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 examining data selection for reward model (RM) training and policy optimization methods (e.g. Direct Preference Optimization (DPO) and Group Relative Policy Optimization (GRPO)) has identified *example difficulty*, measured by the reward gap between chosen and rejected responses, as a key factor. However, findings are contradictory: some studies favor easier examples with larger gaps, while others prefer harder ones. To isolate the role of difficulty from confounding factors, we *assume access to an oracle RM* and systematically study data selection across RM, DPO, and GRPO training. We find that *training on easier pairs consistently leads to better performance* than harder ones, particularly for smaller base models. This advantage persists even when reward estimates are noisy. Notably, using only the top 20% easiest examples often matches or exceeds full-dataset performance while reducing post-training costs by 5×.

1 INTRODUCTION

As large language models (LLMs) become increasingly capable and are applied across a wider range of tasks, guiding their behavior to align with human values and 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 and potential for real-world deployment (Shen et al., 2023; Wang et al., 2023).

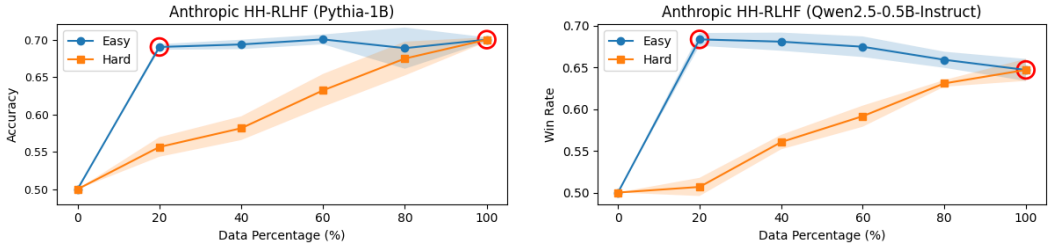
To address these challenges, numerous techniques have been proposed for aligning LLM behavior with human preferences (Zhang & Ranganath, 2025). Reinforcement Learning from Human Feedback (RLHF; Christiano et al. (2017); Ziegler et al. (2019)) is a widely used approach. In RLHF, a reward model (RM) is 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 to RLHF, Direct Preference Optimization (DPO; Rafailov et al. (2023)) directly updates the LLM to match human preference comparisons, streamlining the alignment process by eliminating the need for an explicit RM. Many extensions and variants have been proposed to improve preference alignment, including Sequence Likelihood Calibration with Human Feedback (SLiC-HF; Zhao et al. (2023)), Kahneman-Tversky Optimization (KTO; Ethayarajh et al. (2024)), Group Relative Policy Optimization (GRPO; Shao et al. (2024)), and stepwise DPO (sDPO; Kim et al. (2025)).

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

[*]Work done during an internship at Prescient Design, Genentech

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 RewardBench refer to the datasets introduced in Dubois et al. (2024) and Lambert et al. (2025).

Author(s)	Method(s)	Data	Reward calculation	Evaluation/Benchmark	Recommendation
Gao et al. (2025)	DPO	Conv.	Estimated (trained DPO)	AlpacaEval WR	Use easy
Deng et al. (2025)	DPO	Conv.	Estimated (trained DPO, ext. rewards)	AlpacaEval WR	Use easy
Qi et al. (2025)	RM, DPO	Conv.	Estimated (fixed models)	RewardBench acc., AlpacaEval WR	Use hard
Shen et al. (2024)	RM	Conv.	Response embedding similarity	Test acc.	Inconclusive
Pikus et al. (2025)	GRPO	Math	Verifiable (0 or 1)	Test acc.	Use hard
Ours	**RM, DPO, GRPO**	**Conv.**	**Known, estimated**	**Test acc., test WR (via oracle)**	**Use easy**



(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 reducing post-training compute and time by roughly a factor of 5. 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.

Previous work (summarized in Table 1) has investigated data selection in the context of RM training and policy optimization, focusing on either reinforcement learning or preference-driven approaches. There is broad consensus that *example difficulty*, typically assessed via an explicit or implicit estimation of the reward gap, plays a central role in guiding data selection (e.g., Deng et al. (2025); Gao et al. (2025); Qi et al. (2025)). Interestingly, while some studies recommend focusing on easier examples with larger reward gaps (e.g., Gao et al. (2025)), others find that more challenging examples with smaller gaps are more informative (e.g., Qi et al. (2025)). This disagreement may reflect differences in how reward gaps are defined and measured across studies, leaving it unclear how to select data in practice.

This paper adopts a new perspective on the role of data selection by *assuming access to an oracle RM*, which serves as a surrogate for the true underlying reward or preference signal and removes the confounding effects introduced by reward gap estimation in prior work. This setup allows us to isolate the intrinsic effect of difficulty from the noise introduced by imperfect reward estimation, establishing what holds under ideal conditions before examining how conclusions change when difficulty must be estimated from noisy proxies. In doing so, we directly address the source of disagreement in the existing literature.

Under this setup, we analyze whether data selection continues to play a significant role and whether easier or harder examples are more beneficial. 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 consistently more beneficial* under this setup, particularly for smaller base models. Moreover, this effect persists even when rewards are observed through noisy proxies formed by adding Gaussian noise to the oracle rewards, although the advantage of easier examples diminishes as the noise level increases. Importantly, this finding has significant *practical implications*: beyond reducing post-training compute by $5\times$, requiring only 20% of the data substantially improves efficiency and enables faster iteration during model development.

Our main contributions are as follows: (1) We systematically study data selection for preference alignment by assuming access to an oracle RM, isolating the impact of example difficulty from errors introduced by reward gap estimation; (2) We demonstrate that, when difficulty is assessed using the oracle RM, easier examples consistently yield superior performance across RM training and policy

optimization methods (DPO and GRPO), especially when using smaller base models; and (3) We reconcile conflicting prior findings: the benefit of easier examples is clear with oracle rewards but diminishes under noisy reward estimation, thus accounting for disagreements in previous studies.

The paper is organized as follows. Section 2 positions our work with respect to prior studies on data selection for preference alignment. Section 3 provides a high-level overview of RM, DPO, and GRPO, followed by data processing, difficulty-based data selection, as well as training and evaluation protocol. Section 4 describes the experimental setup and presents data selection results for RM, DPO, and GRPO under oracle-based difficulty measures, including noisy variants. Section 5 investigates data selection when difficulty is estimated using a weak proxy RM trained to approximate the oracle rewards. Finally, Section 6 concludes the paper.

2 Related Work

Prior work has examined data selection and preference alignment across supervised fine-tuning and reinforcement learning settings (e.g., Zhou et al. (2023); Liu et al. (2024a); Cao et al. (2024); Xia et al. (2024); Liu et al. (2024b); Shen et al. (2024); Yu et al. (2024); Li et al. (2024); Wu et al. (2024); Morimura et al. (2024); Khaki et al. (2024); Ankner et al. (2025); Yu et al. (2025); Gao et al. (2025); Deng et al. (2025); Qi et al. (2025); Pikus et al. (2025)). In this section, we focus on five representative studies most closely related to our setup. We review these works along several key aspects: methods, datasets, reward calculations, evaluation metrics, and conclusions, and then position our work in comparison to existing approaches. Refer to Table 1 for a concise overview of these studies alongside our contributions.

Gao et al. (2025) studies data selection for DPO training. They argue that overly difficult examples hinder alignment and propose retaining the easiest 50% of the data (*Selective DPO*). 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 (larger gaps indicate easier examples). They also find that larger models can tolerate a higher proportion of difficult examples. Deng et al. (2025) explores data selection for DPO training. They examine two types of reward margins (external and implicit DPO margins) and find that even strong RMs often disagree on margin estimates. Based on this, they propose *Bayesian Aggregation for Preference Data Selection* (BeeS), which aggregates signals from multiple reward sources and includes an example only when it achieves high reward margins across all sources. They show that BeeS improves alignment while using much less data.

Qi et al. (2025) investigates data selection for RM and DPO training. They measure example difficulty using the DPO implicit reward gap between preferred and rejected responses, computed using fixed Llama-3-based models (Grattafiori et al., 2024). They find that selecting the 10-15% most difficult examples yields the best performance. Shen et al. (2024) examines data selection for RM training. They study how the content of examples affects learning by selecting only samples whose preferred and rejected responses are sufficiently distinct, measured via response embedding similarity. Their results are inconclusive. Pikus et al. (2025) evaluates data selection for GRPO training on tasks with well-defined correctness, such as math problems. They estimate example difficulty using the base model's success rate across multiple completions and find that training on the hardest 10% of examples yields dramatic performance gains.

Our work differs from these studies in two important ways. First, we explicitly *assume access to an oracle RM*, which provides a reliable measure of example difficulty and eliminates confounding errors from proxy RMs. This allows us not only to study the intrinsic impact of difficulty on alignment performance, but also to perform evaluation directly against the oracle. Second, we provide a *unified comparison* across reward-based (RM, GRPO) and preference-based (DPO) optimization methods, which allows us to identify consistent trends in the role of example difficulty and assess how data selection strategies generalize across different model sizes and alignment approaches.

3 Methodology

We provide a high-level summary of RM, DPO, and GRPO in Section 3.1, followed by data processing in Section 3.2, difficulty-based data selection in Section 3.3, and model training and evaluation in Section 3.4.

3.1 PRELIMINARIES

3.1.1 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 (Bradley & Terry, 1952), as given by

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

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.

3.1.2 DIRECT PREFERENCE OPTIMIZATION

Direct Preference Optimization (DPO; Rafailov et al. (2023)) 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

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

Here, β is a parameter controlling the strength of preference enforcement relative to the reference policy π_{ref}. Larger values of β encourage π_θ to more strongly favor y_w over y_l, while smaller values keep π_θ closer to the reference policy.

Although DPO does not explicitly train an RM, it can be viewed as defining an *implicit reward function* (Rafailov et al., 2023). Specifically, the implicit reward assigned to a completion y for prompt x is given by

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

3.1.3 GROUP RELATIVE POLICY OPTIMIZATION

Group Relative Policy Optimization (GRPO; Shao et al. (2024)) is a reinforcement learning-based alignment method proposed as an efficient alternative to Proximal Policy Optimization (PPO; Schulman et al. (2017)). 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

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

where

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

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

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

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.

3.2 Data Processing

Given a preference dataset and a fixed oracle RM (a model that provides a surrogate to the true underlying reward or preference signal), the dataset is pre-processed to ensure consistency and suitability for training and evaluation. This involves the following steps:

Structure filtering. We remove examples that do not follow the expected user-assistant alternation, have mismatched numbers of chosen and rejected responses, or contain empty messages. This ensures that all examples represent well-formed multi-turn conversations.

Token length filtering. Following prior studies (e.g., Shen et al. (2024)), we include only examples where the concatenated prompt and responses (both chosen and rejected) contain at most 512 tokens.

Oracle scoring and relabeling. We score all chosen and rejected responses using the oracle RM. Examples where the original labels disagree with the oracle are relabeled to ensure consistent ground truth for training and evaluation.

Data splitting. We partition the dataset into disjoint parts to ensure that each stage of the pipeline operates on a different data split, thus preventing information leakage.

3.3 Difficulty-Based Data Selection

In this work, we quantify difficulty using the reward gap between chosen and rejected responses, i.e., $\Delta r(x, y_w, y_l) = r(x, y_w) - r(x, y_l)$, where $r(x, y)$ denotes the reward assigned to a response y for prompt x. We define **easy examples** as those with *larger reward gaps*, i.e., examples where the chosen response is clearly better than the rejected one, and **hard examples** as those with *smaller reward gaps*. Formally, given two examples $a = (x_a, y_{w,a}, y_{l,a})$ and $b = (x_b, y_{w,b}, y_{l,b})$, we consider example a *easier than* example b if $\Delta r(x_a, y_{w,a}, y_{l,a}) > \Delta r(x_b, y_{w,b}, y_{l,b})$.

We consider several setups for RM training. In the *oracle* setup, the reward gap Δr is computed using the oracle RM, which is assumed to be known. In the *noisy* setup, we consider a more realistic setting where we do not have direct access to the oracle RM and instead observe oracle rewards corrupted with additive Gaussian noise.

Concretely, we perturb the oracle scores by adding Gaussian noise with mean 0 and standard deviation in $\{0.1\sigma, 0.2\sigma, 0.4\sigma, 0.8\sigma\}$, where σ denotes the standard deviation of the oracle scores. We then compute the reward gaps based on these noisy estimates. This allows us to systematically control the noise level in difficulty estimation. In Section 5, we study an alternative setting in which the oracle rewards are approximated using a weak proxy RM.

To study the effect of example difficulty on RM, DPO, and GRPO training, we create subsets of the preference datasets based on the reward gap Δr (or its estimate in the noisy setup). We consider two types of selections as follows:

- **Easy subsets:** 20%, 40%, 60%, 80%, and 100% of the easiest examples (largest Δr values), where 100% corresponds to the entire dataset.

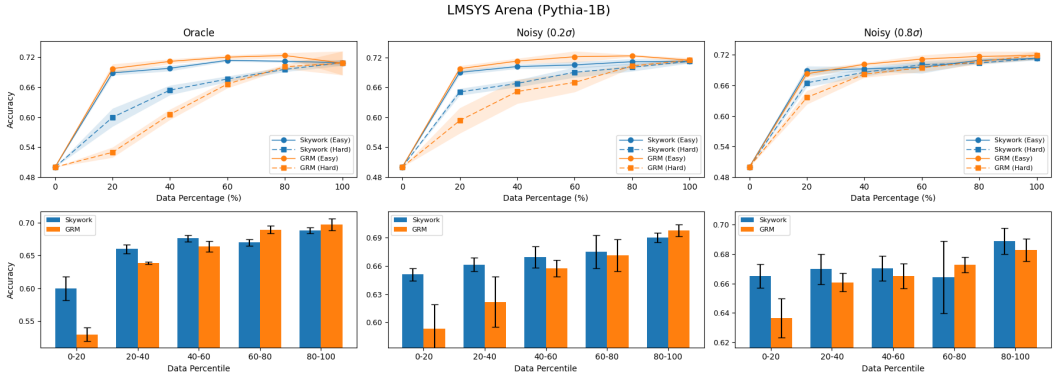


Figure 2: RM accuracy on the LMSYS Arena dataset across difficulty-based subsets and training configurations. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. Results are averaged over five runs, with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

- **Hard subsets:** 20%, 40%, 60%, 80%, and 100% of the hardest examples (smallest Δr values), where 100% corresponds to the entire dataset.

To control for dataset size while varying difficulty, we also create fixed-size slices of the dataset by reward gap percentile: 0-20%, 20-40%, 40-60%, 60-80%, and 80-100%.

3.4 Training and Evaluation Protocol

We study the effect of difficulty-based 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 oracle and noisy setups.

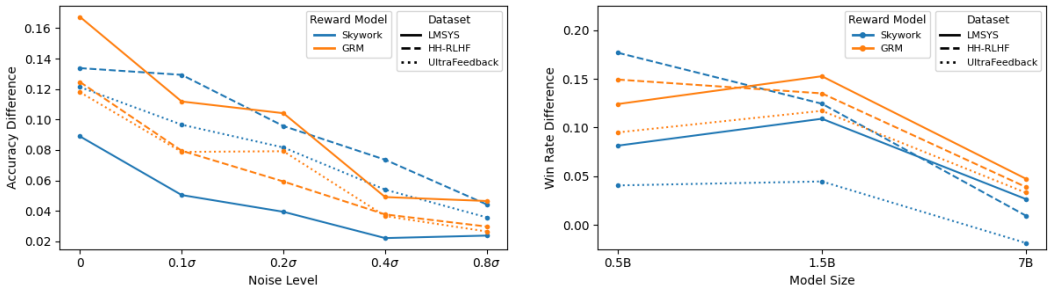
For **RM**, we train a reward model independently on each difficulty-based subset. The resulting RMs are then evaluated on a held-out test set using *accuracy*, defined as the fraction of preference pairs for which the model assigns a higher score to the chosen response than to the rejected one.

For **DPO**, we fine-tune a base LLM on each difficulty-based subset using the DPO objective. Performance is evaluated on a held-out test set by having the fine-tuned model and corresponding base model generate a response for each prompt. We then compare these paired responses using the oracle RM. We report two metrics: *win rate* (the fraction of prompts where the fine-tuned model's response receives a higher score from the oracle RM) and the *average oracle reward score* of the fine-tuned model's responses.

For **GRPO**, we first train RMs on each difficulty-based subset (as described previously) and then use each resulting RM to guide policy optimization. Specifically, we fine-tune a base LLM using GRPO, with rewards provided by the corresponding RM. This setup allows us to assess how the quality of the RM, determined by training data selection, propagates to downstream policy performance. Performance is evaluated on a held-out test set by having both the GRPO-trained policy and the base model generate responses for each prompt, with paired responses compared using the oracle RM. We report the same two metrics as in DPO.

4 Data Selection with Oracle RMs

We describe the components of our experiments in Section 4.1, followed by results for RM, DPO, and GRPO under oracle-based difficulty measures (including noisy variants) in Sections 4.2, 4.3, and 4.4, respectively.



(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 oracle setup. In both cases, the advantage of easier examples diminishes as noise increases or as model size grows.

4.1 Experimental Components

Datasets. We utilize three preference datasets that have been used in prior studies, namely LMSYS Arena (Chiang et al., 2024), Anthropic HH-RLHF (Bai et al., 2022), and HuggingFace UltraFeedback (Cui et al., 2023). Additional details on the data sources are provided in Appendix A.1.

Oracle RMs. We use two oracle RMs, namely Skywork-Reward-V2-Qwen3-4B (Liu et al., 2025) and GRM-gemma2-2B-rewardmodel-ft (Yang et al., 2024b), both of which achieve superior performance on RewardBench (Lambert et al., 2025). See Appendix A.1 for more details.

Data processing. Data processing follows the steps outlined in Section 3.2. Across all datasets and oracle RMs, 20% to 41% of examples are relabeled to ensure consistency in preference labels. Subsequently, each dataset is partitioned into disjoint subsets, with each subset allocated to a distinct part of the pipeline to prevent overlap and information leakage. Details are in Appendix A.1.

Synthetic noise construction. As mentioned in Section 3.3, we experiment with four Gaussian noise levels with mean 0 and standard deviations 0.1σ, 0.2σ, 0.4σ, and 0.8σ, where σ denotes the standard deviation of the oracle scores. The value of σ varies across datasets and oracle RMs, ranging from 5.1 to 9.6. See Appendix A.2 for more details.

4.2 RM Training

For each difficulty-based subset, we train a Pythia-1B model (Biderman et al., 2023) for one epoch with a learning rate of $1e{-}5$, selected via parameter sweeping to maximize RM accuracy. This allows us to evaluate how the choice of examples, from easiest to hardest, affects RM performance in the oracle and noisy setups.

We report RM accuracy across all difficulty-based subsets, datasets, oracle RMs, and training setups. For each configuration, results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation. Figure 2 presents the results on the LMSYS Arena dataset, illustrating the effect of example difficulty on RM performance. For clarity, we display only the oracle, 0.2σ, and 0.8σ noisy setups in the main text; full results are provided in Appendix A.3. Here, the 0% setting corresponds to the non-fine-tuned Pythia-1B model, which serves as an accuracy baseline.

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

Noisy setups: the advantage of easier examples diminishes as noise increases. When difficulty is estimated using noisy versions of the oracle rewards, we observe the same overall trend: training on easier examples generally yields higher RM accuracy than training on harder ones. However,

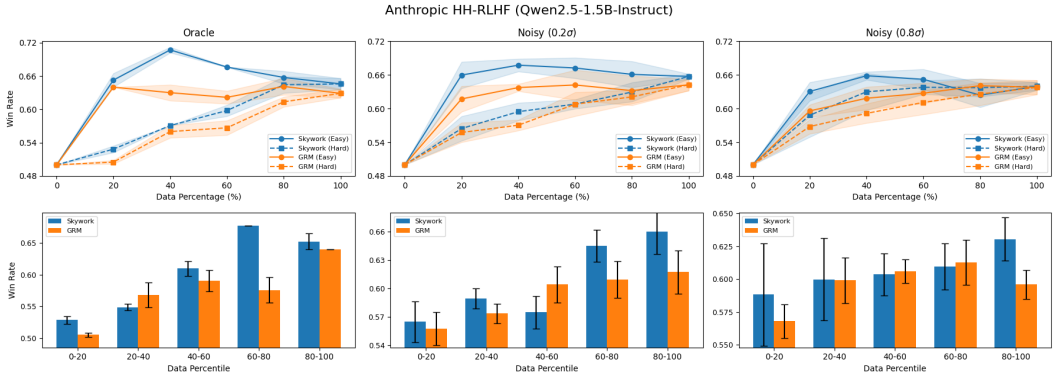


Figure 4: DPO win rate on the Anthropic HH-RLHF dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. Results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

as the noise level increases, the performance gap between easier and harder subsets progressively shrinks, and in some cases becomes negligible (see Figure 3a). This suggests that inaccurate difficulty estimates can obscure the true benefit of easy examples, which may help explain the conflicting conclusions reported in prior studies, such as Qi et al. (2025), that rely on imperfect RMs or proxy difficulty measures.

4.3 DPO Training

We fine-tune Qwen2.5-Instruct models (Yang et al., 2024a) 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 $3\text{e}{-6}$ and a regularization parameter of $\beta = 0.1$, selected via parameter sweeping to maximize win rate. Win rate calculation uses a generation temperature of 0.7, consistent with prior work (Gao et al., 2025).

Figure 4 presents the win rates on the Anthropic HH-RLHF dataset using a 1.5B base model. All results are averaged over five runs, with error bars reporting mean $\pm$ standard deviation. The 0% setting corresponds to the non-fine-tuned Qwen2.5-Instruct model. For clarity, we display only the oracle, 0.2σ, and 0.8σ noisy setups in the main text; full results are in Appendices A.4 and A.5.

Oracle setup: easier examples improve DPO performance, particularly for smaller base models. When difficulty is measured using the oracle reward gap, we find that training on easier examples yields higher win rates and average reward scores, particularly for smaller base models (see Figure 3b). In some cases, using only the 20% easiest examples achieves performance comparable to (or better than) training on the full dataset. For the 7B models, the trend is less consistent, and moderate or harder subsets can outperform the easiest examples in some cases. This trend is broadly in line with Gao et al. (2025), who found that larger models are more robust to harder examples.

Noisy setups: increasing noise weakens the advantage of easier examples. When difficulty is estimated from noisy reward gaps, the qualitative preference for easier examples largely persists, but the performance gap between easier and harder subsets steadily diminishes as the noise level increases. At higher noise levels, the distinction between easy and hard examples becomes increasingly blurred, and in some cases harder subsets can match or outperform easier ones. This may help explain why some prior studies, e.g., Qi et al. (2025), reported that harder examples are more beneficial for DPO training.

4.4 GRPO Training

We fine-tune Qwen2.5-Instruct models (Yang et al., 2024a) with the same model sizes as in the DPO experiments (Section 4.3). Unlike RM and DPO training, we do not construct new training datasets

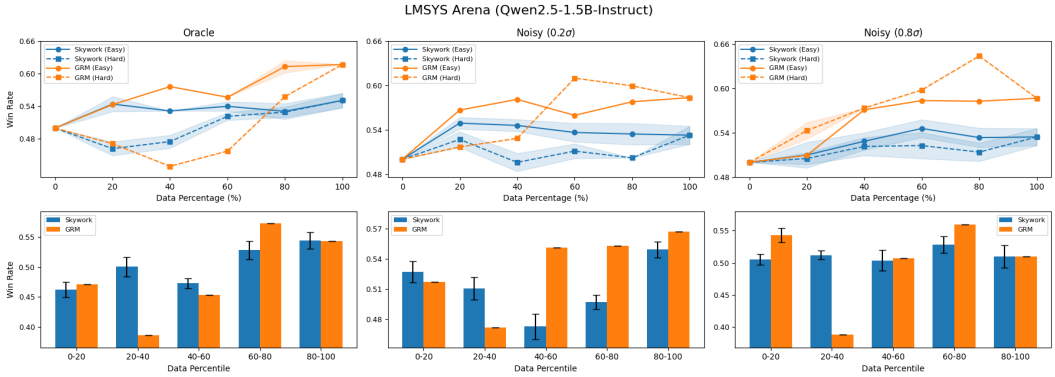


Figure 5: GRPO win rate on the LMSYS Arena dataset across difficulty-based subsets and training setups, using Qwen2.5-1.5B-Instruct base model. Blue and orange denote the Skywork and GRM reward models, respectively. Solid and dashed lines indicate training on the easiest and hardest examples. Results are averaged over five runs, with error bars showing mean $\pm$ standard deviation.

for GRPO. Instead, we use reward models that have already been trained on different difficulty-based subsets (Section 4.2) to guide GRPO optimization.

Training is run for one epoch with a learning rate of $1\mathrm{e}{-6}$ and a regularization parameter of $\beta = 0.1$, selected via parameter sweeping to maximize win rate. We use a group size of $G = 8$ and a sampling temperature of 1.0. We evaluate GRPO-trained models using the same protocol as in Section 4.3. Results are shown in Figure 5 for the LMSYS Arena dataset using a 1.5B base model. Full results are in Appendices A.6 and A.7.

Across both oracle and noisy setups, GRPO shows trends that are generally similar to those reported for DPO. Notably, **the benefit of training on easier examples is strongest for smaller base models and becomes less pronounced or absent for larger models**, indicating that model scale further diminishes sensitivity to example difficulty under GRPO.

5 Data Selection with Weak Proxy RMs

In Section 4, we studied difficulty-based data selection where oracle reward scores are either known or observed through a noisy version with Gaussian noise. We found that easier examples consistently outperform harder ones across RM, DPO, and GRPO, although this advantage diminishes as reward estimation quality decreases. In this section, we investigate how difficulty-based data selection is affected when a weak proxy RM is used to approximate the oracle rewards.

We train a Pythia-1B proxy RM for one epoch using a learning rate of $1\mathrm{e}{-5}$, selected via parameter sweeping to maximize RM accuracy. Depending on the dataset and oracle RM, the proxy achieves 67%–78% test accuracy (see Appendix B.1). For each example, the proxy assigns scores to both the chosen and rejected responses, which are used to simulate oracle feedback.

5.1 RM Training

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 presented in Appendix B.2.

In the proxy setup, we observe a striking and consistent phenomenon across all datasets and oracle RMs. Specifically, training RMs on the hardest subsets yields accuracies far below 50%, indicating severe misalignment between proxy-based difficulty estimates and the true oracle 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 B.3. We observe a reversed trend in the proxy > 0 setup as compared to the oracle 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., Qi et al. (2025), 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, resulting in highly noisy difficulty estimates and less stable data selection. This sensitivity to reward estimation quality likely explains the divergent conclusions reported in prior work, where difficulty is often inaccurately inferred from weak RMs. From a practical perspective, our results underscore the importance of *training sufficiently strong RMs for difficulty estimation*, as reliable estimates enable effective data selection and allow downstream alignment methods to achieve comparable performance using substantially less data.

6 CONCLUSION

In this paper, we revisited difficulty-based data selection for preference alignment by assuming access to an oracle reward model, allowing us to isolate the intrinsic role of example difficulty from errors in reward gap estimation. Across RM, DPO, and GRPO training, we find that when difficulty is measured using the oracle reward gap, easier examples consistently yield better performance, particularly for smaller base models. Notably, in several settings, using only the 20% easiest examples achieves performance comparable to or better than that obtained with all available data, highlighting the potential for substantial gains in efficiency.

We further show that this conclusion is robust to noisy reward signals: when oracle rewards are increasingly corrupted by Gaussian noise, easier examples remain preferable, although the performance gap between easier and harder subsets diminishes as noise increases. Finally, when difficulty is estimated using a weak proxy RM trained on limited data, this pattern no longer holds. These insights help reconcile conflicting findings in prior work and highlight how conclusions about optimal data selection strategies can depend on how difficulty is defined and measured.

Limitations. Our study relies on known oracle RMs to define difficulty and evaluate alignment, a setting that may not fully reflect real-world deployments. While our noisy and proxy-based setups partially bridge this gap, the observed changes in optimal data selection indicate that miscalibrated difficulty estimates can meaningfully affect which examples are most informative.

Future work. These results lay encouraging groundwork for several exciting next steps. A particularly promising direction is to explore adaptive or curriculum-style data selection that adjusts difficulty over training. Another potential direction is to extend this framework to multi-attribute or task-specific notions of difficulty beyond a single scalar reward gap.

REFERENCES

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.

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.

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.

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

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

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.

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

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.

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.

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.

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

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.

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.

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.

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.

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.

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.

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.

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

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*, 2024b.

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

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.

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

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.

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

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.

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

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.

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.

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.

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

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*, 2024a.

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

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.

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

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

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.

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.

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 *Data Selection with Oracle RMs* DETAILS

A.1 DATA STATISTICS AND SPLITTING

Tables 2 and 3 summarize the number of examples retained at each processing stage and the proportion of relabeled examples for each dataset under the two oracle reward models (RMs). Table 4 shows one example from each dataset, highlighting both the chosen and rejected responses.

Table 2: Data processing statistics using Skywork-Reward-V2-Qwen3-4B as the oracle 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 3: Data processing statistics using GRM-gemma2-2B-rewardmodel-ft as the oracle 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 (Lambert et al., 2025). 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 4: 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" }}

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 3.3), 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.

A.2 Oracle Score Distribution

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

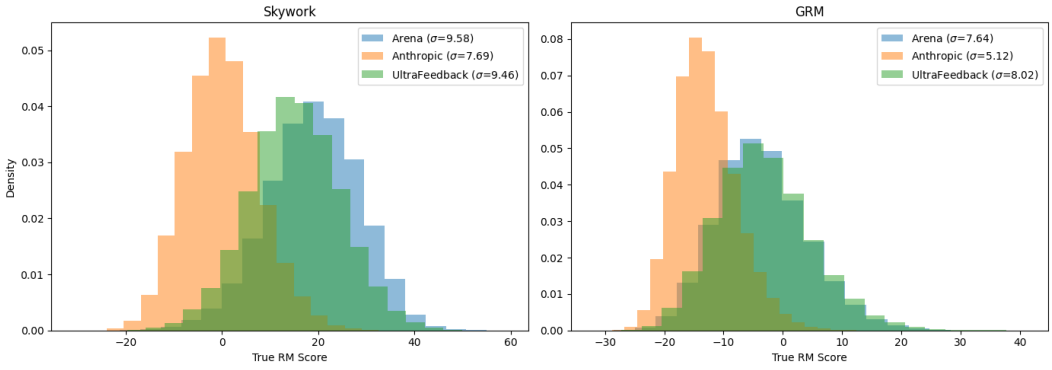
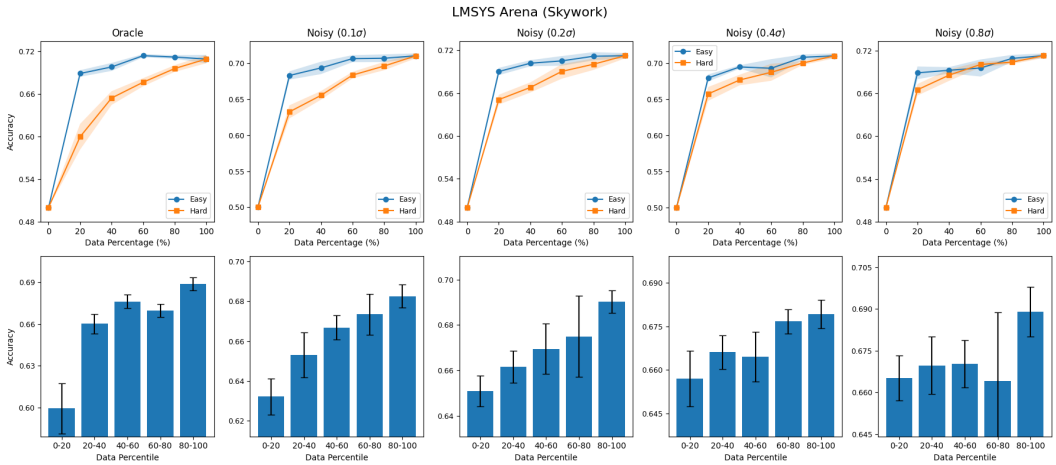


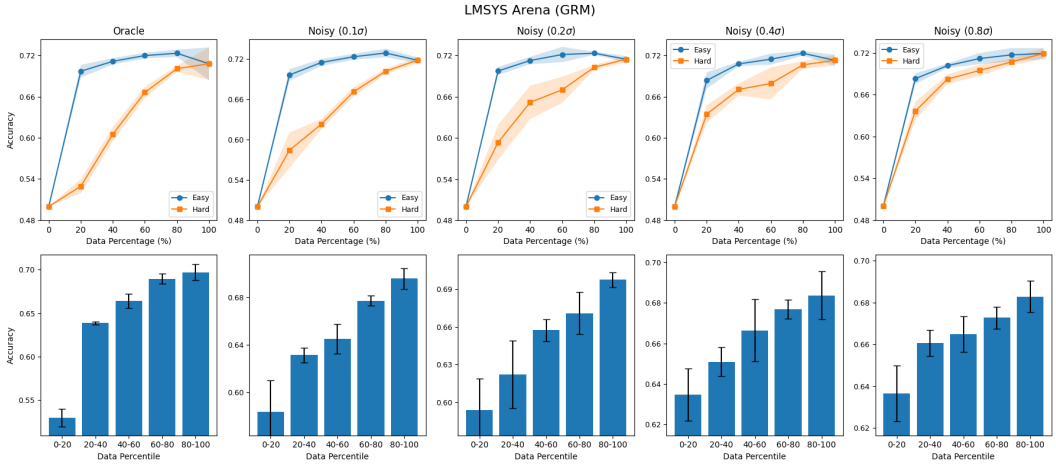
Figure 6: Distribution of reward scores across datasets and oracle RMs. The standard deviation ranges from 5.12 to 9.58. Here, Skywork refers to Skywork-Reward-V2-Qwen3-4B and GRM refers to GRM-gemma2-2B-rewardmodel-ft.

A.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.

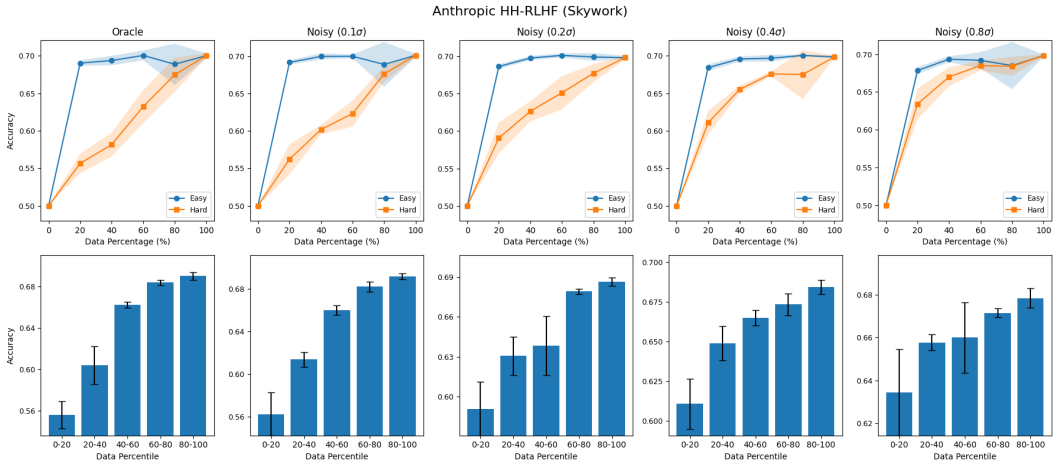


(a) Skywork reward model

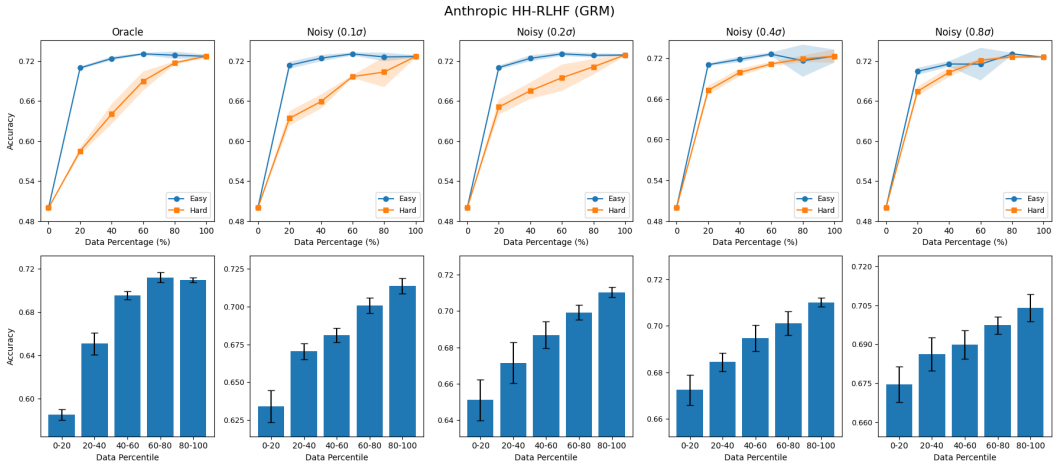


(b) GRM reward model

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 oracle 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.

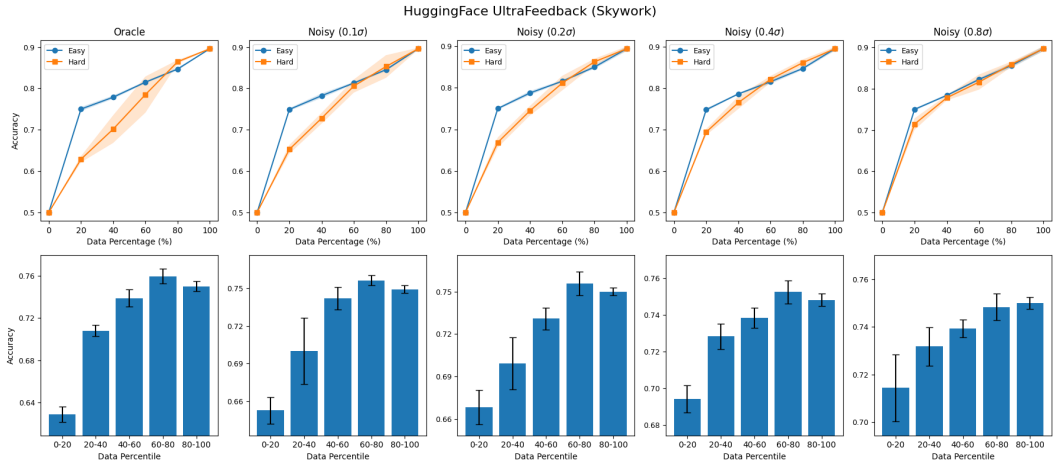


(a) Skywork reward model

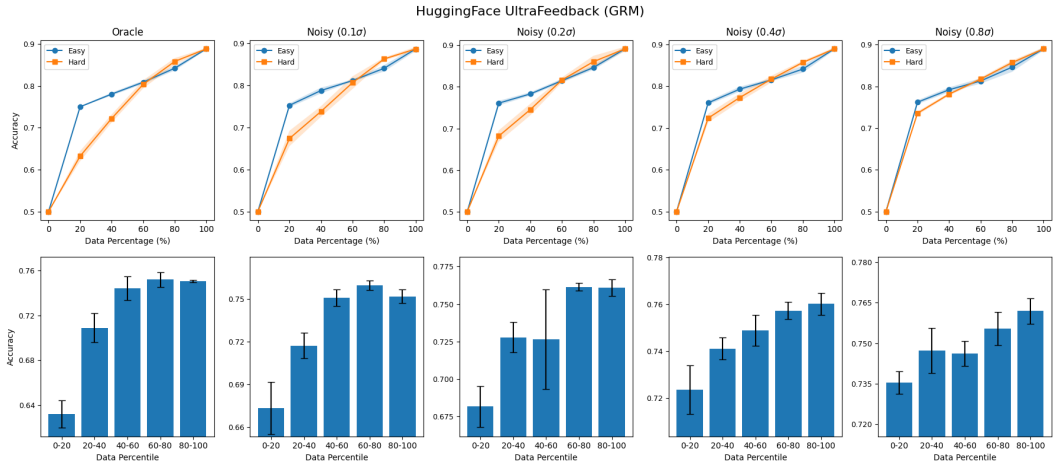


(b) GRM reward model

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 oracle 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

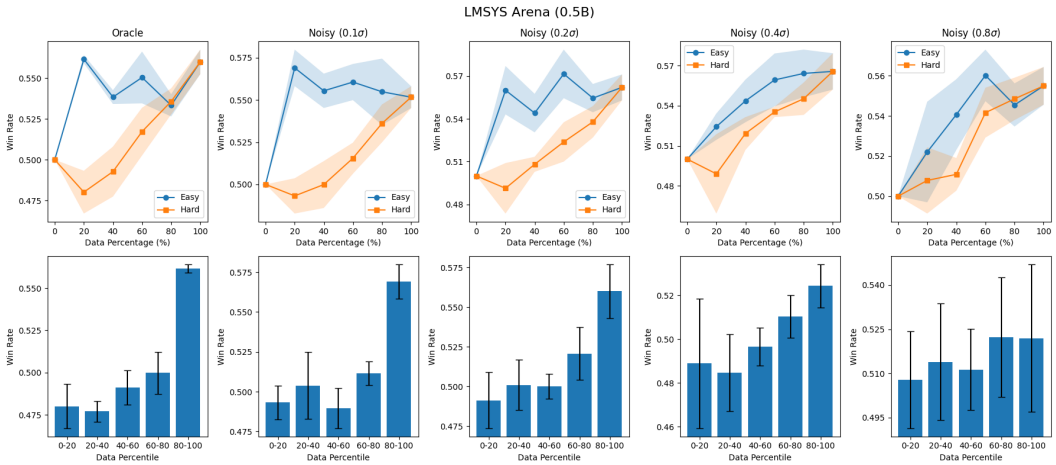


(b) GRM reward model

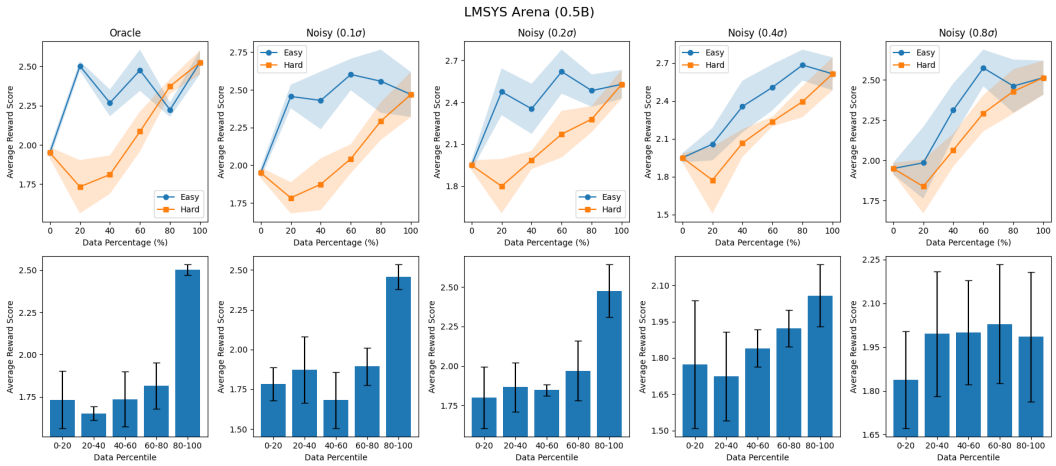
Figure 9: RM accuracy on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM oracle RMs, respectively. Blue and orange lines respectively correspond to training with the easiest and hardest examples. Results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

A.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 oracle RM. For a discussion and interpretation of these results, see Section 4.3.

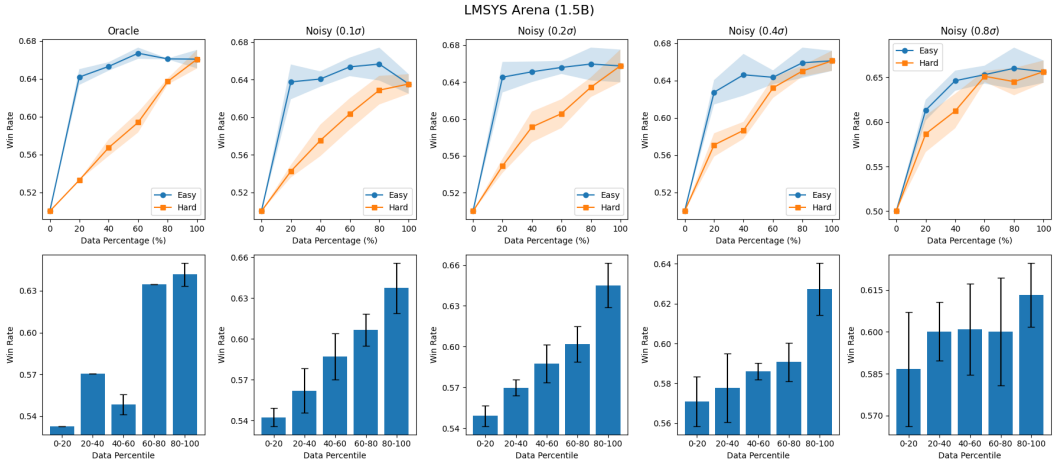


(a) Win rate

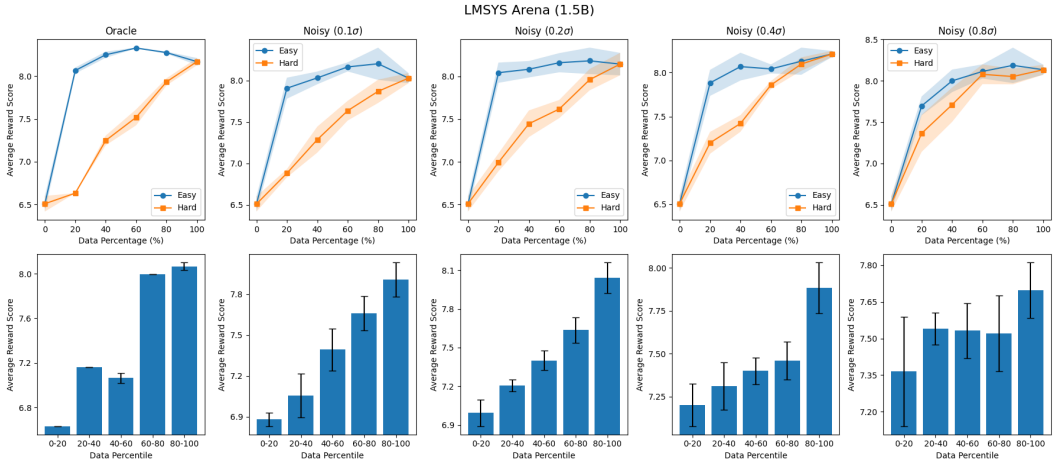


(b) Average reward score

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 oracle 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.

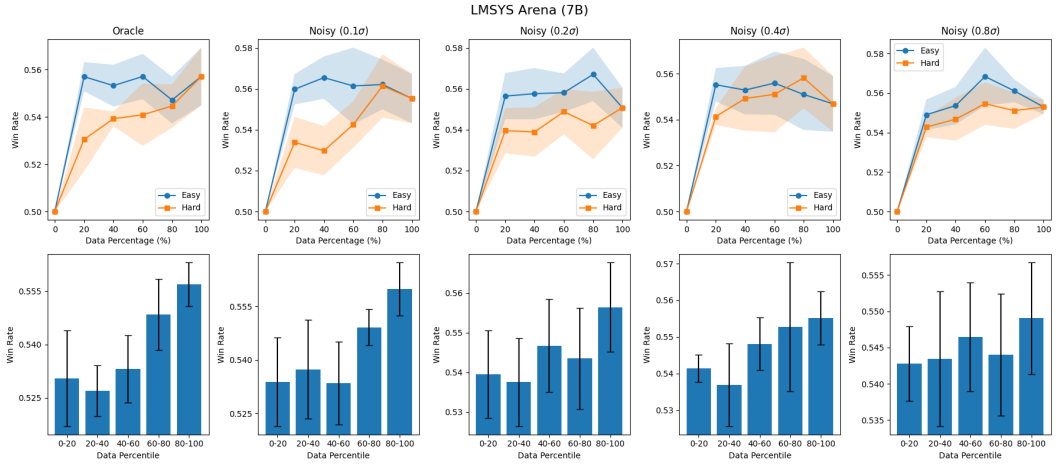


(a) Win rate

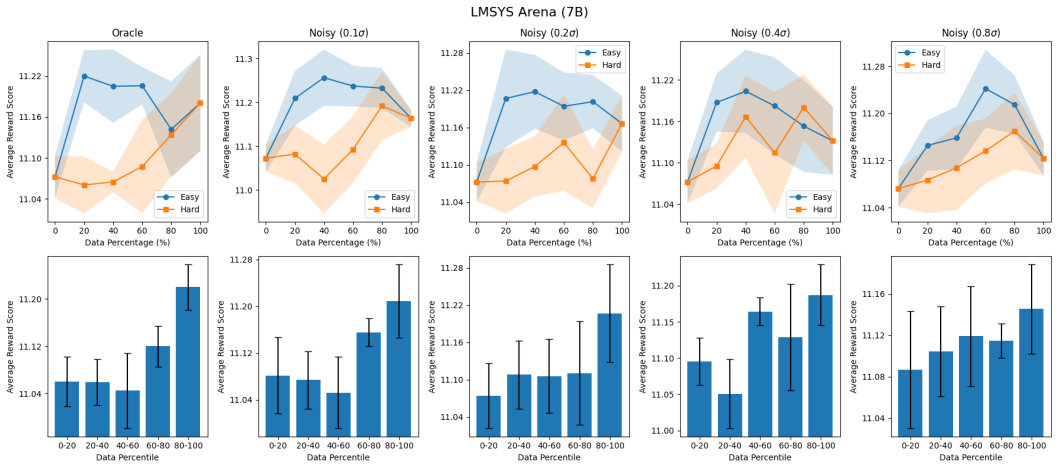


(b) Average reward score

Figure 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 oracle 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.

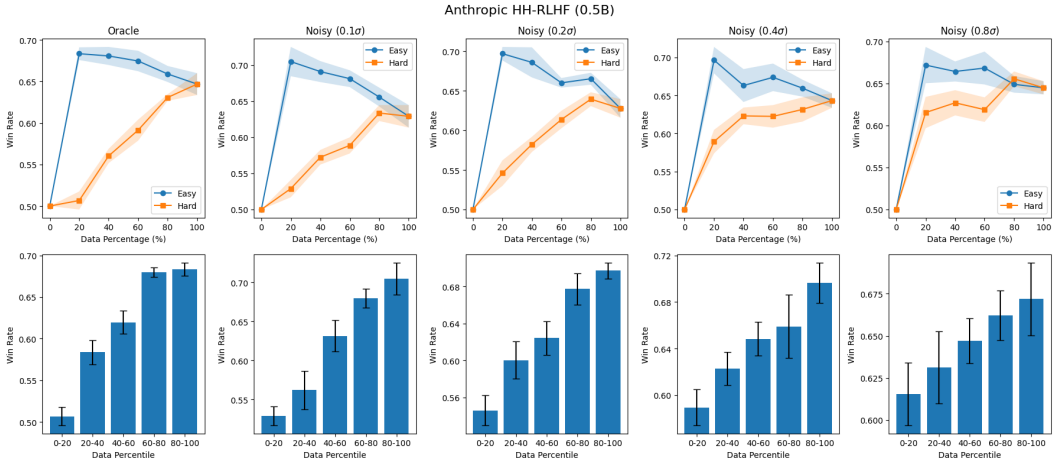


(a) Win rate

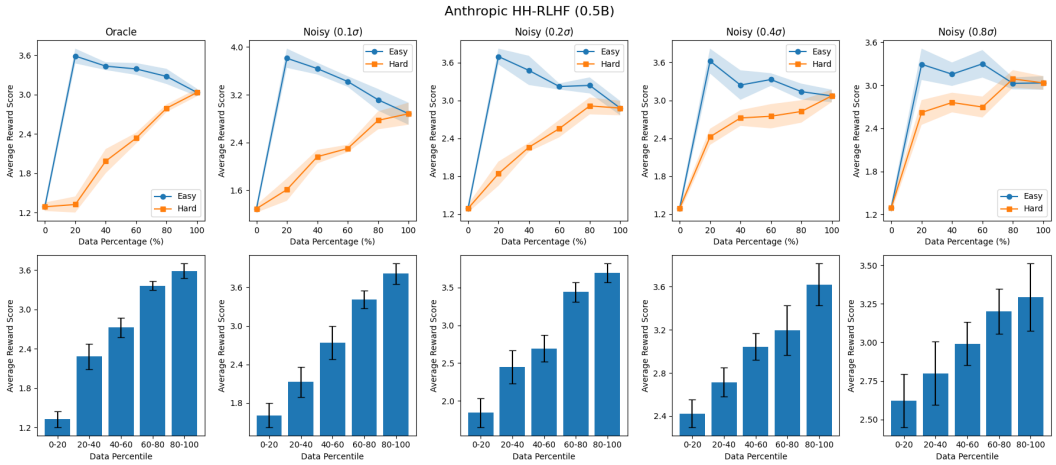


(b) Average reward score

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 oracle 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.

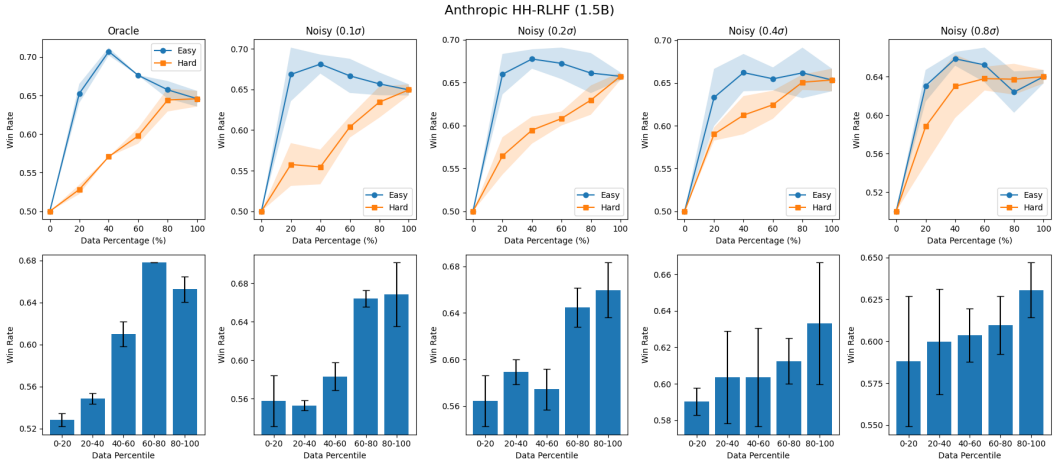


(a) Win rate

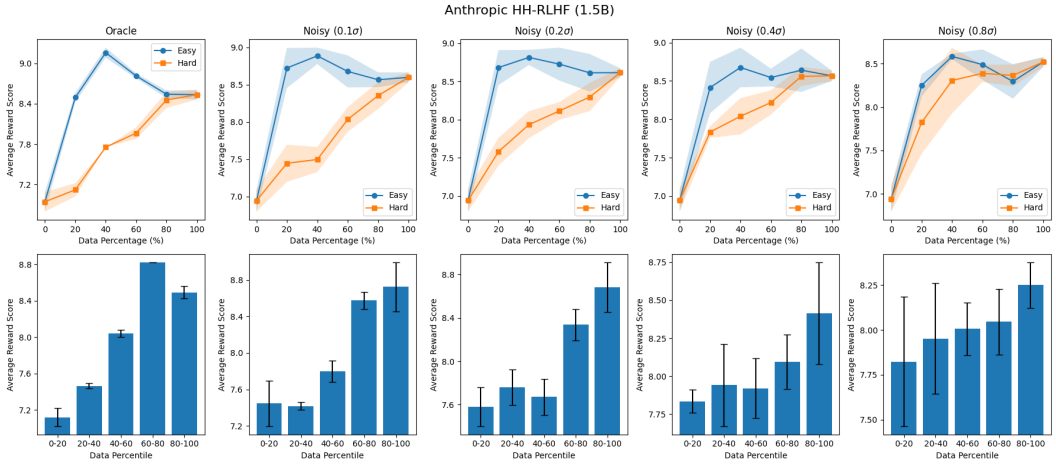


(b) Average reward score

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 oracle 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.

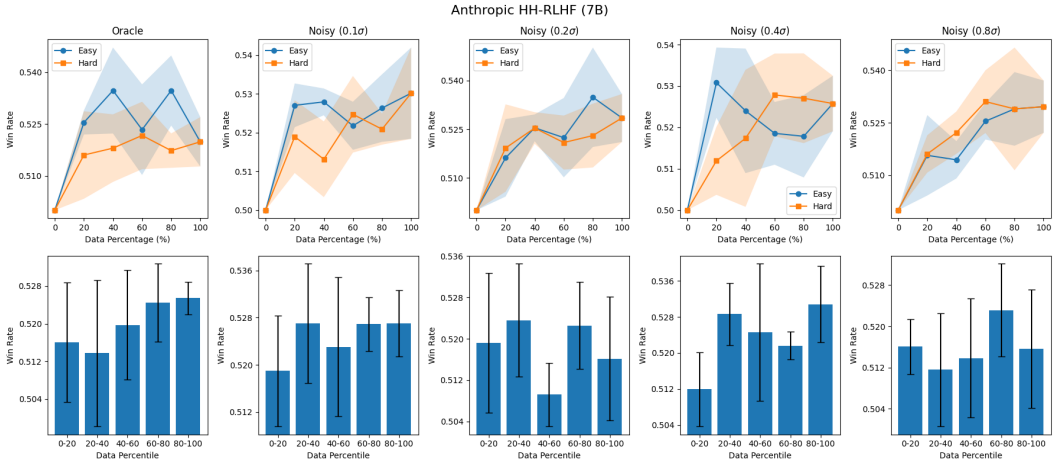


(a) Win rate

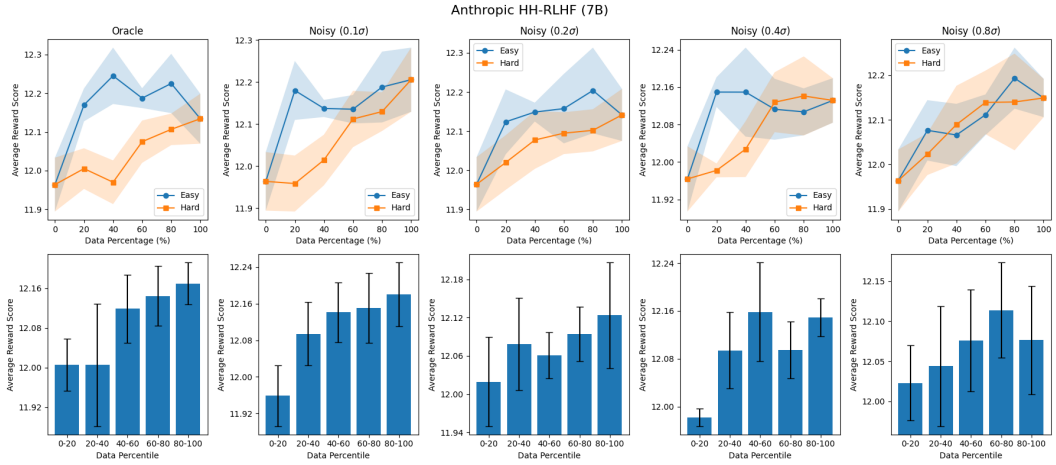


(b) Average reward score

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 oracle 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.

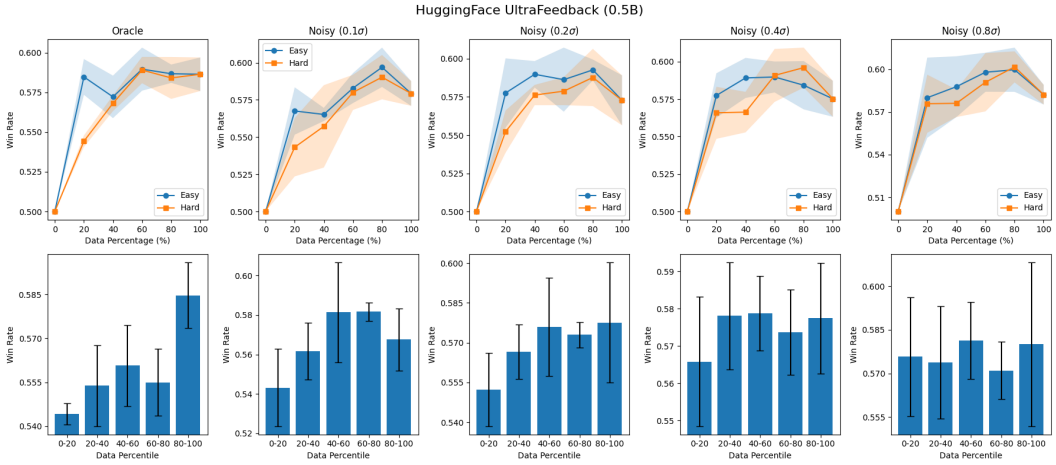


(a) Win rate

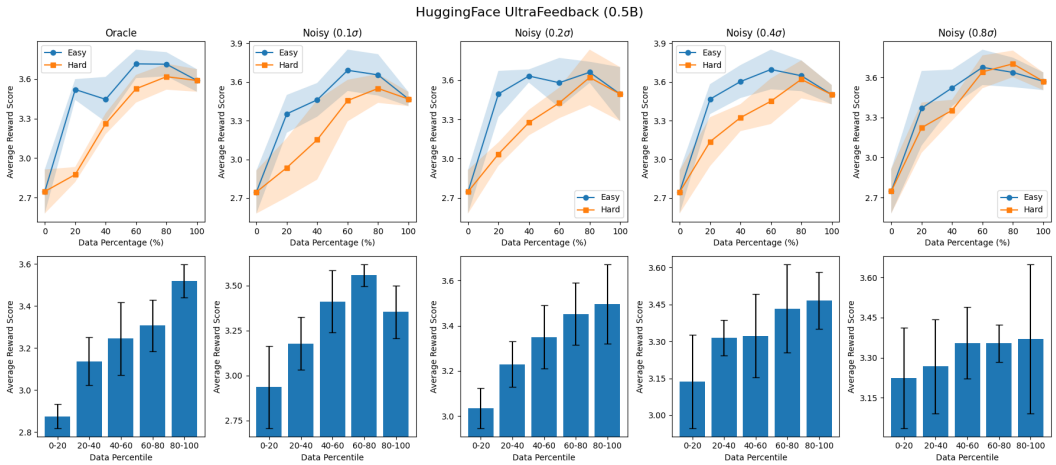


(b) Average reward score

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 oracle 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.

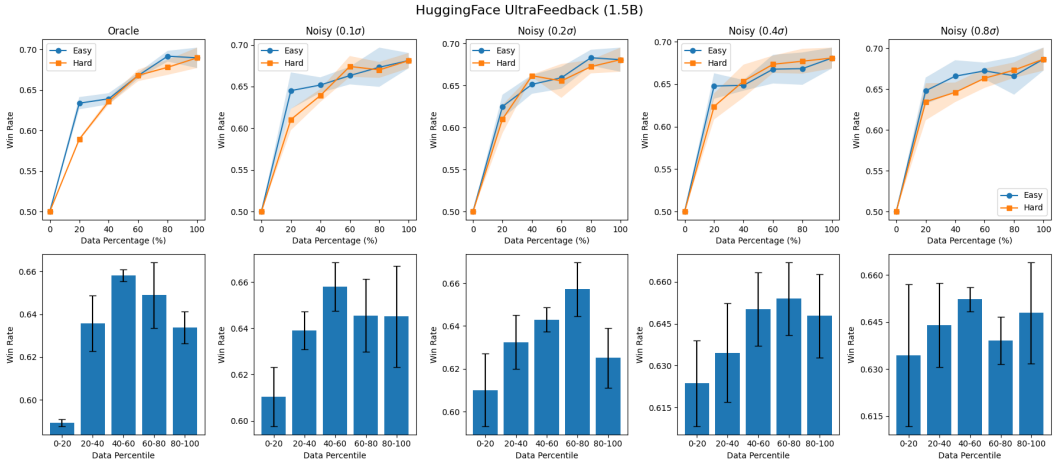


(a) Win rate

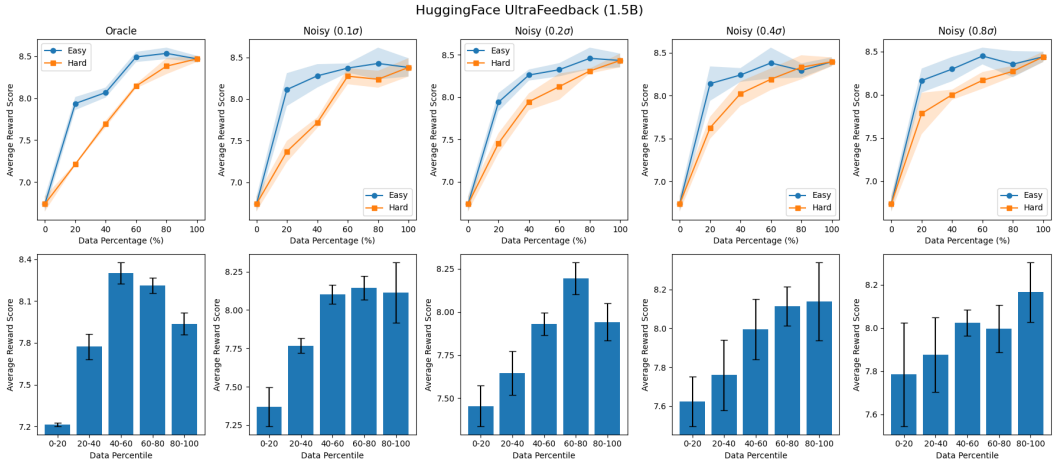


(b) Average reward score

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 oracle 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.

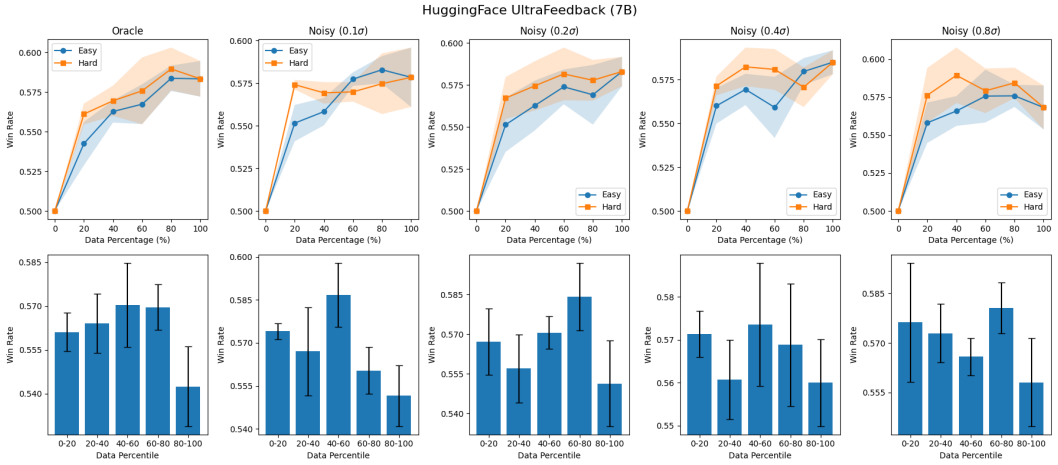


(a) Win rate

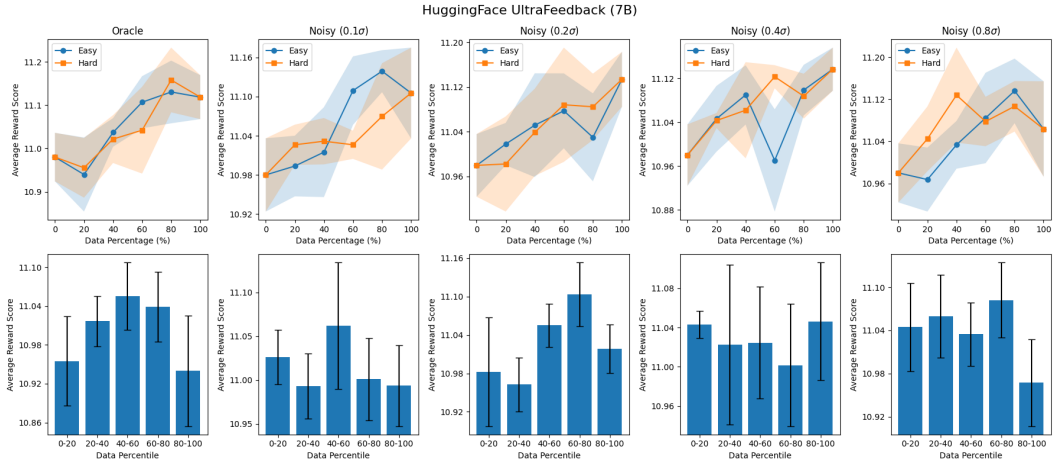


(b) Average reward score

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 oracle 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

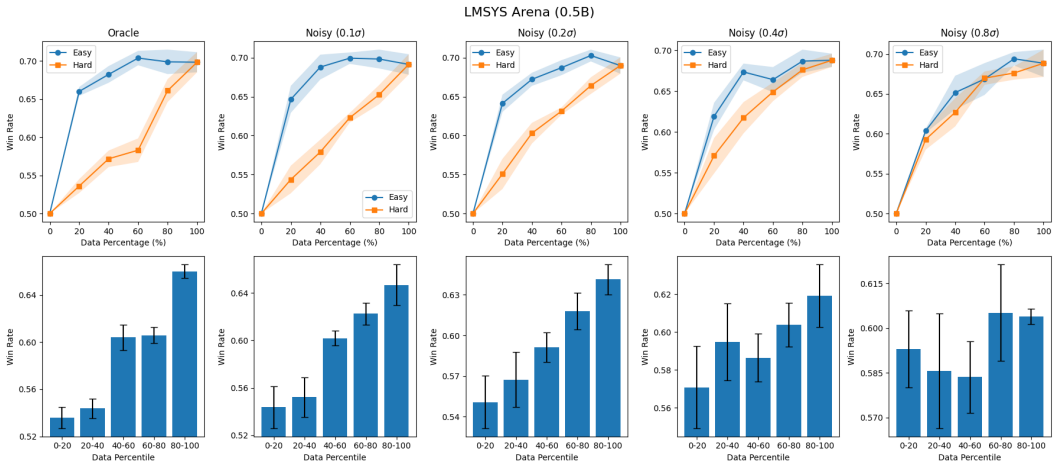


(b) Average reward score

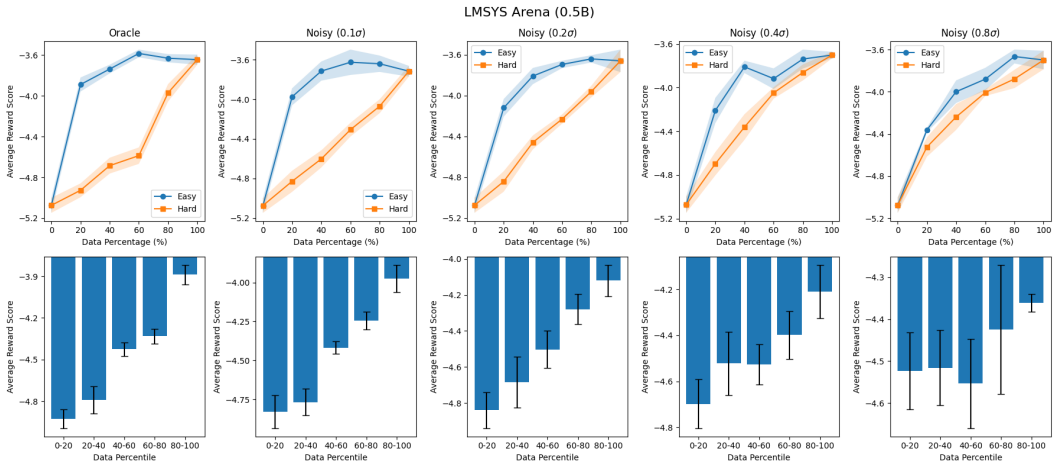
Figure 18: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork oracle RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively.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

A.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 oracle RM. For a discussion and interpretation of these results, see Section 4.3.

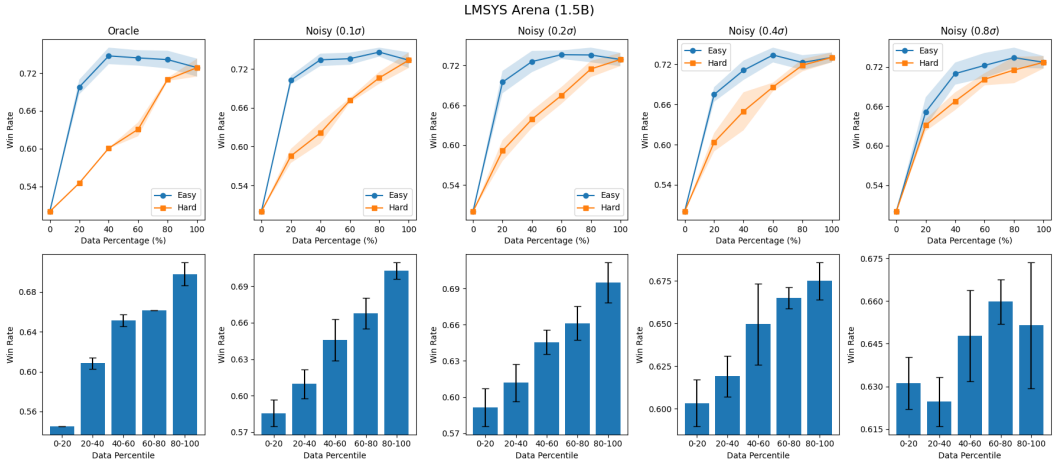


(a) Win rate

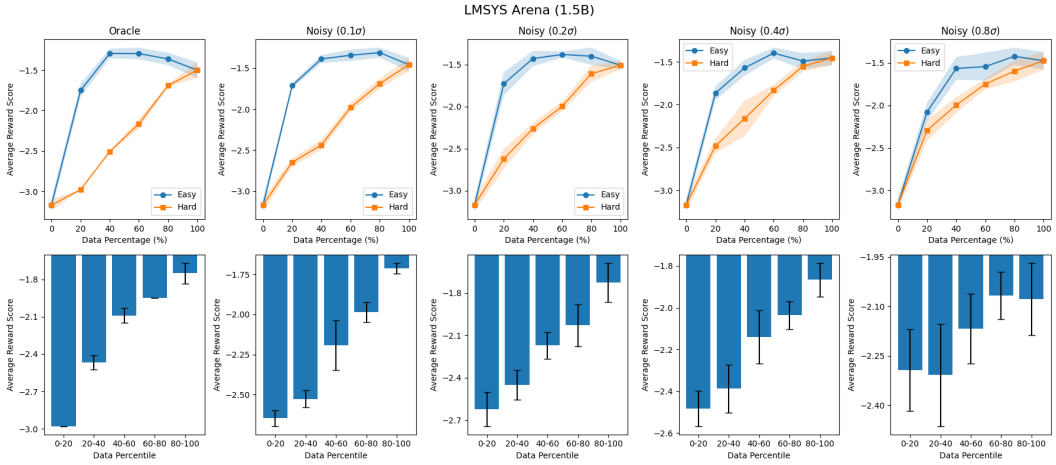


(b) Average reward score

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 oracle 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.

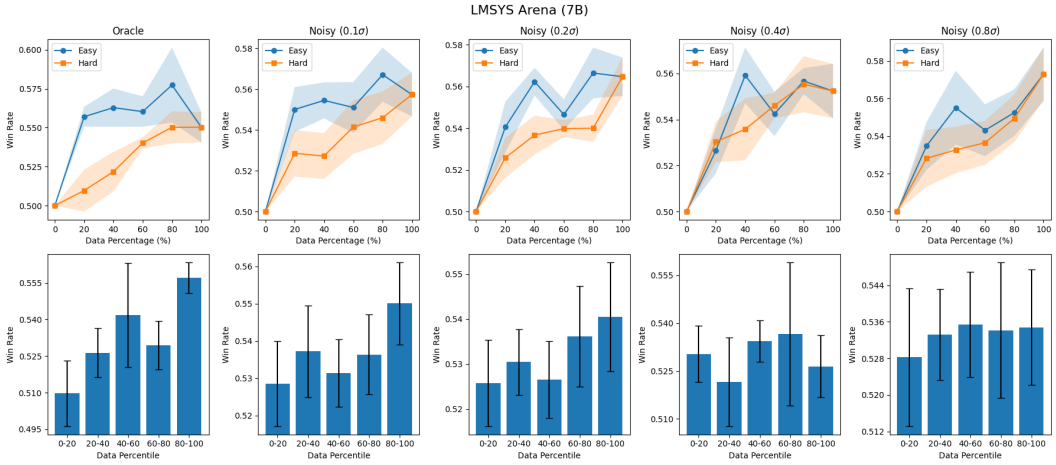


(a) Win rate

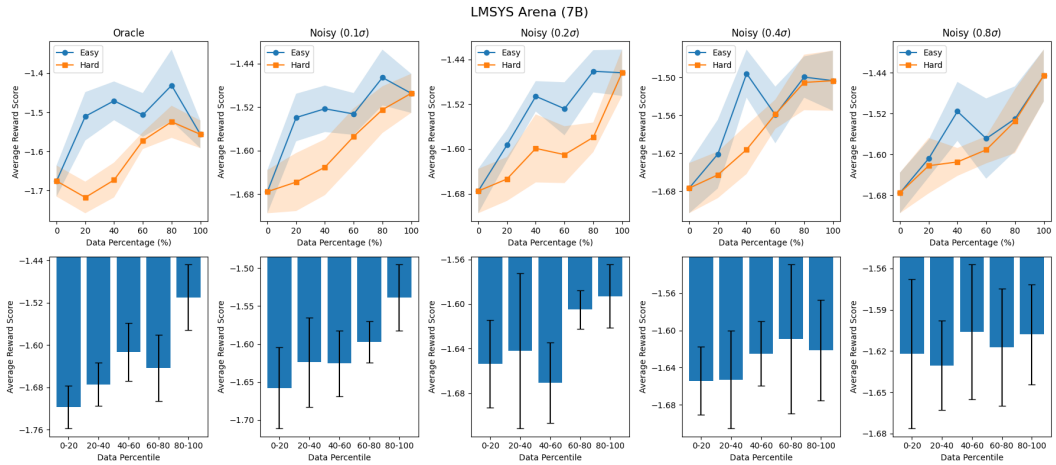


(b) Average reward score

Figure 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 oracle 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.

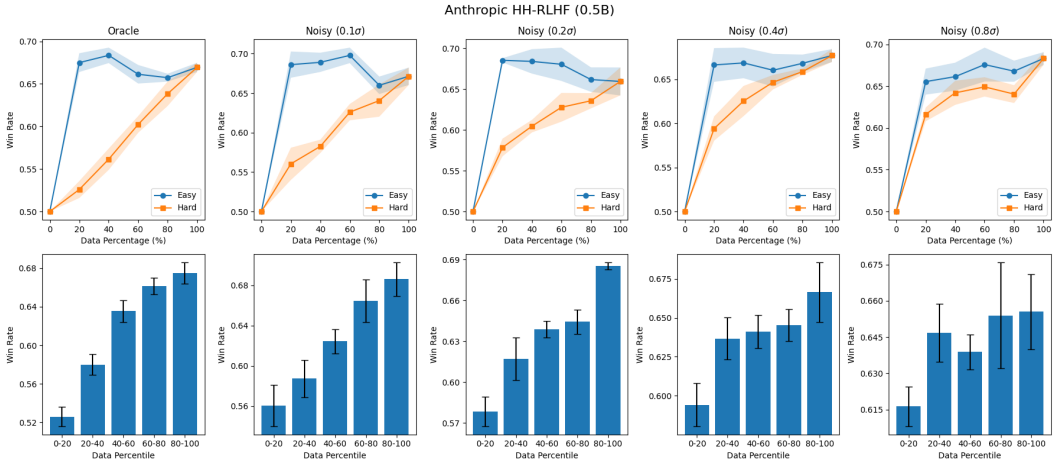


(a) Win rate

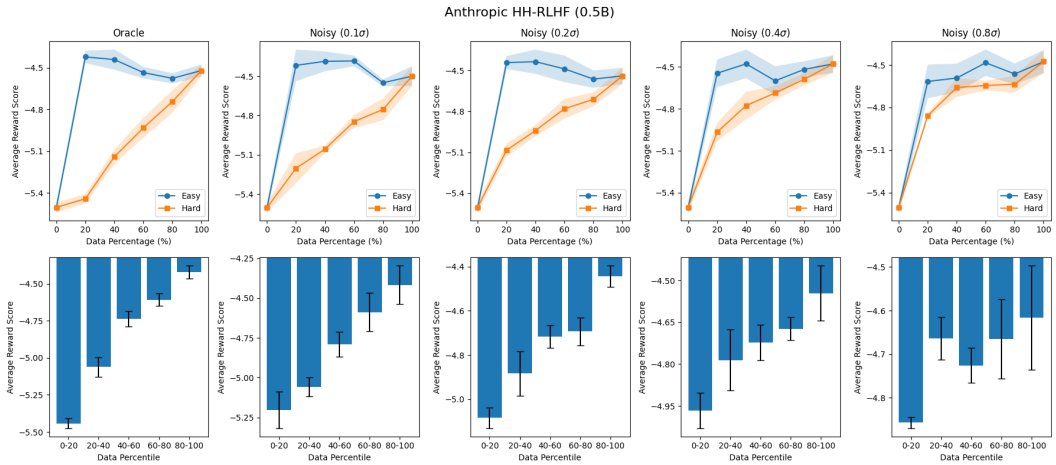


(b) Average reward score

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 oracle 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.

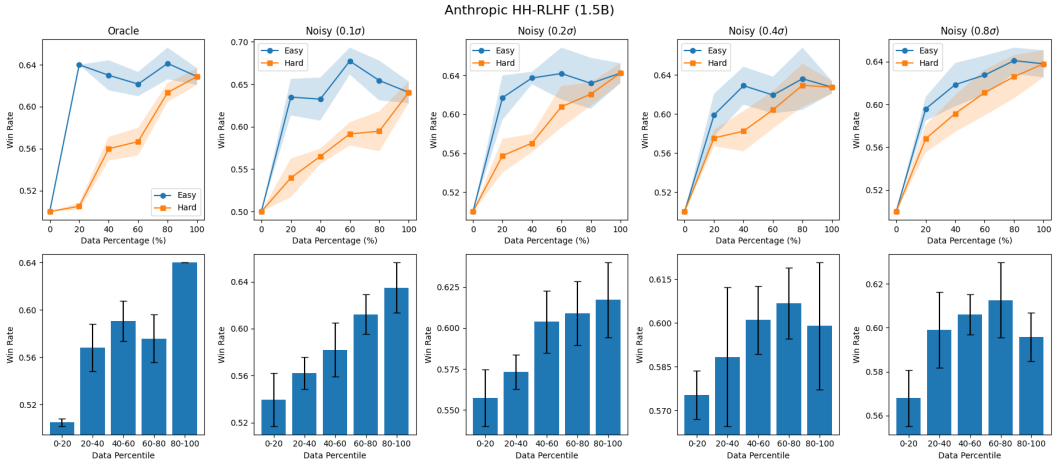


(a) Win rate

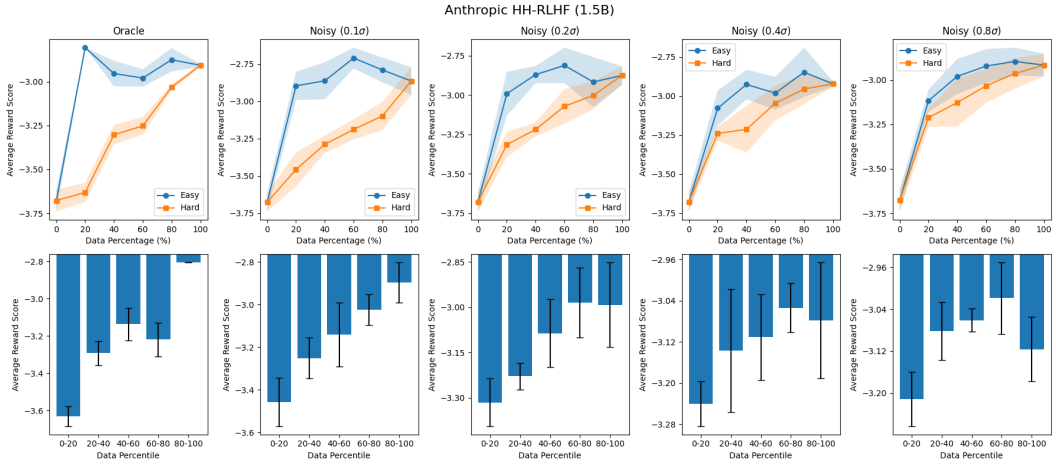


(b) Average reward score

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 oracle 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.

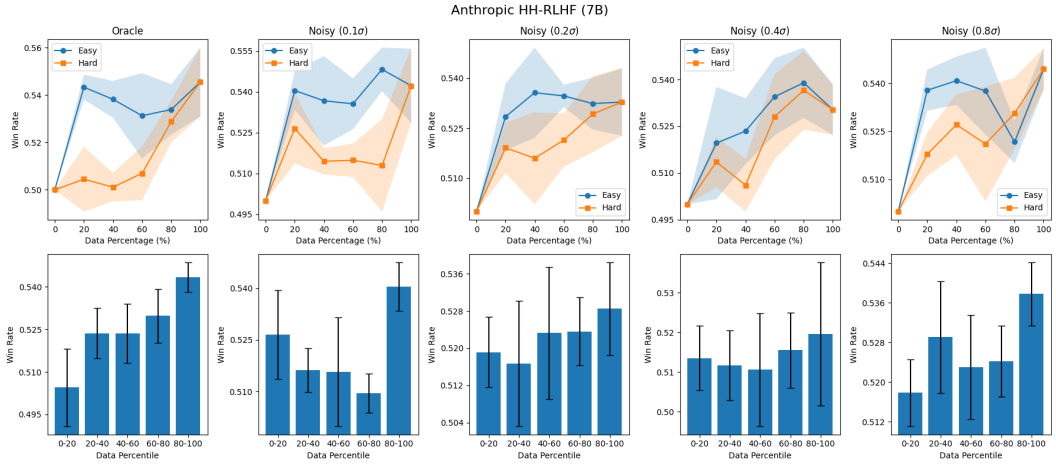


(a) Win rate

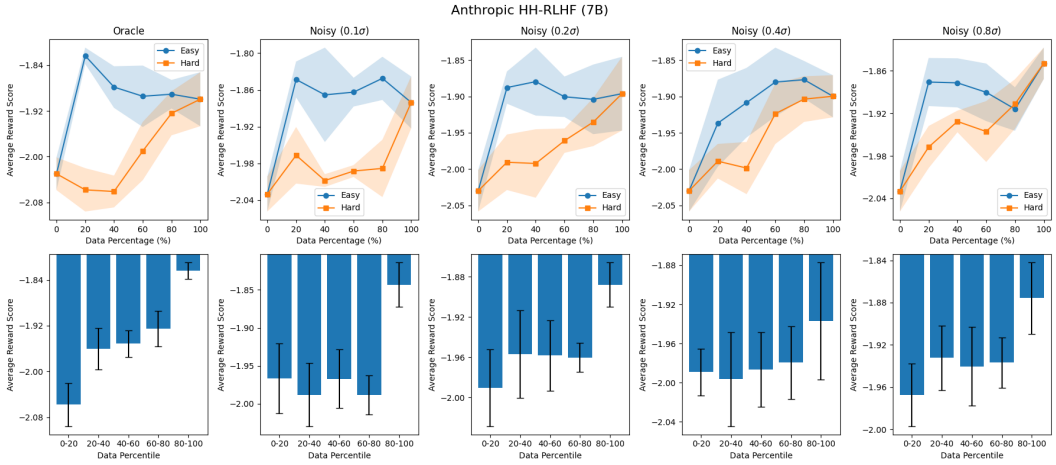


(b) Average reward score

Figure 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 oracle 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.

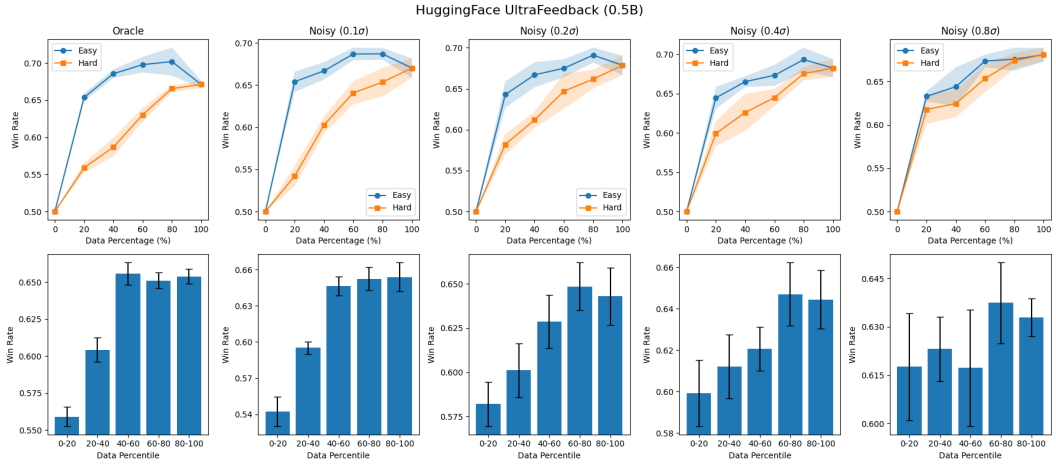


(a) Win rate

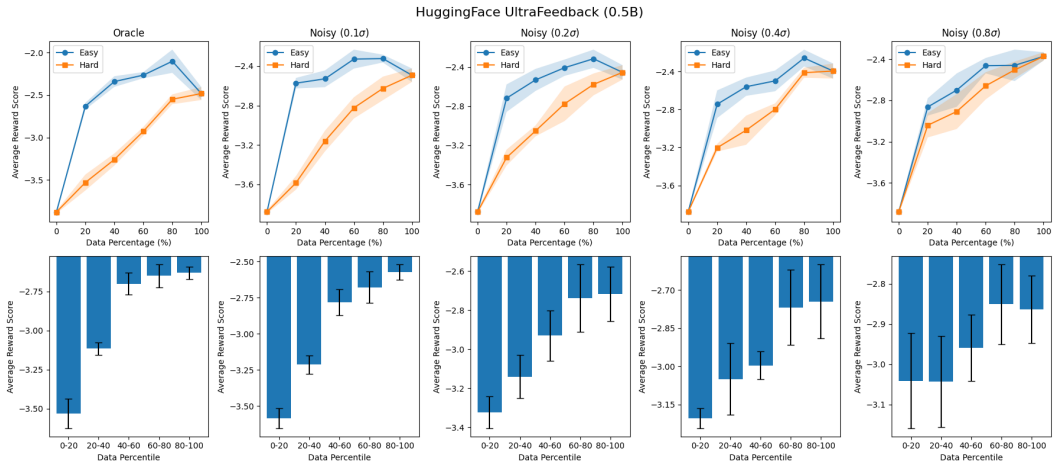


(b) Average reward score

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 oracle 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.

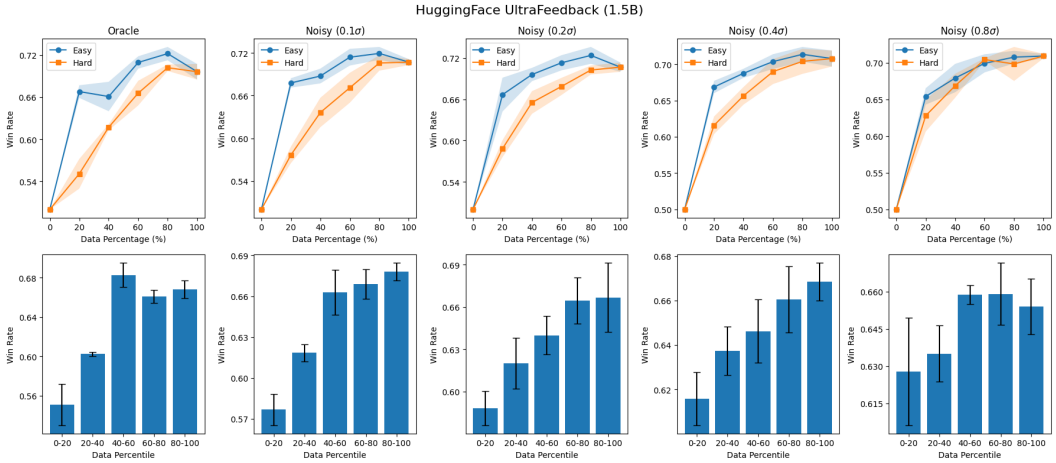


(a) Win rate

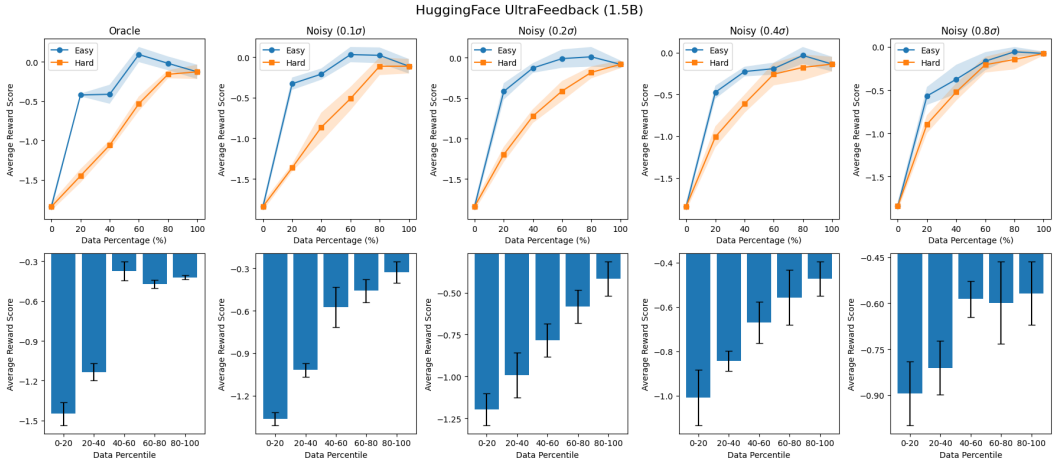


(b) Average reward score

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 oracle 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.

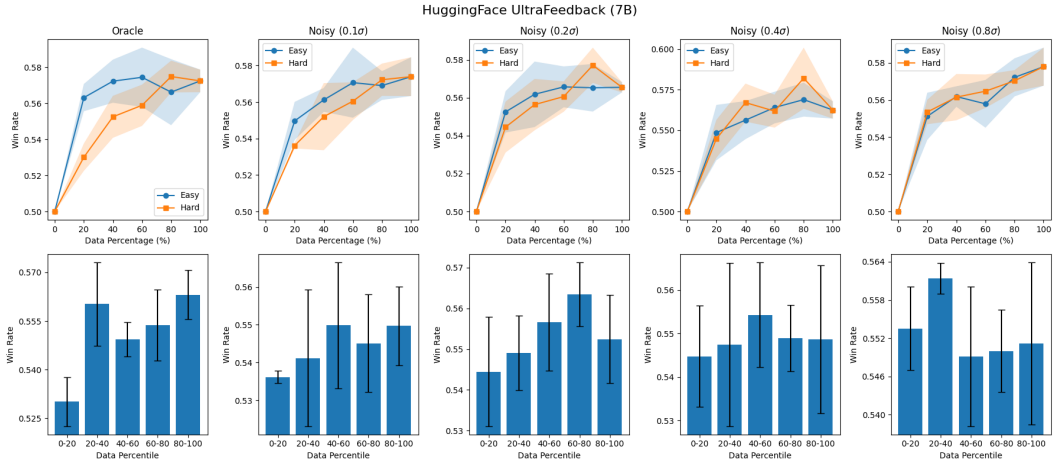


(a) Win rate

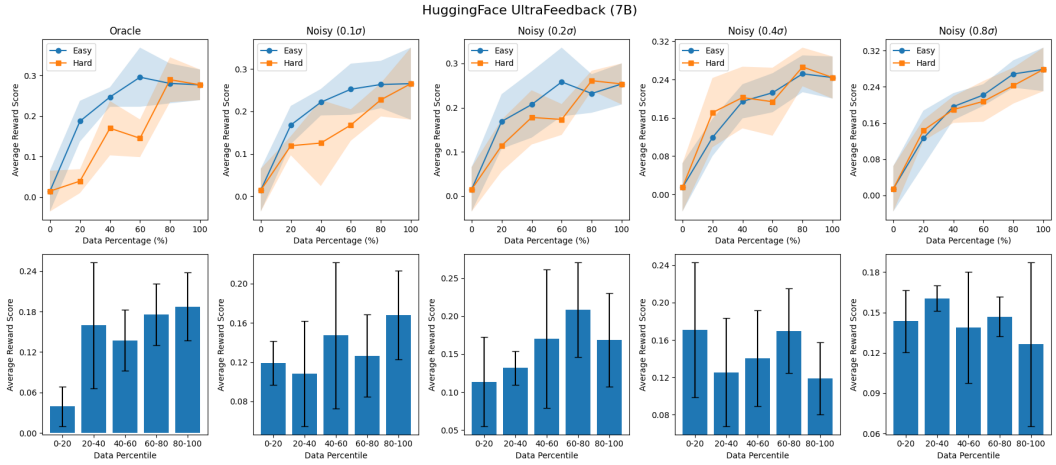


(b) Average reward score

Figure 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 oracle 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

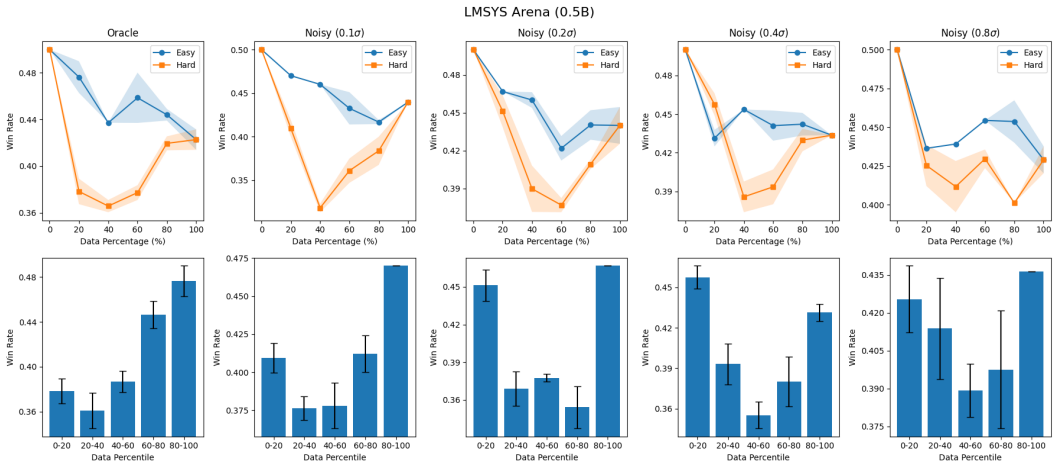


(b) Average reward score

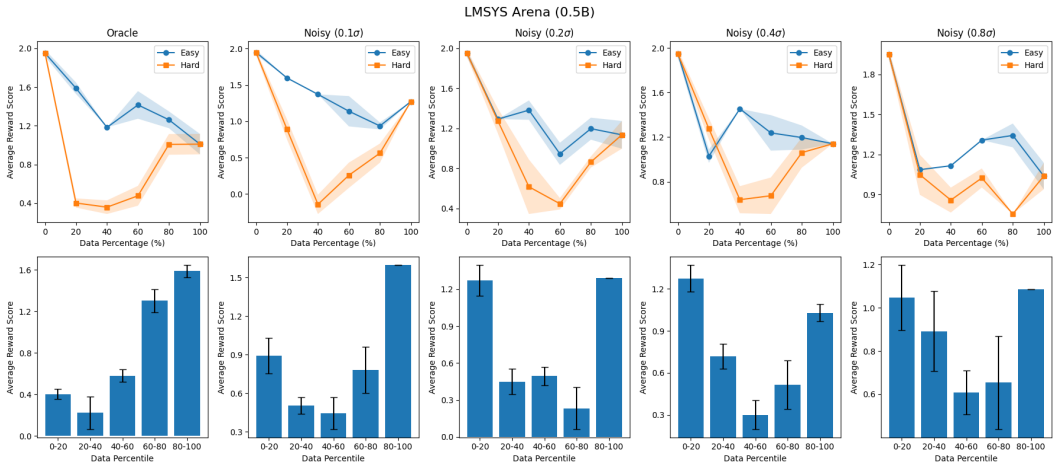
Figure 27: DPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and GRM oracle RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively. Results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

A.6 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 oracle RM. For a discussion and interpretation of these results, see Section 4.4.

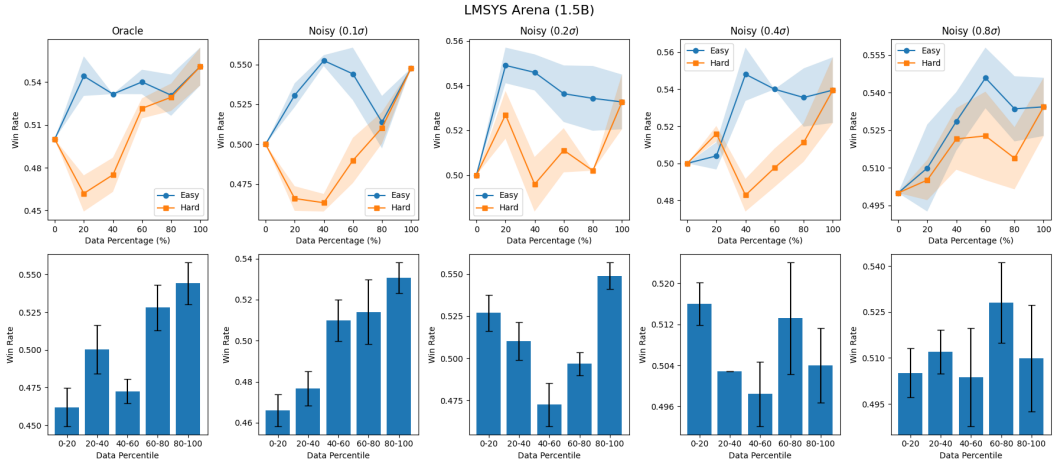


(a) Win rate

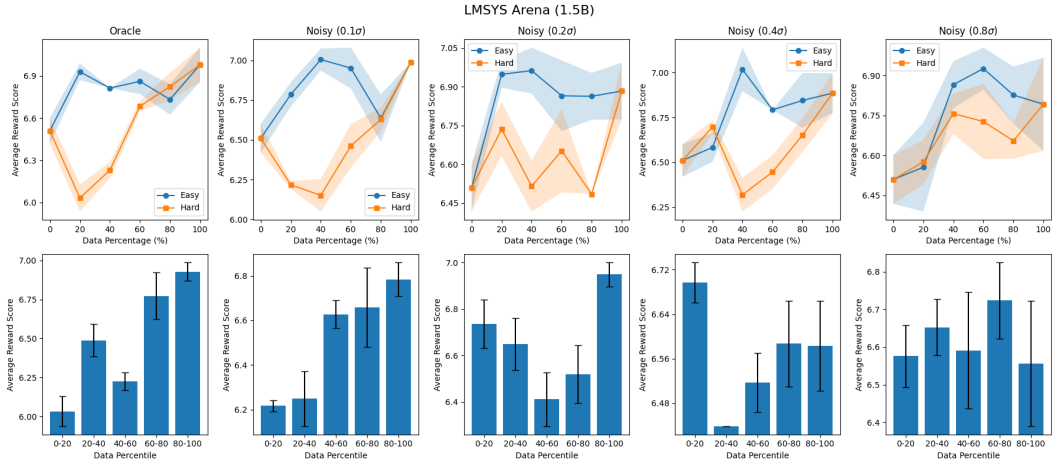


(b) Average reward score

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 oracle 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.

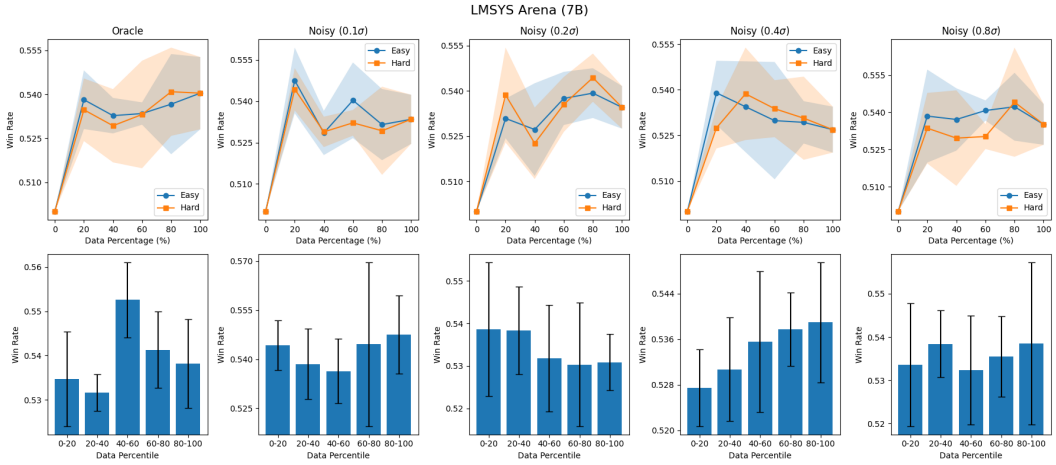


(a) Win rate

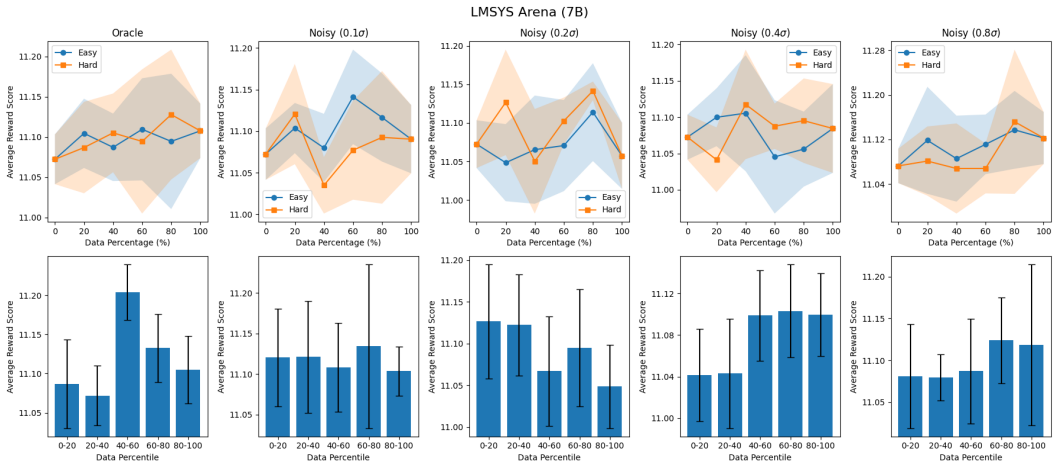


(b) Average reward score

Figure 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 oracle 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.

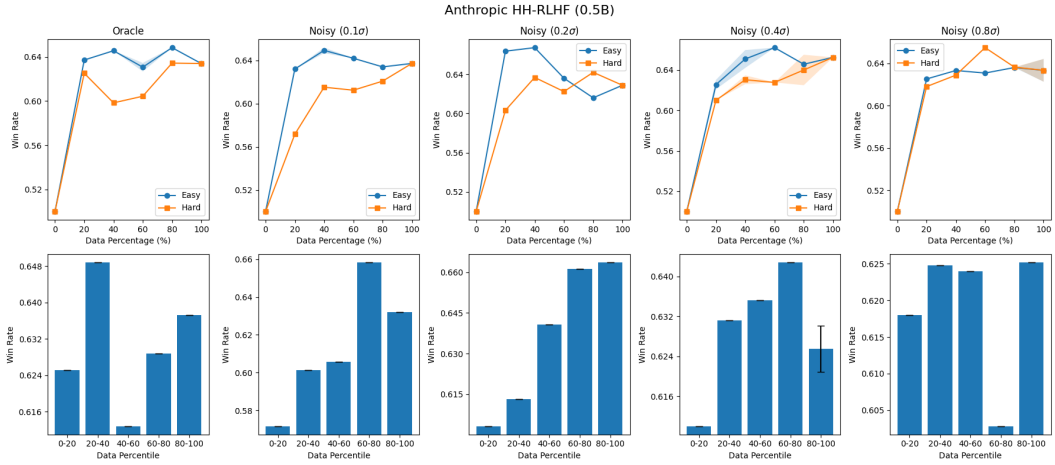


(a) Win rate

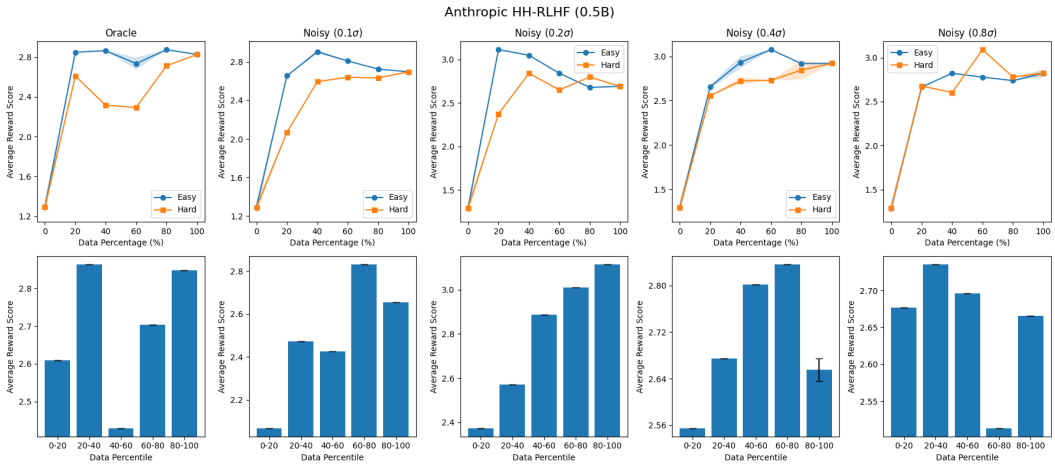


(b) Average reward score

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 oracle 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.

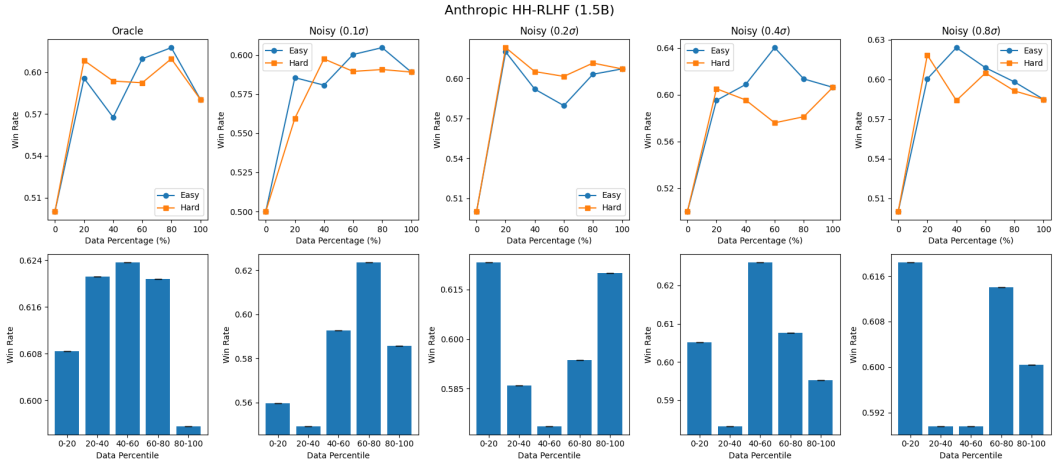


(a) Win rate

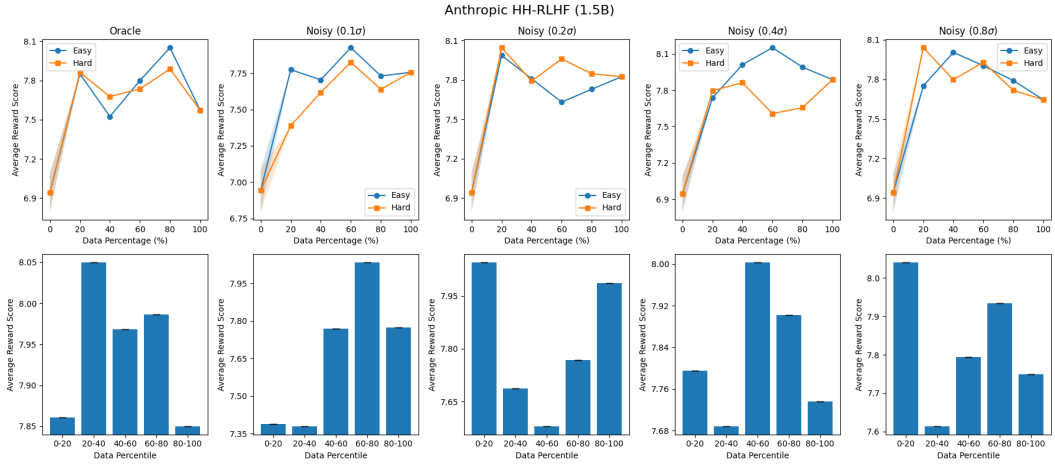


(b) Average reward score

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 oracle 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.

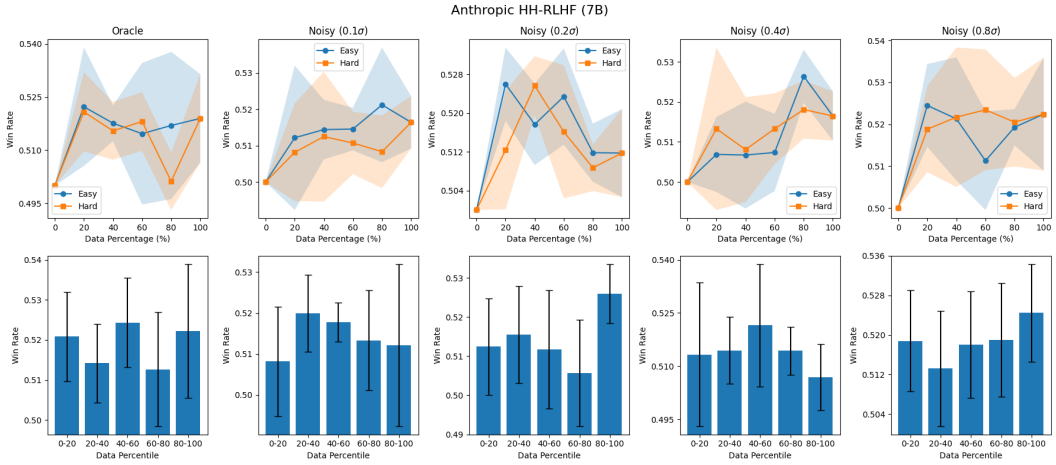


(a) Win rate

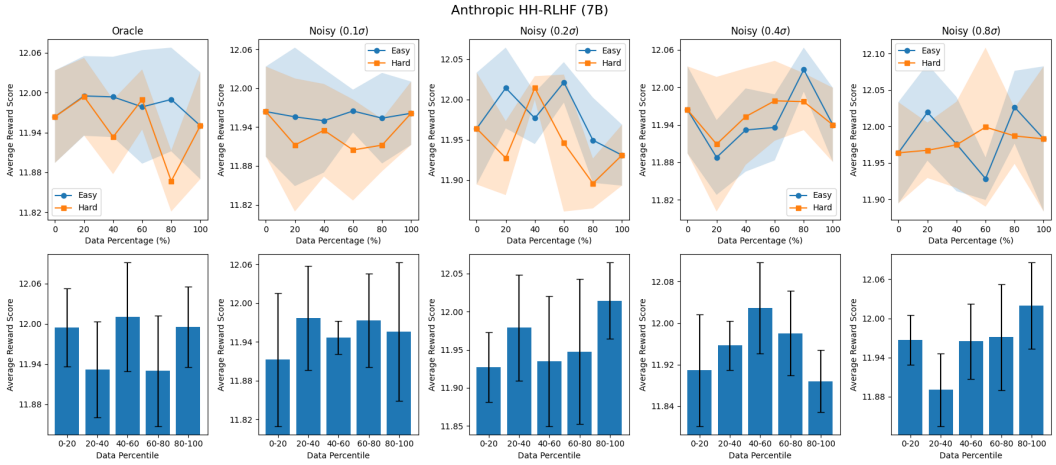


(b) Average reward score

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 oracle 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.

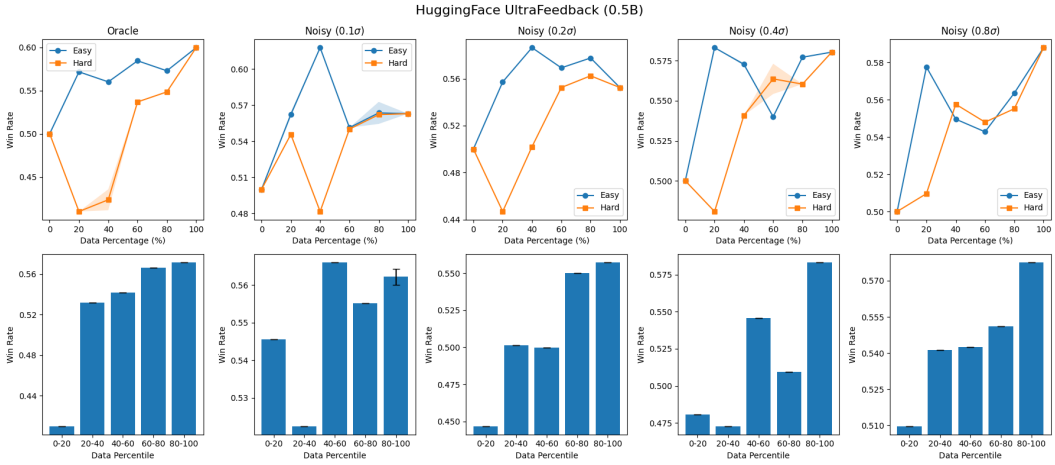


(a) Win rate

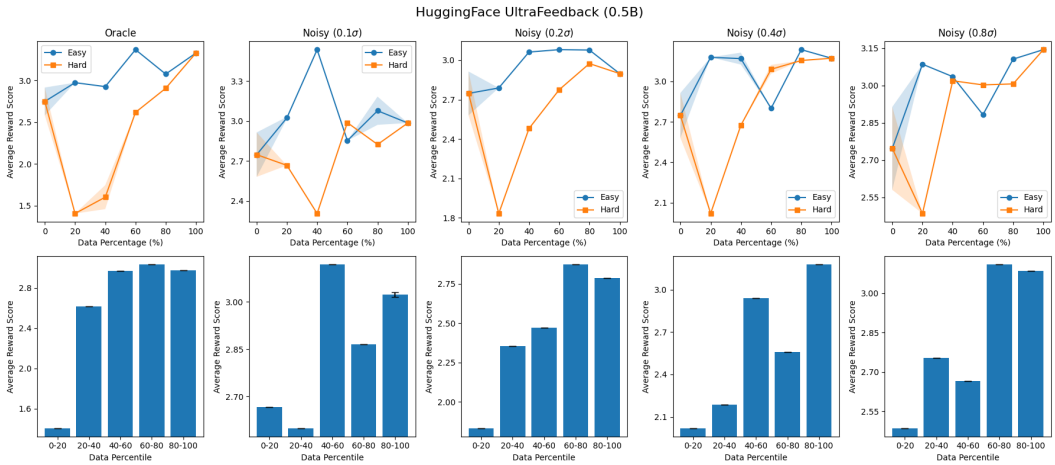


(b) Average reward score

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 oracle 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.

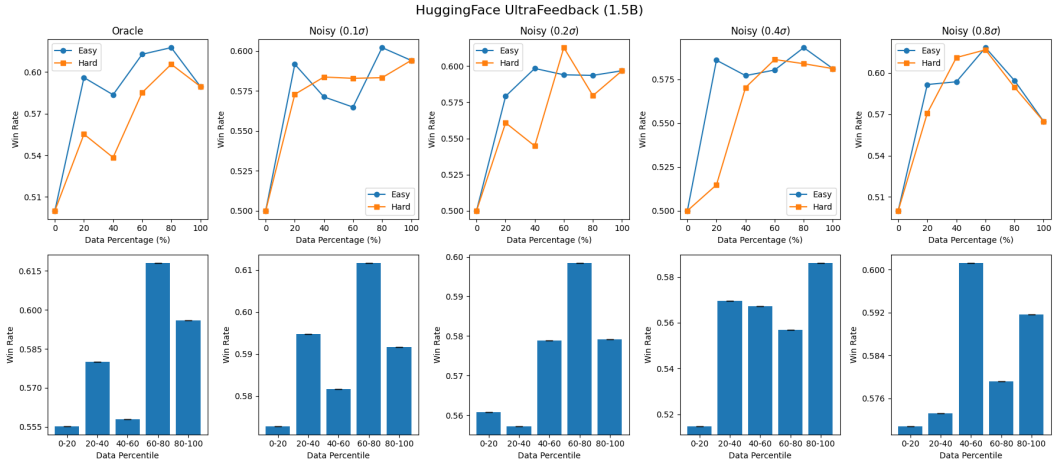


(a) Win rate

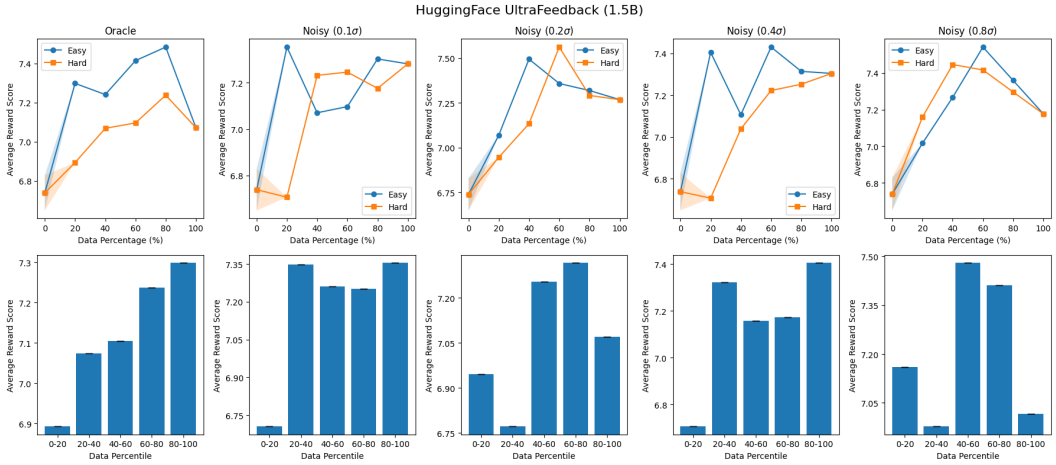


(b) Average reward score

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 oracle 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.

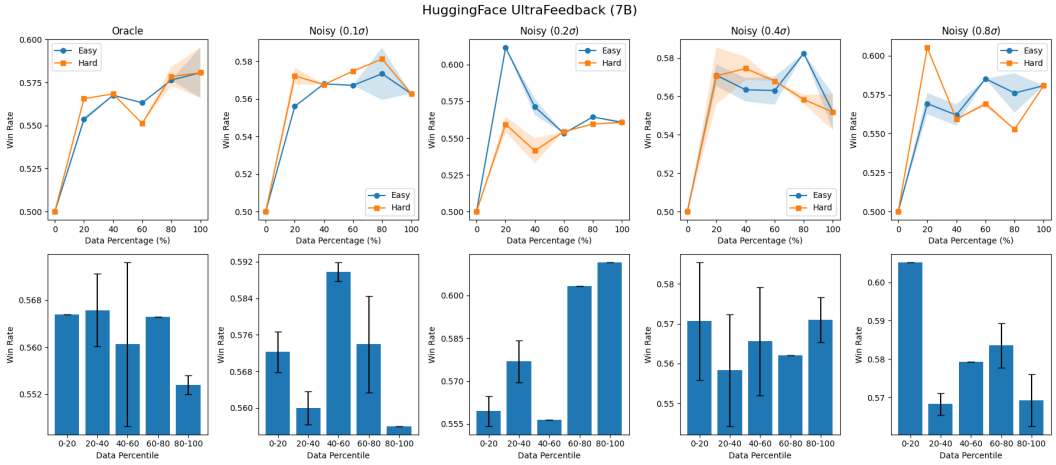


(a) Win rate

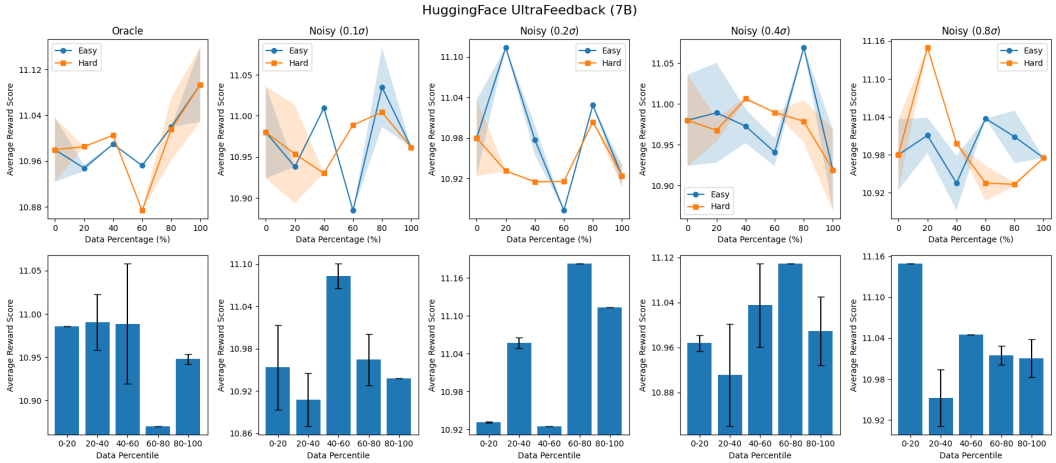


(b) Average reward score

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 oracle 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

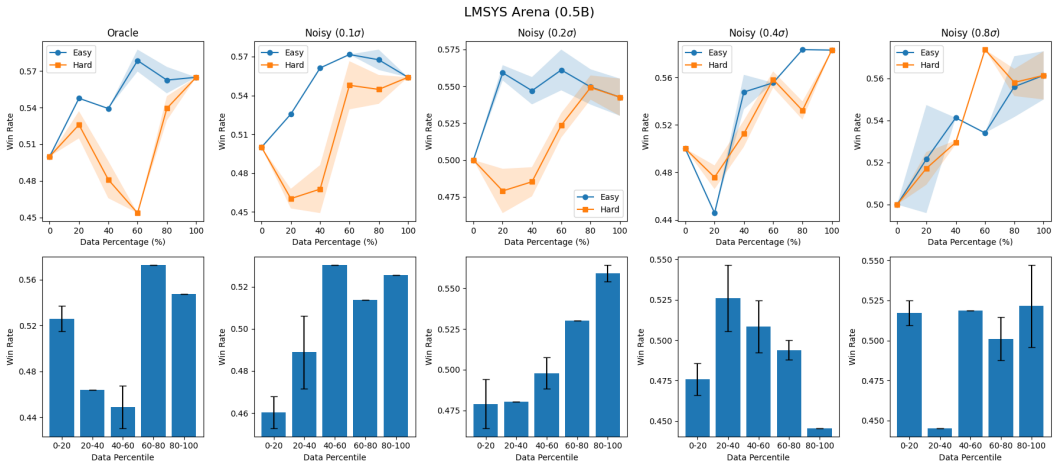


(b) Average reward score

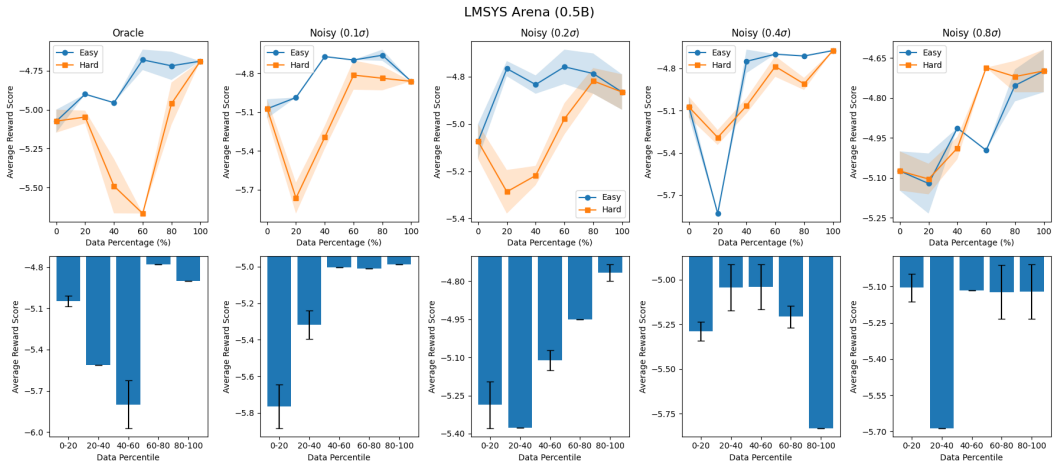
Figure 36: GRPO win rate and average reward score on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups, using Qwen2.5-7B-Instruct base model and Skywork oracle RM. Blue and orange lines correspond to training with the easiest and hardest examples, respectively.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

A.7 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 oracle RM. For a discussion and interpretation of these results, see Section 4.4.

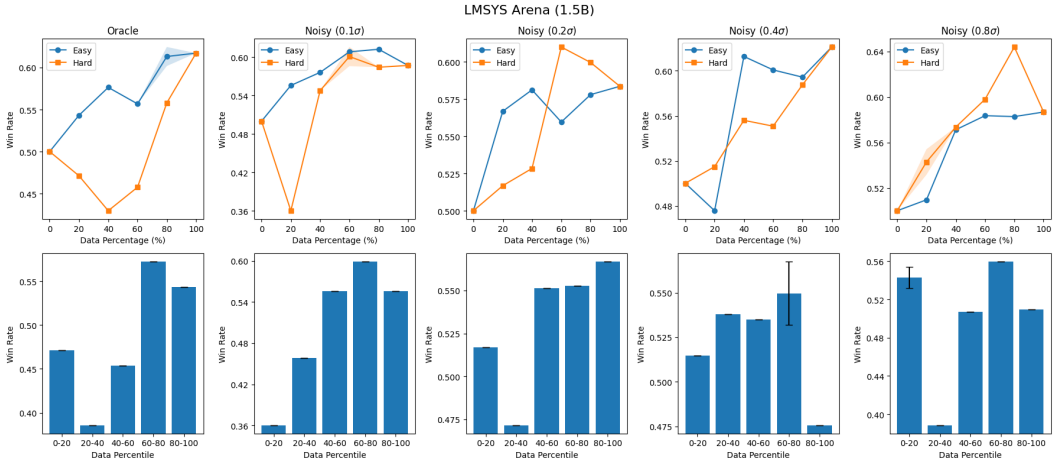


(a) Win rate

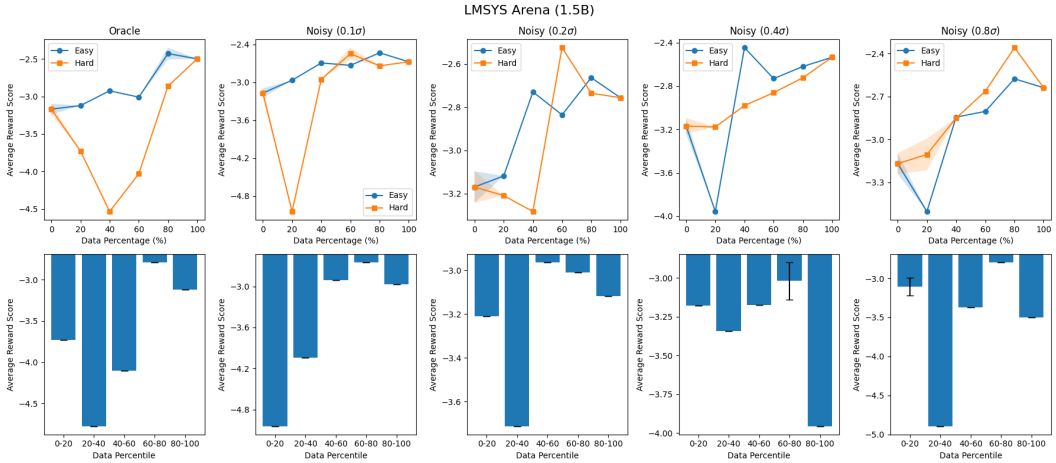


(b) Average reward score

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 oracle 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.

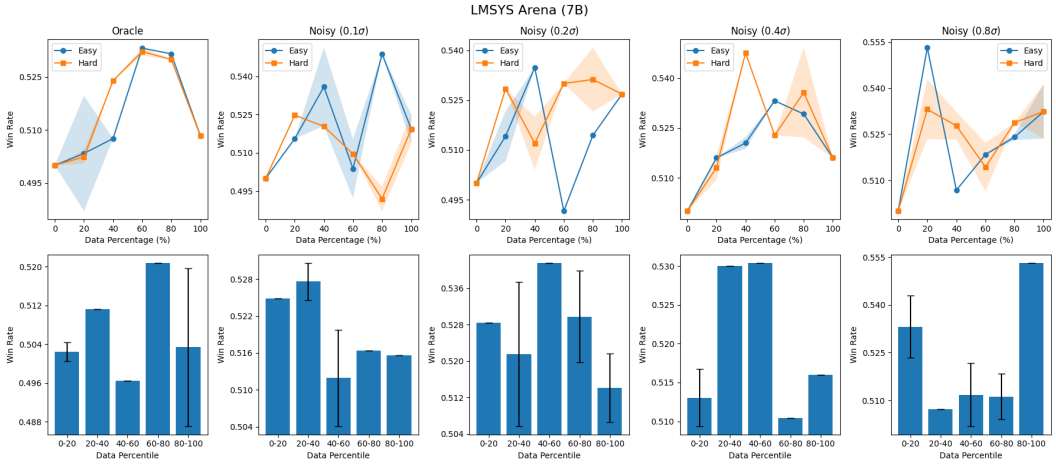


(a) Win rate

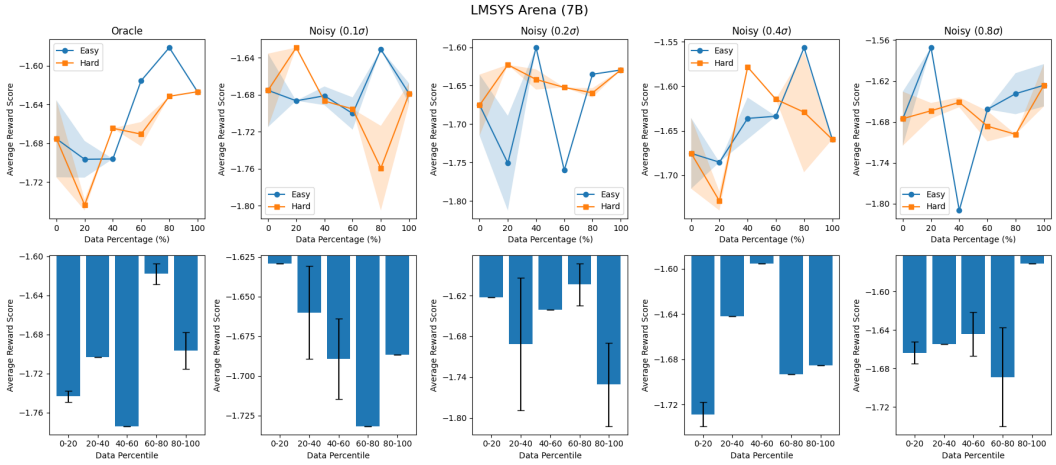


(b) Average reward score

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 oracle 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.

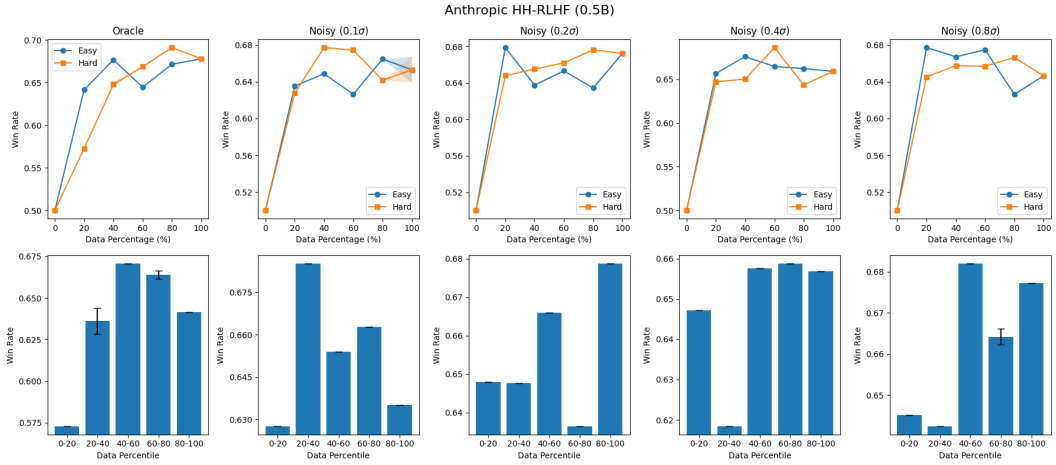


(a) Win rate

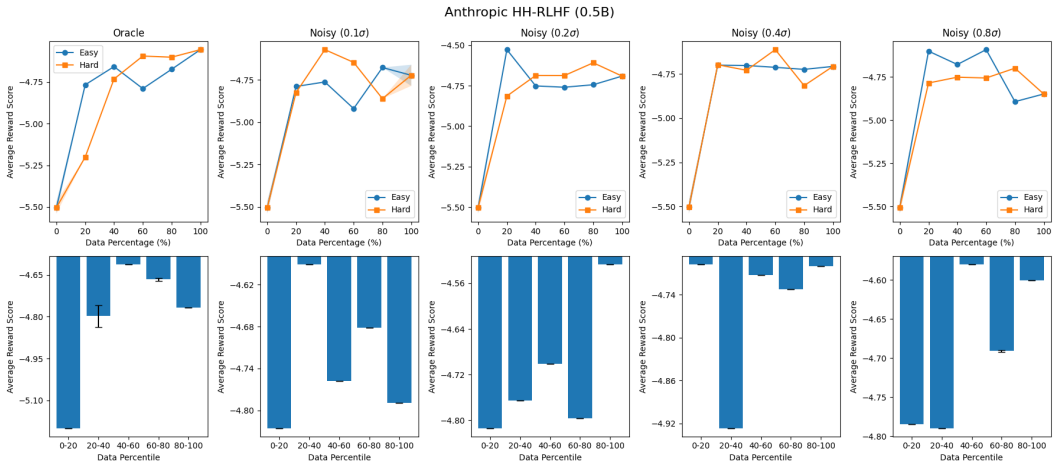


(b) Average reward score

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 oracle 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.

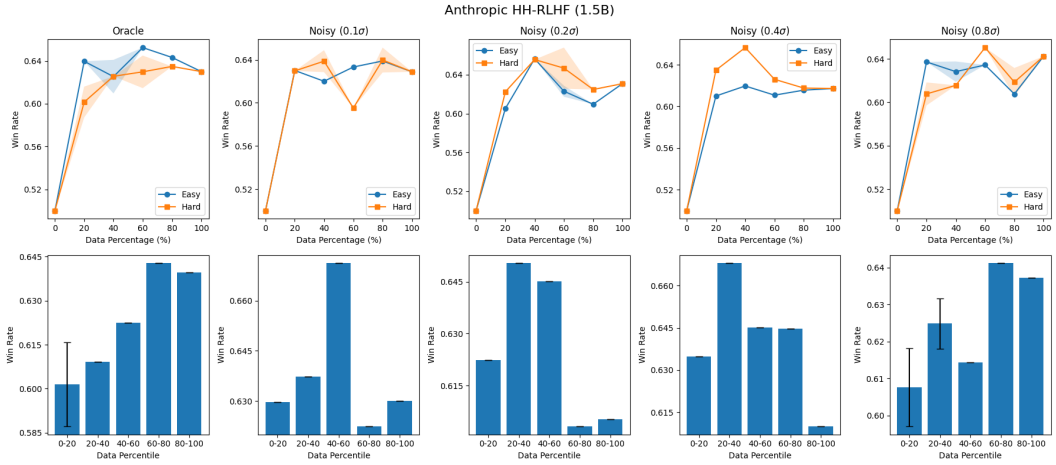


(a) Win rate

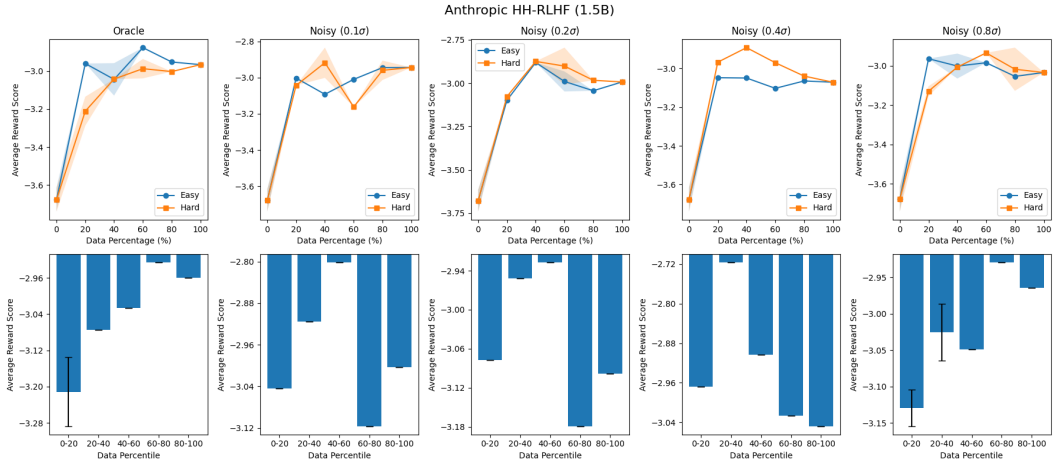


(b) Average reward score

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 oracle 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.

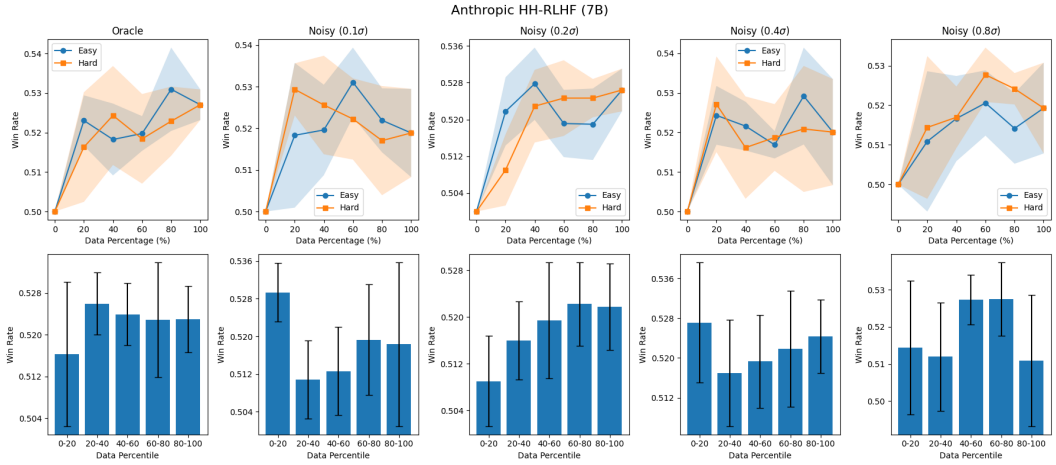


(a) Win rate

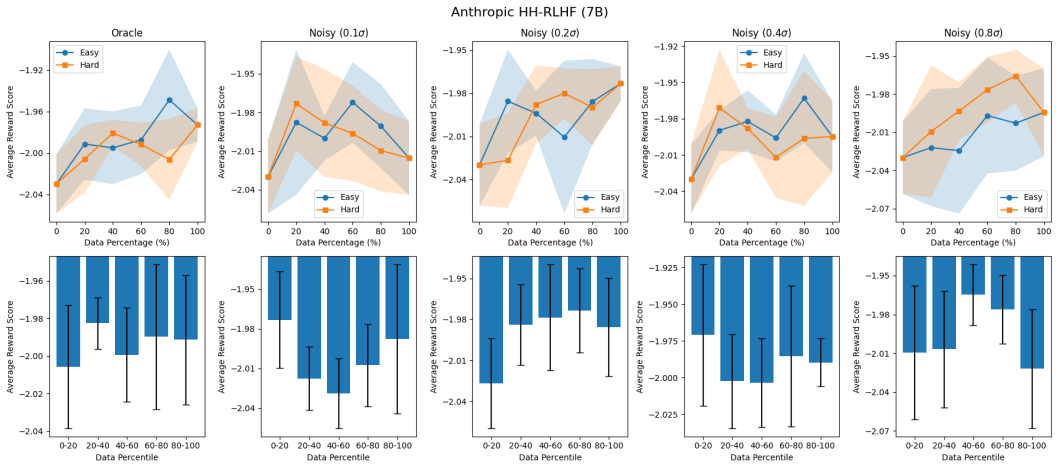


(b) Average reward score

Figure 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 oracle 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.

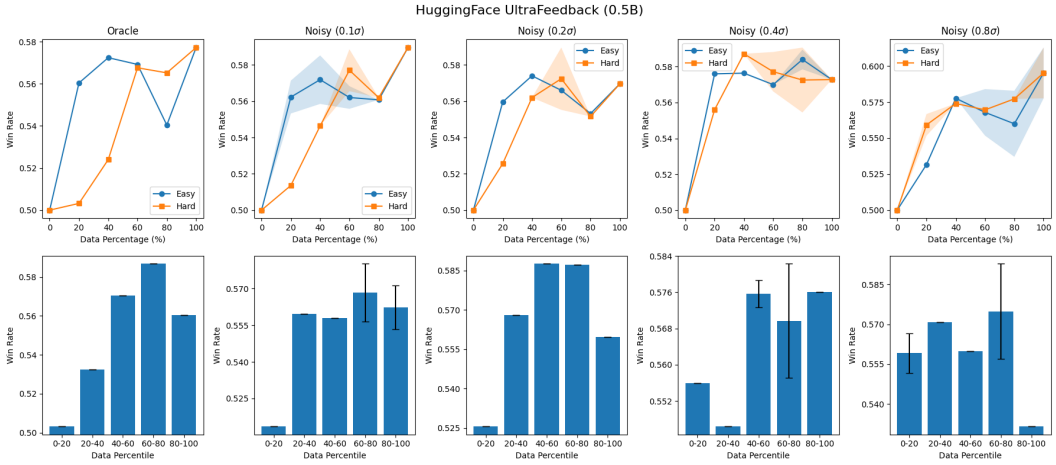


(a) Win rate

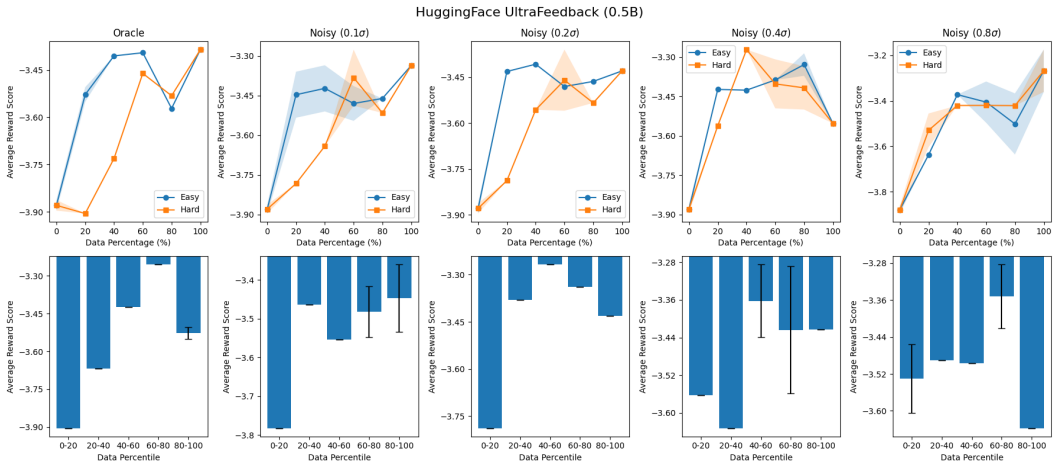


(b) Average reward score

Figure 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 oracle 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.

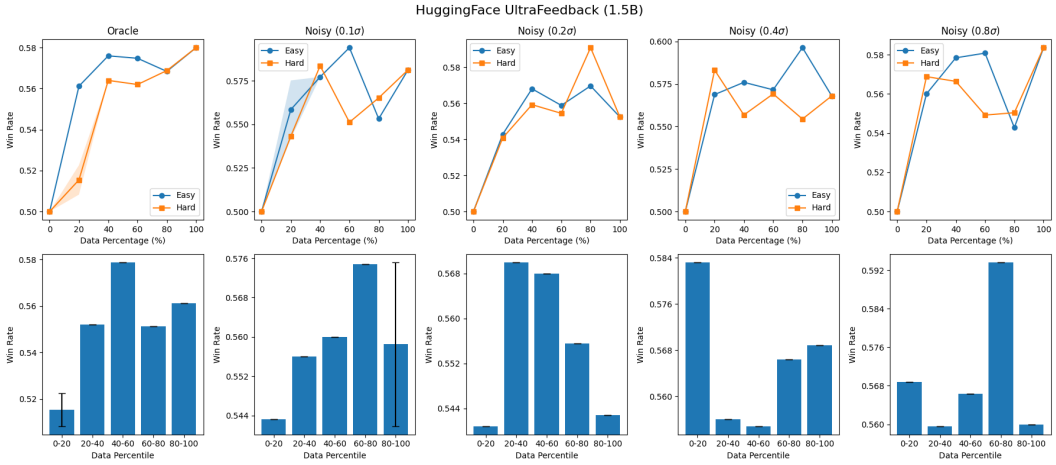


(a) Win rate

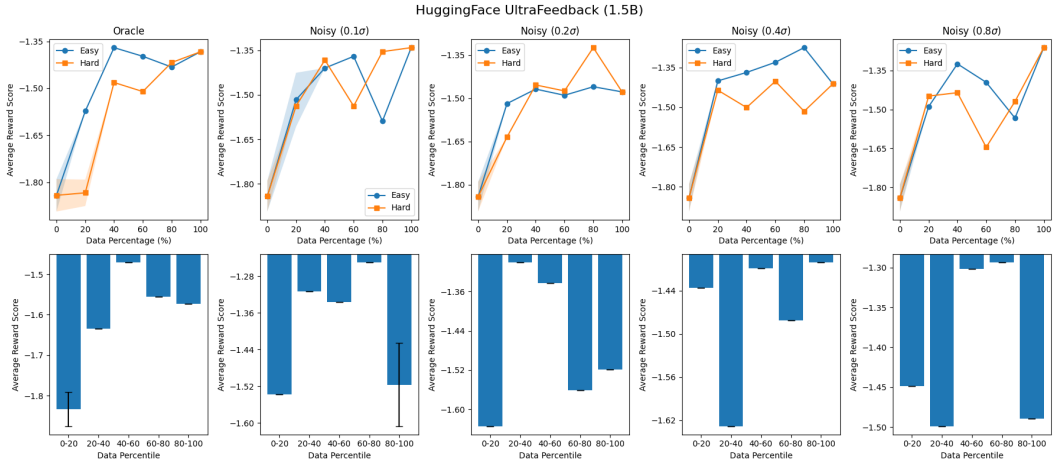


(b) Average reward score

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 oracle 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.

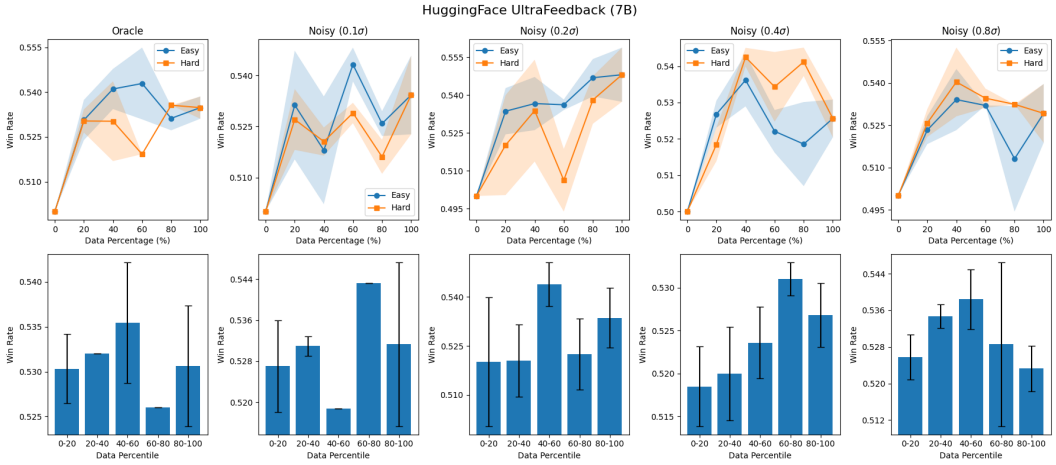


(a) Win rate

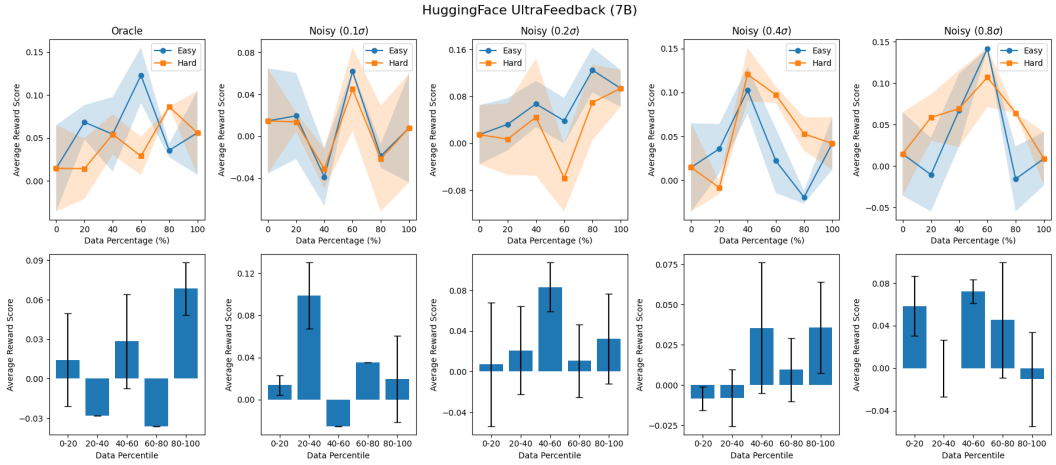


(b) Average reward score

Figure 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 oracle 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 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 oracle 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 *Data Selection with Weak Proxy RMs* DETAILS

B.1 PROXY RM ACCURACY

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

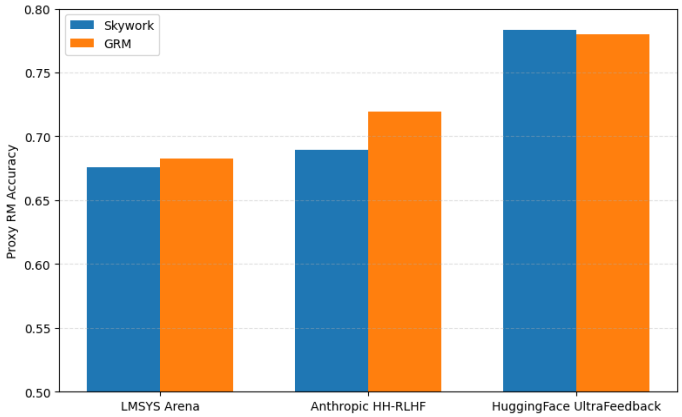
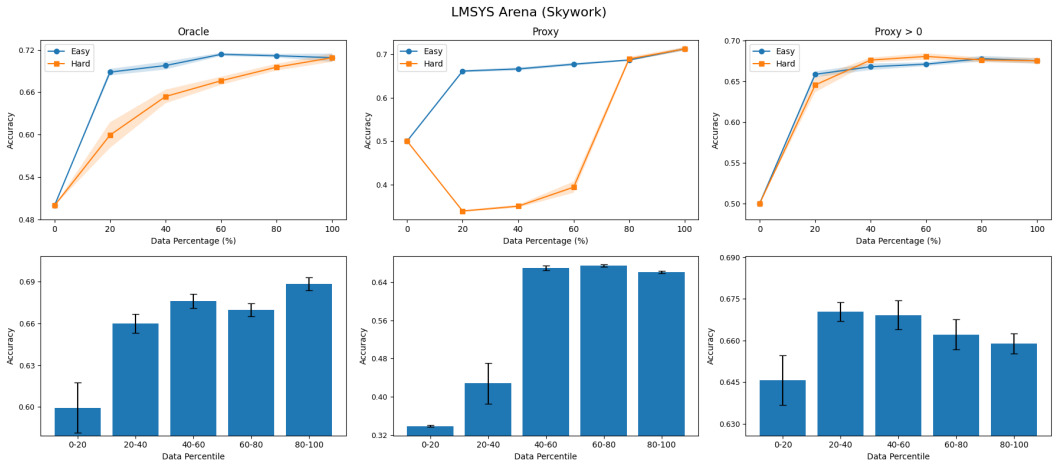


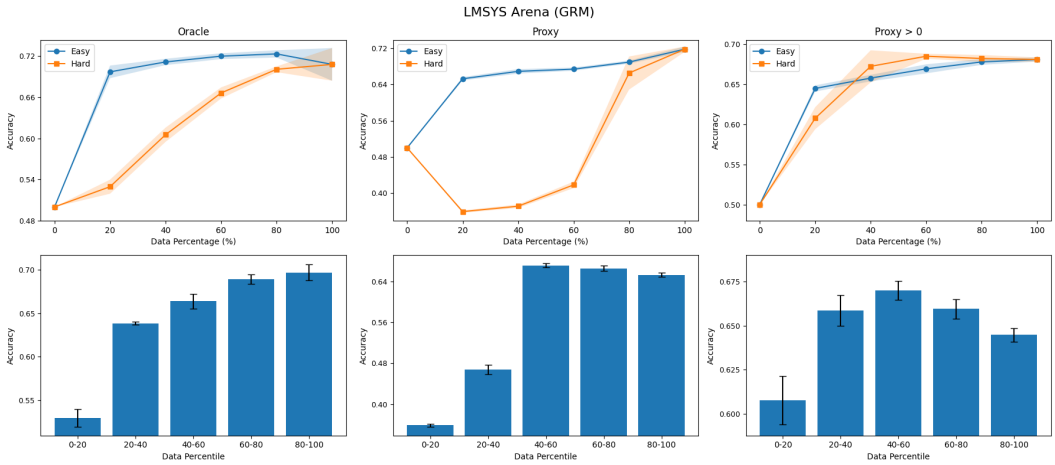
Figure 46: Test set accuracy of the Pythia-1B proxy RM across datasets and oracle RMs. The proxy achieves 67% to 78% accuracy depending on dataset and oracle. Here, Skywork refers to Skywork-Reward-V2-Qwen3-4B and GRM refers to GRM-gemma2-2B-rewardmodel-ft.

B.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.

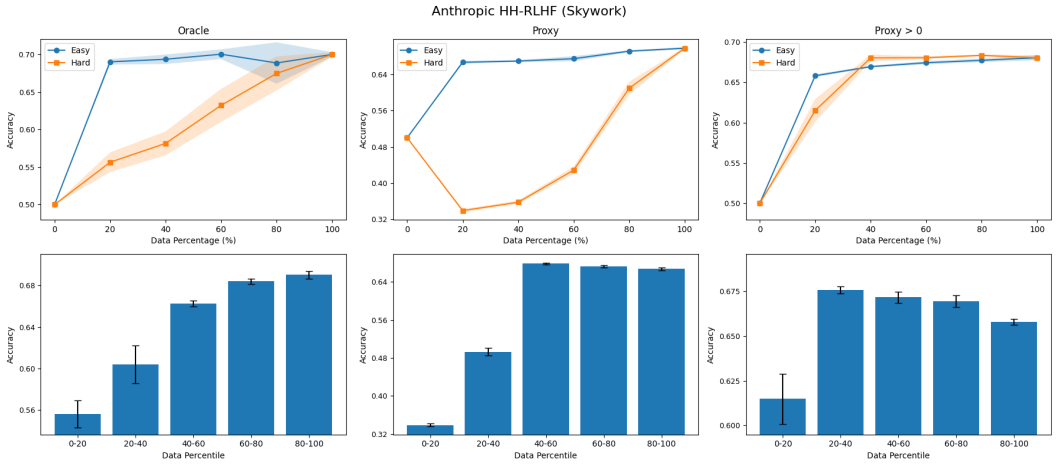


(a) Skywork reward model

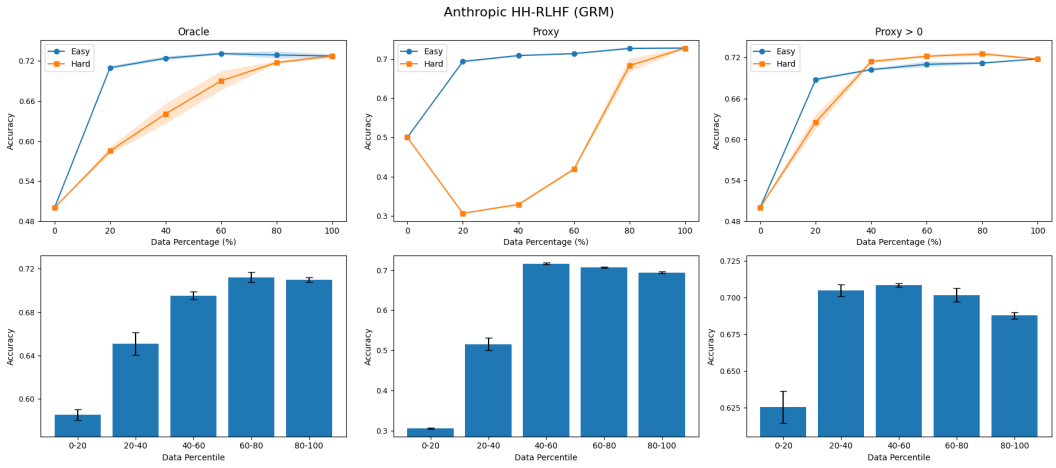


(b) GRM reward model

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 oracle 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.

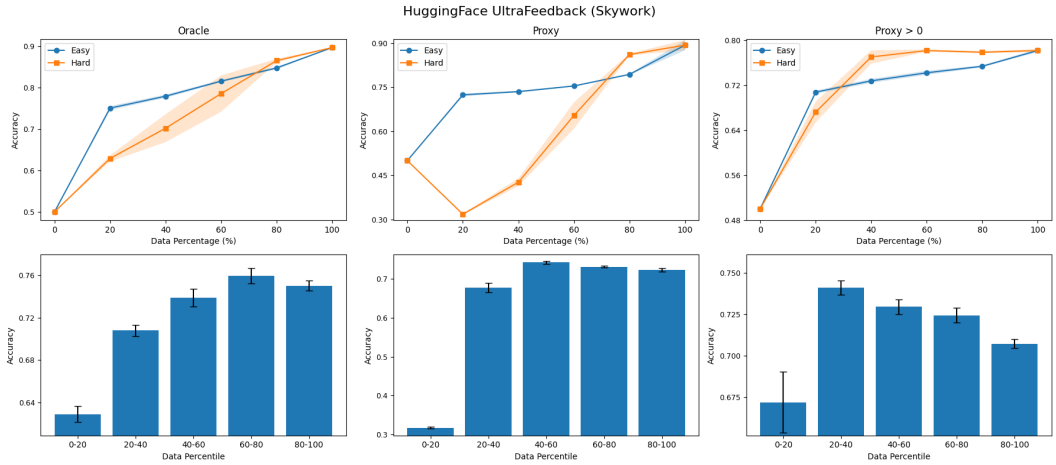


(a) Skywork reward model

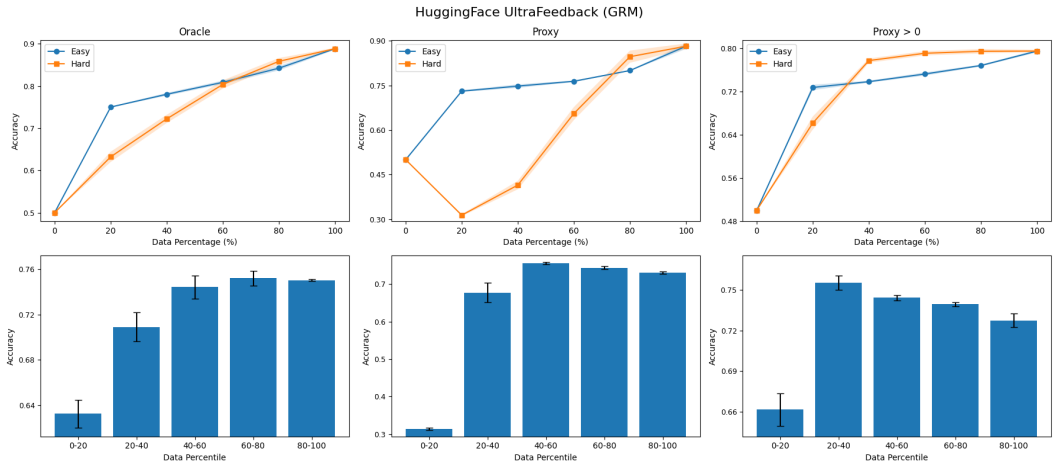


(b) GRM reward model

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 oracle 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

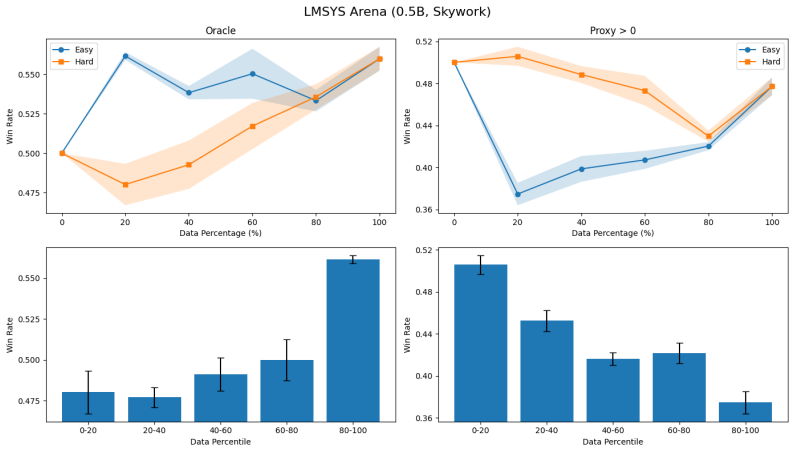


(b) GRM reward model

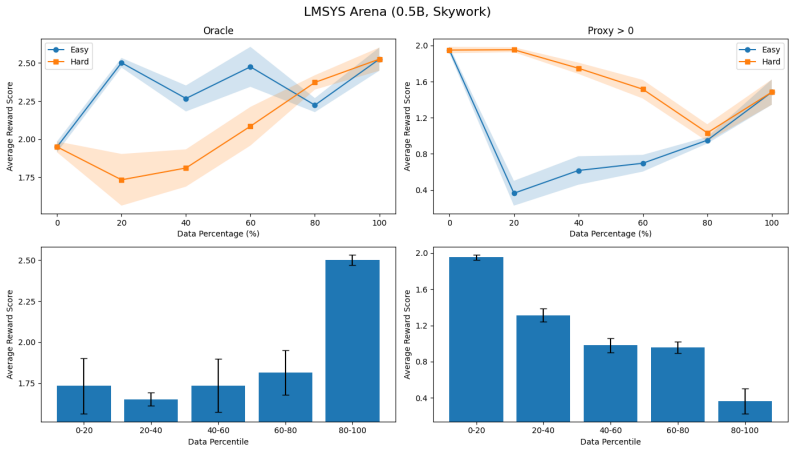
Figure 49: RM accuracy on the HuggingFace UltraFeedback dataset across difficulty-based subsets and training setups. The top and bottom panels correspond to the Skywork and GRM oracle RMs, respectively. Results are averaged over five runs, with error bars indicating mean $\pm$ standard deviation.

B.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.

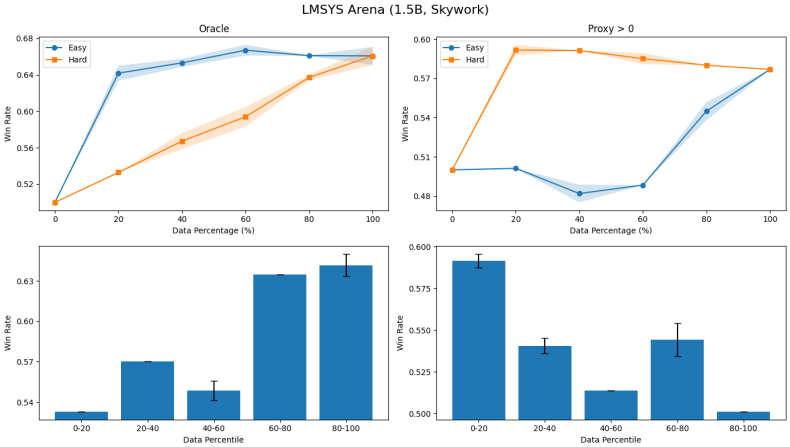


(a) Win rate

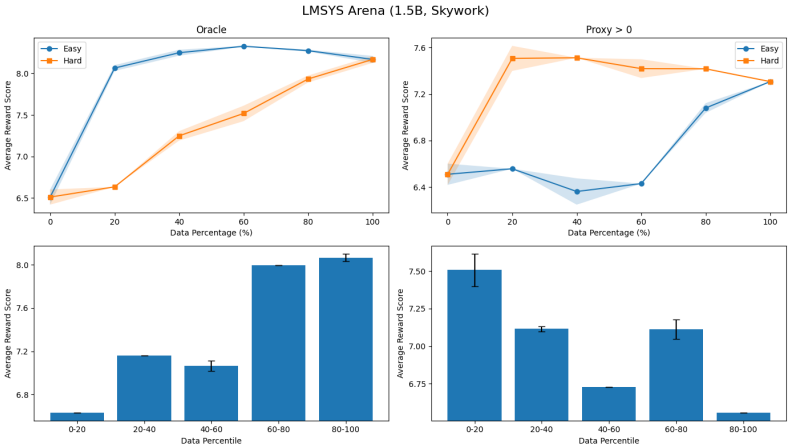


(b) Average reward score

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 oracle 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.

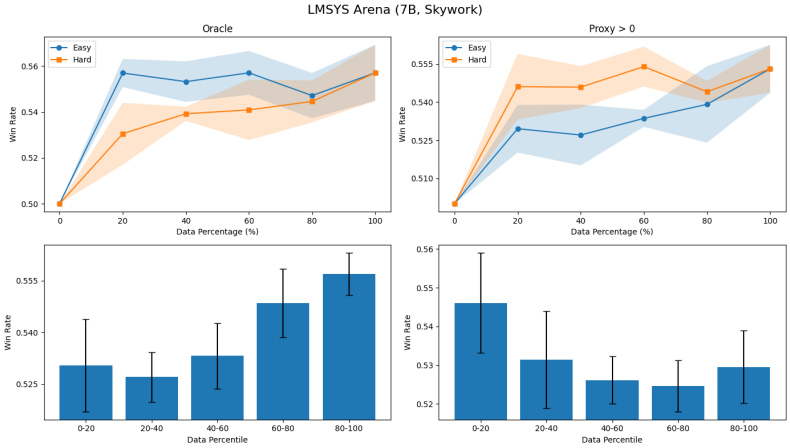


(a) Win rate

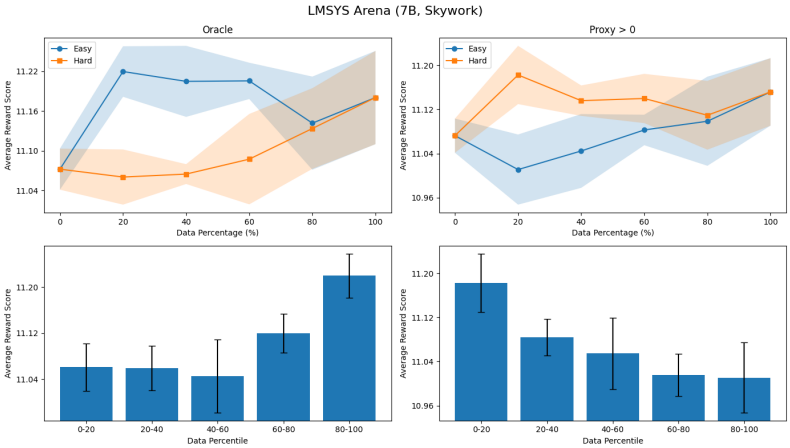


(b) Average reward score

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 oracle 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.

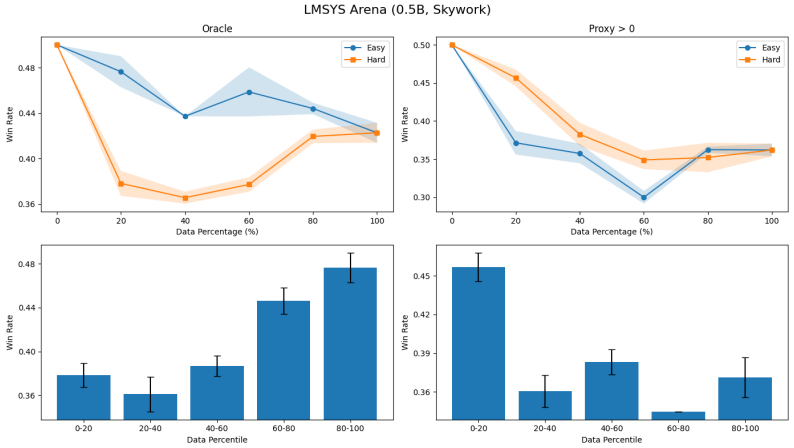


(a) Win rate

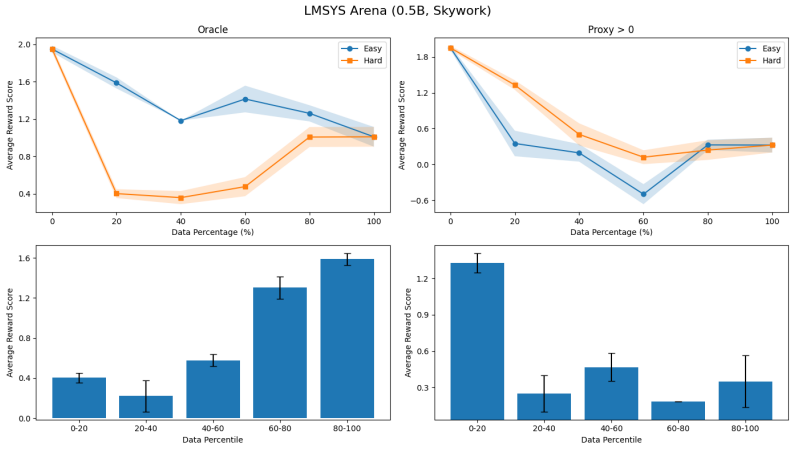


(b) Average reward score

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 oracle 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.

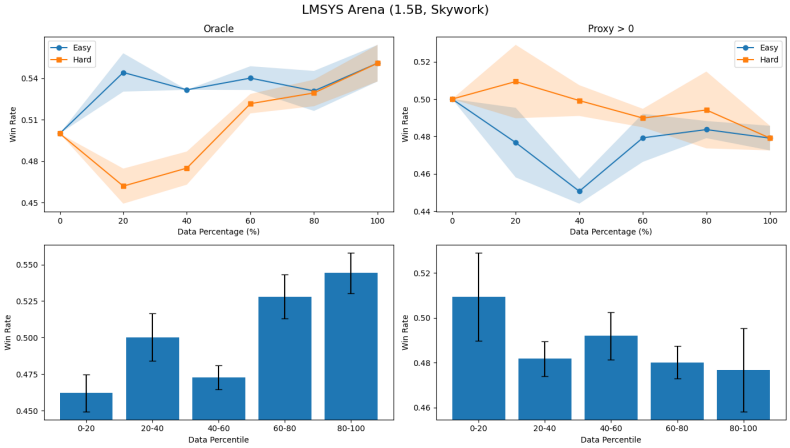


(a) Win rate

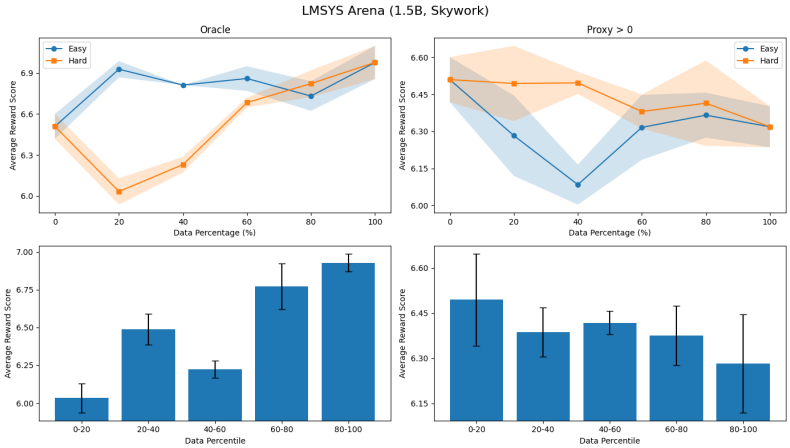


(b) Average reward score

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 oracle 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.

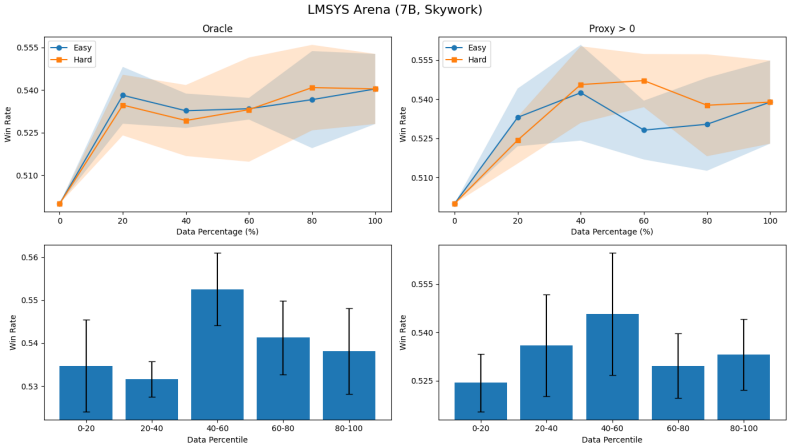


(a) Win rate

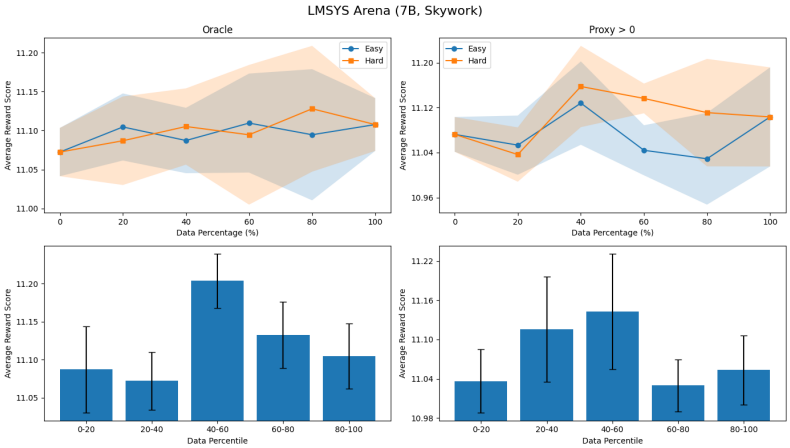


(b) Average reward score

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 oracle 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 oracle 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.