Extended Abstract

Motivation Models trained with SFT and DPO on SmolTalk and UltraFeedback achieved lower loss and higher win rates, but qualitative results fell short. Deeper analysis revealed notable limitations within existing datasets. Although SmolTalk and UltraFeedback are high-quality, they lack sufficient diversity in both prompts and responses.

Method Augment and align. Ultimately, our ultimate goal is to improve the overall user experience in chatbot interactions. Effective data augmentation represents a valuable and practical next step to accelerate our progress. Ideally, we would vastly expand our training corpus by gathering extensive, diverse and high-quality dialogue data from web sources. This approach would significantly improve the understanding of the model and improve its conversational capabilities.

Implementation We begin with a pretrained Qwen2.5-0.5B model, initially fine-tuned (SFT) on the smol-smoltalk dataset (460k pairs), followed by Direct Preference Optimization (DPO) using 61k UltraFeedback samples.

To diversify instructions and expand task coverage, we generate 20k synthetic prompts with Mixtral-8x7B-Instruct, targeting ten categories via a diversity-focused prompt template. Using MiniLM sentence embeddings and FAISS-based filtering, we build a refined prompt set of 40k instructions: 20k synthetic, 10k from smol-smoltalk, and 10k from UltraFeedback.

For each prompt, Mixtral-8x7B-Instruct generates two responses - a high-quality (chosen) and a degraded (rejected) version. Finally, we perform a second round of DPO fine-tuning with these 40k preference pairs, further enhancing model performance.

Results Here are summarized quantitative and qualitative results.

Example input prompt **"Imagine a world where virtual reality has surpassed reality in terms of entertainment and social interaction. Write a third-person narrative about a character's journey as they navigate this new world, including details about their daily life, relationships, and the challenges they face in balancing their virtual and real-life experiences"**.

Model	Win-Rate Over Base	Example Response	Comment
Qwen2.5-0.5B	N/A	The world we live in today is a world of virtual reality. It's a world where we can experience the world as we want to, from anywhere in the world. It's a world where we can travel to other planets and explore them...	The response is verbose, 7,778 characters in total. Miss 3rd person narrative.
Augmented DPO	0.63	**Amari wakes to blinking neon horizons, trading his cramped flat's stale air for Veloria's simulated oceans. He dances with code-friends, yet headaches throb and real muscles weaken. Torn, he logs off to touch sunlight, unsure which life is his.**	This answer is preferable

Table 1: Quantitative & Qualitative Results.

Discussion LLM data augmentation is a more practical way compared to using expensive human-labeled data. The model was pretty sensitive to how we started training, getting hyperparameters exactly right was tricky, and we ran into computational limitations. Going forward, we'll need broader and more varied datasets, ways to train more efficiently, and methods that make the training less delicate and more robust.

Conclusion Our experiments demonstrated that combining SFT, DPO, and targeted data augmentation significantly enhances chatbot performance. Initially, training with SmolTalk and UltraFeedback improved response alignment but lacked diversity. Using Mixtral-8x7B-Instruct for data augmentation and FAISS-based filtering significantly resolved these issues, emphasizing the importance of precise hyper-parameter tuning and leveraging LLM synthetic data.

Augment and Align: Leveraging LLMs for Improved Preference Data in DPO

Li Miao
Stanford University
limiao@stanford.edu

Haoran Qi
Stanford University
ryanqi7@stanford.edu

Yue Shen
Stanford University
ysh2025@stanford.edu

Abstract

Models trained using Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) on the SmolTalk and UltraFeedback datasets achieved reduced loss and higher win rates, yet qualitative performance remained limited. Analysis pinpointed insufficient diversity in both prompts and responses within existing datasets. To address our core goal of enhancing chatbot user experience, it requires effective data augmentation. Expanding the training corpus with diverse, high-quality dialogues sourced from the web and LLM significantly improves the model's comprehension and conversational abilities.

Our experiments revealed substantial improvements in chatbot performance through integrating SFT, DPO, and targeted data augmentation. Initial training on SmolTalk and UltraFeedback improved response alignment but exhibited limited diversity. Employing Mixtral-8x7B-Instruct for generating synthetic data and applying FAISS-based filtering effectively addressed these shortcomings. This strategy underscored the critical importance of precise hyperparameter tuning and leveraging synthetic data from Large Language Models (LLMs), presenting a practical alternative to expensive human-labeled data.

Despite these advancements, the model exhibited sensitivity to initial training conditions and hyperparameter settings, alongside computational constraints. Future work will focus on incorporating broader, more varied datasets, optimizing training efficiency, and developing more robust training methodologies.

1 Introduction

In this project, we implemented Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), Reinforce Leave-One-Out (RLOO), and a novel data augmentation extension based on the Qwen2.5-0.5B language model, using preference datasets (SmolTalk and UltraFeedback). For evaluation, we constructed inference pipelines to generate samples using controlled temperature and sampling methods, and employed the Llama 3.1 Nemotron 70B Reward Model to compute win-rate metrics against the baseline Qwen2.5-0.5B-Instruct model. Each incremental step demonstrated clear performance enhancements.

Most notably, our data augmentation and alignment approach significantly improved model capabilities, achieving the highest win rate compared to the baseline. Our augmentation strategy involved generating high-quality synthetic responses from a mixture of expert language models, coupled with fine-grained distinctions between high-quality and borderline or suboptimal responses. By carefully tuning hyperparameters, our preference algorithms (DPO and RLOO) effectively leveraged these enhancements, demonstrating robust and measurable improvements in performance.

Our models trained with SFT and DPO on the provided datasets (SmolTalk and UltraFeedback) showed significant improvements in terms of loss reduction and win-rate. However, the quantitative results have not fully met our initial expectations. By examining qualitative examples, we gained

Stanford CS224R 2025 Final Report

deeper insights into the challenges and progress, which guided our attention toward the broader goals of our project.

Ultimately, our objective is to develop a chatbot system capable of generating highly satisfying responses given user prompts. SmolTalk offers high-quality dialogues, featuring clear user prompts paired with exemplary assistant responses. Using Qwen2.5-0.5B as our base model, we applied Supervised Fine-Tuning (SFT) to teach the model appropriate response patterns and structures. Before SFT training, the Qwen2.5-0.5B model frequently produced incoherent text, random tokens, or improperly formatted outputs. SFT notably addressed these issues, leading to more structured and meaningful responses.

In subsequent stages, namely DPO and RLOO, we leveraged preference-based datasets containing pairs of preferred and rejected responses to further align the model with human judgments. Particularly in DPO, our goal was to emphasize the differences between good and poor responses, encouraging the model to favor responses aligned with human preferences. As a result, the model not only learned appropriate textual structures but also internalized nuanced human preferences. We observed substantial improvements, including more relevant responses, increased win rates, and richer conversational capabilities.

Nonetheless, deeper analysis revealed notable limitations within the existing datasets. Although SmolTalk and UltraFeedback are high-quality, they lack sufficient diversity in both prompts and responses. Ideally, we would vastly expand our training corpus by gathering extensive, diverse, high-quality dialogue data from web sources. This approach would significantly enrich the model's understanding and enhance its conversational capabilities. Ultimately, our primary aim remains improving the overall user experience in chatbot interactions, and effective data augmentation represents a valuable and practical next step to accelerate our progress.

2 Related Work

Synthetic Preference Data for Model Alignment Dong et al. (2025) introduce an iterative framework called SynPO (Self-Boosting LLMs), where a small supervised model generates synthetic prompt–response pairs and fine-tunes on them for alignment, achieving substantial benchmark gains. This closely parallels our pipeline using Mixtral-generated prompts for preference bootstrapping.

Synthetic Data Generation Strategies Scale AI's study on synthetic data strategies (Chan et al. (2024)) compares techniques like answer augmentation, question rephrasing, and new-question generation. They emphasize that strategy effectiveness depends on seed data size and query budget. Our approach similarly generates large-scale synthetic prompts and applies FAISS-based clustering to ensure diversity and quality.

DPO Theoretical Foundation and Enhancement. A comprehensive survey (Gou and Nguyen (2025)) of DPO reviews theoretical underpinnings, data curation strategies, stability challenges, and emerging variants. In addition, there are many extensions to standard DPO. ODPO (Offset DPO) (Amini et al., 2024) introduces a margin in the DPO loss to better scale preference strength and prevent over-suppression on closely scored pairs. Curry-DPO (Pattnaik et al. (2024)) incorporate curriculum learning by ordering preference pairs from easy-to-hard, showing improvements on UltraFeedback-like benchmarks. Pre-DPO (Pan et al. (2025)) employs a stronger reference model to dynamically reweight samples, improving utilization and achieving gains on AlpacaEval and Arena-Hard.

3 Method

To build a high-quality and diverse prompt set, we use a multi-stage selection process combining semantic clustering, quality filtering, and de-duplication. First, we gather large pools of candidate prompts from the `smol-smoltalk` and `ultrafeedback_binarized` datasets, as well as from our synthetic Mixtral-generated prompts. We encode all candidates into vector embeddings (e.g., with a sentence-transformer MiniLM) and use FAISS-based filtering to group similar instructions. Sampling from each cluster ensures broad topical coverage. This "macro-level" clustering strategy preserves dataset diversity by covering a wide range of instruction types.

- **Embedding-based clustering:** We compute sentence embeddings for all prompts (using MiniLM) and apply clustering (via K-Means/HDBSCAN). Each prompt's embedding serves as a semantic representation, and clustering partitions the dataset into groups of similar instructions. We sample from each cluster to ensure diverse coverage. For instance, we may select the *cluster centroid* or another central example from each cluster, helping to include rare or underrepresented instruction types.
- **Representative selection:** Within each cluster, we select prompts that are central or informative. One approach is to pick the prompt closest to the cluster centroid (medoid). Alternatively, we apply a *farthest-point sampling* strategy: start with a seed prompt, then iteratively add the prompt whose embedding is farthest from all already-selected prompts. This maximizes the minimum pairwise embedding distance and helps avoid near-duplicate selections.
- **Quality filtering:** To ensure high-quality prompts, we optionally rank or score them before selection. We use a small scoring model to evaluate instruction "quality". Simpler heuristics is also used, such as discarding prompts that are too short or malformed. Within each cluster, we choose the top-rated prompts according to our filtering criterion. This two-step "Clustering and Ranking" (CaR) approach has been shown to yield a small but high-quality and diverse subset of prompts.
- **De-duplication (FAISS filtering):** After initial selection, we re-embed the selected prompts and build a FAISS index to detect near-duplicates. Prompts whose nearest neighbors in embedding space exceed a similarity threshold (e.g., cosine similarity > 0.95) are removed. This semantic de-duplication ensures that the final prompt set is not only diverse but also individually distinct.

For synthetic Mixtral-generated prompts, we apply the same pipeline: generate a large pool (e.g., ∼40K) of candidates, embed them, and apply clustering, farthest-point sampling, and FAISS filtering to downsample to 20K unique, high-coverage prompts. Over-generating and pruning based on semantic similarity ensures that the synthetic set is novel and varied.

Altogether, this methodology yields three sets of prompts—10K from `smol-smoltalk`, 10K from `ultrafeedback`, and 20K synthetic—that are maximally informative, high-quality, and diverse, as recommended by recent studies on instruction-tuning data.

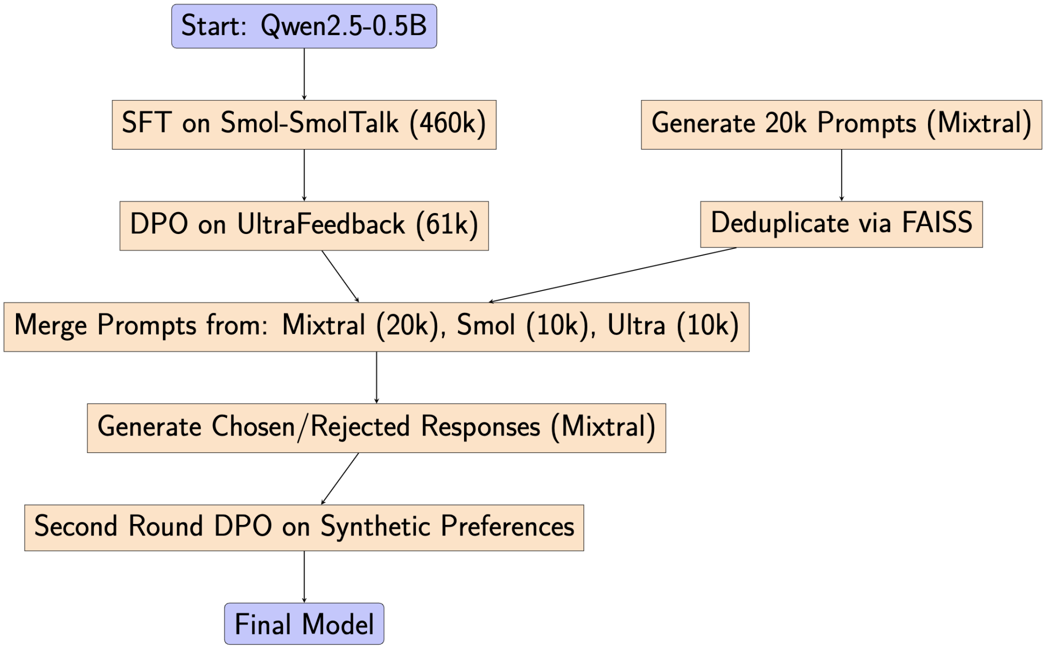

Start: Qwen2.5-0.5B
SFT on Smol-SmolTalk (460k)
Generate 20k Prompts (Mixtral)
DPO on UltraFeedback (61k)
Deduplicate via FAISS
Merge Prompts from: Mixtral (20k), Smol (10k), Ultra (10k)
Generate Chosen/Rejected Responses (Mixtral)
Second Round DPO on Synthetic Preferences
Final Model

Figure 1: Method Overview.

4 Experimental Setup

This project investigates how Direct Preference Optimization (DPO) can improve the instruction-following capabilities of a small language model. Our goal is to train and evaluate a `Qwen2.5-0.5B` model to generate high-quality, helpful, and instruction-compliant responses. The experimental pipeline consists of dataset construction, supervised pretraining (SFT), DPO fine-tuning, DPO fine-tuning with synthetic data augmentation and evaluation.

4.1 Data Collection

To create a diverse and representative instruction dataset, we constructed a total of 40,000 prompt instances from three sources:

- **Smol-Smoltalk (10k prompts):**
  We sample 10,000 prompts from the `HuggingFaceTB/smol-smoltalk` dataset. Sampling is performed using semantic clustering to maximize topical diversity and minimize redundancy.
- **UltraFeedback (10k prompts):**
  Similarly, we sample 10,000 prompts from the `HuggingFaceH4/ultrafeedback_binarized` dataset, ensuring both coverage and diversity using the same clustering-based selection strategy.
- **Synthetic Prompts (20k prompts):**
  We generate 20,000 synthetic instructions using a hosted `Mixtral-8x7B` model. Candidate prompts are filtered using MiniLM embeddings and FAISS-based similarity search to remove near-duplicates and maintain diversity.

Each of the 40,000 final prompts is then passed through `Mixtral-8x7B` with two different system prompts to produce a chosen (higher quality) and rejected (lower quality) response. The result is a dataset of 40k triplets: `(prompt, chosen response, rejected response).`

4.2 Model and Training

The base model used is `Qwen2.5-0.5B`, an instruction-tuned language model. The training pipeline consists of two stages:

Stage 1: Supervised Fine-Tuning (SFT)

We initialize the model with the original `Qwen2.5-0.5B` weights and perform SFT on the 460k prompts from `Smol-Smoltalk`. Training is run for 2 epochs with:

- Learning rate: `1e-5`
- Batch size: `4`
- Max sequence length: `512 tokens`

Stage 2: Direct Preference Optimization (DPO)

We train the model checkpoint acquired from SFT using DPO on the 61K `ultrafeedback binarized` prompt responses triplet. Training is run for 3 epochs with:

- Learning rate: `1e-6`
- Batch size: `3`
- Max prompt length: `128 tokens`
- Max response length: 768 tokens

Stage 3: Direct Preference Optimization (DPO)

We conduct DPO training run on the augmented 40k dataset. It is trained for 3 epochs with:

- Learning rates `1e-6`
- Batch size of `4`
- Max prompt length: `256 tokens`
- Max response length: `1024 tokens`

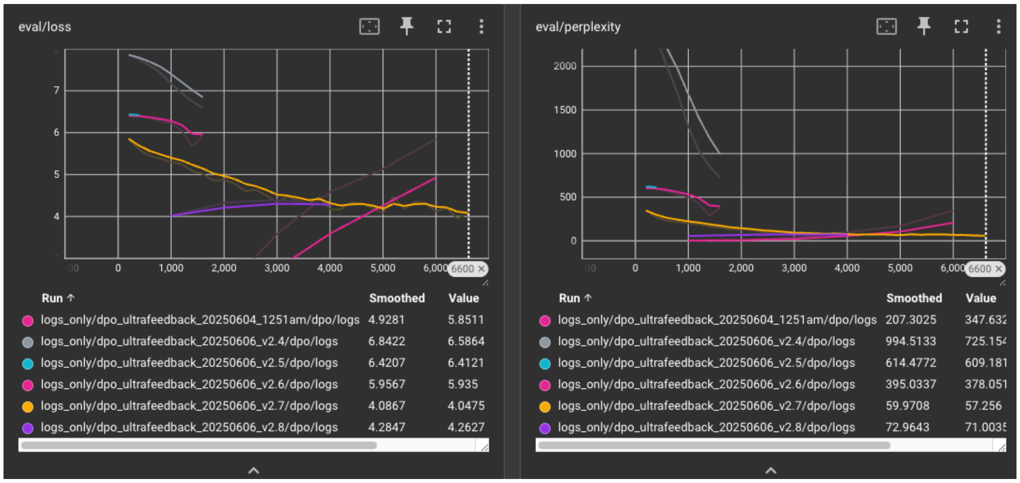
4.3 Evaluation

Model performance is evaluated using a standardized set of 400 held-out prompts provided by the course. For each prompt, we generate a response using the trained model and compare it against a strong baseline model response. Judgments are made by the `Nemotron-70B` model, which acts as a preference evaluator to determine which response better follows the instruction. `Nemotron-70B` model generates reward scores for both the trained and reference models, and we calculate the win-rate as average of the binary label over all prompts.

This setup allows us to quantify how DPO training with different data sources influences instruction-following performance, and to compare synthetic data against real data in a controlled setting.

5 Results

5.1 Quantitative Evaluation



Run ↑	Smoothed	Value
● logs_only/dpo_ultrafeedback_20250604_1251am/dpo/logs	4.9281	5.8511
● logs_only/dpo_ultrafeedback_20250606_v2.4/dpo/logs	6.8422	6.5864
● logs_only/dpo_ultrafeedback_20250606_v2.5/dpo/logs	6.4207	6.4121
● logs_only/dpo_ultrafeedback_20250606_v2.6/dpo/logs	5.9567	5.935
● logs_only/dpo_ultrafeedback_20250606_v2.7/dpo/logs	4.0867	4.0475
● logs_only/dpo_ultrafeedback_20250606_v2.8/dpo/logs	4.2847	4.2627

Run ↑	Smoothed	Value
● logs_only/dpo_ultrafeedback_20250604_1251am/dpo/logs	207.3025	347.632
● logs_only/dpo_ultrafeedback_20250606_v2.4/dpo/logs	994.5133	725.154
● logs_only/dpo_ultrafeedback_20250606_v2.5/dpo/logs	614.4772	609.181
● logs_only/dpo_ultrafeedback_20250606_v2.6/dpo/logs	395.0337	378.051
● logs_only/dpo_ultrafeedback_20250606_v2.7/dpo/logs	59.9708	57.256
● logs_only/dpo_ultrafeedback_20250606_v2.8/dpo/logs	72.9643	71.0035

Figure 2: Ultrafeedback Loss Across DPO Iterations.

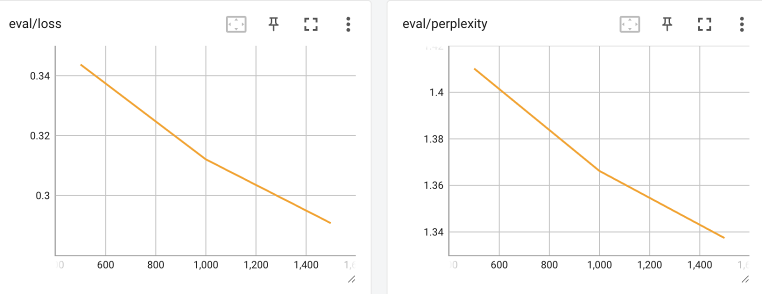


Figure 3: Customized Synthetic Data Loss Across DPO Iterations on Synthetic Augmented Data .

As shown in Figure 2, we conducted numerous experiments, each requiring substantial computational time due to limited GPU resources (AWS g5.xlarge instances). Training runs were frequently interrupted or terminated mid-way due to resource constraints or instance shutdowns. Additionally, each iteration involving hyperparameter tuning or debugging significantly prolonged experiment

Method	Win rate over base model
DPO Ultrafeedback	0.56
DPO synthetic data augmentation	0.63

Table 2: Performance Comparison

durations, typically requiring several hours. Our longest experiment, indicated by the orange line, lasted approximately 19 hours. We experimented with various parameters - learning rate range from 5e-6 to 3e-5, batch size from 1 to 4, accumulation steps from 1 to 8, beta from 0.2 to 0.7, even random initialization seed. With an optimized final setup and stage-specific parameter tuning, we achieved a consistent decrease in evaluation set loss. Note that many experiments presented here are not independent. Ideally, the loss curve would appear continuous; however, due to frequent interruptions and data loss, we had to manually segment and re-continue the model training loss into separate stages, each starting anew from step 0.

The evaluation results from DPO training on the 40k synthetic dataset show a consistent and steady improvement across both loss and perplexity metrics. As depicted in Figure 3, the evaluation loss decreases from approximately 0.343 to 0.296 over the course of training, indicating that the model's ability to align with the chosen responses improves progressively. Similarly, the evaluation perplexity drops from around 1.41 to 1.33, suggesting that the model's predictive confidence and fluency in generating preferred responses increase during training. The monotonic downward trends in both curves imply stable training dynamics without overfitting, and demonstrate that DPO effectively exploits the preference signal embedded in the synthetic prompt-response pairs.

Table 2 summarizes the win-rate analysis we have conducted offline.

5.2 Qualitative Analysis

Table 3 a concrete qualitative example with prompt **"Imagine a world where virtual reality has surpassed reality in terms of entertainment and social interaction. Write a third-person narrative about a character's journey as they navigate this new world, including details about their daily life, relationships, and the challenges they face in balancing their virtual and real-life experiences"**.

6 Discussion

Our experiments demonstrated significant enhancements in chatbot performance using a combined approach of Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and data augmentation techniques. The initial SFT and DPO models, trained on high-quality datasets such as SmolTalk and UltraFeedback, effectively improved structured response patterns and alignment with human preferences. Despite these advancements, qualitative evaluations revealed limitations primarily due to insufficient diversity within the provided datasets.

To address this, we implemented a novel augmentation strategy utilizing the Mixtral-8x7B-Instruct model to generate diverse and high-quality synthetic instructions. Through sophisticated filtering mechanisms, such as FAISS-based approximate nearest neighbor search, we ensured prompt uniqueness and breadth. This approach significantly boosted the performance of our DPO-trained model, evident in its superior win-rate metrics against baseline models. Notably, leveraging larger models for synthetic response generation proved highly effective, suggesting that high-capacity models can substantially enhance the training quality for smaller models.

However, our study encountered several practical challenges. Sensitivity to model initialization was pronounced, as the effectiveness of DPO depended significantly on the quality of the initial SFT checkpoint. Additionally, precise tuning of hyper-parameters - particularly the learning rate and inverse-temperature parameter β - proved critical; even minor deviations substantially affected model performance. Furthermore, computational constraints imposed limits on experimentation by restricting batch sizes and prolonging training durations, underscoring the necessity for more efficient computational strategies or enhanced resources for large-scale training.

Model	Response	Comment
Qwen2.5-0.5B	The world we live in today is a world of virtual reality. It's a world where we can experience the world as we want to, from anywhere in the world. It's a world where we can travel to other planets and explore them...	The response is verbose, 7,778 characters in total. Miss 3rd person narrative.
SFT on SmolTalk	As she stood in the living room, surrounded by her virtual world, Emily felt the thrill of escape. She had made a decision that would change her life forever, and she was going to have to confront the consequences...	The response is incomplete (383 characters), and vague.
DPO on Ultrafeedback	As she sat at her desk, surrounded by sleek, futuristic machines and glowing holographic projections, Emily felt the weight of her virtual reality world slowly suffocating her. She had grown up in this virtual realm, and it was the only place she felt truly at home. Her thoughts were a jumble of cybernetic enhancements, virtual friends, and endless scrolling through her favorite social media feeds ...	This answer is again verbose, which has 2680 characters
DPO on Augmentation	**Amari wakes to blinking neon horizons, trading his cramped flat's stale air for Veloria's simulated oceans. He dances with code-friends, yet headaches throb and real muscles weaken. Torn, he logs off to touch sunlight, unsure which life is his.**	This answer is preferable

Table 3: Response examples.

One limitation of our current experimental setup lies in the way chosen and rejected responses were generated. We relied on system prompts to elicit a high-quality (chosen) and a lower-quality (rejected) response from the Mixtral-8x7B model. While this approach is simple and scalable, it lacks robustness in distinguishing subtle differences in response quality. A more principled approach would involve sampling multiple candidate responses from a smaller model (e.g., Qwen-2.5-0.5B-Instruct), scoring them using a strong evaluator model like Nemotron-70B, and get the mean score as a reference. Specifically, by tuning temperature and top_p to control diversity, we could retain three responses above the mean scofre and three below using Mixtral8x7b model, choosing the highest and lowest scoring ones for DPO supervision. We hypothesize that this method would lead to a more discriminative dataset and thus better preference alignment during DPO training. Unfortunately, due to computational constraints, we were unable to implement this refinement in the current work.

7 Conclusion

This project demonstrated that strategically designed data augmentation combined with preference-based training methods significantly enhances chatbot conversational capabilities. The success of our synthetic data generation and filtering techniques indicates a viable pathway for scaling preference training without costly human annotations. Nonetheless, addressing model initialization sensitivity, hyper-parameter fragility, and computational limitations remains crucial for further progress. Future work should focus on expanding dataset diversity through extensive web-sourced dialogues, refining computational efficiency, and exploring more robust training methods to maximize model performance and generalizability.

8 Team Contributions

- **Li Miao:** Implemented SFT, DPO, and RLOO algorithm, developed preference data loader, built debugging tools, conducted DPO experiments to systematically explore hyperparameters & analyze performance
- **Haoran Qi:** Implemented SFO, DPO algorithm and integrated the end-to-end flow of training. Built the inference and evaluation tooling for our models, and responsible for conducting experiments.
- **Yue Shen:** Implemented the extension part of the project, developed the pipeline to filter prompts from given dataset, generate prompts using open-source model, and produce chosen, rejected response pair. Conducted the DPO experiments on synthetic augmented data.

Changes from Proposal This project ultimately evolved into collaborative teamwork. Initially, team responsibilities were clearly divided — for instance, one member focused exclusively on DPO implementation, another on developing the data loader, and another on running experiments. However, due to the complexity, prolonged experiment durations, and slow turnaround times, we shifted towards a collective approach. We jointly reviewed and refined the data loader, collaborated closely to debug the DPO and RLOO algorithms, coordinated experiments to quickly identify and terminate ineffective runs, and collectively explored optimal hyperparameter configurations. We worked together on preparing the final poster and report documentation.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct Preference Optimization with an Offset. In *Findings of the Association for Computational Linguistics (ACL) 2024*. Association for Computational Linguistics, Bangkok, Thailand, 9954–9972. https://doi.org/10.18653/v1/2024.findings-acl.592

Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. 2024. Balancing Cost and Effectiveness of Synthetic Data Generation Strategies for LLMs. *arXiv preprint* 2409.19759 (2024). Accepted to FITML Workshop @ NeurIPS 2024.

Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. 2025. Self-Boosting Large Language Models with Synthetic Preference Data (SynPO). In *ICLR 2025 Poster*. arXiv:2410.06961.

Qi Gou and Cam-Tu Nguyen. 2025. A Survey of Direct Preference Optimization: Theory, Variants, and Challenges. *arXiv preprint* 2503.11701 (Mar 2025).

Junshu Pan, Wei Shen, Shulin Huang, Qiji Zhou, and Yue Zhang. 2025. Pre-DPO: Improving Data Utilization in Direct Preference Optimization Using a Guiding Reference Model. arXiv:2504.15843 [cs.CL] https://arxiv.org/abs/2504.15843

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-DPO: Enhancing Alignment using Curriculum Learning Ranked Preferences. arXiv:2403.07230 [cs.CL] https://arxiv.org/abs/2403.07230