Extended Abstract

Motivation Despite the rapid progress in Large Language Models (LLMs), current methods for preference alignment and reasoning—especially on complex, multi-step tasks like instruction following and mathematical problem solving—remain limited. Existing techniques such as Direct Preference Optimization (DPO) often require extensive tuning and still struggle to reliably surpass competitive baselines. Motivated by these gaps, our work seeks to systematically evaluate and improve LLM alignment strategies across both preference-driven and reasoning-intensive tasks, highlighting practical challenges and new directions for robust model alignment.

Method We evaluate multiple learning algorithms for language model alignment, including Direct Preference Optimization (DPO), Supervised Fine-Tuning (SFT), and RLOO-based reinforcement learning. In addition to standard DPO, we explore Exploratory Preference Optimization (XPO) and Improved Preference Optimization (IPO), both implemented as minor modifications to the DPO loss in PyTorch. Our experiments are conducted on the `Qwen-2.5-0.5B` base model, using both human and synthetic preference data generated via asynchronous ChatGPT-4.1 Mini calls for the Ultrafeedback and Countdown benchmarks. We systematically tune hyperparameters and apply both LoRA-based parameter-efficient adaptation and full-model fine-tuning, with standardized preprocessing and right-padding for fair comparison across all methods and extensions.

Implementation Our implementation utilized PyTorch and data loaders for SFT. We implemented DPO by constructing the standard pairwise log-ratio loss in PyTorch, optimizing models to prefer human-preferred responses over rejected ones. For XPO and IPO, we introduced minor code modifications to the DPO pipeline to alter the preference loss structure, enabling broader exploration or improved reward matching. To augment training data, we leveraged asynchronous ChatGPT 4.1 Mini API calls to generate additional preference pairs and synthetic feedback to introduce diversity into our datasets. The full pipeline utilizes the Hugging Face Transformers library.

Results We find that with DPO, we were able to achieve just under the 0.1 threshold at 0.0975. We observe that the DPO loss curve does not deviate significantly from 0.69, which we computed is the probability of a model not having a preference between the preferred and dispreferred responses. We also evaluate two variants of DPO, namely eXploratory Preference Optimization (XPO) and Identity Preference Optimization (IPO), both of which achieve around 0.03 on the leaderboard despite an extensive hyperparameter search. Finally, we create a synthetic dataset with `gpt-4.1-mini` with instructions to create preferred generations that align with the criteria of the Nemotron model. While we see loss curves go down, we do not beat the performance of baseline DPO.

Discussion After extensive debugging and printing out model training metrics like KL divergence and gradient norms, we were unable to surmise as to why DPO was not more performant, and by extension, the variants of DPO and the synthetic data generation approach. We would like to note that our DPO model and XPO were comfortably beating the baselines on the old leaderboard. We performed extensive hyperparameter searches, considered tricks like right padding, masking various parts of the prompt and response etc. but did not find critical bugs to our approach.

Conclusion We benchmarked a range of alignment strategies—including SFT, DPO, RLOO, LoRA, and recent extensions such as XPO, IPO, and synthetic data augmentation—on instruction-following and mathematical reasoning tasks. Our results demonstrated that preference alignment with DPO is notably challenging, even with exhaustive hyperparameter tuning due to the quality of data present and the limitations of a 0.5B model. We also explored the countdown task and tried full parameter finetuning and LoRA and found LoRA achieved higher SFT performance. We implemented RLOO with a basic reward function that penalized attributes such as irrelevant English phrases and rewarded the use of valid numbers and found that it did not beat the best SFT baseline of 0.18, achieving only 0.12.

Exploratory Preference Optimization and Synthetic Dataset Generation for Preference Alignment

Pranshu Chaturvedi
Department of Computer Science
Stanford University
pranshu@stanford.edu

Parth Shroff
Department of Electrical Engineering
Stanford University
pmshroff@stanford.edu

Abstract

Large language models (LLMs) have achieved impressive results in instruction following and dialogue, yet optimizing them to align closely with human preferences—especially in challenging domains such as mathematical reasoning—remains difficult. In this work, we systematically evaluate preference-based and verifier-based fine-tuning methods on top of a strong instruction-following baseline,`Qwen-2.5-0.5B`. We first establish Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) baselines using datasets including SmolTalk, CogBehave, and Ultrafeedback. Despite rigorous hyperparameter sweeps, we find that surpassing established benchmarks with DPO remains highly challenging, revealing the sensitivity of preference-based optimization to both data and algorithmic choices. Building on this, we investigate several recent extensions: eXploratory Preference Optimization (XPO) and Identity Preference Optimization (IPO), synthetic data augmentation and parameter-efficient Low-Rank Adaptation (LoRA). We also evaluate reinforcement learning with the RLOO algorithm on the Countdown dataset, a verifier-based arithmetic benchmark. Our experiments highlight both the promise and practical limitations of current alignment techniques, while preference-based objectives like DPO require careful tuning and may present steep optimization barriers. We discuss the implications of these findings for future work in LLM alignment and reward modeling.

1 Introduction

We decided to do the default project which involved preparing and processing diverse datasets, including both preference-based and verifier-based data, and constructing efficient data pipelines for training and evaluation. We implement several core RL methods that fine-tuned the base model for instruction following and mathematical reasoning. In particular, we generated datasets for SFT where we warm-started a pre-trained model and used high-quality instruction-following data. We then utilized DPO Sharma et al. (2023), which aligns models to human preferences using paired response data. For our extensions, we focused on two common variants of DPO including eXploratory Preference Optimization (XPO) Xie et al. (2024) and Identity Preference Optimization (IPO) Gheshlaghi Azar et al. (2024). We also used vanilla DPO with synthetic data augmentation where we had `gpt-4.1-mini` generate 2500 examples using prompts that mirror those present in `Ultrafeedback` Cui et al. (2024) as a preferred and dispreferred completion from `gpt-4.1-mini` for each example. Finally, for extra credit, we implemented an online policy-gradient approach, REINFORCE Leave-One-Out (RLOO), which further explores reward-based optimization and requires either training or explicitly reward model based on the Bradley-Terry objective.

Stanford CS224R 2025 Final Report

Figure 1: SFT Loss

2 Related & Prior Work

`Qwen-2.5-0.5B` Team (2024) is an open-source causal language model released as part of the QWEN 2.5 family, designed for efficient and high-quality text generation in conversational and instruction-following contexts. The model adopts the standard transformer-based decoder-only architecture, comprising several transformer layers with multi-head self-attention, layer normalization, and feed-forward submodules. The models are pre-trained on an 18 trillion token dataset and incorporates intricate supervised finetuning with over one million samples and multistage reinforcement learning. `Qwen-2.5-0.5B` specifically has a 32,768 context length although other variants support longer context lenghts of 128,000 tokens.

2.1 SFT

To enhance the model's ability to follow instructions and engage in dialogue, we employ supervised fine-tuning (SFT) on curated instruction datasets. SFT involves exposing the model to high-quality demonstration data consisting of user prompts and corresponding ground-truth assistant responses. During fine-tuning, the model parameters are updated via standard maximum likelihood estimation, with a cross-entropy loss applied only to the assistant's outputs—ensuring that the model learns to map user queries to contextually appropriate completions. Our training methodology leverages the official Qwen conversational template, formatting each dialogue into interleaved user and assistant turns, demarcated by special tokens (e.g., `<|im_start|>`, `<|im_end|>`).

The first set of experiments prior to the extension involved enabling robust instruction-following capabilities in large language models, we implemented a supervised fine-tuning (SFT) pipeline leveraging the `QWEN2.5-0.5B` base model and utilized the `smol-smoltalk` task-oriented dataset. The pipeline preprocesses each the dataset into a unified chat template format compatible with QWEN's conversational prompting schema. The QWEN 2.5 chat template uses ChatML format with role-based tokens.

```
<|im_start|>system\nYou are a coding assistant<|im_end|>\n
<|im_start|>user\nWrite Python code to reverse a string<|im_end|>\n
<|im_start|>assistant\n
```

The chat template itself is structured as follows. We note that after speaking with Anikait, it appears that system prompts are ineffective for this particular QWEN model.

```
messages = [
    {"role": "system", "content": "You are a helpful assistant"},
    {"role": "user", "content": "Write Python code to reverse a string"}
]
```

We employ a custom masking strategy to ensure that loss is computed exclusively over assistant-generated responses, thus preventing user instructions from influencing the language modeling objective. Training proceeds for one epoch over 460k examples using the AdamW optimizer with a cosine learning rate scheduler and a linear warmup over the first 10% of the dataset. We incorporate gradient accumulation and gradient clipping for stability.

Figure 1 illustrates the training loss trajectory for our SFT-fine-tuned `Qwen-2.5-0.5B` model across approximately 55,000 optimization steps. The loss curve exhibits a clear downward trend during the initial phase of training, with the loss decreasing from above 2.0 to values consistently below 1.0. After the initial rapid decline, the loss continues to decrease gradually, reaching a relatively stable regime with moderate variance—indicative of healthy model convergence and ongoing learning.

We evaluated the fine-tuned model on the leaderboard where it achieved a win rate of 0.05 (5%) over the baseline SFT model. This result demonstrates that the SFT process enables the model to capture additional signal from the instruction-following demonstrations, albeit with relatively modest gains in this challenging setting. We believe that this would provide enough reward signal for DPO to then improve upon for future iterations.

2.2 DPO

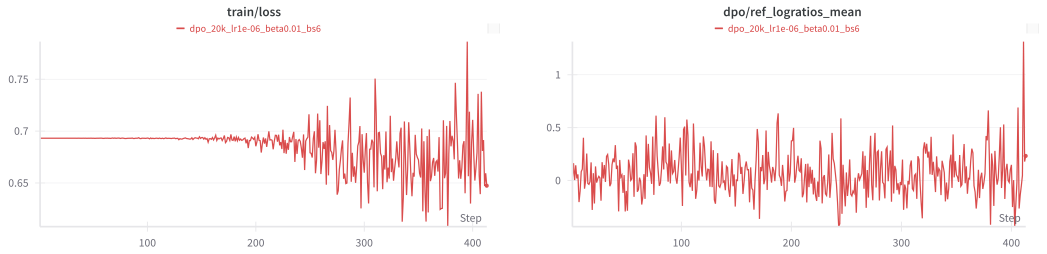
To further align our `Qwen-2.5-0.5B` language model with human preferences, we employed DPO using a synthetically generated ultrafeedback dataset. The DPO framework operates on paired preference data: for each user prompt, the model is shown a preferred ("chosen") and a less preferred ("rejected") response. The training objective encourages the policy model to assign higher log-likelihood to the preferred response relative to the rejected one, compared to a static reference model. This is operationalized through a pairwise loss:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right] \tag{1}$$

where y_w and y_l denote the preferred (winner) and less preferred (loser) completions respectively, π_θ is the policy model, π_{ref} is the reference model, and β is a temperature-like scaling parameter. In our experiments, we set $\beta = 0.05$ to balance stability and preference strength.

The policy and reference models are initialized from an SFT-finetuned `Qwen-2.5-0.5B` checkpoint and are trained with the AdamW optimizer (learning rate 2×10^{-7}), weight decay 0.01, $\epsilon = 1 \times 10^{-7}$. Training is conducted for a single epoch on $20,000$ preference pairs, using a batch size of 6, maximum sequence length of 512, and right-side padding. Gradient accumulation (4 steps) is used to simulate larger effective batch sizes, and gradient norms are clipped to 1.0 to ensure stability. Throughout training, the reference model remains frozen, serving as a baseline for evaluating the improvement in the policy's preference alignment.

Despite extensive experimentation, surpassing the 0.1 leaderboard threshold with DPO on the Ultrafeedback benchmark proved highly challenging, with our best model achieving only a 0.0975 win rate. We conducted broad hyperparameter sweeps—including β values from 0.01 to 0.2, learning rates between 1×10^{-7} and 1×10^{-6}, and various dataset sizes up to 61,000 preference pairs—but DPO consistently fell short. Even with gradient accumulation and clipping, training was unstable, with noisy loss curves and large gradient spikes. These results highlight both the difficulty and sensitivity of preference-based optimization in this setting, as well as the limitations of`Qwen-2.5-0.5B` for challenging human feedback tasks.



(a) DPO Loss (b) DPO Reference Log-Ratio

Figure 2: Training dynamics for DPO: (a) Loss curve; (b) Mean log-ratio between policy and reference models.

Figure 2a presents the training loss curve for DPO fine-tuning under a representative hyperparameter configuration ($\beta = 0.01$, learning rate 1×10^{-6}, batch size 6, $20,000$ preference pairs. The loss remains relatively flat and stable for the majority of training, indicating that the model is able to maintain consistent preference classification performance. In the latter part of training, the loss exhibits increased volatility, which is characteristic of DPO's pairwise objective and the inherent stochasticity of human preference datasets. This heightened noise is expected, as the DPO loss directly reflects fluctuations in preference margin across minibatches, rather than a smooth likelihood surface. Importantly, despite the increased variance, the overall loss does not show signs of instability or divergence, suggesting the training remains under control.

Figure 2b shows the evolution of the mean log-ratio between the policy model and the reference model ($\log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$) during DPO training. This metric tracks the average divergence in likelihood assignments to preferred completions across training steps. For stable and effective DPO training, we expect this curve to fluctuate around zero, indicating that the updated policy does not deviate excessively from the reference while still learning to prefer higher-quality responses.

In this run (DPO, 20k samples, $\text{lr} = 1\text{e}{-6}$, $\beta = 0.01$), the log-ratio mean exhibits considerable short-term variance but remains mostly centered near zero throughout training. This behavior suggests that the policy is making small, controlled updates relative to the reference, without any runaway divergence—a desirable property, especially with a small KL scaling factor ($\beta = 0.01$). Occasional spikes are expected due to the stochastic nature of batch sampling and the discrete nature of preference data, but there is no sustained drift or instability.

Overall, the curve indicates healthy and regularized policy optimization, consistent with a DPO regime that preserves alignment to the base model while incrementally shifting probability mass toward preferred completions.

3 Method

3.1 XPO: eXploratory Preference Optimization

The first extension we considered was Exploratory Preference Optimization (XPO) Xie et al. (2024). XPO is a variant of DPO that explicitly encourages the model to explore generations that are less probable through the addition of a positive term in the loss function that indicates the model's likelihood on a given example. A higher value for this likelihood term indicates the model is more confident about its generation and therefore cause the overall score to be higher. XPO addresses a fundamental limitation of existing alignment methods, particularly Online DPO, which are constrained to behaviors well-supported by the initial model π_{ref} and preference data D_{pref}. The key insight is that RLHF methods need to move beyond this limitation by collecting feedback from responses sampled from the model during training, but existing approaches use only passive exploration. The algorithm proceeds similarly to Online DPO, but at each iteration t, instead of sampling responses $r^{(t)} \sim \pi^{(t)}$ and $\tilde{r}^{(t)} \sim \pi_{\text{ref}}$, XPO samples both responses from the current policy and labels them based on preference feedback.

The XPO objective modifies the standard DPO loss by incorporating a global optimism term. At each iteration, the policy is updated via:

$$\pi^{(t+1)} \leftarrow \arg\min_{\pi \in \Pi} \left\{ \alpha \sum_{i=1}^{t} \log \pi(\tilde{r}^{(i)}) - \sum_{(\tau_+, \tau_-) \in D_{\text{pref}}^{(t)}} \log \left[\sigma \left(\beta \log \frac{\pi(\tau_+)}{\pi_{\text{ref}}(\tau_+)} - \beta \log \frac{\pi(\tau_-)}{\pi_{\text{ref}}(\tau_-)} \right) \right] \right\}$$

where $\alpha \geq 0$ is the optimism parameter, and the first term $\alpha \sum_{i=1}^{t} \log \pi(\tilde{r}^{(i)})$ serves as the exploration bonus that encourages optimistic behavior. For $\alpha = 0$, the algorithm reduces to Online DPO (except for the sampling strategy). When $\alpha > 0$, the term: $\alpha \sum_{i=1}^{t} \log \pi(\tilde{r}^{(i)})$ encourages the policy to behave optimistically and produce diverse responses r.

Given the theoretical justification of XPO in more classical RL settings such as multi-armed bandits and the relative lack of experimental results on XPO for more general language modeling tasks, we concluded that XPO would be a worthwhile extension to pursue. Another reason we found this

algorithm appealing is that it requires only a one-line modification to existing DPO implementations while providing principled exploration that can discover behaviors not well-covered by the initial reference policy.

3.2 IPO: Identity Preference Optimization

The second extension we considered was Identity Preference Optimization (IPO) Gheshlaghi Azar et al. (2024). IPO addresses a fundamental limitation of DPO: its tendency to overfit to preference data, particularly when preferences are deterministic. This overfitting stems from the unboundedness of the preference function Ψ in DPO, combined with the lack of explicit reward function training. Motivation and Problem Statement. Standard DPO is prone to overfitting because it can assign arbitrarily high importance to preference probabilities that are already close to 1, which may be undesirable when preference probabilities are around 50%. The maximization of logit-preferences (Elo scores) can have counter-intuitive effects, even in transitive settings. Additionally, in the finite-data regime where we only have access to sample estimates of preferences $\hat{p}(y \succ y')$, empirical overfitting becomes a substantial issue, especially when context and action spaces are extremely large, as is typical for large language models.

IPO addresses these issues by choosing the identity mapping as the preference function: $\Psi(q) = q$. This choice ensures that the KL regularization in the objective remains effective even when preferences are deterministic, leading to more robust optimization. Similar to XPO, an appealing aspect of IPO is that rather than using the complex theoretical derivation from the paper, IPO can be implemented with a simple squared loss formulation in code. The IPO loss is defined as: $\mathcal{L}_{\text{IPO}} = \mathbb{E}(x, y_w, y_l)\left[\left(\log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)} - \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)} - \frac{1}{2\tau}\right)^2\right]$ where $\tau > 0$ is a hyperparameter that controls the regularization strength. This formulation can be rewritten more clearly as: $\mathcal{L}_{\text{IPO}} = \mathbb{E}(x, y_w, y_l)\left[\left(\Delta_{\text{policy}} - \Delta_{\text{ref}} - \frac{1}{2\tau}\right)^2\right]$ where: $\Delta_{\text{policy}} = \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$ is the log-ratio of chosen to rejected responses under the policy $\Delta_{\text{ref}} = \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)}$ is the log-ratio under the reference model

The constant term $\frac{1}{2\tau}$ acts as a target that the policy difference should achieve relative to the reference model. This squared loss formulation provides several advantages over the DPO objective: it is bounded, prevents overfitting to deterministic preferences, and maintains effective regularization throughout training. Similar to XPO, IPO also has theoretical guarantees including the uniqueness of a solution. Under the assumption that $\text{Supp}(\mu) = \text{Supp}(\pi_{\text{ref}})$, the mapping $\pi \mapsto \mathcal{L}(\pi)$ has a unique global/local minimum, which is π^*. The objective is quadratic as a function of the logits, making it amenable to efficient optimization through standard techniques. Unlike DPO, IPO maintains effective regularization even when preferences are deterministic, preventing the pathological overfitting behaviors observed in standard DPO. This also makes IPO in theory more robust to over-training compared to DPO and RLHF.

3.3 Synthetic Data Augmentation

In `Ultrafeedback-binarized`, preferred and dispreferred responses are selected based on a composite score by `GPT-4` across 4 metrics: helpfulness, honesty, truthfulness, and instruction-following. However, these aren't necessarily the same set of criteria used by the `nvidia/Llama-3.1-Nemotron-70B-Reward` reward model; these criteria are Chat Quality (measured by fluency, relevance and coherence) in Basic Interactions, Adversarial Scenarios, and Multi-turn Consistency. We therefore theorize that creating preferred and dispreferred responses that are mindful of these reward model objectives will lead to a more well-aligned model for our task and an improved leaderboard score. Specifically, we prompted `gpt-4.1-mini` to produce 2500 examples (to ensure API costs were reasonable) with prompts in the theme of Ultrafeedback and preferred and dispreferred responses based on the criteria of the reward model. We then performed DPO on a 2500 example subset of Ultrafeedback for a fair point of comparison.

3.4 Extra Credit: Countdown and RLOO

For extra credit, we explored the Countdown dataset by supervised fine-tuning `Qwen-2.5-0.5B` on the CogBehave "all strategies" split, which features diverse math reasoning problems and human-annotated solutions. Data was preprocessed into model-ready input-output pairs with consistent

tokenization and right padding. We trained with AdamW (3×10^{-5} learning rate), batch size 8, gradient accumulation (8 steps), sequence length 512, and a cosine LR schedule with 10% warmup. Training lasted one epoch with gradient clipping for stability. The resulting model provides a strong baseline for mathematical reasoning and serves as a foundation for subsequent preference optimization and RL experiments on Countdown.

We fine-tuned `Qwen-2.5-0.5B` models using the Reward-weighted Regression Leave-One-Out (RLOO) algorithm on the Countdown dataset. As noted in the paper, RLOO is a policy gradient estimator based on REINFORCE with the following objective:

$$\frac{1}{k}\sum_{i=1}^{k}\left[R(y_{(i)},x)-\frac{1}{k-1}\sum_{j\neq i}R(y_{(j)},x)\right]\nabla\log\pi(y_{(i)}\mid x)\quad\text{for } y_{(1)},\ldots,y_{(k)}\overset{i.i.d.}{\sim}\pi_\theta(\cdot\mid x) \tag{2}$$

For each prompt, the model generated multiple completions, which were scored by a custom reward function emphasizing mathematical correctness and use of provided numbers. Policy updates maximized the advantage-weighted log probability of completions, with a KL penalty to regularize against the initial reference model. We logged training metrics and saved checkpoints for subsequent evaluation.

We conducted a targeted hyperparameter sweep varying four factors: learning rate (1×10^{-5}, 2×10^{-5}, 3×10^{-5}) , KL regularization strength (0.01, 0.02, 0.05), batch size (16, 20, 24), and number of samples per prompt (3 or 4). From the grid, 10 diverse combinations were selected and run on 1,000 Countdown examples each, recording performance metrics for comparative analysis.

4 Experimental Setup

We first evaluate DPO, where models are fine-tuned using synthetic ultrafeedback preference pairs. In this setup, each training sample presents a user prompt and two responses (one preferred, one rejected), and the model is optimized to increase the likelihood of preferred responses as judged by automated reward models. Our DPO experiments use SFT-finetuned `Qwen-2.5-0.5B` as initialization and compare results with supervised fine-tuning. As extensions, we investigate Exploratory Preference Optimization (XPO), Implicit Preference Optimization (IPO), and additional SFT using synthetic data generated by `gpt-4.1-mini`, as well as parameter-efficient LoRA Hu et al. (2022) fine-tuning.

We further assess instruction-following and mathematical reasoning on the Countdown benchmark, where models must generate valid arithmetic expressions using a set of provided numbers to reach a target value. For these experiments, we use the "all strategies" split from CogBehave for supervised fine-tuning, and $1{,}000$ prompts from Countdown for reinforcement learning via RLOO. As baselines, we report SFT, DPO, and RLOO. Evaluation is based on win-rate, the proportion of prompts where the candidate model's output receives a higher reward (using a format- and correctness-aware verifier) than a reference model. All experiments use right-padding, fp32 precision, and standardized chat-style prompting for reproducibility.

We designed a custom reward model for RLOO. The reward model for the Countdown dataset evaluates completions by first checking for invalid HTML artifacts and penalizing those outputs. It then rewards outputs that avoid phrases, correctly use arithmetic operators, and utilize the required numbers from the prompt. Additional reward is given if all provided numbers are used exactly once and if the format matches expected answer tags. The reward further increases as the generated expression's numerical result approaches the target value, with a maximum reward for perfect solutions and scaled partial credit for near-misses. For other datasets, a simpler reward model scores outputs based on their length and penalizes excessive repetition or overly short completions.

5 Results

5.1 XPO: eXploratory Preference Optimization

As XPO is a variant of DPO, the main hyperparameter to control is α where higher α value places more emphasis on high likelihood generations in the loss term, therefore providing the model an

incentive to "explore" generations it is less confident about. For both DPO and XPO, we find that generation quality becomes progressively more incoherent as we scale the number of examples to 61k examples. As DPO progresses for example, we see more repetitions, incoherent generations, and symbols from different human and programming languages appear in the generations. Therefore we run DPO and XPO with 5000 examples to provide a fair comparison. We use the following hyperparameters after doing a hyperparameter sweep with 16 experiments. Specifically we test $\beta \in (0.05, 0.1)$, $lr \in (1e-7, 2e-7, 5e-7, 1e-6)$; for XPO we test $\alpha \in (0.1, 0.01)$ and for IPO we test $\tau \in (0.01, 0.1)$ following the literature.

Table 1: XPO Fine-tuning Parameters

Parameter	**Value**
Algorithm/Dataset	XPO on UltraFeedback (5K examples)
Base Model	QWEN2.5-0.5B-SmolTalk
Learning Rate/Beta/Alpha	2×10^{-7} / 1.0 / 0.01 and 0.1
Batch Size/Grad. Accum.	6 / 4
Max Length/Epochs	512 / 1
Optimizer	AdamW (decay=0.01, $\epsilon = 10^{-7}$)

Figure 3: Results of XPO run with $\alpha = 0.1$

Figure 4: Results of XPO run with $\alpha = 0.01$

We also run a DPO experiment as a baseline seen in figure 5 with 5000 examples and similar hyperparameters except with a batch size of 8.

Starting with $\alpha = 0.01$, we can see that the XPO loss is mostly stable around 0.69 with some spikes down to 0.67. This is to be expected as our DPO loss did not deviate much from 0.69 as well. Looking at the DPO loss formulation for a moment, observe that the inner expression being 0 represents the model not having preference between preferred and dispreferred responses (i.e. when the difference between the preferred and dispreferred quotients is 0). Concretely, we have

$$L_{DPO} = -log(\sigma(0)) = -log(0.5) = log(2) = 0.693$$

Figure 5: DPO baseline trained on 5000 examples

Analyzing each of the plots, we notice that for both values of α the gradient norms are always set to 1. This proved to be necessary to achieve a somewhat stable loss curve, however it indicates that the DPO setup has tendencies to diverge which should be rectified in an alternate manner ideally. Looking at the the exploration term plots, we do see that the contribution of the exploration term does increase by an order of magnitude for $\alpha = 0.1$ which is to be expected given α is 10x larger. Assuming a more stable baseline DPO algorithm, perhaps XPO could have achieved higher performance. On the leaderboard, the better performing model got 0.0350, whereas DPO with 5000 examples got 0.0275; this difference appears to be within the margin of error. On the qualitative side of things, we notice that XPO's theoretical benefit of performing better on hard coding and math problems does not manifest in generations. Specifically, on a prompt such as line 8 in the Ultrafeedback heldout JSON file where the model has to reason about the profit for a bookstore, the SFT model is able to arrive at the right answer whereas the DPO and XPO model are not due to a complete breakdown in mathematical reasoning. These results of XPO, combined with the unfruitful hyperparameter sweep of XPO, led us to consider a simpler DPO formulation.

5.2 IPO: Identity Preference Optimization

Given the lack of success with XPO, we turned to exploring Identity Preference Optimization. Looking at the rewritten formulation carefully, $L_{\text{IPO}}(\pi) = \mathbb{E}_{(y_w,y_l)\sim D}\left[\left(h_\pi(y_w, y_l) - \frac{1}{2\tau}\right)^2\right]$ where $h_\pi(y_w, y_l) = \log\left(\frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right) - \log\left(\frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)$, we see that there is no sigmoid term as in DPO and the only nonlinear terms are quadratic. This led us to believe that in principle IPO may be a more straightforward objective to optimize. After performing an extensive hyperparameter sweep, we settled on the following hyperparameter configuration for IPO.

Table 2: IPO Fine-tuning Parameters

Parameter	**Value**
Algorithm/Dataset	IPO on UltraFeedback (5K examples)
Base Model	QWEN2.5-0.5B-SmolTalk
Learning Rate/Tau	2×10^{-7} / 0.01
Batch Size/Grad. Accum.	8 / 4 (effective: 32)
Max Length/Epochs	512 / 1
Optimizer	AdamW (decay=0.01, $\epsilon = 10^{-7}$)

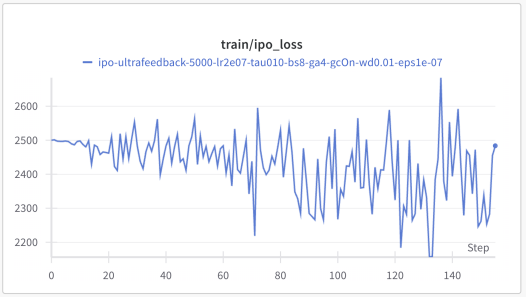


Figure 6: IPO Training Loss

Observing the loss curve, we notice that loss starts at 2500. Concretely this is because the loss at the beginning is only a function of the tau value as the values of the other terms are 0 at the beginning of training. This leads to the expression $\frac{1}{4\tau^2} = \frac{1}{4*(0.01)^2} = 2500$. We see that training loss does not indicate a downward trend here either, and the leaderboard score of 0.0325 reflects the lack of performance. We also saw similar problems in the generations of the IPO model's responses especially on more mathematically oriented tasks. At this point, we considered whether we were getting quality reward signal from the `ultrafeedback-binarized` dataset given that the Nemotron model criteria differs from the criteria used to assess preferred and dispreferred pairs in Ultrafeedback.

5.3 Synthetic Data Augmentation

We consider a setup where we use vanilla DPO on 2500 examples of a synthetic dataset. Specifically, we prompt `gpt-4.1-mini` to generate 2500 prompts similar to the prompts present in the Ultrafeedback dataset and also generate preferred and dispreferred responses also using `gpt-4.1-mini` and the same criteria that `nvidia/Llama-3.1-Nemotron-70B-Reward` uses to grade outputs. For preferred responses we allow for a maximum of 800 tokens to be generated and use a temperature of 0.7 and for dispreferred responses we allow a maximum of 400 tokens to be generated and use a temperature of 0.9, roughly approximating the dataset statistics in the original Ultrafeedback dataset.

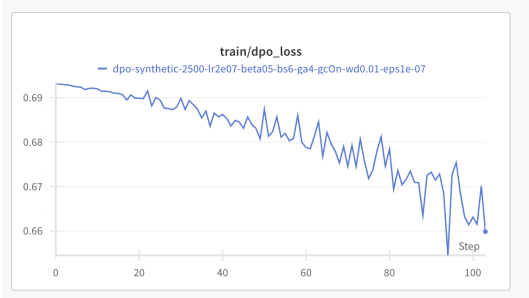


Figure 7: Training loss curve using a synthetic dataset of 2500 examples

For the first time, we observe that the loss decreases in a stable manner. Observing the examples, the more mathematical questions appeared to be more correct compared to XPO and DPO. However, the leaderboard score of 0.025 indicates that the model's generations were not preferred over the reference model. We believe that our prompt for assessing the traits that Nemotron used may have been underspecified and therefore preferred generations may not have embodied enough of these traits.

5.4 Extra Credit

Figure 8: Training loss curve for SFT (Supervised Fine-Tuning) of `Qwen-2.5-0.5B` on the Countdown dataset.

Figure 8 presents the training loss trajectory for supervised fine-tuning (SFT) of the `Qwen-2.5-0.5B` model on the Countdown dataset. The training process employs a batch size of 8 with gradient accumulation over 8 steps and utilizes a learning rate of 3×10^{-5}, with optimization performed using AdamW and cosine learning rate scheduling.

The loss curve exhibits a rapid initial decrease, dropping from approximately 1.25 to below 0.9 within the first few steps. This steep decline is indicative of the model's ability to quickly fit the training data, leveraging strong initialization from the pre-trained `Qwen-2.5-0.5B` checkpoint.

Despite the model learning well with SFT, we noticed that our output responses had average formatting and very poor accuracy (best leaderboard score of 0.12). For our primary extension, we adopted the LoRA technique based on TA suggestions to enable parameter-efficient fine-tuning of large language models on specialized datasets. Rather than updating all model parameters during supervised fine-tuning (SFT), LoRA injects lightweight trainable adapter matrices into the attention projection layers, enabling effective adaptation with a fraction of the training cost and memory footprint.

Specifically, we used the PEFT (Parameter-Efficient Fine-Tuning) library to configure LoRA adapters for the `Qwen-2.5-0.5B` base model. The LoraConfig was set with a rank 16, scaling factor $\alpha = 32$ and dropout rate of 0.1. We targeted all key attention projections (Q, K, V, O) as LoRA modules. During training, only the LoRA adapter weights were updated, while the base model weights were frozen, which drastically reduced the number of trainable parameters and improved training efficiency.

With this change, we observe similar loss metrics but a much greater performance on leaderboard with 0.18 or nearly $96/1000$ accuracy score. Given the limited size of the Countdown dataset, LoRA's parameter-efficient fine-tuning paradigm provides a natural regularization effect, which prevents overfitting and enables better generalization than full SFT.

6 Discussion

Despite substantial effort and an exhaustive hyperparameter sweep, achieving a DPO score surpassing the 0.1 baseline proved difficult. Our experiments explored a broad range of hyperparameters, including β for KL regularization, learning rates, batch sizes, and dataset sizes, gradient accumulation steps, padding, various chat templates, system prompts, masking strategies and more. However, all of our configurations were below the threshold. This substantial computational cost and diminishing returns highlight the sensitivity of DPO to hyperparameter selection especially in the context of ultrafeedback-style datasets. The difficulty in surpassing the DPO baseline had downstream consequences: many of our planned extensions, such as exploratory preference optimization (XPO), were built on top of DPO and thus suffered from poor baseline performance, limiting their potential for improvement. Additional limitations include the heavy compute required for extensive sweeps, potential overfitting in low-data regimes, and the challenge of reliably transferring improvements from leaderboard metrics to real-world performance.

7 Conclusion

In this work, we benchmarked a range of alignment strategies—including SFT, DPO, RLOO, LoRA, and recent extensions such as XPO, IPO, and synthetic data augmentation—on instruction-following and mathematical reasoning tasks. Our results demonstrated that preference alignment with DPO is notably challenging, even with exhaustive hyperparameter tuning due to the quality of data present and the limitations of a 0.5B model. We also showed that finetuning using while parameter-efficient methods like LoRA can be more effective in data-constrained settings. We implemented RLOO with a basic reward function that penalized attributes such as irrelevant English phrases and rewarded the use of valid numbers and found that it did not beat the best SFT baseline of 0.18, achieving only 0.12. These difficulties extend to downstream extensions, suggesting that stronger DPO baselines are critical for further progress. Moving forward, our findings highlight the need for improved optimization techniques, better reward modeling, and more diverse evaluation frameworks to advance LLM alignment in complex problem domains.

8 Team Contributions

- **Group Member 1:** Parth Shroff. Parth primiarly focused on developing SFT and DPO baselines for SmolTalk / Ultrafeedback and the extra credit work of implementing SFT and RLOO on countdown.
- **Group Member 2:** Pranshu Chaturvedi Developed fuctionality in SFT and DPO to support utilities like Weights and Biases, checkpointing, data preprocessing etc. He also implemented DPO and XPO.

Changes from Proposal Initially, our project proposal focused on developing a custom project centered on optimizing and fine-tuning state-of-the-art large language models (LLMs) for challenging domains such as mathematical reasoning and program synthesis. However, after discussing our approach with our TA mentor (Anikait), we decided to pivot toward the default project framework, which offers well-established infrastructure for benchmarking preference optimization algorithms. Anikait had suggested that XPO presents a particularly promising research direction for advancing LLM performance on complex math and coding tasks. By adopting the default project as a foundation, we were able to systematically study baseline techniques such as SFT and DPO, while also

implementing XPO as a natural extension to explore its effectiveness in settings that demand both accuracy and diversity of reasoning.

References

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 *International Conference on Machine Learning (ICML)*. https://doi.org/10.48550/arXiv.2310.01377 arXiv:2310.01377.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Zhaohan Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2024. A General Theoretical Paradigm to Understand Learning from Human Preferences. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 238)*. PMLR, 1–25. https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.2106.09685 arXiv:2106.09685.

Archit Sharma, Eric Mitchell, Albert Xu, Yian Zhang, Stefano Ermon, Chelsea Finn, Christopher D. Manning, Jure Leskovec, and Tatsunori Hashimoto. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2305.18290 arXiv:2305.18290.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. 2024. Exploratory Preference Optimization: Harnessing Implicit Q*-Approximation for Sample-Efficient RLHF. *arXiv preprint arXiv:2405.21046* (2024). arXiv:2405.21046 https://arxiv.org/abs/2405.21046