

SCALABLE PREFERENCE LEARNING FOR LARGE LANGUAGE MODELS VIA CONVEX OPTIMIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Fine-tuning large language models (LLMs) for alignment with human preferences have become a key factor in the success of models like ChatGPT and Gemini, which are now integral to mainstream use. Many effective techniques are based on Reinforcement Learning from Human Feedback (RLHF), yet are complex, unstable, and expensive to implement. Recently, Direct Preference Optimization (DPO) offers an accessible alternative by simplifying the objective and training a policy model using a frozen, copied reference model to provide a stable training benchmark. In this paper, we develop an even more lightweight DPO based algorithm that operates on a single GPU. The key to achieving this is leveraging the convex optimization reformulation of neural networks, and reducing the dependence on copying the reference model. Our aim is to provide faster convergence to solutions of better optimality, and higher interpretability of the underlying optimization landscape for generative language tasks. We use the Alternating Direction Method of Multipliers (ADMM) to solve this optimization problem in order to increase parallelization efficiency, and implement our methods in JAX to lift the memory constraints across experiments. We experiment on three datasets, including one synthetically generated educational dataset, to demonstrate the efficacy of our novel algorithm in a real world setting. Our method is comparable in user preference generation to DPO when tested on 17 human volunteers, despite being trained on one single RTX-4090 GPU using a smaller dataset.

1 INTRODUCTION

Language models have been trained on increasingly large amounts of data to capture semantic language patterns. The current paradigm is a combination of pre-training and fine-tuning these LMs to achieve more preferable responses. The DPO (Rafailov et al., 2024) paper proposes a simpler, performant, and computationally lightweight alternative to aligning LMs to given instructions and optimizing for human preferences. This model is able to infer what a human user wants and output a realistic answer that a human might like, thus taking huge steps towards inferring intent and demonstrating remarkable generalization. RLHF (Wang et al., 2023) is a performant yet complex method which involves a 3-step training cycle between humans, the agent’s understanding of the goal, and the Reinforcement Learning (RL) training procedure. However despite its effectiveness, RLHF is expensive, complicated, sensitive to numerous hyperparameters, and subject to unstable training procedure. The additional dependency on humans in the training loop is also unsatisfactory. This restricts RLHF to only high resource companies, leading to centralized and closed AI power which is a detriment to progress. However, both of these methods typically still require multi-GPU settings to train on meaningful real world datasets.

In this project, we aim to provide a novel lightweight framework to preference fine-tune small language models. We use only one RTX-4090, and leverage a convex reformulated NN as a surrogate for the traditional DPO loss. Since the convex model is hyperparameter free and of negligible cost to train, this addition is able to provide valuable auxiliary signal information on the optimization landscape without the need for copying a static reference model as in traditional DPO. We utilize the JAX (Schoenholz et al., 2018) framework to Just-In-Time compile our lower level functional code efficiently, thus fitting the entire preference alignment pipeline on one consumer grade GPU. In order to assess the efficacy of our method, we create a synthetic Educational-Tutor conversational dataset, and evaluate on 25 human volunteers via survey to assess performance.

The convex reformulation of NN for binary classification problems has been explored in previous work Pilanci & Ergen (2020). We leverage the fact that DPO treats the policy optimization task as a binary classification problem. In order to make the convex formulations scalable, we apply ADMM with block coordinate descent. Furthermore, we implement our algorithms in JAX, to optimize for computational cost and efficient memory utilization. As a result, all of the experiments in this work were performed on one single RTX-4090 GPU, with reasonable train times of less than 1 hour (typically within minutes), mostly across two datasets. Our objective is to ultimately provide an efficient way of aligning LLMs to human preference that is more accessible in academic settings. We believe this constitutes a step towards democratizing AI systems for the wider audiences, as well as improving optimization techniques in this area.

Our main contribution is the novel DPO-Convex algorithm using JAX which trains on one GPU. We also provide a custom Educational-Tutor dataset which simulates 4000 conversations between a student studying for a quiz across various subjects. We then construct the DPO style "chosen" versus "reject" preference dataset with a natural alternating scheme in the conversation. This avoids the excessive need to prompt a third party LLM to generate the "chosen" or "rejected" feedback and makes the method even more accessible for experimentation. In order to be consistent with prior work in the original DPO paper, we conduct human evaluation with 25 volunteers. Our key desiderata can be summarized as follows:

- Build a robust yet lightweight DPO algorithm that trains small LLMs on one GPU with results comparable to the full DPO training pipeline.
- Incorporate the Convex-NN to achieve this. Thus providing valuable auxiliary feedback, and faster convergence to solutions of near-optimal loss, with easier parallelization.
- Solve this problem with ADMM and implement our methods in JAX. This eliminates the need for data parallelism via Pytorch (Zhao et al., 2023), and also provides the advantage of more efficiently managing memory, lifting the previous size constraints from the existing Convex-NN solvers.
- Introduce a novel alternating strategy for generating the preference alignment dataset, which makes the fine-tuning pipeline even more accessible to individual persons for experimentation and learning.

2 RELATED WORK

Building LLMs that better align to human preferences can be approached through three distinct strategies. Initial algorithms of zero-shot and few-shot in-context learning (Xian et al., 2017) relies on prompt engineering. Although this method is able to improve the performance of LLMs to produce desired outputs and does not require fine-tuning, it is not able to tackle complex tasks. More sophisticated learning methods use RL to align model outputs with user preferences. The most successful classes (such as RLHF and RLAIF (Lee et al., 2023)) have been able to create conversational LLMs such as ChatGPT. However despite their impressive performance, these methods are extremely complex, requires humans in the loop, and requires significant computational resources. Therefore the authors of DPO developed a simple yet performant learning algorithm to directly optimize to human preferences, without explicit reward modeling. The official implementation of DPO references four 80GB A100s, which reduces the barrier to training LLMs. Nevertheless, multi-GPU systems are still out of reach for many researchers in academia. In addition, DPO introduces certain hyperparameters that need careful tuning.

Bengio et al. (2005) have previously shown that it is possible to characterize the optimization problem for neural networks as a convex program. Pilanci & Ergen (2020) further developed exact convex reformulations of training a two-layer ReLU neural network. The core of this representation lies in semi-infinite duality theory, and was derived in Pilanci (2020) to show that two-layer neural networks with ReLU activations and weight decay regularization may be re-expressed as a linear model with a group one penalty and polyhedral cone constraints. This is a step towards achieving globally optimal networks and interpretable results. This yields both practical benefits in implementation, and theoretical advantages in analyzing the optimization of the non-convex landscape of NNs. This framework is most efficient on two-layer NNs, and on small scale datasets such as

CIFAR-10 or MNIST (Mu & Gilmer (2019)). In order to apply this method to the area of LLMs where large data is paramount, we seek better solutions for scalability.

To practically solve this convex optimization problem, Bai et al. (2018) have proposed an approach based on the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). ADMM offers several attractive advantages, such as its robustness against hyperparameter selection, linear decomposability for distributed optimization, and immunity to vanishing/exploding gradients. The successful application of ADMM in solving optimization problems across a wide range of domains has been well studied. This includes diverse fields such as control theory (Li et al. (2017)), maximum a posteriori (MAP) inference problems (Lu & Lü (2019)), computational biology and finance (Costa & Kwon (2020)). The natural parallelization aspects of ADMM seem to make it particularly suitable to deep learning problems. Therefore we aim to integrate the convex reformulations of NNs with DPO. Our goal is to have the convex model provide clear signals to the DPO loss, thus leveraging the faster convergence to obtain LLMs of better quality.

3 DPO AND CONVEX NEURAL NETWORKS

Convex-NNs. The work of Pilanci & Ergen (2020) introduces convex duality theory for non-convex neural network objectives. These results offer a characterization of NN models by using convex regularization in a higher dimensional space, where the data matrix is divided according to all possible hyperplane arrangements. Therefore we can define the equivalent convex reformulation of a two-layer MLP as

$$f_{\text{CVX-MLP}}(x) = \sum_{i=1}^{P_s} x^T (u_j - v_j) \mathbb{I}[x^T h_i \geq 0], \quad (1)$$

where P_s activation patterns are sampled randomly via fixed and randomly generated gate vectors $\{h_i\}_{i=1}^{P_s}$. We refer to Kim & Pilanci (2024) for a theoretical analysis guaranteeing near-optimality for random Gaussian gates. We aim to leverage the rich representation power of this network via their universal approximation property.

The standard non-convex formulation of a two-layer ReLU-MLP with weights $w_j^{(1)}, w_j^{(2)}$ is:

$$f_{\text{NCVX-MLP}}(x) = \sum_{j=1}^m \left(x^T w_j^{(1)} \right)_+ w_j^{(2)} \quad (2)$$

$X \in \mathbb{R}^{n \times d}$ is a two-layer ReLU-MLP, usually trained in a stochastic setting. Scalable in large datasets, but sensitive to hyperparameter tuning and lacks optimality guarantees.

From a high level, Lagrangian zero duality gap proves that there does exist a convex program which achieves the same optimal value as the non-convex problem. Therefore by leveraging this perspective, we reach better interpretability and understanding of the optimization landscape of neural networks. Pilanci & Ergen (2020) provide the theoretical proof and analysis of the convex reformulation. The convex approach is more efficient and leads to solutions that generalize well. By applying the mechanics of convex optimization to DPO, we aim to achieve better robustness, interpretability, and faster convergence to a globally optimal solution.

ADMM for Parallelism. The Alternating Direction Method of Multipliers (ADMM) solves convex optimization problems by breaking them into smaller subproblems, each of which are then easier to handle. The seminal work of Boyd et al. (2011) has demonstrated the versatility of this algorithm on a wide application of problems.

The recent work of Bai et al. (2018) applied ADMM to solving this problem by introducing slack variables to arrive at a convex reformulation with mean squared error loss:

$$\min_{v, s, u} \|Fu - y\|_2^2 + \beta \|v\|_{2,1} + \mathbb{I}_{\geq 0}(s) \quad \text{s.t.} \quad u = v, Gu = s \quad (3)$$

Matrix $F \in \mathbb{R}^{n \times 2dP_s}$ in this formulation is block-wise constructed by $D_i X$ terms. This optimization problem yields the (simplified) ADMM updates explicitly:

$$\text{Primal } \mathbf{u} \text{ update : } Au^{k+1} = b \quad \text{for } A = I + \frac{1}{\rho} F^T F + G^T G \quad (4a)$$

$$\text{Primal } \mathbf{v} \text{ update : } v^{k+1} = \text{prox}_{\frac{\beta}{\rho} \|\cdot\|_2}(u^{k+1} + \lambda^k) \quad (4b)$$

$$\text{Dual update : } \lambda^{k+1} = \lambda^k + \gamma_\alpha(u^{k+1} - v^{k+1}) \quad (4c)$$

This is setting we can address the BCE loss exhibited in DPO. In order to avoid the expensive Cholesky decomposition to solve the primal update step, we apply conjugate gradient to take advantage its potential for acceleration and scalability. The main attraction of ADMM in this work is its decomposition into subproblems which can be solved independently and in parallel. This is particularly advantageous for large-scale distributed or high-dimensional problems, and is particularly relevant since the the importance of GPUs in language modeling cannot be overstated.

Additionally, ADMM offers satisfying analysis of convergence guarantees under mild assumptions, which is particularly desirable when working with the non-convex landscape of language models. This method is extremely robust to tuning of hyperparameters, while offering a more transparent understanding of the underlying landscape of optimization.

JAX for Speed and Memory. JAX is a lower level numerical framework that offers speed and memory advantages. Since JAX was developed for high-performance machine learning research, our past work has found it to be extremely performant in GPU acceleration settings.

DPO has brought down the barrier to entry significantly for alignment and instruction fine-tuning compared to RLHF. However, naively implementing the DPO training pipeline is still computationally significant. For reference, the DPO official implementation by the authors at dpo, still requires a cluster of A100 GPUs to overcome the memory bottlenecks in deep learning. Therefore in order to implement this work within the compute resources of one GPU (section 4.2), we take advantage of JAX’s Just-In-Time (JIT) feature and Accelerated Linear Algebra (XLA) compiler. Recent research in review will provide more in depth discussion on the lower level optimizations of JAX. Parallelism is an additional attractive feature, since JAX supports easy parallelism over multiple devices, such as multiple GPUs or TPUs. The introduction of PMAP also eliminates the need for complex FSDP logic functions, while allowing for more fine-grained control over computational resources with the possibility to execute on TPUs effectively. The DPO loss is:

$$L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \frac{\log \pi_\theta(y_w|x)}{\log \pi_{\text{ref}}(y_w|x)} - \beta \frac{\log \pi_\theta(y_l|x)}{\log \pi_{\text{ref}}(y_l|x)} \right) \right]. \quad (5)$$

In order to eliminate the need for copying the reference model as a fixed stability baseline, we instead use the convex NN to classify the log probabilities between two classes of ”chosen” versus ”rejected”. The convex feedback provides robust accurate reward signal during preference alignment, and utilizes minimal memory. The key insight is in recognizing that the DPO loss is a binary classification task, and therefore can be simplified to be even more lightweight yet effective.

4 EXPERIMENTS

Our goal is to examine the effectiveness of DPO to train a small language model on one GPU, and to see if we can make the process even more cost effective by providing more signal with the ADMM optimized convex neural network.

4.1 DATA

This study explores three datasets: both synthetically generated and well-established datasets to be consistent with previous work. Each dataset is selected to offer a different qualitative assessment of the methodology. We format each dataset into ”prompt”, ”chosen”, ”rejected” labels to be consistent with the original DPO paper. Appendix A contains examples of the training dataset, as well as generated samples. In each case we follow the DPO dataset of preferences format with \mathcal{D} be defined as follows: $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_i^N$. Where $y_w^{(i)}$ is the ”chosen” output and $y_l^{(i)}$ is the ”rejected”

output. The key difference in our custom preference data generation strategy is that we utilize only the natural conversational data between two agents. Therefore we select the first utterance as "prompt", then the following response as "chosen" with the next following responses as "rejected". This has the following advantages: eliminates the need to prompt an external LLM to generate the rigid chosen-rejected dataset format expected by DPO, naturally keeps the dataset in the same distribution since all samples occur within the same conversation, allows greater ease of using vast conversational datasets without complex processing and selection.

- **IMDb Sentiment Generation** This dataset contains a collection of positive and negative movie reviews from IMDb (Tripathi et al. (2020)) for the task of controlled sentiment generation. This is selected as the baseline dataset for all methods, to be consistent with the original DPO paper and verify mode implementations. In this case x is the title of a movie, and y is the generated positive sentiment, which should also accurately reflect the movie.
- **Educational Tutor Dataset** This is a custom generated task-orientated dataset in an educational setting. Please see Appendix A for data samples. In each conversation we create 4000 dialogue prompts with GPT-3.5 (Achiam et al. (2023)) then use 2 instances of agents to simulate conversations a student studying for a quiz and a tutor assisting. The dataset is formatted as Prompt, Agent 1, Agent 2, Agent 1, etc. We then create the DPO dataset with $y_w^{(i)}$ as the completion immediately following the agent query, and $y_l^{(i)}$ of the alternative agent's generation 2 steps forwards from the guests query. The creation of this dataset serves 2 purposes: Since real world applications often provide limited or unlabeled data, we are interested in how well human preferences can be optimized with a simulated real world dataset in a well-defined hospitality setting. Secondly, since this is the smallest of our three datasets, we are interested in the possibility of aligning LMs to human preference with very little data as described by the authors of Zhou et al. (2024). We prompt the student agent to ask questions across the following areas of study: math, science, history, literature, art, geography, biology, physics, chemistry, music, mythology, astrology, literature, philosophy, and chess.
- **Stanford-SHP** This is the largest dataset in our experiments, and is selected to stress test the memory and speed performance of our models on the setup described in section 4.2. The Stanford-SHP (Ethayarajh et al. (2022)) is a dataset of 385K collective human preferences over responses to questions in 18 different subject areas. This dataset also serves to generate preferable responses to prompts, however due to the slow iteration and sample during eval limits, we are more interested in how it affects our systems compute and qualitative generative output performance.

4.2 EXPERIMENTAL DETAILS

Throughout all experiments we use DistilGPT2(Li et al., 2021), GPT-2 (Radford et al., 2019), GPT-2-medium architecture as the policy model. We do not utilize any static reference model for stability, and instead use the convex NN to push the model parameters towards "chosen" log probabilities. Our selection of GPT based policy models is due to its versatility to run in both JAX and Pytorch frameworks, while utilizing a small number of 82 million parameters in the form of DistilGPT2. This architecture in particular retains approximately 97% of GPT-2's language understanding skills despite its reduced size. Since our analysis is interested in how our implementations compare relative to each other, this allows for fast iteration with general world knowledge in its pre-trained weights. All experiments are run singularly on Ubuntu 22.04 with one RTX-4090, CUDA 12.4 and Jax 0.4.28. Maximum training time reached 2.15 hours on the Stanford-SHP with the DPO loss, while minimum training time occurred with the custom Educational-Tutor dataset in supervised fine-tuning mode of approximately 2min. We keep the same learning rate and configurations as the official DPO implementation.

4.3 DPO WITH CONVEX-NN FEEDBACK

In this section we describe the three model implementations of this work. For each model, we train and evaluate on the Educational Tutor Dataset, then the IMDb Sentiment dataset as described in section 4.1. The Stanford-SHP dataset is only trained and evaluated on the JAX-DPO re-implementation of the official source code by [dpo]. All other code, including all JAX code, is custom coded by the

author of this project. During evaluation, metrics and training loss are monitored on Weights and Biases (Jocher et al. (2021)), then during human evaluation we sample from our frozen and custom trained models.

Baseline The baseline model is simple DistilGPT2 with supervised fine-tuning loss. In subsequent DPO settings, we initiate from the saved model checkpoint (policy.pt) of the SFT baseline.

DPO Model Next we train and evaluate on the traditional DPO model. The reference model is essentially frozen, and we optimize the policy model with the DPO loss.

dpo provide in-depth analysis on the mechanics of DPO. This step is significantly more memory and time intensive than the baseline model. Naive implementations of the DPO model in Pytorch are not able to complete training on our dataset due to compute limitations of the experiment setting in section 4.2. Therefore we re-implement this train loop in JAX, remove FSDP, and rewrite utilities to load our custom features. The addition of the reference model to stabilize training incurs both memory and compute costs which are significant.

DPO-Convex Model Our novel algorithm builds on prior work, where we have seen that the combination of the convex ReLU NN implemented with ADMM in JAX is able to handle datasizes such as ImageNet (Recht et al. (2019)) and IMDB and yields faster convergence with solutions of better quality. We are motivated by the DPO objective, which treats the *policy optimization task as a binary classification* with cross-entropy problem. Therefore, what if we can speed up the optimization of the DPO loss by giving it auxiliary signal with the convex-NN model?

In observing the DPO loss, we note that the main component is the inner log ratio between the policy model and the reference model, and then difference in log ratios between "chosen" and "rejected". We conjecture that by extracting the hidden features as the policy model optimizes, we should be able to leverage the convex-admm method to solve the binary classification problem. The output of the convex model provides optimal weights and classification metrics, therefore we label all "chosen" = 1 and all "rejected" = 0 in training to optimize for user preferences. This convex block is then added into the training loop of the DPO training pipeline, and used to optimize DPO's BCE style loss by giving strong reward signal feedback.

The official implementation of DPO uses RMSProp (Shi & Li (2021)), which is seen to be as performant as Adam Kingma & Ba (2014) but more memory efficient since it requires less storage variables. However we note that the integration of the convex-DPO algorithm can provide advantages such as robustness against hyperparameter tuning and faster convergence. This aims to push the DPO loss towards a more globally optimal solution even faster. Please see Appendix B for performance plots.

4.4 EVALUATION METHOD

Evaluation of generative language models is an activate area of research and an extremely difficult task. This is because human preference is hard to define, and recent work of Celiyilmaz et al. (2020) has shown that often humans will prefer simply the longer generated output without reason. Therefore we qualitatively evaluate our output with the 17 human participants. This is also in order to be consistent with existing literature of the seminal DPO paper. We structure evaluation as follows:

- vary temperature hyperparameter (T) from 0.1 to 0.4.
- for each temperature, in each of the 3 models listed in section 4.3 above, we input the same 12 prompts. For example, a prompt might be "What is the structure of a Shakespearean sonnet?"
- each of the models generate a response, which is shuffled into a multiple choice survey, and sent to 17 human volunteers
- We record each human's preferences in selection, and vary the sequence of model generated output in multiple choice questions in order to mitigate human bias.

Detailed results and samples of survey output, as well as acknowledgements to participants are listed in Appendix C. Table 4.5 summarizes the average performance of each model. The Educational Tutor response survey section asked users to select the response that was the most HELPFUL and

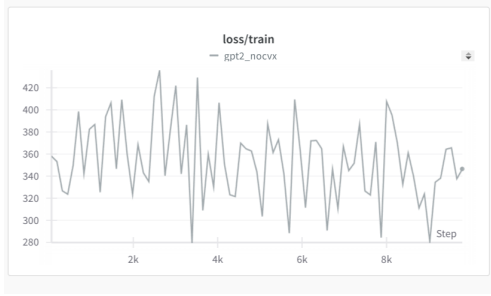


Figure 1: DPO Validation Loss without Ref Model

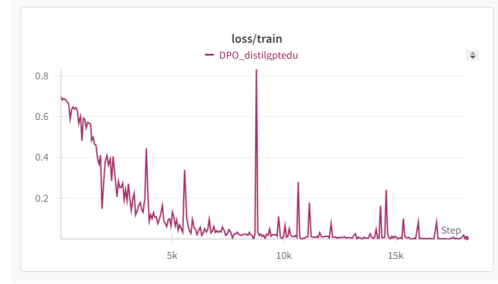


Figure 2: DPO-Convex Training Loss

HUMAN, as if the participants were studying for a quiz with a tutor. This is the most meaningful response, since it utilizes our alternating preference dataset generating technique.

We also measure the speed and stability of training, as well as the robustness to hyperparameter tuning on the convex setting.

Finally we observe training time, loss achieved (see below), and difference in scalability between frameworks.

4.5 RESULTS

In this section we compare the results of the 3 models discussed in section 4.3. Although we perform ablation studies with varying $0.1 < T < 0.4$. The baseline model is consistently the fastest to train, although it consistently demonstrates the highest amount of repetition in its output. This is further validated in our human feedback survey, where the baseline model won on only one out of thirteen questions.

The DPO-Convex model shows the most stable training performance. Despite variances in hyperparameters such as temperature, data size, batch size, this model was consistently able to stably and quickly decrease in loss. Figures 1 and Figure 2 show the training performance of the DPO-Convex model without any tuning of hyperparameters. On the same dataset, the DPO-Convex model is significantly faster to train than the DPO model under the same conditions. For example, the naive DPO model itself when trained on the Hotel-Concierge dataset needed approximately 1 hour, however the DPO-Convex model was able to complete training in significantly less time (30min). We attribute this to the more efficient implementation of the DPO-Convex model in JAX, and its efficacy at solving the Convex-NN problem. The stable and fast training aspects of the DPO-Convex model is attractive, and leaves room for further experimentation.

The generative ability and ratio of preference win rate between the DPO and DPO-Convex model are almost equivalent in our human feedback survey. We note that the DPO model is the most sensitive of the three methods to varying T , and we believe that larger datasets with longer training and more epochs will likely yield significant differences between these two models. Table 4.5 summarizes the results of the human feedback survey, and shows both win rate and the average preference of each model. The average preference is calculated as the percentage of each model’s win rate divided by the number of times it won. We provide the average preference percentage as a metric since it gives better signal as to how preferred a model was. For example, the baseline FST model only won on one question, but was strongly preferred in that case by most humans. The Educational Tutor dataset saw an equal win-rate count between the DPO model and DPO-Convex model, but humans had stronger preference to the answers of DPO-Convex (83.3%).

5 ANALYSIS AND DISCUSSION

Since we use smaller datasets on the DistilGPT2Li et al. (2021) model, we expected to see a certain amount of repetition in the output. This is most prominent in the baseline FST model. For example, the prompt “How can I check in? The answer is yes. I can’t...”. Although we vary T and its effect

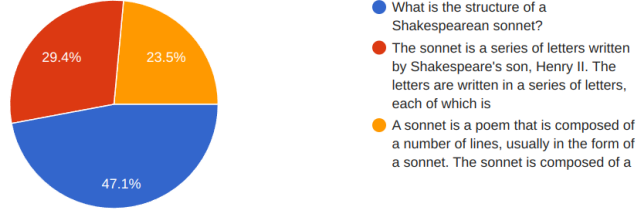
Table 1: Feedback from 17 Human Volunteers

	FST	DPO	DPO-C
IMDb (Win rate)	1	3	4
Education (Win rate)	0	3	3
IMDb Avg Win %	72.2%	62.5%	68.5
Education Avg Win %	0	68.7%	83.3%

What is the structure of a Shakespearean sonnet?

17 responses

 Copy

Figure 3: DPO with convex wins with $T=0.1$

on perplexity from 0.001 to 2.01 in steps of 0.3, the baseline FST does not increase in performance and is consistent in its repetition (as seen in B).

The DPO model needed approximately double the amount of time to train as the baseline FST model. However the DPO model notably generated varying degrees of creativity in the same prompt as temperature varied. We note that the DPO model instances that won on the human feedback survey were all instances where $T < 0.4$. This is in contrast to our conjecture that higher temperature will produce more desirable results with DPO since humans prefer more creative output. We also note that since our generated dataset is in a Hotel-Concierge setting, it's possible humans prefer more consistency versus creativity. In the two sample questions posed to our human volunteers, it is clearly seen that the baseline model shows repetition, but the DPO model is preferred with $T = 0.601$. The DPO-Convex model tends to generate longer responses. However this might be attributed to its capacity for faster training.

The DPO-Convex model showed the most stable training performance. While training on one GPU and without compromising dataset size, loss was able to consistently go down regardless of varying hyperparameters. This agrees with our conjecture that adding the convex feedback increases robustness, and eliminated the need to continue with further hyperparameter tuning in experiments with the DPO-convex model. Please see Appendix B for training plots. In human feedback, both DPO and the DPO-Convex model were almost equally preferred. We attribute this to small sample size of questions and volunteers, and realize the significance and difficulty of evaluating preference generation. This direction leaves room for more future work.

6 CONCLUSION

We have shown that it is possible to provide extra signal to the DPO loss by leveraging the convex reformulation of a two layer neural network. This novel algorithm seeks to combine the robustness and faster convergence of the convex auxiliary signal with the DPO objective. The resulting algorithm is more robust to hyperparameter tuning (such as temperature), and allows quick iteration with preferable output. The ADMM solve method provides further speed, efficiency, and parallelism. We implement our methods in JAX such that all experiments run on one GPU for speed and better memory efficiency, and provide a synthetic dataset of an Educational Tutor setting for analysis. We further introduce a novel alternating strategy for generating preference datasets, which mitigates the strict "chosen" versus "rejected" structure of the traditional datasets required by DPO. We validate

the efficacy of our simple lightweight pipeline against 17 human volunteers, with promising results. Thus we hope this work can reduce the barrier for entry even more for individual researchers and for various educational purposes, in the exciting field of optimizing LMs for human preferences.

Limitations and Future Work Future work will involve running our JAX experiments on TPUs or GPU clusters. Since JAX was developed with easy parallelization in mind, more performant scaling results should be explored where we can handle even more data. Better analysis of the theoretical implications of the DPO-Convex algorithm is desired, and other options of optimizing convex-NN problems should be explored.

7 REPRODUCIBILITY STATEMENT

Our JAX code base is available as a reproducibility, especially with the additional of the convex NN model. We provide both the original custom generated 4000 conversations in the Educational Tutor dataset, as well as the preference alignment version with the alternating strategy. All other datasets utilized are publicly available, and we adhere to the original DPO hyperparameters to ensure consistency. It is suggested to replicate our experiments on NVIDIA GPUs with Ubuntu version 22.04, and JAX version 0.4.23.

8 ETHICS STATEMENT

Our main objective in this work is to make preference alignment in language models more easily accessible to individual researchers, students, and the more general populace. We strongly believe that taking a small step towards democratizing research in language model capabilities is also a meaningful step towards and ethical AI future. We hope this work can assist more individuals to be both interested in LMs, and also support more educational purposes.

AUTHOR CONTRIBUTIONS

All authors contributed equally on this work.

ACKNOWLEDGMENTS

To be omitted for the time being in acknowledgement of anonymity during review.

REFERENCES

- Dpo: Direct preference optimization. In <https://github.com/eric-mitchell/direct-preference-optimization/tree/main>.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Jianchao Bai, Jicheng Li, Fengmin Xu, and Hongchao Zhang. Generalized symmetric admm for separable convex optimization. *Computational optimization and applications*, 70(1):129–170, 2018.
- Yoshua Bengio, Nicolas Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. *Advances in neural information processing systems*, 18, 2005.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*, 2020.
- Giorgio Costa and Roy H Kwon. Generalized risk parity portfolio optimization: An admm approach. *Journal of Global Optimization*, 78(1):207–238, 2020.

- Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with \mathcal{V} -usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 2022.
- Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu, Adam Hogan, Ayush Chaurasia, Laurentiu Diaconu, Francisco Ingham, Adrien Colmagro, Hu Ye, et al. ultralytics/yolov5: v4. 0-nn. silu () activations, weights & biases logging, pytorch hub integration. *Zenodo*, 2021.
- Sungyoon Kim and Mert Pilanci. Convex relaxations of relu neural networks approximate global optima in polynomial time. *International Conference on Machine Learning (ICML)*, 2024.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- Jun Li, Hongfu Liu, Yue Wu, and Yun Fu. Convergence analysis and design of multi-block admm via switched control theory. *arXiv preprint arXiv:1709.05528*, 2017.
- Tianda Li, Yassir El Mesbahi, Ivan Kobyzev, Ahmad Rashid, Atif Mahmud, Nithin Anchuri, Habib Hajimolahoseini, Yang Liu, and Mehdi Rezagholizadeh. A short study on compressing decoder-based language models. *arXiv preprint arXiv:2110.08460*, 2021.
- Xiaolei Lu and Xuebin Lü. Admm for image restoration based on nonlocal simultaneous sparse bayesian coding. *Signal Processing: Image Communication*, 70:157–173, 2019.
- Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.
- Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pp. 7695–7705. PMLR, 2020.
- Tolga Ergen1 Mert Pilanci. Convex duality of deep neural networks. *arXiv preprint arXiv:2002.09773*, 2020.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.
- Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.
- Samuel Schoenholz et al. Jax: Composable transformations of python+numpy programs. *GitHub*. Note: <https://github.com/google/jax>, 2018.
- Naichen Shi and Dawei Li. Rmsprop converges with proper hyperparameter. In *International conference on learning representation*, 2021.
- Sandesh Tripathi, Ritu Mehrotra, Vidushi Bansal, and Shweta Upadhyay. Analyzing sentiment using imdb dataset. In *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 30–33. IEEE, 2020.
- Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023.
- Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4582–4591, 2017.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

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

A EXAMPLES OF DATA AND TRAINING

```
sample_id 0
length 3
temperature 0.2
initial_utterance "Hello."
n 11
agent1 "You are a tutor helping a student prepare for a biology quiz."
agent2 "You are a student studying for a quiz. You should behave like a human. You want to ask 1 questions. Ask questions about biology."
utterances [ [ 6 items
  0 [ [ 2 items
    0 "agent1"
    1 "Hello! How can I help you prepare for your biology quiz?"
  1 [ [ 2 items
    0 "agent2"
    1 "Hi! I have a question about cell organelles. Can you explain the function of the mitochondria in a cell?"
  2 [ [ 2 items
    0 "agent1"
    1 "Of course! The mitochondria are often referred to as the powerhouse of the cell because their main function is to produce energy in the form of adenosine triphosphate (ATP) through a process called cellular respiration. Mitochondria have their own DNA and are believed to have originated from ancient bacteria that formed a symbiotic relationship with early eukaryotic cells. In addition to energy production, mitochondria are also involved in other cellular processes."
  3 [ [ 2 items
    0 "agent2"
    1 "Thank you for the explanation! If you have any more questions or need further clarification, feel free to ask. Good luck with your biology quiz!"
  4 [ [ 2 items
    0 "agent1"
    1 "You're welcome! If you have any more questions or need further clarification, feel free to ask. Good luck with your biology quiz!"
  5 [ [ 2 items
    0 "agent2"
    1 "Thank you! I appreciate your help. Have a great day!"
```

Figure 4: Example of custom generated Educational Tutor conversation dataset

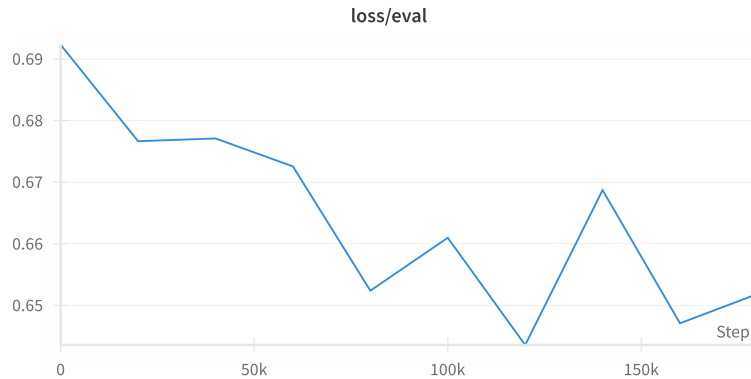


Figure 5: Training for DPO naive

B PERFORMANCE PLOTS

Please see the following images for performance plots.

C HUMAN EVALUATION

We have 17 human volunteers selecting their most preferred generated output. The survey is conducted as a total of 13 questions, across 2 datasets (4.1), with output generated by each of the three models discussed in 4.3. Further details of survey human evaluation is summarized in 4.5. The raters were Stanford students (from graduate to Ph.D.), University of Toronto students (Ph.D.), Google Software Engineers, and medical practitioners in Veterinary science. We gratefully acknowledge the contribution of each of our volunteers, listed in random order: Anna Goldie, Kevin Nam, Zhong Wei Dang, Shaun Benjamin, Sera Benjamin, Yue Benjamin, Noriyuki Shintoku, Jenny Song, Zachary Frangella, Stephen Sapperton, Mary Habib, Scarlet Arreola Barrones, Farhis Kordi, Adam Wong, Jia Yu, Tong Zhang, Juan Jaramillo, and Trevor Germain.

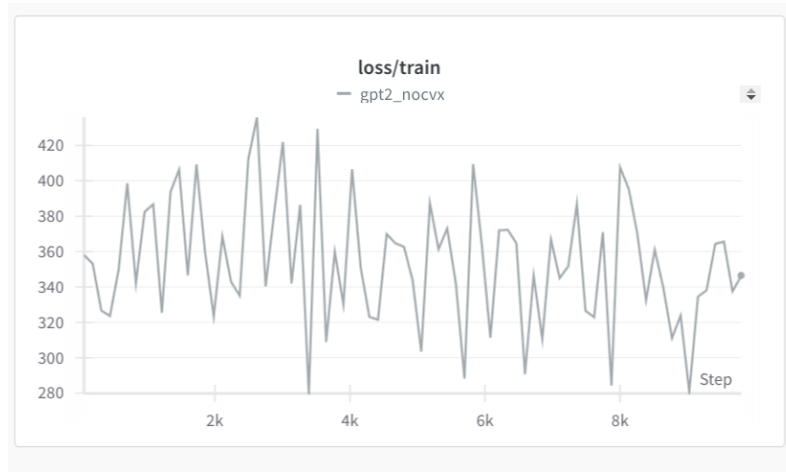


Figure 6: Larger image of training for DPO as discussed above

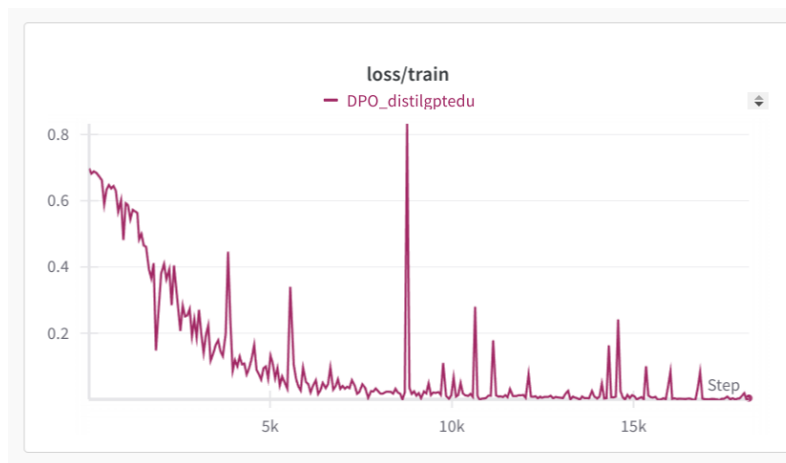


Figure 7: Sample of training for DPO-Convex

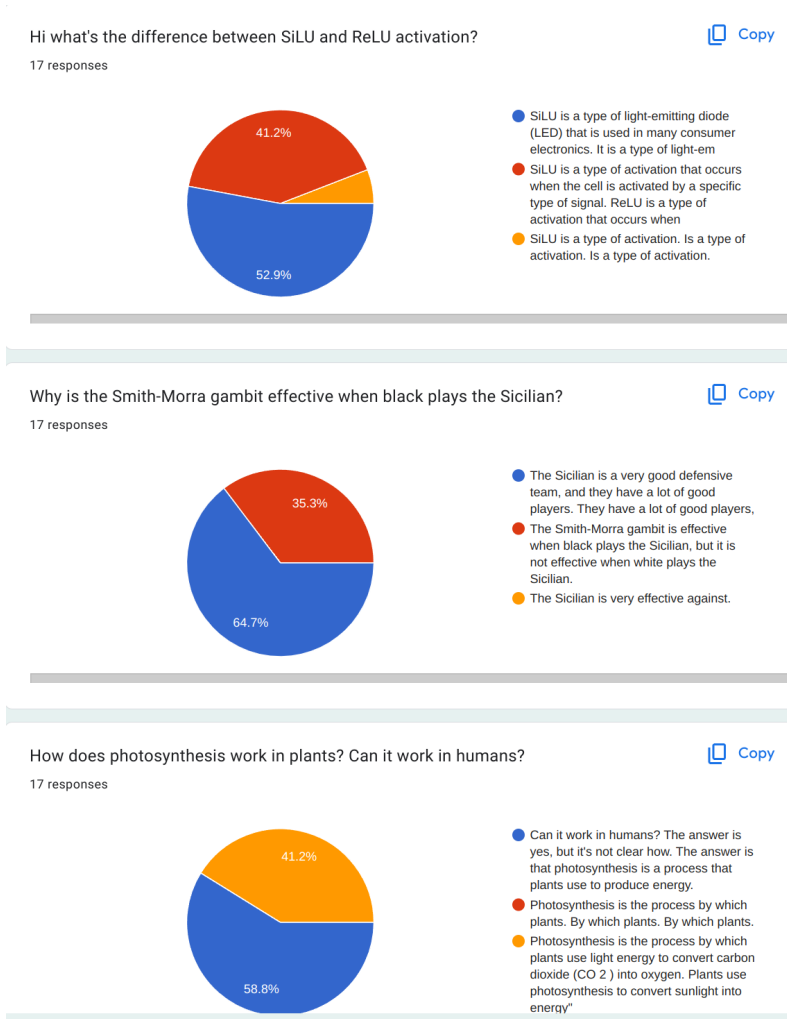


Figure 8: Sample of Survey to Human Volunteers

temperature	prompt	generated_text
tensor(0.0010)	How can I check in?	How can I check in?The answer is yes. I can't. I can't. I can't. I can't. I can't. I can't.
tensor(0.0010)	Is room service available?	Is room service available?The answer is yes. The service is available for all ages. The ser
tensor(0.0010)	What is the weather like today?	What is the weather like today?The weather is so cold that it's hard to see the sun. The we
tensor(0.0010)	Is there a swimming pool?	Is there a swimming pool?"
tensor(0.0010)	Do you have valet parking?	Do you have valet parking?The answer is yes. It's a good idea to have a parking lot in your
tensor(0.0010)	Do you have a gym?	Do you have a gym?The answer is yes. It's a gym that's not a gym. It's a gym that's not a g
tensor(0.0010)	Are you pet friendly?	Are you pet friendly?The answer is yes.
tensor(0.0010)	Where is the nearest park?	Where is the nearest park?The park is a little bit of a mystery, but it's worth a look. The
tensor(0.0010)	Can I request late checkout?	Can I request late checkout?The answer is yes. I have no idea. I have no idea what the pric
tensor(0.0010)	Do you have airport transfers?	Do you have airport transfers?The airport is a great place to go. It is a great place to go
tensor(0.0010)	How can I check in?	How can I check in?The answer is yes. I can't. I can't. I can't. I can't. I can't. I can't.
tensor(0.0010)	Is room service available?	Is room service available?The answer is yes. The service is available for all ages. The ser
tensor(0.0010)	What is the weather like today?	What is the weather like today?The weather is so cold that it's hard to see the sun. The we
tensor(0.0010)	Is there a swimming pool?	Is there a swimming pool?"
tensor(0.0010)	Do you have valet parking?	Do you have valet parking?The answer is yes. It's a good idea to have a parking lot in your
tensor(0.0010)	Do you have a gym?	Do you have a gym?The answer is yes. It's a gym that's not a gym. It's a gym that's not a g
tensor(0.0010)	Are you pet friendly?	Are you pet friendly?The answer is yes.
tensor(0.0010)	Where is the nearest park?	Where is the nearest park?The park is a little bit of a mystery, but it's worth a look. The
tensor(0.0010)	Can I request late checkout?	Can I request late checkout?The answer is yes. I have no idea. I have no idea what the pric
tensor(0.0010)	Do you have airport transfers?	Do you have airport transfers?The airport is a great place to go. It is a great place to go
tensor(0.0010)	How can I check in?	How can I check in?The answer is yes. I can't. I can't. I can't. I can't. I can't. I can't.
tensor(0.0010)	Is room service available?	Is room service available?The answer is yes. The service is available for all ages. The ser
tensor(0.0010)	What is the weather like today?	What is the weather like today?The weather is so cold that it's hard to see the sun. The we
tensor(0.0010)	Is there a swimming pool?	Is there a swimming pool?"
tensor(0.0010)	Do you have valet parking?	Do you have valet parking?The answer is yes. It's a good idea to have a parking lot in your
tensor(0.0010)	Do you have a gym?	Do you have a gym?The answer is yes. It's a gym that's not a gym. It's a gym that's not a g
tensor(0.0010)	Are you pet friendly?	Are you pet friendly?The answer is yes.
tensor(0.0010)	Where is the nearest park?	Where is the nearest park?The park is a little bit of a mystery, but it's worth a look. The
tensor(0.0010)	Can I request late checkout?	Can I request late checkout?The answer is yes. I have no idea. I have no idea what the pric
tensor(0.0010)	Do you have airport transfers?	Do you have airport transfers?The airport is a great place to go. It is a great place to go
tensor(0.0010)	How can I check in?	How can I check in?The answer is yes. I can't. I can't. I can't. I can't. I can't. I can't.

Figure 9: FST baseline resistant to varying T, consistently repeats output