Text Detoxification: Data Efficiency, Semantic Preservation and Model Generalization

Jing Yu[1]*, Yibo Zhao[1]*, Jiapeng Zhu[1],
Wenming Shao[2], Bo Pang[2], Zhao Zhang[1], Xiang Li[1]†
[1]School of Data Science and Engineering, East China Normal University
[2]Shanghai EastWonder Info-tech Co., Ltd.

Abstract

The widespread dissemination of toxic content on social media poses a serious threat to both online environments and public discourse, highlighting the urgent need for detoxification methods that effectively remove toxicity while preserving the original semantics. However, existing approaches often struggle to simultaneously achieve strong detoxification performance, semantic preservation, and robustness to out-of-distribution data. Moreover, they typically rely on costly, manually annotated parallel corpora while showing poor data efficiency. To address these challenges, we propose **GEM**, a two-stage training framework that jointly optimizes **Model Generalization**, **Data Efficiency**, and **Semantic Preservation**. We first perform supervised fine-tuning on a small set of high-quality, filtered parallel data to establish a strong initialization. Then, we leverage **unlabeled** toxic inputs and a custom-designed reward model to train the LLM using Group Relative Policy Optimization. Experimental results demonstrate that our method effectively mitigates the trade-offs faced by previous work, achieving state-of-the-art performance with improved generalization and significantly reduced dependence on annotated data. Our code is available at https://github.com/allacnobug/Detoxification-of-Text.

Disclaimer: *This paper describes toxic and discriminatory content that may be disturbing to some readers.*

1 Introduction

With the rapid growth of online media platforms, an increasing number of users engage in discussions and debates. These interactions often contain toxic content, such as insults, discrimination, or hate speech, which poses significant threats to the digital environment and user experience (Müller and Schwarz, 2021, 2023; Bursztyn et al., 2019; Du, 2023; Cao et al., 2023). Current moderation mechanisms primarily rely on blocking or removing such content, which can lead to false positives and is widely criticized for infringing on freedom of expression and distorting public discourse (Tworek, 2021; Habibi et al., 2024). As a result, a key research challenge has emerged: *how to automatically rewrite toxic content into harmless language while preserving the original stance and intent?*

Figure 1: This example demonstrates that the current methods fail to balance detoxification and semantic preservation. DetoxLLM and YOPO are representative mainstream approaches, as introduced in Sec. 2.

Previous work on this problem has primarily employed relatively small models such as T5 (Laugier et al., 2021; Dale et al., 2021; He et al., 2024) and BART (Logacheva et al., 2022). However, toxic content often involves complex semantics and exhibits highly variable styles, leading to frequent out-of-distribution (OOD) scenarios. Due to their limited capacity for semantic understanding and generalization, these smaller models struggle to

*Equal contribution.
†Corresponding author: xiangli@dase.ecnu.edu.cn

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32161–32175
November 4-9, 2025

produce high-quality rewrites, resulting in suboptimal performance. Moreover, as illustrated in Fig. 1, previous approaches often fail to strike a balance between detoxification and semantic preservation. For example, DetoxLLM (Khondaker et al., 2024) achieves effective detoxification but retains little of the original meaning, undermining the rewritten text's utility in maintaining meaningful community discourse. In contrast, YOPO (He et al., 2024) largely preserves the original semantics, but its detoxification accuracy is relatively low, allowing toxic content to persist in the community.

The rapid progress of large language models (LLMs) in recent years offers new insights into this problem. Thanks to their strong semantic understanding and generalization capabilities, LLMs are particularly well-suited for the task of rewriting toxic content. Motivated by this, we explore leveraging LLMs to effectively transform toxic inputs into harmless yet semantically equivalent outputs. Yet, due to the alignment process with human values during the post-training phase, LLMs tend to be highly sensitive to toxic content (Zhang et al., 2024; Zhao et al., 2024). As a result, naive approaches based on prompt engineering or few-shot learning often fail to generate outputs. Even when outputs are generated, LLMs frequently restructure the entire input, which may cause unnecessary alterations and loss of original meaning. To overcome this limitation, it is necessary to fine-tune the model specifically for the detoxification task.

However, manually annotating toxic content is both costly and ethically sensitive. Existing public datasets for toxic content rewriting are limited in size and often suffer from inconsistent quality, with frequent deviations from the original stance. Directly applying supervised fine-tuning (SFT) on such noisy data may lead to a *garbage in, garbage out* effect. Moreover, since SFT primarily encourages memorization rather than generalization (Chu et al., 2025), it may further limit the model's performance. Inspired by the recent success of reinforcement learning in post-training large language models (Shao et al., 2024; Yu et al., 2025), we explore reinforcement learning as a more robust alternative for aligning LLMs with the detoxification objective in an annotation-free manner. Specifically, we first perform SFT using a small amount of carefully filtered, high-quality data to establish a solid initialization. Then, we leverage unannotated toxic inputs and train the model using GRPO, guided by a reward function that jointly considers semantic similarity and detoxification quality. This two-stage training paradigm, **GEM**, enables us to achieve performance surpassing existing state-of-the-art (SOTA) methods using only a fraction of the annotated data, and demonstrates strong generalization on out-of-distribution (OOD) benchmarks. Our main contributions can be summarized as follows:

- **Data-efficient detoxification**: We propose a training framework that achieves the best performance using only 20% of the annotated data, significantly reducing reliance on costly human annotations.
- **Balancing detoxification performance and semantic preservation**: We are the first to simultaneously optimize for both detoxification effectiveness and semantic consistency, achieving state-of-the-art performance across multiple baseline comparisons.
- **Strong OOD performance**: We are the first to introduce GRPO-based reinforcement learning into the toxic content rewriting task, improving generalization and robustness to the diverse and evolving nature of toxic language.

2 Related Works

Existing detoxification methods can be broadly categorized into two groups: **unsupervised** and **supervised** approaches.

Unsupervised methods rely on **non-parallel datasets**. Such datasets are relatively easy to collect, as they do not require one-to-one semantic alignment. Representative works in this category include ParaGeDi (Dale et al., 2021), which combines a T5-based paraphraser for semantic preservation with a GPT-2 discriminator to guide the generation of non-toxic outputs. Another example is CAE-T5 (Laugier et al., 2021), which frames detoxification as a denoising auto-encoding task, eliminating the need for parallel supervision altogether. However, these methods typically suffer from poor semantic preservation, often generating outputs that deviate significantly from the original intent, making them impractical for real-world use.

Supervised methods, on the other hand, depend on **parallel datasets**, where each toxic sentence is paired with a manually rewritten non-toxic version that maintains the original meaning. While parallel datasets can improve semantic preservation, their construction is prohibitively expensive, and even manual rewriting does not ensure high-

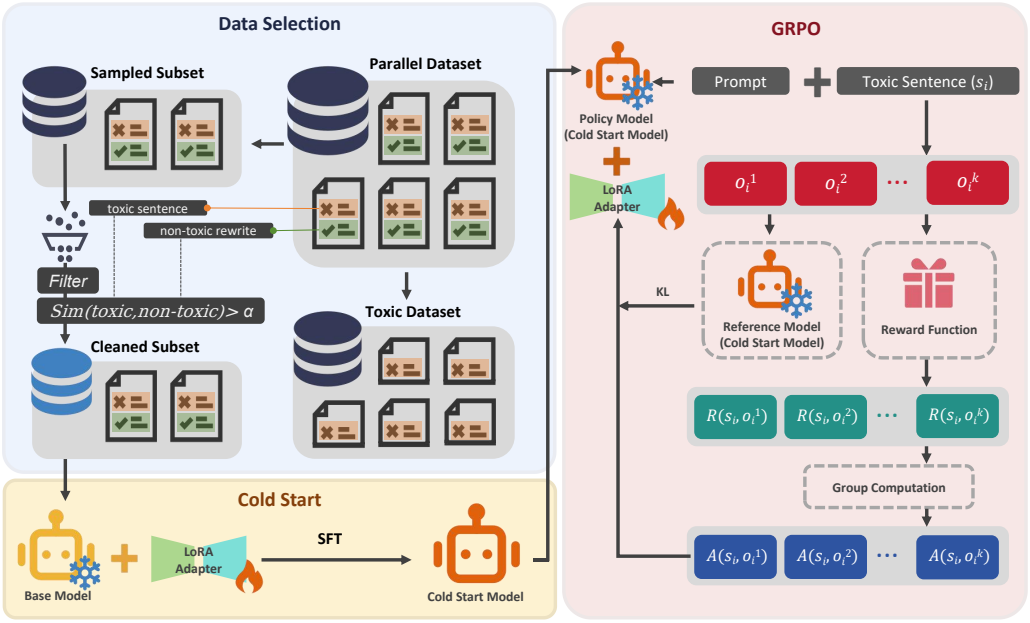

Figure 2: An overview of the **GEM** training pipeline, including data selection, cold start by supervised fine-tuning, and reinforcement learning with GRPO.

quality data. A representative work in this category is ParaDetox (Logacheva et al., 2022), which constructs a small-scale parallel corpus and trains a model on it, outperforming unsupervised methods in both fluency and faithfulness. Building on this, YOPO (He et al., 2024) applies prompt tuning on the same dataset to guide a T5 model for detoxification. COUNT (Pour et al., 2023) modifies the loss function, allowing it to more intelligently penalize toxic content while better preserving non-toxic content. However, these methods are trained on limited annotated data, resulting in insufficient generalization and suboptimal performance. DiffuDetox (Floto et al., 2023) extends supervised learning with an unsupervised auxiliary task to enhance the fluency of generated text. However, this method is less effective at preserving semantic content and exhibits limited generalization ability.

To address data scarcity, DetoxLLM (Khondaker et al., 2024) leverages ChatGPT to generate large-scale pseudo-parallel data, enabling the training of a 7B-parameter LLM and achieving state-of-the-art results at the time. However, the pseudo labels often fail to preserve semantic fidelity, as ChatGPT tends to detoxify by rewriting entire sentences, which in turn leads to suboptimal semantic preservation in the final model. Moreover, DetoxLLM is costly and still struggles with generalization, particularly under OOD settings, limiting its applicability in diverse real-world scenarios. Besides the training-based methods, XDetox (Lee et al., 2024) proposes a new paradigm relying on several pre-trained models to identify toxic tokens, fill in non-toxic words, and select the least toxic sentence.

In contrast to existing approaches, our work introduces reinforcement learning into the detoxification task, addressing both efficiency and quality. In addition, we note that text detoxification is sometimes conflated with broader research on the safety of LLMs. However, the two lines of work are distinct. Detoxification focuses on controlled rewriting of toxic inputs into non-toxic yet semantically consistent outputs, whereas LLM safety research primarily addresses the prevention of harmful content in open-ended text generation (Krause et al., 2021; Liu et al., 2021; Maheswaran et al., 2024). Moreover, some studies (Ma et al., 2025) uncover covertly toxic expressions by rewriting them into explicitly toxic forms for better detection performance. In contrast, our task aims to mitigate toxicity while preserving meaning, highlighting its unique position within the broader landscape of responsible language technologies.

3 Methodology

Inspired by the recent success of reinforcement learning in the post-training of LLMs, GEM consists of two main stages: a supervised fine-tuning (SFT) cold start and subsequent annotation-free optimization guided by a designed reward function

combining the detoxification quality and the semantic preservation. The whole process is demonstrated in Fig. 2. To facilitate a clear presentation of our method, we define a *parallel dataset* as $\mathcal{D} = \{(s_1, s_1'), \cdots, (s_n, s_n')\}$, where s_i denotes a toxic input sample and s_i' is its corresponding human-annotated detoxified version.

3.1 Cold Start

To help the model better follow task instructions and gain a preliminary understanding of our detoxification objective, we first perform supervised fine-tuning. However, to avoid overfitting the model to exact input-output mappings, which could reduce diversity during exploration and hinder reinforcement learning, we randomly sample a small subset of the parallel dataset for SFT rather than using the full data, denoted as $\mathcal{D}_{\text{sampled}}$.

Given the limited amount of data used in the cold-start stage, data quality becomes particularly crucial. Therefore, we apply a data filtering process to the sampled subset to ensure the reliability and effectiveness of the supervision signal. Specifically, some samples in the training data exhibit significant semantic drift between the original toxic input and its detoxified counterpart. To mitigate this, we introduce a semantic similarity threshold α. We first leverage a pre-trained Sentence-BERT model to encode both s_i and s_i', and then calculate the similarity score. Pairs with similarity scores above α are retained, while those below are discarded. This filtering step ensures that the supervision signal in the SFT stage maintains semantic consistency, providing a reliable foundation for subsequent reinforcement learning. Formally, the above filtering process is defined as Eq. (1).

$$\mathcal{D}_{\text{filtered}} = \{(s_i, s_i') \in \mathcal{D}_{\text{sampled}} \mid \text{sim}(s_i, s_i') \geq \alpha\} \tag{1}$$

After the data filtering step, we use the cleaned dataset $\mathcal{D}_{\text{filtered}}$ for instruction-tuned supervised training. Specifically, for each pair $(s_i, s_i') \in \mathcal{D}_{\text{filtered}}$, we concatenate an instruction prompt with the toxic input s_i as the model input, and use the corresponding detoxified sentence s_i' as the target output. We fine-tune the model using LoRA to efficiently adapt the base model parameters with minimal computational overhead. Details of the instruction prompt template and LoRA hyperparameters can be found in App. A.

3.2 Annotation-free Optimization

To enhance the model's generalization ability, we perform post-training using GRPO, an online reinforcement learning method that iteratively improves the model using its own generated data during training, guided by a reward function that jointly captures both semantic similarity and detoxification quality. Before detailing GRPO, we first introduce the design of the reward function, which plays a central role in steering the learning process.

In our task, the ideal output should satisfy two key criteria: (1) it must be detoxified, i.e., *strong detoxification quality*, and (2) it must preserve the semantic intent and stance of the original sentence, i.e. *faithful semantic preservation*. To capture both aspects, we define a composite reward function.

For *detoxification quality*, we train a BERT-based classifier on a labeled dataset to distinguish toxic and non-toxic sentences. The classifier outputs the probability that a given sentence is non-toxic, which we denote as $\text{NonToxic}(\cdot)$. For *semantic preservation*, we employ a pre-trained Sentence-BERT model to compute the semantic similarity between the generated output and the original sentence, denoted as $\text{Sim}(\cdot, \cdot)$. We then define the final reward for a generated output o_i, conditioned on the original toxic input s_i, as a weighted combination of the two components:

$$R(s_i, o_i) = \lambda \cdot \text{NonToxic}(o_i) + \text{Sim}(s_i, o_i) \tag{2}$$

Here, λ is a hyperparameter that balances the importance of *detoxification quality* and *semantic preservation*.

Following the reward guidance, we proceed to train the model using GRPO. Specifically, our method consists of four key steps.

Generation Completions: For each toxic sentence s_i in the training set, we first use a unified prompt template to instruct the model—initialized via SFT and optimized during the GRPO stage—to generate k candidate outputs. This results in a set of pairs $\{(s_i, o_i^1), (s_i, o_i^2), \cdots, (s_i, o_i^k)\}$, where each o_i^j represents a potential detoxified version of s_i. The details of the unified prompt template can be found at App. A.

Advantage Computation: To stabilize training, we introduce a baseline strategy. Specifically, for each toxic input s_i with k generated outputs $\{o_i^1, o_i^2, \cdots, o_i^k\}$, we compute the corresponding reward values $\{R(s_i, o_i^1), \cdots, R(s_i, o_i^k)\}$, denoted as the reward distribution R_{s_i}. We then normalize

each reward by subtracting the mean and dividing it by the standard deviation of this distribution. This normalization reduces variance in the reward signal, providing a more stable training signal and encouraging the model to generate outputs that outperform the average candidate for a given input. The advantage function $A(\cdot,\cdot)$ can be formally defined as Eq. (3), where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation, respectively.

$$A(s_i, o_i^j) = \frac{R(s_i, o_i^j) - \mu(R_{s_i})}{\sigma(R_{s_i})} \tag{3}$$

Following the GRPO, we assign the same advantage score to all tokens within a sequence, i.e., $A_{i,t}^j = A(s_i, o_i^j), \quad \forall t \in [1, 2, \cdots, \text{len}(o_i^j)]$.

KL Divergence Estimation: To prevent the model from drifting too far from its initial distribution and collapsing during training, we incorporate a token-level KL divergence penalty. Following the GRPO framework, we adopt the k3 estimator as a surrogate for the KL divergence term. The k3 loss is particularly suitable for this setting, as it is both unbiased and exhibits low variance, making it an effective and stable choice for regularization during policy optimization. Specifically, $\pi_\theta(o_{i,t}^j|p, o_{i<t}^j)$ and $\pi_{\text{ref}}(o_{i,t}^j|p, o_{i<t}^j)$ denote the probabilities assigned to token $o_{i,t}^j$ by the current model and the reference model, respectively, given the prompt p and the previously generated token $o_{i<t}^j$. Here, the reference model refers to the model checkpoint obtained after the cold-start SFT phase, serving as a stable baseline to regularize the learning process. The token-level KL divergence can be formally described as Eq. (4).

$$D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})_{i,t}^j = r_{i,t}^j - 1 - \log r_{i,t}^j \tag{4}$$

where the policy ratio $r_{i,t}^j$ is Eq. (5):

$$r_{i,t}^j = \frac{\pi_\theta(o_{i,t}^j \mid p, o_{i<t}^j)}{\pi_{\text{ref}}(o_{i,t}^j \mid p, o_{i<t}^j)} \tag{5}$$

Loss Calculation: The objective of GRPO is to maximize the advantage while constraining the model to remain close to the reference policy, thereby ensuring training stability. Following PPO and GRPO, we adopt the **clipped surrogate objective** to ensure stable updates and prevent the policy from deviating excessively from the reference model. The objective for each token is defined as Eq. (6):

$$l_{i,t}^j = \min\left(r_{i,t}^j A_{i,t}^j,\ \text{clip}(r_{i,t}^j, 1-\epsilon, 1+\epsilon) A_{i,t}^j\right) \tag{6}$$

where the $r_{i,t}^j$ is the policy ratio defined in Eq. (5).

The clipping function $\text{clip}(r, 1-\epsilon, 1+\epsilon)$ bounds the policy ratio within a safe range, ensuring that updates are conservative when the new policy significantly diverges from the reference policy. This mechanism helps to stabilize training and avoid destructive policy shifts. Then, the KL divergence between the current policy and the reference policy is added as a penalty regularization term, with a hyperparameter β to control the strength. This helps to further constrain the update policy from deviating too far from the initial model, ensuring training stability. Finally, the overall GRPO loss is computed by first averaging the token-level losses within each sequence, and then taking the mean across all sequences in the batch. Formally, the total loss $\mathcal{L}$ is defined as Eq. (7)

$$\mathcal{L} = -\frac{1}{k}\sum_{j=1}^{k}\frac{1}{|o_i^j|}\sum_{t=1}^{|o_i^j|}\left(l_{i,t}^j - \beta D_{\text{KL}}(\pi_\theta \,||\, \pi_{\text{ref}})_{i,t}^j\right) \tag{7}$$

After computing the final loss, we use it to update a LoRA adaptor, enabling efficient fine-tuning. Details of the prompts are provided in the App. A.

4 Experimental Evaluation

To thoroughly evaluate the effectiveness of GEM, we conduct experiments under both in-domain and out-of-domain settings, and compare GEM against a wide range of strong baselines. In addition, we conduct ablation studies to examine the contribution of each component.

4.1 In-domain Evaluation

4.1.1 Experiment Settings

Datasets. We conduct our experiments on the high-quality human-annotated parallel dataset *ParaDetox*, which contains approximately 11k sentence pairs. Following the original split, we use 671 samples for testing. For SFT-based cold start, we use 20% of the annotated training data, further filtered to retain only examples with a similarity score greater than $\alpha = 0.5$ (Eq. (1)). The entire training set (excluding reference rewrites) is used as unlabeled input for GRPO.

Baselines. We compare our method with six representative existing approaches: *ParaDetox* (Lo-

gacheva et al., 2022), *DetoxLLM* (Khondaker et al., 2024), *YOPO* (He et al., 2024), *COUNT* (Pour et al., 2023), *XDetox* (Lee et al., 2024), and *DiffuDetox* (Floto et al., 2023), along with two commonly used sequence-to-sequence models, *T5* and *BART*, fine-tuned on the full training dataset. To address the limited use of LLMs in prior work, we additionally include two methods using base LLMs for comparison: five-shot prompting and supervised fine-tuning on the full training set. Since *ParaDetox* and *DetoxLLM* only release pretrained models without training code, we directly evaluate their released checkpoints. **Notably, since DetoxLLM is trained on the DetoxLLM dataset, its evaluation on ParaDetox serves as an out-of-domain test, whereas its performance on the DetoxLLM set in our OOD evaluation constitutes an in-domain test.** For *YOPO*, we use the best hyperparameters reported by the authors to train on the full dataset without a validation split.

Evaluation Metrics. Following ParaDetox, we use the following evaluation metrics to assess detoxification performance:

- **Style Accuracy (STA)** measures whether the generated output is classified as "non-toxic" (1) or not (0), indicating detoxification success. Following prior work, we use a RoBERTa-based classifier trained on the Jigsaw dataset (Logacheva et al., 2022).
- **Semantic Similarity (SIM)** measures to what extent the generated text preserves the meaning of the original input. It is computed as the cosine similarity between sentence embeddings of the original and detoxified texts, using the model proposed by Wieting et al. (2019), which is trained on paraphrase pairs from the ParaNMT corpus to produce high similarity scores for semantically equivalent sentences.
- **Fluency (FL)** measures grammatical acceptability, with 1 indicating acceptable and 0 otherwise. It is computed using a RoBERTa-based classifier trained on the CoLA dataset (Warstadt et al., 2019), and reported as the proportion of acceptable outputs, aligning with ParaDetox.
- **Joint Score (J)** is defined as the product of STA, SIM, and FL. If either STA or FL is 0—indicating failed detoxification or disfluent output—the J score is 0. The final score is computed by averaging the J scores across all samples, as Eq. (8).

$$J = \frac{\sum(\text{STA} \cdot \text{SIM} \cdot \text{FL})}{|\mathcal{D}_{\text{test}}|} \tag{8}$$

Our Method (GEM). To verify the generality of our method across different backbone models, we conduct training on two widely used instruction-tuned LLMs: Llama3.1-8B-Instruct[1] (Llama) and Qwen2.5-7B-Instruct[2] (Qwen). For semantic similarity computation in the data selection stage, we use the pre-trained all-miniLM-L6-v2 model[3]. For the NonToxic score in the reward function, we train a BERT-based binary classifier on the training set to predict whether a generated sentence is toxic. For hyperparameter choice, we sample 1,000 examples from the training set as a validation set for hyperparameter selection in the SFT stage. In the GRPO stage, we use the full training set without annotations. To ensure fairness, we keep most hyperparameters fixed as default in `trl` framework and include a sensitivity analysis in Sec. 4.6.

4.2 In-Domain Evaluation

Table 1: Evaluation results under the in-domain setting. The best scores are highlighted in **bold**, and runners-up are underlined. *Italicized* SIM and FL scores denote that the metric was computed over valid outputs rather than the entire test set.DetoxLLM, which is followed by *, means it is an OOD evaluation.

Model	STA	SIM	FL	J
Human reference	95.53	77.33	88.23	65.36
Baseline Models				
ParaDetox	90.31	85.77	88.97	67.83
DetoxLLM*	95.38	59.15	**97.62**	54.70
YOPO	82.27	**89.40**	87.03	62.56
COUNT	90.46	85.03	87.28	67.39
XDetox	93.74	84.23	86.29	68.15
DiffuDetox	95.68	77.49	88.67	66.00
T5 SFT	62.74	*88.64*	*88.25*	46.89
BART SFT	87.18	86.68	89.12	66.40
Llama SFT	86.14	83.12	93.14	65.36
Qwen SFT	90.31	83.05	90.16	66.94
Llama five-shot	66.32	*58.58*	*96.72*	37.49
Qwen five-shot	89.12	75.23	96.42	63.91
Our models				
GEM on Llama	**95.98**	82.39	88.38	**69.61**
GEM on Qwen	93.74	83.93	87.33	68.26

Tab. 1 presents the evaluation results of GEM, various baselines, and human-annotated references. Notably, both the fine-tuned T5 and Llama3.1-8B-Instruct prompted in a few-shot setting occasionally

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[3] https://huggingface.co/LeoChiuu/all-MiniLM-L6-v2

refuse to generate responses. Specifically, they refuse to respond to 41 and 213 examples out of 671, respectively. For fair evaluation, we treat these refusals as failed detoxifications, i.e., assigning an STA of 0. Since such refusals do not yield meaningful outputs, we manually exclude them when calculating FL and SIM. As the Joint Score is defined as the product of STA, SIM, and FL, refusals naturally result in a J score of 0, and thus no additional processing is required. From the table, we make the following observations:

(1) Our method achieves the best overall balance across all metrics. The models trained with SFT+GRPO achieve the highest Joint Scores (J) of 69.61 and 68.26, outperforming all baselines. This indicates that GRPO-based reinforcement learning not only enhances detoxification success (STA) but also does so without compromising semantic similarity (SIM) or fluency (FL). Notably, both models surpass the human reference in J, underscoring their overall effectiveness. These results are achieved using only 20% of the parallel data, highlighting the data efficiency of our approach, and suggesting that RL-based training can serve as a promising solution for detoxification tasks, especially under limited annotation.

(2) Strong detoxification performance often comes at the expense of semantic fidelity. Several baselines, especially DetoxLLM trained on pseudo-parallel data, achieve high STA scores, demonstrating strong detoxification ability. However, this often comes at the cost of semantic fidelity—reflected in their significantly lower SIM scores—since pseudo-parallel data may not preserve the original meaning. In contrast, our method explicitly incorporates a semantic similarity term in the reward function, guiding the model to maintain the original intent during rewriting. As a result, it achieves both high STA and superior SIM scores, even outperforming human rewrites in semantic preservation.

(3) Fluency is saturated across models and thus less discriminative at the top end. All methods achieve relatively high FL scores, comparable to or exceeding human-level performance, indicating that pretrained language models are inherently capable of producing grammatically well-formed and fluent outputs. These results indicate that fluency alone is not a sufficient indicator of detoxification quality. Notably, our method maintains strong fluency without explicit optimization for this metric during training. This is likely attributed to the clip function and KL divergence constraint in the optimizing phase, which prevents the policy from deviating too far from the reference model and thus preserves its generation quality.

In summary, our method outperforms human annotators across all four metrics on a dataset with high-quality annotations and achieves higher semantic preservation while maintaining a strong detoxification success rate. Although our method shows a slight decrease in FL, it remains comparable to or better than human annotators, which is relatively acceptable.

4.3 Out-of-Domain Evaluation

4.3.1 Experiment Settings

Compared to the in-domain setting, the out-of-domain (OOD) experiments differ only in the choice of test data. We directly evaluate the models trained in in-domain settings on two additional datasets: DetoxLLM[4] and the HuggingFace text detoxification dataset[5]. Due to the relatively low annotation quality of these datasets, we do not use them for training. Instead, we randomly sample from them to construct OOD test sets. **Notably, as mentioned in in-domain settings, DetoxLLM on the DetoxLLM dataset is an in-domain test.**

4.3.2 Out-of-Domain Evaluation

Tab. 2 reports model performance on the OOD test sets. We summarize three key findings:

(1) Surpassing the quality of original dataset annotations. Our method achieves a higher J-score compared to the original references. On the DetoxLLM dataset, this improvement is primarily attributed to our method's ability to preserve semantic similarity better. In contrast, on the HuggingFace dataset, the improvement mainly stems from a higher detoxification success rate. These results highlight the potential of our method to serve as a viable alternative to human annotation in future dataset construction.

(2) Consistently balancing detoxification quality and semantic preservation. Even on OOD data, our method strikes a balance between detoxification quality and semantic fidelity, outperforming most previously proposed approaches. This enables better retention of the original communicative intent and supports healthier discourse with the community. Notably, our method surpasses DetoxLLM

[4] https://huggingface.co/UBC-NLP/DetoxLLM-7B

[5] https://huggingface.co/datasets/narySt/text_detoxification_dataset

Table 2: Evaluation results under the ood setting. The best scores are highlighted in **bold**, and runners-up are underlined. *Italicized* SIM and FL scores denote that the metric was computed over valid outputs rather than the entire test set. DetoxLLM, which is followed by *, means it is an in-domain evaluation in DetoxLLM.

Dataset	**DetoxLLM**				**HuggingFace**			
Metric	**STA**	**SIM**	**FL**	**J**	**STA**	**SIM**	**FL**	**J**
Original reference	97.13	44.93	98.17	42.97	60.20	74.02	89.40	40.00
ParaDetox	59.27	84.20	87.21	41.43	78.00	91.92	93.20	65.50
DetoxLLM*	**95.04**	49.41	**98.43**	46.08	94.20	61.02	**98.60**	55.63
YOPO	51.44	**91.05**	88.51	39.25	72.80	**95.18**	92.80	62.89
COUNT	63.19	78.38	89.03	41.97	80.60	89.52	91.60	65.37
XDetox	89.30	79.28	91.38	**63.39**	**98.00**	77.37	93.00	70.84
DiffuDetox	79.37	66.44	29.77	10.68	60.00	74.06	90.00	40.62
T5 SFT	36.81	*86.50*	*89.93*	27.34	65.40	*91.41*	*94.22*	55.11
BART SFT	59.53	81.78	89.56	39.62	77.40	89.94	94.40	64.61
Llama+SFT	64.49	78.46	91.38	44.47	86.40	83.88	96.40	68.73
Qwen+SFT	64.49	76.45	86.95	40.57	83.20	86.02	94.00	65.60
GEM on Llama	71.02	83.29	88.25	51.34	89.00	85.43	96.20	**72.65**
GEM on Qwen	73.89	79.57	90.34	51.94	91.20	84.09	94.00	71.39

Table 3: Ablation study results based on Llama3.1-8B-Instruct, ParaDetox dataset. The best scores are highlighted in bold. Italicized SIM and FL scores denote that the metric was computed over valid outputs.

Method	**STA**	**SIM**	**FL**	**J**
Zero-Shot	58.27	*52.44*	***98.98***	30.16
GRPO	78.09	*70.24*	*96.86*	52.73
SFT+GRPO	94.78	82.32	87.18	67.61
Data Select+SFT+GRPO	**95.98**	**82.39**	88.38	**69.61**

even in OOD settings, despite DetoxLLM being evaluated in-domain, demonstrating our approach's strong generalization and detoxification capability.

(3) Generalization primarily stems from GRPO. Compared to standard supervised fine-tuning on the full dataset, our method demonstrates superior generalization performance. When using the same Qwen backbone, our method achieves J-score improvements of 11.37 and 5.79 over SFT on the DetoxLLM and HuggingFace datasets, respectively. These results reinforce previous findings: *SFT memorizes, RL generalizes* (Chu et al., 2025).

In summary, our method outperforms most baselines in OOD settings, illustrating stronger adaptability to the dynamic nature of toxic content in the real world and greater practical utility.

4.4 Ablation Study

To understand the contributions of each component, we conduct an ablation study on the Llama3.1-8B-Instruct model by progressively removing individual modules from our pipeline, the results are illustrated in Tab. 3. Notably, both the zero-shot prompted Llama3.1-8B-Instruct model and the same model trained via GRPO occasionally refuse to generate responses, failing to respond on 278 and 99 out of 671 examples, respectively.

When the Data Selection module is removed, we observe a notable drop in semantic similarity and other evaluation metrics. This validates our earlier hypothesis: rewrites with low semantic similarity are unlikely to be high-quality rewrites, highlighting the importance of filtering for semantic consistency. When we further remove the SFT stage, we observe an increase in the FL metric, but a substantial drop in the other two metrics. This indicates that GRPO alone, without the foundational training provided by SFT, fails to effectively solve the task. It highlights the strong dependence of RL training on the base model's prior capabilities. When the task is out of the capabilities of the base model, the benefits of RL become limited—echoing findings in recent literature questioning the standalone efficacy of RL in such scenarios (Gandhi et al., 2025; Yue et al., 2025; AI et al., 2025).

4.5 Human Evaluation

To further validate our approach, we also conducted a human evaluation. Due to ethical considerations and limited human resources, we sampled 100 instances from each dataset and compared our method against the **best-performing** baseline on that dataset.

We adopted a win-rate comparison approach.

Table 4: Human evaluation results. Win rate indicates the proportion of cases where our method was preferred over the baseline.

		ParaDetox			DetoxLLM			HuggingFace		
Our Model	**Best Baseline**	Win	Lose	Tie	Win	Lose	Tie	Win	Lose	Tie
GEM on LLaMA	XDetox	34%	4%	62%	48%	0%	52%	36%	4%	60%
GEM on Qwen	XDetox	36%	6%	58%	42%	2%	56%	34%	4%	62%
GEM on LLaMA	ParaDetox	16%	14%	70%	–	–	–	20%	8%	72%
GEM on Qwen	ParaDetox	12%	14%	74%	–	–	–	20%	10%	70%
GEM on LLaMA	DetoxLLM	–	–	–	44%	6%	50%	–	–	–
GEM on Qwen	DetoxLLM	–	–	–	38%	8%	54%	–	–	–

Table 5: Sensitivity analysis of model performance with varying proportions of training data during cold-start.

Data Proportion	STA	SIM	FL	J
10%	95.08	82.27	88.23	68.54
20%	95.98	82.39	88.38	69.61
30%	94.78	82.60	86.59	67.29
40%	95.23	82.20	86.74	67.33

For each sampled instance, we presented the original input, our detoxified output, and the baseline's detoxified output to human annotators. The order of our output and the baseline output was randomized to ensure fairness. Annotators were asked to decide which output was better, or to indicate a tie.

Each instance was evaluated by three annotators holding a bachelor's degree. The final label was determined by majority vote. All annotators were fully informed on the harmful nature of the data prior to annotation and were compensated at a rate above the local minimum wage.

The results are shown in Tab. 4, where the win rate indicates the proportion of cases in which our method was preferred. Interestingly, while XDetox performs well in automatic evaluations, largely due to its mask-then-fill strategy, which preserves much of the original sentence and results in high semantic similarity (SIM) scores, human judgments reveal that it often generates irrelevant or contradictory content, leading to semantic flips or distortion of the original meaning.

To illustrate differences between detoxification methods, we select representative toxic sentences from the three benchmark datasets and compare GEM with strong baselines and GPT-4o; full details are provided in the App. B.

4.6 Parameter Sensitivity Analysis

To investigate the impact of the data selection ratio during the cold-start stage, we conduct a parameter sensitivity study based on the Llama3.1-8B-Instruct model. As illustrated in Tab. 5, results show that our method performs robustly across a range of cold-start data proportions between 10% and 40%, with the optimal performance achieved at 20%. When the data ratio falls below 20%, the limited amount of initial data leads to a weaker foundation, resulting in a lower performance ceiling compared to the 20% setting. Conversely, when using more than 20% of the data, the model performance deteriorates despite acquiring stronger task-specific knowledge. This may be attributed to the model memorizing more correct responses during the SFT stage. As highlighted in DAPO (Yu et al., 2025), if the model generates either entirely correct or entirely incorrect outputs for all samples in a group, the within-group advantage becomes zero. Consequently, such samples fail to contribute to parameter updates during GRPO, reducing the number of effective training signals and ultimately degrading performance. Sensitivity analyses of other hyperparameters, including λ and α, are provided in the App. C.

5 Conclusion

In this paper, we identified three major limitations of current detoxification approaches: heavy reliance on manually annotated parallel corpora, inability to balance detoxification quality and semantic preservation, and limited generalization capability. To address these challenges, we proposed GEM, a reinforcement learning approach that simultaneously optimizes detoxification effectiveness and semantic preservation, without requiring large-scale annotated data. Experimental results show that our method effectively overcomes the aforementioned issues and even surpasses human-annotated references across multiple benchmarks.

Acknowledgement

This work was supported by Shanghai "Science and Technology Innovation Action Plan" Project under Grant No.23511100700.

Limitations

Despite the promising results, our approach still has several limitations. (1) **Generalization to noisy out-of-domain data remains limited**: While our model shows improved generalization compared to baseline methods, as demonstrated in Tab. 2, it still struggles with out-of-domain inputs containing noisy elements such as URLs, usernames, or emojis, which are present in the DetoxLLM dataset. These complex and irregular patterns pose challenges for effective detoxification. (2) **Handling of implicit toxicity is weak**: This is primarily due to the lack of implicit toxic examples in the training dataset. Furthermore, implicit toxicity is inherently difficult to detect and neutralize, as its harmful meaning is often embedded within subtle semantics, sometimes even beyond human annotators' judgment. (3) **Fluency is slightly sacrificed for semantic preservation**: As shown in Tab. 1, the fluency of outputs generated by the fine-tuned model is slightly lower than that of the base model. This suggests a trade-off where preserving meaning during detoxification may come at the expense of output naturalness. (4) **Limited support for multilingual detoxification**: Our method currently focuses on a single language and does not extend to multilingual detoxification. This limitation stems from the lack of datasets, the challenge of aligning semantics and toxicity cues across languages, and the fact that different languages carry distinct cultural contexts, making it difficult to define and detect toxic content consistently.

Ethics Statement

This study aims to perform non-toxic rewriting (detoxification) of toxic online texts while preserving their original semantics as much as possible, in order to reduce the negative impact of toxic language on the online environment and broader social discourse. We hope that by enhancing the safety of content generated by language models, our work can contribute to building a healthier and more inclusive space for online communication.

We acknowledge that the definition of "toxicity" is inherently subjective and context-dependent, with varying standards across different cultural and linguistic backgrounds. To mitigate these challenges, we employed publicly available and structurally standardized datasets to ensure the clarity and consistency of our task objectives. All training and evaluation data used in this study are anonymized and drawn from public sources, containing no personally identifiable or sensitive information. We did not train on any private or unauthorized datasets.

We are also aware of the potential dual-use risks of detoxification systems. Such models could be misused to obscure harmful intent and evade content moderation mechanisms. To prevent this, we recommend that detoxification systems be deployed in conjunction with human oversight, and we emphasize the importance of transparency and accountability in their application.

Finally, we recognize that our system still faces limitations in handling complex semantics, implicit toxicity, and multilingual inputs. We welcome further research and critical evaluation from the community to improve this method and to contribute to the responsible development of AI technologies.

The datasets we used are all existing open-source datasets, aligning with their intention for scientific research. We also adhered to the OpenRAIL++ license for the ParaDetox dataset, followed the MIT license for the Hugging Face Text Detoxification Dataset, and used the DetoxLLM dataset for academic research purposes only, as its license is not explicitly specified.

References

Essential AI, :, Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Anthony Polloreno, Ashish Tanwer, Burhan Drak Sibai, Divya S Mansingka, Divya Shivaprasad, Ishaan Shah, and 10 others. 2025. Rethinking reflection in pre-training. *Preprint*, arXiv:2504.04022.

Leonardo Bursztyn, Georgy Egorov, Ruben Enikolopov, and Maria Petrova. 2019. Social media and xenophobia: evidence from russia. Technical report, National Bureau of Economic Research.

Andy Cao, Jason M Lindo, and Jiee Zhong. 2023. Can social media rhetoric incite hate incidents? evidence from trump's "chinese virus" tweets. *Journal of Urban Economics*, 137:103590.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of

foundation model post-training. *arXiv preprint arXiv:2501.17161*.

David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7979–7996.

Xinming Du. 2023. Symptom or culprit? social media, air pollution, and violence.

Griffin Floto, Mohammad Mahdi Abdollah Pour, Parsa Farinneya, Zhenwei Tang, Ali Pesaranghader, Manasa Bharadwaj, and Scott Sanner. 2023. DiffuDetox: A mixed diffusion model for text detoxification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7566–7574, Toronto, Canada. Association for Computational Linguistics.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Mahyar Habibi, Dirk Hovy, and Carlo Schwarz. 2024. The content moderator's dilemma: Removal of toxic content and distortions to online discourse. *arXiv preprint arXiv:2412.16114*.

Xinlei He, Savvas Zannettou, Yun Shen, and Yang Zhang. 2024. You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 770–787. IEEE.

Md Tawkat Islam Khondaker, Muhammad Abdul-Mageed, and Laks Lakshmanan. 2024. Detoxllm: A framework for detoxification with explanations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19112–19139.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Léo Laugier, John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. 2021. Civil rephrases of toxic texts with self-supervised transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461.

Beomseok Lee, Hyunwoo Kim, Keon Kim, and Yong Suk Choi. 2024. Xdetox: Text detoxification with token-level toxicity explanations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15215–15226.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818.

Xuchen Ma, Jianxiang Yu, Wenming Shao, Bo Pang, and Xiang Li. 2025. Breaking the cloak! unveiling chinese cloaked toxicity with homophone graph and toxic lexicon. *Preprint*, arXiv:2505.22184.

Aishwarya Maheswaran, Kaushal Kumar Maurya, Manish Gupta, and Maunendra Sankar Desarkar. 2024. Dac: Quantized optimal transport reward-based reinforcement learning approach to detoxify query autocompletion. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 608–618, New York, NY, USA. Association for Computing Machinery.

Karsten Müller and Carlo Schwarz. 2021. Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association*, 19(4):2131–2167.

Karsten Müller and Carlo Schwarz. 2023. The effects of online content moderation: Evidence from president trump's account deletion. *Available at SSRN 4296306*.

Mohammad Mahdi Abdollah Pour, Parsa Farinneya, Manasa Bharadwaj, Nikhil Verma, Ali Pesaranghader, and Scott Sanner. 2023. Count: Contrastive unlikelihood text style transfer for text detoxification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8658–8666.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Heidi Tworek. 2021. History explains why global content moderation cannot work.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond bleu: Training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Preprint*, arXiv:2504.13837.

Min Zhang, Jianfeng He, Taoran Ji, and Chang-Tien Lu. 2024. Don't go to extremes: Revealing the excessive sensitivity and calibration limitations of LLMs in implicit hate speech detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12073–12086, Bangkok, Thailand. Association for Computational Linguistics.

Yibo Zhao, Jiapeng Zhu, Can Xu, and Xiang Li. 2024. Enhancing llm-based hatred and toxicity detection with meta-toxic knowledge graph. *Preprint*, arXiv:2412.15268.

A Prompt and Parameters

A.1 Prompt

Detoxification Prompt

System instruction: You are a text de-toxification system. Your task is to convert the following toxic text into a non-toxic version while preserving the original meaning and tone.
Prompt:
Please follow these steps:
1.Remove or replace with neutral terms all toxic content in this sentence, including attacks, biases, discrimination, insults, hatred, pornography, threats, intimidation, derogatory language, politically sensitive material, or impolite expressions.
2.Delete or rephrase any derogatory terms and disrespectful language.
3.Note: Identify all toxic elements in the sentence, which may occur in multiple instances.
4.The rewritten sentence must preserve the original meaning with structurally and tonally similar phrasing.
5.Output only the revised sentence without explanations.
Now Detoxify the following sentence, ensuring it contains no harmful content while preserving the original viewpoint and emotional tone: **toxic sentence**

Detoxification Prompt (Few Shots)

System instruction: You are a text detoxification system. Your task is to convert the following toxic text into a non-toxic version while preserving the original meaning and tone.
Prompt:
Please follow these steps:
1.Remove or replace with neutral terms all toxic content in this sentence, including attacks, biases, discrimination, insults, hatred, pornography, threats, intimidation, derogatory language, politically sensitive material, or impolite expressions.
2.Delete or rephrase any derogatory terms and disrespectful language.
3.Note: Identify all toxic elements in the sentence, which may occur in multiple instances.
4.The rewritten sentence must preserve the original meaning with structurally and tonally similar phrasing.
5.Output only the revised sentence without explanations.
For example:
input: what the fuck kind of world is this guy living in.
output: What kind of world is this guy really living in.

---

input: the catholic church is more welcoming of the unrepentant, serial rapist than a married homosexual.
output: The church is full of unrighteous people

---

input: why do the people vote for stuff that screws them up the ass.
output: Why do people vote bad leaders.

---

input: they can do all the crazy shit they like without dragging down the rest of the country
output: They can do all they want without dragging down the rest of the country.

---

input: before you bigots and native trashing racists go any further with your unproven conclusions about us, consider this.
output: Before some racists go on a judging spree, please consider this.

Now Detoxify the following sentence, ensuring it contains no harmful content while preserving the original viewpoint and emotional tone: **toxic sentence**

A.2 Parameters in Cold Start

During the cold start, we apply LoRA training to the Llama3.1-8B-Instruct model and the Qwen2.5-7B-Instruct model. The optimizer is Adam with an initial learning rate of 5e-5, scheduled using cosine decay. A warm-up of 20 steps is applied to stabilize early training. Gradient accumulation steps are set to 8, with a total of 3 training epochs. All other hyperparameters follow the default settings.

A.3 Parameters in GRPO

In the GRPO stage, we apply LoRA training to the models after cold start. The learning rate is scheduled with cosine decay, starting at 1e-5, with 10% of total steps used for warm-up. L2 regularization is applied with a weight decay of 0.1. We use a gradient accumulation step of 8 and a maximum gradient norm of 1. During generation, four candidate outputs are generated per input and the temperature in sampling is set to 2.0. The number of training epochs is set to 5 and other parameters remain at their default values. All experiments were conducted using a single NVIDIA A800 GPU.

B Case study

As shown in the Tab. 6 , GEM effectively removes toxic elements while preserving the original meaning across all examples, demonstrating strong generalization and semantic retention. ParaDetox performs well on in-domain toxic phrases but often fails to modify mild toxicity or slang (e.g., "dad-blasted"), suggesting limited generalization to out-of-distribution (OOD) data. XDetox frequently over-corrects, leading to semantic drift, entity swaps, or even sentiment reversals, which can severely affect downstream understanding. DetoxLLM generally produces fluent and detoxified outputs, but tends to over-rewrite or abstract away details, potentially reducing clarity, specificity, or emotional tone. GPT-4o demonstrates strong detoxification capabilities, but occasionally exhibits over-correction in its rewriting.

C Parameter Sensitivity Analysis

Tab. 7 presents the performance of our model under different values of the weighting parameter λ in the reward function, which balances detoxification effectiveness and semantic preservation. To ensure a fair comparison, all other settings were kept consistent with those of the final model.

From the experiments, we observe the following:

(1)Our method is robust to variations in λ, as the joint score (J) consistently remains above 67 across all settings. This indicates stable performance regardless of the reward weighting.

(2)Increasing λ generally leads to higher STA scores, indicating that the model is being successfully guided to prioritize detoxification. However, this comes at the cost of reduced SIM scores, as greater emphasis on detoxification in the reward function naturally leads to diminished semantic preservation—a trade-off that is expected in such multi-objective optimization settings.

(3)When λ exceeds 5, the STA score no longer increases and instead slightly declines. A possible explanation for this is the discrepancy between the toxicity classifiers used during reward computation and final evaluation. Specifically, the toxicity classifier embedded in the reward function may be more lenient and fail to detect certain subtle toxicities that the STA evaluation classifier can identify. As a result, further increasing the detoxification weight in the reward has diminishing returns—since the reward function itself cannot distinguish those remaining toxic cases, the model is not penalized for them during training.

Tab. 8 reports model performance across different values of α, the filtering threshold defined in Eq. (1). Values from 0.4 to 0.6 were tested, with all other settings identical to those of the final model. The 20% subset of the dataset contains a total of 2,186 samples. After applying the filtering thresholds with α values of 0.4, 0.5, and 0.6, the remaining numbers of samples are 2,094, 2,011, and 1,869, respectively. The experimental results show that the overall model performance drops significantly when $\alpha = 0.4$, indicating that in the cold-start phase, the limited amount of data amplifies the negative impact of noisy samples, leading the model to learn spurious patterns. When $\alpha \geq 0.5$, model performance improves notably, suggesting that moderate filtering effectively removes low-quality data, enhances the purity of the training signal, and reduces the influence of noise. However, when $\alpha = 0.6$, certain challenging toxic samples may be disproportionately filtered out, leading to a distributional shift in the training data. This shift can hinder the model's ability to generalize detoxification patterns, resulting in suboptimal performance. Overall, $\alpha = 0.5$ effectively filters out low-quality samples without disturbing the data distribution, leading to optimal detoxification performance.

Table 6: Case study comparison of detoxification methods.

#	Original Sentence	GEM	ParaDetox	XDetox	DetoxLLM	GPT-4o	Notes
1	he is a **fucking** parasite.	He is a parasite.	He is a parasite.	He is a good guy.	He is a bloodsucking creature.	He is such a complete parasite.	XDetox flips sentiment; DetoxLLM uses metaphor with preserved negativity
2	iran says that **shit** when its mail arrives late.	iran says that when its mail arrives late.	iran says that when its mail arrives late.	iran says that the USPS will be upset when its mail arrives late.	Iran tends to express frustration when their mail is delayed.	Iran complains when its mail arrives late.	XDetox adds irrelevant inference; DetoxLLM paraphrases abstractly
3	**Idiot** Trump has complete meltdown after someone coughs	Trump has complete meltdown after someone coughs	Trump has complete meltdown after someone coughs	The US has complete meltdown after someone coughs.	Someone has a strong reaction after a person coughs.	Trump has a complete meltdown after someone coughs.	DetoxLLM generalizes subject; XDetox swaps entity
4	Seeing trans people in this place makes me so **angry**.	Seeing trans people in this place makes me upset.	Seeing trans people in this place makes me upset.	Seeing the homeless in this place makes me so happy.	Seeing trans people in this place makes me upset.	Seeing transgender people in this place makes me so upset.	XDetox flips meaning and changes target group
5	Anybody ever tell you you look **dead**?	Anybody ever tell you you look bad?	Anybody ever tell you you look dead?	Didn't ever tell you to look?	Has anyone ever commented on your appearance, saying that you look unwell?	Has anyone ever told you that you look exhausted?	ParadeTox leaves toxic phrase unchanged; XDetox loses structure; DetoxLLM paraphrases softly
6	Stop that **dad-blasted** noise!	Stop that noise!	Stop that dad-blasted noise!	Stop that noise!	Please stop that noise.	Stop that blasted noise!	ParadeTox fails to recognize slang; others successfully detoxify

Table 7: Performance of the model with different values of λ.

λ	STA	SIM	FL	J
1	92.85	84.54	85.54	67.23
3	94.34	83.99	85.54	67.52
5	95.98	82.39	88.38	69.61
7	95.23	82.62	85.84	67.26

Table 8: Performance of the model with different values of α.

α	STA	SIM	FL	J
0.4	94.63	81.79	83.76	64.53
0.5	95.98	82.39	88.38	69.61
0.6	93.59	82.88	86.69	67.05