ToxiGAN: Toxic Data Augmentation via LLM-Guided Directional Adversarial Generation

Peiran Li[1,2], **Jan Fillies**[1,2,3,4], **Adrian Paschke**[1,2,3]
[1]Freie Universität Berlin, Berlin, Germany
[2]Fraunhofer-Institut für Offene Kommunikationssysteme, Berlin, Germany
[3]Institut für Angewandte Informatik, Leipzig, Germany
[4]Stanford University, Stanford, USA
peiran.li@fu-berlin.de

Abstract

Augmenting toxic language data in a controllable and class-specific manner is crucial for improving robustness in toxicity classification, yet remains challenging due to limited supervision and distributional skew. We propose ToxiGAN, a class-aware text augmentation framework that combines adversarial generation with semantic guidance from large language models (LLMs). To address common issues in GAN-based augmentation such as mode collapse and semantic drift, ToxiGAN introduces a two-step directional training strategy and leverages LLM-generated neutral texts as semantic ballast. Unlike prior work that treats LLMs as static generators, our approach dynamically selects neutral exemplars to provide balanced guidance. Toxic samples are explicitly optimized to diverge from these exemplars, reinforcing class-specific contrastive signals. Experiments on four hate speech benchmarks show that ToxiGAN achieves the strongest average performance in both macro-F1 and hate-F1, consistently outperforming traditional and LLM-based augmentation methods. Ablation and sensitivity analyses further confirm the benefits of semantic ballast and directional training in enhancing classifier robustness.

1 Introduction

From comment sections on social media to online gaming chats, toxic language remains alarmingly pervasive, often escaping automated moderation systems. The propagation of such content poses a critical challenge for content moderation, societal safety, and responsible AI development (Wilson and Land, 2020). Automatic detection systems have shown promise in addressing this issue, yet their performance is often hindered by distributional imbalance in training data, most notably, an overrepresentation of neutral or non-toxic samples (Isaksen and Gambäck, 2020; Zampieri et al., 2019; Davidson et al., 2017). This imbalance can lead to majority-class overfitting and poor generalization, especially in low-resource or emerging domains. As a remedy, data augmentation using synthetically generated toxic examples has gained traction for balancing datasets and improving classifier robustness (Rizos et al., 2019).

Yet, turning this solution into practice is far from trivial. Generating toxic text for augmentation is a sensitive and technically challenging task (Vidgen et al., 2019). Synthetic examples must be toxic enough to reflect their target label, while maintaining semantic coherence and linguistic realism to ensure training utility (Rizos et al., 2019). Uncontrolled generation, particularly from language models or GANs, often leads to samples that are keyword-toxic but semantically incoherent or stylistically inconsistent (Gehman et al., 2020). Moreover, traditional GAN-based approaches suffer from mode collapse and semantic drift (Yu et al., 2017; Caccia et al., 2018), which further compromise sub-mode coverage and authenticity of the generated data, ultimately weakening decision-boundary calibration across toxic subtypes.

Given their remarkable fluency and contextual capabilities (Brown et al., 2020), large language models (LLMs) may appear well-suited for toxic text augmentation. However, their application is usually constrained by safety alignment objectives (Ouyang et al., 2022). LLMs are designed to resist producing toxic outputs, and when prompted to do so, tend to yield overly sanitized or generic responses (Achiam et al., 2023; Touvron et al., 2023). Consequently, they are limited in their ability to serve as direct toxic text generators, especially for class-specific data augmentation.

To address the limitations of existing generation methods, we introduce **ToxiGAN**[1], a controllable toxic text augmentation framework. Rather than using LLMs to generate toxic content directly,

[1]https://github.com/Peiran-Li-DS/ToxiGAN

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4029–4044

ToxiGAN leverages LLM-generated neutral exemplars as *semantic ballast* (reference anchors in embedding space). The generator learns to increase toxicity by deviating from these neutral anchors, while a class-aware discriminator enforces alignment with the target label. This design steers generation toward toxic content that preserves sub-mode coverage and remains label-consistent, supporting decision-boundary calibration across toxic subtypes. To mitigate semantic drift and mode collapse, where outputs either gravitate toward neutral semantics or collapse to a narrow toxic niche, we apply a two-step alternating optimization strategy that separately updates semantic deviation and class discrimination. Our main contributions are as follows:

- We propose **ToxiGAN**, a controllable augmentation framework that uses LLM-generated neutral text as *semantic ballast*, to guide stable and diverse generation.
- We design a two-step alternating directional learning algorithm that separates semantic deviation from class alignment, improving training stability and control.
- We evaluate ToxiGAN on four hate classification benchmarks and show that it achieves the best average macro-F1 and hate-F1 among GAN- and LLM-based augmentation methods.

These results highlight the value of class-aware adversarial generation guided by neutral semantic anchors, enabling effective and scalable toxic data augmentation for real-world classification tasks.

2 Related Work

Conventional Toxic Text Generation. Generating toxic or hateful text in a controlled manner has been explored through both supervised and adversarial paradigms. Early work relies on supervised models with prompting or conditional decoding (Gehman et al., 2020; Sheng et al., 2019), but these often lack diversity and controllability. Adversarial frameworks, particularly GAN-based approaches, have emerged as alternatives for generating class-conditioned toxic text. SentiGAN (Wang and Wan, 2018) introduces a sentiment-controlled generator-discriminator architecture, but struggles to produce coherent long-form samples for abstract or domain-specific categories, especially when distributional gaps across targets are large in the same sentiment. CatGAN (Liu et al., 2020) extends the GAN framework to multi-category text generation by introducing a category-aware discriminator and hierarchical training strategy. This allows the model to better handle diverse category labels compared to SentiGAN. However, it remains limited in its reliance on discriminator-only feedback, without leveraging external semantic guidance. Moreover, it is time-consuming and requires complicated tuning in its evolutionary training (Li et al., 2023). HateGAN (Cao and Lee, 2020) further adapts adversarial training to hateful text synthesis, focusing on stylistic features and linguistic variation. But it is constrained to binary classification setups (toxic vs. non-toxic) and lacks scalability to multi-class toxicity generation tasks. Our framework builds upon these insights by integrating LLMs as guidance modules and using a two-step alternating directional learning approach to maintain class consistency and generation quality.

LLMs in Text Augmentation. Recent advances in LLMs have enabled them to serve as effective tools for data augmentation (Ye et al., 2022; Yoo et al., 2021), especially in low-resource or few-shot settings. While many prior works use LLMs as standalone generators or annotators (Min et al., 2022; Sanh et al., 2021), few explicitly integrate them into structured adversarial training pipelines. More importantly, the major deployment-ready LLMs incorporate strict content moderation and safety alignment (Ouyang et al., 2022; Ganguli et al., 2022), which significantly limits their ability to generate or simulate toxic language, even when intended for research or augmentation. Although recent efforts such as ToxiCraft (Hui et al., 2024b) and ToxiLab (Hui et al., 2024a) attempt to generate toxic language directly from LLMs using prompt engineering or auxiliary control modules, these methods are often unstable and limited by content filtering policies and lack robustness in generating diverse, controllable toxic samples. In contrast, our work leverages LLMs not only to generate high-quality neutral examples but also to guide semantic direction and assist in discriminator training. This LLM-as-ballast design improves stability and semantic control during adversarial optimization.

Mode Collapse and Semantic Drift in Text Generation. GAN-based text generation often suffers from mode collapse and semantic drift (Che et al., 2017; Goodfellow et al., 2014; Spataru, 2024), which erode sub-mode coverage and class fidelity, especially in tasks involving toxic language.

Prior work addresses these issues through diversity-promoting objectives (Zhu et al., 2018) or classifier-based rewards (Yu et al., 2017), yet such methods lack semantic grounding. We introduce a semantic anchor via LLM-generated neutral exemplars and mitigate both collapse (concentration into a narrow toxic niche) and drift (gravitating toward neutral semantics) through alternating directional optimization that decouples semantic deviation from class discrimination (Zhang and Bansal, 2019; Dathathri et al., 2019; Spataru, 2024).

3 Methodology

Building on the high-level overview in Section 1, we present the full formulation of **ToxiGAN**, including its architectural components and training dynamics.

3.1 Problem Formulation

Let $\mathcal{D}_{real} = \{(x_i, y_i)\}$ denote the training dataset, where x_i is a text input and $y_i \in \{\text{neutral}, \text{toxic}_1, \ldots, \text{toxic}_K\}$. Our goal is to train a generator module G that, given a class label y_k and random noise $z \sim P_z$, generate a toxic sample that is (1) semantically authentic, and (2) representative of toxic class y_k.

3.2 Overall Framework

Figure 1 illustrates the overall architecture. ToxiGAN consists of the following components:

- **Toxic Generator Module** (G): Consists of multiple LSTM-based toxic generators and learns to generate samples for each toxic class from a noise distribution. Each class has a dedicated decoding branch.
- **Multi-class Discriminator** (D): Classifies input text into $K+2$ classes: K toxic classes, one neutral class, and one fake class to capture unrealistic generations.
- **LLM-based Neutral Text Provider**: A pre-trained LLM (e.g., Llama 3.2) is used to generate neutral in-domain examples for training D and guiding G via few-shot learning from the real neutral texts.

Real and generated samples are passed through D during training. A backward penalty is applied on the "fake" and "neutral" evaluations to encourage clearer decision boundaries and better generation quality.

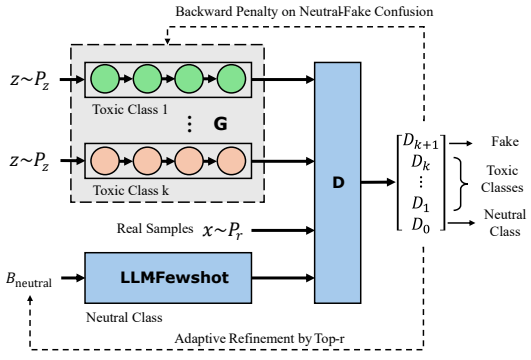

Figure 1: ToxiGAN with k toxic generators, one neutral texts provider, and one multi-class discriminator.

3.3 LLM as Ballast: Preventing Mode Collapse and Semantic Drift

To address mode collapse and semantic drift, we introduce a **LLM-based neutral text provider** that offers high-quality, fluent in-domain exemplars. These samples serve as *semantic anchors* during both generation and discrimination, improving stability and realism under domain shifts while **preserving sub-mode coverage** in representation space.

It contributes in three complementary ways: (1) **Neutral Text Generation**: using a small set of real examples as prompts, the LLM generates fluent, contextually appropriate neutral samples. These exemplars act as semantic ballast for both the generator and discriminator. (2) **Discriminator Enhancement**: LLM-generated neutral samples are included during discriminator training to sharpen its separation of target classes and authenticity, which supports more reliable decision-boundary calibration in low-resource or noisy regimes. (3) **Semantic Filtering**: when the neutral data are noisy or partially mislabeled, the LLM provides a soft constraint that downweights samples inconsistent with natural-language regularities, mitigating drift and maintaining coverage across toxic sub-modes. Taken together, these roles help prevent collapse and drift, preserve sub-mode coverage and authenticity, and yield label-faithful toxic text that better supports downstream boundary calibration.

Adaptive Refinement of Neutral Pool. To ensure semantic divergence is measured against high-quality anchors, we employ a dynamic filtering strategy guided by discriminator evaluation of neutral class D_0. Starting from a large pool $\mathcal{X}_{\text{neutral}}$ of real neutral texts, we compute per-sample neutrality scores: $s(x) = D_0(x)$, where $D_0(x)$ reflects how similar the perceived x is to the real neutral data

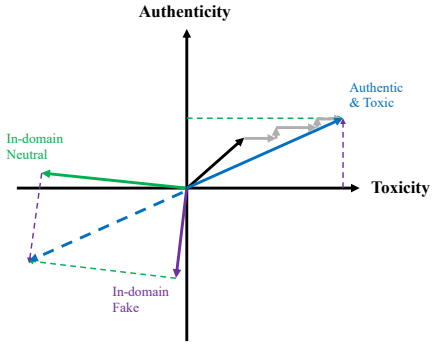

Figure 2: Illustration of Two-Step Alternating Directional Learning in Embedding Space. The black arrow shows the initial generation after pretraining. Gray arrows represent updates during alternating optimization: shifting toward toxicity and authenticity directions by penalizing unexpected directional evaluations.

evaluated by the discriminator. After each adversarial epoch, we retain only the top-$r\%$ of neutral candidates (by $s(x)$), halving r until a fixed-size ballast pool (e.g., 100 samples) is reached:

$$\mathcal{B}^{(t)}_{\text{neutral}} = \text{Top}_r\left(\mathcal{X}_{\text{neutral}}, s(x)\right) \tag{1}$$

All LLM few-shot prompts are drawn from this final refined pool.

3.4 Two-Step Alternating Directional Learning

While prior work e.g. SentiGAN has explored using penalty-based function, it simply relies on evaluating how well the synthetic text aligns with in-domain authenticity by the discriminator. In the context of toxic text generation, it is not sufficient to merely generate text classified as "authentic" by a discriminator; the generated output should also semantically diverge from neutral language in a meaningful and controlled direction.

One of our core contributions is a semantic directional constraint that guides generation in embedding space, as shown in Figure 2. Instead of rewarding toxic and authentic outputs jointly, we propose a Two-Step Alternating Directional Learning strategy that disentangles and alternates two core training objectives: semantic toxicity and linguistic authenticity. Crucially, these objectives rely on distinct evaluation signals, cosine distance from neutral exemplars and class probabilities from the discriminator, making joint optimization nontrivial. Alternating updates allow each direction to be optimized according to its own metric, preserving interpretability while promoting both control and domain-authenticity.

Alternating vs. joint objective. A natural alternative to our alternating toxicity and authenticity steps is to optimize a single joint objective for the generator:

$$\mathcal{L}^{\text{joint}}_{G_i} = \lambda\, \mathcal{L}^{\text{tox}}_{G_i} + (1-\lambda)\, \mathcal{L}^{\text{auth}}_{G_i}, \tag{2}$$

where $\mathcal{L}^{\text{tox}}_{G_i}$ corresponds to the toxicity-driven divergence objective (Eq. 3), $\mathcal{L}^{\text{auth}}_{G_i}$ corresponds to the authenticity objective (Eq. 4), and $\lambda \in [0,1]$ controls their relative importance. However, as illustrated by the loss dynamics in Figure 7 (Appendix A.2), the scale and evolution of $\mathcal{L}^{\text{tox}}_{G_i}$ and $\mathcal{L}^{\text{auth}}_{G_i}$ are not well aligned during training: at different stages, one term can dominate the other both in magnitude and in gradient variability. In such a setting, a fixed global weighting λ in Eq. 2 is difficult to tune and can easily bias the generator toward either overly aggressive toxicity (ignoring authenticity) or overly conservative changes (staying too close to neutral exemplars). We therefore adopt an alternating optimization scheme, where we update G_i with $\mathcal{L}^{\text{tox}}_{G_i}$ in the toxicity step and with $\mathcal{L}^{\text{auth}}_{G_i}$ in the authenticity step, providing a simple and robust way to balance the two objectives over the course of training without committing to a single hand-tuned trade-off parameter.

Let $G_i(z)$ be a generated sample for toxic class i by random noise z and x_{neutral} be a neutral sentence sampled via LLM few-shot prompting. At each training step t, the generator is updated based on one of two alternating objectives:

- **Toxicity Step (odd t):** The generator is guided to semantically diverge from neutral content by minimizing the cosine similarity between the generated sentence and a neutral reference:

$$\mathcal{L}^{(t)}_{G_i} = \mathbb{E}[\max_{x \in \mathcal{B}_{\text{neutral}}} \cos\left(\Phi(G_i(z)), \Phi(x)\right)], \quad \text{if } t \bmod 2 = 1 \tag{3}$$

  where $\Phi(\cdot)$ denotes the sentence embedding function (e.g., all-MiniLM-L6-v2). $\mathcal{B}_{\text{neutral}}$ is a set of fluent, LLM-generated neutral sentences serving as *Semantic Ballast*.

- **Authenticity Step (even t):** The generator is optimized to improve naturalness and realism by maximizing the discriminator's belief that the output is real:

$$\mathcal{L}^{(t)}_{G_i} = \mathbb{E}[1 - D_i(G_i(z))], \quad \text{if } t \bmod 2 = 0 \tag{4}$$

  where $D_i(\cdot)$ is the discriminator output of toxic class i.

This alternation allows the generator to progressively move away from LLM-neutral semantics while remaining within the bounds of in-domain authenticity.

3.5 Adversarial Training

ToxiGAN employs a class-conditional adversarial training framework comprising K class-specific generators $\{G_i\}_{i=1}^{K}$ and a unified multi-head discriminator D. The training process alternates between two optimization goals: promoting semantic divergence from neutral anchors (toxicity direction) and aligning with real data distribution (authenticity), as illustrated in Algorithm 1 (Appendix A.1).

Each generator G_i is initialized via maximum likelihood estimation (MLE) pretraining on real toxic samples from class i. In parallel, the discriminator D is pre-trained using a mixture of real, generated, and LLM-synthesized neutral texts. Neutral exemplars are dynamically selected from a refined ballast pool $\mathcal{B}^{(t)}_{\text{neutral}}$ (as previously described), ensuring adaptive semantic contrast throughout training. This design maintains meaningful toxicity direction signals and improves class fidelity over static or randomly sampled baselines.

During adversarial training, at each epoch t, we iterate over classes $i \in \{1, \ldots, K\}$ and generate toxic texts $\mathcal{F}_i$ from G_i. The generator objective alternates across epochs:

$$J_{G_i}(\theta_{g_i}) = \mathcal{L}^{(t)}\big(G_i(z; \theta_{g_i})\big) \tag{5}$$

where $\mathcal{L}^{(t)}$ corresponds to a toxicity-inducing loss at odd steps (see Eq. 3) and an authenticity-promoting loss at even steps (see Eq. 4). This two-step directional optimization prevents semantic drift and encourages category fidelity across generation stages.

The discriminator D receives three types of inputs: (1) real labeled data from each class i, (2) synthetic toxic texts from $\{G_i\}$, and (3) neutral texts generated by the $\mathcal{LLM}$. It consists of $K + 2$ output heads: one per toxic class (D_i), one for fake samples (D_{k+1}), and one for LLM-neutral detection. Its training objective is:

$$\begin{aligned} J_D(\theta_d) = & -\mathbb{E}_{z\sim P_z}\big[\log D_{k+1}(G_i(z); \theta_d)\big] \\ & -\mathbb{E}_{\tilde{x}\sim \mathcal{LLM}(\mathcal{B}^{(t)}_{\text{neutral}})}\big[\log D_{k+1}(\tilde{x}; \theta_d)\big] \\ & -\sum_{i=1}^{K} \mathbb{E}_{x\sim P_{r_i}}\big[\log D_i(x; \theta_d)\big] \end{aligned} \tag{6}$$

After each epoch, the ballast pool $\mathcal{B}^{(t)}_{\text{neutral}}$ is dynamically refined by filtering candidate prompts based on discriminator confidence scores (Eq (1)). This mechanism ensures that the LLM continues to provide diverse and semantically representative neutral anchors, supporting both stable optimization and class-specific control.

4 Experimentation

4.1 Experiment Setup

Datasets. We evaluate our approach on four publicly available hate speech datasets with diverse origins and annotation schemes:

WZ (Waseem and Hovy, 2016) contains tweets annotated by experts into *racism*, *sexism*, or *neither*, and is widely used for binary and multiclass toxic language detection.

DC (Discord Chat) (Fillies et al., 2023) is collected from gaming chat communities and annotated along linguistic dimensions such as *stereotype*, *violence*, *normalized discrimination*, *slander*, and *irony*, providing a fine-grained perspective on in-domain toxicity styles.

HX (HateXplain) (Mathew et al., 2021) combines social media posts from Twitter and Gab, annotated through crowd-sourced rationales, and categorized into *offensive*, *general hate*, and targeted categories such as *gender/sex*, *race*, and *religion*.

OR (Offensive Reddit) (Qian et al., 2019) introduces a structured Reddit dataset with conversational dynamics. Each instance is labeled as *non-hate*, *initiating-hate*, or *responding-hate*, reflecting intervention scenarios in real-world moderation.

Table 1 summarizes the key attributes of these datasets after processing. Together, they span multiple source domains (e.g., Twitter), annotation paradigms, and toxicity taxonomies, allowing us to comprehensively evaluate both the controllability and generalizability of ToxiGAN.

Baselines. We compare ToxiGAN against a broad set of data augmentation methods commonly used in text generation and toxic content detection.

Dataset	Source	Guideline	Category
WZ	Twitter	Hate Targets	**3**-class: racism, sexism, neither
DC	Discord	Linguistic Forms	**7**-class: no-hate, stereotype, dehumanization, violence, discrimination, irony, slander
HX	Twitter, Gab	Hate Targets	**5**-class: normal / offensive, general-hate, gender / sex, race, religion
OR	Reddit	Initiate or not	**3**-class: non-hate, initiating, responding

Table 1: Overview of the datasets. Categories cover a diverse range of toxicity taxonomies.

These include:

- **No Augmentation (Base)**: Training directly on the limited toxic dataset without synthetic data.
- **Gold Labels (Ideal)**: A hypothetical upper-bound setting where all original toxic samples are preserved across different ratios.
- **Conventional Methods: Oversampling**: Duplicating real toxic samples to match the target augmentation ratio. **EDA** (Wei and Zou, 2019): A light heuristic method applying synonym replacement and word swapping. **Back-Translation**: Translating sentences to another language (e.g. German in this case) and back to create paraphrases, via WMT-19 translator[2]. **T5-Paraphrase** (Piedboeuf and Langlais, 2023; Scherrer, 2020): Using a fine-tuned T5 model[3] for paraphrasing toxic data. **SentiGAN** (Wang and Wan, 2018): A GAN-based method that controls sentiment via multiple generators and a discriminator.
- **LLM-Based Generation**: Generating toxic samples using **Mistral-v0.3**[4] via **ZeroGen** (Ye et al., 2022) (ZG; zero-shot synthesis from class definitions and constraints without seed examples), and **SunGen** (Gao et al., 2022) (SG; ZG followed by self-guided reweighting to downweight noisy synthetic samples), then **Fewshot** (FS) generation with randomly selected 5 examples in each corresponding toxic class; **LLaMA3.2**[5], **GPT-4.1**[6], and **GPT-4o**[7] via carefully crafted prompts (*ToxiCraft* (Hui et al., 2024b), denoted as TC), due to their tight moderation.

We split each dataset into 80% training, 10% validation, and 10% testing. To simulate low-resource settings, only 50% of the training set is used as labeled data; the remaining 50% is replaced with augmented samples from each method. All results are averaged over 5 runs.

Evaluation Metrics. We evaluate model performance using the following metrics:

- **Toxicity Score**: The average toxicity scores of a group, computed by external toxicity evaluator[8].
- **Macro-F1**: The unweighted average F1-scores across all classes, capturing overall balance.

[2] https://huggingface.co/facebook/wmt19-de-en
[3] https://huggingface.co/hetpandya/t5-small-tapaco
[4] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[5] https://huggingface.co/meta-llama/Llama-3.2-1B
[6] https://platform.openai.com/docs/models/gpt-4.1-nano
[7] https://platform.openai.com/docs/models/gpt-4o
[8] https://github.com/unitaryai/detoxify

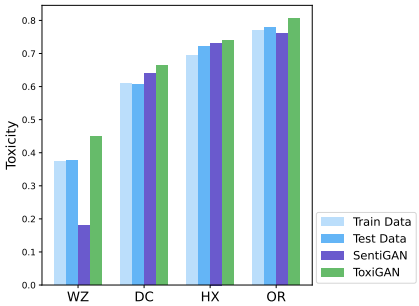

Figure 3: Average toxicity scores of real and synthetic samples across datasets.

- **Hate-F1**: The F1-score computed specifically for the toxic or hate class, highlighting the model's ability to detect rare but critical instances.

4.2 Toxicity of Synthetic Texts

To assess whether the generated texts exhibit sufficient toxicity, we computed toxicity across four datasets: WZ, DC, HX, and OR. Figure 3 compares the toxicity levels of training data, test data, and synthetic texts produced by SentiGAN and ToxiGAN.

ToxiGAN consistently produces samples with toxicity levels comparable to or higher than those in the original training and testing sets. In contrast, SentiGAN-generated texts often display reduced toxicity, especially in datasets with low toxicity originally, e.g WZ, where SentiGAN has difficulty to "interpret" how to generate toxic texts. This suggests that ToxiGAN is more effective at preserving the intended toxic signal, contributed by its directional training and LLM-guided neutral anchoring. These results confirm that ToxiGAN not only preserves authenticity but also enhances class-level toxicity control in the generated samples.

4.3 Result of Augmentation Performance

Table 2 reports performance comparisons across four datasets and two backbone classifiers (BERT and RoBERTa), under various data augmentation strategies. We evaluate both Macro-F1 and Hate-F1 to capture class-level balance and minority class performance.

On average, ToxiGAN outperforms all baselines across both Macro-F1 and Hate-F1, achieving the best mean results across datasets and classifiers. Notably, on DC and OR datasets with implicit categories (by linguistic forms and initiate or not), ToxiGAN outperforms all baselines in both metrics. This demonstrates its advantage in mod-

Classifier	Augmentation	WZ		DC		HX		OR		*Avg.*	
		H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1
BERT	Base (No Aug.)	68.1	74.8	26.7	34.8	34.6	41.6	46.5	61.6	44.0	53.2
	Ideal (Gold Labels)	69.7	76.0	28.5	35.8	37.7	43.6	49.7	63.8	46.4	54.8
	Oversampling	68.3	75.0	27.8	35.3	35.1	42.3	47.2	62.0	44.6	53.6
	EDA	71.9	77.2	26.3	35.9	36.0	42.4	47.2	62.0	45.4	54.4
	Back-Translate	73.4	78.4	27.0	36.4	34.7	41.8	47.5	62.2	45.6	54.7
	T5-Paraphase	70.4	76.3	27.1	36.8	36.0	40.7	47.6	62.2	45.3	54.0
	Mistral-v0.3-ZG	71.3	76.9	28.2	37.4	33.6	41.3	44.6	60.2	44.4	54.0
	Mistral-v0.3-SG	71.9	77.2	28.5	37.6	33.3	41.6	45.5	61.0	44.8	54.3
	Mistral-v0.3-FS	71.5	76.8	28.0	37.6	36.1	43.1	46.6	61.5	45.5	54.8
	Llama3.2-TC	71.9	77.3	27.4	37.0	34.9	40.9	47.1	61.6	45.3	54.2
	GPT4.1-TC	71.8	77.2	26.9	35.2	26.0	37.0	46.2	60.9	42.7	52.6
	GPT4o-TC	72.6	77.6	27.5	35.7	29.3	38.4	46.2	61.1	43.9	53.2
	ToxiGAN (Ours)	72.2	77.7	29.7	37.7	36.9	42.8	47.8	62.7	46.7	55.2
RoBERTa	Base (No Aug.)	71.2	76.9	29.2	38.6	39.6	44.4	45.6	61.0	46.4	55.2
	Ideal (Gold Label)	72.5	77.9	30.7	39.7	42.6	49.1	49.4	63.7	48.8	57.6
	Oversampling	71.7	77.3	29.5	38.6	39.0	45.1	46.2	61.4	46.6	55.6
	EDA	71.3	77.0	29.3	39.0	40.6	47.6	46.8	62.0	47.0	56.4
	Back-Translate	72.8	77.9	29.4	39.1	40.9	47.0	45.9	61.1	47.2	56.3
	T5-Paraphase	73.1	78.3	29.4	38.9	40.0	46.9	46.9	61.8	47.3	56.5
	Mistral-v0.3-ZG	71.3	77.0	28.2	37.6	41.9	46.8	44.9	60.6	46.6	55.5
	Mistral-v0.3-SG	72.0	77.5	29.1	38.3	44.0	47.3	46.1	61.5	47.8	56.2
	Mistral-v0.3-FS	71.7	77.2	28.2	37.9	44.2	48.8	47.5	62.2	47.9	56.5
	Llama3.2-TC	72.8	77.7	27.5	37.3	41.0	45.7	44.5	59.7	46.4	55.1
	GPT4.1-TC	72.4	77.6	27.3	36.3	37.4	42.6	45.5	60.6	45.6	54.3
	GPT4o-TC	73.4	78.3	28.7	38.1	39.1	44.4	46.6	61.5	47.0	55.6
	ToxiGAN (Ours)	72.8	77.9	31.0	40.1	41.6	48.1	48.3	62.9	48.4	57.3

Table 2: Augmentation results compared to other baselines. The best and second best are highlighted in green and yellow. Note: LLM-based augmentation methods may be constrained by internal safety alignment, affecting their ability to generate truly toxic samples despite prompt customization.

eling nuanced toxicity directions. Among traditional augmentation techniques, back-translation and T5-based paraphrasing yield competitive results, while LLM-based generation (e.g., Mistral, GPT-4) shows less consistent improvements.

Compared to the Ideal (Gold Label) setting, ToxiGAN closes the performance gap and even surpasses it in many cases. This suggests that semantically guided synthetic samples can be as effective as real annotated data in low-resource scenarios. The improvements are particularly pronounced for Hate-F1, indicating stronger capability in capturing toxic-specific signal.

We further observe that several LLM-based augmentation methods (e.g., **GPT4o**, **LLaMA3.2**) do not consistently outperform simpler techniques such as back-translation or T5-based paraphrasing. We hypothesize that this may be attributed to the internal moderation mechanisms or alignment procedures embedded in modern LLMs. Despite prompt engineering and the use of frameworks like **ToxiCraft** to elicit toxic samples, models such as GPT-4 and LLaMA-3 still exhibit reluctance or failure to generate explicitly toxic content. This results in samples that are grammatically fluent but often semantically neutral or diluted, thereby reducing their efficacy in contrastive training. Even models like **Mistral**, which are not tightly moderated, may still inherit instruction-tuning biases toward politeness or neutrality due to alignment with general-purpose pre-training corpora. These implicit constraints likely inhibit the generation of toxic-specific features, explaining their relatively weaker performance in Hate-F1.

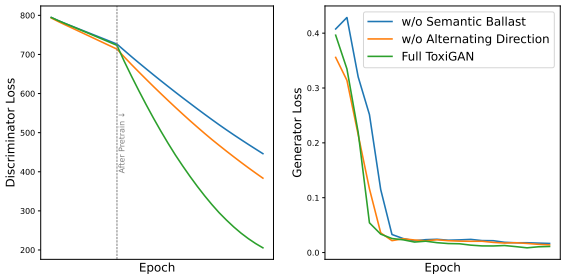

Figure 4: Training curves of ToxiGAN and ablations on OR. (ToxiGAN w/o semantic ballast degrades to SentiGAN.)

4.4 Training Stability and Convergence

We analyze the training dynamics of ToxiGAN and its ablated variants to understand the impact of semantic guidance and alternating optimization on

Classifier	Ablation	Sem. Bal.	Alt.-Dir.	WZ		DC		HX		OR		*Avg.*	
				H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1	H.-F1	M.-F1
BERT	w/o LLM (⇔ *SentiGAN*)	x	x	69.5	75.7	28.9	36.4	31.8	39.2	47.0	61.9	44.3	53.3
	w/o Toxicity Step	✓	x	70.9	76.6	29.6	37.1	35.0	41.8	47.4	62.2	45.7	54.4
	Full ToxiGAN	✓	✓	**72.2**	**77.7**	**29.7**	**37.7**	**36.9**	**42.8**	**47.8**	**62.7**	**46.7**	**55.2**
RoBERTa	w/o LLM (⇔ *SentiGAN*)	x	x	71.6	77.0	29.4	38.4	40.5	45.9	46.6	61.9	47.0	55.8
	w/o Toxicity Step	✓	x	72.1	77.4	30.3	39.1	41.4	46.7	47.3	62.2	47.8	56.4
	Full ToxiGAN	✓	✓	**72.8**	**77.9**	**31.0**	**40.1**	**41.6**	**48.1**	**48.3**	**62.9**	**48.4**	**57.3**

Table 3: Ablation study on four datasets. Best performance scores of each testing are in bold.

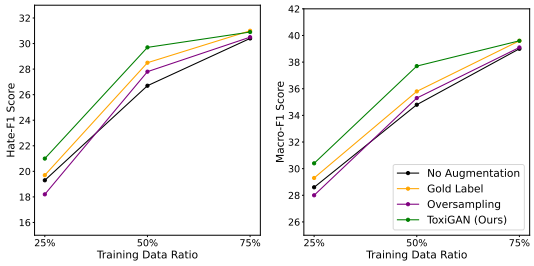

Figure 5: Sensitivity to Various Augmentation Data Ratio on DC. Ratios refer to proportion of original data; remainder is filled with synthetic/duplicated toxic data.

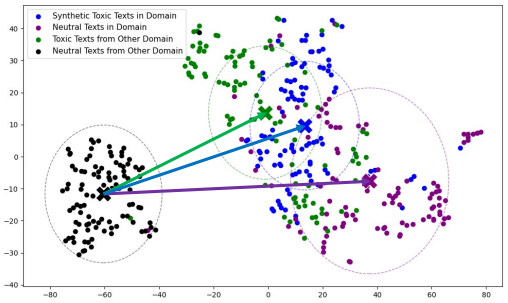

Figure 6: t-SNE visualization of real and synthetic texts. Arrows indicate semantic shifts: neutral to toxic (green), out-of-domain to in-domain (purple), and their composite (blue).

convergence behavior. Figure 4 illustrates the loss curve of the generator and discriminator on the OR dataset. The full ToxiGAN model demonstrates significantly smoother generator loss and faster convergence compared to the ablations. Meanwhile, the discriminator loss steadily decreases and maintains a lower variance, suggesting more stable and effective adversarial training. These observations indicate that the semantic ballast from LLM exemplars and the directional learning mechanism not only improve generation quality but also enhance optimization stability, both crucial for reliable toxic text augmentation.

4.5 Ablation Study

We ablate two core components of ToxiGAN: the LLM-based semantic ballast (Sem. Bal.) and the alternating directional learning strategy (Alt.-Dir.). Table 3 reports results on four datasets using BERT and RoBERTa classifiers. **Removing the semantic ballast** degrades the model to SentiGAN, resulting in substantial drops across all metrics, highlighting the role of LLM exemplars in stabilizing training and guiding generation. **Omitting the toxicity step** also weakens performance, particularly in Hate-F1, indicating the necessity of explicit semantic deviation. The full model consistently outperforms its ablated variants across classifiers and datasets, validating the effectiveness of both semantic guidance and directional optimization.

4.6 Sensitivity Analysis on Data Ratio

We evaluate how varying real-data availability affects ToxiGAN's effectiveness. Using 25%, 50%, and 75% of labeled data, we augment the remainder with (i) oversampled toxic samples, (ii) gold-labeled data (ideal upper bound), or (iii) ToxiGAN-generated samples. As shown in Figure 5, ToxiGAN consistently outperforms oversampling, particularly under low-resource settings (25%), and even rivals gold-label augmentation at higher ratios. This highlights the utility of our generation strategy in preserving class-specific signals and improving robustness under data scarcity.

4.7 Visualization in Semantic Space

To examine the semantic behavior of generated samples, we visualize sentence embeddings using t-SNE on both an in-domain dataset (DC) and an out-of-domain dataset (Jigsaw[9]), sourced from Wikipedia comments. As shown in Figure 6, ToxiGAN's synthetic toxic texts (blue) occupy a clear intermediate position between in-domain neutral texts (purple) and out-domain toxic texts (green). We observe directional trends from neutral to toxic, and from out-of-domain to in-domain, aligning with our intended semantic shift (as in Figure 2).

[9] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

This supports the effectiveness of our two-step directional learning and confirms that the generated samples are both toxic and domain-coherent.

5 Conclusion

In this work, we propose **ToxiGAN**, a directional adversarial framework for controllable toxic text augmentation. By incorporating a semantic ballast from LLM-generated neutral exemplars and a two-step alternating training strategy, ToxiGAN improves optimization stability and reduces semantic drift and mode-collapse tendencies. The resulting synthetic samples remain label-consistent and preserve sub-mode coverage in representation space, thereby supporting more reliable decision-boundary calibration for toxicity classifiers. Across four benchmark datasets, ToxiGAN outperforms traditional and LLM-based augmentation baselines, especially under low-resource settings. These results highlight the value of integrating structured semantic guidance with adversarial learning to achieve robust, scalable augmentation for toxicity detection.

6 Limitations

Our study has several limitations. (1) **Data coverage.** Experiments focus on a limited set of English, social-media-centric datasets; generalization to other domains, genres, and languages remains open. (2) **Model and metric dependence.** Findings may be sensitive to the chosen backbones and to toxicity scorers with known bias profiles; stronger or fairer evaluators could change conclusions. (3) **Guidance design.** The efficacy of our ballast/embedding choices and hyperparameters (e.g., pool filtering) has not been systematically audited, so variance across alternatives needs to be further explored. To maintain a focus on the availability of ToxiGAN and its improvement of downstream classification tasks, we only employ light model for semantic embedding and neutral exemplar provider. More powerful models are expected to investigate as replacement in ToxiGAN. (4) **Evaluation scope.** A more comprehensive treatment of human and fairness assessments (e.g., across dialects and protected attributes) is beyond our current scope. (5) **Practicality and risk.** The method adds training/generation overhead and, despite safeguards, synthetic toxic text introduces curation and misuse risks that require careful data handling. Some alternative strategies rely on jailbreaking large language models to directly emit toxic outputs, but this raises additional safety and compliance concerns and may be infeasible under typical API usage policies; we therefore keep the base LLM in a neutral role and delegate toxic generation to a separate GAN. Deployed systems should combine ToxiGAN with appropriate data governance, access control, and safety safeguards.

7 Ethical Statement

Toxic text generation poses ethical concerns due to the risk of misuse and potential harm. We describe below the steps taken to ensure responsible use of our proposed framework.

Purpose and Research Motivation. ToxiGAN is developed solely for **data augmentation in toxicity classification**, to address data imbalance and improve model robustness. Generated texts are used only for **controlled classifier training and evaluation**, not for human-facing applications.

Handling of Harmful Content. To avoid direct generation of toxic content by large language models (LLMs), we use LLM-generated neutral examples as semantic ballast. Toxic samples are generated adversarially using task-specific discriminators within a closed environment, without user input or external deployment.

Responsible Release and Usage Policy. We acknowledge that generative frameworks such as ToxiGAN could be misused if deployed without constraints. To minimize this risk:

- We will release the code and models only under a research license.
- The repository will include a clear usage policy discouraging misuse, aligned with community guidelines for safe and ethical NLP.
- All datasets used are publicly available and have been ethically sourced as per original licenses.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot

learners. *Advances in neural information processing systems*, 33:1877–1901.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2018. Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Rui Cao and Roy Ka-Wei Lee. 2020. Hategan: Adversarial generative-based data augmentation for hate speech detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6327–6338.

Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jan Fillies, Silvio Peikert, and Adrian Paschke. 2023. Hateful messages: A conversational data set of hate speech produced by adolescents on discord. In *International Data Science Conference*, pages 37–44. Springer.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Self-guided noise-free data generation for efficient zero-shot learning. *arXiv preprint arXiv:2205.12679*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, Lin Ai, Yinheng Li, Julia Hirschberg, and Congrui Huang. 2024a. Toxilab: How well do open-source llms generate synthetic toxicity data? *arXiv preprint arXiv:2411.15175*.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, and Congrui Huang. 2024b. Toxicraft: A novel framework for synthetic generation of harmful information. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16632–16647.

Vebjørn Isaksen and Björn Gambäck. 2020. Using transfer-based language models to detect hateful and offensive language online. In *Proceedings of the fourth workshop on online abuse and harms*, pages 16–27.

Xinze Li, Kezhi Mao, Fanfan Lin, and Zijian Feng. 2023. Feature-aware conditional gan for category text generation. *Neurocomputing*, 547:126352.

Zhiyue Liu, Jiahai Wang, and Zhiwei Liang. 2020. Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8425–8432.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Frédéric Piedboeuf and Philippe Langlais. 2023. Is chatgpt the ultimate data augmentation algorithm? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15606–15615.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764.

Georgios Rizos, Konstantin Hemker, and Björn Schuller. 2019. Augment to prevent: short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 991–1000.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Yves Scherrer. 2020. Tapaco: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6868–6873.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412.

Ava Spataru. 2024. Know when to stop: A study of semantic drift in text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3656–3671.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the third workshop on abusive language online*. Association for Computational Linguistics.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Richard Ashby Wilson and Molly K Land. 2020. Hate speech on social media: Content moderation in context. *Conn. L. Rev.*, 52:1029.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

A Appendix

A.1 Training Procedure of ToxiGAN

We first pretrain the class-conditional generators with MLE and a multi-head discriminator on a mixture of labeled toxic instances and LLM-synthesized neutral exemplars (Alg. 1). Then conduct adversarial training with alternating objectives: at odd steps the generator is pushed away from neutral anchors by minimizing cosine similarity (toxicity step, Eq. 3), and at even steps it is optimized for

Algorithm 1 Training of ToxiGAN

Input: Generators $\{G_i\}_{i=1}^{K}$, discriminator D, large language model $\mathcal{LLM}$, real dataset $\mathcal{D}_{real}$;
Output: Well trained generators $\{G_i^{'}\}_{i=1}^{K}$;

Initialize: $\{G_i\}_{i=1}^{K}$, D, ballast set $\mathcal{B}_{\text{neutral}}$;
Pre-train $\{G_i\}_{i=1}^{K}$ on $\mathcal{D}_{real}$ using MLE;
Generate: fake toxic texts $\{\mathcal{F}_i\}_{i=1}^{K}$ by $\{G_i\}_{i=1}^{K}$, fake neutral texts $\mathcal{F}_0$ by $\mathcal{LLM}$ with samples from $\mathcal{B}_{\text{neutral}}$;
Pre-train D on $\{\mathcal{D}_{real}, \mathcal{F}\}$, $\mathcal{F} = \{\mathcal{F}_i\}_{i=0}^{K}$;
for epoch = 1 to max_epoch **do**
  for each class $i = 1$ to K **do**
    Generate fake toxic texts $\mathcal{F}_i$ = $\{G_i(z)\}$;
    if epoch is odd **then**
      Compute $\mathcal{L}$ according Eq (3) # *Toxicity Step*;
    else
      Compute $\mathcal{L}$ according Eq (4) # *Authenticity Step*;
    end if
    Update G_i by minimizing Eq (5);
  end for
  Generate fake neutral texts $\mathcal{F}_0$ by $\mathcal{LLM}$ with samples from $\mathcal{B}_{\text{neutral}}$, merge into $\{\mathcal{D}_{real}, \mathcal{F}\}$;
  Update D by minimizing Eq (6);
  Update $\mathcal{B}_{\text{neutral}}$ according Eq (1);
end for;
return $\{G_i\}_{i=1}^{K}$;

realism under the discriminator (authenticity step, Eq. 4. While periodically refreshing the discriminator and the neutral ballast pool (Eq. 6, 1).

A.2 Rationale and Theoretical Support for Alternating Optimization

ToxiGAN optimizes two distinct objectives for each generator: semantic toxicity (via directional deviation from neutral exemplars) and linguistic authenticity (via discriminator feedback). Rather than combining these objectives into a single joint loss or reward, we employ an alternating optimization strategy: each training step updates the generator based on either toxicity or authenticity feedback, but never both simultaneously. Below, we justify this design from both a policy gradient perspective and empirical observations.

(1) Reward Signal Imbalance. ToxiGAN uses a policy gradient formulation inspired by SeqGAN and SentiGAN, where the generator is updated using REINFORCE:

$$\nabla_\theta \mathcal{L}_{\text{PG}} = \mathbb{E}_{x \sim G_\theta} \left[\nabla_\theta \log P_\theta(x) \cdot R(x) \right],$$

with $R(x)$ representing the reward signal, computed as either toxicity or authenticity depending on the step. A naive joint formulation like $R(x) = \alpha R_{\text{tox}}(x) + \beta R_{\text{auth}}(x)$, often leads to signal imbalance, where the more stable reward (typically authenticity) dominates the learning signal, suppressing meaningful semantic deviation. Alternating updates ensure that each reward signal receives full gradient feedback without competition, which is crucial in early training.

(2) Gradient Variance and Directional Conflict. Policy gradient methods are known for high variance. When two reward signals reflect objectives that act in different or even conflicting regions of semantic space, joint updates may suffer from noisy or oscillatory learning. Alternating updates reduce this variance by decoupling the reward sources, enabling the generator to stably explore toxic semantic directions without interference from stylistic constraints, and vice versa.

(3) Multi-objective Decomposition and Interpretability. In standard multi-objective optimization, joint training seeks to minimize a convex combination of objectives:

$$\min_\theta \mathbb{E}_{x \sim G_\theta} \left[\alpha R_{\text{tox}}(x) + \beta R_{\text{auth}}(x) \right].$$

However, this does not guarantee optimality with respect to either reward individually. Alternating optimization can be interpreted as a form of multi-objective decomposition or coordinate-wise reinforcement, which helps the generator approximate both objectives more effectively and improves the interpretability of training dynamics—particularly for controllable generation tasks.

(4) Theoretical Stability and Convergence Considerations. We now provide a simplified convergence analysis of our alternating policy gradient training scheme. At each step, the generator is updated via REINFORCE with one active reward function (either toxicity or authenticity):

$$\mathcal{L}_{\text{PG}}^{(t)} = -\mathbb{E}_{x \sim G_{\theta_t}} \left[R_t(x) \log P_{\theta_t}(x) \right],$$

where $R_t(x) \in \{R_{\text{tox}}, R_{\text{auth}}\}$ depends on the current step. The gradient estimator is:

$$\nabla_\theta \mathcal{L}_{\text{PG}}^{(t)} = -\mathbb{E}_{x \sim G_{\theta_t}} \left[R_t(x) \nabla_\theta \log P_{\theta_t}(x) \right].$$

Assumptions:

1. $R_t(x) \in [0, R_{\max}]$: reward is bounded.
2. $\log P_\theta(x)$ is L-Lipschitz in θ.
3. The policy has sufficient exploration, i.e., all actions have non-zero probability.

Applicability in ToxiGAN:
(In our implementation, these assumptions are satisfied.)

1. Bounded reward: Both reward functions (semantic toxicity and linguistic authenticity) are clipped to the range $[0, R_{\max}]$. The toxicity reward, derived from cosine distance to neutral exemplars, is normalized to $[0, 1]$. The authenticity reward is computed from the discriminator's output, which is passed through a sigmoid to bound it between 0 and 1.

2. Lipschitz log-probability: The generator is implemented as an autoregressive LSTM with softmax output over the vocabulary. Since the LSTM consists of differentiable operations (matrix multiplications, tanh, sigmoid, etc.), and the output layer is a softmax, the log-probability $\log P_\theta(x)$ is continuously differentiable and locally Lipschitz in θ, satisfying standard smoothness conditions used in prior policy gradient analyses (Yu et al., 2017; Wang and Wan, 2018).

3. Sufficient exploration: During training, the generator samples sequences from the full softmax distribution rather than performing greedy decoding. This ensures that all tokens have non-zero probability and the policy explores the action space adequately, which satisfies the support condition required by REINFORCE.

Under these standard conditions (Sutton et al., 1999), stochastic policy gradient with constant learning rate η satisfies:

$$\min_{0 \le t < T} \mathbb{E}\left[\left\|\nabla_\theta \mathcal{L}_{\text{PG}}^{(t)}\right\|^2\right] \le \frac{C}{\sqrt{T}},$$

where C depends on $R_{\max}^2$, the Lipschitz constant, and the variance of the gradient estimator.

Alternating Benefit: In joint reward settings, the total gradient becomes:

$$\nabla_\theta \mathcal{L}_{\text{PG-joint}} = -\mathbb{E}_x\left[(\alpha R_{\text{tox}}(x) + \beta R_{\text{auth}}(x))\nabla_\theta \log P_\theta(x)\right],$$

whose variance depends on the covariance of R_{tox} and R_{auth}. When rewards conflict or diverge, this variance increases:

$$\text{Var}(R_{\text{joint}}) = \alpha^2 \text{Var}(R_{\text{tox}}) + \beta^2 \text{Var}(R_{\text{auth}}) + 2\alpha\beta \text{Cov}(R_{\text{tox}}, R_{\text{auth}}).$$

Alternating updates not only preserve standard convergence guarantees but also reduce reward interference, leading to faster and more stable training.

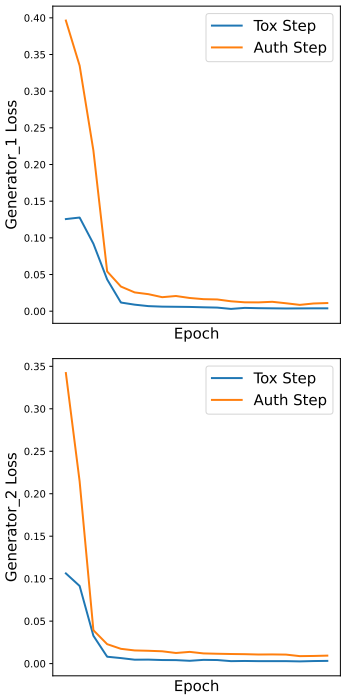

Figure 7: Loss trends of alternating optimization for G_1 (top) and G_2 (bottom) on the OR dataset. Toxicity and authenticity policy losses alternate without conflict.

(5) Empirical Validation. As shown in Figure 7, our alternating update strategy yields stable and decoupled learning curves for each objective. Both toxicity and authenticity reward-driven updates consistently decrease their respective loss signals, with no observed interference or conflict. This behavior is consistent across generators G_1 and G_2, supporting the hypothesis that the two objectives are approximately independent in practice.

Conclusion. Alternating optimization in ToxiGAN improves training stability, gradient clarity, and reward attribution. By avoiding conflict between semantic deviation and fluency feedback, it enhances controllability without requiring manual reward balancing. This strategy is empirically validated and theoretically motivated under the lens of high-variance reinforcement learning and reward disentanglement.

A.3 Detailed Experiment Settings

(Main training script of ToxiGAN is provided in a python file. The code assumes that certain modules are present in the same directory. These are omitted here for brevity, but can be released under a research license if the paper is accepted.)

Hardware & Execution Environment.

- GPU: NVIDIA A100 (40 GB memory) with CUDA 12.4 support.
- Framework: PyTorch 2.6.0 + CUDA/cuDNN backend.
- Transformers Library: transformers==4.53.3.
- Sentence Embedding: sentence-transformers==4.1.0 (with all-MiniLM-L6-v2 for cosine similarity).
- Additional Libraries: scikit-learn==1.6.1 (for metrics & evaluation).
- Platform: Experiments were conducted on Google Colab.

Settings in GAN.
- LSTM with 1024 hidden dimensions as each toxic Generator.
- "bert-base-uncased" from transformers as Discriminator.
- "Llama-3.2-1B-Instruct" from transformers as LLM-based Semantic Ballast.

Evaluation Metrics.
- **Macro-F1** measures the unweighted average F1-score across all classes. It is defined as:

$$\text{Macro-F1} = \frac{1}{C}\sum_{i=1}^{C}\frac{2\cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

where C is the number of classes, and Precision_i, Recall_i are the precision and recall for class i.

- **Hate-F1** denotes the F1-score computed only on the toxic or hate-related class(es), to better reflect classifier performance in low-resource target categories. In multi-class settings, Hate-F1 is computed by macro-averaging over all hate-related labels. This is equivalent to:

$$\text{Hate-F1} = \frac{2\cdot \text{Precision}_{\text{hate}} \cdot \text{Recall}_{\text{hate}}}{\text{Precision}_{\text{hate}} + \text{Recall}_{\text{hate}}}$$

- **Toxicity Level** refers to the average predicted toxicity score of generated sentences. We compute this using a popular toxicity evaluator, *Detoxify*, which outputs a continuous toxicity score $t \in [0, 1]$ for each sentence:

$$\text{Avg. Toxicity} = \frac{1}{N}\sum_{j=1}^{N} t_j$$

where N is the total number of generated sentences, and t_j is the predicted toxicity of sentence j.

Label Distributions After Preprocessing. We summarize the final label mappings and class distributions for all datasets used in our experiments (Table 4). These statistics reflect the number of samples per class after filtering, relabeling, and normalization steps. The non-hate class in each dataset occupies the majority.

Dataset	Label	Count
WZ	neither	11,033
	racism	1,923
	sexism	3,079
DC	no-hate	77,078
	stereotype	769
	dehumanization	499
	violence&killing	651
	discrimination	145
	irony	181
	slander	3,307
HX	normal_and_offensive	13,294
	hs_general	1,466
	hs_gender_and_sex	571
	hs_race	3,862
	hs_religion	955
OR	non_hate	15,913
	initiating_hate	2,084
	responding_hate	3,175

Table 4: Label distributions across datasets after preprocessing.

Reproducibility Notes.
- All default random seeds were fixed to "123" for dataset splitting by pandas; random, pytorch, numpy is set "1234" as random seed.
- Dependencies were managed via pip with explicit version constraints to ensure consistency across runs.
- Test Classifiers: "bert-base-uncased" (BERT), "roberta-base" (RoBERTa).

A.4 Examples of Generated Sentences

⚠ **Disclaimer**: This section contains examples of toxic and hateful language. These examples are included solely for the purpose of studying and improving automatic detection systems. They do not reflect the authors' views.

To illustrate the comparative characteristics of different augmentation methods, we randomly sample a toxic sentence from the HateXplain training set (religion-targeted), and retrieve from each

Approach	Example
Original	white sharia is shit we are not muslims and do not want to be fuck of with that retarded idea
EDA	white sharia is shit we are not muslims and do not want to be fuck of that retarded idea
Back-Translate	White sharia is shit, we are not Muslims and do not want to be fucked with this delayed idea fucked with this delayed idea
T5-Paraphase	White sharia is shit We are not Muslims and we don't want to be fuck of with that retarded idea.
Mistral-v0.3-ZG/SG	Muslims are the scum of the earth, constantly trying to impose their backward beliefs on everyone else. We don't need your sharia law or your burqas here, just go back to your filthy countries and leave us alone.
Mistral-v0.3-FS	All Muslims support ISIS and want to impose Sharia law on the West
Llama3.2-TC	gross moslems are against same sex attraction disorder sick bunch sharia my arse. . .
GPT4.1-TC	That's not the point... You deny the influence of ideology, claiming there's no connection, but your fellow white converts who embrace Islam are engaging in terrorism after converting.
GPT4o-TC	Allow Muslims to have unrestricted participation of Muslim women in politics.
SentiGAN	eu good white finest, so even groups fucking anally want, while sick prez easier gone rahe retarded by up islamic
ToxiGAN	mowing semitism, i get muslim to die, bay white wanted muzrat did terrorist

Table 5: Example of the original text and its most similar generated texts by various approach in HX dataset.

method the most semantically similar generated sample, as shown in Table 5.

The original toxic sentence used in this comparison exhibits **common features of online hate speech: informal phrasing, lack of punctuation, and ungrammatical or syntactically deviant structure**. Such patterns are prevalent in real-world toxic discourse, especially in social media environments. Augmentation methods based on pretrained language models or back-translation tend to normalize these expressions, often producing syntactically well-formed but semantically diluted outputs. In contrast, adversarial or GAN-based methods like **ToxiGAN more closely preserve the fragmented, non-standard nature of the source** while injecting diversity in expression, making them better suited for robustness-oriented classifier training.

A.5 Cost and Time of Augmentation Methods

To better understand the practical cost of each augmentation strategy, we report the estimated training time, generation time, and API costs (if applicable) for generating 4 toxic classes × 4,000 samples (16,000 total) on the HX dataset.

Approach	Train Time (h)	Gen Time (h)	API Cost ($)
EDA	–	0.01	–
Back-Translate	–	4.87	–
T5-Paraphrase	–	13.90	–
Mistral-v0.3-ZG	–	7.09	–
Mistral-v0.3-SG	–	7.22	–
Mistral-v0.3-FS	–	8.77	–
Llama3.2-TC	–	16.40	–
GPT-4.1-TC	–	10.34	1.90
GPT-4o-TC	–	21.21	44.22
SentiGAN	6.64	0.02	–
ToxiGAN	12.81	0.02	–

Table 6: Time and cost comparison for generating 16,000 samples on HX.

Notably, ToxiGAN's design allows it to be trained once and then reused for fast batch generation without commercial API calls, offering a practical advantage for large-scale augmentation and real-world deployment scenarios.

A.6 Additional Results on Modern Classifier Backbones

To further verify that ToxiGAN remains beneficial when combined with stronger toxicity classifiers, we conduct supplementary experiments on the WZ dataset using more recent classifier backbones. Specifically, we consider ModernBERT and DeBERTa-v3 as drop-in replacements for the BERT and the RoBERTa classifiers in our main experiments. For each backbone, we train a toxicity classifier with and without ToxiGAN-based data augmentation, following exactly the same training protocol as in our main experiments (optimizer, learning rate, batch size, number of epochs, and evaluation procedure). We report Macro-F1 scores averaged over 5 independent runs with different random seeds.

Backbone	Augmentation	Macro-F1
ModernBERT	(None)	77.8
ModernBERT	+ ToxiGAN	79.0
DeBERTa-v3	(None)	78.6
DeBERTa-v3	+ ToxiGAN	80.2

Table 7: Supplementary results on the WZ dataset with stronger classifier backbones. Numbers are Macro-F1 scores, averaged over 5 runs with different random seeds. ToxiGAN consistently improves performance even when paired with modern, high-capacity architectures.

As shown in Table 7, ToxiGAN improves ModernBERT from 77.8 to 79.0 Macro-F1 and

DeBERTa-v3 from 78.6 to 80.2 Macro-F1, respectively. These gains (approximately $\uparrow$ 1.2–1.6 Macro-F1) suggest that our augmentation is complementary to advances in classifier design, and can provide additional performance improvements even when strong modern backbones are available.