A Practical Analysis of Human Alignment with *PO

Kian Ahrabian[1]* Xihui Lin[2] Barun Patra[2]

Vishrav Chaudhary[2] Alon Benhaim[2] Jay Pujara[1]

Xia Song[2]

[1]University of Southern California, Information Sciences Institute
[2]Microsoft
ahrabian@usc.edu,{xihlin,barun.patra@microsoft.com}
{vchaudhary,alonbenhaim}@microsoft.com,jpujara@isi.edu,xiaso@microsoft.com

Abstract

At the forefront of state-of-the-art human alignment methods are preference optimization methods (*PO). Prior research has often concentrated on identifying the best-performing method, typically involving a grid search over hyperparameters, which can be impractical for general practitioners. In this paper, we examine the robustness of existing state-of-the-art methods to varying hyperparameters in a realistic out-of-distribution (OOD) scenario that mirrors real-world applications of human alignment. Our goal is to empirically find the method that increases the likelihood of achieving better results through the lens of various metrics, such as KL divergence and response length. We also introduce LN-DPO, a simple length-normalized version of DPO that is more stable across hyperparameters, effectively reduces the average response length, and improves performance. Our analysis of state-of-the-art reference-free (*i.e.,* SimPO) and reference-dependent (*i.e.,* DPO and LN-DPO) methods reveals that they perform similarly at their peak (*i.e.,* best possible scenario). However, we uncover that the pattern of change in performance greatly varies as we move away from the best possible scenario.

1 Introduction

In recent years, the quality of large language models (LLMs) has been constantly increasing (Chiang et al., 2024), achieving impressive results across tasks and benchmarks (Abdin et al., 2024; AI@Meta, 2024; Achiam et al., 2023; Team, 2023; Yang et al., 2024). However, even with the most rigorous filtering heuristics, the training data (Computer, 2023; Penedo et al., 2024) is typically contaminated with undesirable content that can lead to unacceptable behaviors (Bender et al., 2021; Gehman et al., 2020). To improve the model's alignment with human preferences, the de-facto approach has been to learn from human/AI-generated preference data (*e.g.,* a chosen and a rejected response for each prompt). In particular, off-policy preference optimization methods (*PO) have been prevalent given their good performance and ease of implementation (Rafailov et al., 2024; Hong et al., 2024; Meng et al., 2024).

	DPO	LN-DPO	SimPO
Mean Score	1.6	+0.3%	+2.7%
Mean Length	119.8	-15.9%	-22.9%
KL Divergence	55.0	-26.0%	-20.7%
Win vs. Chosen	77.1%	+0.8%	+3.1%
Win vs. SFT	60.7%	+2.1%	+5.0%

Table 1: **Best *PO Performance**. The metrics are normalized by the respective DPO performance. The underlined values indicate the best performance.

One commonly occurring practice when reporting the performance of new methods is to compare their best-performing variant (after a hyperparameter grid search) to a default baseline with a fixed set of hyperparameters. However, from a practical perspective for future users, these comparisons do not provide a good answer to the problem of which method is expected to achieve higher performance, given a fixed budget for hyperparameter search, as doing broad grid searches is often computationally infeasible for many practitioners. To this end, in this work, we aim to empirically identify the more robust method to hyperparameter variations while still being competitive in performance.

We set up our experiments in a realistic out-of-distribution (OOD) setting, focused on safety and helpfulness domains, where the train and test datasets share a common core goal, but their samples are generated from different distributions (*e.g.,* AI and human expert). This setting resembles real-

[*] Work done during an internship at Microsoft.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 8028–8036
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

Method	Objective	Hyperparameters
DPO	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\text{ref}}(y_w\mid x)} - \beta\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\text{ref}}(y_l\mid x)}\right)$	$\beta \in \{0.01, 0.05, 0.1, 0.3, 0.5\}$
SimPO	$-\log\sigma\left(\frac{\beta}{\lvert y_w\rvert}\log\pi_\theta(y_w\mid x) - \frac{\beta}{\lvert y_l\rvert}\log\pi_\theta(y_l\mid x) - \gamma\right)$	$\beta \in \{1.0, 1.5, 2.0, 2.5\}$ $\gamma \in \{0.5, 0.8, 1.0, 1.2, 1.4, 1.6\}$
LN-DPO	$-\log\sigma\left(\frac{\beta}{\lvert y_w\rvert}\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\text{ref}}(y_w\mid x)} - \frac{\beta}{\lvert y_l\rvert}\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\text{ref}}(y_l\mid x)}\right)$	$\beta \in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5\}$

Table 2: ***PO Optimization Objectives.** The preference data is formulated as $D = (x, y_w, y_l)$, where x is the prompt and y_w and y_l are the chosen and rejected responses.

world scenarios as it simulates the release of large generative models for public use. Moreover, to better understand the behavior of the state-of-the-art models, we take the best-performing reference-free and reference-dependent models (as reported by Meng et al. (2024)) and analyze them through the lens of standard metrics such KL divergence, response length, and win rate. We also introduce an embarrassingly simple length-normalized extension of vanilla Direct Preference Optimization (DPO) (Rafailov et al., 2024), LN-DPO, that effectively mitigates the issue of lengthy generations without any apparent performance degradation[1]. In summary, our contributions are as follows:

- We examine state-of-the-art reference-free and reference-dependent preference optimization methods across a wide range of hyperparameters in a real-world setup.
- We analyze the performance of these methods on critical metrics such as mean response length, mean score on a gold reward model, win rate vs. chosen and SFT, and KL vs. SFT.
- We introduce and examine LN-DPO, a simple length-normalized version of DPO that is more stable across hyperparameters, effectively reduces the average response length and improves performance.

2 Related Work

Since the introduction of DPO (Rafailov et al., 2024), there has been a body of works with new optimization objectives improving the performance and efficiency (Azar et al., 2024; Tang et al., 2024; Hong et al., 2024; Rosset et al., 2024; Meng et al., 2024; Xu et al., 2024a; Ethayarajh et al., 2024). These methods can be partitioned into two groups: reference-free (Meng et al., 2024; Hong et al., 2024) and reference-dependent (Rafailov et al., 2024; Park et al., 2024). Reference-free methods generally benefit from fast training runs, while reference-dependent methods have terms baked into their objective to control divergence from the reference model. In this work, we compare SimPO (Meng et al., 2024), a recent state-of-the-art reference-free method, with DPO and LN-DPO as reference-dependent methods (see Appendix A for extended related work).

3 Experimental Setup

3.1 Datasets

For our datasets, we follow the setup introduced by Xu et al. (2024b). Specifically, we use the double safe/unsafe filtered train subset of SafeRLHF (Dai et al., 2024) for training and the test subset of HH-RLHF (Ganguli et al., 2022) for evaluation. This setup closely resembles real-world scenarios where even though models are trained on various domains (*e.g.,* safety and helpfulness in our experiments), they have to generalize to similar unseen queries while interacting with the users.

3.2 Models

For all our experiments, we chose the Phi-3 Medium model (Abdin et al., 2024) due to its high performance across benchmarks and small size, ensuring computational tractability. To evaluate the trained models, we use the OpenAssistant reward model (Köpf et al., 2024) to score the quality of their generated responses. We chose this model due to its small size and use in prior works (Xu et al., 2024b), ensuring fast and correct evaluations.

3.3 Optimization Objectives

Considering the performances reported by Meng et al. (2024), we choose DPO as our reference-dependent method and SimPO as our reference-free

[1] Concurrently, Meng et al. (2024) have added a similar method to their experiments (updated on July 7th, 2024). Here, we present a more thorough analysis and comparison.

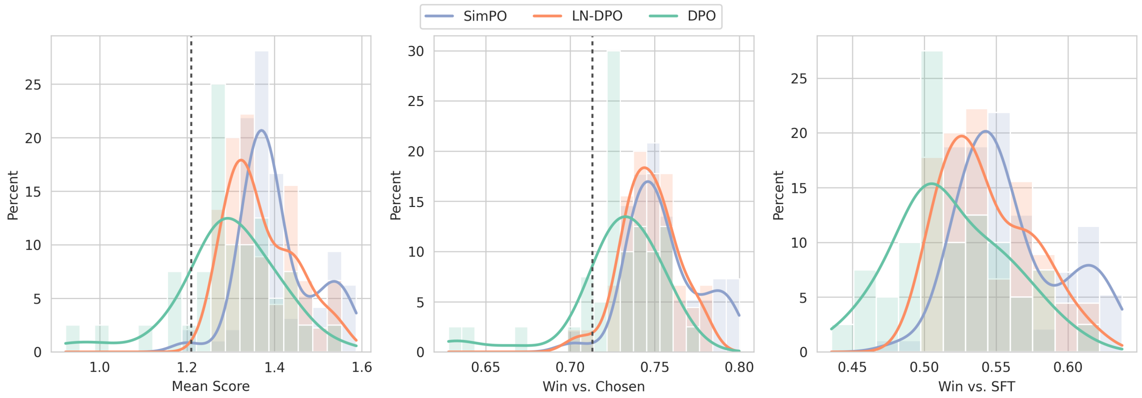

Figure 1: ***PO Performance Distribution**. Each sample in the distribution represents the performance of one set of hyperparameters on the denoted metric. The dashed line indicates the performance of the initial SFT model.

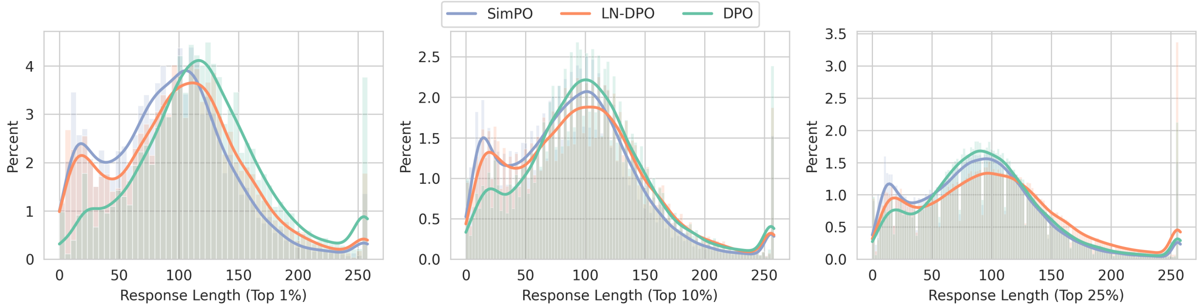

Figure 2: **Response Length**. The top k% ($k \in \{1, 10, 25\}$) denotes the percentage of best-performing hyperparameters taken from each method's runs.

method. While DPO has an implicit length normalization through the reference model, the variance of the reward (*i.e.,* $\log \frac{\pi_\theta}{\pi_{\text{ref}}}$) increases with response length. As such, inspired by explicit length regularization in SimPO and R-DPO (Park et al., 2024), we further normalize it with the response length similar to SimPO, which we call LN-DPO (see Section 3.4 for more details).

3.4 Connection between LN-DPO and SimPO

LN-DPO is similar to an adaptive margin version of SimPO with per sample margin defined as

$$\gamma_{w,l} = \log \frac{\pi_{\text{ref}}(y_w|x)}{|y_w|} - \log \frac{\pi_{\text{ref}}(y_l|x)}{|y_l|} \, . \quad (1)$$

Essentially, this adaptive margin encourages larger margins for pairs with large margins in the reference policy. Depending on the quality of the reference model and the labels, this change could be beneficial compared to SimPO's constant margin. The adaptive margin focuses more on "easier" pairs (*i.e.,* pairs that have some prior evidence to be different) while less on "harder" pairs (*i.e.,* pairs that are closer), which means that LN-DPO is potentially less prone to overfitting and less sensitive to wrong labels.

4 Training Regimen

Following the common practice, before the preference optimization step we do a supervised finetuning (SFT) step. Specifically, we first run a grid search over the following hyperparameters: epochs $\in \{1, 3\}$ and learning rate $\in \{1e-6, 3e-6, 1e-5, 2e-5\}$. Then we evaluate the final checkpoints against the test set and choose the one with the highest performance. This procedure ensures that the preference optimization methods are initialized from a good checkpoint. For the preference optimization methods, we run a grid search using 1) the same ranges as SFT for epochs and learning rate and 2) common values for method-specific hyperparameters as used in prior works (Meng et al., 2024; Rafailov et al., 2024; Hong et al., 2024). Table 2 presents the method-specific ranges used in our experiments. In all of our experiments, the batch size is set to 256.

5 Metrics

Our analysis focuses on the following five metrics:

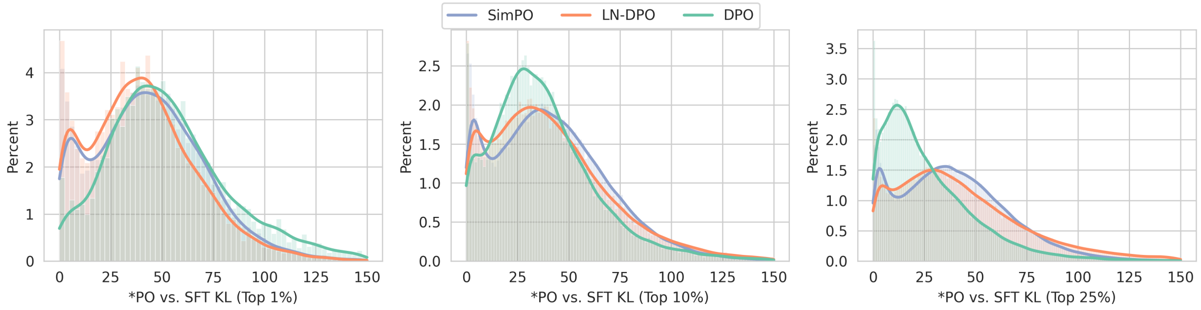

Figure 3: **KL Divergence**. The top k% ($k \in \{1, 10, 25\}$) denotes the percentage of best-performing hyperparameters taken from each method's runs.

- **Mean Score:** The average score of the generated responses, as judged by the gold reward model.
- **Win vs. Chosen:** The fraction of samples where the gold reward model assigns a higher score to the generated response compared to the chosen response in the dataset.
- **Win vs. SFT:** The fraction of samples where the gold reward model scores the generated response higher than the initial SFT model's response.
- **KL divergence:** The summed difference of log probabilities between the SFT and the trained models over the samples.
- **Response length:** The number of tokens in the generated response under the tokenization space of the base model.

6 Implementation Details

We generate all the responses by sampling with a `temperature` $= 0.7$, and `top_p` $= 0.95$. Moreover, `max_generation_length` is set to 256 across all experiments, following the setup by Xu et al. (2024b). All our experiments are carried out on a cluster with 256×A100 80GB GPUs. Finally, we implemented our code using the Transformers (Wolf et al., 2020), TRL (von Werra et al., 2020), and PyTorch (Paszke et al., 2019) libraries.

7 Experimental Results

7.1 Hyperparameter Robustness

Best Performance. Following the common practice, we compare the best performance achieved by each method in Table 1. As evident, at their peaks, SimPO, LN-DPO, and DPO score similarly (within a 0.05 point on average). However, SimPO and LN-DPO show an edge in terms of the rest of the metrics. Specifically, we can observe the effectiveness of the length normalization term. We also notice a significant decrease in KL divergence. However, KL for SimPO decreases less than LN-DPO, showcasing a more significant divergence from SFT. For more details on tuning these models, see Appendix B.

%	DPO	LN-DPO	SimPO
DPO	-	49.04	47.51
LN-DPO	49.47	-	46.43
SimPO	51.12	51.09	-

(a) Best

%	DPO	LN-DPO	SimPO
DPO	-	45.72	44.33
LN-DPO	51.77	-	47.28
SimPO	54.34	50.13	-

(b) 75th Percentile

Table 3: **Head-to-head *PO Comparison.** Each cell represents the win rate of the row method over the column method. The underlined values indicate the row method beating the column method.

Head-to-head Performance. While comparing the pure performances achieved on the desired metrics is usually good enough to contrast different methods, there are potential cases where the averaging could be exploited (*e.g.,* outliers with high rewards). Hence, it is essential also to do a head-to-head per sample comparison, which provides more fine-grained insights. Table 3 compares each method's best and 75th percentile performance. Notably, we observe a sharp performance drop in DPO from the best to the top 25% model, in contrast to the other two. This occurrence highlights the practical flaw in only comparing the best performances.

Expected Performance. Given the limited resources that most users have, it is extremely difficult to run broad hyperparameter searches to find the best-performing combination. As such, it becomes crucial to analyze hyperparameter robustness, which provides insights into the expectation of finding good hyperparameters set from a limited search. Figure 1 presents the performance distribution *PO methods following a grid search over the hyperparameters denoted in Table 2 and Section 4. As evident, SimPO and LN-DPO effectively increase the average performance (i.e., shifting the distributions to the right) across hyperparameters, showcasing their superiority. Note that we stretched the range of hyperparameters until a plateau or an extreme variance was observed.

7.2 Response Length

Since length exploitation is a critical issue (Park et al., 2024), we compare the response lengths across samples generated by the top k% ($k \in \{1, 10, 25\}$) of each method's best-performing hyperparameters. As illustrated in Figure 2, on the best set of hyperparameters (*i.e.,* top 1%), the non-DPO methods showcase a left shift in length distribution (compared to DPO), which is a desired effect. However, this phenomenon starts to diminish as we include worse-performing hyperparameters. For example, LN-DPO has a higher rate than DPO in the tail-end of the top 25% distribution. Overall, we observed that both length-normalized models perform superior to DPO, with SimPO producing the shortest responses across the distribution.

7.3 KL Divergence (vs. SFT)

Since reference-free methods are not normalized against a reference policy (*e.g.,* the SFT model), reward hacking might occur (*i.e.,* lower loss with degraded performance). Therefore, we compare the KL divergence in Figure 3 across samples generated by the top k% ($k \in \{1, 10, 25\}$) of each method's best-performing hyperparameters. As evident, both SimPO and LN-DPO achieve lower KLs at their peak. However, as we move toward worse-performing models, DPO achieves lower KL (at 10%). This phenomenon is due to many DPO runs failing to learn beyond the SFT model.

8 When to use LN-DPO over SimPO?

While SimPO achieves superior performance on most metrics compared to LN-DPO, the lack of a reference policy regularization could lead to drastic divergence from the initial checkpoint, as also shown in our experiments. This issue then could cause a degradation of performance on other benchmarks, which is a critical pitfall (as also observed in Korbak et al. (2022)). As such, we believe there are various scenarios where LN-DPO should be preferred to SimPO. We leave further experiments over this direction to future works.

9 Conclusion

In this work, we introduce LN-DPO, a length-normalized variation of DPO that reduces the average response length while staying reference-dependent. Moreover, we present a thorough analysis of LN-DPO and two state-of-the-art reference-dependent and reference-free preference optimization methods in a simulated real-world scenario for safety and helpfulness domains. Specifically, we cover the behavior of these methods across a wide range of hyperparameters under metrics such as mean response length, KL divergence (vs. SFT), and win rate (vs. chosen and SFT). Our experiments showcase state-of-the-art methods' strengths and weaknesses and provide insights for other practitioners.

Limitations

Due to the extremely high costs of running such experiments (*i.e.,* roughly 86000 GPU hours for the current experiments), in this work, we only experimented with a small set of models, methods, and datasets. While this might limit generalizability, we believe the existence of such analysis is critical to help practitioners save costs. Moreover, since the conclusion of our experiments, new reward models with higher performance have been released (*e.g.,* ArmoRM (Wang et al., 2024)); however, we still rely on older, smaller models to keep the evaluation tractable on such a high number of runs.

Acknowledgements

This work was partially funded by the Defense Advanced Research Projects Agency with the award HR00112220046.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

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

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

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

Shengyi Huang, Michael Noukhovitch, Arian Hosseini, Kashif Rasul, Weixun Wang, and Lewis Tunstall. 2024a. The n+ implementation details of rlhf with ppo: A case study on tl; dr summarization. *arXiv preprint arXiv:2403.17031*.

Shengyi Costa Huang, Tianlin Liu, and Leandro von Werra. 2024b. The n implementation details of rlhf with ppo. In *ICLR Blogposts 2024*. Https://d2jud02ci9yv69.cloudfront.net/2024-05-07-the-n-implementation-details-of-rlhf-with-ppo-130/blog/the-n-implementation-details-of-rlhf-with-ppo/.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *arXiv preprint arXiv:2406.09279*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. 2022. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Preprint*, arXiv:2406.17557.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*.

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

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024b. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Extended Related Work

Online Algorithms. Reinforcement learning from human/AI feedback (RLHF/RLAIF) is among the common approaches for aligning LLMs to human preferences (Christiano et al., 2017; Bai et al., 2022a; Stiennon et al., 2020; Bai et al., 2022b), and has been used to train models such as GPT-4 (Achiam et al., 2023) and Llama-3 (AI@Meta, 2024). In most cases, these approaches are comprised of three stages: 1) supervised fine-tuning (Taori et al., 2023; Zhou et al., 2024; Xia et al., 2024), 2) reward modeling (Gao et al., 2023; Chen et al., 2024; Lightman et al., 2023), and 3) policy optimization (Schulman et al., 2017). The prominent method for policy optimization is Proximal Policy Optimization (PPO), an online on-policy approach (Schulman et al., 2017). While PPO has shown promising performances (Stiennon et al., 2020; Ouyang et al., 2022; Achiam et al., 2023), it suffers from problems such as having too many subtle details for reproducibility (Huang et al., 2024b), 2) taking a long time for training (Huang et al., 2024a), and 3) reward over-optimization (Skalse et al., 2022).

Offline Algorithms. To address the drawbacks of RLHF/RLAIF, recent works have proposed simpler and more efficient offline algorithms, particularly Direct Preference Optimization (DPO) (Rafailov et al., 2024), which is based on the Bradley-Terry model (Bradley and Terry, 1952). These offline algorithms directly optimize an objective on the preference data with an implicit reward model without needing to have separate stages. Some recent works have focused on making a broad comparison between PPO and DPO. Specifically, they showcase the potential for PPO with a gold reward model ($\sim +10\%$) while underlying the similarity to DPO ($\sim +1\%$ averaged across benchmarks) when trained on the same data (Ivison et al., 2024; Xu et al., 2024b).

B Hyperparameter Tuning Considerations

DPO. As presented in Figure 4, lower β leads to higher performances; however, as β decreases, the performance variance increases, which showcases the method's instability. Overall, $\beta = 0.05$ provides the best balance of stability and performance.

LN-DPO. While we initially borrowed β's range from SimPO (Meng et al., 2024), more experiments showed benefits in further decreasing its value. Figure 5 presents the performance spread across different runs. From these experiments, $\beta \in [1.0, 2.0]$ contains most of the best-performing models. Moreover, we observe the relatively low (compared to DPO) variance across the performances, showcasing another benefit of LN-DPO.

SimPO. In contrast to the other two methods, SimPO has two method-specific hyperparameters: β and γ. As illustrated in Figure 6, on average, lower β values lead to better performance. We believe the performance uptick in the lower range is due to a difference in the average length of this work's and the original work's training sets. Moreover, as showcased in Figure 7, the best performing models have a $\gamma \in [1.0, 1.4]$, in line with the suggestion by Meng et al. (2024). Notably, β and γ have a relatively low variance across experiments, another upside of SimPO.

C The Answer to the Ultimate Question

Based on our collective empirical results, we believe SimPO to be the best starting point among the three methods, mainly due to its robustness toward hyperparameter variations and effective length reduction. As for SimPO's hyperparameters, we recommend $\beta \in \{1.0, 1.5\}$ and $\gamma \approx 1.2$. Moreover, while LN-DPO is consistently second-best in most of our experiments, we discuss scenarios for choosing it over SimPO in Section 8.

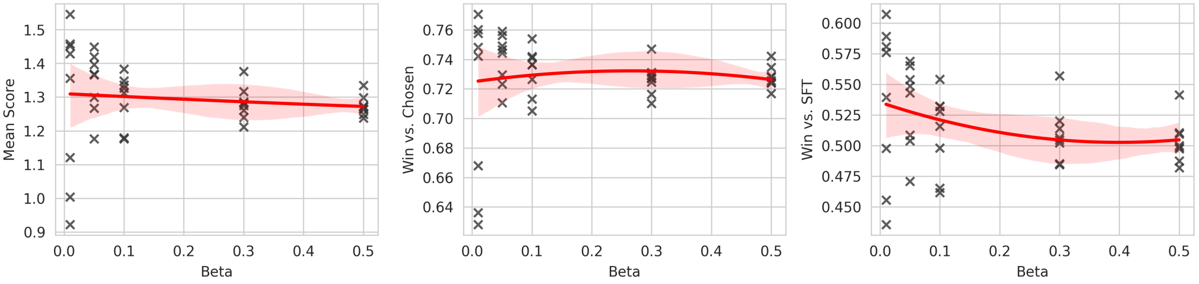

Figure 4: **DPO** β. Each point indicates a run with the corresponding β value.

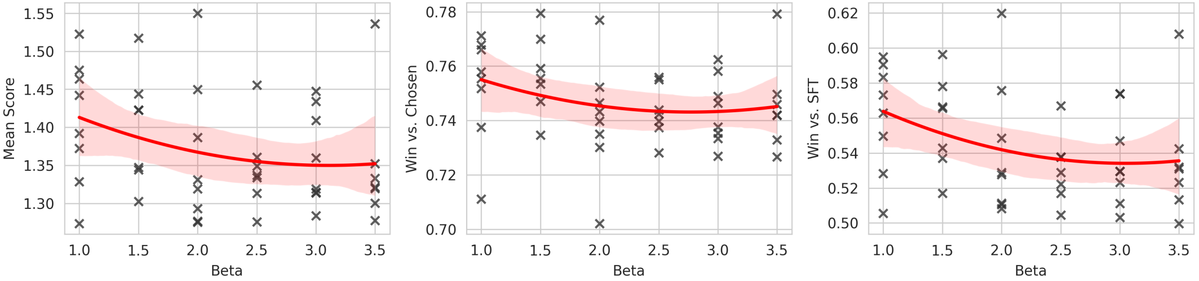

Figure 5: **LN-DPO** β. Each point indicates a run with the corresponding β value.

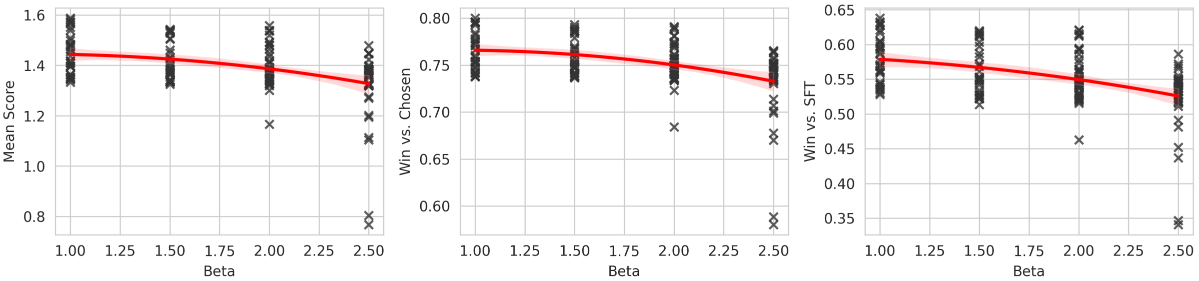

Figure 6: **SimPO** β. Each point indicates a run with the corresponding β value.

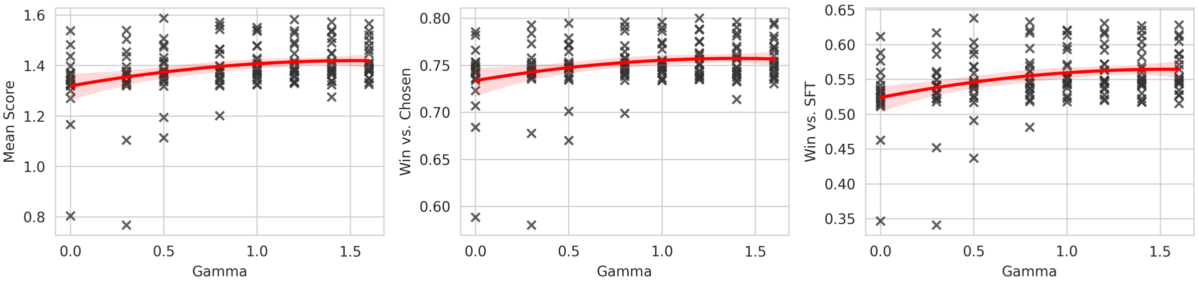

Figure 7: **SimPO** γ. Each point indicates a run with the corresponding γ value.