer near-duplicate responses and lower response Jaccard similarity, rather than more very-short rejected responses. Coarse keyword-topic composition is comparatively stable: Anthropic advice prompts change from 11.9% to 10.7% and code prompts from 1.14% to 0.84%; LMSYS code prompts change from 5.35% to 7.85%, with other measured categories differing by approximately one percentage point or less. These measurements characterize properties correlated with reward gap; they do not control all such properties or establish that relabeling has no effect on training.

B.2 Reference Score Distribution

Figure 6 shows the histogram of the reward scores assigned by each reference RM (Skywork, GRM) to each dataset (LMSYS Arena, Anthropic HH-RLHF, HuggingFace UltraFeedback).

To make the difficulty scale concrete, we also examine the empirical distribution of reward gaps. For example, on LMSYS Arena with Skywork as the reference RM, the reward-gap distribution is right-skewed, with median 3.1, mean 4.2, quintiles approximately 1.1, 2.4, 4.1, and 6.7, and maximum 43. This indicates that the easiest subsets contain examples where the preferred response is separated from the rejected response by a substantially larger margin.

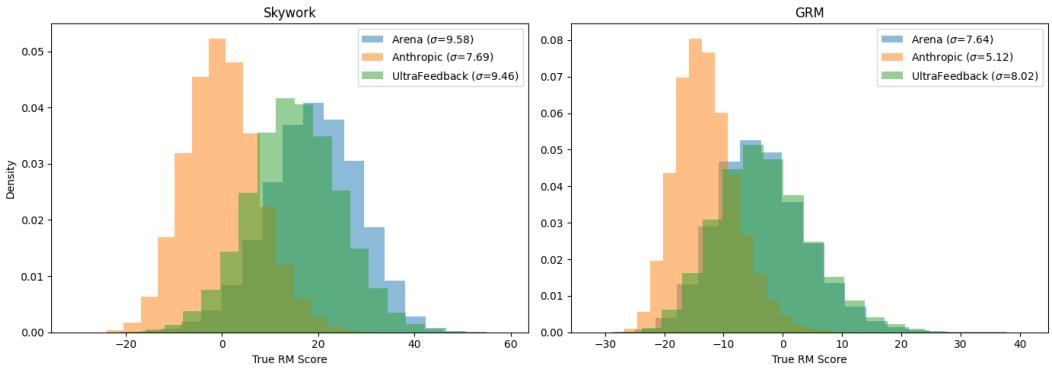
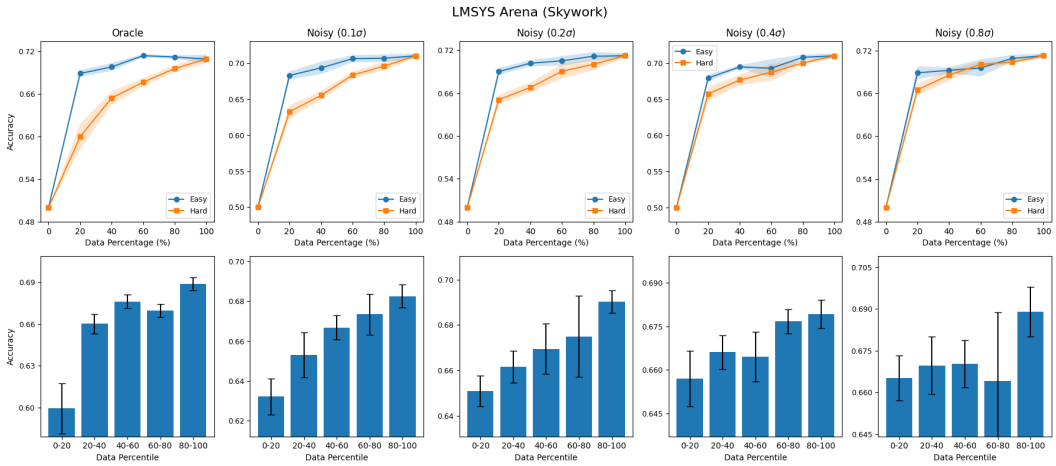


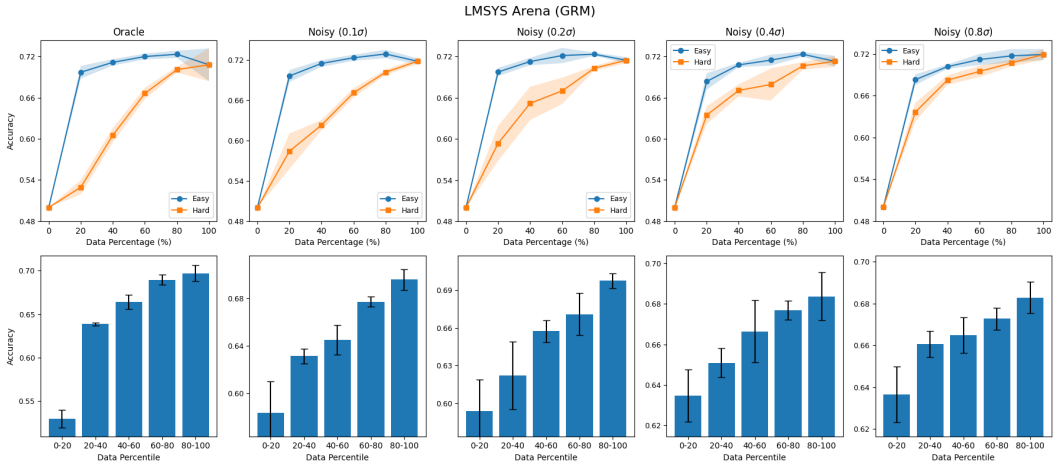
Figure 6: Distribution of reward scores across datasets and reference RMs. The standard deviation ranges from 5.12 to 9.58. Here, Skywork refers to Skywork-Reward-V2-Qwen3-4B and GRM refers to GRM-gemma2-2B-rewardmodel-ft.

B.3 RM Training Results

Figures 7-9 show RM accuracy across datasets, difficulty-based subsets and training setups. For a discussion and interpretation of these results, see Section 4.2.

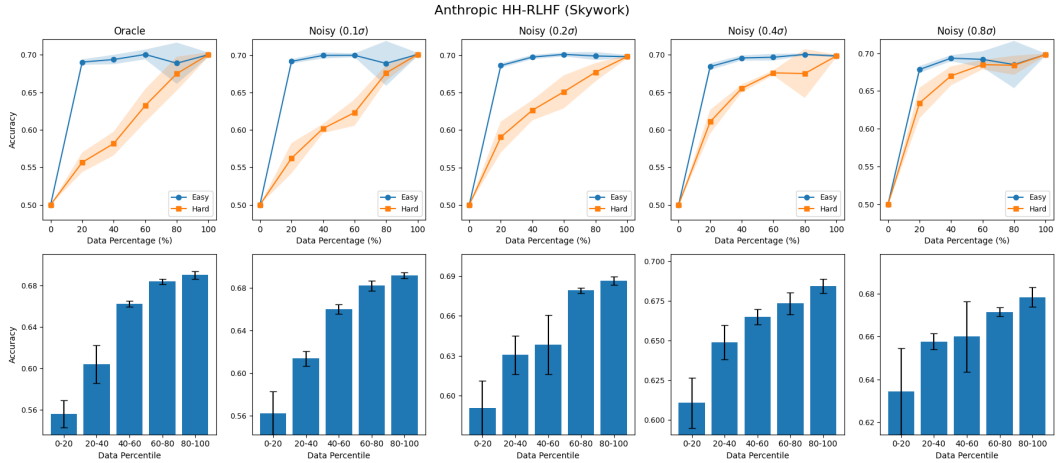


(a) Skywork reward model

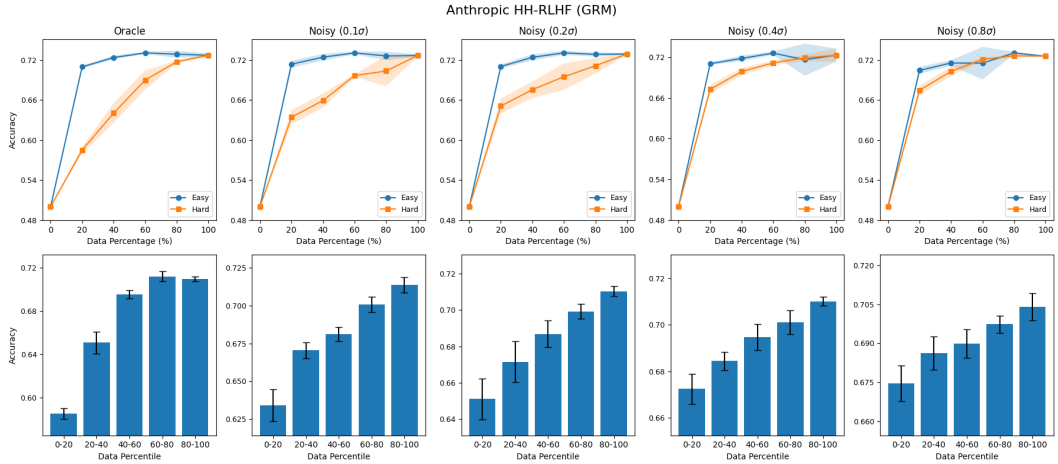


(b) GRM reward model

Figure 7: RM accuracy on the LMSYS Arena dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

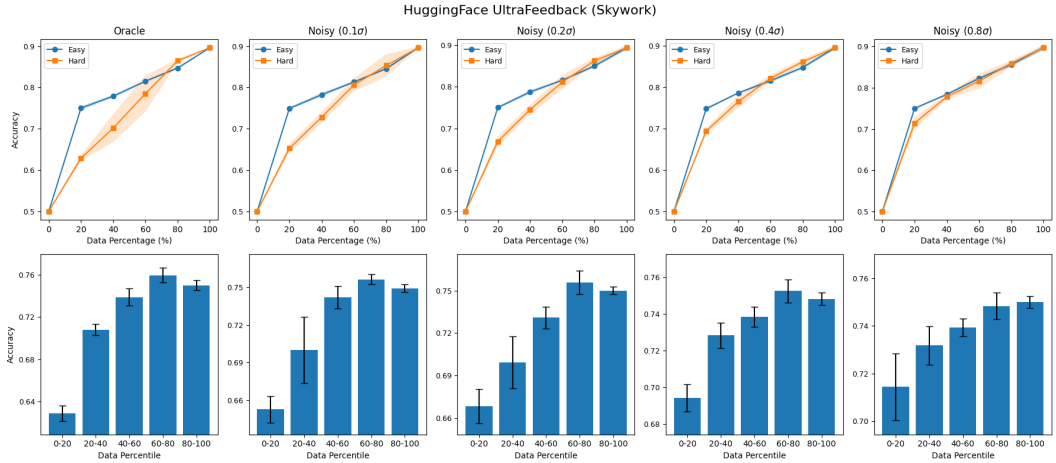


(a) Skywork reward model

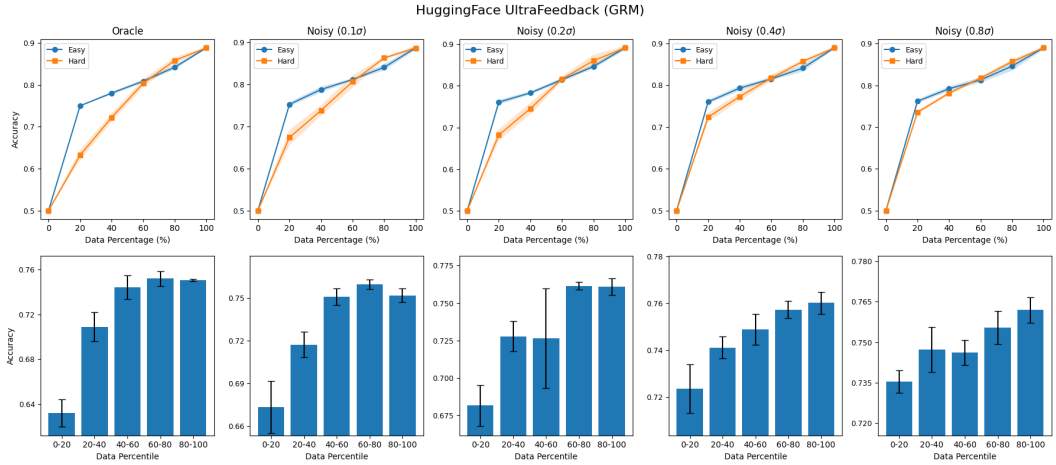


(b) GRM reward model

Figure 8: RM accuracy on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Skywork reward model

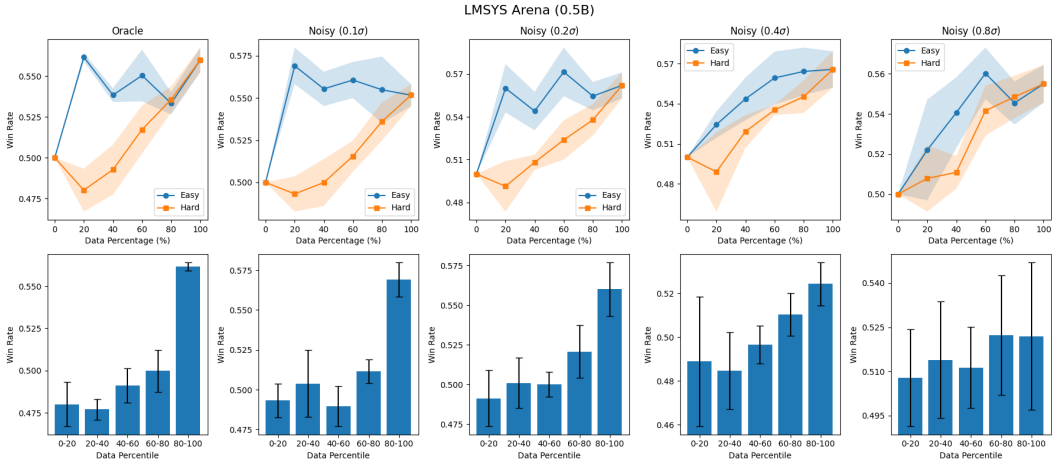


(b) GRM reward model

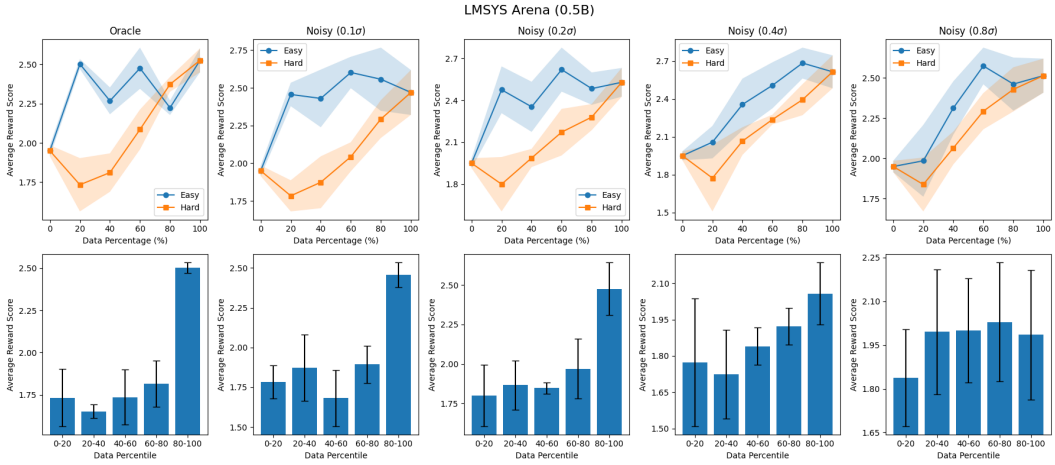
Figure 9: RM accuracy on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

B.4 DPO Training Results: Skywork Reward Model

Figures 10-18 show DPO win rate and average reward score across difficulty-based subsets, training setups and base model sizes, using Skywork reference RM. For a discussion and interpretation of these results, see Section 4.3.

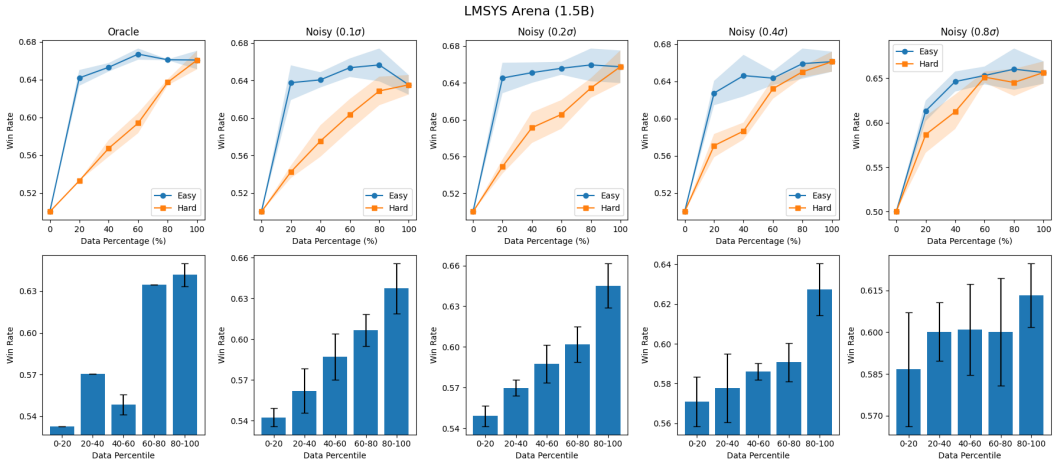


(a) Win rate

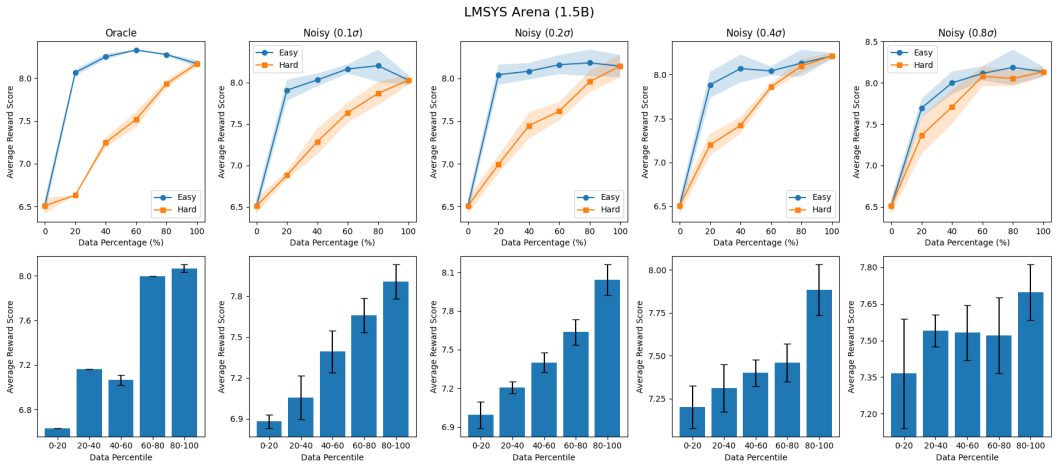


(b) Average reward score

Figure 10: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

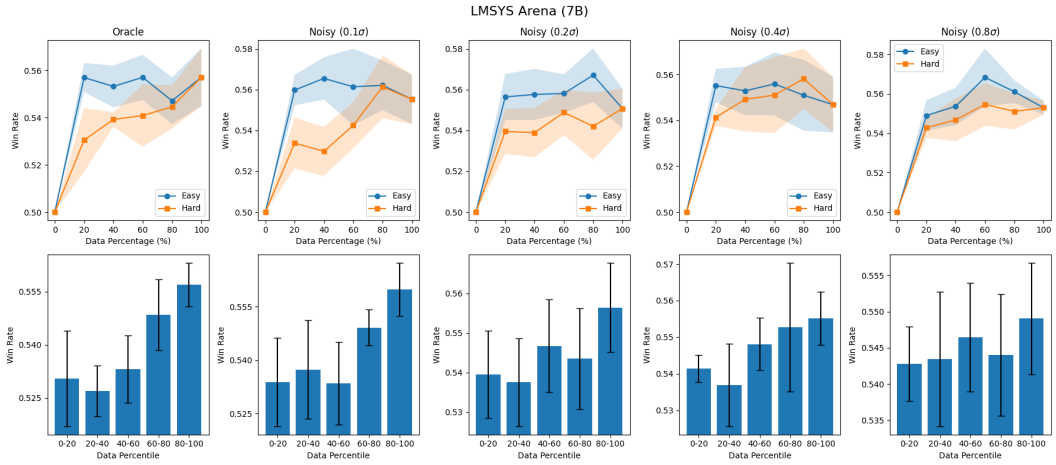


(a) Win rate

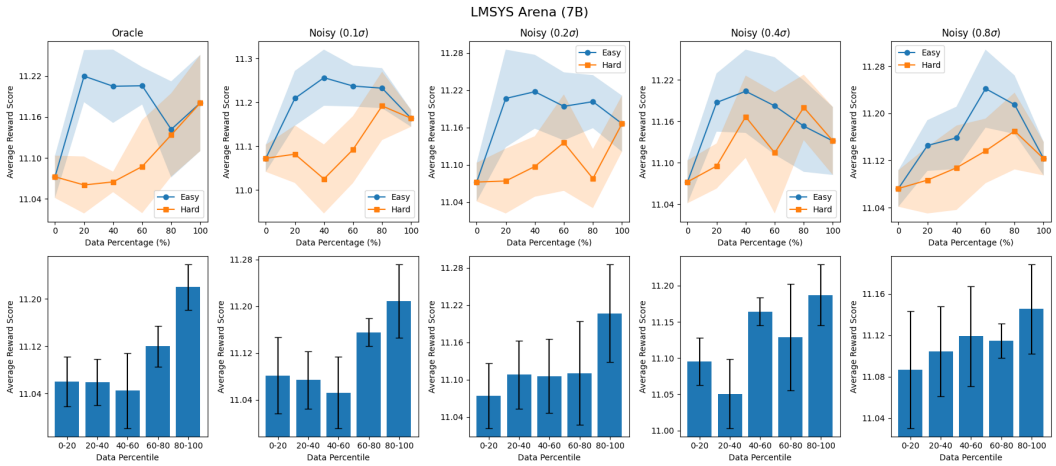


(b) Average reward score

Figure 11: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

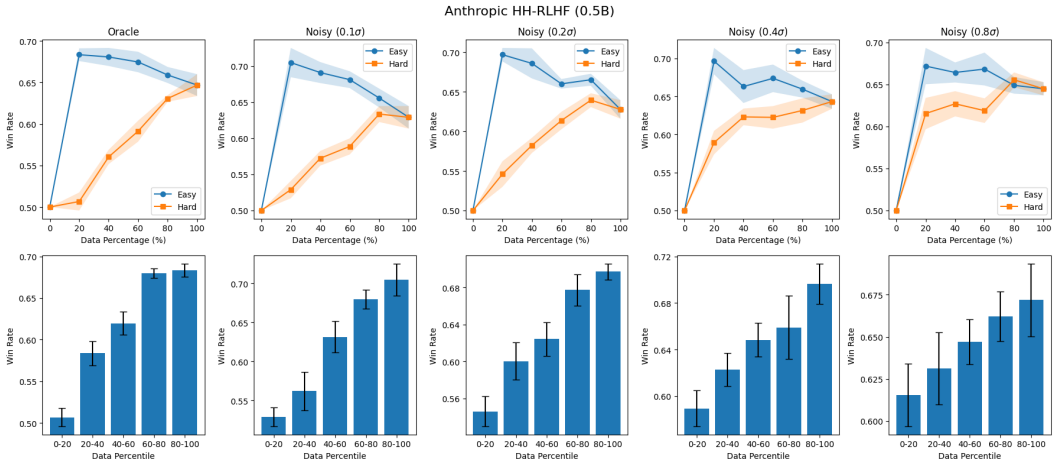


(a) Win rate

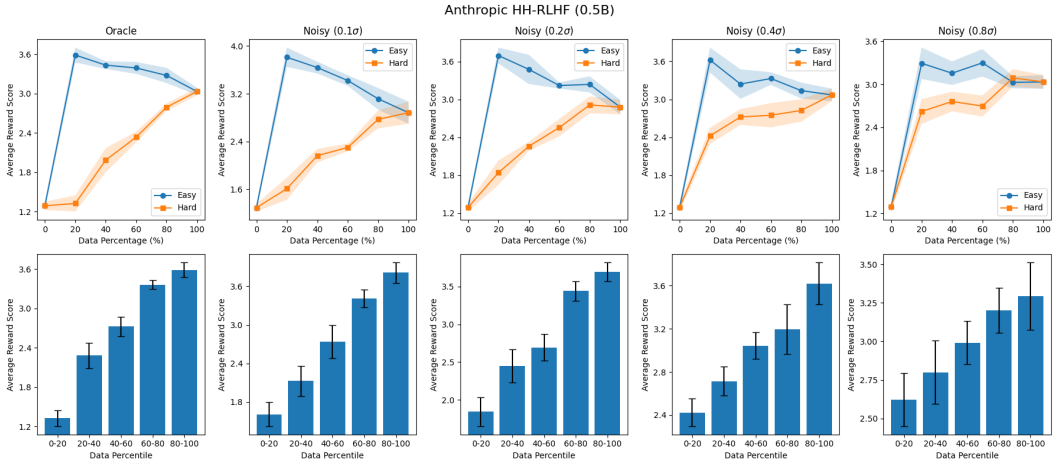


(b) Average reward score

Figure 12: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

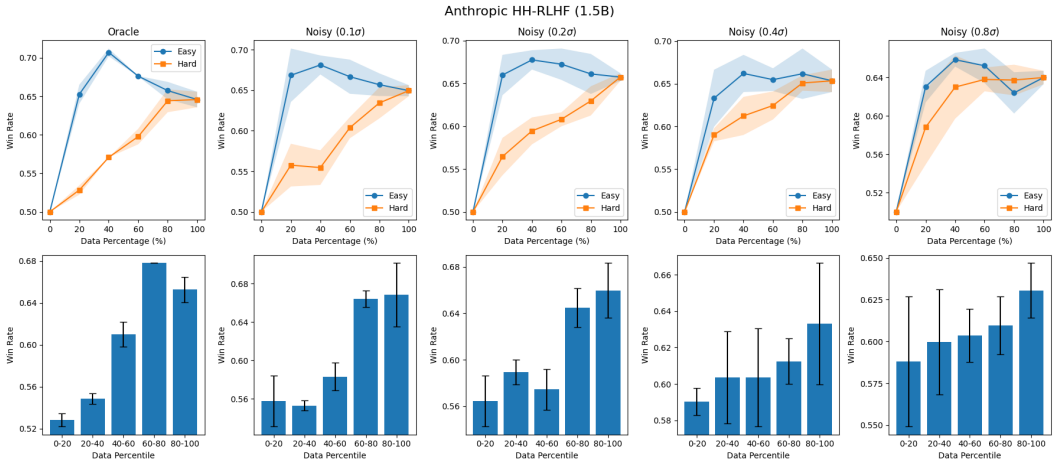


(a) Win rate

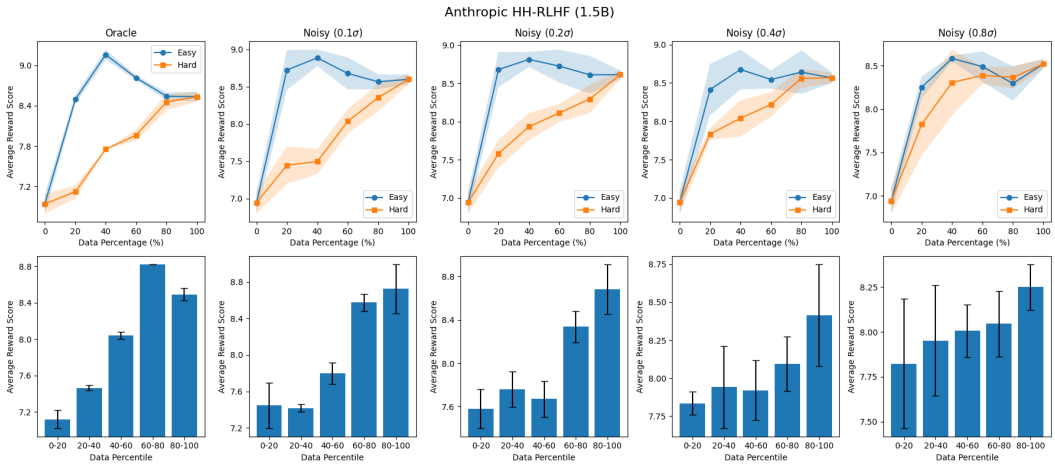


(b) Average reward score

Figure 13: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

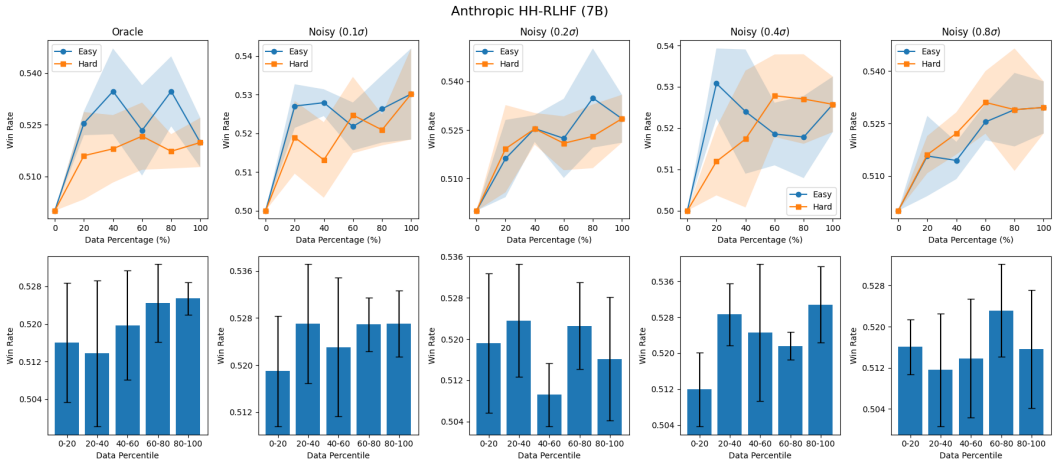


(a) Win rate

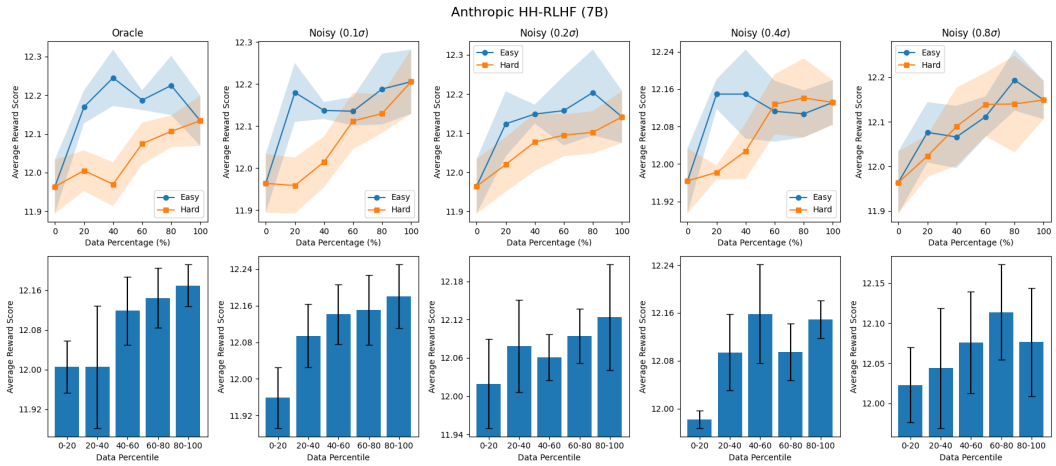


(b) Average reward score

Figure 14: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

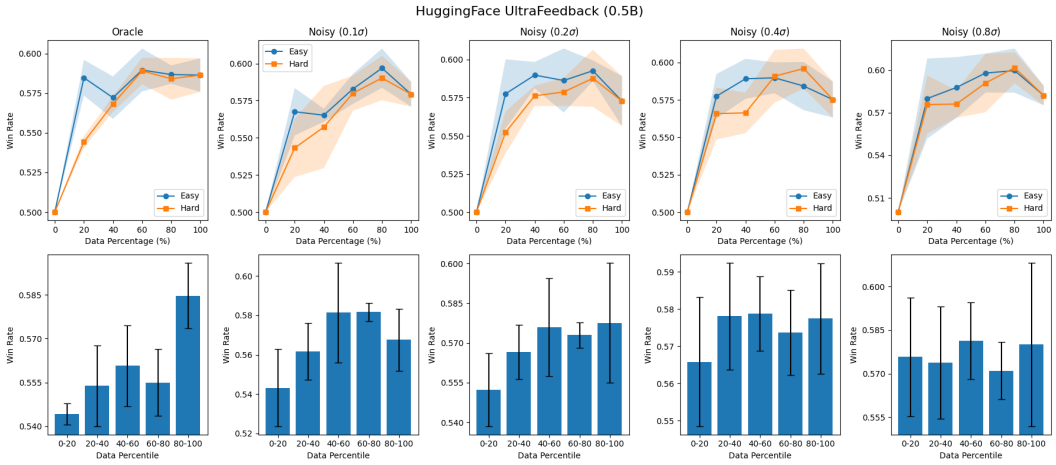


(a) Win rate

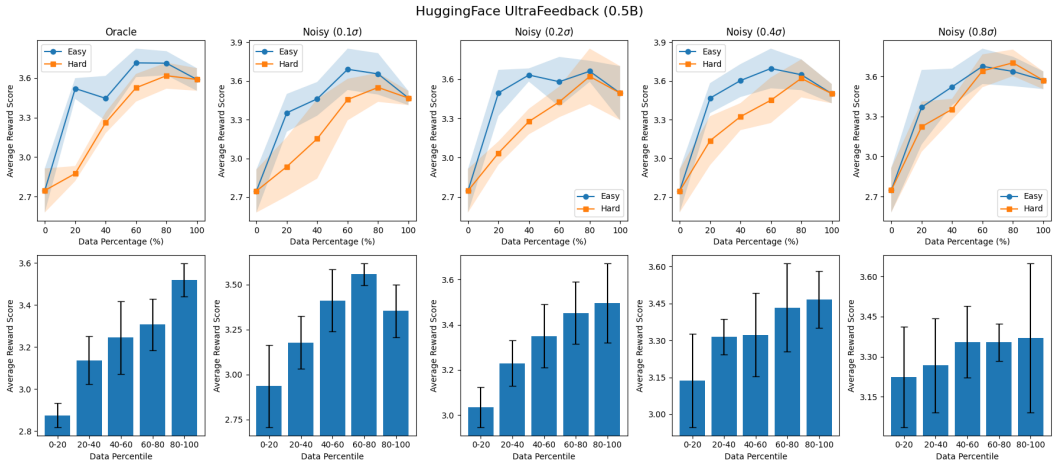


(b) Average reward score

Figure 15: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

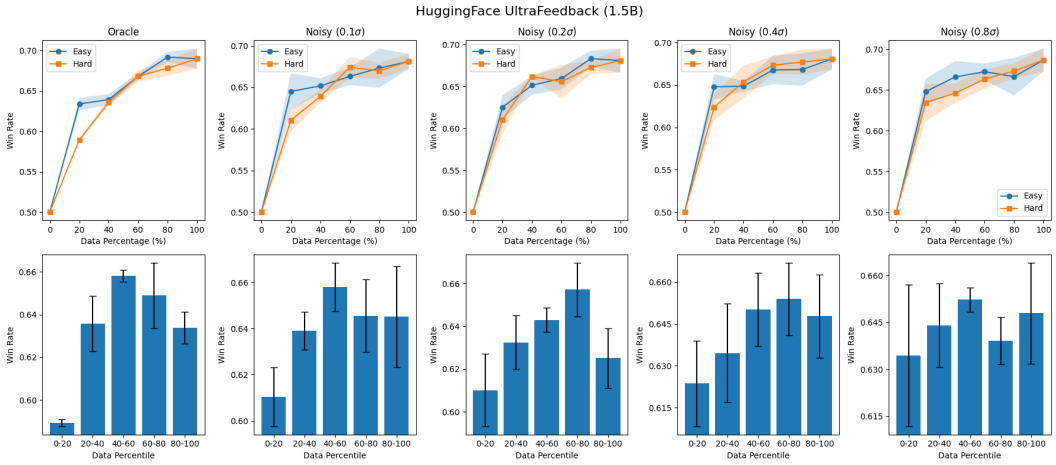


(a) Win rate

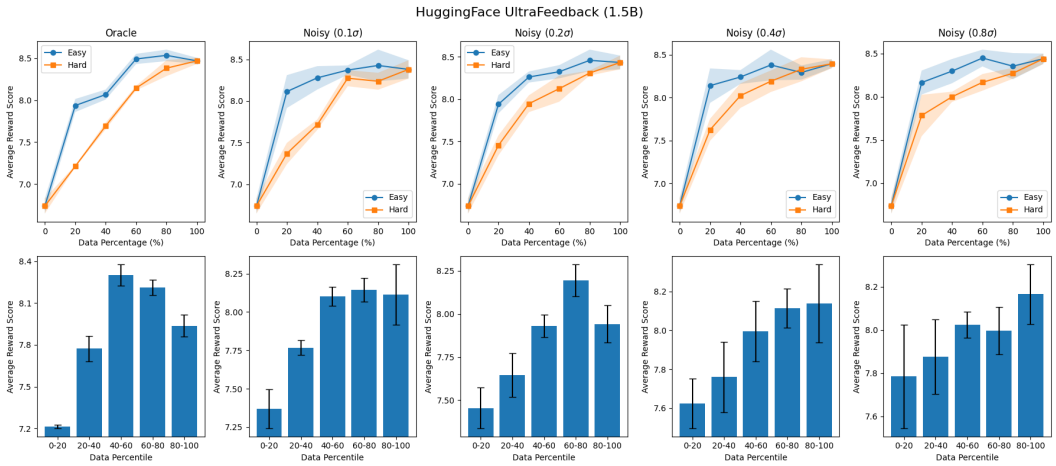


(b) Average reward score

Figure 16: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

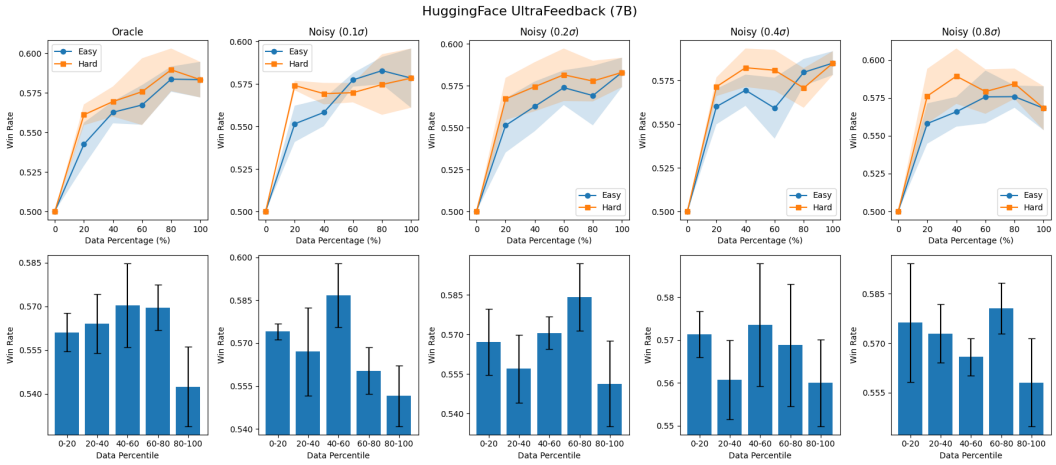


(a) Win rate

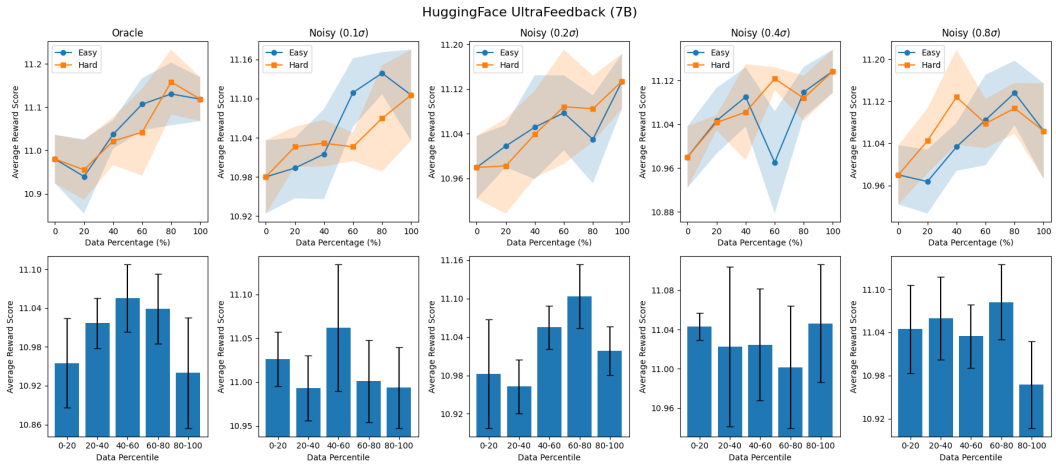


(b) Average reward score

Figure 17: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Win rate

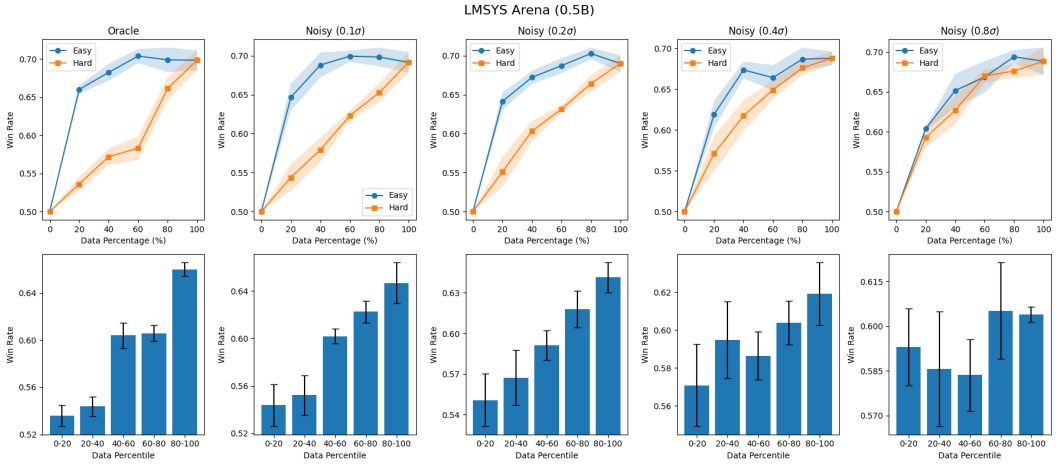


(b) Average reward score

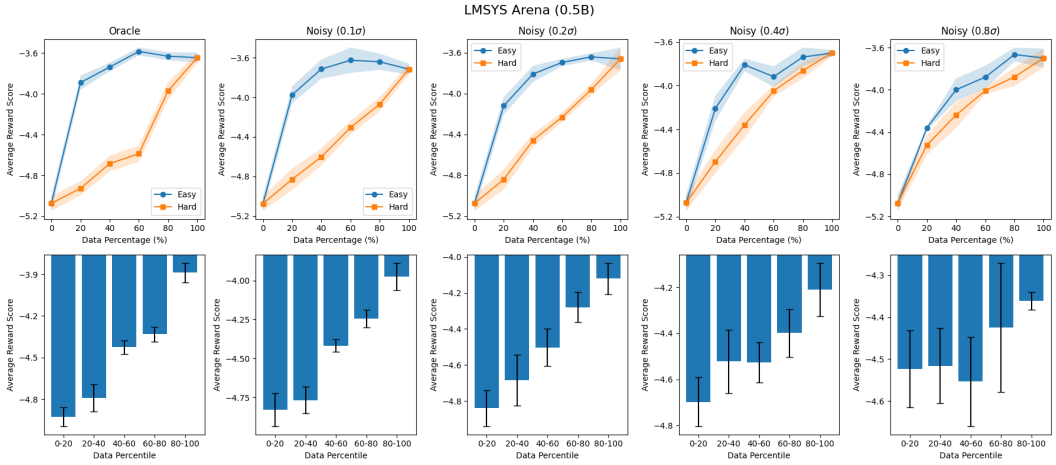
Figure 18: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

B.5 DPO Training Results: GRM Reward Model

Figures 19-27 show DPO win rate and average reward score across difficulty-based subsets, training setups and base model sizes, using GRM reference RM. For a discussion and interpretation of these results, see Section 4.3.

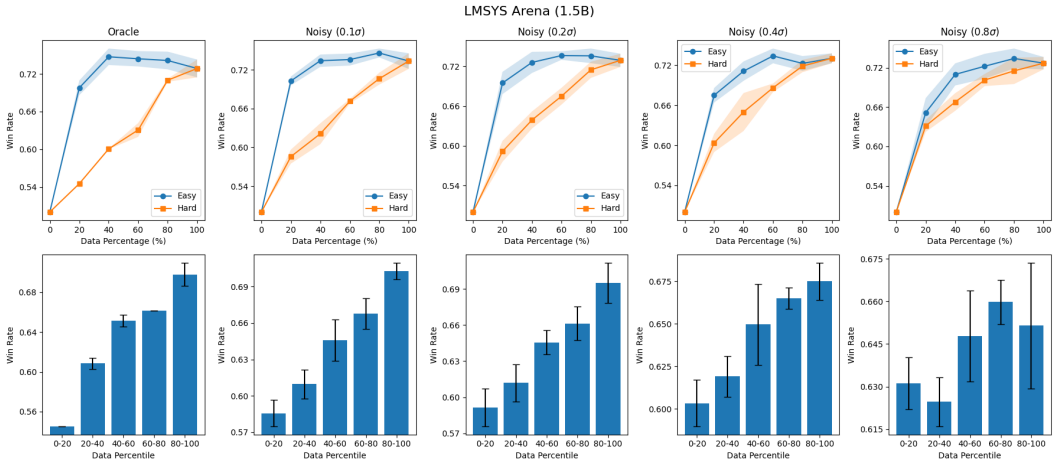


(a) Win rate

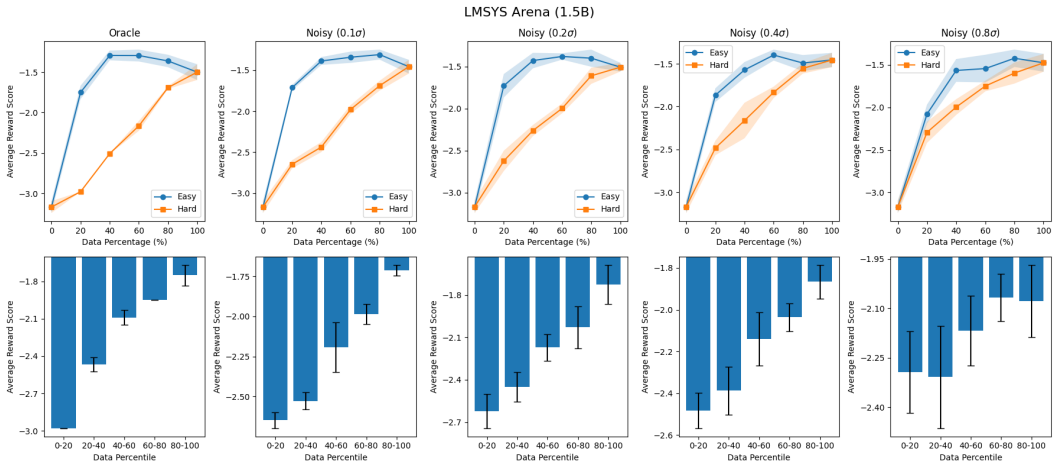


(b) Average reward score

Figure 19: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

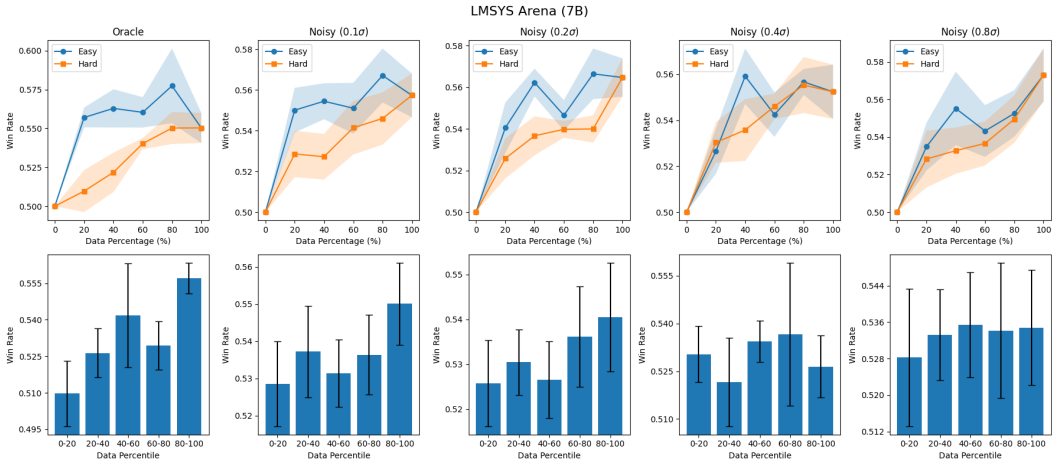


(a) Win rate

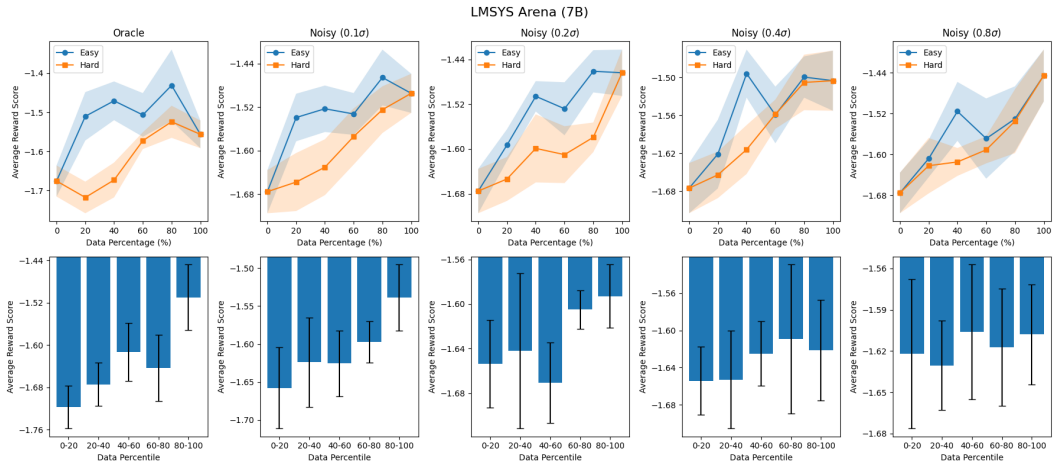


(b) Average reward score

Figure 20: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

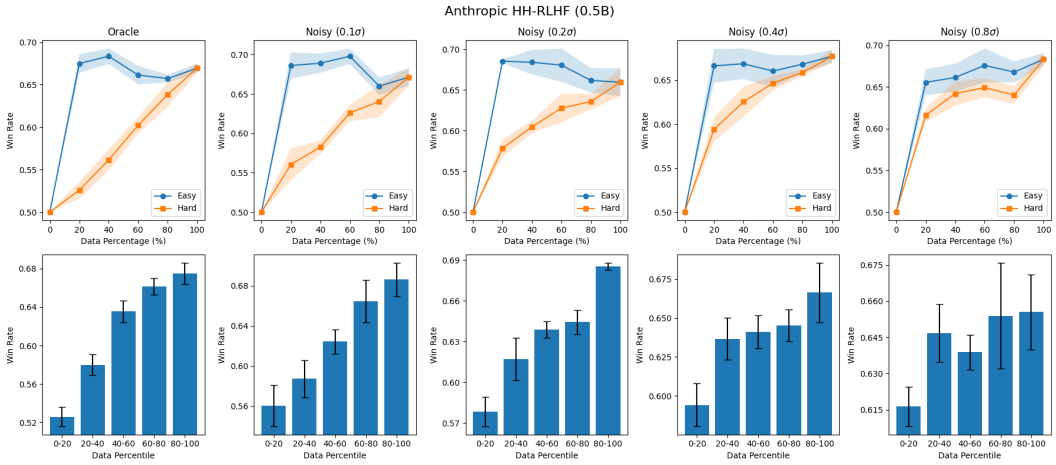


(a) Win rate

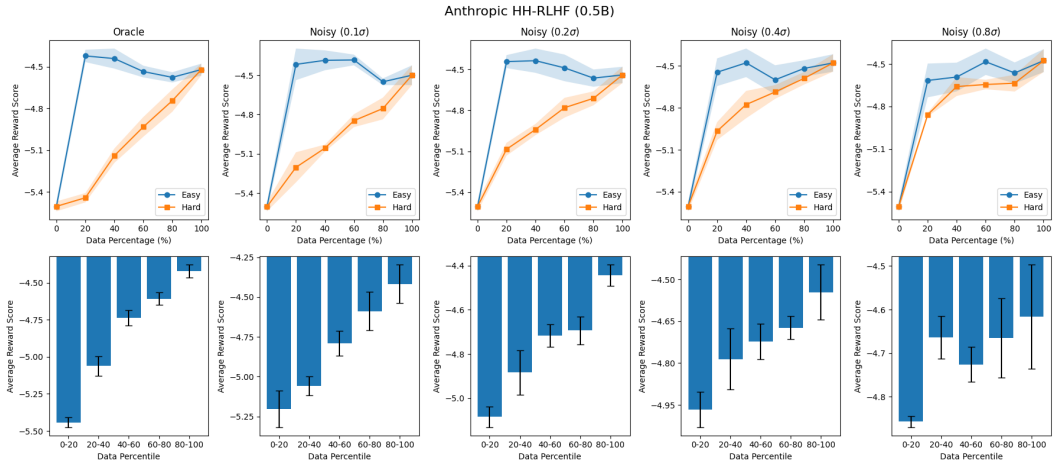


(b) Average reward score

Figure 21: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

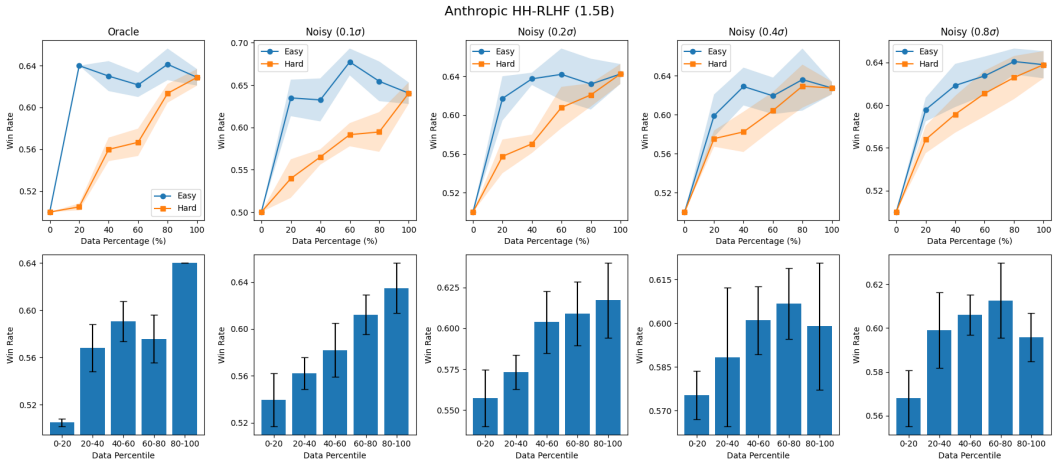


(a) Win rate

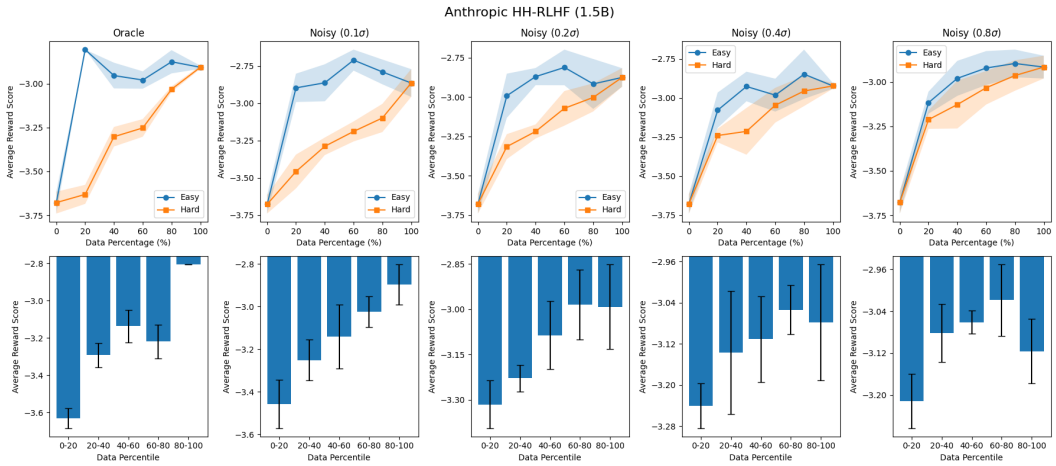


(b) Average reward score

Figure 22: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

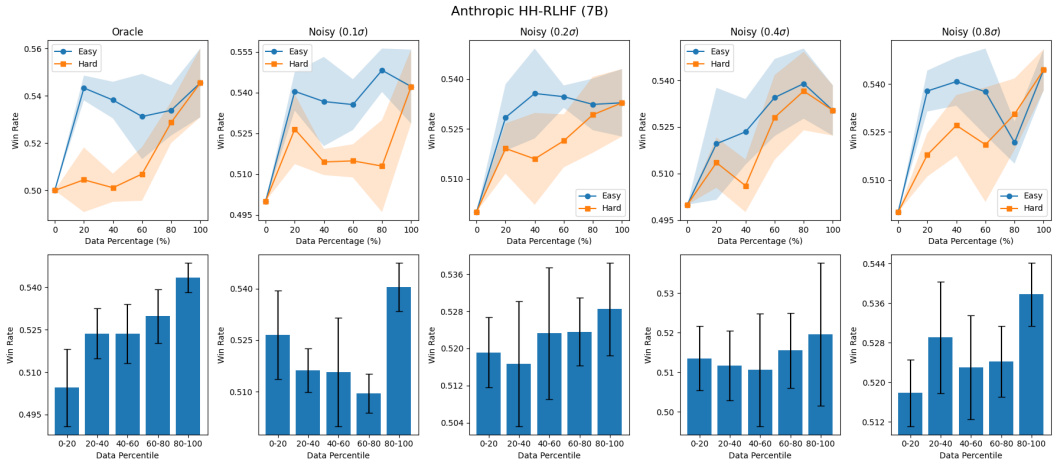


(a) Win rate

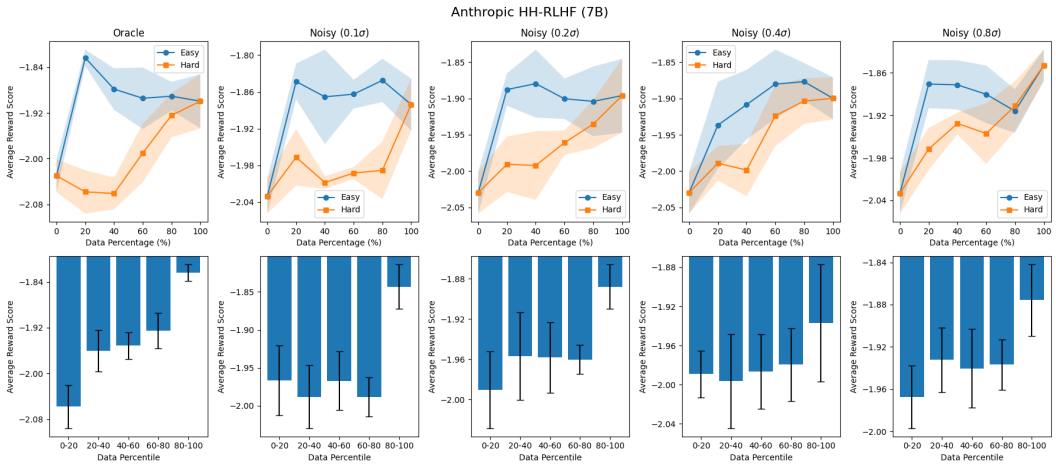


(b) Average reward score

Figure 23: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

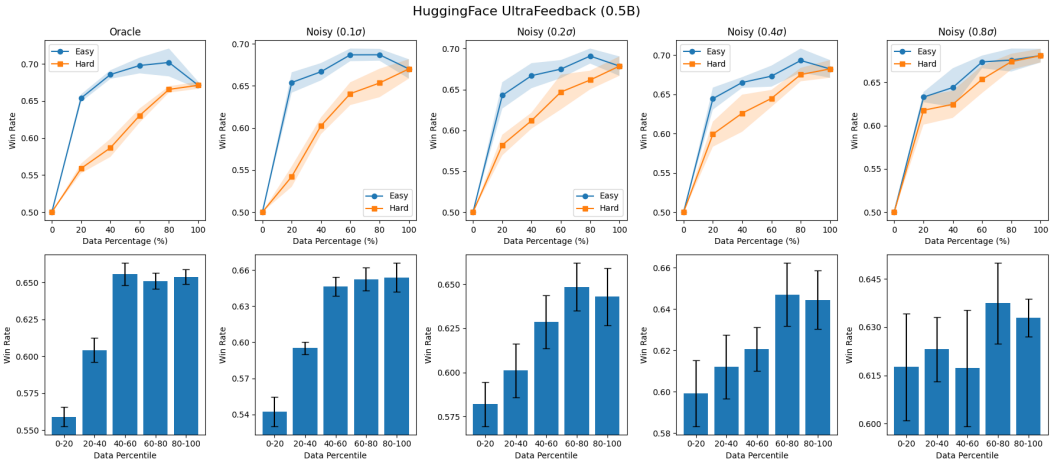


(a) Win rate

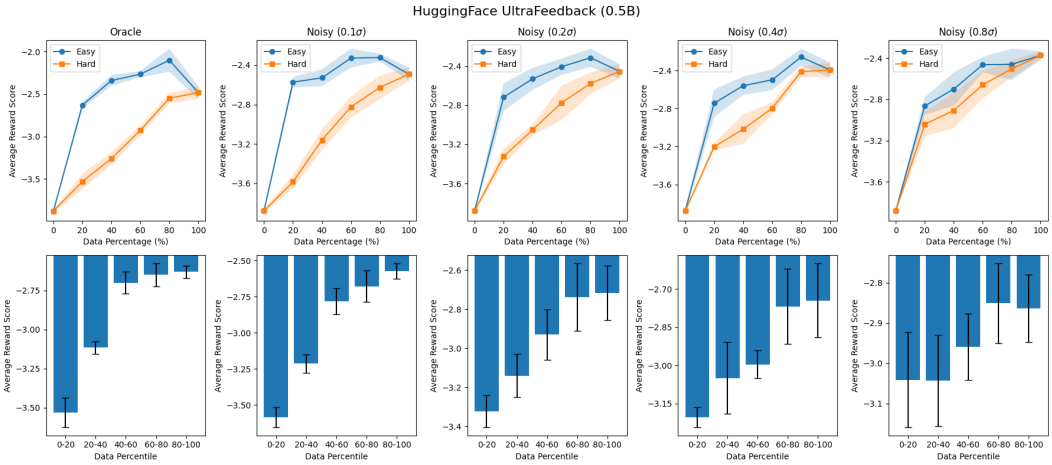


(b) Average reward score

Figure 24: DPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

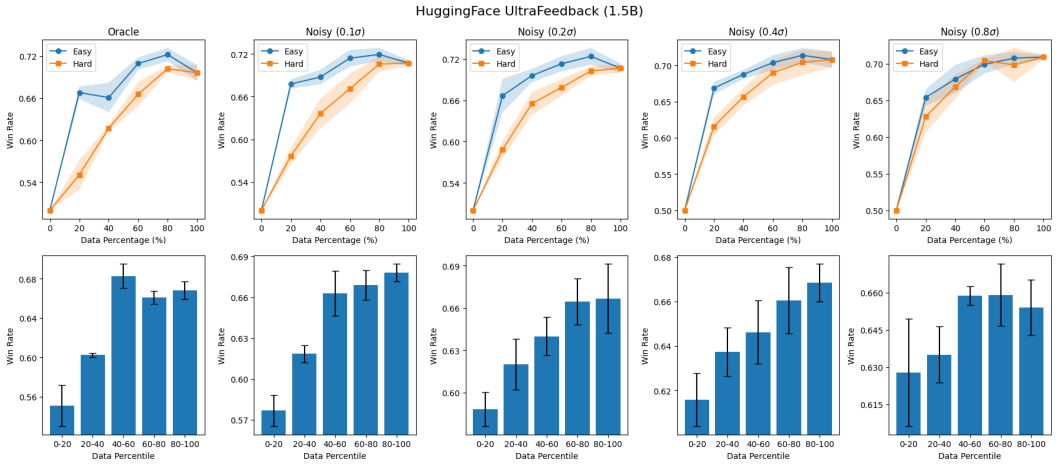


(a) Win rate

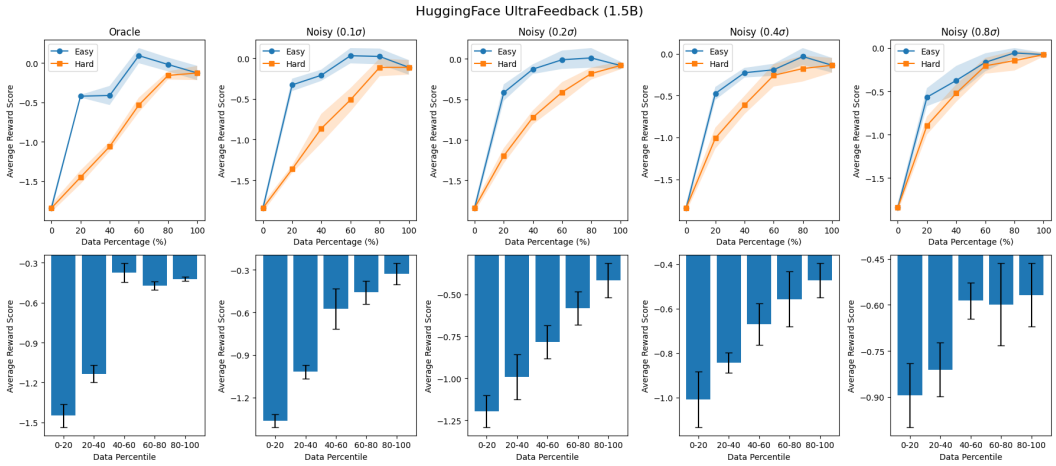


(b) Average reward score

Figure 25: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

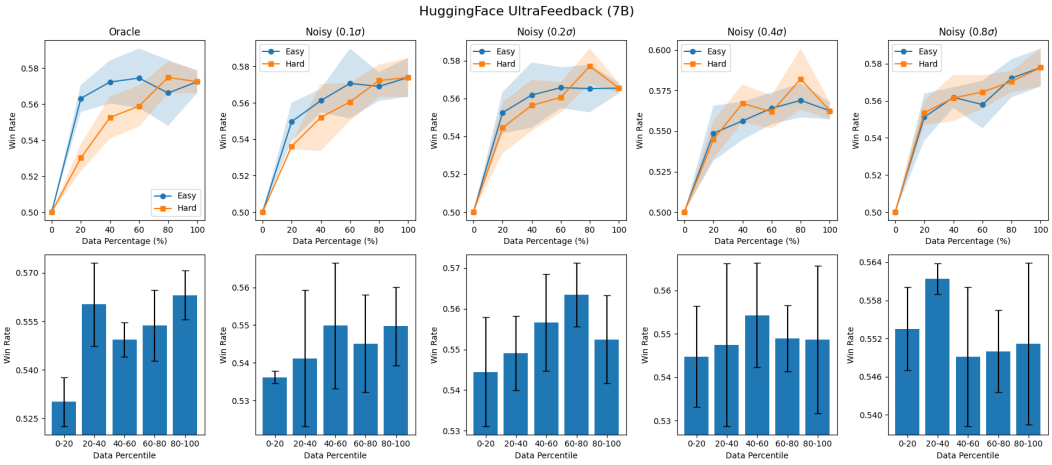


(a) Win rate

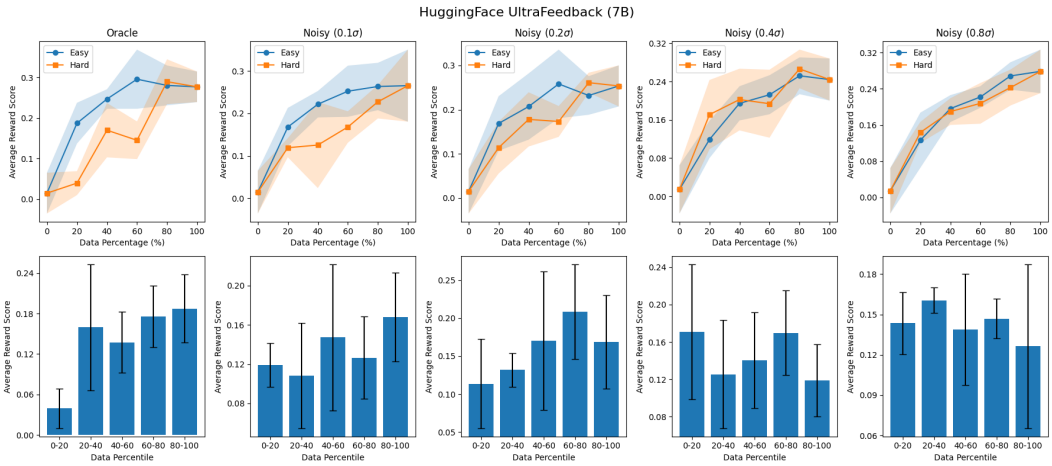


(b) Average reward score

Figure 26: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Win rate



(b) Average reward score

Figure 27: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

B.6 Reference-Independent Evaluation

Original-label RM evaluation. We train Pythia-1B RMs using Skywork-defined bins and Skywork-consistent labels, but evaluate against untouched original dataset preferences. Table 7 reports fixed-size difficulty slices. Easiest-minus-hardest accuracy differences are +6.1 percentage points on Anthropic and +4.9 on LMSYS, positive in both RM seeds for each dataset. The agreement-stratified results in Table 2 refer to held-out evaluation pairs, not an ablation that removes relabeled training pairs. On disagreement pairs, opposite binary labels imply opposite correctness, apart from ties; the stratification diagnoses target mismatch rather than independently establishing a causal effect of filtering on bias.

Table 7: RM accuracy against original dataset preferences. Training labels and fixed-size difficulty bins use Skywork. Columns run from hardest to easiest.

Dataset	0–20%	20–40%	40–60%	60–80%	80–100%
Anthropic	0.523	0.536	0.560	0.572	0.584
LMSYS	0.510	0.536	0.543	0.542	0.559

External-judge DPO evaluation. In the main experiments, DPO performance is evaluated using the same reference RM that defines example difficulty. To assess whether the observed trends persist under an evaluation signal that is independent of the reference RM, we additionally evaluate DPO-trained models using AlpacaEval. We focus on models trained on LMSYS Arena with difficulty bins defined by the Skywork reference RM.

Table 8 shows AlpacaEval win rates for DPO models trained on fixed-size difficulty slices. The bins are ordered from hardest to easiest according to the reference reward gap: 0–20% denotes the hardest examples and 80–100% denotes the easiest examples. For the 0.5B and 1.5B models, AlpacaEval performance improves monotonically as training data shifts from harder to easier examples. Specifically, the 0.5B model improves from 0.12 on the hardest slice to 0.22 on the easiest slice, while the 1.5B model improves from 0.57 to 0.70. For the 7B model, performance is high across all bins and largely saturates, suggesting that larger models may be less sensitive to difficulty-based data selection under this evaluation.

These results suggest that the advantage of easier examples is not solely an artifact of evaluating with the same reference RM used for data selection. Indeed, the same qualitative trend persists under AlpacaEval, an external reference-independent evaluation, especially for smaller models.

Table 8: AlpacaEval win rates for DPO models trained on LMSYS Arena using difficulty slices defined by the Skywork reference RM. Bins are ordered from hardest to easiest according to the reference reward gap: 0–20% denotes the hardest examples and 80–100% denotes the easiest examples.

Model	0–20%	20–40%	40–60%	60–80%	80–100%
0.5B	0.12	0.15	0.17	0.21	0.22
1.5B	0.57	0.62	0.64	0.67	0.70
7B	0.94	0.95	0.95	0.95	0.94

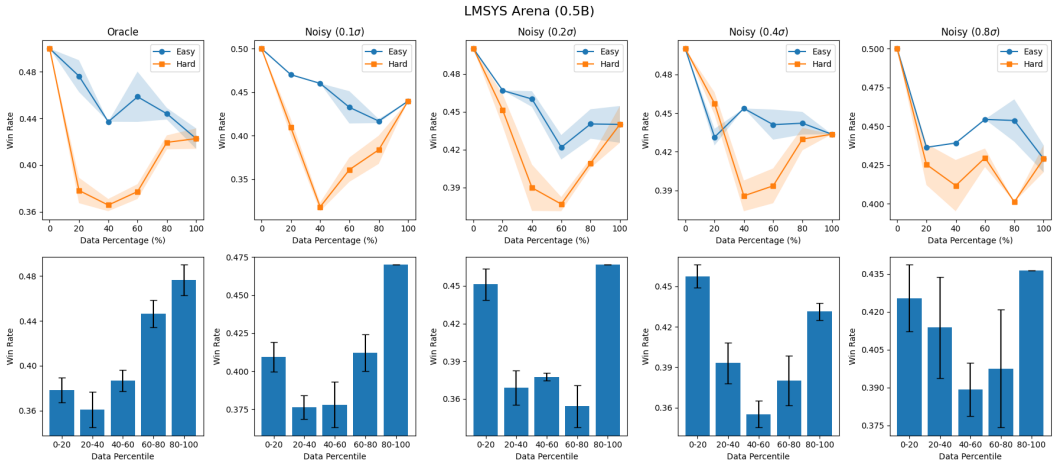
We also evaluate DPO models trained on Anthropic (Table 9). The easiest-minus-hardest differences are +11.5 points at 0.5B, +16.8 at 1.5B, and -1.5 at 7B. Intermediate bins are not strictly monotone. Together with LMSYS, these results support a smaller-model advantage in the tested configurations, without a consistent advantage at 7B. They compare easy and hard slices, not easy subsets with full-data training.

Table 9: AlpacaEval win rates for DPO models trained on Anthropic using Skywork-defined labels and fixed-size difficulty slices, from hardest to easiest.

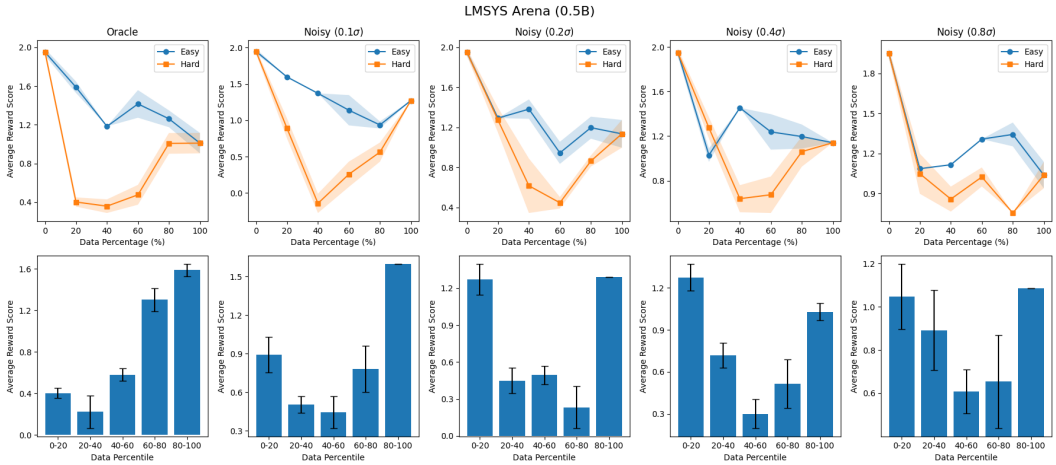
Model	0–20%	20–40%	40–60%	60–80%	80–100%
0.5B	0.09	0.12	0.20	0.16	0.20
1.5B	0.41	0.44	0.56	0.59	0.58
7B	0.94	0.93	0.94	0.93	0.92

B.7 GRPO Training Results: Skywork Reward Model

Figures 28-36 show GRPO win rate and average reward score across difficulty-based subsets, training setups and base model sizes, using Skywork reference RM. For a discussion and interpretation of these results, see Section 4.4.

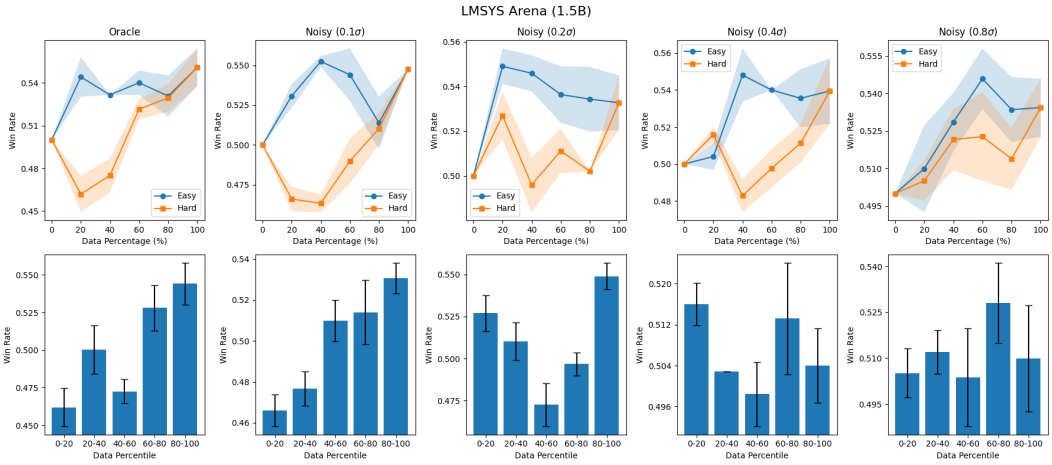


(a) Win rate

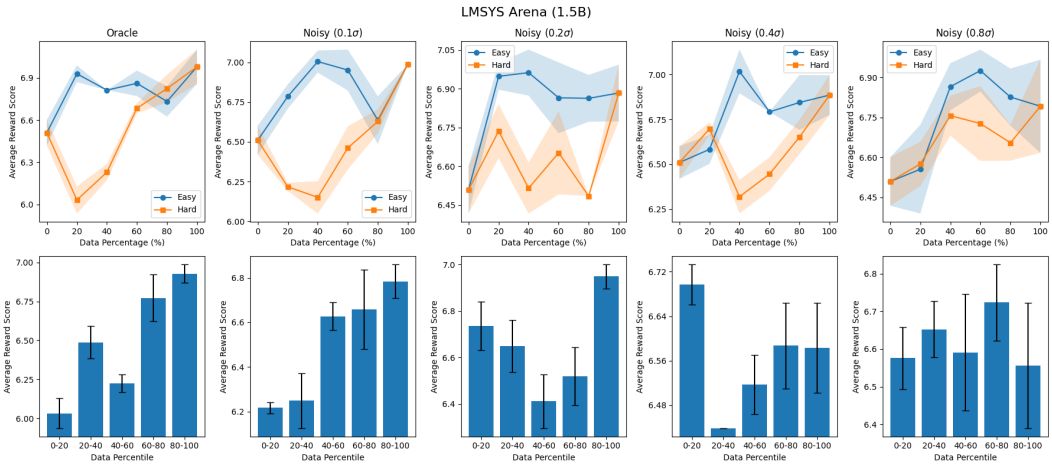


(b) Average reward score

Figure 28: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

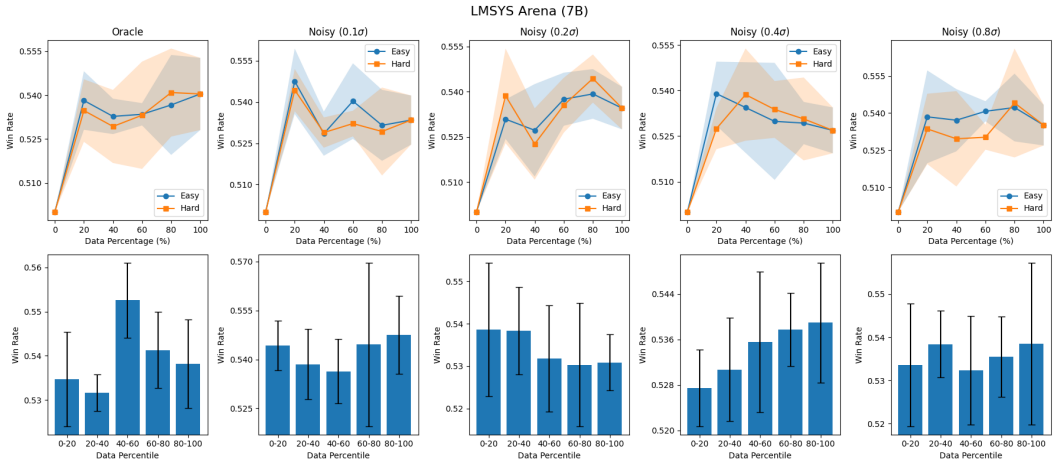


(a) Win rate

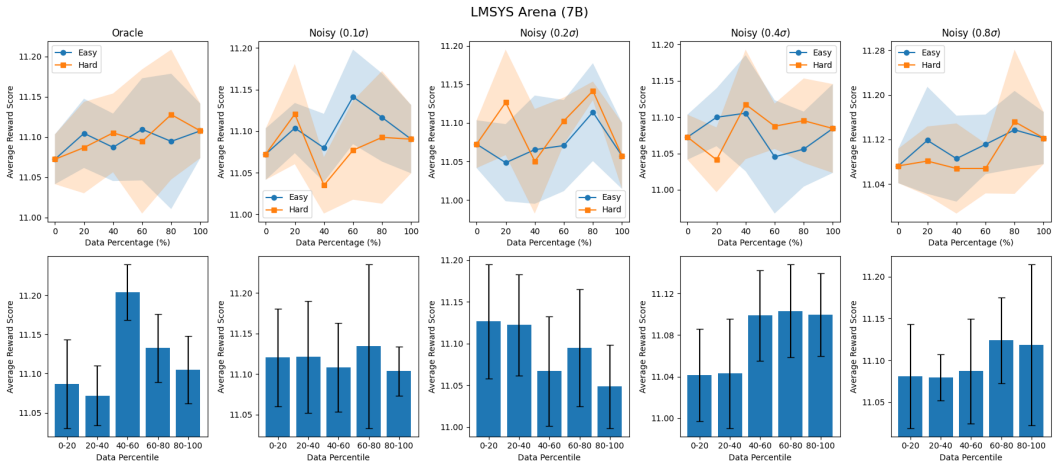


(b) Average reward score

Figure 29: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

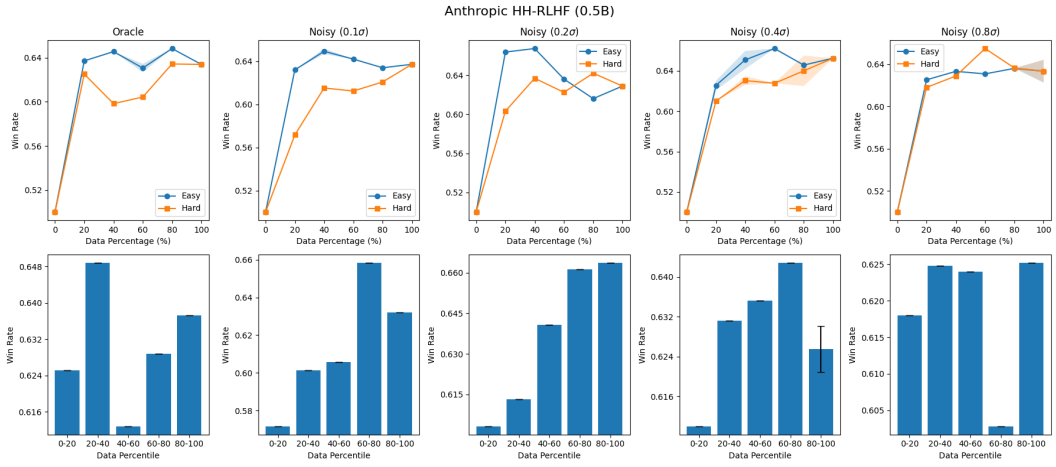


(a) Win rate

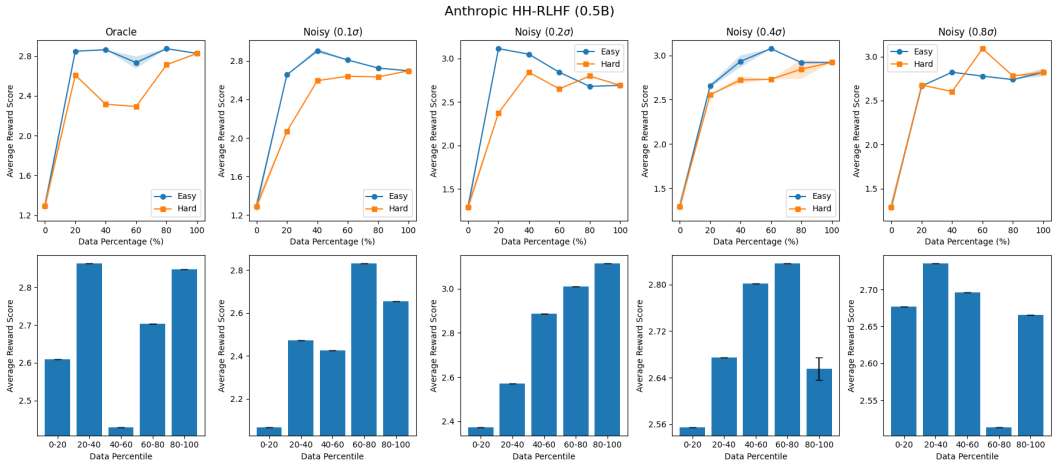


(b) Average reward score

Figure 30: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

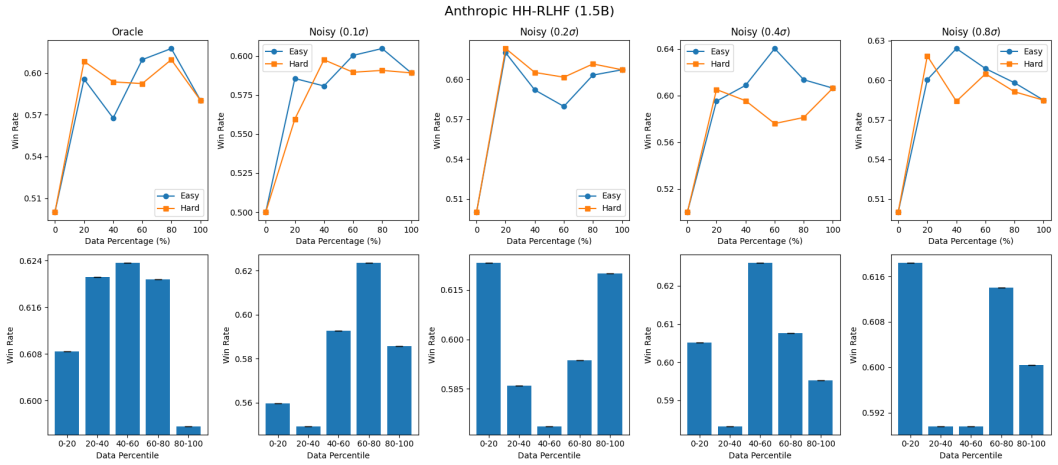


(a) Win rate

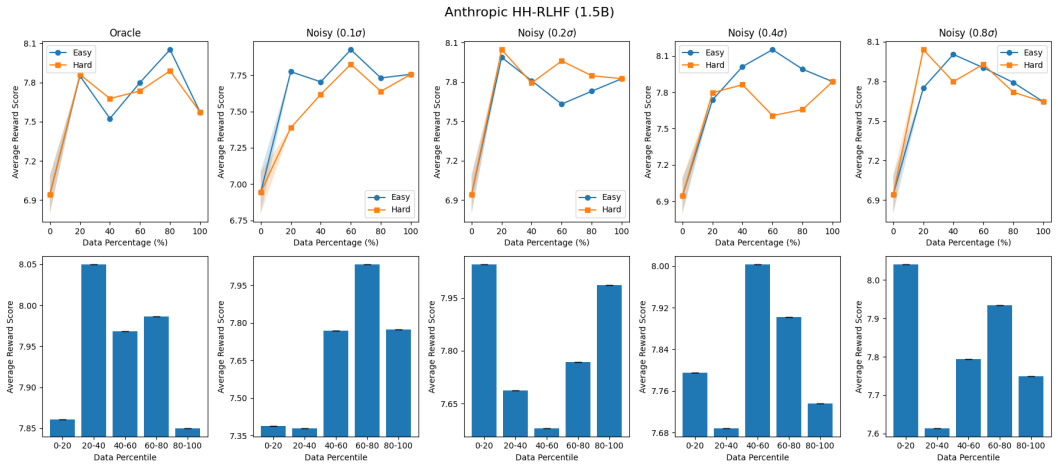


(b) Average reward score

Figure 31: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

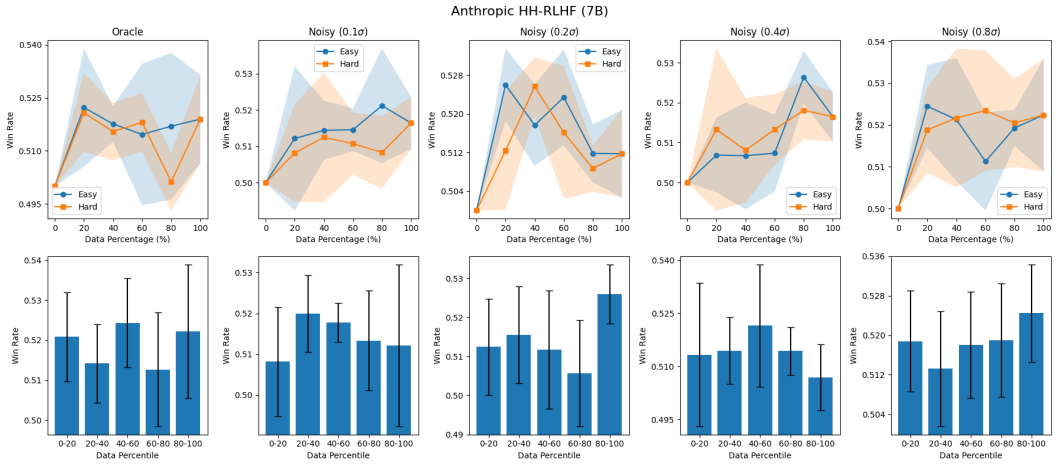


(a) Win rate

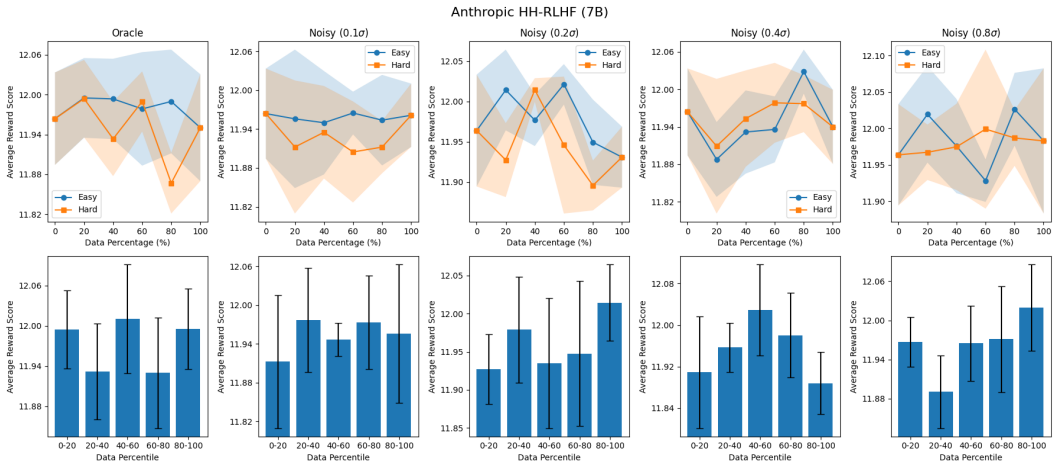


(b) Average reward score

Figure 32: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

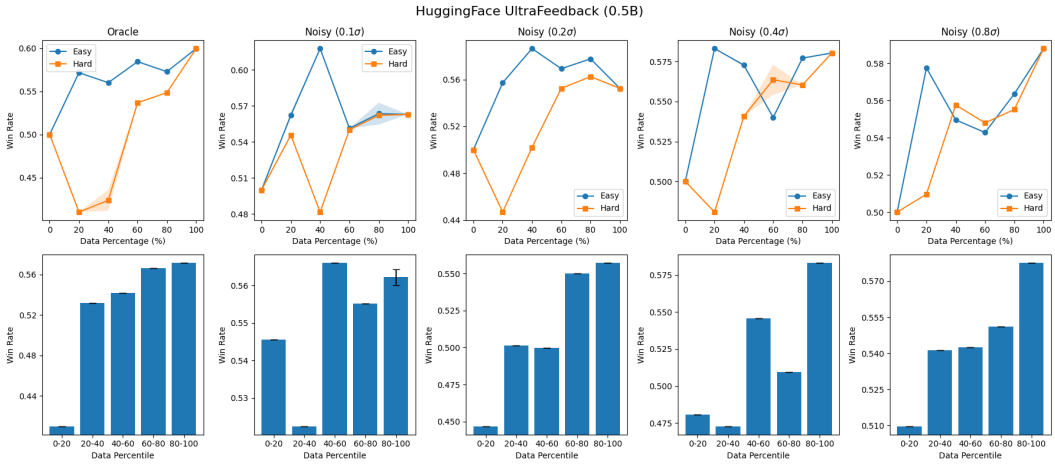


(a) Win rate

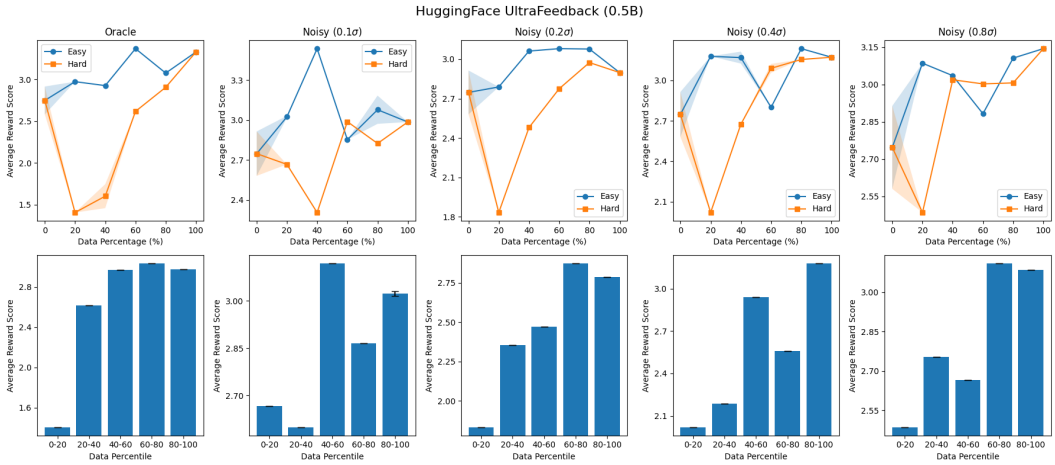


(b) Average reward score

Figure 33: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

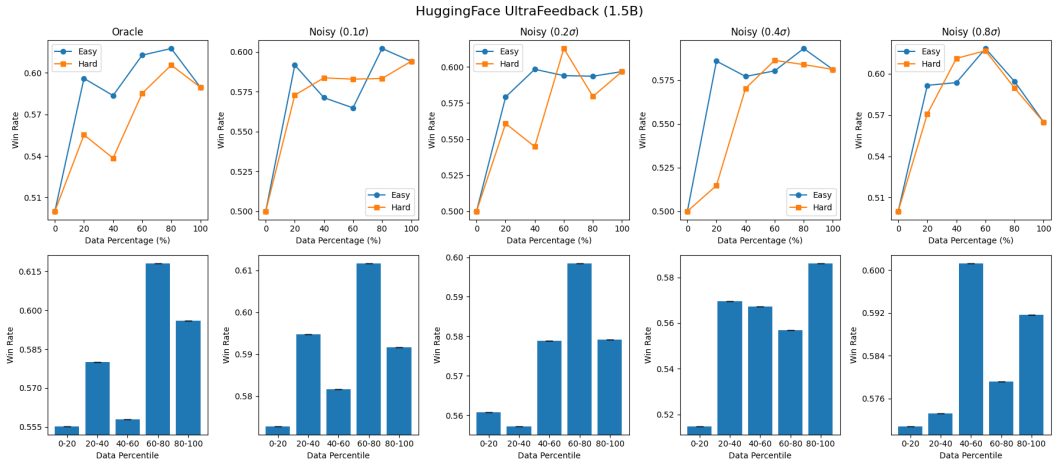


(a) Win rate

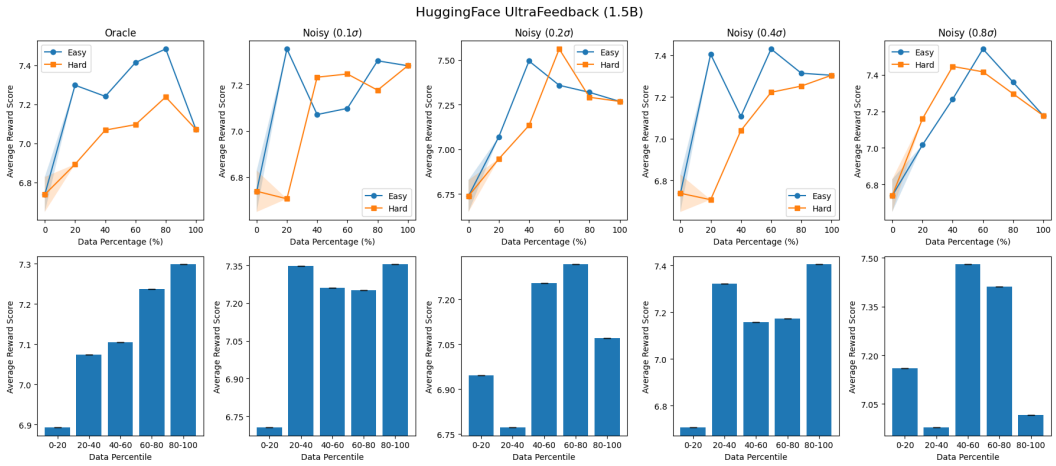


(b) Average reward score

Figure 34: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

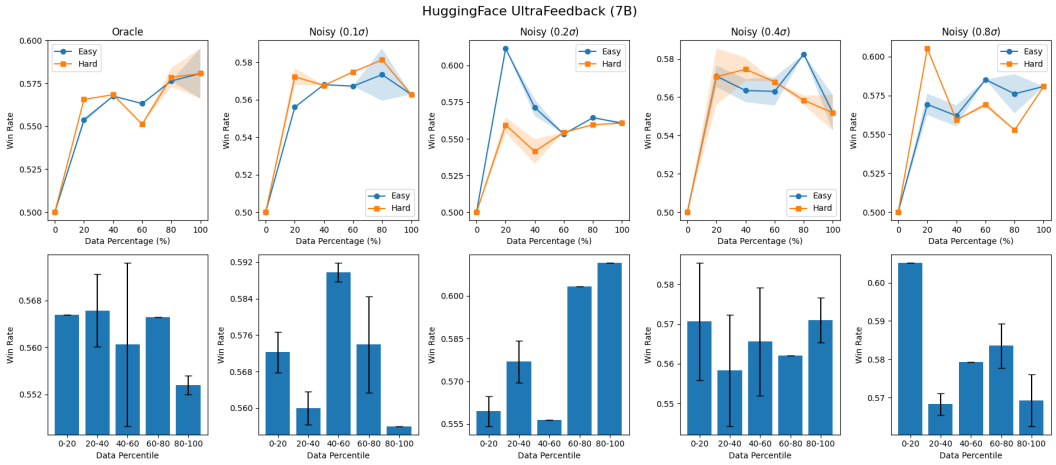


(a) Win rate

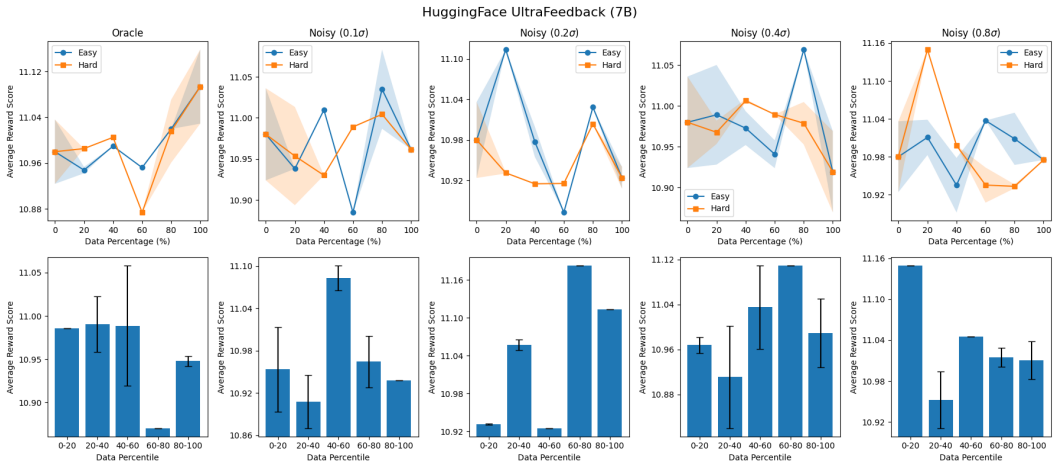


(b) Average reward score

Figure 35: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Win rate

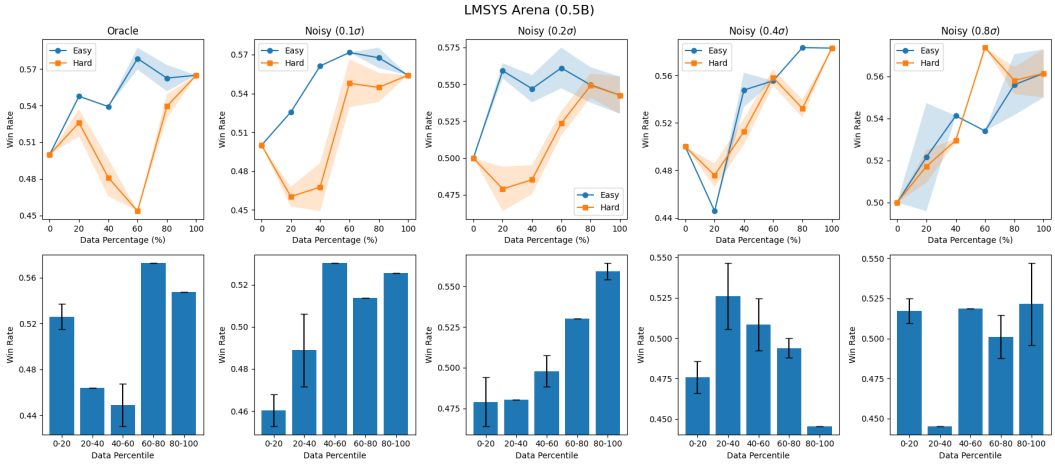


(b) Average reward score

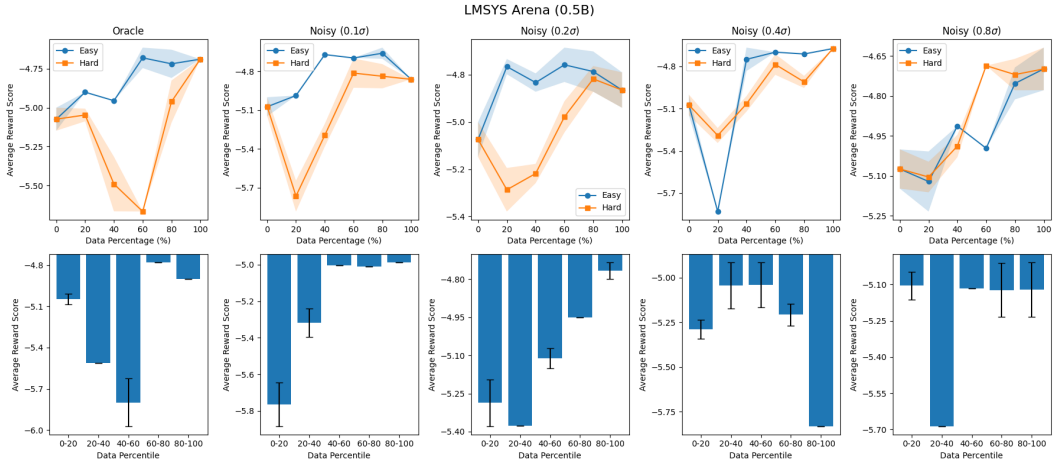
Figure 36: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

B.8 GRPO Training Results: GRM Reward Model

Figures 37-45 show GRPO win rate and average reward score across difficulty-based subsets, training setups and base model sizes, using GRM reference RM. For a discussion and interpretation of these results, see Section 4.4.

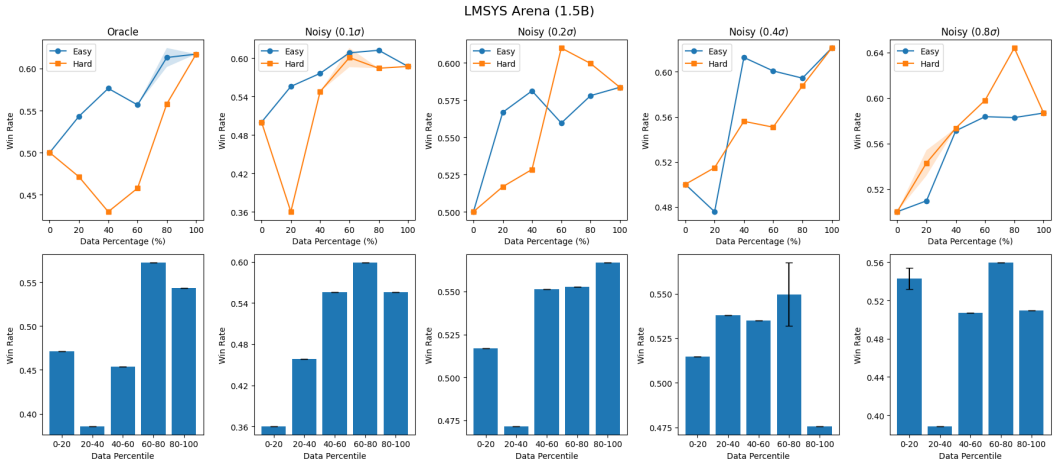


(a) Win rate

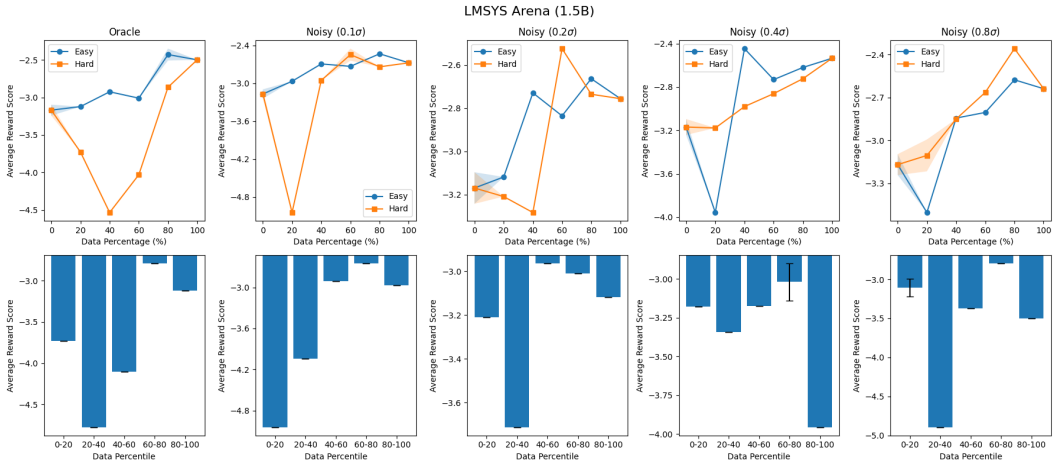


(b) Average reward score

Figure 37: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

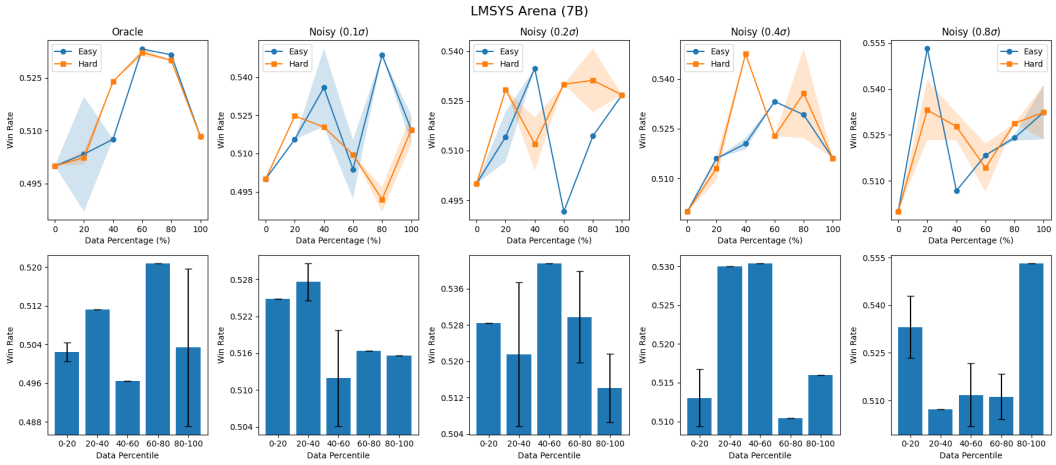


(a) Win rate

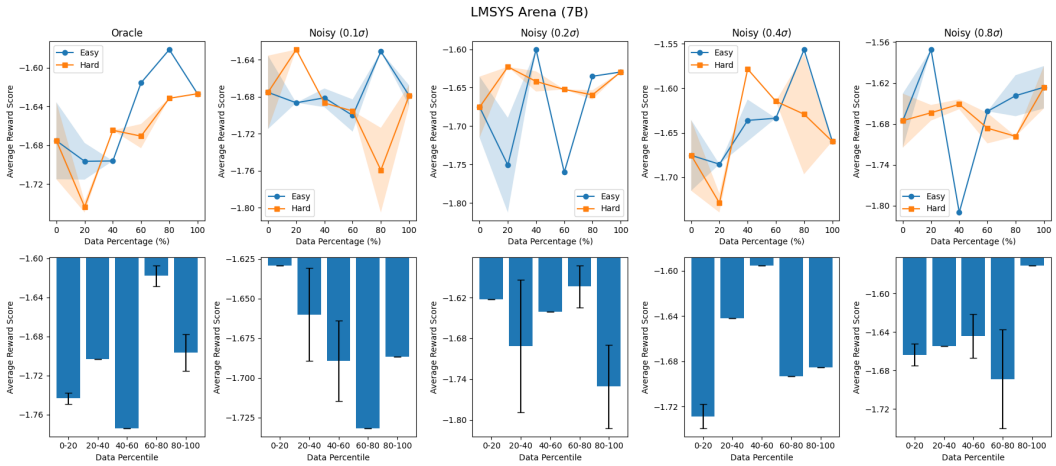


(b) Average reward score

Figure 38: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

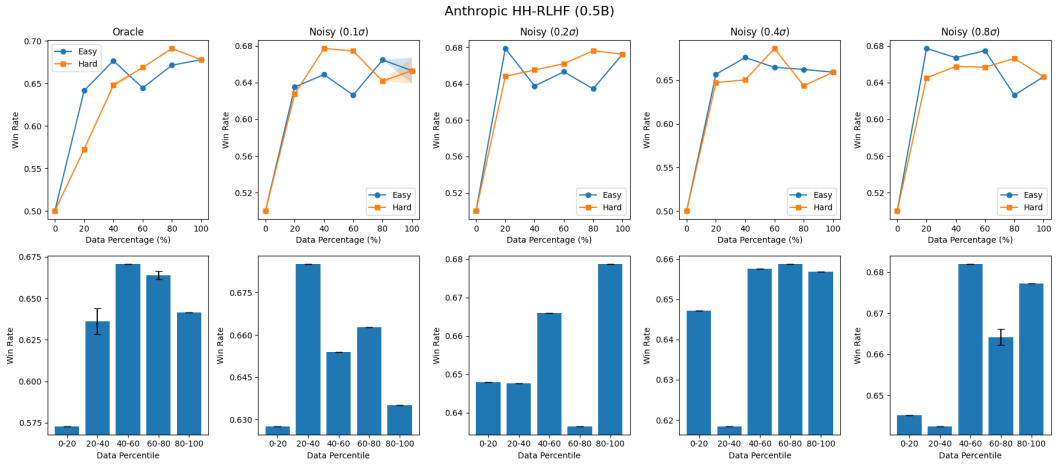


(a) Win rate

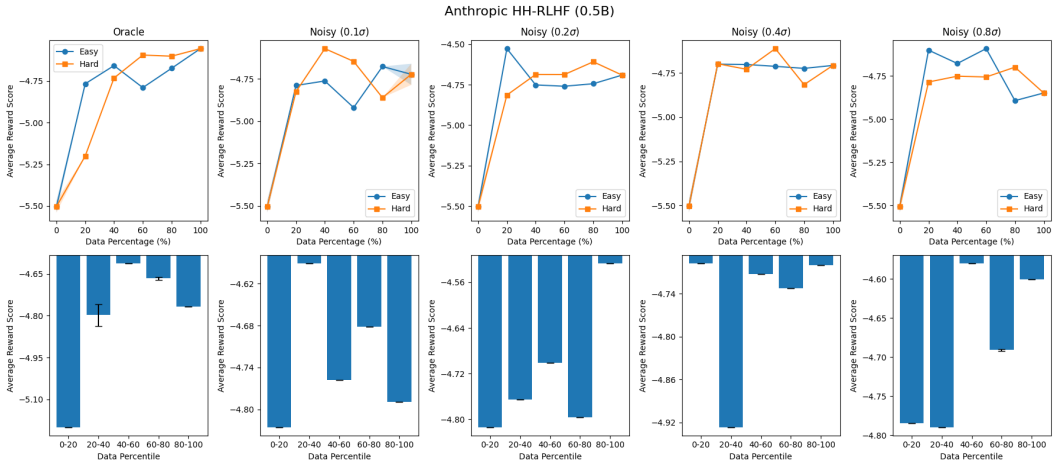


(b) Average reward score

Figure 39: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

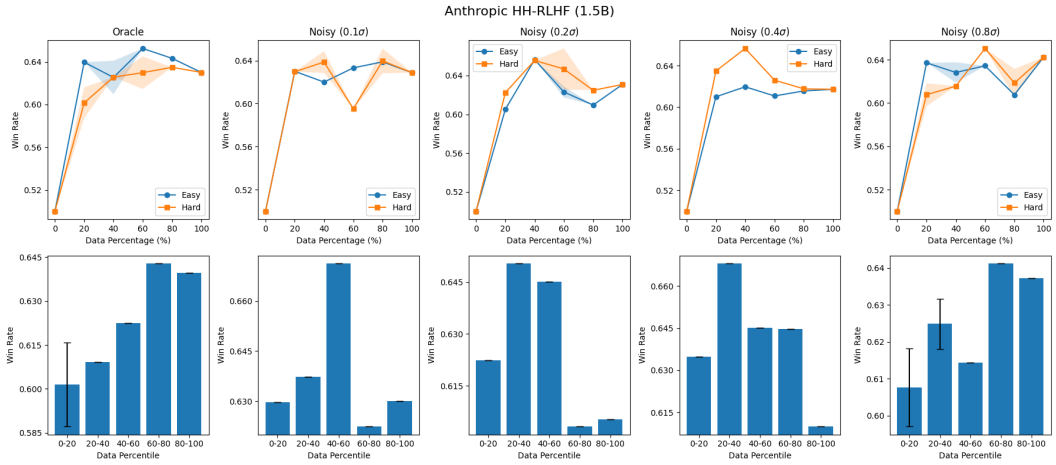


(a) Win rate

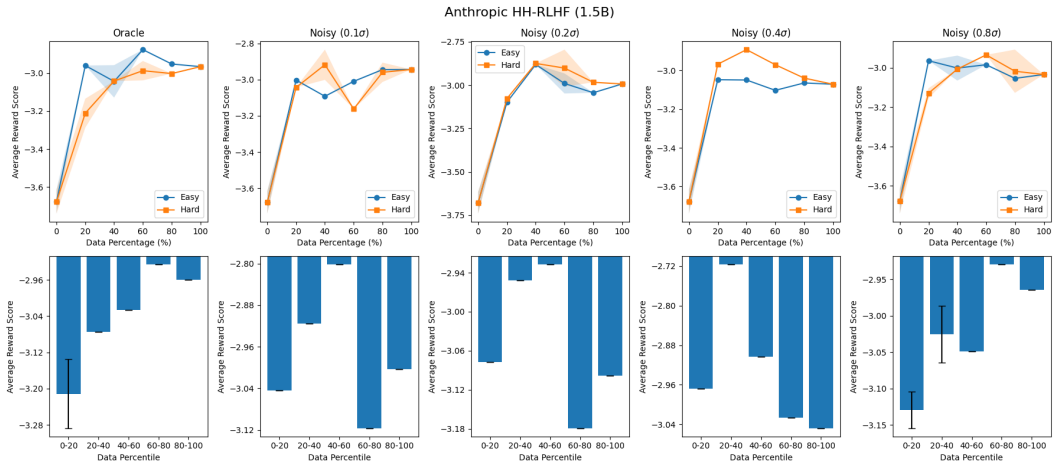


(b) Average reward score

Figure 40: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

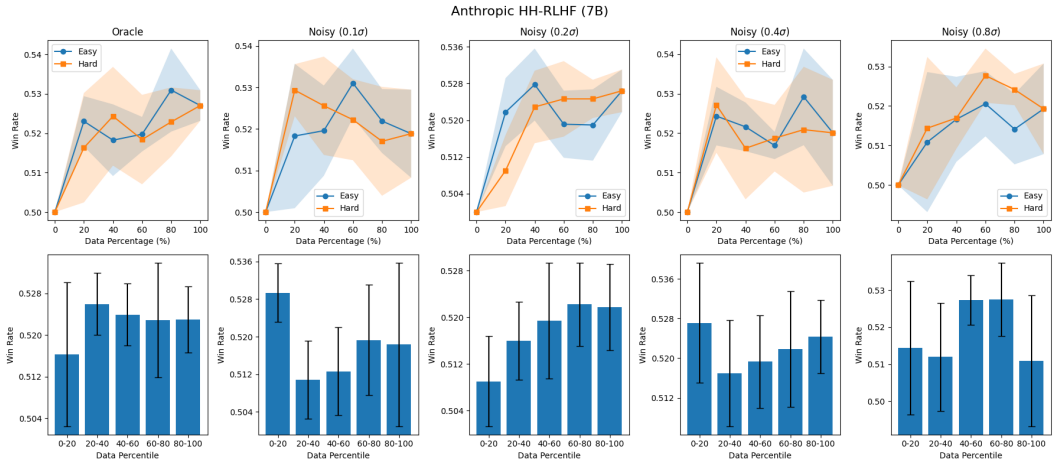


(a) Win rate

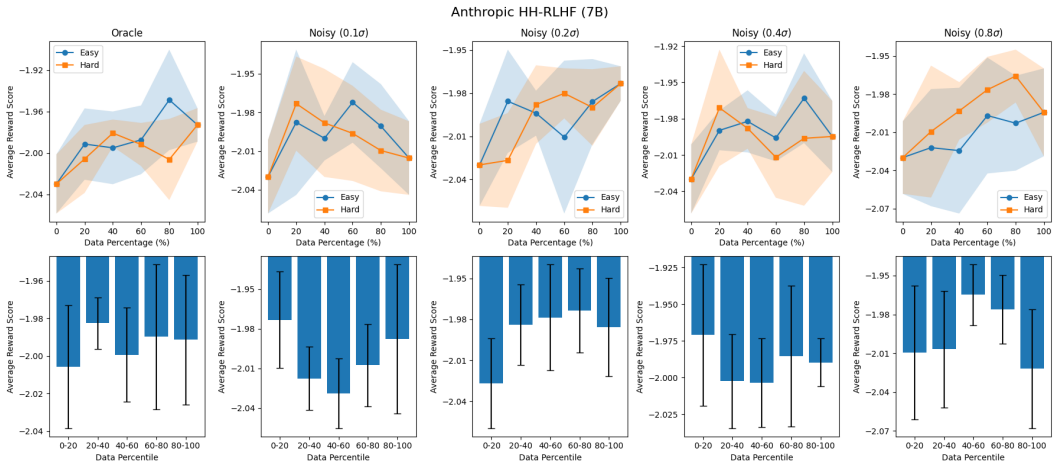


(b) Average reward score

Figure 41: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

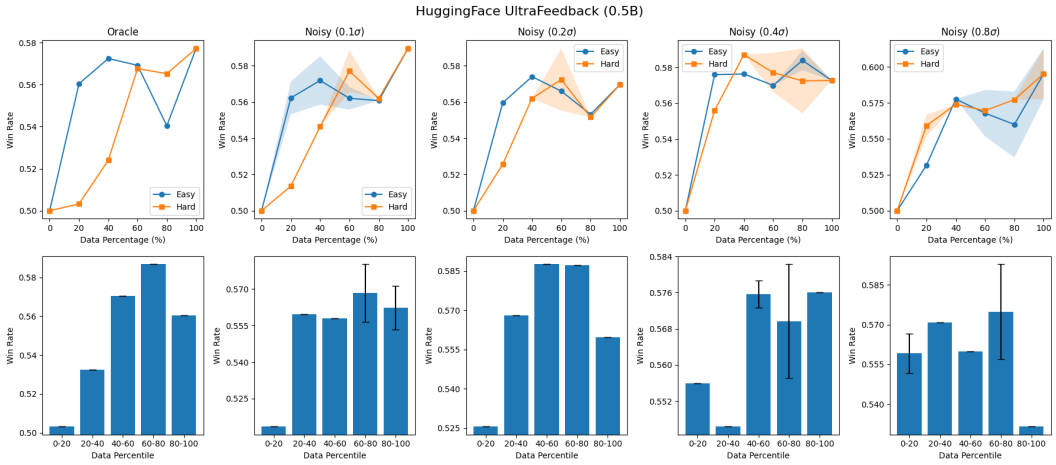


(a) Win rate

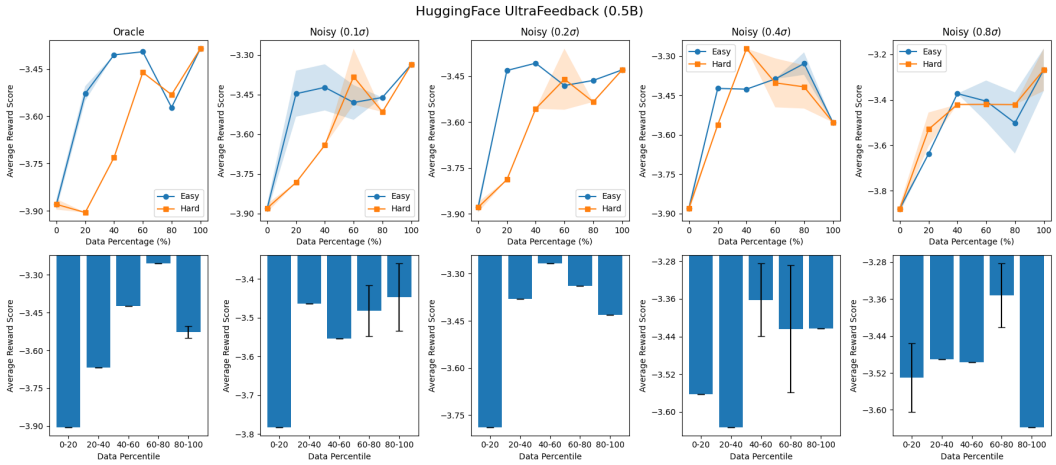


(b) Average reward score

Figure 42: GRPO win rate and average reward score on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

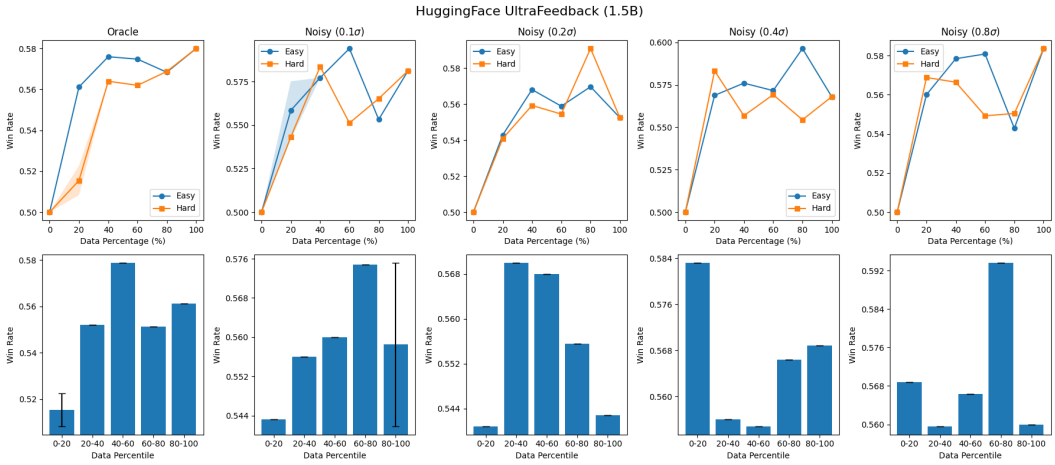


(a) Win rate

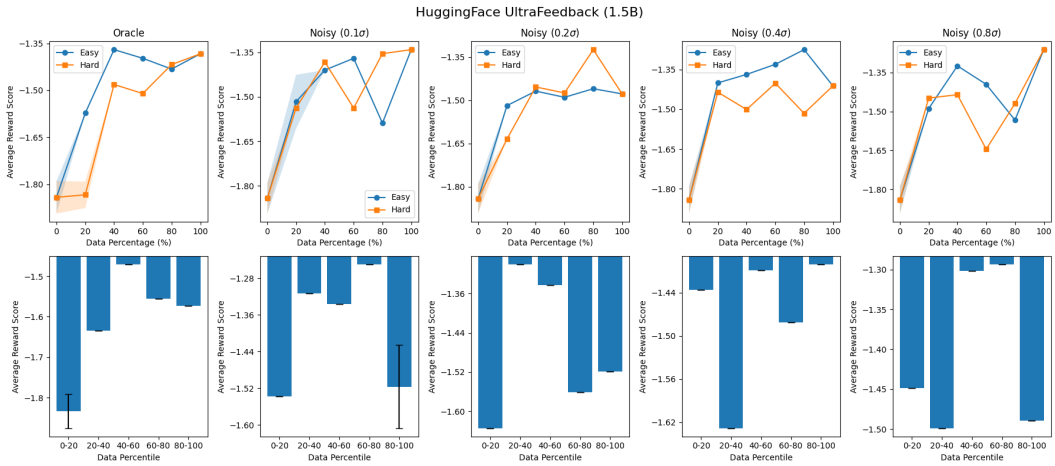


(b) Average reward score

Figure 43: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

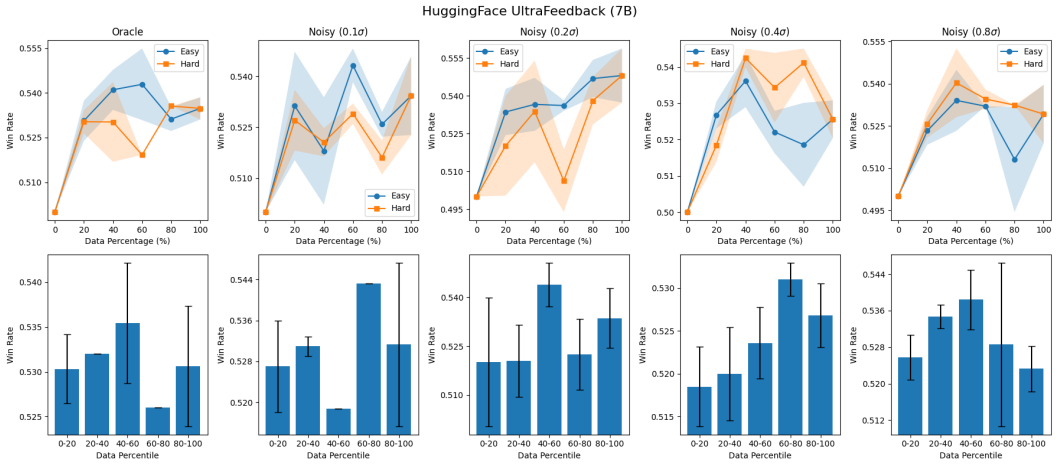


(a) Win rate

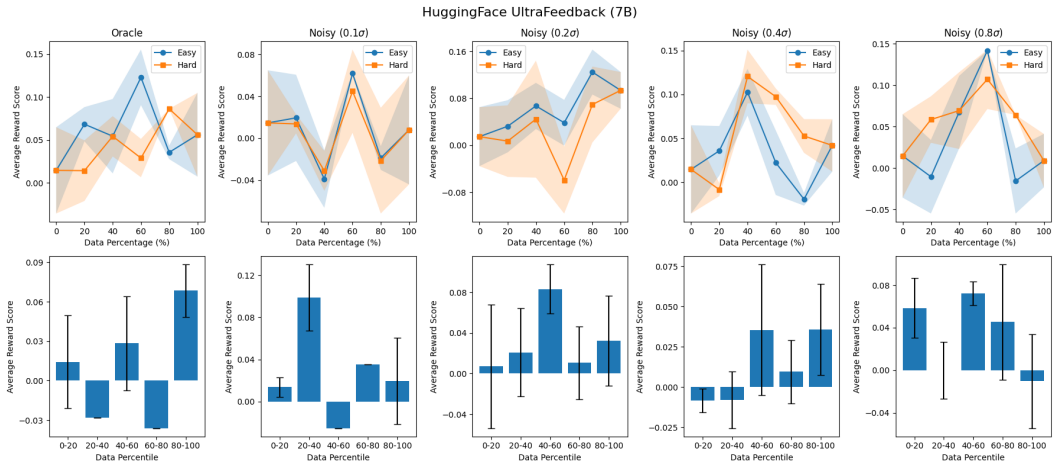


(b) Average reward score

Figure 44: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.



(a) Win rate



(b) Average reward score

Figure 45: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean ± standard deviation.

B.9 Theoretical Details

In this section, we provide formal statements for the signal-to-noise analysis summarized in Section 4.5. Note that the goal is not to exactly model the full nonconvex training dynamics of RM, DPO, or GRPO, but rather to formalize a simple mechanism: larger reference reward gaps are more robust to reward-estimation noise, and therefore yield more reliable pairwise update directions under a standard sign-gradient approximation.

B.9.1 Setup and Assumptions

Consider a single preference pair (x,y^w,y^l). Let r^* denote the fixed reference reward function and define the reference reward gap

$$\Delta^* = r^*(x,y^w) - r^*(x,y^l) > 0. \tag{2}$$

Here, larger values of Δ^* correspond to easier examples. In our study, the estimated gap used for difficulty-based selection is

$$\widehat{\Delta} = \Delta^* + \eta, \tag{3}$$

where η represents reward estimation error.

Pairwise reward model gradient. A reward model r_θ is trained using the Bradley-Terry loss

$$\mathcal{L}_{\mathrm{RM}}(\theta) = -\mathbb{E}[\log\sigma(\delta_\theta)], \qquad \delta_\theta := r_\theta(x,y^w) - r_\theta(x,y^l), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function. For a correctly ordered pair, the single-example gradient is

$$g^+(\theta) = -\big(1-\sigma(\delta_\theta)\big)\mathbf{h}, \qquad \mathbf{h} := \nabla_\theta \delta_\theta. \tag{5}$$

If the preference direction is flipped, the gradient becomes

$$g^-(\theta) = \sigma(\delta_\theta)\mathbf{h}. \tag{6}$$

Thus, a preference direction error reverses the update direction, but not necessarily its magnitude. In particular,

$$g^-(\theta) = -\frac{\sigma(\delta_\theta)}{1-\sigma(\delta_\theta)} g^+(\theta) = -e^{\delta_\theta} g^+(\theta). \tag{7}$$

Therefore, $g^-(\theta) = -g^+(\theta)$ holds exactly only at the preference boundary $\delta_\theta = 0$.

Near-boundary directional approximation. To obtain a transparent signal-to-noise calculation, we analyze the near-boundary regime in which $\delta_\theta \approx 0$. In this regime, $e^{\delta_\theta} \approx 1$, so the flipped and correctly ordered gradients have approximately equal magnitude and opposite direction:

$$g^-(\theta) \approx -g^+(\theta). \tag{8}$$

This approximation is exact at $\delta_\theta = 0$ and is most relevant early in training or for examples that the current model has not yet separated. More generally, even when the magnitudes differ, a preference direction error still reverses the pairwise update direction. Thus, the calculation below should be interpreted as a directional SNR analysis rather than an exact description of gradient magnitudes throughout training.

Assumption B.1 (Admissible noise distribution)**.** *The additive noise has a common continuous CDF F, independent of the clean reference gap, satisfying:*

- *(A1)* Symmetry: $F(-t) = 1 - F(t)$ *for all t, so* $F(0) = 1/2$.
- *(A2)* Strict monotonicity: F *is strictly increasing on* $\mathbb{R}$.
- *(A3)* Tail decay: $F(-t) \to 0$ *as* $t \to \infty$.

Examples include Gaussian, Laplace, logistic, and Student-t noise.

B.9.2 Preference Direction Flip Probability

Proposition B.2 (Preference direction flip probability)**.** *Under Assumption B.1, the probability that the estimated gap $\widehat{\Delta} = \Delta^* + \eta$ reverses the reference preference direction is*

$$p_{\mathrm{flip}}(\Delta^*) := \mathbb{P}(\widehat{\Delta} < 0 \,|\, \Delta^* > 0) = F(-\Delta^*). \tag{9}$$

Moreover, $p_{\mathrm{flip}}(\Delta^*)$ is strictly decreasing in Δ^*, with

$$p_{\mathrm{flip}}(\Delta^*) \to 0 \quad \textit{as} \quad \Delta^* \to \infty, \qquad p_{\mathrm{flip}}(\Delta^*) \to \frac{1}{2} \quad \textit{as} \quad \Delta^* \to 0^+.$$

Proof. By definition,

$$p_{\text{flip}}(\Delta^*) = \mathbb{P}(\Delta^* + \eta < 0) = \mathbb{P}(\eta < -\Delta^*) = F(-\Delta^*).$$

Since F is strictly increasing, $F(-\Delta^*)$ is strictly decreasing in Δ^*. Tail decay gives $F(-\Delta^*) \to 0$ as $\Delta^* \to \infty$. Symmetry and continuity give $F(-\Delta^*) \to F(0) = 1/2$ as $\Delta^* \to 0^+$. □

Interpretation. Proposition B.2 captures the main reason larger-gap examples are more reliable: they are farther from the preference boundary. Under symmetric reward estimation noise, a small-gap pair is more likely to have its estimated preference direction reversed, while a large-gap pair is more stable. Thus, the reference reward gap acts as a margin or confidence measure for the pairwise preference signal.

B.9.3 Gradient-Direction Signal-to-Noise Ratio

Theorem B.3 (SNR monotonicity)**.** *Under the near-boundary directional approximation and Assumption B.1, let*

$$g^*(\theta) := g^+(\theta)$$

denote the clean gradient direction for a correctly ordered preference pair. For an example with reference gap Δ^*, suppose the observed update is $+g^*(\theta)$ with probability $1 - p_{\text{flip}}(\Delta^*)$ and $-g^*(\theta)$ with probability $p_{\text{flip}}(\Delta^*)$. Define the gradient direction signal-to-noise ratio as

$$\text{SNR}(\Delta^*) := \frac{\|\mathbb{E}[g(\theta)]\|^2}{\mathbb{E}\left[\|g(\theta) - \mathbb{E}g(\theta)\|^2\right]}. \tag{10}$$

Then

$$\mathbb{E}[g(\theta)] = \big(1 - 2p_{\text{flip}}(\Delta^*)\big) g^*(\theta), \tag{11}$$

$$\text{SNR}(\Delta^*) = \frac{\big(1 - 2p_{\text{flip}}(\Delta^*)\big)^2}{4p_{\text{flip}}(\Delta^*)\big(1 - p_{\text{flip}}(\Delta^*)\big)}. \tag{12}$$

Furthermore, $\text{SNR}(\Delta^*)$ is strictly increasing in Δ^*. Consequently, for any fixed subset size K, the subset maximizing the sum of per-example directional SNR scores consists of the K examples with the largest reference reward gaps.

Proof. Let $p = p_{\text{flip}}(\Delta^*)$. Under the sign-gradient approximation,

$$g(\theta) = \begin{cases} +g^*(\theta), & \text{with probability } 1-p, \\ -g^*(\theta), & \text{with probability } p. \end{cases}$$

Therefore,

$$\mathbb{E}[g(\theta)] = (1-p)g^*(\theta) + p(-g^*(\theta)) = (1-2p)g^*(\theta).$$

Next,

$$\mathbb{E}\left[\|g(\theta)\|^2\right] = \|g^*(\theta)\|^2,$$

and hence

$$\mathbb{E}\left[\|g(\theta) - \mathbb{E}g(\theta)\|^2\right] = \mathbb{E}\left[\|g(\theta)\|^2\right] - \|\mathbb{E}g(\theta)\|^2 = 4p(1-p)\|g^*(\theta)\|^2.$$

Substituting into equation 10 gives

$$\text{SNR}(\Delta^*) = \frac{(1-2p)^2\|g^*(\theta)\|^2}{4p(1-p)\|g^*(\theta)\|^2} = \frac{(1-2p)^2}{4p(1-p)}.$$

To prove monotonicity, let $q = 1 - 2p \in [0,1)$. Then

$$4p(1-p) = 1 - q^2, \qquad \text{SNR} = \frac{q^2}{1-q^2}.$$

The function $q \mapsto q^2/(1-q^2)$ is strictly increasing for $q \in (0,1)$. By Proposition B.2, $p_{\text{flip}}(\Delta^*)$ is strictly decreasing in Δ^*, so $q = 1 - 2p_{\text{flip}}(\Delta^*)$ is strictly increasing in Δ^*. Thus, $\text{SNR}(\Delta^*)$ is strictly increasing in Δ^*.

Finally, for any subset S of size K, the subset score is

$$\sum_{i \in S} \text{SNR}(\Delta_i^*).$$

Since each term is strictly increasing in Δ_i^*, this sum is maximized by choosing the K examples with the largest reference gaps. □

Interpretation. Theorem B.3 formalizes the basic mechanism behind the advantage of easy examples. Larger reference reward gaps are less likely to have their preference direction corrupted, so their expected update is less attenuated and has higher directional SNR. In this sense, easy examples provide cleaner and more coherent pairwise update directions than hard examples. The fixed-size subset result shows that selecting the largest-gap examples is optimal for maximizing the total directional SNR under a reliable reference reward. Note that this sum is not generally the SNR of the aggregated mini-batch gradient, which also depends on gradient directions, magnitudes, and dependence.

Corollary B.4 (Noise sensitivity)**.** *Suppose the noise distribution belongs to a scale family $F_\sigma(t)=F(t/\sigma)$ with $\sigma>0$. Then, for every fixed finite $\Delta^*>0$:*

1. *$p_{\text{flip}}(\Delta^*;\sigma)=F(-\Delta^*/\sigma)$ is strictly increasing in σ;*
2. *$\mathrm{SNR}(\Delta^*;\sigma)$ is strictly decreasing in σ;*
3. *as $\sigma\to\infty$,*

$$p_{\text{flip}}(\Delta^*;\sigma)\to\frac{1}{2},\qquad \mathrm{SNR}(\Delta^*;\sigma)\to 0.$$

Consequently, over any bounded range of reference reward gaps, the SNR advantage of easy over hard examples vanishes as the noise scale grows.

Proof. For $\sigma_2>\sigma_1>0$ and fixed $\Delta^*>0$,

$$-\Delta^*/\sigma_2>-\Delta^*/\sigma_1.$$

Since F is strictly increasing,

$$F(-\Delta^*/\sigma_2)>F(-\Delta^*/\sigma_1),$$

so $p_{\text{flip}}(\Delta^*;\sigma)$ is strictly increasing in σ. Since the SNR in Theorem B.3 is decreasing in p_{flip} for $p_{\text{flip}}\in(0,1/2)$, the SNR is strictly decreasing in σ. Finally, as $\sigma\to\infty$, $-\Delta^*/\sigma\to 0$, so by continuity and symmetry,

$$F(-\Delta^*/\sigma)\to F(0)=1/2.$$

Substituting into equation 12 gives $\mathrm{SNR}(\Delta^*;\sigma)\to 0$. The bounded-range statement follows because this convergence is uniform over any compact interval $\Delta^*\in[0,M]$. □

Interpretation. Corollary B.4 explains why the empirical gap between easy and hard subsets shrinks as reward estimation noise increases. As the noise scale grows, even large reference gaps become harder to distinguish reliably, and the flip probability moves toward $1/2$. Consequently, the directional SNR of all examples decreases, and the absolute directional SNR difference between easy and hard examples vanishes. This matches the noisy-reference experiments, where easy examples remain beneficial under moderate noise but their advantage diminishes at larger noise levels.

B.9.4 Selection Failure Under Large Noise

Proposition B.5 (Selection failure under large noise)**.** *Let examples be ranked for selection by*

$$\widehat{\Delta}_i=\Delta_i^*+\eta_i,\qquad \eta_i\overset{\text{iid}}{\sim}F_\sigma,$$

where F_σ is a continuous scale family. For fixed reference gaps $\Delta_1^*,\dots,\Delta_N^*$ and fixed K, as $\sigma\to\infty$, the top-K subset selected by largest $\widehat{\Delta}_i$ converges in distribution to a uniformly random subset of size K. In particular, the selected subset becomes asymptotically independent of the reference gaps.

Proof. Write $\eta_i=\sigma Z_i$, where $Z_i\overset{\text{iid}}{\sim}F$. Then

$$\frac{\widehat{\Delta}_i}{\sigma}=\frac{\Delta_i^*}{\sigma}+Z_i.$$

For fixed Δ_i^*, $\Delta_i^*/\sigma\to 0$, so

$$\left(\frac{\widehat{\Delta}_1}{\sigma},\dots,\frac{\widehat{\Delta}_N}{\sigma}\right)\Rightarrow(Z_1,\dots,Z_N).$$

Scaling by the positive constant σ does not change the ranking, so the ranking of $\widehat{\Delta}_1,\dots,\widehat{\Delta}_N$ converges in distribution to the ranking of i.i.d. continuous random variables $Z_1,\dots,Z_N$. Since the Z_i are exchangeable and ties occur with probability zero, every permutation of the N examples is equally likely. Therefore, the top-K subset converges to a uniformly random subset among all $\binom{N}{K}$ subsets, independent of $(\Delta_1^*,\dots,\Delta_N^*)$. □

Interpretation. Proposition B.5 concerns a different failure mode from the flip-probability analysis. Even if the training labels are not explicitly flipped, large estimation noise can make the ranking induced by $\widehat{\Delta}$ nearly independent of the ranking induced by the reference gaps Δ^*. In this regime, selecting the top-K examples according to the proxy no longer preferentially selects high-SNR examples; it behaves like random subset selection. This provides a theoretical explanation for why difficulty-based filtering requires reliable reward gap estimates.

Remark B.6 (Implication for weak proxy RMs)**.** *Proposition B.5 formalizes the limiting case in which the selector RM is so noisy that its estimated gaps no longer preserve the ordering induced by the reference RM. In such regimes, the subset labeled "easy" by the proxy need not contain the examples with the highest reference-gap SNR. This limit explains loss of a ranking advantage under the stated noise model; systematic reversal under structured learned-proxy errors is an empirical observation beyond this result.*

B.9.5 Connection to DPO

While the preceding analysis pertains to the Bradley-Terry loss, the same directional argument also applies to DPO. Given a policy π_θ and reference policy π_{ref}, DPO uses the policy-dependent implicit reward gap

$$\Delta_\theta^{\mathrm{DPO}} = \beta\left(\log\frac{\pi_\theta(y^w\,|\,x)}{\pi_{\mathrm{ref}}(y^w\,|\,x)} - \log\frac{\pi_\theta(y^l\,|\,x)}{\pi_{\mathrm{ref}}(y^l\,|\,x)}\right). \tag{13}$$

For a correctly ordered pair (y^w, y^l), the single-example DPO gradient has the form

$$g_{\mathrm{DPO}}^{+}(\theta) = -\beta\big(1-\sigma(\Delta_\theta^{\mathrm{DPO}})\big)\big[\nabla_\theta \log\pi_\theta(y^w\,|\,x) - \nabla_\theta\log\pi_\theta(y^l\,|\,x)\big]. \tag{14}$$

If the preference direction is flipped, DPO treats y^l as preferred and y^w as rejected. The corresponding gradient becomes

$$g_{\mathrm{DPO}}^{-}(\theta) = \beta\sigma(\Delta_\theta^{\mathrm{DPO}})\big[\nabla_\theta \log\pi_\theta(y^w\,|\,x) - \nabla_\theta\log\pi_\theta(y^l\,|\,x)\big]. \tag{15}$$

Thus, a preference direction error reverses the pairwise update direction: the correctly ordered gradient is proportional to the negative of the log-probability contrast, whereas the flipped gradient is proportional to the positive contrast. As in RM training, however, the magnitudes are not generally equal. Comparing equation 14 and equation 15, we have

$$g_{\mathrm{DPO}}^{-}(\theta) = -\frac{\sigma(\Delta_\theta^{\mathrm{DPO}})}{1-\sigma(\Delta_\theta^{\mathrm{DPO}})}\, g_{\mathrm{DPO}}^{+}(\theta) = -e^{\Delta_\theta^{\mathrm{DPO}}}\, g_{\mathrm{DPO}}^{+}(\theta). \tag{16}$$

Therefore, $g_{\mathrm{DPO}}^{-}(\theta) = -g_{\mathrm{DPO}}^{+}(\theta)$ holds exactly only at the DPO preference boundary $\Delta_\theta^{\mathrm{DPO}} = 0$. In the near-boundary regime $\Delta_\theta^{\mathrm{DPO}} \approx 0$, we have $e^{\Delta_\theta^{\mathrm{DPO}}} \approx 1$, and hence

$$g_{\mathrm{DPO}}^{-}(\theta) \approx -g_{\mathrm{DPO}}^{+}(\theta). \tag{17}$$

This shows that DPO has the same directional structure as in the RM case: a preference direction error pushes the pairwise update in the opposite direction, and this opposite update has approximately the same magnitude near the preference boundary. The key difference from RM training is that the margin $\Delta_\theta^{\mathrm{DPO}}$ is policy-dependent and evolves during training. Therefore, conditioning on the current policy parameters, the same directional reliability argument applies: larger reference reward gaps are less likely to suffer preference direction errors under noisy reward estimation, and therefore yield more reliable pairwise update directions.

Interpretation. The DPO calculation shows that the same directional reliability idea is not specific to explicit RM training. DPO also uses a pairwise preference loss, so an error in the preference direction pushes the update toward increasing the rejected response relative to the chosen response. The main difference is that the DPO margin is policy-dependent, so the exact gradient magnitude changes during training. Nevertheless, the stability of the pairwise update direction is still controlled by the reliability of the underlying preference ordering. Thus, larger reference gaps can be understood as providing more stable DPO update directions, especially in early or near-boundary regimes.

B.9.6 Empirical Preference-Flip Measurements

We compare analytic and empirical flip rates over 50 independent Gaussian-noise draws on Anthropic and LMSYS. Quintiles in the gap-stratified comparison use clean reference gaps. These measurements test synthetic-noise preference stability; they do not directly measure learned-proxy flip rates or gradients throughout training.

Table 10: Overall analytic and empirical flip probabilities. Empirical results use 50 independent noise draws.

Dataset	Calculation	0.1σ	0.2σ	0.4σ	0.8σ
Anthropic	Analytic	.0468	.0917	.1688	.2720
Anthropic	Empirical	.0469	.0919	.1684	.2722
LMSYS	Analytic	.0518	.1011	.1804	.2800
LMSYS	Empirical	.0517	.1011	.1808	.2794

Table 11: Empirical flip rates at 0.4σ by clean-reference-gap quintile.

Dataset	Hardest quintile	Easiest quintile
Anthropic	41.7%	0.27%
LMSYS	42.7%	0.28%

C *Data Selection with Weak Proxy RMs* Details

C.1 Proxy RM Accuracy

Figure 46 summarizes the test set accuracy of the Pythia-1B proxy RM for each dataset and reference RM.

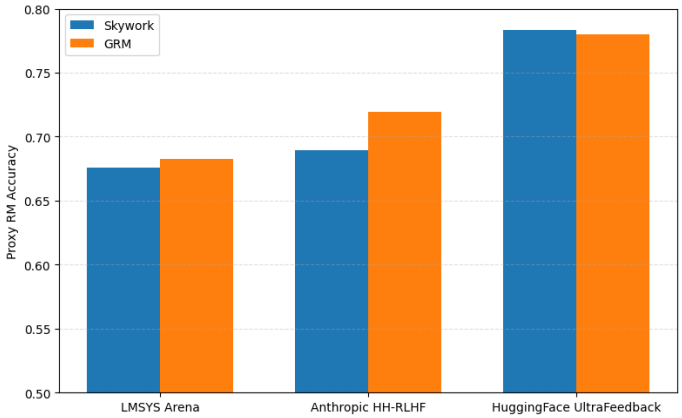
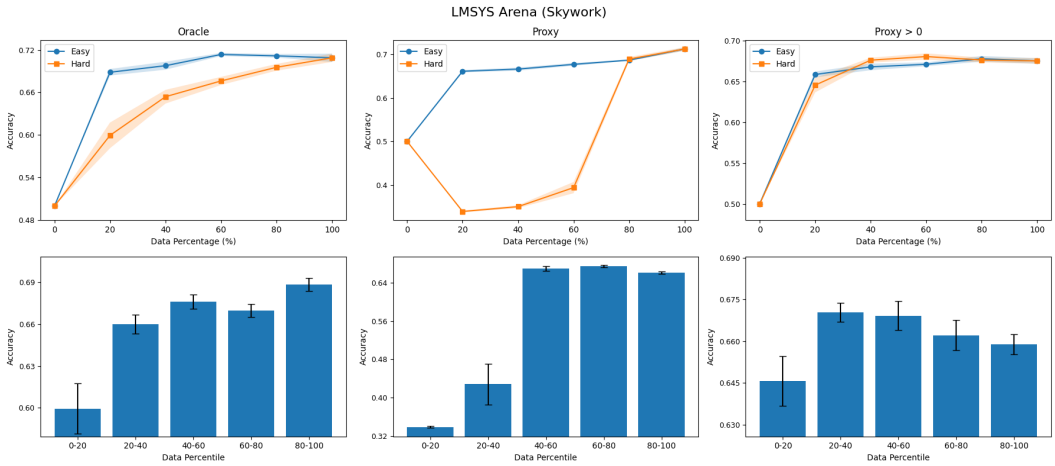


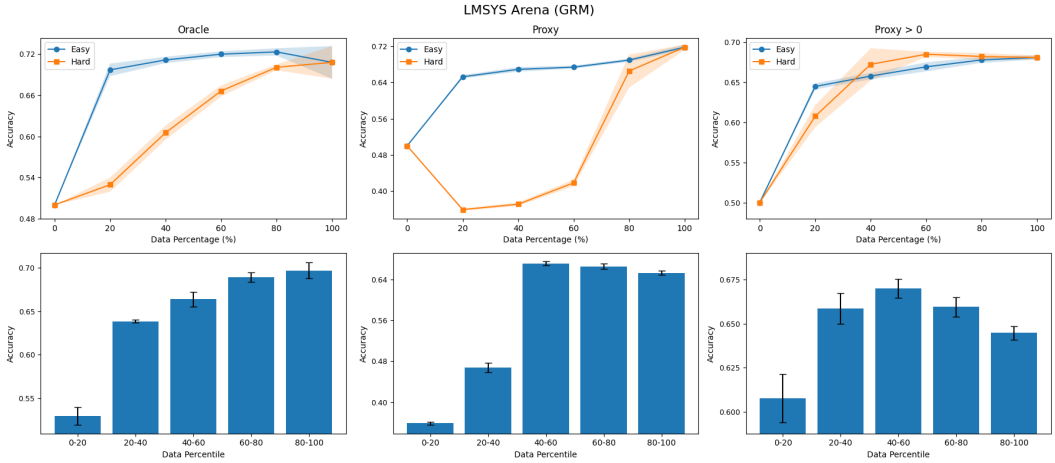
Figure 46: Test set accuracy of the Pythia-1B proxy RM across datasets and reference RMs. The proxy achieves 67% to 78% accuracy depending on dataset and reference. Here, Skywork refers to Skywork-Reward-V2-Qwen3-4B and GRM refers to GRM-gemma2-2B-rewardmodel-ft.

C.2 RM Training Results (Learned Proxy)

Figures 47-49 show RM accuracy across datasets, difficulty-based subsets and training setups under the learned proxy setting. For a discussion and interpretation of these results, see Section 5.1.

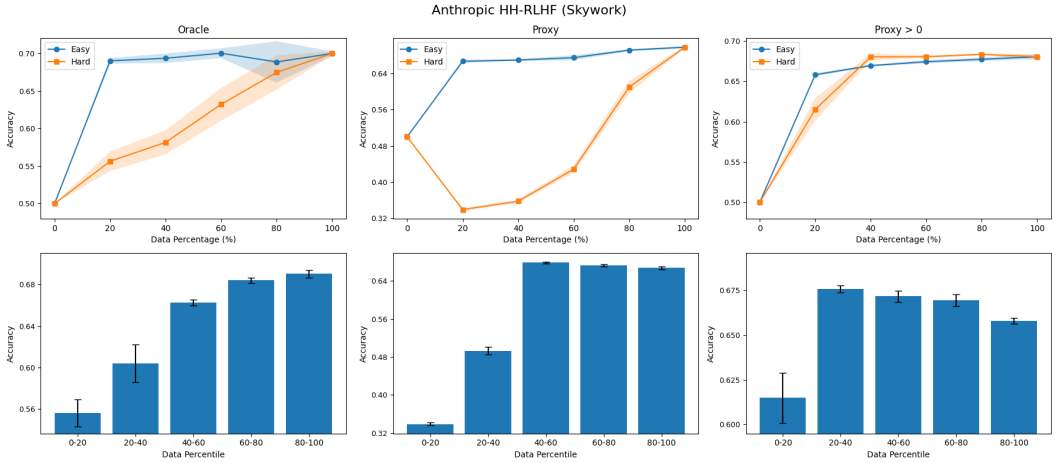


(a) Skywork reward model

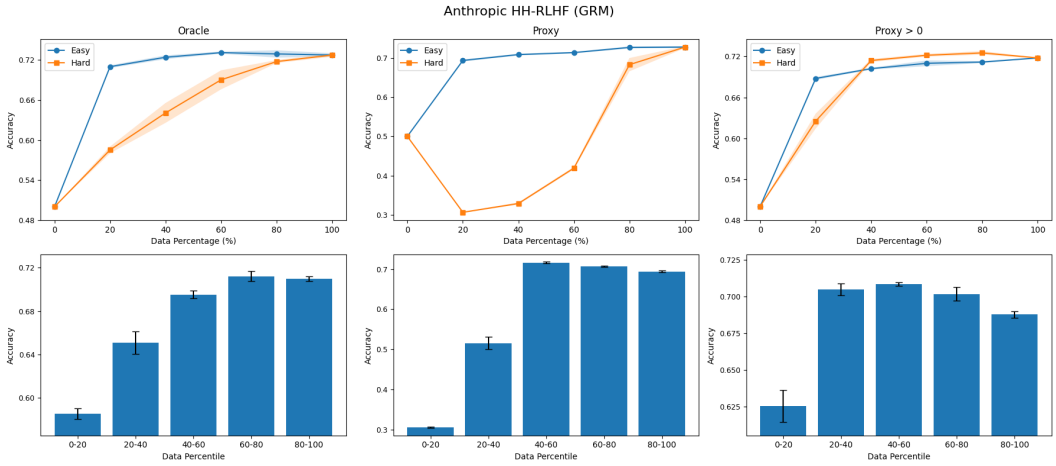


(b) GRM reward model

Figure 47: RM accuracy on the LMSYS Arena dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

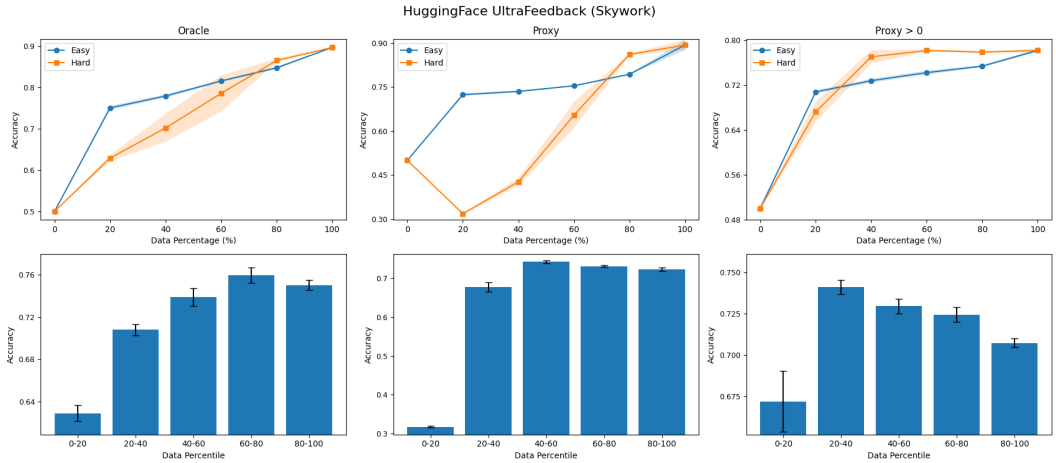


(a) Skywork reward model

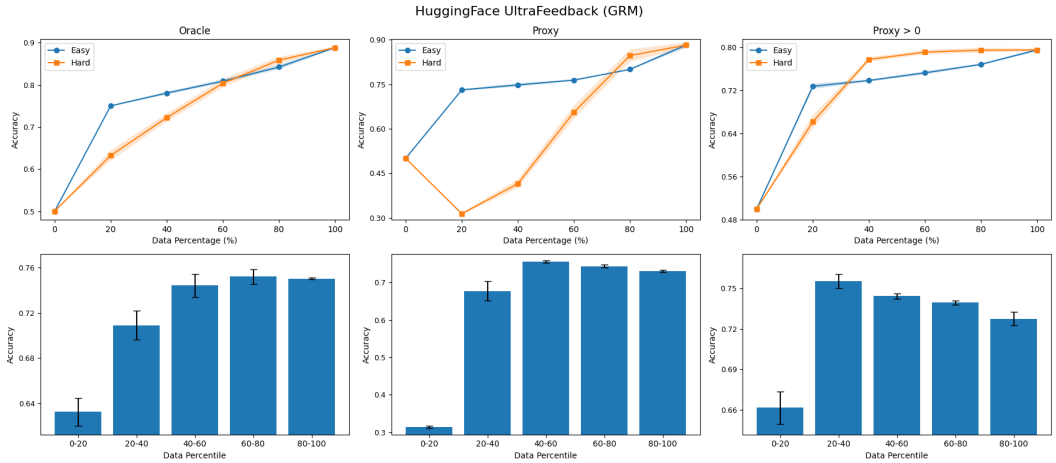


(b) GRM reward model

Figure 48: RM accuracy on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Skywork reward model

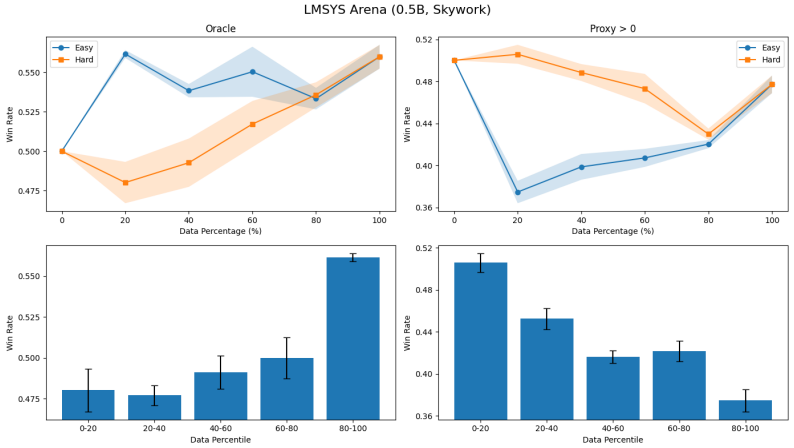


(b) GRM reward model

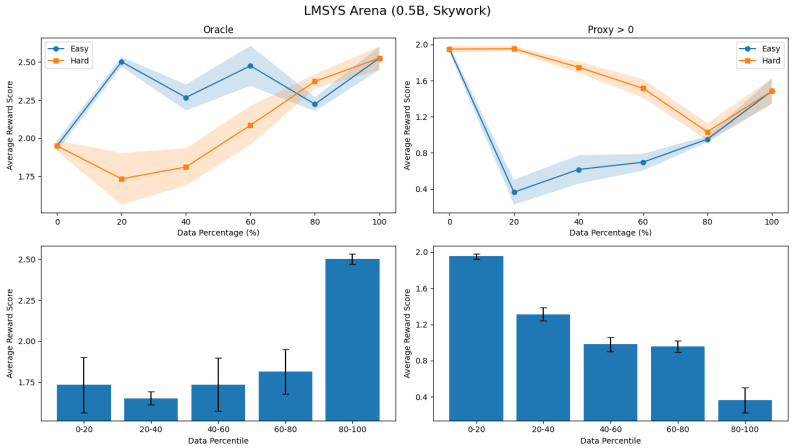
Figure 49: RM accuracy on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM reference RMs, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

C.3 DPO and GRPO Training Results (Learned Proxy, LMSYS Arena, Skywork)

Figures 50-55 show DPO and GRPO win rates and average reward scores across difficulty-based subsets and training setups under the learned proxy setting. For a discussion and interpretation of these results, see Section 5.2.

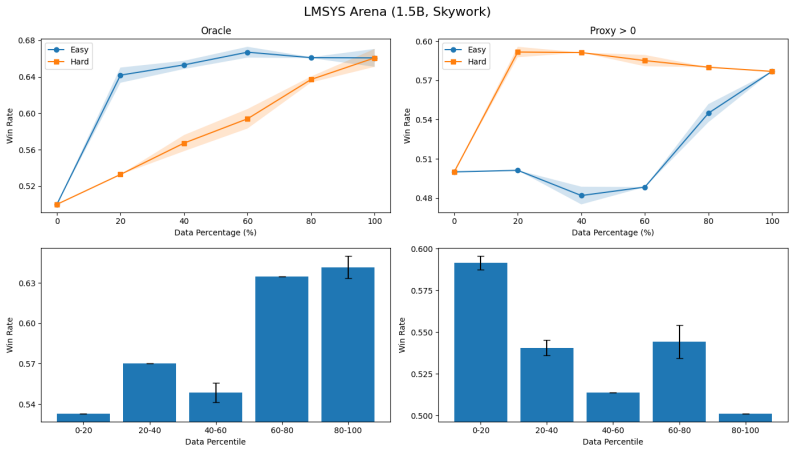


(a) Win rate

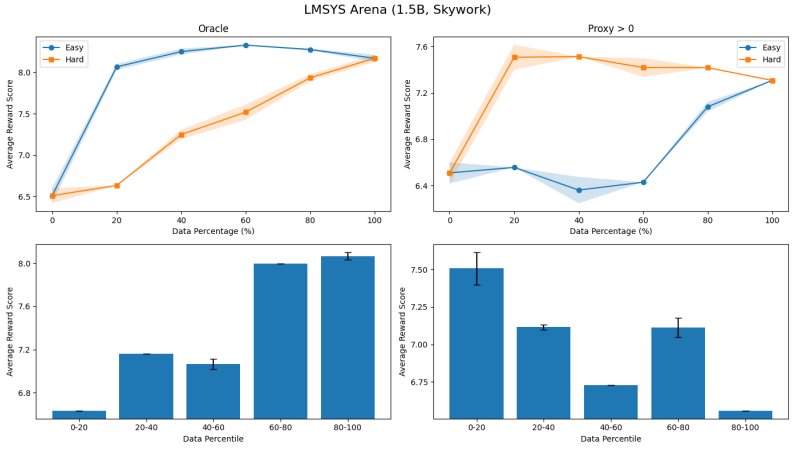


(b) Average reward score

Figure 50: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

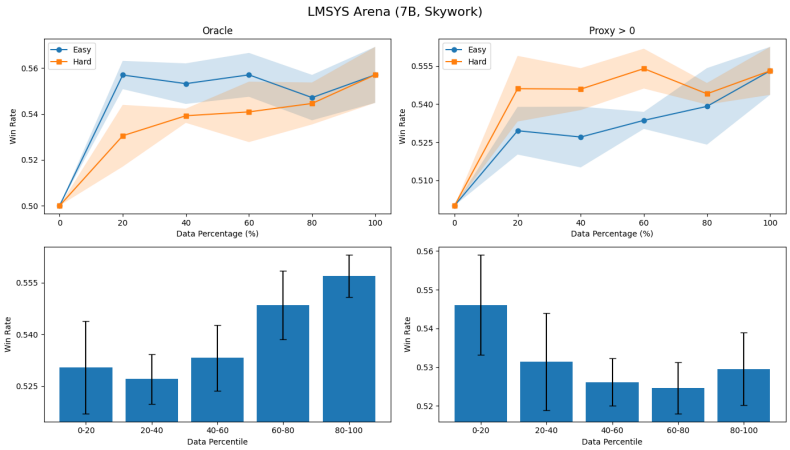


(a) Win rate

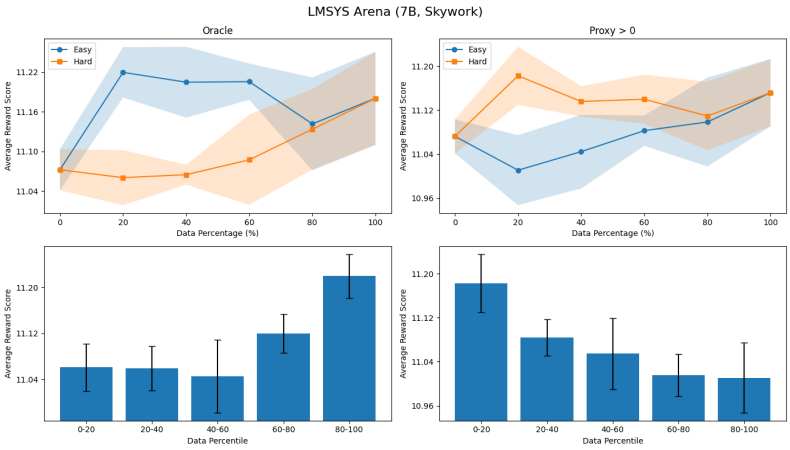


(b) Average reward score

Figure 51: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

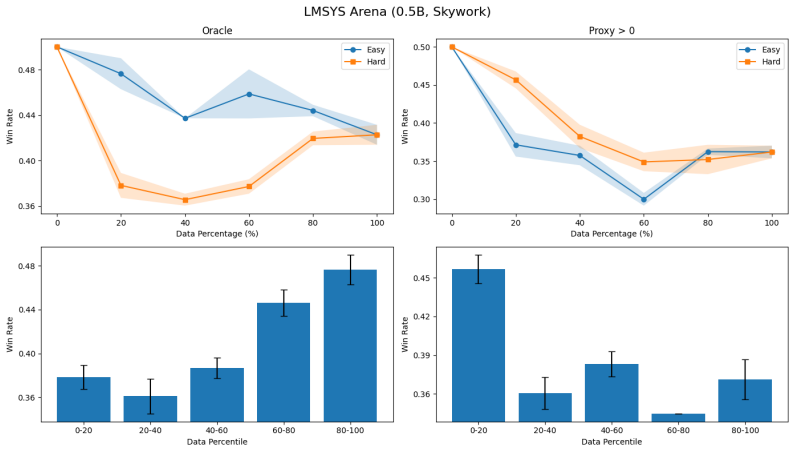


(a) Win rate

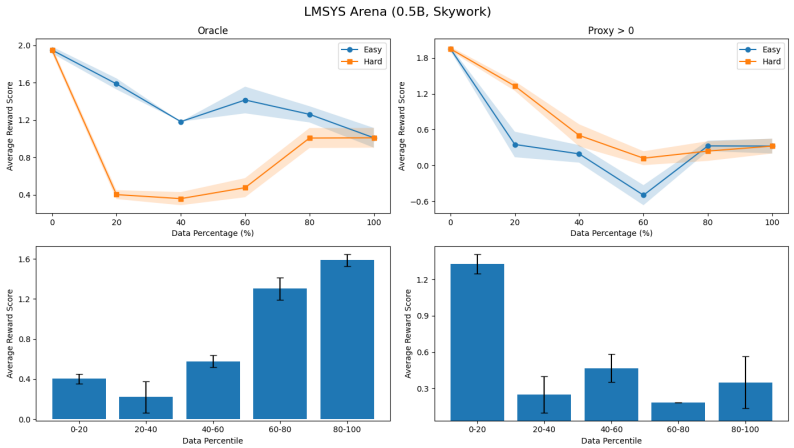


(b) Average reward score

Figure 52: DPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

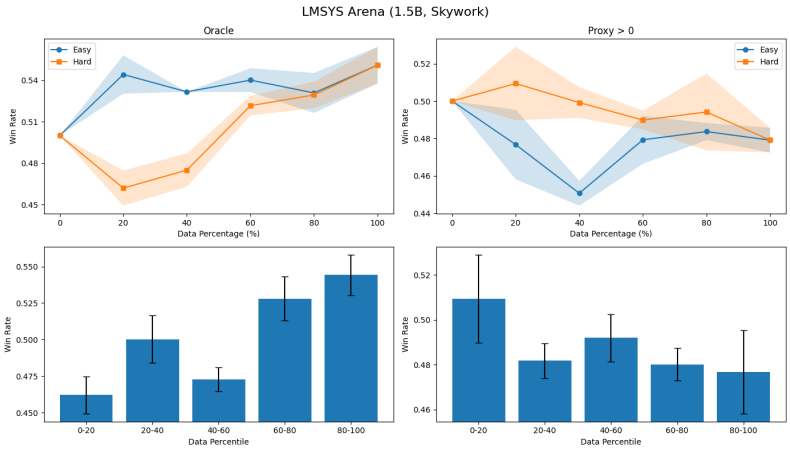


(a) Win rate

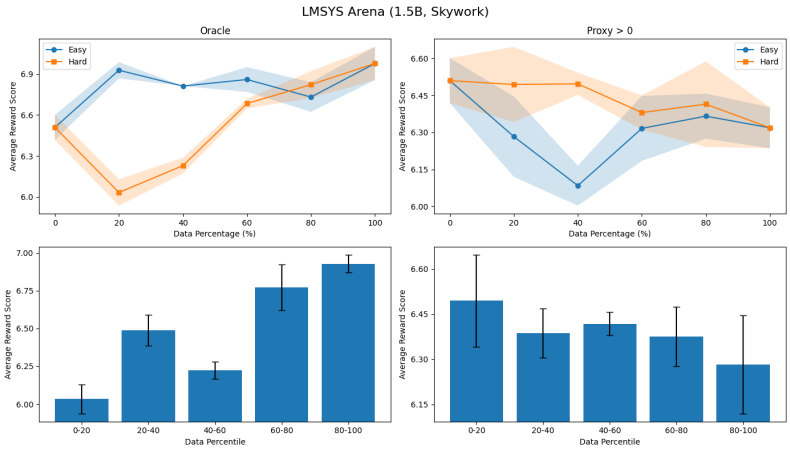


(b) Average reward score

Figure 53: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-0.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

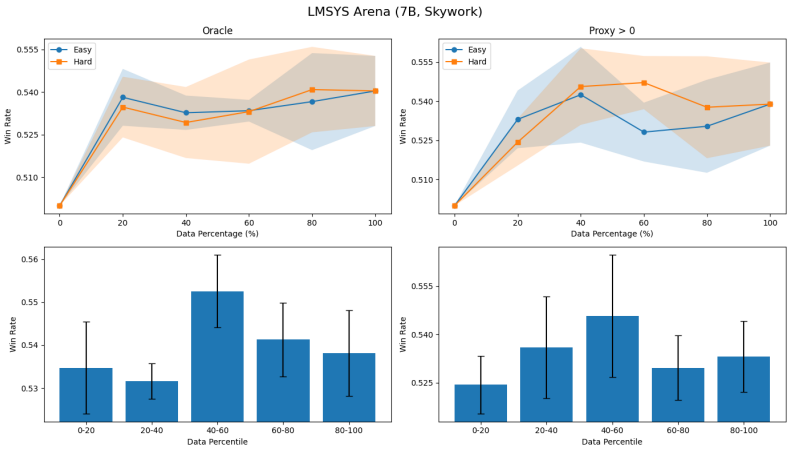


(a) Win rate

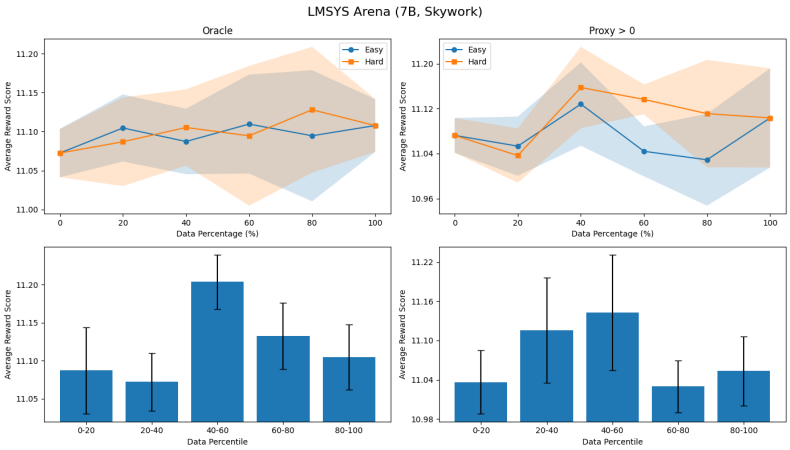


(b) Average reward score

Figure 54: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.



(a) Win rate



(b) Average reward score

Figure 55: GRPO win rate and average reward score on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork reference RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

D Existing Assets and Licenses

We use only existing publicly available datasets, models, and evaluation benchmarks. The preference datasets used in our experiments are LMSYS Arena [6], Anthropic HH-RLHF [2], and UltraFeedback [8]. The base and reward models are Pythia-1B [3], Qwen2.5-Instruct [34], Skywork-Reward-V2-Qwen3-4B [20], and GRM-gemma2-2B-rewardmodel-ft [35]. We also use RewardBench [16] and AlpacaEval [10] for reporting and comparison. We cite the original creators of all assets and use them according to their stated licenses and terms of use. Table 12 summarizes the assets used in this work.

Table 12: Existing assets used in this paper. We cite the original creators and use each asset according to its stated license and terms of use.

Asset	Use in this paper	Stated license
LMSYS Arena	Preference dataset	CC-BY-4.0, CC-BY-NC-4.0
Anthropic HH-RLHF	Preference dataset	MIT
UltraFeedback	Preference dataset	MIT
Pythia-1B	Reward model backbone	Apache-2.0
Qwen2.5-Instruct	Policy backbone for DPO/GRPO	Apache-2.0
Skywork-Reward-V2-Qwen3-4B	Reference reward model	Apache-2.0
GRM-gemma2-2B-rewardmodel-ft	Reference reward model	Apache-2.0
RewardBench	Reward model benchmark/reference	Mixed licenses across subsets
AlpacaEval	Oracle-independent evaluation	Apache-2.0

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state conditional findings involving reward reliability, target alignment, and model scale. They distinguish RM and smaller-model DPO results from the more variable downstream GRPO evidence and qualify the directional theoretical explanation. The abstract and introduction state the main contributions: studying difficulty-based data selection under a reference RM, showing that easier examples are more effective when reward gap estimates are reliable, and explaining conflicting prior results through reward reliability and signal mismatch. The claims are matched to the empirical results in Sections 4-5 and the theoretical explanation in Section 4.5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper includes a limitation discussion in Section 6. It explicitly notes the use of fixed reference RMs, the gap between the controlled reference setting and real-world noisy or mismatched reward signals, and the focus on open-ended preference settings with continuous reward scores. The discussion also identifies the model scale boundary, reference-target disagreement, and the limited independent evaluation coverage.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important

role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Section 4.5 states the main signal-to-noise intuition and references the formal theoretical results in Appendix B.9. The appendix provides the assumptions and proofs for the propositions, theorem, and corollary supporting the directional signal-to-noise explanation.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Sections 3 and 4 describe the datasets, reference reward models, preprocessing, relabeling, data splits, difficulty-based subset construction, training protocols, evaluation metrics, and noisy/proxy difficulty setups. Additional experimental details and full results are provided in the appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The datasets and models used in our experiments are publicly available, and their sources are cited in the paper. Experiment code is not publicly available. However, the manuscript provides detailed descriptions of the data processing, difficulty-based selection procedure, training setups, hyperparameters, and evaluation protocols, which should allow researchers to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper specifies the datasets, preprocessing rules, data splits, reward-gap construction, subset definitions, model families, learning rates, number of epochs, DPO/GRPO regularization parameters, generation temperatures, and evaluation metrics in Sections 3-5 and the appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The main figures report results averaged over five runs, with error bars defined as mean $\pm$ standard deviation. The text accompanying the experimental sections states how these error bars are computed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: The paper provides the main training configurations needed to understand the experiments, including model sizes, number of epochs, learning rates, subset sizes, and evaluation protocols. However, the current version does not provide full compute resource details such as GPU type, memory requirements, per-run execution time, total compute, or compute spent on preliminary experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research uses existing public datasets and models for empirical analysis of data selection in preference alignment. To the best of our knowledge, the work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A].

Justification: The work is a standard methodological study of data selection for LLM preference alignment and does not introduce a deployed system, new dataset, or new model release. We therefore do not identify direct societal impacts beyond the general implications of improving alignment efficiency.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not introduce or release a new high-risk pretrained model, image generator, or scraped dataset. The experiments use existing datasets and model families to study data selection behavior.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper cites the original creators of the datasets, models, and evaluation benchmarks used in the experiments. Appendix D summarizes the stated licenses and terms of use for these existing assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce a new dataset, benchmark, or released model asset. The contribution is an empirical and theoretical study of difficulty-based data selection using existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not conduct new crowdsourcing experiments or new human-subject studies. It uses existing preference datasets released by prior work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve new human-subject data collection or interaction with study participants. Therefore, IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.