Extended Abstract

Motivation Preference-based learning has emerged as a powerful alternative to supervised fine-tuning for aligning language models with human intent, particularly for open-ended and subjective tasks. However, models trained with preference data often produce outputs of inconsistent quality, especially when evaluated on challenging benchmarks such as UltraFeedback, which rely on relative preferences without access to a ground truth reward signal. These challenges are amplified in smaller-scale models like Qwen2.5-0.5B, which lack the capacity of larger LLMs and may struggle to internalize alignment objectives. This project investigates whether a hybrid approach, in combining preference optimization at training time with sampling and reranking at inference, can lead to more robust and consistent instruction-following behavior, even in resource-constrained settings.

Method We fine-tune the Qwen2.5-0.5B base model using two distinct alignment strategies. First, we apply supervised fine-tuning (SFT) on the SmolTalk dataset to establish a baseline instruction-following model. To further align the model with human preferences, we apply Direct Preference Optimization (DPO) on top of the SFT-tuned model. DPO, which contains pairwise human preferences across diverse prompts and completions, is initialized from the SFT checkpoint, allowing it to leverage the strong instruction-following foundation established through supervised fine-tuning. To further boost performance at inference, we introduce a Best-of-N reranking strategy that samples multiple candidate responses per prompt and selects the highest-scoring one using the Nemotron-70B reward model. We vary the sampling temperature to study the trade-off between output diversity and alignment fidelity, analyzing how test-time decoding interacts with preference-optimized training.

Implementation The SFT model was trained using a batch size of 2, gradient accumulation of 8, and a learning rate of 5×10^{-5} on 8000 instruction-response pairs from the SmolTalk dataset. For DPO, we used the full UltraFeedback dataset, training with a batch size of 4, gradient accumulation of 8, and a learning rate of 1×10^{-6}. The DPO model used the SFT model as its reference policy. For the Best-of-N extension, we generated $N = 4$ completions per prompt using top-k sampling, then reranked them with the Nemotron-70B reward model. We experimented with temperatures of 0.25, 0.6, and 1.0 to analyze the effect of sampling entropy on final quality. Evaluation included reward accuracy, reward margins, preference-based loss, and win rates from reward model comparisons.

Results Quantitative results demonstrate that DPO significantly outperforms SFT, achieving a win rate of 0.6200 over the SFT model. When combined with Best-of-N reranking at temperature 0.6, the win rate improved to 0.9350, showcasing the complementary value of decoding-time selection. Additional experiments with alternative temperatures (0.25 and 1.0) revealed robustness across sampling configurations. Submissions to the UltraFeedback class leaderboard corroborated these gains: the DPO model achieved a score of 0.1150, while the reranked variant reached 0.2000, both exceeding the official 0.1 benchmark threshold. Qualitative analysis further confirmed that DPO reduced repetition and hallucination, and reranking improved coherence, instruction adherence, and factual reliability across tasks.

Discussion Our findings show that preference-based optimization and test-time reranking are highly complementary. DPO effectively aligns model behavior with human preferences during training, while Best-of-N sampling enables dynamic selection of higher-quality outputs at inference. Together, they help mitigate typical small model limitations such as verbosity, hallucination, and misalignment. However, limitations remain. The need to generate and evaluate multiple completions increases inference cost, which may be a concern in latency-sensitive applications. In addition, reliance on reward models introduces concerns around their accuracy, calibration, and domain generalization. The binary framing of preferences in DPO also oversimplifies complex quality judgments, motivating future work in richer feedback modeling.

Conclusion This project demonstrates that combining Direct Preference Optimization with reward-based reranking at inference significantly improves instruction-following performance in smaller language models. Our results highlight the importance of integrating both training-time preference alignment and decoding-time generation control. These complementary strategies offer a scalable pathway toward more helpful, coherent, and reliable language model outputs, particularly in low-resource or compact-model scenarios.

CS224R Final Report

Renee Qin
Department of Computer Science
Stanford University
reneeqin@stanford.edu

Nicole Garcia
Department of Computer Science
Stanford University
nicolejg@stanford.edu

Abstract

We investigate the effectiveness of combining preference-based training and test-time inference/reranking as an extension to improve instruction-following in compact language models. Using the Qwen2.5-0.5B as a base model, we compare supervised fine-tuning (SFT) on the SmolTalk dataset with Direct Preference Optimization (DPO) built on top of the SFT model using the UltraFeedback dataset. As an extension, we introduce a test-time reranking strategy that samples multiple completions per prompt and selects the best one using the Nemotron-70B reward model. Our results show that DPO largely outperforms SFT (win rate 0.6200), and that reranking further boosts performance (up to 0.9350 at temperature 0.6). This work highlights the complementary strengths of preference alignment during training and reward-guided selection at inference, offering a practical pathway to improving output quality in low-resource LLMs.

1 Introduction

Training reward-aligned language models remains a central challenge in the development of reliable AI systems, particularly in settings where reward signals are noisy, sparse, or unavailable at inference time. This difficulty is especially pronounced in preference-based learning methods such as Direct Preference Optimization (DPO), where models are trained to prefer responses ranked higher in pairwise human comparisons rather than optimizing for a known reward function. While such approaches have shown promise in improving alignment and response quality, output variability and unstable instruction-following behavior remain persistent issues, especially for smaller models and open-ended prompts.

These challenges are exacerbated when using benchmarks like UltraFeedback, which provide only relative preference labels without a ground-truth reward function. This makes it difficult to both train and evaluate models reliably, particularly in low-resource situations. Even with fine-tuning, the quality of generated responses may vary widely. This shows the need for strategies that enhance robustness beyond training alone.

In this project, we focus on the task of Instruction Following and explore multiple approaches to improve performance: supervised fine-tuning (SFT), preference-based learning via DPO, and a test-time reranking strategy. The central hypothesis we examine is whether generation quality can be improved by deferring part of the decision-making process to test time through Best-of-N reranking, wherein multiple completions are sampled and the highest-scoring one is selected according to a strong reward model. Specifically, we use the Nemotron-70B reward model to evaluate and rerank outputs from the Qwen2.5-0.5B base model and its SFT and DPO variants. This setup allows us to assess both the training-time benefits of DPO and the inference-time gains of sampling-based reranking, especially under resource-constrained conditions.

2 Related Work

Instruction-following and alignment in language models have received increasing attention in recent years as researchers seek to make large language models more helpful, truthful, and safe. A foundational approach in this area is Supervised Fine-Tuning (SFT) on instruction-tuning datasets, as demonstrated in models like InstructGPT (Ouyang et al., 2022) and FLAN (Wei et al., 2022). While effective, SFT often falls short of capturing nuanced human preferences, particularly for subjective or open-ended tasks.

To address this, preference-based methods have emerged as a more fine-grained alternative. Reinforcement Learning from Human Feedback (RLHF), most notably implemented via Proximal Policy Optimization (PPO) in InstructGPT, enables models to learn from pairwise comparisons instead of fixed targets. However, RLHF is complex and computationally expensive, often requiring careful tuning of reward models and training stability mechanisms (Bai et al., 2022; Christiano et al., 2023).

Direct Preference Optimization (DPO) (Rafailov et al., 2024) offers a simpler and more stable alternative by directly optimizing for preference logits using a contrastive loss between preferred and dispreferred responses. This has been shown to yield comparable or better alignment performance than PPO-based methods, with much lower computational overhead. DPO has been particularly effective in instruction-following and dialog tasks, making it a compelling choice for low-resource or smaller model scenarios.

Recent work has further extended DPO with methods such as fDPO for filtering noisy preferences (Morimura et al., 2024) and ODPO for preference weighting (Amini et al., 2024). These efforts aim to better handle real-world preference distributions and training noise. In addition, reward model calibration has emerged as an important area of concern, with techniques such as Cal-DPO (Xiao et al., 2024) and DICE (Chen et al., 2025) improving reward quality and stability, especially for downstream tasks.

Our work extends DPO by incorporating a test-time reranking strategy inspired by approaches such as Best-of-N sampling (Jinnai et al., 2025; Ichihara et al., 2025; Gui et al., 2024; Sessa et al., 2024), which generate multiple completions and select the most aligned response using an external reward model. This idea builds on the insight that sampling-based decoding can significantly affect model outputs, even after alignment training.

Recent works such as UltraFeedback (Cui et al., 2024) have proposed evaluating language models using large reward models instead of human annotators, enabling more scalable preference benchmarking. We adopt this evaluation setup using Nemotron-70B as a reward model to assess the alignment and fluency of our DPO and DPO in combination with reranking models. This method aligns with the growing shift toward automatic preference evaluation at scale, which leverages alignment-focused reward models.

Finally, while alignment research often centers on large models, recent works highlight the unique challenges and opportunities in applying alignment techniques to smaller models. Our study contributes to this direction by demonstrating how decoding control and reward reranking can compensate for the limited capacity of compact architectures. Together, our method draws from this evolving ecosystem of SFT, DPO, reward modeling, and decoding strategies to explore whether smaller LLMs can still achieve strong alignment through a combination of efficient training and inference-time control.

3 Method

Our approach to improving instruction-following performance involves three components: Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and test-time reranking using Best-of-N sampling. We fine-tune the Qwen2.5-0.5B base model using two distinct alignment strategies. First, we apply supervised fine-tuning on the SmolTalk dataset to establish a baseline instruction-following model. Next, we build on this foundation by applying Direct Preference Optimization using the UltraFeedback dataset, initializing DPO from the pretrained SFT model. This hierarchical setup allows DPO to refine instruction-following behavior by explicitly modeling human preferences over the outputs of an already instruction-tuned model. Below, we describe each component in detail, including model architectures, loss functions, and decoding strategies.

3.1 Supervised Fine-Tuning (SFT)

We begin with a pretrained base model (Qwen2.5-0.5B) and fine-tune it using supervised learning on instruction-response pairs. The SFT objective is to maximize the likelihood of human-written responses given a prompt, defined by the loss below:

$$\underset{\theta}{max} x\mathbb{E}_{x,y\in D} \sum_{t=1}^{|y|} \log \pi_\theta(y_t|x, y_{<t}) \tag{1}$$

where the supervised learning objective is optimized over queries x and completions y drawn from an expert distribution.

3.2 Direct Preference Optimization (DPO)

To align the model more closely with human preferences, we apply Direct Preference Optimization (DPO) using a dataset of preference pairs (x, y_w, y_l), where y_w is the preferred response over y_l for prompt x. DPO eliminates the need for reinforcement learning by directly optimizing a contrastive loss between the logits of the chosen and rejected responses. The objective is:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)})] \tag{2}$$

and here, π_θ is the policy that is being optimized and π_{ref} is the reference policy. Implicitly, the reward margin for a preference pair (x, y_w, y_l) is defined as $r(x, y_w, y_l) = \beta \log \left(\frac{\pi_\theta(y_w|x)/\pi_{\text{ref}}(y_w|x)}{\pi_\theta(y_l|x)/\pi_{\text{ref}}(y_l|x)} \right)$. And the reward accuracy for DPO is the proportion of preference pairs in the dataset for which $r(x, y_w, y_l) > 0$. These two values can be used to measure the extent of DPO's learning of human preferences.

3.3 Test-Time Reranking with Best-of-N Sampling

To further improve generation quality, we implement a decoding-time strategy where the model generates N candidate responses per prompt using top-k sampling with temperature. These candidates are then scored using an external reward model (Nemotron-70B) and a response among the highest-scoring ones is selected. This approach decouples generation and selection, leveraging diversity in outputs while enforcing alignment through reward-based selection.

Formally:

$$\hat{y} = \arg \max_{y_i \in \{y_1,\ldots,y_N\}} R(y_i, x) \tag{3}$$

where $R(y_i, x)$ is the reward model score for a candidate response y_i given prompt x.

By combining preference-based learning with decoding-time optimization, our method seeks to produce outputs that are not only instruction-following but also aligned with downstream preferences.

4 Experimental Setup

4.1 Datasets

Our experiments use two datasets: SmolTalk and UltraFeedback. The SmolTalk dataset is used for supervised fine-tuning (SFT). For preference training with Direct Preference Optimization (DPO), we use the full UltraFeedback dataset, which contains pairwise human preference comparisons over a diverse range of instructions and completions.

4.2 Training Configurations

We initialize our models with the Qwen2.5-0.5B architecture and fine-tune using the SmolTalk dataset. SFT is performed with a batch size of 2 and a gradient accumulation step of 8 to simulate an effective batch size of 16. We use a learning rate of 5×10^{-5} and a max token length of 512. Due to computational constraints, we only trained SFT on a random subset with 8000 pairs from smolTalk for 1 epoch. This model is also used as the reference model for subsequent DPO training.

For DPO training, we use the full UltraFeedback preference dataset. The training is conducted with a batch size of 4 and gradient accumulation of 8, giving an effective batch size of 32. We use a smaller learning rate of 1×10^{-6} to ensure stable optimization. The DPO temperature hyperparameter β is set at 0.1 and the max token length is set to 1024. And we trained DPO for 1 epoch. Reward margin and reward accuracy metrics are logged during training to monitor model progression and preference alignment.

To enhance model performance at inference time, we adopt a Best-of-N sampling strategy as a test-time extension. Specifically, for each input prompt, we generate $N = 4$ completions using top-k sampling and rerank them using the Nemotron-70B reward model. A response among the highest-scoring ones is selected as the final output. Our primary configuration uses a temperature of 0.6, but we also experimented with lower temperatures of 0.25 and 0.1 to examine the trade-offs between diversity and precision.

4.3 Evaluation

For evaluation, we compute win rates using the LLaMA-3.1-Nemotron-70B reward model to score pairwise outputs. We report both reward margins and reward accuracy metrics for DPO. In addition to quantitative analysis, we perform qualitative inspection on a held-out set of UltraFeedback prompts to assess fluency, instruction-following, and factual consistency across model variants.

5 Results

We present both quantitative and qualitative evaluations to assess the effectiveness of Direct Preference Optimization (DPO) and its enhancement via top-N decoding. The quantitative results leverage an automated reward model to benchmark comparative response quality, while the qualitative analysis provides deeper insight into the kinds of improvements and failure modes exhibited by each variant. Together, these evaluations offer a comprehensive view of model behavior across instructional, factual, and generative tasks.

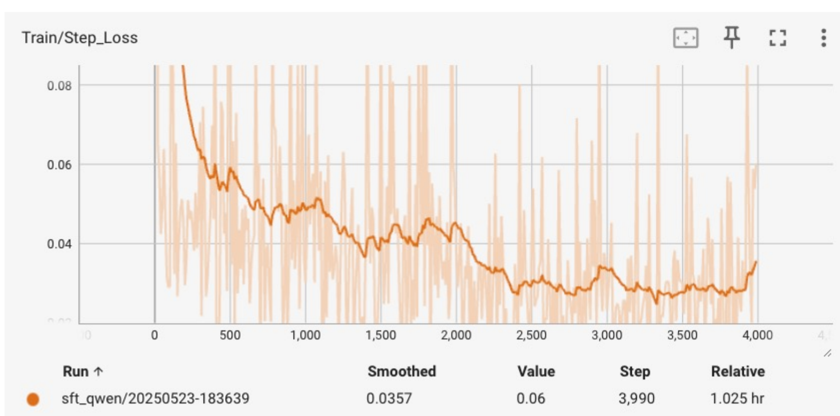

Figure 1: SFT training loss curve on the SmolTalk dataset.

Figure 1 and Figure 2 show the training loss curves for the SFT and DPO stages, respectively. The SFT model exhibits smooth convergence on the SmolTalk dataset, confirming stable learning under the supervised objective.

Figure 2: DPO training step loss with exponential smoothing (factor 0.95).

The DPO model, trained on the UltraFeedback dataset, demonstrates effective optimization with a smoothed training loss curve (smoothing factor of 0.95). Evaluation metrics show promising alignment performance: the chosen response reward reaches 0.1878 while the rejected response receives a much lower reward of -0.3919 (more negative is better), yielding a reward margin of 0.5798. The corresponding evaluation loss is 0.563, with a reward accuracy of 0.6923. These results suggest successful preference alignment during training and support the efficacy of DPO even under resource-constrained conditions.

5.1 Quantitative Evaluation

To evaluate overall model quality, we performed quantitative comparisons using the LLaMA-3.1-Nemotron-70B reward model, which scores pairwise generations by estimating human preference alignment. This automated evaluation helps benchmark improvements across various training and decoding strategies. Results are reported as win-rates, such that it measures the proportion of times the first model's response is preferred over the second.

Model A	**Model B**	**Win Rate (A over B)**
SFT	Qwen2.5 Baseline	0.9450
SFT + Top-N	Qwen2.5 Baseline	0.9700
DPO	SFT	0.6200
DPO + Top-N ($T = 0.25$)	SFT	0.9300
DPO + Top-N ($T = 0.6$)	SFT	0.9350
DPO + Top-N ($T = 1.0$)	SFT	0.8750

Table 1: Pairwise Win Rates from Nemotron-70B Reward Model Evaluation

First, comparing supervised fine-tuning (SFT) outputs to a Qwen2.5 baseline, we find that both the original and reranked (top-N) versions of SFT responses perform strongly. The base SFT model achieved a score of 0.9450, while SFT with top-N reranking further improved to 0.9700, indicating that top-N sampling contributes to a measurable boost in quality even without any preference optimization.

Direct Preference Optimization (DPO) showed promising results as well. When comparing DPO outputs to SFT outputs directly, the DPO model achieved a win rate of 0.6200, meaning it was preferred 62 percent of the time over SFT, which is a decent improvement. This suggests that DPO training alone meaningfully improves response quality over standard supervised fine-tuning.

Moreover, combining DPO with top-N sampling and temperature tuning yielded further gains. DPO with top-N decoding at temperature = 0.6 scored 0.9350 against SFT, while variations at temperature = 0.25 and temperature = 1.0 achieved 0.9300 and 0.8750, respectively. These results show that top-N decoding consistently enhances DPO outputs, and moderate temperature values such as 0.6

offer a strong trade-off between diversity and reliability. This finding is consistent with the results in Rafailov et al. (2024). Due to computational constraints, we did not ablate for higher values of N.

In addition to our offline evaluations, we submitted both the DPO and DPO combined with top-N models to the class leaderboard for the UltraFeedback benchmark. The base DPO model achieved a score of 0.1150, while the test-time reranking extension with Best-of-N sampling reached a score of 0.2000. Both models exceed the minimum required threshold of 0.1, demonstrating the practical effectiveness of preference optimization and reranking techniques under the official evaluation protocol. These results further validate the alignment between our controlled experimental findings and real-world benchmark performance.

Overall, the quantitative evaluation confirms that DPO with top-N decoding surpasses SFT, both in head-to-head comparisons and in alignment with reward model preferences. While DPO on its own already offers a notable improvement over SFT, the combination with top-N sampling and careful temperature tuning diminishes the remaining quality gap. These findings underscore the importance of not only preference-aligned training but also well-calibrated decoding strategies in maximizing model quality. The alignment between these quantitative scores and the qualitative improvements observed in structure, fluency, and factual correctness further validates top-N DPO as a robust and scalable approach to fine-tuning high-quality language models.

5.2 Qualitative Analysis

To complement quantitative evaluations of model performance, we conducted a qualitative analysis of two output files, one using the DPO approach and the other using an extension with a test-time reranking approach on top of DPO. These outputs contain model-generated responses to a shared set of held-out prompts. This comparison focuses on aspects such as fluency, factual accuracy, instruction-following behavior, and coherence. While both files reflect models trained with direct preference optimization, the Top-N version incorporates reranking and sampling improvements. Our goal is to assess how these differences manifest in real-world tasks, from creative writing and multilingual reasoning to coding and factual question-answering. Below, we present representative examples and observations that highlight key distinctions in output quality.

The generated responses with DPO contain lengthy, occasionally repetitive texts, and tend to over-explain and hallucinate technical details. For instance, in the response to the prompt “How is augmented reality being used to enhance museum experiences and engage visitors with interactive exhibits?”, the model repeats the idea of “AR being used to create 360-degree views” and “AR helping visitors explore 3D models” in multiple paragraphs, using phrasing such as: “...AR can be used to create immersive, 360-degree views of artifacts...” and “...explore a piece of art in a 3D model, manipulate the digital version, and see how it would look in real life...” This repetition dilutes the clarity and makes the response feel padded rather than informative. Furthermore, hallucinations are present in technical answers. This issue is present in questions such as those related to using Python scripts to complete tasks in Blender. Similarly, the chocolate cake enhancement prompt results in a bizarre output where “pineapple” is repeated dozens of times. As such, some of the responses it generates are based on inaccurate claims. Overall, while generated responses based only on DPO contain some completed texts with decent content, they tend to involve excessive repetition, inaccurate claims, and some mismatches between instructions and responses.

On the other hand, the top-N model follows a tighter narrative arc and avoids unnecessary repetition that appears in multiple paragraphs of the DPO only output. In particular, instruction-following behavior is more reliable in the top-N method, such as in a cause-effect task in Spanish, the model correctly answers the prompt. On factual precision, although the top-N outputs give more coherent and logical responses, it is still not perfect. For instance, it repeats the same hallucinated explanation. But such errors are rarer compared to the base DPO responses, where hallucinated claims, repeated text, and logic gaps are common. There is still some occasional overgeneralization and some redundancy in its generations. However, the top-N approach does offer more structured, concise, and coherent answers with fewer hallucinations and better task completion. While both methods share similar base content, the top-N variant clearly reflects more robust decoding or ranking logic, likely reflecting better use of sampling and reranking techniques.

Overall, the top-N DPO variant consistently demonstrates improved alignment with user intent, stronger factual grounding, and cleaner execution across a range of tasks. It generates responses that are not only more concise but also more contextually appropriate and logically coherent, suggesting a better calibration of language generation under task constraints. While it does not eliminate all errors, particularly with respect to subtle factual inaccuracies or occasional verbosity, it represents a clear qualitative step forward compared to the base DPO outputs. This complements our quantitative findings and reinforces the value of combining preference optimization with decoding strategies for high-quality generation.

6 Discussion

Our findings demonstrate that Direct Preference Optimization consistently improves the quality and alignment of language model outputs over standard supervised fine-tuning, even when used in isolation. Across all configurations, DPO outperformed SFT in head-to-head comparisons using a reward model proxy, highlighting its effectiveness at aligning model behavior with human preferences. Furthermore, combining DPO with top-N decoding strategies yielded additional gains. In particular, DPO with top-N decoding at moderate temperatures achieved win rates exceeding SFT baselines. These improvements are also supported by qualitative analysis, which shows that top-N DPO responses are more concise, structured, and instruction-following, with noticeably fewer hallucinations and logical inconsistencies compared to base DPO generations. Samples of our generated responses can be found in the Implementation Details section of our appendix.

Another key insight from this work is that training-time preference alignment and inference-time decoding strategies act synergistically. While DPO steers the model toward more aligned outputs during training, top-N decoding helps better realize these improvements by encouraging diverse, high-quality completions at inference. This highlights the importance of not only optimizing learning objectives but also carefully tuning generation strategies to fully leverage the benefits of preference-based training.

Despite these promising results, several limitations remain. First, while the reward model used for evaluation (LLaMA-3.1-Nemotron-70B) serves as a scalable and consistent proxy for human preferences, it may not capture the full nuance of subjective or creative tasks. Human evaluation remains essential for validating alignment in more open-ended settings. Second, our qualitative analysis, while informative, is based on a limited prompt set and manual review; more rigorous user studies would be needed to assess generalization across domains. Third, although top-N decoding improves reliability and controllability, it introduces additional inference cost due to the need to sample and rank multiple outputs per prompt. This may present practical challenges for deployment in latency-sensitive applications. Moreover, even the best-performing configurations exhibit limitations, including factual inaccuracies, shallow reasoning, and limited ability to handle multi-step or compositional tasks. Finally, our experiments focus on a single base model (Qwen2.5 0.5B) and one reward model evaluator. Further work is needed to assess how well these conclusions generalize across different architectures, training scales, and preference optimization paradigms.

Another important set of limitations stems from model size, dataset characteristics, and training constraints. While the use of the Qwen 0.5B model enables faster experimentation and lower resource requirements, its relatively small parameter count may limit its capacity to internalize complex preference structures, especially when compared to the larger models typically used in DPO studies. Furthermore, although the UltraFeedback dataset provides a large and diverse set of comparisons, it is built on crowd-sourced preference data, which may introduce annotator biases or inconsistencies in quality judgments. On the implementation side, computational constraints restricted us to smaller batch sizes, fewer training epochs, and a reduced training set, all of which may prevent the model from fully converging on optimal preference-aligned behavior. The binary nature of preference pairs used in DPO training also introduces simplification, such that it reduces a spectrum of response quality to a discrete choice, which may fail to capture finer-grained distinctions. Finally, our evaluation metrics focus primarily on pairwise preference win rates and do not directly assess important properties such as output diversity, calibration, or whether the model retains its general-purpose capabilities after DPO training. Addressing these limitations is essential for understanding the full trade-offs and potential of preference optimization methods in broader deployment scenarios.

Overall, our results substantiate the effectiveness of DPO over SFT and demonstrate that decoding strategies, like top-N sampling, play an important role in extracting the full benefits of preference-aligned training. These findings suggest a practical and scalable direction for improving the usability of instruction-tuned language models. However, future work is needed to extend these insights to broader settings and evaluate their robustness in real-world applications.

6.1 Future Work

Future work could explore several directions to improve the effectiveness of DPO training, particularly in the context of smaller language models. One promising avenue is the development of hybrid approaches that integrate DPO with other preference learning methods such as Proximal Policy Optimization (PPO) or rejection sampling. These techniques could help mitigate some of the limitations that arise when applying DPO to smaller models with limited representational capacity.

Another area of interest is the design of adaptive KL regularization strategies that dynamically adjust the penalty term based on the training dynamics. Such mechanisms could help maintain a more stable balance between optimizing for preferences and preserving the pretrained knowledge of the base model—an especially important consideration when working with less robust architectures.

Improving training efficiency is also a key goal. Investigating few-shot or low-resource preference learning techniques could reduce the computational burden of training, making it more feasible to deploy preference-aligned models in settings with limited resources. Alongside this, studying the impact of different preference data curation strategies—such as filtering, clustering, or curriculum learning—could reveal methods for maximizing performance using smaller, more targeted datasets.

Finally, future research could focus on developing model architectures or pretraining objectives that better support preference learning in compact models. This includes exploring parameter-efficient fine-tuning methods such as LoRA or adapter-based techniques specifically tailored to DPO or similar objectives. By enhancing the adaptability of smaller models to preference-based training, these efforts could broaden access to alignment techniques and enable more sustainable deployment of preference-tuned language models.

7 Conclusion

In this work, we explored instruction-following performance across supervised fine-tuning, Direct Preference Optimization, and extending it to a test-time reranking strategy using Best-of-N sampling. Our results show that DPO consistently outperforms standard supervised fine-tuning in both quantitative and qualitative evaluations, and that further gains can be achieved through test-time reranking with a strong reward model. These findings highlight the complementary roles of training-time preference optimization and inference-time generation control. While limitations remain in terms of model capacity, data efficiency, and evaluation coverage, our study suggests that even smaller language models can benefit substantially from preference-based learning when paired with effective decoding strategies. Together, these insights contribute to more reliable, aligned, and controllable generation for instruction-following tasks.

8 Team Contributions

- **Renee Qin:** Implemented SFT training loop, developed DPO model pipeline, incorporated Best-of-N reranking experiments with varying sampling temperatures, led reward model evaluation and generated performance visualizations.
- **Nicole Garcia:** Implemented dataset preprocessing for SFT and DPO, implemented and iterated upon DPO training and response generation loop, improved reward margin analysis, and managed model checkpoints, tokenizer setup, and EC2 training environment.

Changes from Proposal While our initial proposal focused on extending the RLOO algorithm by using replay buffers to enable multiple gradient updates from each collected sample, we ultimately pivoted to improving DPO with test-time reranking. We made this shift because we believed it offered a more compelling opportunity to explore how the strengths of different models—specifically preference-aligned training and reward-based selection could be combined to improve instruction-following behavior in compact LLMs.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct Preference Optimization with an Offset. arXiv:2402.10571 [cs.CL] `https://arxiv.org/abs/2402.10571`

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862 [cs.CL] `https://arxiv.org/abs/2204.05862`

Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. 2025. Bootstrapping Language Models with DPO Implicit Rewards. arXiv:2406.09760 [cs.CL] `https://arxiv.org/abs/2406.09760`

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. arXiv:1706.03741 [stat.ML] `https://arxiv.org/abs/1706.03741`

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. arXiv:2310.01377 [cs.CL] `https://arxiv.org/abs/2310.01377`

Lin Gui, Cristina Gârbacea, and Victor Veitch. 2024. BoNBoN Alignment for Large Language Models and the Sweetness of Best-of-n Sampling. arXiv:2406.00832 [cs.CL] `https://arxiv.org/abs/2406.00832`

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, Kenshi Abe, Mitsuki Sakamoto, and Eiji Uchibe. 2025. Evaluation of Best-of-N Sampling Strategies for Language Model Alignment. arXiv:2502.12668 [cs.CL] `https://arxiv.org/abs/2502.12668`

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. 2025. Regularized Best-of-N Sampling with Minimum Bayes Risk Objective for Language Model Alignment. arXiv:2404.01054 [cs.CL] `https://arxiv.org/abs/2404.01054`

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered Direct Preference Optimization. arXiv:2404.13846 [cs.LG] `https://arxiv.org/abs/2404.13846`

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL] `https://arxiv.org/abs/2203.02155`

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG] `https://arxiv.org/abs/2305.18290`

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos, Amélie Héliou, Aliaksei Severyn, Matt Hoffman, Nikola Momchev, and Olivier Bachem. 2024. BOND: Aligning LLMs with Best-of-N Distillation. arXiv:2407.14622 [cs.LG] `https://arxiv.org/abs/2407.14622`

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL] `https://arxiv.org/abs/2109.01652`

Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. 2024. Cal-DPO: Calibrated Direct Preference Optimization for Language Model Alignment. arXiv:2412.14516 [cs.LG] `https://arxiv.org/abs/2412.14516`

A Additional Experiments

Effect of Temperature on Top-N Reranking. To explore the trade-off between diversity and reward quality, we evaluated our DPO+Top-N decoding strategy across different sampling temperatures: $T = 0.25$, $T = 0.6$, and $T = 1.0$. We found that $T = 0.6$ achieved the best win rate (0.9350 vs. SFT), balancing diversity and alignment. Lower temperatures like 0.25 yielded slightly lower win rates (0.9300), possibly due to reduced generation diversity. Higher temperatures such as 1.0 decreased performance (0.8750), likely from increased incoherence.

Reward Margin Monitoring During Training. Throughout DPO training, we tracked reward margins between chosen and rejected responses using the Nemotron-70B model. We observed steady margin improvement early in training, with diminishing returns beyond a certain step threshold. This suggests the importance of early-stage preference alignment and diminishing benefits from extended training on the same data distribution.

Ablation: SFT with Top-N Only. We also evaluated SFT paired with Top-N sampling (without DPO). While SFT+Top-N improved over SFT alone (win rate of 0.9700 vs. 0.9450 against base), it underperformed DPO+Top-N. This reinforces that preference-optimized training and test-time reranking are complementary.

B Implementation Details

All models were trained and evaluated using the Qwen2.5-0.5B architecture. Training was conducted either on an AWS EC2 g4dn.xlarge instance or an AWS EC2 instance with A100 GPU, using PyTorch and Hugging Face Transformers. We used mixed-precision training where applicable to optimize memory usage.

SFT Training. The SFT model was trained on the SmolTalk dataset for 1 epoch with a batch size of 2, gradient accumulation of 8, and a learning rate of 5×10^{-5}. Tokenization was handled with Qwen's chat template, using right-padding and truncation at 512 tokens.

DPO Training. The DPO model was initialized from the SFT checkpoint and trained on the UltraFeedback dataset using a batch size of 4, gradient accumulation of 8, and a learning rate of 1×10^{-6}. The DPO implementation included prompt token masking (max length at 1024) and reward margin logging.

Inference and Reranking. For test-time reranking, we generated 4 completions per prompt using top-k sampling and experimented with temperatures ranging from 0.25 to 1.0. Completions were reranked using the Nemotron-70B reward model via the NVIDIA inference API. All inference experiments were run on GPU with max input length of 1024 tokens and max new tokens set to 1024.

Codebase. The codebase was modularized into training, evaluation, preprocessing, and utilities, with checkpoints logged at regular intervals. Dataset maps and preprocessing were debugged using logging local executions for inspection purposes prior to full-scale training.

Response Generations. The following are sample prompts and responses generated by our DPO model and DPO+Top-N model from our test set for reference purposes.

Prompt 1: How is augmented reality being used to enhance museum experiences?

DPO Model Response	DPO+Top-N Response
Augmented reality (AR) is being used to enhance museum experiences in numerous ways, from immersive virtual tours to interactive exhibits that transport visitors to new and unexplored realms. One of the most exciting applications is in museums, where AR can be used to create immersive, 360-degree views of artifacts, historical events, and cultural significance.\n\nFor instance, museums can use AR to create a 3D model of a piece of art or a historical artifact, allowing visitors to explore it in a more immersive and engaging way. This can include virtual tours, guided tours, or even 360-degree views of the exhibit. AR can also be used to create a sense of depth and context, making visitors feel like they're experiencing the exhibit in a whole new way.\n \nAR can also be used to create interactive exhibits that encourage visitors to engage with the content in a more hands-on and interactive way. For example, a museum could create an AR experience where visitors can explore a piece of art in a 3D model, manipulate the digital version, and see how it would look in real life. This could help visitors develop skills in areas such as art, architecture, and engineering.\n \nAR can also be used to create a sense of social connection and community among visitors. For example, a museum could create an AR experience where visitors can share their own stories and experiences related to a particular exhibit, fostering a sense of community and a sense of belonging. \n\nAnother exciting application of AR in museums is in the realm of education and outreach. For instance, museums can use AR to create virtual reality experiences that transport visitors to ancient civilizations, where they can explore the history and culture of the past. This can help students and visitors learn about different cultures, history, and history of science in a more immersive and engaging way.\n\n Overall, the potential of AR in museums is vast and exciting, offering a new and innovative way to engage visitors with historical and cultural significance. By creating immersive, interactive, and engaging experiences, museums can not only expand their reach and attract more visitors but also become leaders in the field of AR and AR technology.\n	Augmented reality (AR) is transforming the museum experience by seamlessly integrating digital content with physical spaces, creating immersive and engaging experiences for visitors. Here are some ways AR is being used to enhance museum experiences:\n \n1. Interactive exhibits: AR allows visitors to explore exhibits in immersive and dynamic ways by overlaying digital information onto the real world. For example, a museum could create an AR experience where visitors can explore a piece of art in a new and captivating way, or see how a historical event unfolds in a 360-degree virtual reality environment.\n \n2. Virtual tours: AR can provide visitors with immersive virtual tours of museums and historical landmarks, allowing them to explore hidden gems and explore new areas of interest without ever leaving their digital devices. This technology is especially beneficial for museums that want to expand their reach and attract a younger demographic.\n \n3. Collaborative experiences: AR can be used to create collaborative experiences between museum visitors and staff or other users. For instance, a museum could create an AR experience where visitors can collaborate with a virtual assistant to explore a particular exhibit or visit a particular exhibit from different angles.\n \n4. Location-based experiences: AR can be used to create location-based experiences that take visitors to specific locations within the museum. For example, a museum could create an AR experience where visitors can explore a particular exhibit in a specific location, or visit a particular exhibit from a specific angle.\n \n5. Reminders and notifications: AR can also be used to provide visitors with reminders and notifications about upcoming events, exhibitions, and activities, making the museum experience more engaging and interactive.\n \n6. Educational and educational: AR can be used to educate visitors about historical events, cultural heritage, and other topics in a fun and immersive way. For example, a museum could create an AR experience that takes visitors through a virtual museum exhibit, teaching them about a particular historical event or topic.\n \nOverall, AR is a powerful tool for museums that want to make their experiences more engaging, interactive, and immersive for their visitors. By leveraging AR, museums can create new revenue streams, attract a new demographic, and stay at the forefront of technological innovation.\n

Prompt 2: Name a scripting language that is commonly used for web development and supports asynchronous programming.

DPO Model Response	DPO+Top-N Response
One popular scripting language that is commonly used for web development and supports asynchronous programming is JavaScript. JavaScript is a dynamic, interpreted, and cross-platform programming language that can be used to create interactive and dynamic web pages. It is widely used in web development, including creating user interfaces, handling user input, and running JavaScript code in response to user actions.\n \nJavaScript also supports asynchronous programming through its event loop, which allows for the execution of multiple tasks concurrently. This enables the creation of asynchronous JavaScript and CSS (ASGI) applications, which can handle multiple requests concurrently without blocking the main thread.\n \nSome popular JavaScript frameworks and libraries that support asynchronous programming include Node.js, which is a JavaScript runtime environment that allows for the execution of asynchronous code, and React, a JavaScript library for building user interfaces.\n \nAdditionally, JavaScript can also be used with other programming languages, such as Node.js and Python, to create scalable and high-performance web applications.\n	One popular scripting language that is commonly used for web development and supports asynchronous programming is JavaScript. JavaScript is a dynamic, interpreted, and cross-platform programming language that can be used to create interactive and dynamic web pages. It is widely used in web development, including creating user interfaces, handling user input, and running JavaScript code in response to user actions.\n \nJavaScript also supports asynchronous programming through its event loop, which allows for the execution of multiple tasks concurrently. This enables the creation of asynchronous JavaScript and CSS (ASGI) applications, which can handle multiple requests concurrently without blocking the main thread.\n \nSome popular JavaScript frameworks and libraries that support asynchronous programming include Node.js, which is a JavaScript runtime environment that allows for the execution of asynchronous code, and React, a JavaScript library for building user interfaces.\n \nAdditionally, JavaScript can also be used with other programming languages, such as Node.js and Python, to create scalable and high-performance web applications.\n

Prompt 3: [Q]: Write a multi-choice question for the following article: Article: Lisa is a 10-year-old girl who loves to read books. Every Saturday, she visits the local library to borrow new books. Her favorite genre is fantasy. This Saturday, she borrowed a book about dragons and finished reading it in just one day. Her mother was surprised and asked how she read it so quickly. Lisa smiled and said, \"I just couldn't put it down!\" \n****\n[A]: Question: Why did Lisa finish the book so quickly?Options:A she didn't like the bookB she read it with her motherC she couldn't put it downD she skipped many pagesAnswer:C

DPO Model Response	DPO+Top-N Response
[A]: Question: Where did Lisa borrow the book from?Options:A the libraryB the bookstoreC the schoolD the parkAnswer:A\n	[C]\n

Prompt 4: Create a metaphor to describe the feeling of nostalgia.

DPO Model Response	DPO+Top-N Response
The weight of memories, like autumn leaves on a tree, slowly settles in, a poignant reminder of the past. It’s a weight that feels heavier than the weight of the world, a crushing burden that refuses to let go. The nostalgia, a lingering ache, gnaws at the edges of my mind, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of nostalgia, a crushing burden that refuses to let go, is a poignant reminder of the past, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched. The weight of memories, a poignant reminder of the past, a crushing burden that refuses to let go, a lingering ache that refuses to be satisfied. It’s a feeling that’s hard to shake, a longing that’s never truly quenched.	The weight of memories, like autumn leaves, slowly unfurl, revealing a tangled tapestry of stories and sights that have been hidden away in the past. The ache of longing, like a longing ache, gnaws at the mind, but the weight of those memories is a gentle reminder, a comforting comfort, that holds the past in place, waiting for the day when it can be relived, cherished, and loved.\n