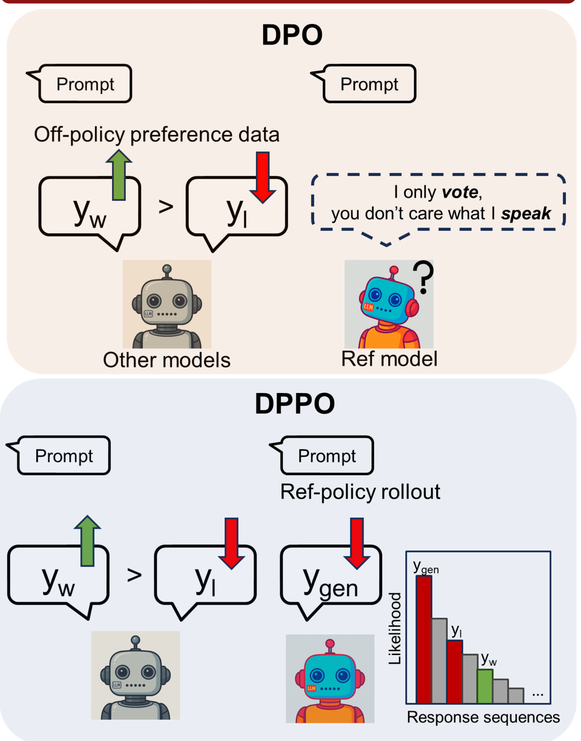
# DPPO: Direct Preference and Penalization Optimization

*Pengwei Sun*

*Program of Biomedical Physics, Stanford University*

Stanford Computer Science

## Project Overview




## Datasets & Metrics

- SFT dataset: SmolTalk [1], a collection of high-quality chat responses from GPT-4o. We use one-turn conversation between "user" and "assistant". The context token length is truncated to 576, which is the 95th percentile of token length in the training set.
- DPO dataset: Ultrafeedback [2], a preference dataset to study the instruction following abilities of LLMs. We truncate the context token length to 989, the 95th percentile.
- Evaluation: we use a parametric reward model for scoring with the Llama 3.1 Nemotron 70B Reward Model [3]. The prompts and responses are constructed as a chat template. Nemotron generates a reward score for both reference model (our SFT model) and the evaluated model. We report the win rate across the evaluation set where "win" means the reward of the evaluated model is higher than the reference model.

## Methods & Experiments

- For SFT, we use the next-token prediction loss with mask on the queries. For vanilla DPO, we use the same preference classification loss as described in [4].

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log \pi_\theta(y_t|y_{<t}, x) \qquad \mathcal{L}_{\text{DPO}} = -\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)})$$

- When the model fails to vote the winning sequence, the most likely response from the reference model is penalized. The response is generated as below, which is collected before the training to avoid unacceptable-long inference time for on-policy rollouts while maintains **policy-specific** penalization.

$$y_{\text{gen}} = \arg\max_y \pi_{\text{ref}}(y|x)$$

- **1. Minimum likelihood** The loss function is tailored to minimize the likelihood of reference response, with a reward-weighted function to control the penalty strength based on *r*.

$$\mathcal{L}_{\text{minll}} = \mathcal{L}_{\text{DPO}} + \alpha \cdot f(r) \log \pi_\theta(y_{\text{gen}}|x) \qquad r = \log \frac{\pi(y_w|x)}{\pi(y_l|x)}$$

- **2. Unlikelihood**

$$\mathcal{L}_{\text{unll}} = \mathcal{L}_{\text{DPO}} - \alpha \cdot f(r) \log(1 - \pi_\theta(y_{\text{gen}}|x))$$

$$f(r) = \begin{cases} -\mathbb{1}\{r<0\}r & \text{linear} \\ \mathbb{1}\{r<0\}r^2 & \text{quadratic} \\ \mathbb{1}\{r<0\}(-r)^{0.5} & \text{square root} \\ (\zeta - \mathbb{1}\{r<0\})r^2 & \text{expectile} \end{cases}$$

- **3. Contrastive likelihood**

$$\mathcal{L}_{\text{conll}} = -\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \alpha\beta \cdot f(r) \log \frac{\pi_\theta(y_{\text{gen}}|x)}{\pi_{\text{ref}}(y_{\text{gen}}|x)})$$

## Discussions & Reference

### Discussions:

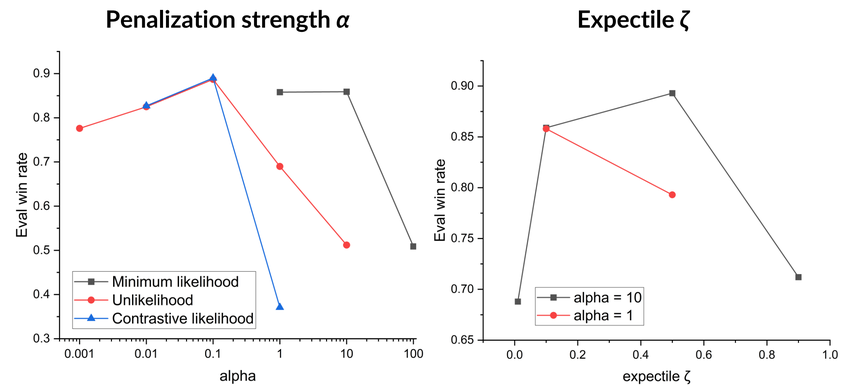
- Choice of penalization strength $\alpha$ is essential to balance between the DPO and penalization loss.
- Negative rewards significantly decrease compared to DPO, which proves better preference optimization.
- Unlikelihood trained with 8k samples yields higher win rate compared to DPO trained with 60k samples. Unlikelihood training with 60k samples + DPO post-training with 8k samples yields highest win rate (0.977).

### References

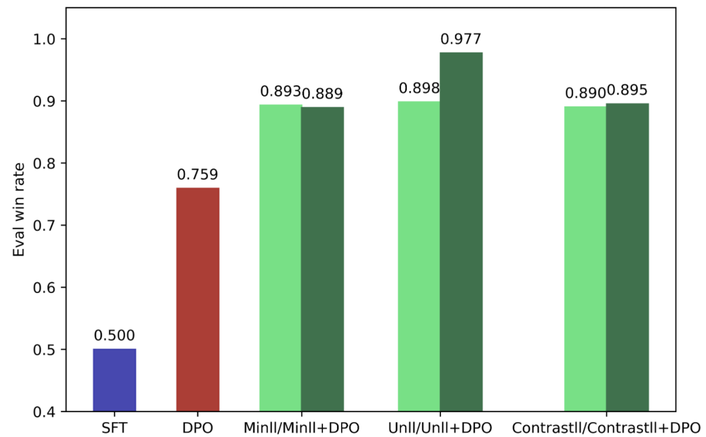
[1] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlí˘cek, Agustín Piqueres Lajarín, Vaibhav Srivastav,Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, HugoLarcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, andThomas Wolf. 2025. SmolLM2: When Smol Goes Big – Data-Centric Training of a SmallLanguage Model. arXiv:2502.02737 [cs.CL]
[2] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. AlpacaFarm: A Simulation Framework forMethods that Learn from Human Feedback. arXiv:2305.14387 [cs.LG]
[3] https://huggingface.co/nvidia/Llama-3. 1-Nemotron-70B-Reward
[4] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model.arXiv:2305.18290 [cs.LG]

## Results

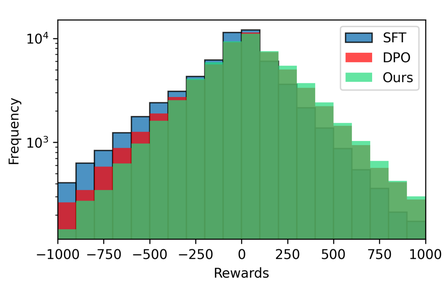
### Hyperparameter search




### Performance of different training paradigms




### Reward histogram




### Data Efficiency