

Preference Optimization for Parameter-Efficient Multi-Task Learning of GPT-2

Stanford CS224N Default Project

Jiecong Tan
Department of MS&E
Stanford University
jctan03@stanford.edu

Yichang M. Yang
Department of MS&E
Stanford University
yangmar@stanford.edu

Abstract

We investigate how preference optimization methods can improve Shakespearean sonnet generation beyond standard supervised fine-tuning (SFT) of GPT-2. We conduct a systematic study of hyperparameter selection and parameter-efficient fine-tuning, finding that GPT-2-large outperforms GPT-2-medium under SFT, and LoRA achieves competitive performance with only 0.38% of trainable parameters. We apply Direct Preference Optimization (DPO) and Simple Preference Optimization (SimPO) using heuristically constructed preference pairs, with SimPO achieving a final test CHRF score of 42.658. A concern is that the SFT model largely memorizes the small training corpus, limiting the effectiveness of DPO and SimPO. Overall, our results demonstrate that combining SimPO and LoRA is a promising direction for structured poetry generation with limited data and compute.

1 Key Information to include

- Mentor: Fang Wu
- External Collaborators (if you have any): None
- Sharing project: No
- Late Days (if you are using any):

Team member	Late days remaining
Jiecong Tan	4
Yichang M. Yang	3
Total pooled	7
Team size	2
Late days applied for the report	3

2 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in natural language generation, yet producing high-quality, structured creative text remains a significant challenge. In this work, we first implement core components of the GPT-2 architecture, then finetune it for downstream tasks including sentiment analysis and paraphrase detection. Then, we turn our primary focus to the open-ended task of Shakespearean sonnet generation, where we investigate how preference optimization methods can push generation quality beyond what standard fine-tuning achieves.

While GPT-2 is capable of fluent text generation, producing Shakespearean sonnets requires careful fine-tuning and alignment. The model must simultaneously capture Shakespearean vocabulary and stylistic conventions, and respect rigid structural requirements. Standard supervised fine-tuning (SFT) on a small training dataset of Shakespeare's 130 sonnets makes overfitting a persistent concern, and cross-entropy training provides little signal about structural quality or stylistic preference.

Prior work on language model alignment has largely focused on Reinforcement Learning from Human Feedback (RLHF), which can be implemented through Proximal Policy Optimization (PPO). While effective, PPO-based approaches require training and maintaining multiple large models simultaneously, making them computationally prohibitive at our scale. Direct Preference Optimization (DPO) has emerged as a compelling alternative that eliminates the need for an explicit reward model by directly optimizing on preference data. Building on DPO, Simple Preference Optimization (SimPO) further simplifies the pipeline by removing the need for a reference model entirely, instead using length-normalized reward margins to distinguish preferred from dispreferred outputs.

Our approach explores how these two preference optimization methods can improve sonnet generation quality beyond what standard SFT achieves. We fine-tune GPT-2-large on Shakespeare’s sonnets, then apply SimPO with LoRA adapters to align the model toward higher quality outputs using preference pairs constructed from heuristic corruptions of training sonnets. We additionally investigate the effect of model scale, learning rate, and generation hyperparameters on output quality, evaluating all models using the CHRF metric against held-out sonnets.

Our key findings are as follows. Fine-tuning GPT-2-large outperforms GPT-2-medium, achieving a CHRF score of 42.28 compared to 41.76. Applying SimPO with carefully tuned hyperparameters further improves performance to 43.16 CHRF, demonstrating that preference optimization provides meaningful gains even with heuristically constructed preference data rather than human annotations. We also find that LoRA fine-tuning achieves competitive results with only 0.38% of trainable parameters, suggesting that parameter-efficient methods are a viable approach for creative text generation tasks with small datasets.

3 Related Work

The foundation of modern preference-based alignment was established by Stiennon et al. [1], who showed that training on pairwise human preferences produces outputs significantly preferred over supervised fine-tuning alone. Ouyang et al. [2] scaled this approach to general-purpose instruction-following with InstructGPT, demonstrating that RLHF can align language models with human intent across diverse tasks. However, PPO-based RLHF requires simultaneously maintaining a policy model, reference model, and reward model, making it computationally expensive. Casper et al. [3] provide a comprehensive overview of these practical limitations, arguing that this accessibility barrier has significantly constrained adoption among researchers with limited resources, which is a constraint directly applicable to our setting.

Rafailov et al. [4] introduced Direct Preference Optimization (DPO), which eliminates the separate reward model by reparameterizing the RLHF objective as a simple binary cross-entropy loss over winning and losing completions. However, subsequent work has identified practical limitations: Ivison et al. [5] found PPO outperforms DPO by up to 2.5% in mathematics and 1.2% in general domains, and Xu et al. [6] identify sensitivity to distribution shift as a fundamental weakness, showing DPO struggles on challenging tasks where properly tuned PPO succeeds.

These findings motivated our choice to also explore SimPO [7], which further simplifies preference learning by removing the reference model entirely and using length-normalized rewards with a target margin. This is particularly suited to sonnet generation, where output length is a key quality criterion and memory constraints of our virtual machine preclude maintaining multiple large models simultaneously. While preference optimization has been extensively studied for summarization, dialogue, and instruction-following, its application to structured creative text generation has potential to be further explored, and our work examines whether heuristically constructed preference pairs can provide meaningful alignment signal in the absence of human annotations.

4 Approach

4.1 Base Model and Supervised Fine-Tuning

Our base model is GPT-2-large [8], which was chosen over GPT-2-medium on the basis of its greater capacity for capturing the complex stylistic patterns of Shakespearean poetry. After implementing the core GPT-2 architecture including multi-head self-attention, we first validate our implementation on sentiment analysis and paraphrase detection, where both tasks are trained using the same standard

language modeling objective, minimizing cross-entropy loss over next-token predictions. We then apply the same objective to fine-tune the model on a corpus of 130 of Shakespeare’s sonnets.

During fine-tuning, we use the AdamW optimizer with a learning rate of 1×10^{-5} and batch size of 4. Based on validation CHRF scores, we select the checkpoint at epoch 25 as our base model, which achieves a CHRF score of 42.3 on the held-out dev set. We observe that training loss continues to decrease well beyond this point (reaching 0.066 by epoch 30) while dev CHRF degrades, confirming overfitting on the small training corpus.

For generation, we use top- p (nucleus) sampling with temperature $\tau = 0.7$ and $p = 0.9$. At each step, we sample from the smallest set of tokens whose cumulative probability mass exceeds p , after scaling logits by temperature:

$$p_i \propto \exp(\log p_\theta(x_t = i)/\tau)$$

4.2 LoRA Fine-Tuning

To address overfitting on the small sonnet corpus, we implement Low-Rank Adaptation (LoRA) [9]. Rather than updating all model parameters, LoRA freezes the pretrained weights and injects trainable low-rank decompositions into the query and value projection matrices of each self-attention layer. For a pretrained weight matrix $W \in \mathbb{R}^{d \times k}$, the adapted forward pass becomes:

$$h = Wx + \frac{\alpha}{r}BAx$$

where $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{d \times r}$ are the trainable low-rank matrices, r is the rank, and α is a scaling factor. We initialize A with Kaiming uniform and B with zeros, so the LoRA contribution is zero at the start of training. With rank $r = 16$ and $\alpha = 32$, this reduces trainable parameters for a GPT-2-large model to approximately 2.9M out of 774M total (0.38%), providing strong regularization while preserving the expressivity of the pretrained model.

4.3 Preference Pair Construction

For DPO, we use a direct contrast strategy; the model’s own generated continuation serves as the dispreferred completion y_l and the ground-truth sonnet as the preferred one. This encourages the model to prefer completions with stronger rhyme, meter, and fluency over its own current outputs.

For SimPO, we generate dispreferred completions using five heuristic corruption strategies: (1) *shuffle* — randomly reorder lines while preserving the final couplet; (2) *truncate* — keep only 8-10 lines; (3) *repeat* — duplicate 2-3 random lines; (4) *extra lines* — append 3 additional randomly sampled lines to exceed 14; (5) *wrong ending* — replace the final word of alternating lines to break the rhyme scheme. Applying all five strategies to each of 130 training sonnets yields 650 preference pairs.

4.4 DPO

DPO [4] eliminates the need for a separate reward model by reparameterizing the RLHF objective as a direct optimization over preference pairs. Given a prompt x , preferred completion y_w , and dispreferred completion y_l , DPO optimizes:

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

where π_θ is the policy being trained, π_{ref} is a frozen reference model (the SFT checkpoint), β controls deviation from the reference policy.

DPO requires maintaining both the policy model and a frozen reference model simultaneously, which doubles memory requirements relative to SFT. Due to this constraint, we were unable to apply DPO to GPT-2-large and instead use GPT-2-medium for DPO experiments. While this limits direct comparison with our SimPO results, it reflects a practical limitation of reference-model-based methods and motivates our primary focus on SimPO, which removes the reference model entirely. We train DPO with a batch size of 5 preference pairs, 20 epochs, and a learning rate of 1×10^{-6} .

4.5 SimPO

SimPO [7] addresses the memory overhead of DPO by eliminating the reference model entirely, instead using length-normalized rewards to distinguish preferred from dispreferred completions:

$$\mathcal{L}_{\text{SimPO}}(\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log p_{\theta}(y_w | x) - \frac{\beta}{|y_l|} \log p_{\theta}(y_l | x) - \gamma \right) \right]$$

where β controls the scale of the reward difference, $\gamma > 0$ is a target margin enforcing a minimum gap between winning and losing rewards, and $|y|$ denotes sequence length used for normalization. The length normalization prevents the model from preferring longer or shorter outputs regardless of quality, which is directly relevant to enforcing the 14-line sonnet structure.

We apply SimPO on top of our GPT-2-large SFT checkpoint, with LoRA adapters ($r = 16$, $\alpha = 32$) attached to the query and value projections of each attention layer. The base model weights remain fully frozen during SimPO training, so only the LoRA parameters are updated. We use AdamW with learning rate 1×10^{-6} , gradient clipping at 1.0, and train for 25 epochs with batch size 8.

5 Experiments

5.1 Data

We evaluate on three downstream tasks using standard datasets. For sentiment analysis, we use SST-2 [10], and for paraphrase detection, we use the Quora Question Pairs (QQP) dataset [11], formulated as a cloze-style generation task where the model generates “yes” or “no”. For sonnet generation, we use a corpus of Shakespeare’s 154 sonnets, split into 130 training sonnets and a held-out evaluation set. At inference time, the model is given the first 3 lines of a held-out sonnet as a prompt and must generate the remaining lines.

5.2 Evaluation methods

For sentiment analysis and paraphrase detection we report accuracy on the respective test sets. For sonnet generation we evaluate using CHRF [12], which measures character n -gram F-score between generated and reference sonnets, rewarding Shakespearean vocabulary and phrasing even when exact word overlap is sparse. We also conduct qualitative analysis of generated sonnets examining structural adherence and rhyming coherence, as CHRF alone does not capture these aspects.

5.3 Experimental details

5.3.1 Paraphrase Detection

We experiment with two configurations for paraphrase detection. Configuration 1 uses GPT-2-small with LoRA applied to the last 4 attention layers ($r = 8$, $\alpha = 8$), a constant learning rate of 1×10^{-5} , and batch size 16. Configuration 2 scales to GPT-2-medium with LoRA applied to the last 8 attention layers ($r = 16$, $\alpha = 16$), a learning rate that decays linearly from 1×10^{-4} to 1×10^{-5} over the first 10 epochs to address training instability, weight decay of 0.01, gradient clipping at 1.0, and batch size 32. We additionally explore a contrastive auxiliary loss weighted at 0.1 that encourages geometrically meaningful sentence representations by computing cosine similarity between independently encoded sentence embeddings.

5.3.2 Sonnet Generation

All experiments are run on a single NVIDIA GPU. We first establish a strong SFT baseline by iterating over batch size, model scale, learning rate, and generation hyperparameters before applying preference optimization.

Batch size. We compare batch sizes of 4 and 8 for SFT, finding that batch size 4 yields lower training loss and use it for all subsequent experiments.

Model scale. We train both GPT-2-medium and GPT-2-large with learning rate 1×10^{-5} and batch size 4 for up to 50 epochs, saving checkpoints every 5 epochs and selecting the best dev CHRF checkpoint.

Learning rate. For GPT-2-large, we compare learning rates of 1×10^{-5} and 5×10^{-6} , evaluating dev CHRF at each saved checkpoint.

Generation hyperparameters. Using the best SFT checkpoint (GPT-2-large, epoch 25, lr = 1×10^{-5}), we experiment with multiple values of temperature $\tau \in \{0.70, 0.75, 0.80\}$ and top- $p \in \{0.85, 0.90, 0.95\}$, evaluating dev CHRF for each combination.

LoRA. We apply LoRA to GPT-2-large with two rank configurations: $r = 4, \alpha = 8$, lr = 1×10^{-5} ; and $r = 16, \alpha = 32$, lr = 1×10^{-4} . Both configurations train for up to 50 epochs with batch size 4.

DPO. We apply DPO without LoRA to the best SFT checkpoint of GPT-2-medium. During training, we use gradient clipping at 1.0 to stabilize optimization, a DPO scaling parameter of $\beta = 0.1$ to control the strength of the preference objective, a batch size of 8 preference pairs, and a learning rate of 1×10^{-6} . We train for 10 epochs and select the checkpoint with the best development-set CHRF score.

SimPO. We apply SimPO on top of the best SFT checkpoint for GPT-2-large with LoRA adapters ($r = 16, \alpha = 32$), gradient clipping at 1.0, $\beta = 2.0$, $\gamma = 0.5$, and batch size 8. We compare three learning rates: 5×10^{-7} , 8×10^{-7} , and 1×10^{-6} , training for up to 55 epochs and selecting the best dev CHRF checkpoint for each.

5.4 Results

5.4.1 Paraphrase Detection

Model	LoRA Config	Training Time	10 Epochs	20 Epochs	40 Epochs
GPT-2 small	last 4 layers, $r = 8$	8 min/epoch	0.809	0.835	—
GPT-2 medium	last 8 layers, $r = 16$	11 min/epoch	0.823	0.845	0.872

Table 1: Paraphrase detection dev accuracy on the Quora dataset across training epochs.

Table 1 shows that GPT-2-medium outperforms GPT-2-small at every comparable epoch, confirming that model scale yields meaningful gains for this task. GPT-2-medium continues to improve with extended training, reaching 0.872 at 40 epochs, which we attribute to the expanded LoRA coverage across more layers with higher rank enabling more expressive task-specific adaptation. Our best configuration (GPT-2-medium with LoRA applied to the last 8 layers ($r = 16, \alpha = 16$), decaying learning rate, and 40 epochs of training) achieves a test accuracy of 0.869 on the QQP leaderboard.

5.4.2 Sonnet Generation

SFT baseline. Table 2 summarizes SFT results. GPT-2-large outperforms GPT-2-medium, achieving a best dev CHRF of 42.278 at epoch 25 compared to 41.759 for medium at epoch 40. This confirms that model scale is a key factor for SFT quality on this task. For GPT-2-large, the learning rate of 1×10^{-5} outperforms 5×10^{-6} (best CHRF 41.938), suggesting that faster convergence is beneficial given the small training corpus. In both cases, dev CHRF peaks well before training loss stops decreasing; for GPT-2-large at 1×10^{-5} , loss continues falling from 0.099 at epoch 25 to 0.049 at epoch 35 while CHRF degrades, providing clear evidence of overfitting to the 130 training sonnets (see Table 6 in the Appendix for full training curves). Our GPT-2-large SFT model achieves a test set CHRF of 41.273 on the leaderboard.

Model	LR	Best Epoch	Train Loss	Dev CHRF
GPT-2-medium	1×10^{-5}	40	0.334	41.759
GPT-2-large	1×10^{-5}	25	0.099	42.278
GPT-2-large	5×10^{-6}	50	0.087	41.938

Table 2: SFT results across model sizes and learning rates.

Generation hyperparameters. Tuning temperature and top- p on the epoch 25 GPT-2-large checkpoint, we evaluate each combination across multiple random seeds to account for sampling variance. We find $\tau = 0.75$, $p = 0.9$ produces the highest single-run score of 42.794, however $\tau = 0.7$, $p = 0.9$ achieves the highest average CHRF across seeds and is adopted as our default setting for all subsequent experiments (see Table 7 in the Appendix for full results).

LoRA. Full fine-tuning of GPT-2-large takes approximately 22 seconds per epoch, whereas both LoRA configurations reduce this to 11 seconds per epoch due to the significantly fewer trainable parameters. We find that LoRA with $r = 4$, $\alpha = 8$, and $\text{lr} = 1 \times 10^{-5}$ underfits severely, with loss still at 3.8 after 40 epochs, suggesting the rank is too low and the learning rate too small.

We therefore increase to $r = 16$, $\alpha = 32$, and $\text{lr} = 1 \times 10^{-4}$, which achieves a best dev CHRF of 41.743 at epoch 50, competitive with full fine-tuning despite using only 0.38% of trainable parameters (see Table 8 in the Appendix for full training curves). Also, since training time was identical for both rank configurations, we used the higher rank for greater expressivity for subsequent experiments.

DPO. Due to memory constraints from maintaining both policy and reference models, DPO experiments use GPT-2-medium. Table 3 shows training results across two runs of DPO fine-tuning, each preceded by SFT on GPT-2-medium. The first run achieves a dev CHRF of 37.981 after 5 DPO epochs, and the second achieves 41.184 after 10 DPO epochs. The SFT loss values across both runs remain high (~ 3.7 – 4.6), consistent with the weaker capacity of GPT-2-medium compared to GPT-2-large. The improvement from 37.981 to 41.184 across runs suggests DPO benefits from more training epochs, though performance still falls short of our GPT-2-large SFT baseline of 42.278, likely due to the model size constraint rather than a limitation of DPO itself.

Phase	Epoch	Loss	LR	Batch Size	Dev CHRF
SFT	0	4.561	1×10^{-5}	4	–
SFT	10	4.106	1×10^{-5}	4	–
SFT	20	3.954	1×10^{-5}	4	–
DPO	5	0.567	1×10^{-6}	1	37.981
SFT	30	4.112	1×10^{-5}	4	–
SFT	40	3.894	1×10^{-5}	4	–
SFT	50	3.720	1×10^{-5}	4	–
DPO	10	0.398	1×10^{-6}	1	41.184

Table 3: DPO training results on GPT-2-medium. SFT and DPO losses are not directly comparable as they use different objectives.

SimPO. Table 4 summarizes the best SimPO results across learning rates. SimPO consistently improves over the SFT baseline across all learning rates, with the best result of 43.158 achieved at $\text{lr} = 1 \times 10^{-6}$, epoch 20, representing a +0.88 CHRF improvement over the SFT baseline of 42.278. Lower learning rates converge more slowly but maintain competitive performance over a wider range of epochs: $\text{lr} = 5 \times 10^{-7}$ peaks at 43.054 at epoch 35, and $\text{lr} = 8 \times 10^{-7}$ peaks at 42.812 at epoch 30. In all cases, SimPO loss decreases monotonically but dev CHRF eventually degrades, consistent with the overfitting pattern observed in SFT. We additionally re-evaluate generation hyperparameters on the best SimPO checkpoint, but find that the default $\tau = 0.7$, $p = 0.9$ setting of 43.158 achieves the highest, confirming our original hyperparameter choice. Full training curves are provided in Table 9.

SimPO LR	Best Epoch	SimPO Loss	Dev CHRF
5×10^{-7}	35	0.446	43.054
8×10^{-7}	30	0.423	42.812
1×10^{-6}	20	0.432	43.158

Table 4: SimPO results across learning rates. All use GPT-2-large SFT (epoch 25) as base with LoRA $r = 16$, $\alpha = 32$.

We also investigated a sample-and-rank approach for preference pair construction, where candidates are generated at multiple temperatures (0.6, 0.7, 0.8 and 0.9) and ranked by CHRF score. However,

we found this ineffective; the SFT model achieves near-perfect CHRF (≈ 99.5) on training sonnets even at different temperatures due to memorization, leaving no meaningful quality gap between temperature-sampled candidates. When requiring a minimum CHRF difference of 0.5 between the best and worst candidate across four temperature settings, only 7 out of 130 sonnets produced a valid preference pair. This confirms that the base model has largely memorized the 130 training sonnets, and that heuristic corruptions are necessary to construct preference pairs with sufficiently large quality gaps to provide a reliable learning signal.

Qualitative analysis. Sonnets generated by our best SimPO model (see Appendix A.2 for examples) demonstrate strong stylistic similarities to Shakespearean conventions, such as archaic vocabulary (“thee”, “thou”). The couplet structure at the end is also mostly preserved, with the final two lines frequently indented.

However, structural limitations remain: line counts occasionally exceed 14, suggesting the model has not fully internalised the strict structural constraint despite SimPO training with truncation and extra-line corruption strategies. Also, rhyme scheme adherence is inconsistent, and thematic coherence sometimes degrades in later lines, with the model occasionally producing near-identical repeated lines within the same sonnet. This likely reflects the residual memorization of training sonnets negatively impacting novel generations.

Summary. Table 5 compares best results across all methods. SimPO applied to GPT-2-large achieves the strongest performance, outperforming both the SFT baseline and DPO. The gap between DPO and SimPO is partly attributable to the model size constraint forcing DPO to use GPT-2-medium, making it difficult to draw conclusions about the relative merits of the two preference optimization approaches independent of model scale.

Method	Dev CHRF	Test CHRF
GPT-2-medium SFT	41.759	–
GPT-2-medium SFT + DPO	41.184	–
GPT-2-large SFT	42.278	41.273
GPT-2-large LoRA	41.743	–
GPT-2-large SFT + SimPO	43.158	42.658

Table 5: Summary of best results across all methods.

6 Analysis

Our results suggest that model scale is one of the strongest drivers of baseline sonnet generation quality. Under supervised fine-tuning, GPT-2-large outperforms GPT-2-medium, indicating that larger models better capture the stylistic and structural regularities of Shakespearean sonnet completion. The training curves reveal a consistent overfitting pattern across all experiments: dev performance peaks before the end of training and then declines even as training loss continues to decrease, suggesting that token-level cross-entropy alone is insufficient to preserve generation quality once the model begins to memorize the limited training set.

Decoding hyperparameters matter, though their effect is smaller than model scale or preference optimization. Lower-temperature, moderate top-p sampling produces the most stable results, and our multi-seed evaluation reveals meaningful run-to-run variance, suggesting that model comparison should consider robustness across runs rather than isolated best-case outputs.

The LoRA experiments reveal a efficiency-quality tradeoff: stronger LoRA configurations perform close to full fine-tuning while updating only 0.38% of parameters and halving training time per epoch, making LoRA an attractive foundation for preference-based optimization under compute constraints.

Comparing DPO and SimPO highlights both the promise and practical constraints of preference optimization. DPO improves over its SFT checkpoint but is constrained by the memory overhead of maintaining a reference model, which prevented us from applying it to GPT-2-large. SimPO combined naturally with LoRA on GPT-2-large and produced the best overall results.

Crucially, the SimPO paper [7] identifies learning rate as the most critical hyperparameter for preference optimization, and our results support this; 1×10^{-6} achieves the fastest improvement

but overfits earliest, while 5×10^{-7} and 8×10^{-7} converge more slowly but sustain competitive performance over a wider range of epochs.

An important open question is whether this overfitting is partly a consequence of applying SimPO to an already heavily overfit base model. Our finding that the SFT model achieves near-perfect CHRF (≈ 99.5) on training sonnets suggests it may have largely memorized the training corpus, leaving little room for preference optimization to meaningfully reshape the model’s distribution. It is possible that SimPO would be more effective when applied to a less overfit base; an earlier SFT checkpoint or a LoRA-finetuned model that retains more generalization capacity. Investigating this interaction between base model overfitting and the effectiveness of preference optimization is a promising direction for future work.

Preference pair construction also proves critical. Sample-and-rank was ineffective because the SFT model produced outputs too similar in quality to create a meaningful preference signal. The heuristic corruptions we used were more effective, introducing clear structural failures that gave the model a reliable learning signal. This is especially important for sonnet generation, where quality depends on multiple factors line count and rhyme consistency, not just local fluency.

Overall, SFT teaches the model the language and style of Shakespearean sonnets, while SimPO adds a second layer of supervision that helps the model prefer outputs with stronger structural and stylistic qualities. SimPO was especially effective in our setting because it avoided the overhead of a reference model while still producing meaningful preference signal from heuristically-modified pairs.

7 Conclusion

In this project, we studied whether preference optimization can improve Shakespearean sonnet generation beyond standard supervised fine-tuning. We built SFT baselines with GPT-2 models of different scales, explored LoRA for parameter-efficient adaptation, and evaluated both DPO and SimPO for preference-based training. Our best model was SimPO with LoRA built on top of a strong SFT checkpoint, which outperformed both the standard SFT baseline and our DPO setup.

Our main takeaway is that preference optimization can improve structured creative generation even without human-annotated preferences, as long as the preference pairs capture meaningful quality differences. In our setting, heuristic corruptions were effective because they targeted common sonnet-specific errors such as broken structure, truncation, repetition, and rhyme disruption. We also found that compute constraints strongly affect which methods are practical: DPO was limited by higher memory cost, while SimPO integrated more naturally with a parameter-efficient setup.

Our work has several limitations. Most critically, the small training corpus of 130 sonnets makes memorization a persistent concern; our finding that the SFT model achieves near-perfect CHRF (≈ 99.5) on training sonnets suggests that preference optimization may be operating on an already saturated base model. Future work could explore whether leaving more of the finetuning process to SimPO could lead to more effective learning and greater generalization capacity on the held-out set.

Our comparison between DPO and SimPO is also not perfectly controlled, as memory constraints forced DPO onto a smaller model; a cleaner comparison on the same model size and compute budget would better isolate the contribution of each objective.

Finally, future work could explore richer preference pair construction strategies, such as mixing heuristic corruptions with model-generated negatives from a less overfit checkpoint where sample-and-rank would be more viable, or collecting human annotations over a subset of completions to better align the preference signal with true poetic quality.

Team contributions (Required for multi-person team)

We worked together on the paraphrase detection task, baseline sonnet generation modeling, and separately experimented with different extensions (DPO and SimPO). We also jointly prepared all project reports and the final poster.

References

- [1] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [2] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [3] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krashennnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. In *Transactions on Machine Learning Research*, 2023.
- [4] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [5] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [6] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is DPO superior to PPO for LLM alignment? A comprehensive study. In *ICML’24: Proceedings of the 41st International Conference on Machine Learning*, 2024.
- [7] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [8] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- [9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [10] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, 2013.
- [11] Quora. First quora dataset release: Question pairs. <https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>, 2017.
- [12] Maja Popovic. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 2015.

A Appendix

A.1 Results

Model	Epoch	Train Loss	Dev CHRF
GPT-2-medium (1×10^{-5})	10	2.960	–
	20	1.982	–
	30	0.928	40.325
	40	0.334	41.759
	50	0.120	41.290
GPT-2-large (1×10^{-5})	10	1.278	41.176
	15	0.472	41.430
	20	0.194	40.779
	25	0.099	42.278
	30	0.066	41.532
	35	0.049	–
GPT-2-large (5×10^{-6})	25	0.637	41.238
	30	0.387	40.860
	35	0.247	41.880
	40	0.169	40.965
	45	0.122	41.203
	50	0.087	41.938
	55	0.064	41.715

Table 6: Full SFT training curves across model sizes and learning rates.

Temp	Top-p	Run 1	Run 2	Run 3	Run 4	Average
0.70	0.90	42.278	41.800	42.286	–	42.121
0.70	0.95	42.333	41.945	42.392	41.484	42.039
0.75	0.90	42.794	41.698	41.975	41.054	41.880
0.75	0.95	40.945	42.080	–	–	41.513
0.80	0.90	41.888	–	–	–	41.888
0.70	0.85	41.408	–	–	–	41.408

Table 7: Dev CHRF scores across temperature and top- p combinations on the GPT-2-large epoch 25 checkpoint. Each run uses a different random seed. $\tau = 0.7$, $p = 0.9$ is selected based on highest average score.

Config	Epoch	Train Loss	Dev CHRF
$r = 4, \alpha = 8, \text{lr} = 1 \times 10^{-5}$	40	3.800	–
$r = 16, \alpha = 32, \text{lr} = 1 \times 10^{-4}$	25	2.716	41.501
	30	2.512	41.013
	35	2.326	41.574
	40	2.164	41.696
	45	1.972	41.723
	50	1.832	41.743

Table 8: Full LoRA training curves on GPT-2-large. Both configurations train for 11 seconds per epoch compared to 22 seconds for full fine-tuning.

LR	Epoch	SimPO Loss	Dev CHRF
5×10^{-7}	5	0.529	41.931
	10	0.512	41.550
	15	0.500	42.365
	20	0.482	42.679
	25	0.466	42.405
	30	0.454	42.910
	35	0.446	43.054
	40	0.435	42.301
	45	0.424	42.491
	50	0.414	42.080
8×10^{-7}	20	0.460	41.912
	25	0.440	42.390
	30	0.423	42.812
	35	0.405	41.568
	40	0.395	41.164
1×10^{-6}	10	0.502	41.983
	15	0.467	42.563
	20	0.432	43.158
	25	0.392	42.038
	30	0.366	41.621

Table 9: Full SimPO training curves across learning rates. All experiments use GPT-2-large SFT epoch 25 as base with LoRA $r = 16$, $\alpha = 32$, $\beta = 2.0$, $\gamma = 0.5$.

A.2 Generated Sonnet Examples

The following sonnets were generated by our best SimPO model (GPT-2-large, $\tau = 0.7$, $p = 0.9$) on the test set, given the first three lines as prompt.

Example 1:

Those lips that Love’s own hand did make
Breathed forth the sound that said “I hate”
To me that languished for her sake;
Those lips that taught my first song their shape and form
To repeat on another’s tongue the praise I gave.
Now are those lips blank, and in their place begin
New writers to tell of thee, others of me.
They begin with thy sweet face, which in themselves
Doth more beauty teach than any outward form;
And they go on to thy fingers, thy hair, and thy toes,
Increasing in number the beauty of thy face.
Then, if there be nothing more to say, there is no need
Of my further praise; but, so much the better, I will add,
Thy teeth are well worthy of that beauty that doth dwell in thee.

Example 2:

My love is as a fever, longing still
For that which longer nurseth the disease,
Feeding on that which doth preserve the ill,
Which it erst began. If it have no end,
What will its nurse’s care be but fleeting?
For she whose face with time’s change is pictured
In his former beauty will, I fear, be rude

And lack the delight of his sight.
Love's stillness is my appetite,
And death my lodging. She keeps me warm
By day's motion, and turns cold
In his absence. I am her food,
 And she is her eater. If she eat me not,
 My love dies also, and I alone perish.