

Learning to Rhyme with Token-Weighted DPO

Stanford CS224N Default Project

Fisher Marks

Department of Computer Science
Stanford University
ftm@stanford.edu

Abstract

Performing supervised fine-tuning (SFT) on pretrained language models can teach Shakespearean-like sonnet completions, but this method fails to reliably produce the Shakespearean rhyme scheme at any level. We investigate whether an adaptation of Direct Preference Optimization (DPO) can be used to align for this subsequence level preference. We introduce token-weighted DPO, a modification that heuristically weights the tokens belonging to line-final words in order to more strongly emphasize rhyme behavior. Our evaluations demonstrate that while standard and token-weighted DPO improve CHRF scores over an SFT-only baseline, token-weighted DPO fails to promote rhyme accuracy in generated completions. Attention distribution analyses shows that problems arising from a limited corpus dominate the weak rhyme signal in spite of introducing token weights.

1 Key Information to include

- Mentor: Simon Kim
- External Collaborators (if you have any): N/A
- Sharing project: N/A
- Late Days (if you are using any):

Team member	Late days remaining
Fisher Marks	1
–	N/A
–	N/A
Total pooled	<>
Team size	<>
Late days applied for the report	<>

2 Introduction

For the task of Shakespearean sonnet completion, a simple GPT-2 language model from pretrained weights can reliably produce Shakespearean sonnet-like text after standard supervised fine-tuning (SFT) on a small subset of Shakespeare’s corpus. However, the ABAB CDCD EFEF GG rhyme scheme that Shakespeare’s 154-sonnet corpus follows is does not reliable emerge under SFT on such a small amount of training data. So given a passable base model that can reproduce the Shakespearean style and vocabulary in a fluent way, we would like to be able to retain this ability while also picking up the ability to rhyme at the end of lines.

Direct Preference Optimization (DPO) [Rafailov et al., 2024] has provided a clean and effective solution for aligning language models to human preferences when using examples of preferred and nonpreferred responses. As such, this makes DPO an obvious candidate for aligning our model towards Shakespearean rhyme structure. Facing an absence of real rhyme-less Shakespearean data,

a standard approach would generate rhymeless completions from our reference model to use as nonpreferred responses. But because these generations will somewhat fail to be authentic in style, the broader preference signal for being more Shakespearean might dominate the token-level signal for rhyme. Traditional DPO has no means for focusing on learning a more localized preference over a broader stylistic preference.

To overcome the granularity mismatch between the DPO objective and our preference signal, we propose token-weighted DPO that assigns larger weights to tokens belonging to line-final rhyme words in hopes of providing a more targeted preference signal. We construct a pairwise dataset in which real Shakespearean lines are paired with rhymeless lines generated by a reference model, and evaluate whether token weightings can better align our policy. Our experiments find that although DPO improves CHRF scores over an SFT baseline by around 0.5 points, introducing token weights doesn't show a clear impact on either rhyme frequency or CHRF scores. Further analysis suggests that token weights alone are not enough to make the rhyme signal stronger than the errors of reference model generations.

3 Related Work

The foundational work for the idea of a pre-trained language model trained in an unsupervised manner before fine-tuning for specific tasks comes from the paper that introduced GPT-1 [Radford et al., 2019]. This work was the first to demonstrate that if you train a model on next-token prediction over a large amount of textual data, it can develop an understanding of language that makes it significantly easier to fine-tune for specific tasks. The weights our base model loads comes from this general next-token pre-training. The GPT-1 paper also outlines the decoder-only transformer architecture that is scaled up in the follow-up GPT-2 paper by Radford et al. [2019] which demonstrated that scaling up the decoder-only transformer architecture allows it to perform many tasks even without supervised fine-tuning. In our work we use a GPT-2 sized base model.

The DPO formulation we adapt in this work comes from the paper by Rafailov et al. [2024]. The DPO paper demonstrates that by making a simple assumption about the human preferences, we can derive a single maximum likelihood objective that bypasses an explicit reward model as a simplification of the traditional Reinforcement Learning from Human Feedback (RLHF) pipeline introduced by Ziegler et al. [2020]. Utilizing the DPO formulation not only simplifies the training pipeline, but also is more stable, less computationally expensive, and was shown by the authors of the original DPO paper to match or outperform the dominant language-alignment method at the time which relied on Proximal Policy Optimization [Schulman et al., 2017].

The inspiration our work in attempting to achieve alignment towards human preferences at a granularity below the sequence level comes from the work done by Yang et al. [2025] where they observe a discrepancy between the tendency of human preferences to be at the sequence level, while autoregressive training and generation are done at the token level. To address this and improve preference learning, they learn a token-level reward model to weight tokens and even apply this idea in their follow-up paper which relies on DPO [Yang et al., 2024]. Other papers learn token-level weights for DPO through contrastive models [Liu et al., 2025] and optimal transport [Li et al., 2025]. Our work positions itself as the only heuristically-labelled token weighting applied for DPO.

Previous works to align language models to poetic rhyme schemes focus on constrained decoding [Agnew et al., 2023] and model generation-revision Zugarini et al. [2021] rather learning rhyme alongside poetic style and language. A token-weighted DPO approach, if successful, would avoid pushing the model into a low probability space and maintains global coherence, potential limitations of constrained decoding and generation-revision respectively.

4 Approach

4.1 Base Model with Supervised Fine-Tuning

Our base model is a GPT-2 language model [Radford et al., 2019] initialized from pre-trained weights that we fine-tune on a set of 131 of Shakespeare's sonnets. The model generates sonnets autoregressively using top-p sampling [Holtzman et al., 2020] and softmax temperature. See the default project handout for full base model details.

Prompt (x)
 Thine eyes I love, and they, as pitying me,
 Knowing thy heart tormenting me with disdain,
 Have put on black, and loving mourners be,

Preferred Response (y^+)
 Looking with pretty ruth upon my pain.

Non-preferred Response (y^-)
 My daughter’s indulgence impresses the frown,

Figure 1: Example preference pair used for DPO training with prompt length 3. Prompts vary in length between 3 and 13 to provide training data for all parts of the continuation.

The modifications made to the supervised fine-tuning training regime outlined by the default project include a validation split to monitor overfitting alongside an early stopping condition and dropout to help regularize the model. Other than that, results for all three tasks submitted to the leaderboard come from extensive tuning of hyperparameters.

4.2 Rhyme Alignment with DPO Fine-Tuning

While the SFT-only model is able to roughly produce Shakespearean sonnet-like text, this regime fails to enforce the Shakespearean sonnet rhyme scheme at all. To promote the correct rhyme behavior, we further align our SFT-only model using DPO after undergoing supervised fine-tuning described in 4.1.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\log \sigma \left(\beta \left[\log \frac{\pi_\theta(y^+ | x)}{\pi_{\text{ref}}(y^+ | x)} - \log \frac{\pi_\theta(y^- | x)}{\pi_{\text{ref}}(y^- | x)} \right] \right) \right] \quad (1)$$

Under a Bradley-Terry assumption of human preferences, the DPO paper [Rafailov et al., 2024] outlines a DPO objective seen in Equation 1. While the original paper already gives a full derivation of how this assumption allows us to reach this objective, the main intuition is that for a policy π_θ , we want to increase the log-probability difference between preferred y^+ and nonpreferred responses y^- conditioned on a prompt x relative to a frozen reference policy π_{ref} .

We use our base model trained for Shakespearean sonnet generation as both our initial policy π_θ and the frozen reference policy π_{ref} . The idea behind the reference policy is that while we want to encourage rhyming, we don’t to drift too far away from generating good Shakespearean sonnets in the pursuit of always rhyming. β is a parameter that allows us to control the strength of this penalty for diverging too far from the reference model.

For thoroughness and for reference later, we can explicitly define the log probability of a policy’s response in Equation 2. The log probability of the response is the sum of the log probability of each of the tokens conditioned on the prompt and all previously generated tokens.

$$\log \pi_\theta(y | x) = \sum_{t=1}^T \log \pi_\theta(y_t | x, y_{<t}) \quad (2)$$

We create a dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-), \dots (x_n, y_n^+, y_n^-)\}$ made up of prompts with preferred and nonpreferred responses in such a way as described in Section 4.3.

4.3 Preference Pair Construction

An example of a preference pair generated for this work is shown in Figure 1. To create our prompts, we splice each Shakespearean sonnet in the training set to its first n lines for all $n \in [3, 13]$. This allows us to have examples for all stages of the sonnet completion starting from three lines. We use the next line following each partial sonnet as the preferred response y^+ .

Prompt	
Token:	Thine eyes I love, and they, as pitying me,
Weight:	0 0 0 0 0 0 0 0 0 0 0 0 0 0
Token:	Knowing thy heart tormenting me with disdain,
Weight:	0 0 0 0 0 0 0 0 0 0 0 0 0 0
Token:	Have put on black, and loving mourners be,
Weight:	0 0 0 0 0 0 0 0 0 0 0 0 0 0
Nonpreferred	
Token:	My daughter's indulgence impresses the frown,
Weight:	1 1 1 1 1 1 1 1 1 10 1

Figure 2: Tokenized negative example with weights applied for $w_r = 10$, $w_n = 1$. Note prompt tokens are masked for DPO objective.

Our nonpreferred response y^- comes from the SFT-only model. We present the SFT-only model with each of the partial Shakespearean sonnets and generate a one-line completion. When generating these nonpreferred examples, we discard any responses that rhyme correctly using Carnegie Mellon’s CMUDict [Carnegie Mellon University, 2014] so that we have a clear signal for preferring rhymes. Additionally, we discard any response with less than 5 words in order to present nontrivial preferences.

4.4 Token-Weighted DPO

We introduce an adaptation to the standard DPO formulation that applies token-level weights to the log probability of each response. The motivation for this modification is the difficulty of attributing token-level credit to sequence-level preferences in the DPO objective in Equation 1. Since we aim to train a policy that both preserves Shakespearean style and increases rhyme correctness, we could simply perform traditional DPO using nonpreferred responses from our reference model. But because our reference model isn’t authentic, a broader quality difference might overpower the more localized rhyme behavior. To address this, we introduce heuristically-labeled token weights that more heavily weight tokens belonging to line-final rhyme words, allowing for an objective that more clearly assigns credit to tokens responsible for the preference signal.

$$w \in \mathbb{R}^T \quad (3)$$

$$\log s_\pi(y|x) = \frac{1}{T} \sum_{t=1}^T w_t \log \pi(y_t|x, y_{<t}) \quad (4)$$

$$\mathcal{L}_{\text{TWDPDPO}(\pi_\theta; \pi_{ref})} = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\log \sigma \left(\beta \left[\log \frac{s_{\pi_\theta}(y^+|x)}{s_{\pi_{ref}}(y^+|x)} - \log \frac{s_{\pi_\theta}(y^-|x)}{s_{\pi_{ref}}(y^-|x)} \right] \right) \right] \quad (5)$$

We introduce a weight vector $w \in \mathbb{R}^T$ where each token gets a weight w_r if it is a token part of a rhyme word, and w_n if it is a token part of a non-rhyme word. The implementation details of marking these special tokens just involve a heuristic encoding the last word of the line alongside whitespace and punctuation and counting out the length of the token sequence starting from the back and excluding the punctuation token. Manual check over 100 sequences saw a 100% accuracy in marking the intended tokens. We normalize the weights by dividing by the number of tokens in the response in order to prevent inadvertently rewarding line-final words with large token counts.

To further reason about why this modification might be effective, we can observe the gradient of the weighted scores in Equation 6. Tokens with a larger weight produce larger parameter updates, increasing the probability of the rhyme tokens in preferred lines while decreasing the probability of non-rhyme alternatives.

$$\nabla \log s_\pi(y|x) = \frac{1}{T} \sum_{t=1}^T w_t \nabla \log \pi(y_t|x, y_{<t}) \quad (6)$$

4.5 Theoretical Justification

Just as the original DPO paper demonstrates its objective is a maximum likelihood estimator of a Bradley-Terry model, we can also observe that this modification remains the negative log-likelihood of a Bradley-Terry preference model.

$$P(y^+ \succ y^- | x) = \sigma(r(x, y^+) - r(x, y^-)) \quad (7)$$

The Bradley-Terry assumption shown in Equation 7 requires that preferences can be modeled as the logistic of the difference between two rewards for some $r(x, y) \in \mathbb{R}$. The original DPO formulation uses $r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$, while we substitute the log probabilities with weighted log probabilities as shown in Equations 4 and 5. From here, there is no difference between the original DPO paper’s derivation and our own justification for our new objective as a maximum likelihood estimator. Our modification preserves the same relation.

5 Experiments

5.1 Data

Our work relies on four datasets, three for validation of the GPT-2 implementation (Stanford Sentiment Treebank [Socher et al., 2013], CFIMDB [Pang and Lee, 2004], Quora Question Pairs [Quora, Inc., 2017]) and one for the primary evaluation of our token-weighted DPO method (Shakespeare’s sonnet corpus [Shakespeare, 2015]).

The Stanford Sentiment Treebank and CFIMDB datasets contain 11,855 and 2,434 excerpts respectively from movie reviews, and our task is to predict one label out of set of sentiment labels. The Quora Question Pairs dataset compares 400,000 question pairs with our task being to predict a binary label describing whether the two questions are paraphrases of the other. The Shakespeare sonnet corpus consists of Shakespeare’s 154 sonnets, used for fine-tuning our language model for completion of a sonnet given the first 3 lines.

5.2 Evaluation Method

In order to evaluate the sentiment analysis and paraphrase detection tasks to validate our GPT-2 implementation, we will rely on simply label accuracy.

For the sonnet generation task, we will rely on both CHRF score as a measure of overall sonnet quality and a "rhyme accuracy" to measure the effect of our token-weighted DPO approach on improving the rhyme structure of our generated sonnets. Rhyme accuracy will be measured as the number of line-final words that rhyme as they should in the Shakespearean sonnet structure. Not including the rhyme provided by the 3 line prompt, we will check the other 6 pairs of line-final words for accuracy. To evaluate rhyme accuracy, we will rely on CMUDict’s pronunciation library Carnegie Mellon University [2014] to check pairs of words for rhyme. Because Shakespeare’s sonnets use an older form of English with differences in modern pronunciation, we will also manually inspect our generated sonnets for any missed rhymes. For example, Shakespeare uses "west" as a rhyme for "east" despite modern pronunciation being different.

5.3 Experimental details

For the sentiment analysis tasks, we make no modification to the default experimental setup. For the paraphrase detection task, we use the gpt2 size model, training over 3 epochs with a learning rate of $1e-5$, a batch size of 8, and additional dropout of 0.1 on the output layer for regularization. Each epoch takes approximately 50 minutes, for a total training time of approximately 2 and a half hours.

For the sonnet generation task, we create our base model by performing SFT on our GPT2 implementation. We use a softmax temperature of 1.2, a top-p of 0.9, a batch size of 8, and a learning rate of $3e-5$ to train the gpt2 sized model over 3 epochs. Each epoch takes approximately 3 seconds, for a total training time of less than 10 seconds.

We then perform our token-weighted DPO approach on the base model having already undergone the SFT regime as described above. We use a softmax temperature of 1.2, a top-p of 0.9, a batch size of

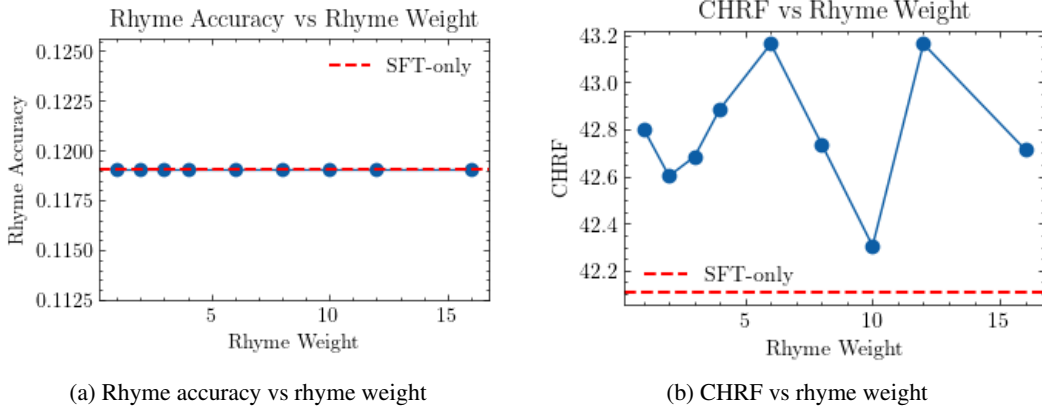


Figure 3: Rhyme accuracy and CHRF scores in relation to w_r on the Dev set.

8, and a learning rate of $3e-5$ to train the gpt2 sized model over 3 epochs. Each DPO epoch still only takes approximately 3 seconds, for a total training time of less than 10 seconds. We found that a β (DPO parameter for controlling the divergence from reference model) of 0.1 saw the best results and different values had no impact on the trends discussed in the Section 5.4 where we explore the impact of different rhyme weights w_r .

5.4 Results

For the sentiment analysis and paraphrase detection experiments, we were able to achieve scores than the staff baselines for both Stanford Sentiment Treebank (SST) and CFIMDB as shown in Table 1.

Table 1: Sentiment and Paraphrase Results (Test Set)

Experiment	Accuracy (%)	Staff Baseline (%)
SST Last-Linear	46.7	35.0
SST Full Model	53.6	45.0
CFIMDB Full Model	97.1	90.0
Quora Full Model	85.7	85.6

Because our token-weighted DPO extension did not focus on these tasks and improvements were only made by tuning hyperparameters or adding additional dropout, achieving scores slightly above the baseline by was expected and allowed us to confirm the effectiveness of our GPT2 implementation.

Table 2: Sonnet Results (Test Set)

Experiment	CHRF	Rhyme Acc (%)
Staff Baseline	41.208	—
SFT Model	42.004	11.9%
DPO ($w_r = 1$)	42.437	11.9%
TW-DPO ($w_r = 6$)	42.496	11.9%

Figure 3 and Table 2 demonstrate the results of our experiments for token-weighted DPO on the sonnet completion task. Despite our intuition that increasing token-weights for tokens making up the final word of each line, we found that the rhyme accuracy does not increase across any of our test rhyme weights $w_r \in \{1, 2, 3, 4, 6, 8, 10, 12, 16\}$.

While rhyme accuracy itself doesn’t increase, we did find that both traditional DPO and token-weighted DPO outperform both the staff baseline results and the SFT-only model. Figure 3b demonstrates that any form of DPO consistently outperforms the SFT-only baseline.

While we were able to achieve the highest results using our token-weighted DPO approach, Figure 3a demonstrates that varying the rhyme weight w_r is more like noise than an actual improvement and traditional DPO with $w_r = 1$ performs on average just as well.

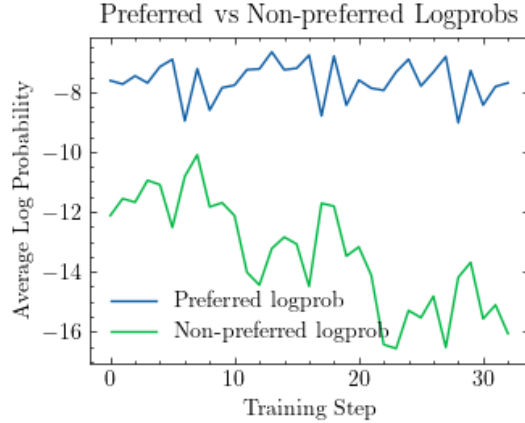


Figure 4: Log Probability Margin over DPO Training ($w_r = 6$)

6 Analysis

To dive deeper into why our token-weighted DPO might not be having its expected impact, we can observe that by plotting out the log probabilities of preferred and nonpreferred responses during DPO training as shown in Figure 4, we can see that our policy seems to only get better at separating responses by learning which responses are bad while the probability of preferred responses remains roughly the same.

Examining a pair of completions from the SFT-only model and the best token-weighted DPO model, we can qualitatively see why this is happening: the policy appears to improve primarily by avoiding low-quality continuations rather than by learning the rhyme preference signal.

SFT-only:

So, now I have confessed that he is thine
And I myself am mortgaged to thy will,
Myself I'll forfeit, so that other mine wont have
mercy on
Like to silent cruel pointers not.
Wead all A gull mee, sorrowfully locked off by
time,
By him mistaken words measured what thou
wilt bore,
For, pity, but his ex(Qawnblad is still root').
Should old time enfruit fief he his new own
beast pride?
My \$10 let keep its merit with worth of virtue
E' t doth written home up by his eyes that can
work
Lured away by myself past illness, out o'en
silence,
E' like such action

TW-DPO:

So, now I have confessed that he is thine
And I myself am mortgaged to thy will,
Myself I'll forfeit, so that other mine
That paid the due fee of my owner's interest put
In theft's lawful care;
My crown, being invested with commodities to
pay,
Blowed to him that put his esteem in me,
That show me far better that I do not spoil this
wretched spoils of th' prize.
Beyond all this, I will make restitution of those I
stole,
For either to pay what is fair and equal to man's
sole worth;
That while there can be peace among buyers in
the sea,
That the true bargain makes all profits to his
friend,
Which the true thief may rob most monopolists
of there beloved,
And truly hold true his bounty to himself forever
by stealing.

Towards the end of the completion, the SFT-only model starts to make many obvious mistakes. The rhyme signal during DPO is likely to be muddled by the signal of not making these types of mistakes, so it is difficult for the policy to see any improvement in rhyme accuracy using these nonpreferred examples.

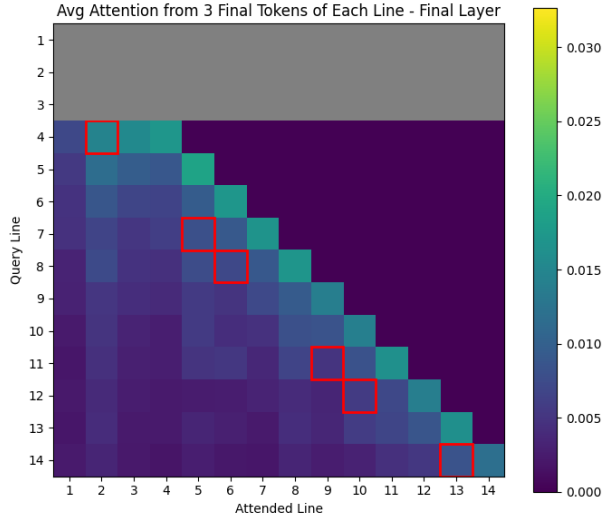


Figure 5: Attention heatmap displaying how the last 3 tokens of each line attend to each other in the final layer. Boxes in red highlight where we’d expect earlier rhyme words to influence future rhymes.

A further analysis of the attention distribution of the last three tokens of the final layer of our GPT-2 model in Figure 5 help confirm this suspicion. Boxes in red highlight where previous rhyme words should be attending to their future counterparts, but there is no noticeable peak in attention that might suggest the model is learning it should rhyme with these previous words.

This speaks to a potential limitation to the approach in terms of the available training data. While we had hoped to use SFT-only completions as Shakespearean-like examples that didn’t include rhymes, clearly they are not high-quality enough for the policy to pick up on the signal for the intended rhyming behavior. For further evaluation of this approach on other tasks, we will focus on finding a setting for which we can completely isolate a token-level preference signal.

7 Conclusion

In this work we propose an adaptation to traditional DPO, using heuristically-decided token-weights to modify the traditional DPO objective in the hopes of promoting rhyming behavior in a GPT2-style model trained for Shakespearean sonnet generation. While we demonstrated the theoretical soundness of the approach, experimental results found no clear improvement over traditional DPO. Further analysis speaks to the difficulty of the tasks given limitations in available data. The gap in quality between the reference model and authentic Shakespeare is too wide for token-level weights to boost the rhyme signal enough to increase rhyme accuracy. Limitations in mind, we hope to perform future experiments to evaluate the effectiveness of this method on other tasks that do not suffer from the same issues. One such avenue could be to focus on poetry outside of the Shakespearean corpus which both is greater in volume and includes language that might be more in-distribution.

8 Acknowledgments

Thanks to John Garret for his SciencePlots python package for pyplot styling he made publicly available for the use in projects and papers. [Garrett, 2021]. I treated AI as a collaborator in this project for things like latex and plot formatting, and to err on the side of caution I added several citations within my code in a couple places for when I thought I might have received more than completely insignificant assistance. All code for the extension was done myself except for the external libraries mentioned in this report, CMUDict for rhyme checking and SciencePlots for plot styling.

References

- 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, 2024. URL <https://arxiv.org/abs/2305.18290>.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *arXiv preprint arXiv:1901.11373*, 2019.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL <https://arxiv.org/abs/1909.08593>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Shentao Yang, Shujian Zhang, Congying Xia, Yihao Feng, Caiming Xiong, and Mingyuan Zhou. Preference-grounded token-level guidance for language model fine-tuning, 2025. URL <https://arxiv.org/abs/2306.00398>.
- Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference, 2024. URL <https://arxiv.org/abs/2402.08265>.
- Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Xiaoming Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip S. Yu, and Meng Cao. TIS-DPO: Token-level importance sampling for direct preference optimization with estimated weights. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=oF6e2WwxX0>.
- Meng Li, Guangda Huzhang, Haibo Zhang, Xiting Wang, and Anxiang Zeng. Optimal transport-based token weighting scheme for enhanced preference optimization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21311–21334, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1035. URL <https://aclanthology.org/2025.acl-long.1035/>.
- Edwin Agnew, Michelle Qiu, Lily Zhu, Sam Wiseman, and Cynthia Rudin. The mechanical bard: An interpretable machine learning approach to Shakespearean sonnet generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1627–1638, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.140. URL <https://aclanthology.org/2023.acl-short.140/>.
- Andrea Zugarini, Luca Pasqualini, Stefano Melacci, and Marco Maggini. Generate and revise: Reinforcement learning in neural poetry, 2021. URL <https://arxiv.org/abs/2102.04114>.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL <https://arxiv.org/abs/1904.09751>.
- Carnegie Mellon University. The cmu pronouncing dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>, 2014. Version 0.7b.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Association for Computational Linguistics, 2013.
- Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278, 2004.
- Quora, Inc. Quora question pairs. <https://kaggle.com/competitions/quora-question-pairs>, 2017. Kaggle dataset.

William Shakespeare. Shakespeare's sonnets. <https://shakespeare.folger.edu/shakespeares-works/shakespeares-sonnets/>, 2015. Created July 31, 2015 from FDT version 0.9.0.1.

John D. Garrett. garrettj403/SciencePlots. September 2021. doi: 10.5281/zenodo.4106649. URL <http://doi.org/10.5281/zenodo.4106649>.