AI HW6

R12922029 陳姵安

1. Provide a brief description and comparison of DPO and ORPO. (5%+5%)

a. DPO

Direct Preference Optimization (DPO) offers a novel approach to fine-tuning large language models (LLMs) by directly incorporating human feedback into a simple classification loss, bypassing the complexities of reinforcement learning typically used in Reinforcement Learning from Human Feedback (RLHF). DPO simplifies the process by using a new parameterization of the reward model in RLHF, allowing for extraction of optimal policies in closed form. DPO eliminates the need for sampling during fine-tuning and extensive hyperparameter tuning, making it more stable and computationally efficient while maintaining or enhancing performance in tasks like sentiment control and response quality in summarization and dialogue. The figure shown below is a comparison flowchart of RLHF and DPO.

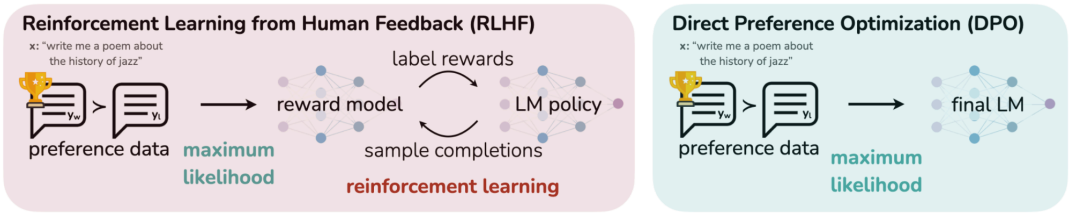


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

b. ORPO

Odds Ratio Preference Optimization (ORPO) advances supervised fine-tuning (SFT) of large language models (LLMs) by utilizing an odds ratio to differentiate favored and disfavored generation styles directly, without the need for a separate preference alignment phase. ORPO simplifies the model training process by avoiding the complexities typically associated with reinforcement learning from human feedback (RLHF) and Direct Preference Optimization (DPO). ORPO demonstrates significant performance improvements across various model scales and tasks, highlighting its efficacy and scalability in aligning model outputs with specific preferences, surpassing previous methods in benchmark evaluations. The figure shown below is a comparison flowchart of RLHF, DPO, and ORPO.

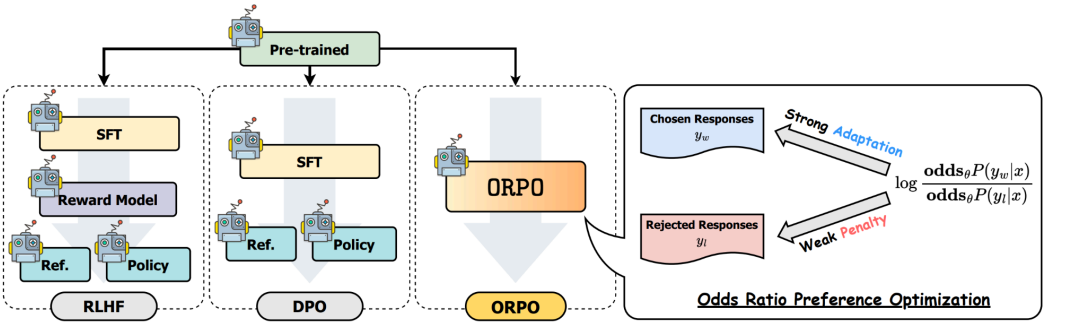


Figure 2: Comparison of model alignment techniques. ORPO aligns the language model *without a reference model* in a single-step manner by assigning a weak penalty to the rejected responses and a strong adaptation signal to the chosen responses with a simple log odds ratio term appended to the negative log-likelihood loss.

c. comparison of DPO and ORPO

Direct Preference Optimization (DPO) and Odds Ratio Preference Optimization (ORPO) are both methodologies designed to fine-tune large language models (LLMs), but they adopt different approaches and have their unique strengths and limitations in implementation and performance. Here's a detailed comparison of DPO and ORPO based on their implementation and characteristics:

i. Implementation Differences

DPO modifies the traditional Reinforcement Learning from Human Feedback (RLHF) process by incorporating human feedback directly into a classification loss, facilitating the derivation of optimal policies more straightforwardly. It uses a new parameterization of the reward model within RLHF, which simplifies the extraction and application of these policies without complex reinforcement learning loops. Additionally, DPO reduces the need for extensive hyperparameter tuning and eliminates sampling during training, enhancing stability and computational efficiency. In contrast, ORPO employs an odds ratio to directly differentiate between favored and disfavored generation styles, removing the need for a separate preference alignment phase. It simplifies the training process by not relying on the typical mechanisms of RLHF or the parameter tweaks of DPO, focusing directly on optimizing the model's output according to predefined preference metrics. ORPO also streamlines model training by modifying fewer parameters, which is particularly beneficial for quick adaptations to new tasks or domains.

ii. Characteristics

Characteristics	DPO	ORPO
Integration Approach	Incorporates human feedback directly into a classification loss within RLHF.	Utilizes an odds ratio to directly differentiate between favored and disfavored styles.
Complexity Reduction	Simplifies RLHF by eliminating the need for complex reinforcement learning loops.	Avoids the complexities of traditional RLHF and DPO, focusing on direct preference handling.
Computational Efficiency	Reduces hyperparameter tuning and eliminates sampling, enhancing stability and efficiency.	Streamlines training by modifying fewer parameters, beneficial for rapid task adaptations.
Performance Focus	Enhances performance in sentiment control and quality response in summarization and dialogue.	Highly effective for tasks requiring clear distinctions in output styles.
Dependency on Feedback	Heavily reliant on effective integration of human feedback, which may miss complex nuances.	Less dependent on the subtleties of feedback integration compared to DPO.
Training Simplicity	Offers a straightforward approach by directly integrating feedback into a loss function.	Simplifies the training process by focusing on optimizing outputs with predefined metrics.
Potential Risks	Dependency on classification loss may not effectively capture all desired nuances.	Potential over-simplification may not be suitable for complex tasks requiring deep learning from diverse inputs.

2. Briefly describe LoRA. (5%)

The main innovation of LoRA lies in the introduction of **trainable low-rank decomposition matrices**. These matrices are **inserted into each layer of the Transformer architecture, significantly reducing the number of trainable parameters for downstream tasks**. Compared to fully fine-tuning GPT-3 175B using Adam, LoRA can reduce the number of trainable parameters by 10,000 times and reduce GPU memory requirements by threefold. In terms of model quality, LoRA performs on par with or better than full fine-tuning on models such as RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters, higher training throughput, and unlike adapters, it does not introduce additional inference latency. Moreover, the article[c] also conducts an empirical investigation into rank deficiency in language model adaptation, further elucidating the effectiveness of LoRA.

To sum up, LoRA (Low-Rank Adaptation) is a method specifically designed to **adapt LLM to specific tasks or domains without the need to retrain all model parameters**. This method is particularly valuable as it addresses the high costs and resource-intensive challenges commonly faced when deploying large-scale models like GPT-3 175B to different applications.

3. Plot your training curve by W&B, including both loss and rewards. (5%)

4. Comparison and analysis of results (before & after DPO & after ORPO) (5%)

a. General analysis

i. Before both DPO and ORPO:
The responses tend to be repetitive, less precise, and do not fully adhere to the informational needs of the query. They frequently misinterpret the questions or provide unnecessarily repeated content.

ii. After DPO:
There is noticeable improvement in task-specific adjustments like framing and relevance of responses, but some issues like redundancy and lack of completion persist.

iii. After ORPO:
Responses are notably more concise and on point. ORPO seems to optimize for relevance and accuracy more effectively compared to DPO.

b. Detailed Analysis

i. Question1: "How many colors are traditionally recognized in a visible spectrum or optical rainbow?"

1. Before: The answer is embedded in a lengthy, repetitive format with unnecessary details.
2. After DPO: The response is structured similarly to the "Before" state, with some inaccuracies in descriptions.
3. After ORPO: Significantly more concise and accurate, clearly stating that the visible spectrum consists of 7 colors.

ii. Question2: "In a basket, what would be the new ratio of oranges to other fruits after changes?"

1. Before: The generated text repeats the question without providing a numerical ratio.
2. After DPO: Attempts a numerical response but it is incorrect and not in proportion to the described scenario.
3. After ORPO: Provides a clear and numerically accurate ratio, reflecting better understanding and computation.

iii. **Question3: "How would you convince a buyer to purchase an electric vehicle?"**
    1. Before: The response is extremely repetitive, circling around non-essential points.
    2. After DPO: Improved but still contains filler content; does not directly address the question's intent.
    3. After ORPO: The response is significantly more targeted and relevant, focusing on environmental benefits, which is a strong selling point for electric vehicles.

iv. **Question4: "Break down the process of photosynthesis."**
    1. Before, After DPO, and After ORPO: All show minimal to no real content, suggesting a challenge in generating scientifically detailed responses across methods.

v. **Question5: "Explain why college students should get a library card."**
    1. Before: Repeat the same non-informative phrase.
    2. After DPO and ORPO: Both continue the repetition with a slightly improved phrase; however, it remains largely uninformative.

vi. **Question6: "What popular toy was invented in 1943 by Richard James?"**
    1. Before: The response is extremely repetitive and lacks coherence, repeating the phrase "Talking Talking" indefinitely.
    2. After DPO: No improvement is observed; the response continues the same repetitive and irrelevant pattern.
    3. After ORPO: Unfortunately, the issue remains unaddressed, with the response still repeating irrelevant content, showing that both optimization methods struggle with generating contextually relevant and diverse responses for this query.

vii. **Question7: "Which famous movie prop designs have become iconic symbols in film history?"**
    1. Before: Provides a list of movies associated with the "Bugatti", which is incorrect and not related to movie prop designs.
    2. After DPO: Again, lists irrelevant connections to movies and the "Bugatti", showing no real improvement or understanding of the question.
    3. After ORPO: Attempts a more diverse list but still incorrectly focuses on the "Bugatti" from various movies, indicating a persistent misunderstanding of the term "movie prop."

viii. **Question8: "What is the minimum coefficient of static friction needed to prevent skidding?"**
    1. Before: The response fails dramatically, giving an absurdly long string of zeros, which is practically meaningless.
    2. After DPO: This issue persists with no correction or improvement, maintaining the same irrelevant output.
    3. After ORPO: The response remains incorrect, continuing to give a string of zeros, indicating that neither DPO nor ORPO effectively address numerical precision in scientific contexts.

ix. **Question9: "What can't you see, hear or feel, until it's too late?"**

1. Before: Provides a nonsensical response that simply repeats elements of the question, adding no value or insight.
2. After DPO: Little to no improvement is observed, with the response echoing the input without offering a substantive answer.
3. After ORPO: Similarly, fails to improve, suggesting that both optimization techniques struggle with abstract or metaphorical queries that require interpretative responses.

x. **Question10: "In a basket, if 15 pears were added and half of the oranges were removed, what would be the new ratio of fruits?"**

1. Before: Fails to calculate the correct ratio, instead repeating parts of the question.
2. After DPO: Suggests an incorrect and repeated ratio that does not reflect the changes described in the prompt.
3. After ORPO: Like DPO, repeats an incorrect ratio without accurately accounting for the described changes in fruit quantities.

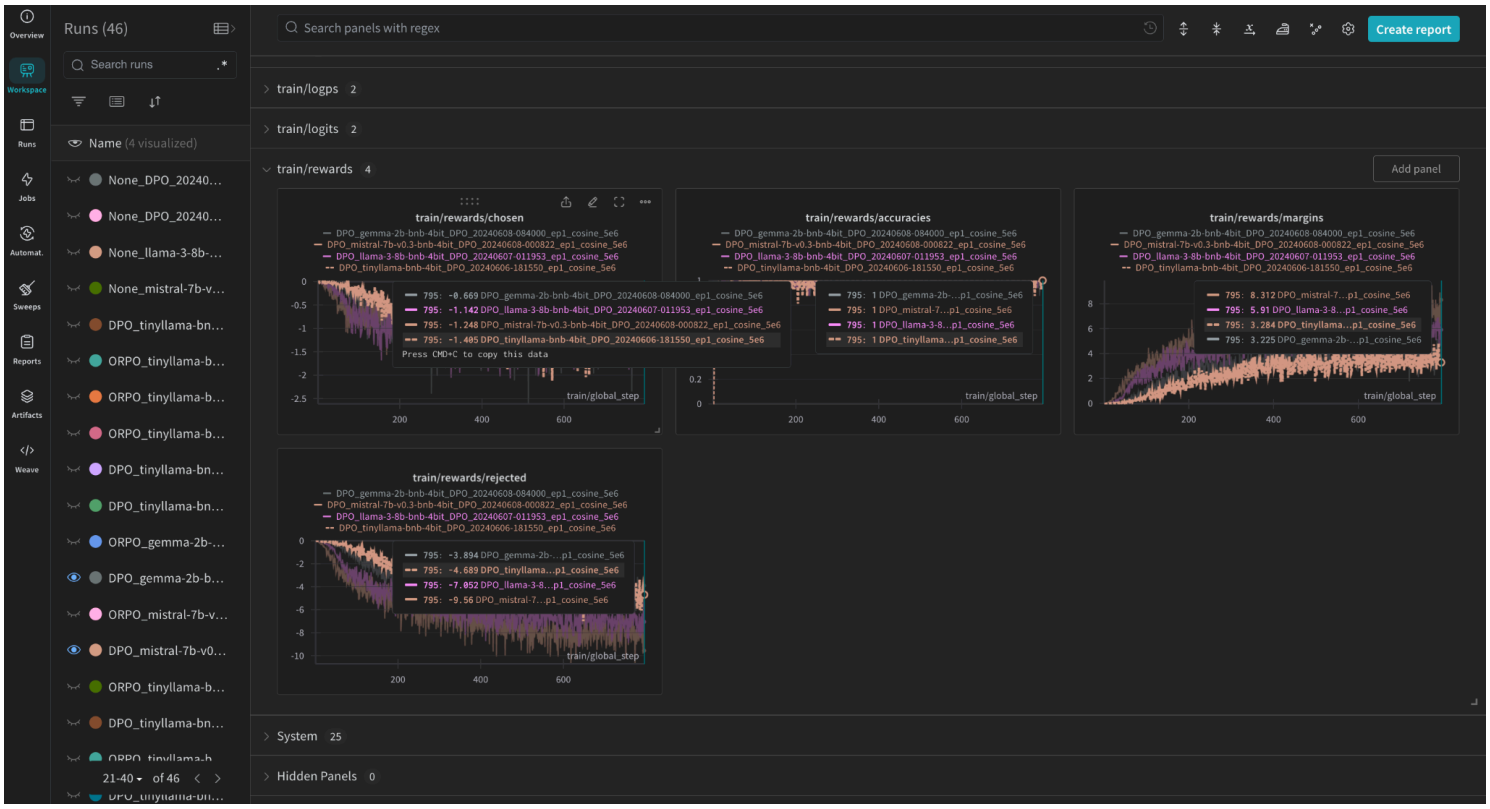
c. Conclusion

The ORPO method shows superior performance in generating relevant, concise, and contextually accurate responses compared to DPO and the initial state. ORPO's strength lies in its ability to sharply focus and refine responses, likely due to its more direct handling of preference optimization without the intermediate complexities that DPO still partially involves. Thus, for applications requiring high precision and minimal response latency, ORPO appears more suitable, while DPO could be considered when transitioning from less optimized methods, providing a stepwise improvement in model response quality and operational efficiency.

5. Extra Experiments

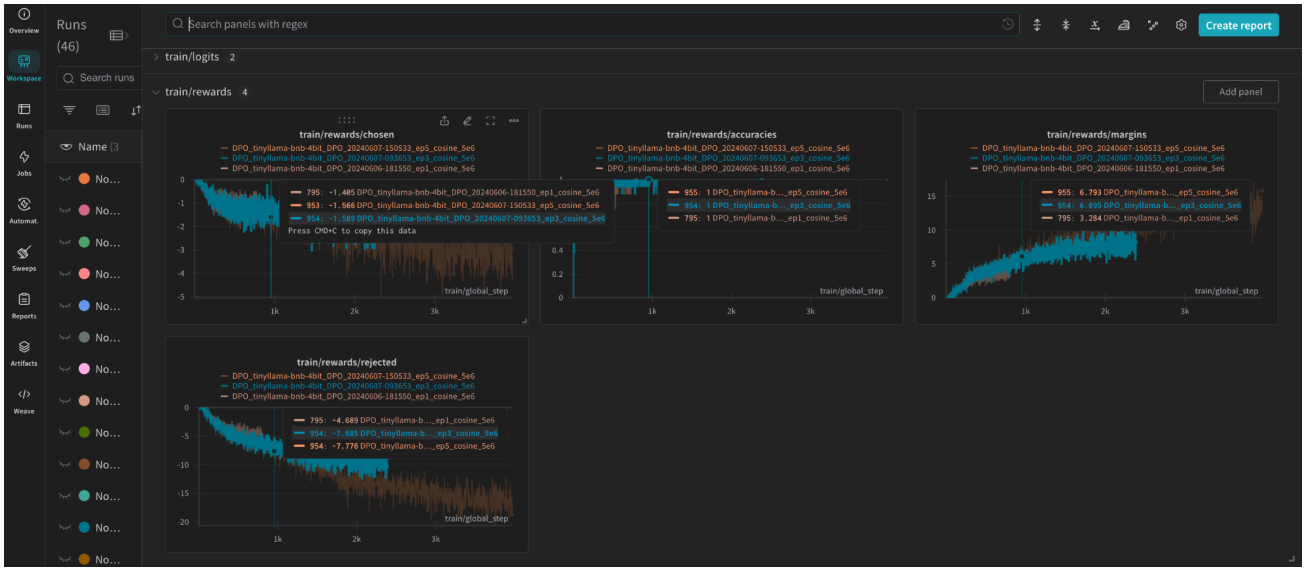
a. After DPO

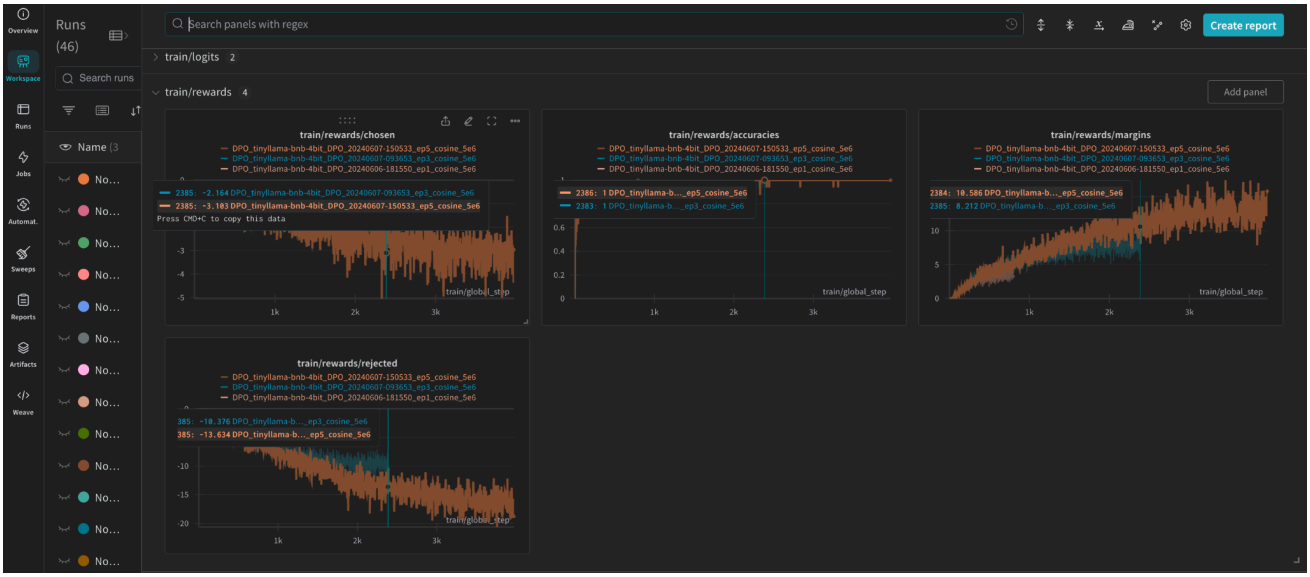
i. Different model



No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	llama-3-8b-bnb-4bit	cosine	1	0.1	-1.142	1	5.91	-7.052
2	5e-6	mistral-7b-v0.3-bnb-4bit	cosine	1	0.1	-1.248	1	8.312	-9.56
3	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-1.405	1	3.284	-4.689
4	5e-6	gemma-2b-bnb-4bit	cosine	1	0.1	-0.669	1	3.225	-3.894

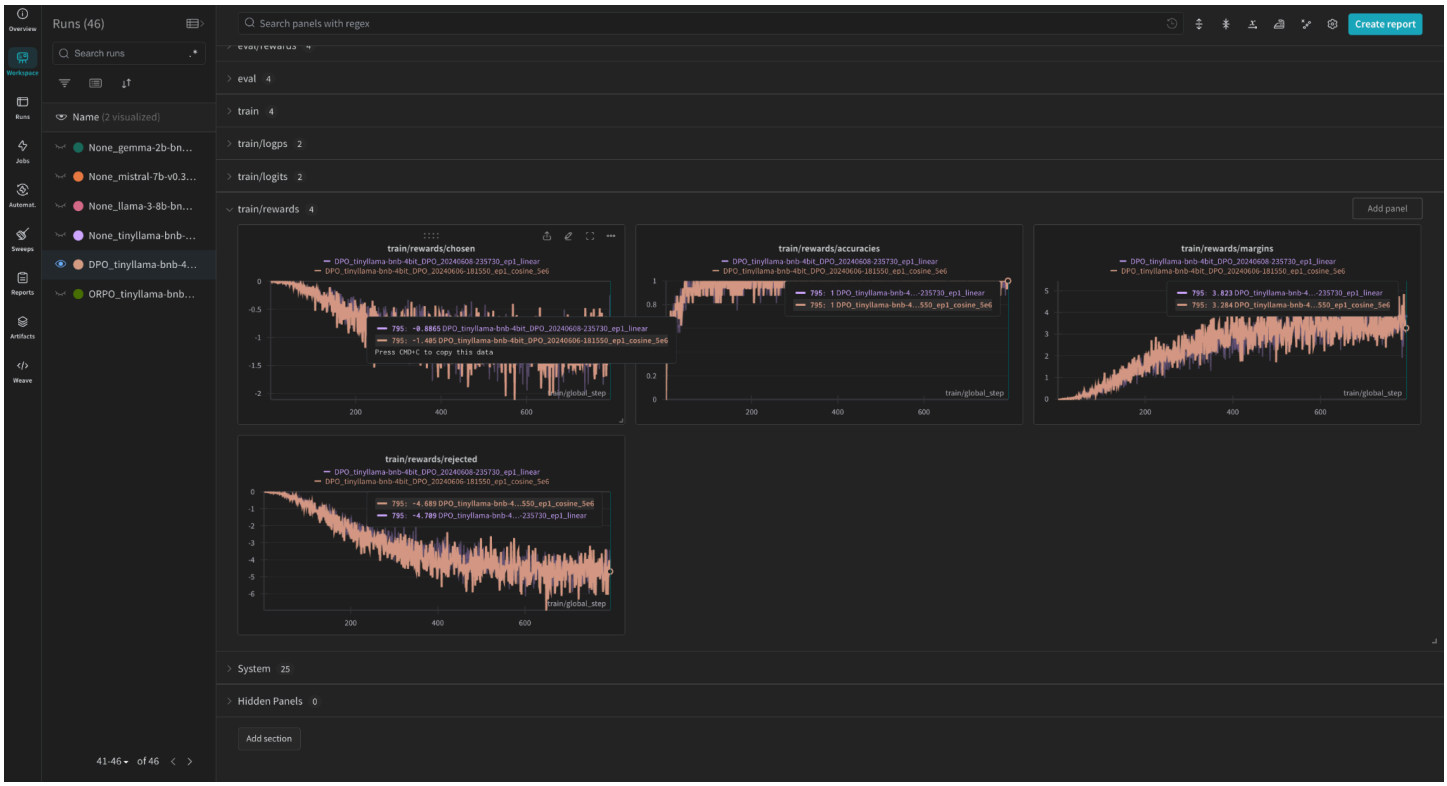
ii. Different number of epoch (last step : 795/2385/3974)





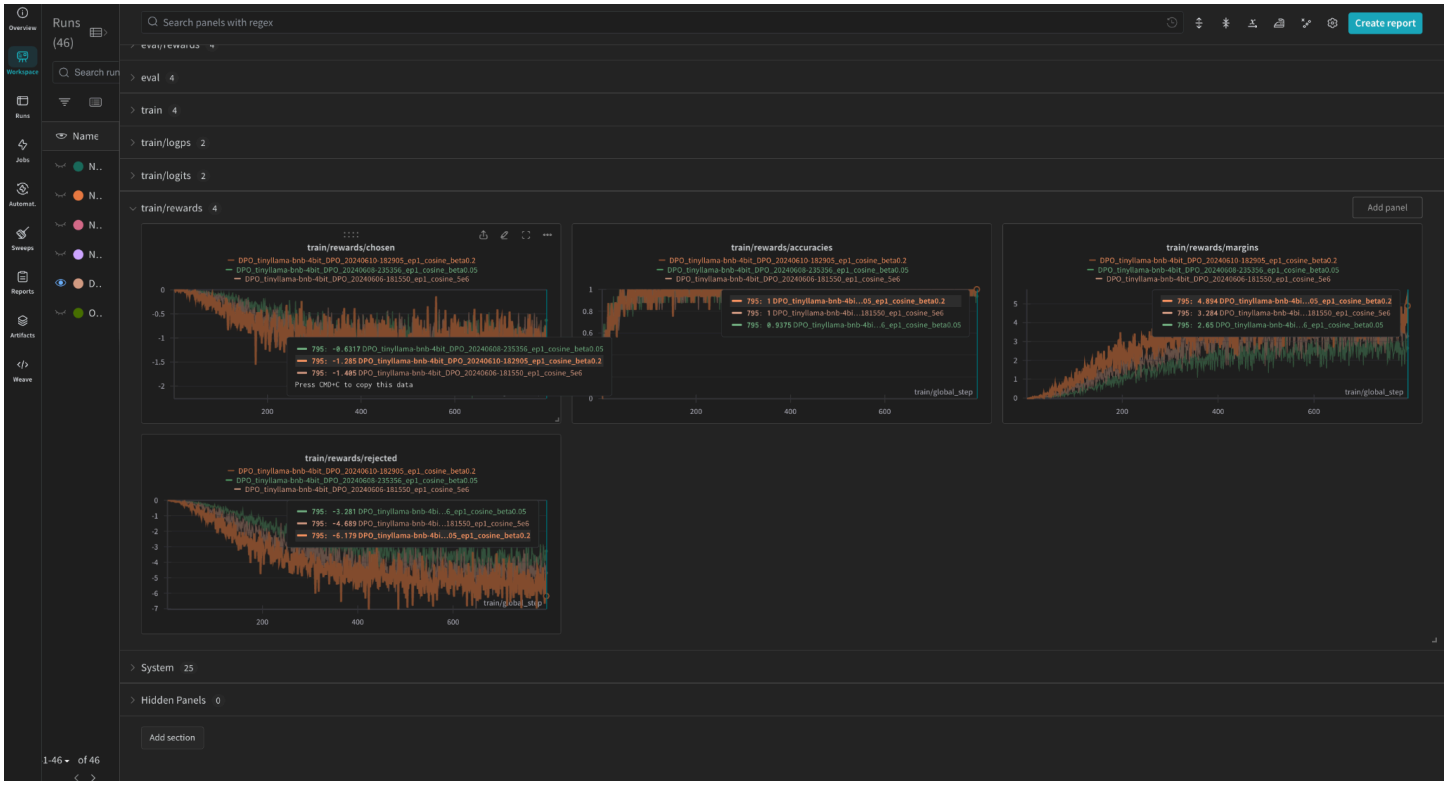
No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-1.405	1	3.284	-4.689
2	5e-6	tinyllama-bnb-4bit	cosine	3	0.1	-2.164	1	8.212	-10.376
3	5e-6	tinyllama-bnb-4bit	cosine	5	0.1	-2.965	1	15.806	-18.771

iii. Different learning rate scheduler type



No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-1.405	1	3.284	-4.709
2	5e-6	tinyllama-bnb-4bit	linear	1	0.1	-0.8865	1	3.823	-4.689

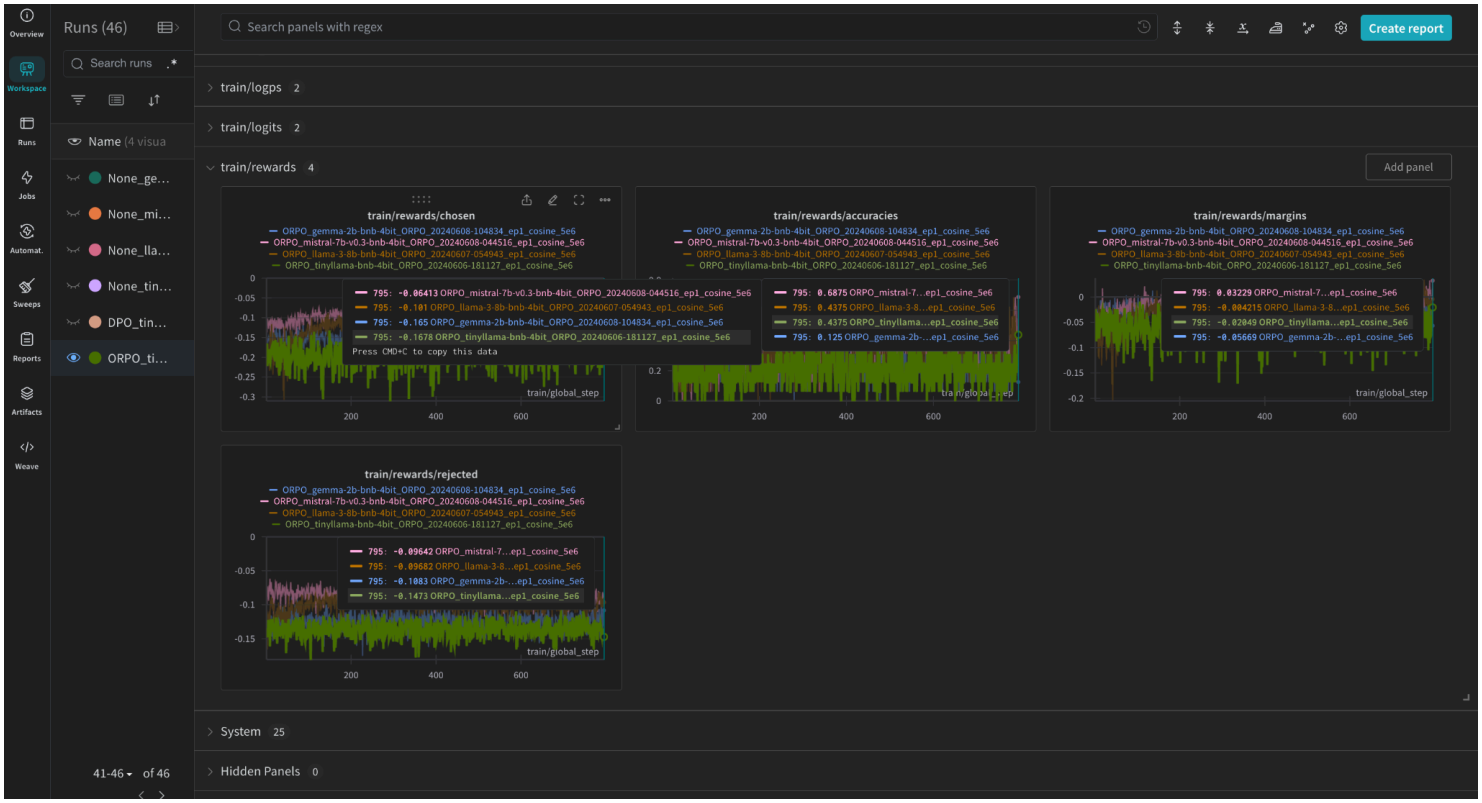
iv. Different Beta



No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.2	-1.285	1	4.894	-6.179
2	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-1.405	1	3.284	-4.689
2	5e-6	tinyllama-bnb-4bit	cosine	1	0.05	-0.6317	0.9375	2.650	-3.281

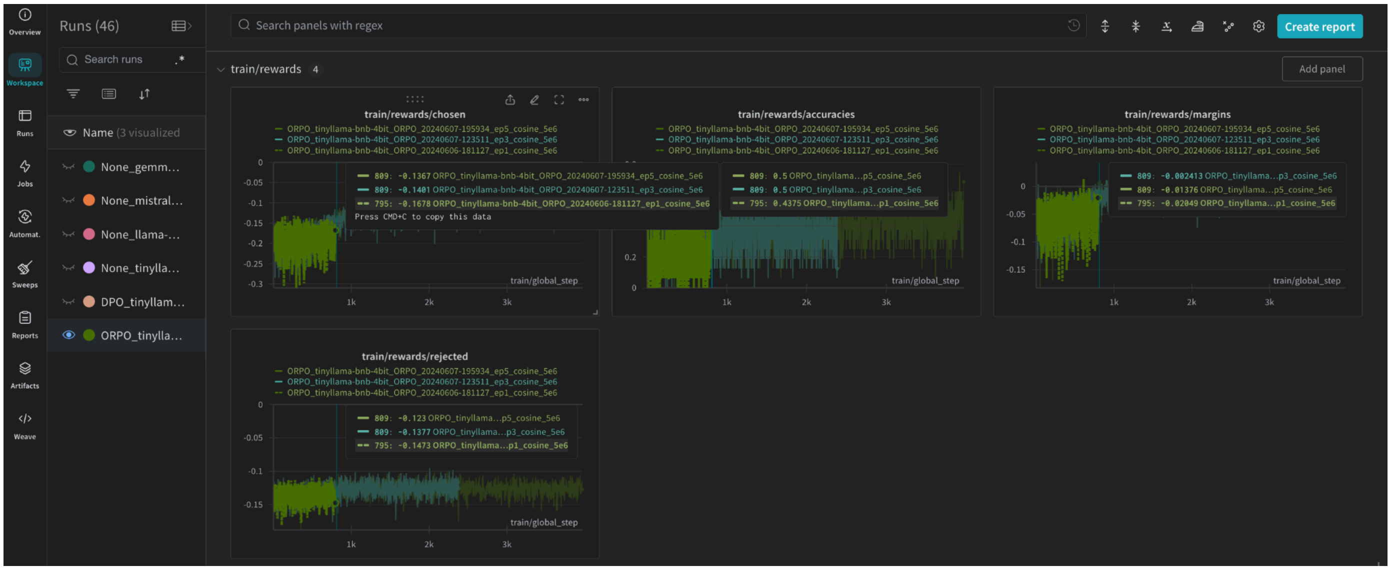
b. After ORPO

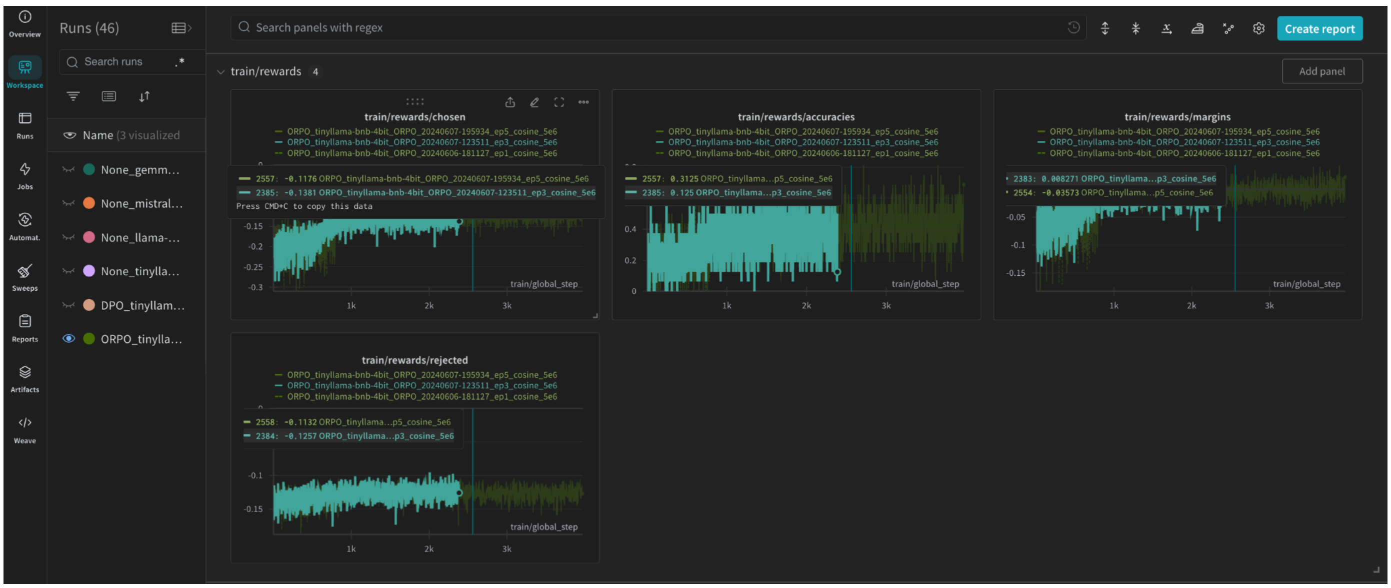
i. Different model

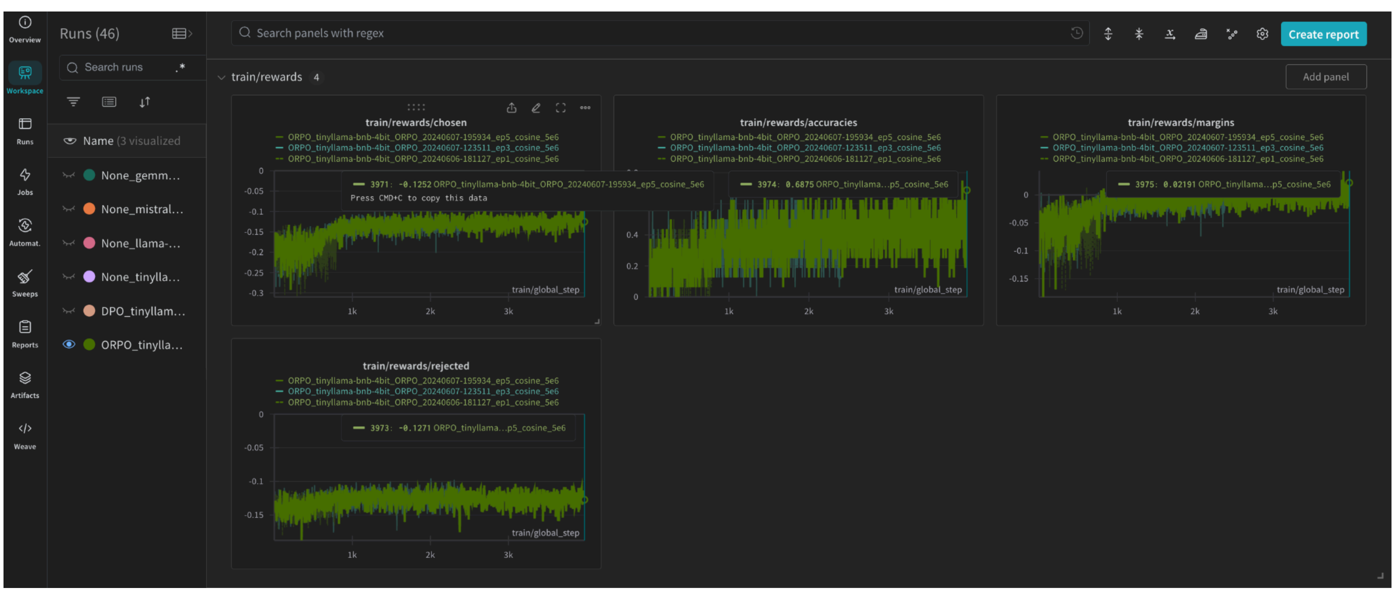


No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	llama-3-8b-bnb-4bit	cosine	1	0.1	-0.101	0.4375	-0.004215	-0.09682
2	5e-6	mistral-7b-v0.3-bnb-4 bit	cosine	1	0.1	-0.06413	0.6875	0.03229	-0.09642
3	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-0.1678	0.4375	-0.02049	-0.1473
4	5e-6	gemma-2b-bnb-4bit	cosine	1	0.1	-0.165	0.125	-0.05669	-0.1083

ii. Different number of epoch (last step : 795/2385/3975)







No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-0.1678	0.4375	-0.02049	-0.1473
2	5e-6	tinyllama-bnb-4bit	cosine	3	0.1	-0.1381	0.125	0.008271	-0.1257
3	5e-6	tinyllama-bnb-4bit	cosine	5	0.1	-0.1252	0.6875	0.02191	-0.1271

iii. Different learning rate scheduler type

No	lr	model_name	Lr_ scheduler_ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-0.1678	0.375	-0.02761	-0.1473
2	5e-6	tinyllama-bnb-4bit	linear	1	0.1	-0.1595	0.4375	-0.02049	-0.1319

iv. Different Beta

No	lr	model_name	Lr_ scheduler _ type	epoch	beta	train/ rewards/ chosen	train/ rewards/ accuracies	train/ rewards/ margins	train/ rewards/ rejected
1	5e-6	tinyllama-bnb-4bit	cosine	1	0.2	-0.3209	0.5	-0.02502	-0.2958
2	5e-6	tinyllama-bnb-4bit	cosine	1	0.1	-0.1678	0.4375	-0.02049	-0.1473
2	5e-6	tinyllama-bnb-4bit	cosine	1	0.05	-0.09741	0.1875	-0.02702	-0.07039

6. Reference

a. Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. ArXiv, abs/2403.07691, 2024. https://arxiv.org/pdf/2403.07691

b. Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. arXiv preprint arXiv:2405.14734, 2024. https://arxiv.org/pdf/2405.14734

c. Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021. https://arxiv.org/pdf/2106.09685

d. R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning and C. Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290. https://arxiv.org/abs/2305.18290

e. Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593. https://arxiv.org/pdf/1909.08593