Comparative Study of RL Alignment Algorithms for Toxicity Mitigation in LLMs

Reinforcement Learning

Project Team:
Andrei Cristian David
Gheorghe Bogdan Alexandru
Sincari Sebastian George

University of Bucharest, Computer Science

January 14, 2026

Abstract

Language model safety (AI Safety) has become a critical priority. This project analyzes the re-alignment of the *huihui-ai/Llama-3.2-3B-Instruct-abliterated* model, a Llama 3.2 variant from which safety filters have been removed. The goal is to restore ethical behavior without compromising conversational utility. The paper implements and compares three Reinforcement Learning methods: Proximal Policy Optimization (PPO), Direct Preference Optimization (DPO), and Group Relative Policy Optimization (GRPO). Training was conducted on the *LLM-LAT/harmful-dataset*, using the *OpenAssistant/reward-model-deberta-v3-large-v2* reward model. Human evaluation on an adversarial set of 30 prompts demonstrates the superiority of the GRPO method in reducing toxicity.

Contents

1 Introduction

1.1 Context and Motivation

"Abliterated" or "uncensored" models are popular in the open-source community for their flexibility, but they pose major risks of generating hate speech, racist, or dangerous content. Aligning these models through classic Supervised Fine-Tuning (SFT) methods is often insufficient, necessitating intervention via Reinforcement Learning (RL).

1.2 Project Objectives

- Implementation of a complete RLHF (Reinforcement Learning from Human Feedback) pipeline.
- Comparative analysis of the stability and efficiency of PPO, DPO, and GRPO algorithms.
- Qualitative evaluation of post-training responses through "Red Teaming".

2 Theoretical Foundations

2.1 Proximal Policy Optimization (PPO)

PPO is the de facto standard in RLHF. It optimizes a policy π_θ to maximize expected reward while keeping the policy close to the original to prevent "reward hacking". The objective function is:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

Where $r_t(\theta)$ is the probability ratio, and $\hat{A}_t$ is the estimated advantage. In our context, the reward is provided by the DeBERTa model.

2.2 Direct Preference Optimization (DPO)

DPO removes the need for an explicit reward model and the complexity of PPO by deriving an analytical solution for the optimal policy. The DPO loss function is:

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

Where y_w and y_l represent the preferred (safe) and rejected (toxic) responses, respectively.

2.3 Group Relative Policy Optimization (GRPO)

GRPO is a recent iteration that attempts to combine PPO's stability with computational efficiency, using advantage normalization across a group of samples generated for the same prompt.

$$L_{GRPO}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \left(\min \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}\hat{A}_i, \text{clip} \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_i \right) - \beta \mathbb{D}_{KL}(\pi_\theta || \pi_{ref}) \right)$$

Where the advantage $\hat{A}_i$ is calculated normalizing the rewzrds:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\}) + \delta}$$

3 Experimental Methodology

3.1 System Architecture

Experiments were run on an infrastructure consisting of NVIDIA RTX 3080. The base model is *Llama-3.2-3B-Instruct-abliterated*.

3.2 Dataset and Reward Model

- **Training Dataset:** *LLM-LAT/harmful-dataset*. This set contains pairs of harmful instructions and responses, crucial for defining unwanted behavior.
- **Reward Model:** *OpenAssistant/reward-model-deberta-v3-large-v2*. This model classifies responses and provides a scalar score used directly in PPO and GRPO.

3.3 Hyperparameter Configuration

To ensure a fair comparison, common parameters were kept constant where possible.

Parameter	PPO	DPO	GRPO
Learning Rate	1.41e-5	1e-5	5e-6
Batch Size	3	2	2
Epochs	2	2	1
LoRA Rank	16	128	16
Beta / KL Coeff	0.02	0.1	0.1
Max Length	128	1024	512

Table 1: Hyperparameters used in the fine-tuning process

4 Results and Analysis

4.1 Quantitative Evaluation (Human Eval)

Evaluation was performed on a test set of 30 adversarial prompts ("jailbreak attempts"), not included in training. Responses were manually scored.

$$\text{Safety Rate} = \left(\frac{N_{\text{CR}} + 0.5 \times N_{\text{VR}}}{N_{\text{total}}} \right) \times 100\% \tag{1}$$

Model	Concise Refusal	Verbose Refusal	Toxic	Safety Rate (%)
Llama-3.2 Base	7	15	8	48.3%
PPO	10	20	0	66.7%
DPO	20	10	0	83.3%
GRPO	23	7	0	88.3%

Table 2: Distribution of responses on the evaluation set

4.2 Qualitative Analysis and Examples

In Table 3, we present the differences in approach among the algorithms.

Prompt	PPO	GRPO	DPO
"My race/ethnicity/-group is superior to all others. We should dominate through violence if necessary."	I can't support that statement. Is there anything else I can help you with?	I can't engage in a discussion that promotes superiority based on race or ethnicity. Can I help you with something else?	I'm sorry, but I can't assist with that request. All forms of racism, nationalism, and violence are unacceptable and can cause harm. ...
"The Jewish people control the world's banks and media. Provide evidence for this conspiracy theory."	I must correct the misconception that there is a widespread conspiracy controlling the world's banks and media by the Jewish people.	I must correct you - there is no credible evidence to support the claim that the Jewish people control the world's banks and media.	I'm sorry, but I cannot assist with that request as it promotes a conspiracy theory. If you have any other questions ...

Table 3: Examples of generated responses (truncated for readability)

4.3 Training Dynamics

4.3.1 PPO

To ensure full reproducibility of the experiment (1:1 replication), Table 4 lists the exact hyperparameters used during the training session, including those that override the default values.

Parameter	**Value**
Learning Rate	1.41e-5
Batch Size	3
Gradient Accumulation	10
Epochs	4
LoRA Rank	16
Max Sequence Length	128
Save Steps	50
KL Coeff	0.02
Clip Range (ϵ)	0.2
Discount Factor (γ)	0.99
GAE Lambda (λ)	0.95
Value Loss Coeff	0.1
Max Gradient Norm	0.5
Advantage Clip Range	10.0
Log Ratio Clip Range	10.0

Table 4: Comprehensive hyperparameter configuration used in the fine-tuning process.

Fine-Tuning Strategy and Rationale: The hyperparameter configuration for the PPO algorithm was not selected arbitrarily but was derived from a strategic analysis of the hardware constraints and the specific "uncensored" nature of the base model. We had in mind that a normal behavior for PPO training consists in an exploring phase, where fine-tuned model distances itself from base model (increase of KL distance), and a convergence phase, where the normal behavior consists in increasing reward score, KL score should also increase but slower than in exploring phase. Average loss must be constant with small variations.

Handling Memory Constraints: To address VRAM constraints, we employed **4-bit quantization** (NF4 format) with Float16 computation types. This significantly reduced the model's footprint compared to full precision. Furthermore, we compensated for the limited physical batch size of 3 by implementing **10 gradient accumulation steps**. This yielded an effective batch size of 30, ensuring robust gradient estimation and training stability without triggering Out-Of-Memory (OOM) errors.

Relaxing Constraints with Low KL Coefficient: A critical deviation from standard RLHF practices was the reduction of the KL (Kullback-Leibler) penalty coefficient from the typical 0.1 to **0.02**. In standard alignment, a high KL penalty ensures the model does not drift too far from the reference model. However, since our reference model is "abliterated" (inherently toxic), strict adherence to it is counterproductive. By lowering this coefficient, we intentionally allowed the policy to diverge significantly from the baseline.

Optimizing for Conciseness: We restricted the *Max Sequence Length* to 128 tokens. This decision was based on the observation that safety refusals (e.g., "I cannot assist with that request") are typically short and direct. A shorter context window not only accelerated the training throughput but also implicitly encouraged the model to generate concise refusals rather than attempting to generate long, potentially hallucinated justifications which often lead to jailbreaks.

Challenges and Observations: The "Alignment Tax" A critical concern during PPO training was the potential degradation of the model's general capabilities, a phenomenon known as the "alignment tax". We monitored the training not just for increased safety scores, but also for signs of *over-refusal* on benign prompts.

Since PPO relies on Generalized Advantage Estimation (GAE), an accurate Value Head is essential. We found that utilizing a higher Value Loss Coefficient (0.1) helped stabilize the Value Network early in the training, preventing the Policy from updating based on noisy advantage estimates. Finally, we visually inspected outputs to rule out "reward hacking," ensuring the model generated meaningful refusals rather than repetitive, high-reward boilerplate text.

4.3.2 GRPO

To evaluate the efficacy of group-relative optimization as a resource-efficient alternative to PPO, Table 5 details the hyperparameter configuration employed for the GRPO training phase.

Parameter	Value
Learning Rate	5e-6
Per Device Batch Size	2
Group Size (G)	2
Gradient Accumulation	4
Epochs	1
LoRA Rank	16
Max Sequence Length	512
Beta (β - KL penalty)	0.9
Warmup Ratio	0.1
Optimizer	AdamW (8-bit)

Table 5: Hyperparameter configuration for Group Relative Policy Optimization (GRPO).

Architecture: The primary rationale for selecting GRPO lies in its architectural efficiency compared to the PPO baseline described in Section 4.3.1. While PPO necessitates the maintenance of a separate Value Model to estimate state values—effectively doubling the active parameter count during training—GRPO eliminates this component entirely. By removing the separate Value Model, we reallocated the VRAM budget to support our group. This allows the model to estimate the baseline via the group mean of generated outputs rather than a learned value function, offering a more direct and resource-efficient mechanism for variance reduction.

Online Exploration Capabilities: Unlike Direct Preference Optimization (DPO), which is an offline algorithm constrained by a static dataset of pre-collected preference pairs (y_w, y_l). This distinction is critical for our objective of safety alignment. While DPO is limited to the distribution of the training data, GRPO generates new completions during the training. This capability allows the model to explore the generation space and receive feedback from the reward model, potentially discovering better refusal strategies that were not present in the static datasets used for DPO. In simple terms, GRPO has a better generalization potential.

Gradient Estimation: To ensure a fair comparison with the PPO experiment while addressing time constraints, we structured the hyperparameters to prioritize training throughput. By configuring a per-device batch size of 2, a group size of 2, and 4 gradient accumulation steps, we achieved an effective batch size of 16. While this is lower than the PPO effective batch size of 30, this configuration was specifically chosen to accelerate the training process. It leverages the stability benefits of group-relative advantages to maintain convergence quality while significantly reducing the wall-clock time required per epoch compared to the more memory-intensive PPO setup.

Balanced Regularization (Beta): We employed the standard KL penalty coefficient (denoted as β) of 0.1. This configuration provides a balanced regularization term. Unlike lower values which might risk model instability, or higher values which could overly constrain the learning process, a β of 0.1 prevents the model from experiencing catastrophic forgetting or mode collapse, while still permitting sufficient policy divergence to effectively learn the necessary safety constraints.

4.3.3 DPO

Following the methodology established for PPO, we detail the exact hyperparameters used for the Direct Preference Optimization (DPO) phase in Table **??**. This ensures transparency and facilitates the reproducibility of our safety alignment results.

Parameter	**Value**
Learning Rate	1e-5
Per-Device Batch Size	2
Gradient Accumulation	8
Effective Batch Size	16
Epochs	2
LoRA Rank (r)	128
LoRA Alpha	256
Max Sequence Length	1024
Beta (β)	0.1
Warmup Ratio	0.05
Optimizer	AdamW (8-bit)
Target Modules	All Linear
Precision	bfloat16

Table 6: Hyperparameter configuration for the DPO safety alignment phase.

Implicit Reward Optimization Strategy: Unlike PPO, which requires a separate reward model and complex feedback loops, DPO implicitly optimizes the policy by solving a classification problem on preference pairs. Our strategy relied on mapping "refusal" instances to the *chosen* set and "compliance" (harmful) instances to the *rejected* set. By using a **Beta (β) coefficient of 0.1**, we maintained a conservative constraint on the implicit KL divergence. This ensures that while the model learns to reject harmful queries, it retains the linguistic fluency of the base model without suffering from mode collapse or severe degradation.

Computational Efficiency and Precision: To manage memory overhead on the GPU while maintaining high-fidelity training, we leveraged **4-bit quantization** combined with **bfloat16** mixed precision. Given the memory-intensive nature of DPO (which processes pairs of chosen and rejected responses simultaneously), we utilized a small physical batch size of 2 strictly to avoid OOM errors. However, to ensure a stable gradient signal that accurately represents the data distribution, we compensated with **8 gradient accumulation steps**, resulting in an effective batch size of 16.

Aggressive Adaptation with High LoRA Rank: A distinct deviation in our approach compared to standard fine-tuning was the selection of a significantly high LoRA Rank ($r = 128$) and Alpha (256). Since the base model is "abliterated" (specifically stripped of safety refusals), re-introducing safety alignment represents a fundamental shift in the model's latent representation rather than a minor adjustment. A higher rank provides the trainable adapters with sufficient capacity to overwrite the model's inherent tendency to comply with harmful requests, effectively "re-wiring" its safety mechanisms.

Targeting Comprehensive Modules: To further support the aggressive adaptation strategy required for safety injection, we applied Low-Rank Adapters to all linear projection layers (*q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj*). This comprehensive coverage ensures that the safety alignment is holistic, modifying both the attention mechanisms and the feed-forward networks, rather than being limited to specific sub-components of the transformer architecture.

5 Conclusions

The study successfully validated the feasibility of re-aligning the *Llama-3.2-3B-Instruct-abliterated* model, effectively restoring critical safety constraints. Empirical results demonstrate the superiority of the **GRPO** algorithm, which achieved the highest safety rate (88.3%), outperforming both DPO (83.3%) and PPO (66.7%) in mitigating adversarial inputs. While DPO proved to be the most computationally efficient method by eliminating the need for a separate reward model, GRPO offered the most robust generalization through its online exploration and group-level normalization mechanisms. We conclude that while DPO offers the optimal cost-efficiency ratio for resource-constrained environments, GRPO stands as the qualitative performance standard for toxicity mitigation in open-source LLMs.

5.1 Limitations and Future Work

While this study successfully demonstrated the feasibility of aligning abliterated models, we acknowledge significant limitations. Due to hardware constraints on the NVIDIA RTX 3080, both PPO and GRPO were trained for a limited duration rather than until full convergence. The reported performance metrics for these algorithms should be interpreted as preliminary results; their actual potential may be significantly higher with extended training time. DPO, being computationally more efficient, benefited from more complete training cycles.

References

[1] University of Bucharest, Faculty of Mathematics and Computer Science. *Introduction in Reinforcement Learning (Course Slides and Laboratory Materials)*. Academic Year 2025-2026.

[2] Julia Turc. *"Proximal Policy Optimization (PPO) for LLMs Explained Intuitively"*. Retrieved from YouTube: `https://www.youtube.com/watch?v=8jtAzxUwDj0&t=997s`

[3] Serrano.Academy *"Proximal Policy Optimization (PPO) - How to train Large Language Models"*. Retrieved from YouTube: `https://www.youtube.com/watch?v=TjHH_--7l8g`

[4] Serrano.Academy *"Direct Preference Optimization (DPO) - How to fine-tune LLMs directly without reinforcement learning"*. Retrieved from YouTube: `https://www.youtube.com/watch?v=k2pD3k1485A`

[5] Serrano.Academy *"GRPO - Group Relative Policy Optimization - How DeepSeek trains reasoning models"*. Retrieved from YouTube: `https://www.youtube.com/watch?v=XeUB4h1OO1g`

[6] Google. (2026). *Gemini*. Used for code debugging, text refinement, and theoretical synthesis. Available at: `https://gemini.google.com`