P-Aligner: Appendix

Feifan Song
Peking University
Beijing, China

Bofei Gao
Peking University
Beijing, China

Yifan Song
Peking University
Beijing, China

Yi Liu
Peking University
Beijing, China

Weimin Xiong
Peking University
Beijing, China

Yuyang Song
University of Southampton
Southampton, Hampshire, UK

Tianyu Liu
Peking University
Beijing, China

Guoyin Wang
Alibaba Group
Seattle, WA, USA

Houfeng Wang*
Peking University
Beijing, China

ACM Reference Format:
Feifan Song, Bofei Gao, Yifan Song, Yi Liu, Weimin Xiong, Yuyang Song, Tianyu Liu, Guoyin Wang, and Houfeng Wang. 2018. P-Aligner: Appendix. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

A Effect of Iterative Optimization

An interesting observation in Cheng et al. [2] is that repeatedly applying BPO can yield further gains. While empirically useful, this practice is economically unattractive: each additional round incurs additional latency of the rewriter.

We hypothesize that the diminishing-returns improvement of iteratively running BPO stems from the corpus in its training. To be specific, each refined instruction is produced by a single heuristic rewrite step, whose direction is therefore implicit and slight. Consequently, multi-time BPO also functions like a low-resolution search process. In contrast, resources in UltraPrompt, which support P-Aligner, are already near-optimal with iteratively search in data synthesis while requiring no human annotation. We check this point by replicate the iterative experiment with P-Aligner, using Gemma-2-SimPO and measuring performance via win-rate on four benchmarks involved in Cheng et al. [2].

Figure 1 illustrates the results. Unlike BPO which shows gradual improvement with additional passes, P-Aligner exhibits no consistent trend across iterations. For example, performance on BPO Test (BT) and Dolly Eval (DE) remains stable, while scores on Self-Instruct Eval (SE) and Vicuna Eval (VE) show clear fluctuation, or even decrese (in VE). These indicate that P-Aligner delivers near-optimal instructions in a single step, eliminating the need for and consumption of iterative refinement.

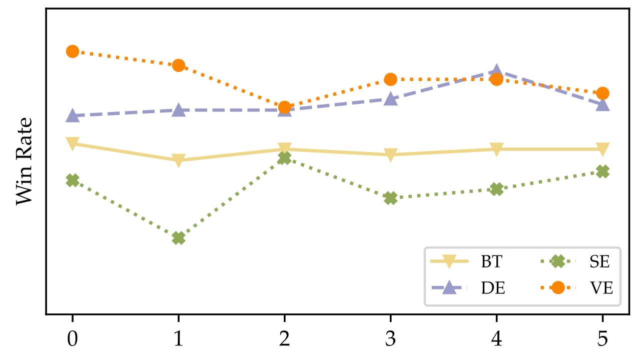

Figure 1: Effect of iterative instruction optimization with P-Aligner, where P-Aligner can almost achieve the highest benefit without multiple runs. The X-axis represents the index of iteration.

B Dataset Construction

In this work, we propose **UltraPrompt**, a preference dataset synthesized through our synthesis pipeline. It contains 10000 seed instructions selected from various sources: UltraFeedback [5], HH-RLHF [1], Glaive-code-assistant[1], and MathInstruct [15], which is completed according to N-grams diversity in Song et al. [13] to cover various domains, as shown in Table 1.

UltraPrompt has an additionaly split that is used to train **SinglePO**. In detail, we reuse the 10000 search trees and collect all 104602 positive transitions, i.e., where the next instruction has a higher reward than the current instruction. The distribution of principles in the *single-step* split is shown in Figure 2.

C Principles

The principles are pre-defined to cover commonly recognized positive aspects for achieving human preference. It involves multiple domains: Harmlessness, Helpfulness, Honesty, Coding & Debugging, and Math. Each domain contains several priciples. Importantly, the principles of Helpfulness are intended to be universally applicable across all defined domains. A detailed categorization is presented below:

*Correspondence author.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

[1] https://huggingface.co/datasets/glaiveai/glaive-code-assistant

Sources	# Instructions	Category
FalseQA [8]	250	Honesty
TruthfulQA [11]	250	Honesty
FLAN [12]	300	Helpfulness
HH-RLHF [1]	1500	Harmlessness
UltraChat [6]	1800	Helpfulness
ShareGPT [3]	3000	Helpfulness
GSM-RFT	300	Math
Math50k-camel	300	Math
MATH [7]	300	Math
Glaive-code-assistant	2000	Coding & Debugging

Table 1: Statistics of the seed instruction sources in UltraPrompt.

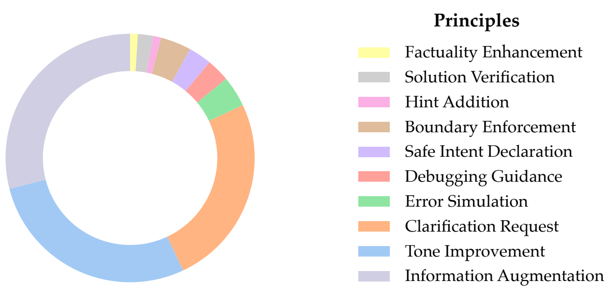

Figure 2: Distribution of principles in the *single-step* split of UltraPrompt.

Harmlessness:

- **Safe Intent Declaration** - adding a safety-oriented preface (e.g., "Please respond respectfully and avoid harmful or unethical content...") requiring the response to meet ethical guidelines.
- **Boundary Enforcement** - appending explicit refusal instructions for any content violating privacy/ethics (e.g., "If this request involves offensive content, decline politely").

Helpfulness:

- **Clarification Request** - making it more clear and instructive, leading the agent to answer it in detail.
- **Information Augmentation** - making it more detailed and informative, such as adding more background information and so on, which may help the agent better understand the content.
- **Tone Improvement** - improving its tone to be more polite, helpful, honest, and friendly.

Honesty:

- **Factuality Enhancement** - encouraging objective facts instead of fake material and subjective interpretations.

Coding & Debugging:

- **Error Simulation** - adding requests about considering common bugs or edge cases related to the prompt.
- **Debugging Guidance** - offering step-by-step debugging instructions, error analysis, and troubleshooting tips to resolve coding issues.

Math:

- **Hint Addition** - introducing hints, examples or multi-step pre-thinking to reduce the difficulty of problems.
- **Solution Verification** - asking the agent to verify the correctness of their response or provide a detailed explanation of their reasoning.

D Further Explanation of Evaluation Settings

In this work, we evaluate our methods on five instruction-following benchmarks. Four of them are also adopted by Cheng et al. [2], which we list as follows:

- **Dolly Eval** contains 200 samples drawn from the Dolly dataset [4] to test general instruction following performance.
- **Vicuna Eval** [3] contains 80 samples spanning role-play, common-sense, creative writing, coding, mathematics, and other categories.
- **Self-Instruct Eval** [14] contains 252 carefully crafted samples, each pairing a context with an instruction.
- **BPO Test** contains 200 samples introduced by Cheng et al. [2].

We further include **ArenaHard** [9], a popular benchmark containing 500 challenging prompts and an automated evaluation framework that yields a scalar score of instruction-following performance.

Beyond vanilla inference, we further equip Llama-3.1-8B with two decoding-time alignment strategies. **Best-of-N** samples N candidate responses and returns the highest-scoring one as judged by ArmoRM-Llama3-8B-v0.1. **URIAL** [10] steers the decoding distribution via in-context learning on human-preferred demonstrations. Both methods function at generation time and are orthogonal to BPO and P-Aligner, which work by refining the prompts, so they can be well aggregated for a more comprehensive evaluation and robust validation of our findings.

E Prompt Templates

Please see Figure 3, 4 and 5 for reference.

F Cases

Please see Table 2 and 3 for reference.

The user query to be paraphrased is {The given query}.
You should optimize this query by {Content of the selected rule}.
You should also return the optimized version directly, without any prefix.

Figure 3: The prompt template used for GPT-4 to improve the given query.

You are an expert prompt engineer. Please help me optimize this prompt to get better response:

[The Start of Raw Prompt]
{The given query}
[The End of Raw Prompt]

You should optimize this prompt by {Content of the selected rule}.

Figure 4: The prompt template used for SinglePO to improve the given query.

You are an expert prompt engineer. Please help me optimize this prompt to get better response:

[The Start of Raw Prompt]
{The given query}
[The End of Raw Prompt]

Figure 5: The prompt template used for MCTS-p to improve the given query.

Raw Prompt	**User:** How can governments utilize fiscal and monetary policies to combat economic recessions? **Llama-3.1-8B:** Governments can utilize fiscal and monetary policies to combat economic recessions by increasing government spending, reducing taxes, and increasing the money supply. These policies can help stimulate the economy and create jobs, which can help to reduce the severity of a recession.
Prompt by BPO	**User:** How can governments utilize fiscal and monetary policies to combat economic recessions? Provide a detailed explanation of the specific measures governments can implement through fiscal and monetary policies to address economic downturns. Include examples of how these policies can impact the economy, such as increasing demand and reducing costs. **Llama-3.1-8B:** Governments can utilize fiscal and monetary policies to combat economic recessions by implementing measures such as increasing government spending, reducing taxes, and providing stimulus packages. These policies can help increase demand in the economy and reduce costs, which can help stimulate economic growth and reduce unemployment. For example, increasing government spending on infrastructure projects can create jobs and boost economic activity, while reducing taxes can put more money in the hands of consumers, who can then spend it on goods and services, further stimulating the economy. Additionally, providing stimulus packages can provide financial support to businesses and individuals, helping to keep them afloat during difficult times. Overall, these policies can help governments combat economic recessions and promote economic stability and growth.

Prompt by P-Aligner	**User:** Could you kindly provide a detailed explanation on the various fiscal and monetary policies that governments can implement to mitigate the effects of economic downturns or recessions? I'm particularly interested in understanding the specific actions they can take to stimulate economic growth, manage inflation, and maintain financial stability. Your expertise on this topic is greatly appreciated. **Llama-3.1-8B:** Certainly! Here are some of the key fiscal and monetary policies that governments can use to address economic downturns and recessions: 1. **Fiscal Stimulus:** Governments can use fiscal stimulus measures such as tax cuts, increased government spending, or direct cash transfers to individuals or businesses to boost consumer spending and investment. This can help stimulate economic growth and create jobs. 2. **Monetary Policy:** Central banks can use monetary policy tools such as interest rate adjustments, quantitative easing, or open market operations to influence the supply of money and credit in the economy. Lowering interest rates can encourage borrowing and spending, while quantitative easing can increase the money supply and stimulate economic activity. 3. **Exchange Rate Policy:** Governments can adjust their exchange rates to make their exports more competitive or to attract foreign investment. This can help boost exports and stimulate economic growth. 4. **Regulatory Policy:** Governments can use regulatory policies to encourage investment and innovation, such as tax incentives for research and development or streamlined permitting processes for new businesses. 5. **Social Safety Net:** Governments can strengthen their social safety net programs, such as unemployment insurance, food stamps, or housing assistance, to support individuals and families during economic downturns. 6. **Infrastructure Investment:** Governments can invest in infrastructure projects such as roads, bridges, and public transportation to create jobs and stimulate economic growth. 7. **Trade Policy:** Governments can use trade policies such as tariffs or import quotas to protect domestic industries or to encourage exports. 8. **Financial Sector Regulation:** Governments can regulate the financial sector to prevent excessive risk-taking and to ensure that financial institutions are well-capitalized and able to withstand economic shocks. 9. **Public Debt Management:** Governments can manage their public debt levels to ensure that they are sustainable and do not become a burden on future generations. 10. **Economic Forecasting:** Governments can use economic forecasting models to anticipate economic downturns and recessions and to develop appropriate policy responses. Overall, governments have a range of fiscal and monetary policy tools at their disposal to address economic downturns and recessions. The specific policies that are most effective will depend on the nature and severity of the economic downturn, as well as the specific circumstances of each country or region.

Table 2: Case demonstration from Vicuna Eval.

Raw Prompt	**User:** Who was the lead singer for the band Aerosmith? **Gemma-2-SimPO:** The lead singer for the band Aerosmith is **Steven Tyler**.
Prompt by BPO (identical to the Raw Prompt)	**User:** Who was the lead singer for the band Aerosmith? **Gemma-2-SimPO:** The lead singer for the band Aerosmith is **Steven Tyler**.
Prompt by P-Aligner	**User:** Could you please provide information on the lead vocalist of the renowned American rock band Aerosmith? **Gemma-2-SimPO:** The lead vocalist of the renowned American rock band Aerosmith is **Steven Tyler**. Known for his powerful, raspy vocals, flamboyant stage presence, and signature wide vocal range, Tyler is a defining element of Aerosmith's iconic sound and image. He's not just a vocalist; he's a charismatic frontman who heavily contributes to the band's energetic performances and enduring appeal.

Table 3: Case demonstration from Dolly Eval.

References

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

[2] Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-Box Prompt Optimization: Aligning Large Language Models without Model Training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3201–3219. doi:10.18653/v1/2024.acl-long.176

[3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

[4] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks* (2023).

[5] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=BOorDpKHiJ

[6] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3029–3051. doi:10.18653/v1/2023.emnlp-main.183

[7] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=7Bywt2mQsCe

[8] Shengding Hu, Yifan Luo, Huadong Wang, Xingyi Cheng, Zhiyuan Liu, and Maosong Sun. 2023. Won't Get Fooled Again: Answering Questions with False Premises. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5626–5643. doi:10.18653/v1/2023.acl-long.309

[9] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2025. From Crowdsourced Data to High-quality Benchmarks: Arena-Hard and Benchbuilder Pipeline. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=KfTf9vFvSn

[10] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The Unlocking Spell on Base LLMs: Rethinking Alignment via In-Context Learning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=wxJ0eXwwda

[11] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3214–3252. doi:10.18653/v1/2022.acl-long.229

[12] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 22631–22648. https://proceedings.mlr.press/v202/longpre23a.html

[13] Feifan Song, Bowen Yu, Hao Lang, Haiyang Yu, Fei Huang, Houfeng Wang, and Yongbin Li. 2024. Scaling Data Diversity for Fine-Tuning Language Models in Human Alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 14358–14369. https://aclanthology.org/2024.lrec-main.1251/

[14] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13484–13508. doi:10.18653/v1/2023.acl-long.754

[15] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=yLClGs770I