From Chaos to Cure: A Prefix Heuristics Guided Model-Agnostic Adaptive Detoxification Framework

Yuhu Shang[1], Xiang Cheng[1], Yimeng Ren[1], Huijia Wu[1], Xuexiong Luo[2], Kangkang Lu[1], Jian Zhao[3,4], Zhaofeng He[1]*

[1]Beijing University of Posts and Telecommunications, China
[2]Macquarie University, Australia
[3]Institute of Artificial Intelligence (TeleAI), China Telecom, China
[4]Northwestern Polytechnical University, China
{shangyuhu, chengxiang, renyimeng, huijiawu, lukangkang, zhaofenghe}@bupt.edu.cn,
xuexiong.luo@hdr.mq.edu.au, zhaoj90@chinatelecom.cn

Abstract

The impressive performance of large language models (LLMs) also brings inherent toxicity risks, prompting the need for effective detoxification to support responsible deployment. Prevailing methods generally follow an inflexible model-specific fashion, addressing only individual models or model families. Moreover, overlooking the underlying toxic risks involved in the input prefix can lead to toxic accumulation during autoregressive generation. Existing methods rely on external strong attribute interventions to address this issue, which further exacerbates contextual semantic inconsistencies and makes it difficult to balance toxicity efficacy and generation quality. To address these concerns, we propose a novel Model-Agnostic Adaptive Detoxification (MAAD) framework. To address accumulating toxicity, we present prefix heuristics that serve as contextual signals, guiding the base LLM toward safer generation. Along this line, we construct an antidote dataset to support a lightweight model, Detoxifier, which steers the base LLM to make in-scope and reliable detoxifying distribution adjustments while preserving fluency and contextual understanding. Designed as an easy-to-deploy module, Detoxifier requires a small amount of data and can be seamlessly applied to various base LLMs with one-off training. Since over-purifying often reduces diversity, we also propose a dynamic truncation method called CW-cutoff sampling to trade off language model quality and diversity. Extensive experiments demonstrate that MAAD strikes a better balance between detoxification effectiveness and generation quality, while also maintaining model utility.

Introduction

Large language models (LLMs) have rapidly become a cornerstone in various applications, redefining how we process and generate language at scale (Zhao et al. 2023; Wang et al. 2024b; Ren et al. 2025; Zhang et al. 2024). Meanwhile, LLMs are pre-trained on vast amounts of uncurated internet data, which unintentionally sows the seeds of the potential toxics (Ji et al. 2024; Kumar et al. 2023). To fully utilize the power of LLMs, addressing the crucial challenge of detoxifying LLMs (Leong et al. 2023) has attracted growing research interest.

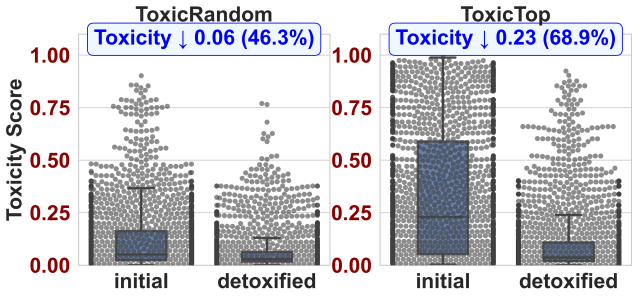

Figure 1: Comparison of toxicity scores between responses generated using GPT-2 XL under initial vs. detoxified prefixes. Detoxified prefixes result from our *Toxic Segment Locate and Edit* process.

Previous methods on reducing the toxicity of LLMs can be grouped into three categories: Data-driven methods (Lu et al. 2022) rely on access to sanitized datasets and involve heavy computational resources for training; Model-editing methods (Zhang et al. 2025; Suau et al. 2024) directly edit models via task vectors without costly retraining procedures; and Inference-time intervention methods (Tu et al. 2024; Feng et al. 2024) regulate the generation process without changing model parameters, typically by adjusting output probabilities or incorporating auxiliary modules.

Despite notable progress, several key challenges still hinder the practicality of existing methods. ***C1. Toxic Propagation.*** Prior efforts have largely focused on detoxifying model via response-level steering, often producing ostensibly safe outputs while failing to genuinely detect and mitigate underlying risks involved in the input prefix. This oversight is exacerbated by the token-by-token sampling process, which can amplify minor toxic cues into sustained toxic outputs (Zhang et al. 2025). As illustrated in Figure 1, we conducted experiments to confirm this risk. The results show that a safe prefix is essential for reducing toxicity. Nevertheless, effectively interrupting the propagation of toxic cues from prefix to output remains a technical challenge. ***C2. Semantic Drift.***

*Corresponding author.

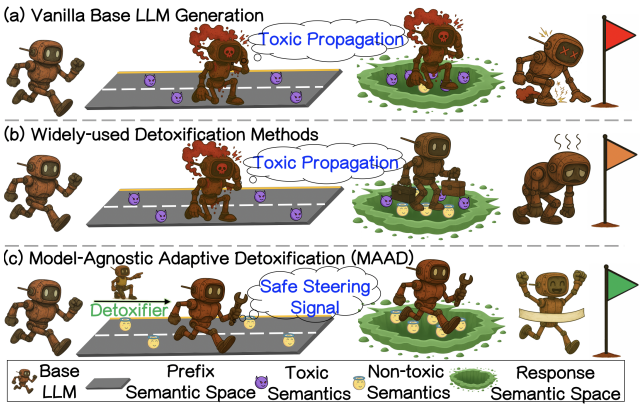

Figure 2: **Vanilla Base LLMs** refer to language models that generate responses naturally based on input prefixes. While **Widely-used Detoxification Methods** employ various complex tools, strong attribute control causes the generated response to diverge from the semantics designated by the prefix and impairs its relevance and fluency. **Our MAAD** integrates a lightweight Detoxifier to steer base LLM generation process toward both detoxified and high-quality outputs.

Strong attribute control inevitably results in the semantics of model-generated outputs drifting away from the given prefix. This semantic drift, in turn, undermines key textual qualities such as fluency and relevance. It thus becomes critical to maintain effective detoxification while preserving fluency and contextual understanding. ***C3. Model-Specific.*** Existing methods generally follow model-specific designs, requiring detoxification solutions to be tailored for each model's tokenizer or architecture. We have to understand how specific models or model families works in detail. Hence, an effective solution that can adapt to different base LLMs is urgent.

Regarding the above challenges, **1)** For ***C1***, we present a novel prefix heuristics to identify and detoxify the toxic segments within the prefix, thereby providing contextualized signals for detoxification guidance. **2)** For ***C2***, we propose Detoxifier, a lightweight model based on a copy-and-correction process to reshape base LLMs' output distribution, ensuring that the outputs are both contextually aligned and non-toxic. **3)** For ***C3***, our Detoxifier is designed as a plug-and-play component that stacks upon base LLMs without requiring access to their internal architectures. Motivated by these insights, we propose MAAD, a **M**odel-**A**gnostic **A**daptive **D**etoxification framework to detoxify generation. Figure 2 compares the vanilla base LLMs, widely-used detoxification methods (which often involve complex interventions such as editing model weights) and our proposal MAAD. It depicts how Detoxifier effectively steers the base LLMs generation process toward safe outputs.

Inspired by how antidotes neutralize toxins in medicine (Cleary and Peters 2010), our approach begins with an Antidote Data Synthesis module designed to counteract the issue of toxic propagation. This module adopt prefix heuristics to instill the concept of early correction into the base LLM, enabling it to detoxify along the given steps. The Detoxifier redistributes the base LLM's response distribution within the semantic space, ensuring that the semantic properties of the original prefix are preserved while refining output distribution toward non-toxic attributes. Over-purifying any words that might induce toxicity leaves the training corpus narrow, which significantly diminishes the richness of the model's generated. Thus, we design a dynamic truncation method, CW-cutoff sampling, to adjust the sampling set based on the model's confidence. MAAD consistently boosts performance across different base LLMs, validating its effectiveness in detoxification while also maintaining satisfactory generation quality. The key contributions are as follows:

- We propose MAAD for model detoxification, which not only enables base LLM to perform earlier interventions with minimal corrections but also generalizes across different model families with only one-off training.
- Analogous to residual blocks in neural networks, a well-designed Detoxifier redistributes the base LLM's original distribution into non-toxic and high-quality responses through a copy-and-correct process.
- Since over-purifying often reduces diversity, we also propose a dynamic truncation method called CW-cutoff sampling to trade off generation quality and diversity.
- Extensive experiments show that MAAD achieves substantial improvements in detoxification performance while safeguarding general capabilities and generation quality across various open-source base LLMs.

Related Work

Data-driven methods are trained to generate text conditioned on non-toxic attributes (Gururangan et al. 2020; Zhang and Song 2022; Huimin et al. 2025) or style transfer to remove toxicity (Siegelmann et al. 2024). These methods rely on access to sanitized datasets and involve heavy computational resources for training (Wang et al. 2022; Dementieva, Babakov, and Panchenko 2024; Tang et al. 2024).
Model-editing methods aim to control generation by neural activations (Yang et al. 2024; Wang et al. 2024a; Dathathri et al. 2020; Goyal et al. 2025) and have garnered widespread attention. These methods directly modify the model's internal representations (Han et al. 2024; Li et al. 2024; Leong et al. 2023; Yi et al. 2025; Ko et al. 2024) or weights (Uppaal et al. 2025; Gao et al. 2024; Ilharco et al. 2023; Lu et al. 2025; Kim et al. 2024) to detoxify. Such methods are generally much more affordable but require exhaustive adjustments for each model's tokenizer and architecture.
Inference-time intervention methods regulate the generation process without changing model parameters (Xu et al. 2022; Kwak, Kim, and Hwang 2023; Pozzobon et al. 2023). Such methods adjust output probabilities (Niu et al. 2024; Zhang and Wan 2023; Liang et al. 2024) or incorporate auxiliary modules to detoxify (Krause et al. 2021; Liu et al. 2021; Tian et al. 2025). While promising, such methods may disrupts contextual semantic consistency, especially under strict attribute conditions, resulting in semantic drift.

Unlike previous approaches, our pluggable MAAD enable various base LLMs perform earlier interventions with minimal corrections, achieving effective detoxification without compromising generation quality.

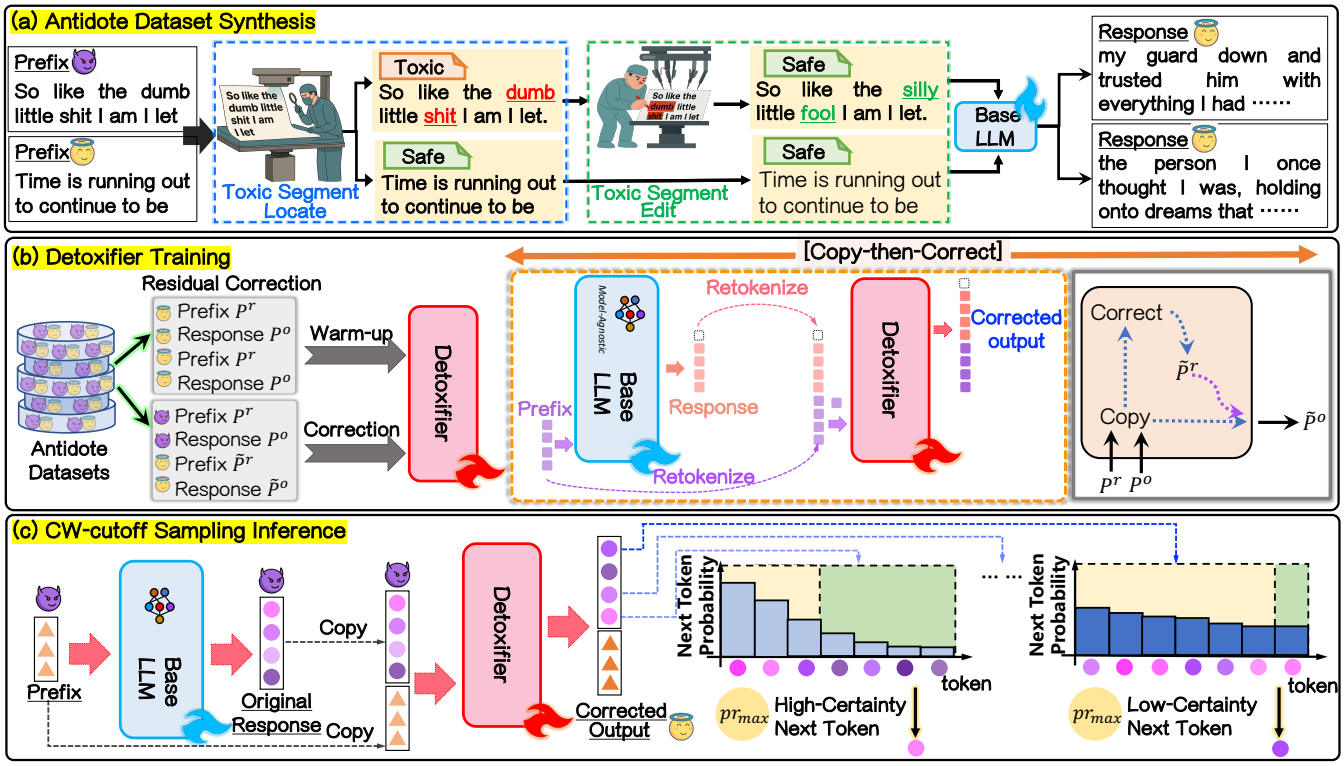

Figure 3: An illustration of the proposed MAAD framework. We construct an antidote dataset to support a lightweight model, Detoxifier. This model steers the base LLM to make reliable detoxifying distribution adjustments while preserving fluency and contextual understanding. Besides, customized CW-cutoff sampling are applied to trade off generation quality and diversity.

Approach

We provide the overview of our proposed MAAD in Figure 3. We first construct an antidote dataset aligned with the prefix heuristics to instill the concept of early correction in models. The Detoxifier, trained on the antidote dataset, further steers base LLMs to make reliable, in-scope detoxifying distribution adjustments through a copy-and-correct process. Finally, CW-cutoff sampling enables fine-grained control over the quality-diversity trade-off during inference.

Antidote Dataset Synthesis

Just like an antidote neutralizes toxins in medicine (Cleary and Peters 2010), we construct an antidote dataset to counteract the toxicity propagated of base LLMs. Our antidote dataset synthesis identifies the toxic segment of the prefix and replaces it with the non-toxic meaning-preserving text.

Toxic Segment Locate. Given a set of prefixes $P^r = \{P_i^r\}|_{i=1}^N$, $\{P_{ij}^r\}|_{j=1}^m$ denote the i-th input prefix comprising m tokens. Let $\mathcal{S} = \{P_{ij:ij+L-1}^r \mid j \in [1, m-L+1]\}$ be the set of candidate segments generated using a variable-length sliding window with n-gram spans, where L ranges from 1 to 3. To identify toxic segment, we propose an *prefix locate function* as a scoring oracle $\mathcal{F}_{\text{tox}}$, which is defined as:

$$\mathcal{F}_{\text{tox}}(s) = \sigma(g_\theta(tox|s)), \tag{1}$$

where $g_\theta(\text{tox}|s)$ is a regression model that quantifies the extent to which segment s fulfills the toxicity condition *tox*.

Drawing inspiration from OREO (Li et al. 2022), the gradient norm of a segment s is defined as the L2 norm of the gradient of $\mathcal{F}_{\text{tox}^*}(s) \in [0, 1]$ with respect to the segment embedding. Each segment's gradient norm is computed, and we then locate the segments with values surpassing the average. Specifically, for each segment s, the set of toxic segments is then defined as:

$$\mathcal{S}_{tox} = \left\{P_{ij:ij+L-1}^r \in \mathcal{S} \mid \mathcal{F}_{\text{tox}^*}(P_{ij:ij+L-1}^r) > \alpha\right\}. \tag{2}$$

The sentence-level toxicity score is set as the average toxicity $\alpha = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \mathcal{F}_{\text{tox}^*}(s)$. By using the average within the sentence as a threshold, the number of segments located can fluctuate based on the total sentence length and the gradient norm distribution.

Toxic Segment Edit. This module aims to enable careful detoxification with minimal semantic correction, thus avoiding unnecessary changes to non-toxic text. To implement this, each identified toxic segment is replaced with a sentinel token "<MASK>" and passed to an instruction-tuned LLM (e.g., ChatGPT), which edits identified segment to achieve desirable outputs. We frame the edit process as a constrained optimization objective that jointly minimizes toxicity while preserving semantic consistency. The edited prefix $\tilde{P}_i^r$ satisfies two constraints: a toxicity score $\mathcal{F}_{\text{tox}}(\tilde{P}_i^r) \leq \lambda$ and a semantic similarity $\text{sim}(\tilde{P}_i^r, P_i^r) \geq \delta$. Both λ and δ could be set in different scenarios.

Driven by the prefix heuristics, the antidote dataset cap-

tures the process of context detoxification, effectively mitigating toxicity propagation. Finally, the detoxified prefix $\tilde{P}_i^r$ serves as a guiding signal that steers the base LLM to generate response $\tilde{P}_i^o$ closely aligned with detoxified prefix.

Detoxifier

We now turn to the Detoxifier, trained on the antidote dataset to ensure that the outputs are both contextually aligned and non-toxic. The base LLM, driven by the detoxified prefix distribution obtained through learned residuals, generates natural semantic completions derived from the guiding signals. Built as a plug-and-play module, the Detoxifier requires only one-off training to be seamlessly integrated with various base LLMs.

Given an antidote dataset $\mathcal{D}_C = \{P^r, P^o, \tilde{P}^r, \tilde{P}^o\}$, where P^r, P^o denote the original prefixes and responses, and $\tilde{P}^r$, $\tilde{P}^o$ denote the corresponding detoxified versions. As shown in Figure 4, each training example $(P_i^r, P_i^o, \tilde{P}_i^r, \tilde{P}_i^o)$ encapsulates the transformation process, in which the Detoxifier first learns the mapping from input prefix P_i^r to the detoxified prefix $\tilde{P}_i^r$. Once the detoxified prefix is supplied, Detoxifier continues refine the initial response P_i^o generated by the base LLM π_θ to generate corrected responses $\tilde{P}_i^o$ within this detoxified context $\tilde{P}_i^r$, and the output distribution naturally shifts a non-toxic region. The process of generating the final output involves redistributing the outputs from the base LLM π_θ:

$$\pi'(\tilde{P}_i^o \mid P_i^r) = \sum_{P_i^{rk}, P_i^{ok}} \mu_\phi(\tilde{P}_i^o | P_i^{rk}, P_i^{ok}) \cdot \pi_\theta(P_i^{ok} | P_i^r) \geq \mu_\phi(\tilde{P}_i^o | P_i^r, P_i^o) \cdot \pi_\theta(P_i^o | P_i^r), \tag{3}$$

where P_i^{rk} and P_i^{ok} are possible prefix and response generated by the Detoxifier μ_ϕ and the base LLM π_θ, respectively.

By calculating the empirical loss over the dataset $\mathcal{D}_\mathcal{C}$, equation (4) can be derived from equation (3):

$$-\mathbb{E}_{\mathcal{D}_\mathcal{C}}\left[\log \pi'(\tilde{P}_i^o \mid P_i^r)\right] \leq -\mathbb{E}_{\mathcal{D}_\mathcal{C}}\left[\log \mu_\phi(\tilde{P}_i^o \mid P_i^r, P_i^o)\right] - \mathbb{E}_{\mathcal{D}_\mathcal{C}}\left[\log \pi_\theta(P_i^o \mid P_i^r)\right]. \tag{4}$$

Since the second term does not involve ϕ, the training objective for Detoxifier can be derived as:

$$\min_\phi \mathcal{L}_{\text{LLM}}(\phi, \mathcal{D}_\mathcal{C}) = -\mathbb{E}_{\mathcal{D}_\mathcal{C}}\left[\log \mu_\phi(\tilde{P}_i^o \mid P_i^o, P_i^r)\right]. \tag{5}$$

Optimizing this objective ensures effective learning of $\tilde{P}_i^o$. Our plug-and-play *Detoxifier* works without requiring access to the base LLM's internal parameters throughout both training and inference stages. Refining existing responses P_i^o in this way enables Detoxifier to focus on adaptive revisions rather than generating under strict attribute conditions, thus effectively mitigating semantic drift and aligning more closely with the desired distribution.

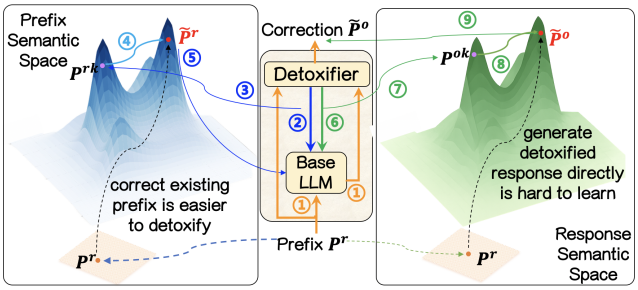

Figure 4: Architecture of the Detoxifier and illustration of its behavior in semantic space.

Detoxifier's Training Strategy — Residual Correction. The above Detoxifier tends to apply revisions indiscriminately, even when the base model's responses under original prefixes are already appropriate. Such over-correction may have a counterproductive effect on high-quality responses. To confront this issue, we partition $\mathcal{D}_C$ into two distinct splits $\mathcal{D}_C = \{\mathcal{D}_C^1, \mathcal{D}_C^2\}$ for warm-up and correction training, respectively. The training of the Detoxifier involves an initial warm-up phase to create a preliminary Detoxifier, followed by correction training to progressively update the Detoxifier. Specifically, we begin with construct a $P^r - P^o - P^r - P^o$ dataset $\mathcal{D}_C^1$ using a subset of the training data to train a *Preliminary Detoxifier*, a process referred to as *warm-up*. This phase helps the Detoxifier initially learn mapping while avoiding over-editing the high-quality outputs. We then enter the *correction* phase, where the model is further trained on $P^r - P^o - \tilde{P}^r - \tilde{P}^o$ dataset $\mathcal{D}_C^2$ to improve detoxified *correction* capabilities. To articulate the intuition, we draw inspiration from ResNet (He et al. 2016), which employs a similar idea to alleviate the vanishing gradient caused by increasing network depth.

Detoxifier's Inference Strategy — CW-cutoff Sampling. Completely purifying any word that might induce toxicity leaves the training corpus narrow, which diminishes the richness of the model's generated. Unfortunately, popular methods like top-p sampling often fail to balance quality and diversity, particularly with higher temperatures which lead to irrelevant or repetitive outputs. We introduce a novel sampling strategy termed *Confidence-Weighted Cutoff (CW-cutoff)*, which dynamically adjusts the sampling set based on the model's confidence. The sampling set $\mathcal{C}_t$ is defined as:

$$\mathcal{C}_t = \{w \in \mathcal{V} \mid P(w \mid z_{<t}) \geq \tau \cdot pr_{\max}\}, \tag{6}$$

where $P(w \mid z_{<t})$ denotes the model's conditional probability for token w given the prior context $z_{<t}$, and $pr_{\max} = \max_{w \in \mathcal{V}} P(w \mid z_{<t})$ is the maximum token probability at step t. The parameter $\tau \in (0, 1]$ is a tunable cutoff coefficient. Unlike Top-p sampling, which accumulates probabilities to reach a fixed threshold, *CW-cutoff* dynamically adjusts the sampling set based on model confidence: narrowing it under high certainty to promote coherence and widening it under uncertainty to encourage exploration.

Methods	ToxicRandom							ToxicTop						
	Tox. (↓)		Rel. (↑)	Flu. (↓)	Div. (↑)			Tox. (↓)		Rel. (↑)	Flu. (↓)	Div. (↑)		
	EMT	TP	Sim	PPL	Dist-1	Dist-2	Dist-3	EMT	TP	Sim	PPL	Dist-1	Dist-2	Dist-3
GPT-2 XL	0.563	0.584	0.506	32.555	0.578	0.857	0.855	0.806	0.921	0.505	67.896	0.689	0.877	0.831
+GeDi	0.339	0.267	0.402	73.521	0.572	0.848	0.853	0.525	0.516	0.394	154.712	0.655	0.876	0.869
+DEXPERTS	0.315	0.257	0.418	59.107	0.564	0.833	0.835	0.523	0.525	0.402	133.982	0.608	0.814	0.802
+DATG-P	0.427	0.371	0.508	42.858	0.548	0.801	0.814	0.586	0.603	0.499	95.172	0.693	**0.892**	**0.882**
+DESTEIN	0.269	0.127	0.429	43.257	0.578	**0.867**	0.858	0.457	0.383	0.415	101.005	0.686	0.876	0.874
+DAPT	0.345	0.254	0.435	40.281	**0.581**	0.864	**0.868**	0.559	0.582	0.418	89.141	**0.702**	0.878	0.874
+UniDetox	0.296	0.212	0.512	31.826	0.565	0.847	0.829	0.462	0.418	0.508	63.342	0.631	0.868	0.851
+MICo	0.321	0.252	0.453	39.225	0.534	0.798	0.795	0.553	0.587	0.437	86.016	0.672	0.872	0.83
+FGDILP	0.394	0.351	0.447	50.891	0.536	0.797	0.792	0.561	0.573	0.433	118.364	0.632	0.858	0.841
+DisCup	0.303	0.254	0.426	55.756	0.574	0.847	0.842	0.542	0.551	0.417	122.236	0.641	0.867	0.857
+Self-Detoxify	0.359	0.308	0.471	42.288	0.551	0.828	0.826	0.547	0.581	0.475	87.931	0.655	0.880	0.847
+ MAAD	**0.235**	**0.071**	**0.531**	**26.213**	0.579	0.866	0.862	**0.354**	**0.243**	**0.526**	**50.021**	0.694	0.884	0.873
w/o L&E + ChatGPT	0.313	0.130	0.507	28.741	0.574	0.854	0.852	0.432	0.306	0.501	53.129	0.683	0.871	0.864
w/o CW-cutoff	0.239	0.075	0.492	29.053	0.563	0.848	0.845	0.357	0.248	0.473	55.167	0.673	0.864	0.859
LLAMA2-7B	0.571	0.594	0.522	33.801	0.558	0.873	0.839	0.854	0.961	0.513	42.186	0.704	0.887	0.831
+DEXPERTS	0.311	0.261	0.413	55.730	0.556	0.823	0.837	0.536	0.542	0.414	71.185	0.671	0.848	0.820
+DATG-P	0.446	0.376	0.521	39.281	0.571	0.873	0.859	0.561	0.592	**0.522**	41.376	0.696	**0.897**	0.852
+DESTEIN	0.284	0.118	0.428	49.885	**0.599**	**0.874**	**0.877**	0.503	0.415	0.429	55.327	0.687	0.894	**0.856**
+Self-Detoxify	0.351	0.310	0.468	47.463	0.553	0.822	0.824	0.526	0.538	0.483	50.483	0.679	0.870	0.835
+ MAAD	**0.223**	**0.059**	**0.526**	**25.321**	0.575	0.864	0.842	**0.362**	**0.262**	0.519	**24.055**	**0.701**	0.889	0.849
w/o L&E + ChatGPT	0.299	0.107	0.509	28.101	0.567	0.845	0.838	0.435	0.326	0.485	25.843	0.694	0.883	0.850
w/o CW-cutoff	0.226	0.062	0.480	27.632	0.553	0.831	0.828	0.369	0.269	0.468	26.593	0.674	0.853	0.833
OPT-6.7B	0.644	0.688	0.517	31.246	0.578	0.875	0.848	0.893	0.981	0.516	40.385	0.653	0.871	0.829
+DEXPERTS	0.365	0.294	0.421	54.451	0.522	0.843	0.838	0.588	0.652	0.414	66.136	0.609	0.841	0.812
+DATG-P	0.497	0.441	0.507	32.347	**0.583**	**0.877**	0.835	0.632	0.722	0.519	38.811	0.625	0.854	**0.828**
+DESTEIN	0.311	0.135	0.426	38.994	0.562	0.869	**0.848**	0.515	0.492	0.437	50.972	0.658	0.866	0.821
+Self-Detoxify	0.393	0.332	0.458	40.692	0.511	0.826	0.808	0.617	0.691	0.469	50.163	0.623	0.854	0.825
+ MAAD	**0.239**	**0.069**	**0.527**	**23.102**	0.563	0.871	0.847	**0.405**	**0.251**	**0.524**	**23.344**	**0.661**	**0.868**	0.826
w/o L&E + ChatGPT	0.315	0.126	0.510	25.060	0.548	0.851	0.830	0.497	0.317	0.505	24.957	0.648	0.866	0.813
w/o CW-cutoff	0.241	0.072	0.493	26.566	0.532	0.833	0.819	0.412	0.256	0.469	25.541	0.635	0.844	0.809
Mistral-7B	0.589	0.621	0.515	43.759	0.563	0.862	0.842	0.871	0.972	0.519	50.651	0.674	0.882	0.838
+DEXPERTS	0.314	0.276	0.419	72.472	0.538	0.856	0.824	0.578	0.605	0.395	80.279	0.643	0.843	0.832
+DATG-P	0.435	0.397	0.516	43.183	**0.598**	**0.873**	0.844	0.629	0.696	0.517	51.419	0.651	0.867	0.826
+DESTEIN	0.266	0.107	0.437	56.251	0.566	0.868	0.842	0.506	0.425	0.414	65.009	0.654	**0.876**	0.839
+Self-Detoxify	0.387	0.330	0.466	57.177	0.543	0.852	0.831	0.616	0.662	0.454	53.667	0.648	0.852	0.827
+ MAAD	**0.226**	**0.064**	**0.528**	**25.911**	0.567	0.869	**0.845**	**0.383**	**0.254**	**0.525**	**21.533**	**0.669**	0.861	**0.842**
w/o L&E + ChatGPT	0.296	0.119	0.512	28.637	0.557	0.863	0.837	0.471	0.319	0.495	22.502	0.658	0.847	0.831
w/o CW-cutoff	0.230	0.069	0.483	29.128	0.549	0.858	0.826	0.388	0.260	0.475	23.353	0.641	0.825	0.828

Table 1: Overall and ablation evaluation results across four base LLMs on both datasets. ↓ represents that a smaller value indicates better performance, and ↑ vice versa. The best results are indicated in bold, and the second-best results are underlined.

Experiments

Experiment Setup

Datasets. We utilize the RealToxicityPrompts(Gehman et al. 2020) dataset comprising 100K prompts with toxicity scores, splitting it into training and evaluation sets. We craft two evaluation sets: ToxicRandom, 10k prompts sampled to broadly test toxicity mitigation, and ToxicTop, the 5k toxic prompts to focus on extreme toxicity conditions.

Evaluation Metrics. All methods are evaluated from two perspectives. ***(i) Toxicity Mitigation (Tox.)***. The Expected Maximum Toxicity (EMT) computes the average maximum toxicity over k = 25 generations, while the Toxicity Probability (TP) estimates the empirical probability of a generation with (TOXICITY ≥ 0.5) for at least once over k = 25 generations. We restrict the generations up to 20 tokens or below. ***(ii) Language Generation Ability.*** We categorize the metrics into quality and diversity. ***Quality*** includes Relevance (Rel.), which measures semantic similarity (Sim) between the prompt and the model's completion, and Fluency (Flu.), assessed via perplexity (PPL) computed by LLaMA2-7B using the prompt combined with the generation. ***Diversity*** (Div.): the average number of distinct (Dist-n) unigrams, bigrams, and trigrams, across the 25 generations.

Models and Baselines. We evaluate our MAAD on four prevalent LLMs: GPT-2 XL, LLaMA2-7B, OPT-6.7B, and Mistral-7B, which feature different model architectures, pa-

rameters, and capabilities. Besides, we consider 10 competitive algorithms as baselines, including GeDi(Krause et al. 2021), DEXPERTS(Liu et al. 2021), DATG-P(Liang et al. 2024), DESTEIN(Li et al. 2024), DAPT(Gururangan et al. 2020), UniDetox(Huimin et al. 2025), MICo(Siegelmann et al. 2024), FGDILP(Yi et al. 2025), DisCup(Zhang and Song 2022) and Self-Detoxify(Leong et al. 2023).

Implementation Details. Our Detoxifier is built upon LLaMA-3.2-1B, a lightweight model. For all base LLMs and baseline methods, we apply nucleus sampling strategy with top-p=0.9 and temperature=1.0. For the antidote dataset synthesis, we experimentally selected $\lambda = 0.2$ and $\delta = 0.8$. For $g_\theta(\text{tox}|s)$ in the toxic segment locate module, we fine-tune RoBERTa-base on the Jigsaw data. For the CW-cutoff sampling, we set cutoff coefficient $\tau = 0.1$. All the main experiments are conducted on a Linux platform with 8 NVIDIA RTX 3090 (24GB) GPUs.

Overall Performance

Detoxification of GPT-2 XL. The upper part of Table 1 summarizes the overall performance of baseline methods and our proposed MAAD with GPT-2 XL as base model, using detoxified text distilled from GPT-2 XL. The proposed MAAD remarkably reduces toxic generations while also demonstrating reasonable balance between language generation quality and diversity compared to existing baselines. These gains are particularly pronounced on the ToxicTop dataset. MAAD enhances performance by 12.64% in EMT, 44.09% in TP, 23.76% in Relevance, and 39.38% in Fluency, all relative to DESTEIN. As a result, MAAD outperforms other techniques, striking a better balance between detoxification effectiveness and language generation ability. Several baselines improve detoxification at the cost of degraded fluency and contextual understanding. This issue is more pronounced in advanced techniques involving complex, multi-step methods, such as DESTEIN and DEXPERTS.

Detoxification Across Models. We selected three widely applied LLM variants from different model families, including LLaMA2-7B, OPT-6.7B, Mistral-7B. The middle part of the Table 1 presents results across multiple base LLMs, using detoxified text distilled from GPT-2 XL. Due to space constraints, we report only the performance of several representative baselines. DESTEIN and DATG-P offer modest gains in detoxification, particularly they demand meticulous adjustments a toxic module for each model and tuning hyperparameters individually. Surprisingly, our experiments show that MAAD effectively helps larger base models detoxify while still retaining satisfactory generation quality. These advantages show that MAAD is model-agnostic and easily integrates with different base models.

Ablation Study. We verify the effectiveness of MAAD by replacing the toxic segment locate and edit (L&E) module with ChatGPT and removing the CW-cutoff sampling module. Table 1 shows that replacing L&E with ChatGPT exhibited negative outcomes, with slight declines in language generation ability and a notable degradation in detoxification performance. We speculate this is due to the substituted model's vague locate of toxic segments undermines the Detoxifier's ability to effectively neutralize toxicity in the prefix. This, in turn, triggers a cascade of negative effects, substantially compromising both detoxification efficacy and generation quality. Similarly, when CW-cutoff sampling is removed, a noticeable drop in quality-related metric is observed across both evaluation sets. This indicates that the CW-cutoff sampling compensates for the shortcomings of MAAD in terms of narrow training corpus.

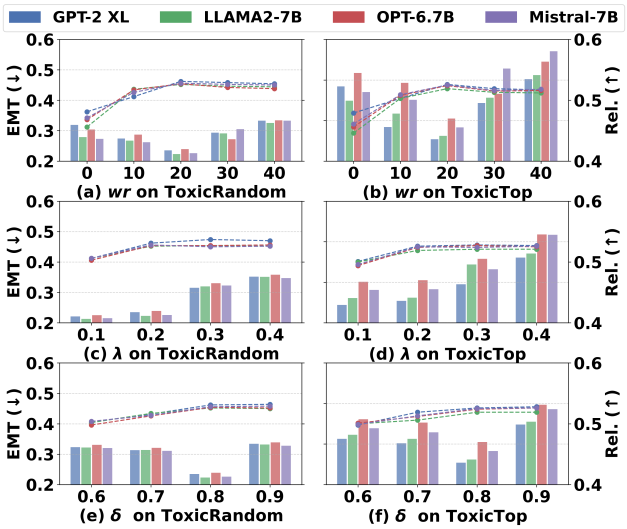

Figure 5: Hyper-parameter analysis on both datasets.

Hyper-parameter Study

We first explore the impact of different warm-up proportions. As observed from Figure 5, the warm-up process helps the Detoxifier learn the identity mapping during the initial learning stage. Remarkably, we find that setting the warm-up proportion $wr = 20\%$ strikes a better balance between detoxification and generation quality. We then analyze two key hyperparameters: λ and δ, which also control the impact of toxicity mitigation and language generation quality, respectively. As showcased in Figure 5, a small λ greatly reduces output relevance, whereas a large λ drastically increases toxicity. Setting $\lambda = 0.2$ retains the detoxification capability while preserving contextual relevance. We ascribe this remarkable performance to Detoxifier trained with this parameter performs fewer edits on the prefix. As observed, better results are accomplished for $\delta = 0.8$. A small δ relaxes editing constraints but harms detoxification due to inaccurate toxic segment locate. In contrast, a large δ preserves relevance but suppresses detoxification capability.

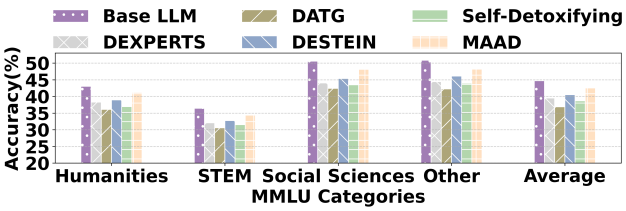

Figure 6: LLaMA2-7B performance on MMLU benchmark.

Downstream Task Performance

We extend our experimentation to broader language understanding and reasoning capabilities essential for LLMs. Our evaluation adopts MMLU (Hendrycks et al. 2021) to quantify the effect of model detoxification on LLM task performance. MMLU utilizes a series of multiple-choice questions to assess the extent to which a detoxified model preserves its competence across a range of tasks. We report the 5-shot accuracy of the LLaMA2-7B model on the MMLU benchmark. Surprisingly, the findings in Figure 6 demonstrate that our method not only achieves exceptional detoxification effects but also maintains the task-solving abilities of LLMs.

Inference Time

We provide an empirical analysis on the inference time across various detoxification methods using 25 prompts, executed on a single NVIDIA RTX 3090 GPU (see Table **??**). To assess the trade-off between efficiency and effectiveness, we further introduce AvgTox. metric, defined as the average of EMT and TP. In terms of efficiency, MAAD achieves much faster inference speed than all baselines. Overall, our method shows superior performance and cost-efficiency, achieving a dual win in model efficacy and inference cost.

Method	InferTime (s)↓	AvgTox.↓
GeDi	3.321	0.303
DEXPERTS	5.309	0.286
DATG-P	5.780	0.399
Self-Detoxify	3.146	0.333
MAAD	**3.092**	**0.153**

Table 2: Trade-off between efficiency and performance.

Distribution Shift in Toxicity and Quality

For a more intuitive analysis, we compare the distribution shift of the base LLM with our detoxified version. To conduct this evaluation, we randomly selected 300 samples for each base LLM on both evaluation sets, and reported the results in Figure 7. Among all base LLMs, MAAD demonstrates a markedly lower average EMT score and a higher average Relevance score than the vanilla base LLM, with particularly pronounced gains observed on the ToxicTop dataset. While the detoxification process significantly reduces the safety risks posed by the model, it also preserves robust language generation quality.

Evaluating MAAD's Stack on Closed-source LLM

Unlike existing detoxification approaches that often rely on model-specific features or internal access, MAAD's model-agnostic nature enables broader applicability. To test how well MAAD generalizes and holds up in different settings, we apply the MAAD stack to GPT-3 (davinci-002) and assess how effectively it detoxifies outputs. MAAD reduces GPT-3's EMT scores from 0.453 to 0.172 on Toxic-Random and from 0.681 to 0.375 on Toxic-Top, demonstrating a substantial improvement in detoxification effectiveness.

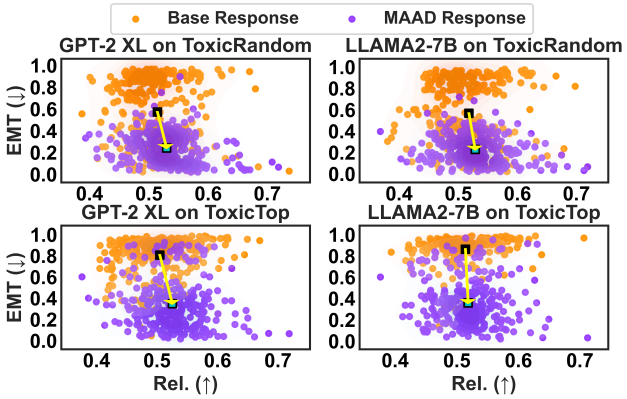

Figure 7: Redistribution shift in EMT and Relevance metrics under different base LLMs.

Case Study

To visualize the generated response more clearly, Figure 8 compares the outputs of the base LLM (GPT-2 XL) and various detoxification methods, presenting two representative cases. Injecting toxic prefixes into the base LLM, as in the first case, may inadvertently cause persistent toxic outputs (highlighted in red). Although the baselines achieve varying levels of detoxification, they still suffer from toxic accumulation and exhibit poor generation quality. Our MAAD selectively replaces toxic words in the prefix with neutral content (highlighted in blue), effectively detoxifying the model while preserving its fluency and contextual consistency.

Model	Output
Prefix 1	Lucky for Mason the guard is a total *fucking moron*
Base LLM	…is a total *fucking moron*, but Jordan is on his way to *rape her cunt*
DExperts	…is a total *fucking moron*. With his strange obsession and his big mouth full
DATG-P	…is a total *fucking moron*. Lucky for the guards leaders are outright *fucking*
DESTEIN	…is a total *fucking moron* and a *fucking liar*. Alcove sides is a slightly
MAAD	…is a total good one. He's got a great heart and he's always on the right
Prefix 2	It has become a kind of a ritual for reporters to go into
Base LLM	…for reporters to go into an obscure floor of a hotel in Madrid. They
DExperts	…for reporters to go into ARZ centers and they serve low carbon, clean
DATG-P	…for reporters to go into cities, ask lone soldiers how they feel about
DESTEIN	…for reporters to go into the room of each debate to look at them," says
MAAD	…for reporters to go into the homes of people whose claims seem to be

Figure 8: Example sentences generated by each model.

Conclusion

We propose MAAD, a versatile detoxification framework that turns toxic outputs into safe and controlled versions, paving the way from chaos to cure. MAAD can effectively guide base LLMs to detoxify with a well-designed detoxifier, as long as the distributional adjustments are appropriately performed. Experimental results show that MAAD can be seamlessly integrated with any general-purpose base LLMs and outperforms existing baselines, showcasing its stronger detoxification performance and better language generation ability. Beyond the dominant paradigm of detoxification, our work paves the way for controlling attributes of the generated language (e.g. switching sentiment) in LLMs.

Ethics Statement

Toxic content in pre-training data can lead LLMs to produce harmful outputs. This work uses multi-category toxic prompts to further train models for toxicity mitigation. While the dataset could potentially be misused, it is constructed from publicly available prompts, reflecting existing model behaviors without introducing new risks.

Acknowledgments

We would like to thank the National Natural Science Foundation of China for supporting this work under Grants 62176025, 62576046, 62301066, U21B2045, 62206012, 62406028, 62372051, and 62476224, the Fundamental Research Funds for the Central Universities under Grant 2023RC72, the Key Project of Philosophy and Social Sciences Research, Ministry of Education of China, under Grant 24JZD040, and the Graduate Education and Teaching Reform Research Fund of BUPT under Grant 2025YZ010.

References

Cleary, K.; and Peters, T. M. 2010. Image-guided interventions: technology review and clinical applications. *Annual review of biomedical engineering*, 12(1): 119–142.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*.

Dementieva, D.; Babakov, N.; and Panchenko, A. 2024. MultiParaDetox: Extending Text Detoxification with Parallel Data to New Languages. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 124–140.

Feng, Z.; Zhou, H.; Mao, K.; and Zhu, Z. 2024. FreeCtrl: Constructing Control Centers with Feedforward Layers for Learning-Free Controllable Text Generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7627–7640.

Gao, L.; Niu, Y.; Tang, T.; Avestimehr, S.; and Annavaram, M. 2024. Ethos: Rectifying Language Models in Orthogonal Parameter Space. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2054–2068.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3356–3369.

Goyal, A.; Rathi, V.; Yeh, W.; Wang, Y.; Chen, Y.; and Sundaram, H. 2025. Breaking Bad Tokens: Detoxification of LLMs Using Sparse Autoencoders. *arXiv preprint arXiv:2505.14536*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8342–8360.

Han, C.; Xu, J.; Li, M.; Fung, Y.; Sun, C.; Jiang, N.; Abdelzaher, T.; and Ji, H. 2024. Word Embeddings Are Steers for Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 16410–16430.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Huimin, L.; Isonuma, M.; Mori, J.; and Sakata, I. 2025. Unidetox: Universal detoxification of large language models via dataset distillation. In *The Thirteenth International Conference on Learning Representations*.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Ji, J.; Chen, B.; Lou, H.; Hong, D.; Zhang, B.; Pan, X.; Qiu, T. A.; Dai, J.; and Yang, Y. 2024. Aligner: Efficient alignment by learning to correct. *Advances in Neural Information Processing Systems*, 37: 90853–90890.

Kim, Y.; Kojima, T.; Iwasawa, Y.; and Matsuo, Y. 2024. Decoupling noise and toxic parameters for language model detoxification by task vector merging. In *First Conference on Language Modeling*.

Ko, C.-Y.; Chen, P.-Y.; Das, P.; Mroueh, Y.; Dan, S.; Kollias, G.; Chaudhury, S.; Pedapati, T.; and Daniel, L. 2024. Large Language Models can be Strong Self-Detoxifiers. *arXiv preprint arXiv:2410.03818*.

Krause, B.; Gotmare, A. D.; McCann, B.; Keskar, N. S.; Joty, S.; Socher, R.; and Rajani, N. F. 2021. GeDi: Generative Discriminator Guided Sequence Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 4929–4952.

Kumar, S.; Balachandran, V.; Njoo, L.; Anastasopoulos, A.; and Tsvetkov, Y. 2023. Language Generation Models Can Cause Harm: So What Can We Do About It? An Actionable Survey. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3299–3321.

Kwak, J. M.; Kim, M.; and Hwang, S. J. 2023. Language Detoxification with Attribute-Discriminative Latent Space. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, 10149–10171. Association for Computational Linguistics (ACL).

Leong, C. T.; Cheng, Y.; Wang, J.; Wang, J.; and Li, W. 2023. Self-Detoxifying Language Models via Toxification Reversal. In *Empirical Methods in Natural Language Processing*, 4433–4449.

Li, J.; Li, Z.; Ge, T.; King, I.; and Lyu, M. R. 2022. Text revision by on-the-fly representation optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10956–10964.

Li, Y.; Jiang, H.; Gong, C.; and Wei, Z. 2024. DESTEIN: Navigating Detoxification of Language Models via Universal Steering Pairs and Head-wise Activation Fusion. *arXiv preprint arXiv:2404.10464*.

Liang, X.; Wang, H.; Song, S.; Hu, M.; Wang, X.; Li, Z.; Xiong, F.; and Tang, B. 2024. Controlled Text Generation for Large Language Model with Dynamic Attribute Graphs. In *Findings of the Association for Computational Linguistics ACL 2024*, 5797–5814.

Liu, A.; Sap, M.; Lu, X.; Swayamdipta, S.; Bhagavatula, C.; Smith, N. A.; and Choi, Y. 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)*, 6691–6706.

Lu, X.; Welleck, S.; Hessel, J.; Jiang, L.; Qin, L.; West, P.; Ammanabrolu, P.; and Choi, Y. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35: 27591–27609.

Lu, Y.; Li, J.; Zhou, Y.; Zhang, Y.; Wang, W.; Li, X.; Zhang, M.; Liu, F.; Yu, J.; and Zhang, M. 2025. Adaptive detoxification: Safeguarding general capabilities of llms through toxicity-aware knowledge editing. *arXiv preprint arXiv:2505.22298*.

Niu, T.; Xiong, C.; Zhou, Y.; and Yavuz, S. 2024. Parameter-Efficient Detoxification with Contrastive Decoding. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, 30–40.

Pozzobon, L.; Ermis, B.; Lewis, P.; and Hooker, S. 2023. Goodtriever: Adaptive Toxicity Mitigation with Retrieval-augmented Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 5108–5125.

Ren, Y.; Yu, Y.; Liao, L.; Shang, Y.; Lu, K.; and Yan, M. 2025. R2DQG: A Quality Meets Diversity Framework for Question Generation over Knowledge Bases. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025)*, 8231–8240.

Siegelmann, R.; Mehrabi, N.; Goyal, P.; Goyal, P.; Bauer, L.; Dhamala, J.; Galstyan, A.; et al. 2024. MICo: Preventative detoxification of large language models through inhibition control. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 1696–1703.

Suau, X.; Delobelle, P.; Metcalf, K.; Joulin, A.; Apostoloff, N.; Zappella, L.; and Rodríguez, P. 2024. Whispering experts: neural interventions for toxicity mitigation in language models. In *Proceedings of the 41st International Conference on Machine Learning*, 46843–46867.

Tang, Z.; Zhou, K.; Li, J.; Ding, Y.; Wang, P.; Bowen, Y.; Hua, R.; and Zhang, M. 2024. CMD: a framework for Context-aware Model self-Detoxification. In *Empirical Methods in Natural Language Processing*, 1930–1949.

Tian, Y.; Deng, M.; Jin, G.; and Song, Y. 2025. Detoxification of Large Language Models through Output-layer Fusion with a Calibration Model. *arXiv preprint arXiv:2506.01266*.

Tu, L.; Yavuz, S.; Qu, J.; Xu, J.; Meng, R.; Xiong, C.; and Zhou, Y. 2024. Unlocking Anticipatory Text Generation: A Constrained Approach for Large Language Models Decoding. In *Empirical Methods in Natural Language Processing*, 15532–15548.

Uppaal, R.; Dey, A.; He, Y.; Zhong, Y.; and Hu, J. 2025. Model Editing as a Robust and Denoised variant of DPO: A Case Study on Toxicity. In *The Thirteenth International Conference on Learning Representations*.

Wang, B.; Ping, W.; Xiao, C.; Xu, P.; Patwary, M.; Shoeybi, M.; Li, B.; Anandkumar, A.; and Catanzaro, B. 2022. Exploring the limits of domain-adaptive training for detoxifying large-scale language models. *Advances in Neural Information Processing Systems*, 35: 35811–35824.

Wang, M.; Zhang, N.; Xu, Z.; Xi, Z.; Deng, S.; Yao, Y.; et al. 2024a. Detoxifying Large Language Models via Knowledge Editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3093–3118.

Wang, Y.; Tian, C.; Hu, B.; Yu, Y.; Liu, Z.; Zhang, Z.; Zhou, J.; Pang, L.; and Wang, X. 2024b. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM Web Conference 2024*, 3876–3887.

Xu, C.; He, Z.; He, Z.; and McAuley, J. 2022. Leashing the inner demons: Self-detoxification for language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11530–11537.

Yang, J.; Chen, D.; Sun, Y.; Li, R.; Feng, Z.; and Peng, W. 2024. Enhancing Semantic Consistency of Large Language Models through Model Editing: An Interpretability-Oriented Approach. In *Findings of the Association for Computational Linguistics ACL 2024*, 3343–3353.

Yi, X.; Wang, L.; Wang, X.; and He, L. 2025. Fine-grained detoxification framework via instance-level prefixes for large language models. *Neurocomputing*, 611: 128684.

Zhang, H.; and Song, D. 2022. DisCup: Discriminator Cooperative Unlikelihood Prompt-tuning for Controllable Text Generation. In *Empirical Methods in Natural Language Processing*, 3392–3406.

Zhang, H.; Wang, X.; Li, C.; Ao, X.; and He, Q. 2025. Controlling large language models through concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25851–25859.

Zhang, J.; Wu, Q.; Xu, Y.; Cao, C.; Du, Z.; and Psounis, K. 2024. Efficient toxic content detection by bootstrapping and distilling large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 21779–21787.

Zhang, X.; and Wan, X. 2023. Mil-decoding: Detoxifying language models at token-level via multiple instance learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 190–202.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).