ToxiTrace: Gradient-Aligned Training for Explainable Chinese Toxicity Detection

Boyang Li[1], **Hongzhe Shou**[1], **Yuanyuan Liang**[2], **Jingbin Zhang**[1], **Fang Zhou**[1]*
[1]School of Data Science and Engineering, East China Normal University
[2]School of International Chinese Studies, East China Normal University
{byli1024, hzshou, jingbinzhang}@stu.ecnu.edu.cn
yyliang@chinese.ecnu.edu.cn, fzhou@dase.ecnu.edu.cn

Abstract

Existing Chinese toxic content detection methods mainly target sentence-level classification but often fail to provide readable and contiguous toxic evidence spans. We propose **ToxiTrace**, an explainability-oriented method for BERT-style encoders with three components: (1) **CuSA**, which refines encoder-derived saliency cues into fine-grained toxic spans with lightweight LLM guidance; (2) **GCLoss**, a gradient-constrained objective that concentrates token-level saliency on toxic evidence while suppressing irrelevant activations; and (3) **ARCL**, which constructs sample-specific contrastive reasoning pairs to sharpen the semantic boundary between toxic and non-toxic content. Experiments show that ToxiTrace improves classification accuracy and toxic span extraction while preserving efficient encoder-based inference and producing more coherent, human-readable explanations. The core training code is available at https://github.com/ZhouF-ECNU/ToxiTrace.

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

1 Introduction

In the era of pervasive digital social media, toxic user-generated content (UGC)—such as cyberbullying and hate speech—has become increasingly prevalent, posing tangible risks to online communities and society at large. As a result, toxic content detection has been extensively studied (Arora et al., 2023; Kirk et al., 2022; Azumah et al., 2023), with Transformer-based pre-trained language models (PLMs) (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019) and, more recently, large language models (LLMs) further advancing classification performance.

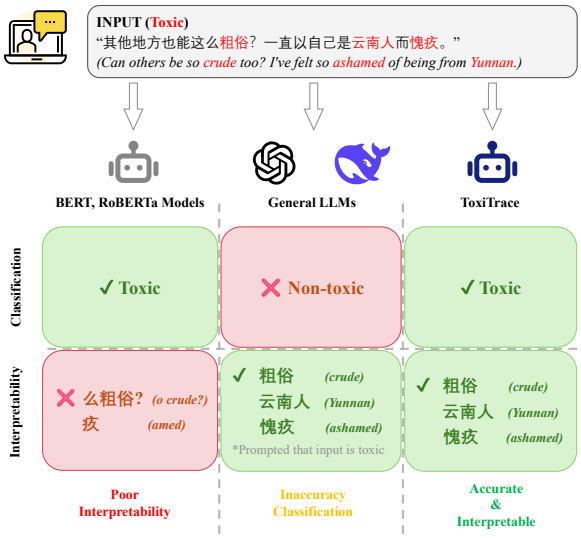

Figure 1: Existing encoder-based detectors struggle to reliably extract fine-grained toxic expressions within a sentence; LLMs can better extract spans when toxicity is given but are limited in direct classification and efficiency; our method preserves classification performance while enabling contiguous toxic span extraction.

Despite these advances, most existing work focuses on sentence-level toxicity classification, while providing little insight into which specific parts of a sentence constitute the toxic content. However, identifying fine-grained toxic evidence is crucial for explainability, moderation transparency, and downstream interventions. This challenge is particularly pronounced for Chinese.

Unlike English, where words serve as the basic semantic units, Chinese toxic expressions are typically realized as multi-character phrases, while individual characters are often semantically ambiguous. However, mainstream Chinese PLMs adopt character-level tokenization (Cui et al., 2020; Sun et al., 2021). Consequently, attribution signals such as gradients or attention weights are fragmented across individual characters rather than coherent semantic spans, producing rationales that are difficult for humans to interpret (Ding and

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 7121–7138

Koehn, 2021).

As a result, existing Chinese toxic content detection approaches—despite achieving strong sentence-level performance through fine-tuning (Deng et al., 2022), knowledge distillation (Deng et al., 2023), or glyph-aware modeling (Wullach et al., 2022; Xue et al., 2025)—remain limited in their ability to accurately extract the true toxic expressions within a sentence (Figure 1, left). In contrast, LLMs often exhibit stronger capabilities in explanation and span extraction (Creswell and Shanahan, 2022; Chuang et al., 2025), but they typically underperform on direct toxicity classification and incur substantially higher inference costs (Schmidhuber and Kruschwitz, 2024; Zhang et al., 2025; Sun et al., 2023) (Figure 1, middle).

These limitations call for a method that preserves the classification strength and efficiency of encoder-based models while enabling reliable extraction of contiguous, human-readable toxic spans (Figure 1, right). To this end, we propose **ToxiTrace**, a span-extraction-oriented framework built on BERT-style encoders for Chinese toxic content detection, designed to produce coherent and interpretable toxic rationales without requiring fine-grained span supervision. Our main contributions are as follows:

- We propose a cue-guided span annotation strategy with gradient-aware training, which leverages attribution signals from encoders to induce consistent saliency on toxic tokens without requiring explicit span annotations.
- We introduce a bidirectional cliff-based span extraction algorithm to identify contiguous toxic spans based on saliency transitions, alleviating the span fragmentation issue inherent in prior top-k selection methods.
- We develop an adversarial reasoning contrastive learning objective with adaptive InfoNCE, which aligns sample-specific toxic and non-toxic reasoning representations, sharpening the semantic boundary and further enhancing span-level interpretability.
- Experiments on multiple Chinese toxic content benchmarks demonstrate that **ToxiTrace** consistently outperforms strong baselines.

2 Related Work

2.1 Toxic Content Detection

Toxic content detection has evolved from early Bag-of-Words and conventional classifiers (Kwok and Wang, 2013; Waseem and Hovy, 2016; Davidson et al., 2017) to neural and Transformer-based models that achieve strong sentence-level accuracy (Badjatiya et al., 2017; Zimmerman et al., 2018; Caselli et al., 2021; Sarkar et al., 2021).

Chinese toxic content detection follows a similar trajectory, supported by datasets such as COLD (Deng et al., 2022), ToxiCN (Lu et al., 2023), and CNTP (Yang et al., 2025b), as well as encoder-centric methods including character-level modeling (Wullach et al., 2022), domain feature fusion (Zhang et al., 2024), LLM-assisted rewriting (Chao et al., 2024), and distillation for robustness (Deng et al., 2023). In contrast to prior work that primarily optimizes sentence-level detection, we target fine-grained toxic span extraction by training an encoder to produce evidence-aligned saliency and readable spans.

Although CNTP (Yang et al., 2025b) provides limited span annotations, most Chinese detection pipelines still lack reliable supervision and methods for extracting *contiguous* toxic spans within sentences. This gap leaves models accurate yet poorly grounded in human-readable evidence. We address this by using cue-guided span signals to support training under weak supervision and by enforcing higher saliency on toxic evidence.

2.2 Attribution Method

Attribution methods, initially developed in computer vision, have been widely adopted for NLP interpretability. Representative lines include perturbation-based explanations such as LIME (Ribeiro et al., 2016), gradient-based explanations (Ross et al., 2017), and task-calibrated attention-saliency alignment that improves faithfulness (Chrysostomou and Aletras, 2021a,b). A practical issue is that many pipelines extract explanations by selecting top-k salient tokens, where k is either fixed (Jesus et al., 2021; Bastings et al., 2022) or set by heuristics such as a fixed fraction of sentence length (Krishna et al., 2025); dynamic alternatives such as peak-based top-k have been proposed to improve consistency (Kamp et al., 2023). Building on gradient-based attribution, we explicitly *train* the encoder to yield more evidence-

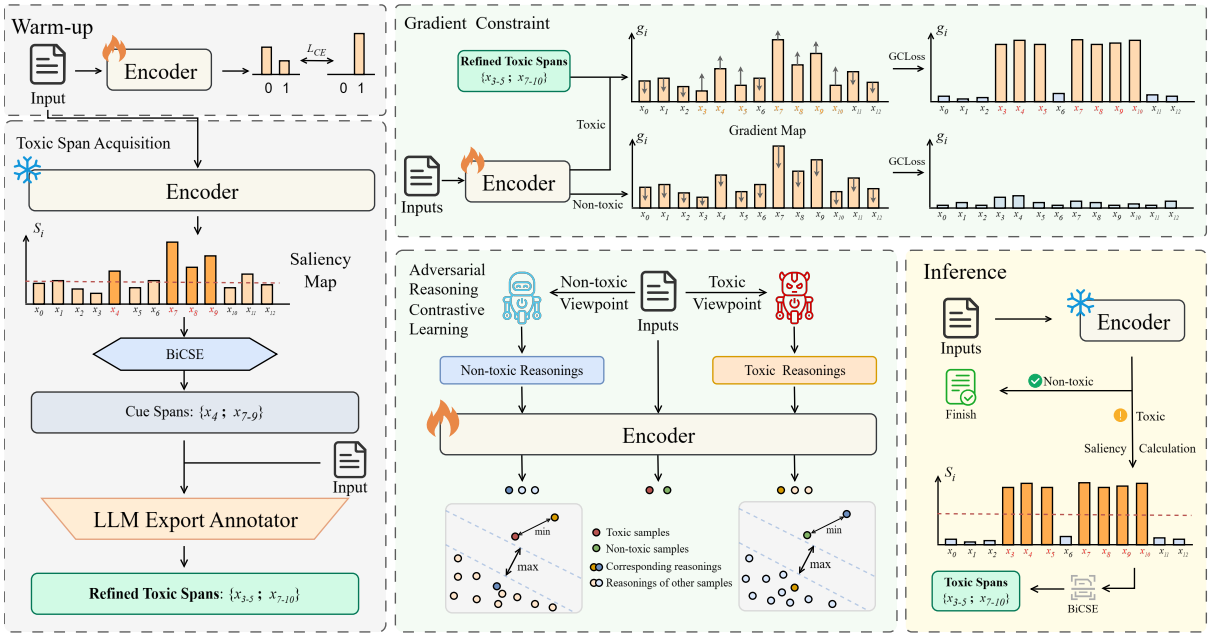

Figure 2: Framework of the proposed ToxiTrace method. During training, we warm up an encoder classifier, acquire weak span annotations with CuSA using BiCSE-extracted saliency cues, and jointly optimize GCLoss and ARCL to concentrate saliency on toxic evidence. During inference, the model predicts toxicity and, for toxic inputs, extracts contiguous spans via BiCSE from the saliency map.

aligned gradients, rather than only post-hoc selecting salient tokens.

Despite progress, top-k selection methods often produces fragmented highlights and cannot recover contiguous, human-readable spans—an issue that especially pronounced under character-level tokenization. We therefore propose a bidirectional scanning algorithm that identifies consecutive locally high-saliency spans, enabling stable contiguous toxic span extraction beyond discrete token selection.

3 Methodology

As shown in Figure 2, our ToxiTrace framework contains following four steps: (1) We first warm up the encoder with standard classification training to obtain robust sentence-level discrimination. (2) After warming-up, we compute saliency maps and apply a **Bi**directional **C**liff-based **S**pan **E**xtraction algorithm (BiCSE) to obtain initial high-saliency spans, which are used as cues to prompt an LLM to refine boundaries and recover coherent toxic spans, yielding *refined toxic spans* as weak span annotations. (3) We then introduce **G**radient **C**onstraint **Loss** (GCLoss) to explicitly increase gradient responses on toxic evidence while suppressing spurious activations on non-toxic tokens, shaping a more concentrated and extractable saliency. (4) In parallel, we adopt **A**dversarial **R**easoning **C**ontrastive **L**earning (ARCL) with adaptive InfoNCE to align each input with sample-specific reasoning of opposing stances, sharpening the toxic/non-toxic semantic boundary.

At inference time, the model first predicts toxicity; if toxic, the saliency map will be calculated, and toxic spans will be extracted using BiCSE.

3.1 Cue-guided Span Annotation (CuSA)

Existing toxic content datasets only have coarse-grained labels (toxic/non-toxic) and cannot accurately locate toxic spans. CuSA constructs span-level signals by using the model's attribution map as cues and letting an LLM refine span boundaries. It consists of two steps: (1) warm-up fine-tuning to obtain a reliable sentence-level classifier; and (2) cue-guided span refinement, where we feed the toxic text together with initially extracted salient spans as cues to an LLM for span annotation.

Warm-up training. We train the encoder with binary cross-entropy. Given an input text sequence $\mathcal{X} = \{x_1, \ldots, x_n\}$, the embedding layer maps tokens to $\mathcal{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n]$, the model ϕ generate its contextual representation $\mathcal{H} = \phi(\mathcal{E}) = [\boldsymbol{h}^{cls}, \boldsymbol{h}^1, \boldsymbol{h}^2, ...\boldsymbol{h}^n]$. This representation is then fed into a classification head ψ to yield the predicted probability $P(y|\mathcal{X}) = \psi(\boldsymbol{h}^{cls})$.

Saliency cues for span annotation. After warm-up, following existing attribution methods (Chrysostomou and Aletras, 2021a; Sikdar et al., 2021), we compute a token-level *saliency* score for each token and form a saliency map, which serves as cues for span extraction. The saliency s_i of each token x_i can be calculated via Eq. (1):

$$s_i = \left\| \boldsymbol{e}_i \odot \frac{\partial \log P(y|\mathcal{X})}{\partial \boldsymbol{e}_i} \right\|_2 . \tag{1}$$

Existing attribution methods select the top-k most salient tokens (Kamp et al., 2023); however, these selected tokens tend to be scattered. To address this limitation, we propose BiCSE (detailed in Appendix A), a bidirectional cliff-based scanning algorithm that tracks saliency transitions to identify the optimal start and end boundaries of contiguous spans. Moreover, longer and more continuous toxic spans are captured by taking the union of results from two sequential scans (left-to-right and right-to-left).

To help the LLM to find more potential toxic spans, both the raw text and the initially extracted spans (cues) $\mathcal{T}_{cue}$ are fed into the LLM. In our experiments, we use Gemini 2.5 Pro (Team, 2025) as an expert annotator to integrate and refine these cues. The refined toxic spans are formulated as: $\mathcal{T}_{refined} = \text{LLM}(\boldsymbol{x}, \mathcal{T}_{cue})$. By leveraging the LLM's superior interpretability, we can achieve more accurate annotation of toxic spans.

3.2 Gradient Constraint Loss

The fine-tuned model attains satisfactory classification performance overall; yet, its token-level toxicity discrimination remains imprecise (Appendix C shows the saliency maps before and after applying our ToxiTrace method). CuSA provides refined toxic spans as annotations, which allow us to explicitly shape the model's token-level attribution for span extraction. Concretely, GCLoss consists of two complementary components: (i) a **Pairwise Gradient Ranking (PGR)** term that enforces a margin between toxic and non-toxic tokens, and (ii) a **Push–Pull Threshold (PPT)** term that regularizes their gradient ranges within each sample. During training, both terms operate on the gradient norm of the log-predicted probability with respect to the *input embeddings* $\boldsymbol{e}_i$:

$$g_i = \left\| \frac{\partial \log P(y|\mathcal{X})}{\partial \boldsymbol{e}_i} \right\|_2 \tag{2}$$

as the training signal to increase responses on toxic spans and suppress spurious activations on non-toxic tokens, thereby shaping more concentrated and extractable attribution.

PGR Loss. The objective of this loss is to penalize cases where, within a single sentence, the gradient of a toxic token exceeds that of a non-toxic token by a margin smaller than the predefined threshold m. The specific formula is given as follows:

$$\mathcal{L}_{PGR} = \frac{1}{|\mathcal{P}| \cdot |\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \max(0, g_j - g_i + m), \tag{3}$$

where $\mathcal{P}$ and $\mathcal{N}$ denote the sets of toxic and non-toxic tokens, respectively; g_i, g_j is calculated according to Eq. (2); and m is set to 1 in this study.

PPT Loss. The PGR Loss introduced above captures the relative gradient relationship between toxic and non-toxic tokens, yet it cannot constrain the gradient value ranges of either token type. Given the substantial discrepancies in gradient scores across different sentences, employing a fixed range to regulate the gradients of toxic and non-toxic tokens is inherently flawed. To address this dual limitation, we propose the intra-sentence PPT Loss that leverages gradient information of tokens within each single sample to separately guide the gradient learning of toxic and non-toxic tokens. Specifically, we first calculate the 15th percentile of the gradient values of all tokens in a single sample as the threshold τ, which serves to constrain the gradient values of non-toxic tokens to stay below this threshold. The detailed formula is given as follows:

$$\mathcal{L}_{neg} = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} [\max(0, g_j - \tau)]. \tag{4}$$

For toxic tokens, we expect their gradient values to fall within a relatively high range. Thus, we adopt the maximum gradient $g_{\max}$ as the reference and set $\alpha \cdot g_{\max}$ as the lower bound, where α denotes a positive target coefficient. Our goal is to push the gradient values of toxic tokens above this threshold. However, given the possibility of an excessively large $g_{\max}$, we further introduce a gradient cap τ_{cap} to prevent gradient explosion caused by overly high gradient values of toxic tokens. The formula for toxic tokens is given as follows:

$$\mathcal{L}_{pos} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} [\max(0, t_n - g_i)], \tag{5}$$

where the target value $t_n = \min(\alpha \cdot g_{max}, \tau_{cap})$. The overall form of the PPT Loss is:

$$\mathcal{L}_{PPT} = \frac{1}{2}(\mathcal{L}_{pos} + \mathcal{L}_{neg}) \tag{6}$$

Remark. GCLoss constrains *gradients* during training (Eq. (2)), because gradients directly capture the model's sensitivity to token-level evidence. For span extraction at inference time, we compute a *saliency* score with the same embedding-level gradients.

This formulation reflects both sensitivity and contribution, and BiCSE is applied to the saliency sequence $\{s_i\}_{i=1}^{n}$ to produce contiguous spans.

3.3 Adversarial Reasoning Contrastive Learning

While the aforementioned loss functions address token-level gradient constraints, they are confined to individual sentences and cannot capture the semantic boundary differentiating toxic from non-toxic sentences. Inspired by the work of (Rusak et al., 2025), we adopt an adaptive InfoNCE loss to implement semantic contrastive learning. Existing data augmentation methods (Zhou et al., 2021; Hu et al., 2023; Fang et al., 2024) primarily rely on substituting, modifying or add noise to certain words in a sentence. Such operations lack a thorough understanding of the sentence's inherent semantics. To address this limitation, we propose leveraging the **LLM debate mechanism** (Moniri et al., 2025) to generate **adversarial reasoning** as augmented samples. This approach enables the model to better explore the intrinsic semantic information of sentences and learn to distinguish the differences between toxic and non-toxic texts and sharpening the semantic boundary.

Specifically, we first use two opposing reasoning prompts: *"Assuming the text is {toxic, normal}, generate explanations to support this judgment"*[1], and feed them to the LLM (Gemini 2.5 Flash) to obtain semantically augmented positive and negative samples. This way, the model can produce targeted reasoning content, instead of generating generic descriptions such as *"This sentence is just an exaggerated joke and does not intend to attack any group"*.

Using the two opposing reasoning prompts obtained above, we adopt the following contrastive loss to facilitate sentence-level semantic learning:

$$\mathcal{L}_{\{tox,nor\}} = -\log \frac{\exp(\boldsymbol{h}_t \cdot \boldsymbol{h}_{\{p,n\}}/\tau)}{\exp(\boldsymbol{h}_t \cdot \boldsymbol{h}_{\{p,n\}}/\tau) + \sum_{\boldsymbol{h}^k_{\{n,p\}} \in \mathcal{B}} \exp(\boldsymbol{h}_t \cdot \boldsymbol{h}^k_{\{n,p\}}/\tau)}, \tag{7}$$

where $\boldsymbol{h}_t$, $\boldsymbol{h}_p$ and $\boldsymbol{h}_n$ denote the semantic embeddings of the target text, its positive reasoning explanation and its negative reasoning explanation, respectively; $\boldsymbol{h}_p^k$ and $\boldsymbol{h}_n^k$ denote the final-layer `[CLS]` token embeddings of the positive and negative reasoning explanations corresponding to other toxic and non-toxic sentences within the training batch $\mathcal{B}$. τ denotes the temperature parameter, which is set to 0.05 in our experiments.

The overall semantic contrastive loss is defined as the average of these two loss components:

$$\mathcal{L}_{con} = \frac{1}{2}(\mathcal{L}_{tox} + \mathcal{L}_{nor}) \tag{8}$$

3.4 Joint Training Objective

In the joint training phase, we simultaneously optimize three loss components: the gradient-based binary classification loss, the gradient constraint losses, and the contrastive learning loss. The overall training objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_{grad}(\mathcal{L}_{PGR} + \mathcal{L}_{PPT}) + \lambda_{sem}\mathcal{L}_{con}, \tag{9}$$

where λ_{grad} and λ_{sem} are hyperparameters that balance the contributions of gradient constraint losses and semantic contrastive loss, respectively.

The overall training pipeline is designed as follows: (1) First, perform a warm-up training using the cross-entropy loss to equip the model with basic toxicity classification capability. (2) Subsequently, the GCLoss is introduced to enforce the model to generate higher gradient values for toxic tokens and lower gradient values for non-toxic tokens. (3) Meanwhile, ARCL is adopted to sharpen the semantic boundary between toxic and non-toxic texts. Detailed training hyperparameters are provided in appendix B.

4 Experiments

4.1 Experimental Setup

Dataset. For the binary toxicity classification task, we evaluate our model on two Chinese datasets: COLD (Deng et al., 2022) (32,480 instances in total, with 5,323 test instances) and ToxiCN (Lu et al., 2023) (12,011 instances in total, with 2,411 test instances).

[1]Detailed prompt templates are provided in Appendix D.

Models	COLD					ToxiCN				
	Acc	*R*	*P*	F_1	**Macro-F_1**	*Acc*	*R*	*P*	F_1	**Macro-F_1**
Qwen3-8B	$74.03_{0.23}$	$59.90_{0.28}$	$70.15_{0.32}$	$64.62_{0.19}$	$72.33_{0.21}$	$71.44_{0.33}$	$82.94_{0.29}$	$69.13_{0.27}$	$75.41_{0.16}$	$71.51_{0.12}$
LLaMA3.1-8B	$73.91_{0.31}$	$51.46_{0.29}$	$74.58_{0.35}$	$60.90_{0.22}$	$72.00_{0.28}$	$65.18_{0.28}$	$44.29_{0.29}$	$81.61_{0.24}$	$57.42_{0.20}$	$68.25_{0.22}$
DeepSeek-V3.2	$74.23_{0.16}$	$59.23_{0.10}$	$70.91_{0.13}$	$64.55_{0.12}$	$72.51_{0.09}$	$60.66_{0.11}$	$36.19_{0.14}$	$77.35_{0.16}$	$49.30_{0.08}$	$64.14_{0.11}$
GPT-4o	$74.49_{0.18}$	$67.79_{0.20}$	$66.98_{0.21}$	$67.38_{0.12}$	$73.22_{0.13}$	$73.20_{0.21}$	$65.66_{0.18}$	$66.98_{0.16}$	$67.38_{0.11}$	$73.22_{0.09}$
RoBERTa	$82.68_{0.42}$	$86.37_{0.86}$	$74.42_{0.86}$	$79.80_{0.27}$	$82.56_{0.38}$	$82.70_{0.45}$	$84.38_{0.30}$	$83.40_{0.55}$	$83.89_{0.40}$	$82.81_{0.45}$
Qwen3-8B (SFT)	$82.25_{0.22}$	$82.68_{0.22}$	$75.02_{0.38}$	$78.66_{0.16}$	$81.87_{0.10}$	$82.08_{0.54}$	$83.52_{1.33}$	$82.74_{0.93}$	$83.12_{0.52}$	$82.02_{0.42}$
LLaMA2-7B (SFT)	$82.43_{0.28}$	$86.28_{0.18}$	$73.78_{0.22}$	$79.54_{0.10}$	$82.46_{0.13}$	$81.25_{0.37}$	$84.62_{0.44}$	$80.81_{0.40}$	$82.67_{0.29}$	$81.18_{0.22}$
LLaMA3.1-8B (SFT)	$83.00_{0.10}$	$87.13_{0.16}$	$74.37_{0.18}$	$80.19_{0.09}$	$82.19_{0.10}$	$83.02_{0.17}$	$86.08_{0.20}$	$82.52_{0.24}$	$84.26_{0.12}$	$82.96_{0.14}$
LLaMA3.1-8B + ToxiTrace	$82.15_{0.24}$	$\mathbf{88.42}_{0.32}$	$72.52_{0.37}$	$79.68_{0.20}$	$81.89_{0.27}$	$82.33_{0.47}$	$84.93_{0.53}$	$82.22_{0.42}$	$83.55_{0.35}$	$82.23_{0.32}$
Qwen3-8B + ToxiTrace	$82.96_{0.29}$	$82.49_{0.37}$	$\mathbf{76.38}_{0.41}$	$79.41_{0.25}$	$82.39_{0.21}$	$82.91_{0.62}$	$83.44_{0.85}$	$\mathbf{84.10}_{0.77}$	$83.77_{0.31}$	$82.86_{0.36}$
RoBERTa + ToxiTrace	$\mathbf{83.84}_{0.27}$	$86.19_{0.48}$	$76.14_{0.34}$	$\mathbf{80.85}_{0.14}$	$\mathbf{83.68}_{0.16}$	$83.62_{0.15}$	$87.05_{1.28}$	$82.82_{0.78}$	$84.88_{0.27}$	$83.56_{0.14}$
MacBERT + ToxiTrace	$83.22_{0.19}$	$86.19_{0.48}$	$75.10_{0.49}$	$80.27_{0.10}$	$83.13_{0.17}$	$\mathbf{83.87}_{0.12}$	$\mathbf{87.99}_{0.83}$	$82.61_{0.41}$	$\mathbf{85.21}_{0.20}$	$\mathbf{83.83}_{0.14}$

Table 1: The average classification results of different models on COLD and ToxiCN datasets(%) across 3 independent runs. Subscripts denote standard deviations.

For toxic span extraction, we use the span annotations provided in CNTP (Yang et al., 2025b) as gold labels, which include 2,533 samples annotated with toxic spans, to assess the model's ability to extract fine-grained toxic expressions within toxic sentences.

Models and implement. To verify the effectiveness of our proposed method in both binary classification and span-level extraction, we applied both fine-tuning solely on binary classification labels and our training strategy across different BERT-based models: Chinese-RoBERTa-wwm-ext **(RoBERTa)** and Chinese-MacBERT-base **(MacBERT)** (Cui et al., 2020). Concurrently, a set of LLMs were chosen to conduct comparative analysis: **LLaMA2-7B** (Touvron et al., 2023), **LLaMA3.1-8B** (Grattafiori et al., 2024), **Qwen3-8B** (Yang et al., 2025a), **DeepSeek-V3** (Liu et al., 2024) and **GPT-4o** (Achiam et al., 2023).

For open-source LLMs, both their direct inference capabilities and their performance after fine-tuning on the respective datasets (denoted as SFT) were evaluated. Fine-tuning of LLMs was conducted using the open-source toolkit LLaMA-Factory[2] to implement LoRA (Hu et al., 2022). Closed-source LLMs were evaluated with zero-shot setting; the prompt templates used were detailed in Appendix D. Models denoted with "(ToxiTrace)" were trained using the method proposed in this paper. DeepSeek and GPT-4o were accessed via official APIs, while all other experiments were conducted using four NVIDIA A800 80GB GPUs.

Evaluation metrics. To evaluate toxic content detection performance, we used five widely adopted metrics: Accuracy (*Acc*), Recall (R), Precision (P), F_1 and Macro-F_1 Score.

[2]https://github.com/hiyouga/LLaMA-Factory

4.2 Overall Classification Performance

The classification performance of ToxiTrace and competing methods is summarized in Table 1. Overall, ToxiTrace achieves the best results across all five metrics on both COLD and ToxiCN. In the zero-shot setting, both open- and closed-source LLMs perform poorly on Chinese toxicity classification (and LLaMA2-7B fails to complete the task due to strict safety alignment), whereas fine-tuning brings LLMs to a level comparable with encoder-based models. In terms of efficiency, encoder models finish inference within 20 seconds on both datasets, while LLMs require 2–9 minutes, reflecting differences in model scale, architecture, and the use of LoRA adapters.

4.3 Applicability to LLMs

Since LLMs are also Transformer-based, we conduct an exploratory study on transferring ToxiTrace to decoder-only LLMs. Due to resource constraints, LoRA was applied for parameter-efficient adaptation; we then replaced the decoding head with a binary classification head and optimized the same ToxiTrace objectives. As shown in Table 1, ToxiTrace achieves performance comparable to instruction fine-tuning on LLM, yet still falls short of encoder-based models trained with ToxiTrace. One possible reason is that LoRA updates only a tiny fraction of parameters (typically $\leq$ 0.5%), limiting its ability to reshape embedding-level gradients distributions required by our gradient-oriented training. We leave a systematic study of more effective parameter-efficient gradient shaping for future work.

Models	Overlap			Character-level				Inference Time
	R	P	F_1	R	P	F_1	IoU	
CRF (Liu et al., 2021)	44.48	**78.01**	56.66	39.24	**81.85**	53.05	36.10	01m 10s
LIME (Ribeiro et al., 2016)	41.45	35.99	38.53	81.62	33.93	47.93	31.52	27m 20s
Attention (Yamada et al., 2020)	39.01	36.15	37.52	25.43	47.68	33.17	19.88	00m 42s
IG (Sikdar et al., 2021)	58.30	60.01	59.14	36.99	76.46	49.86	33.21	About 1.2 hours
Llama3.2-3B	41.03	71.02	52.01	37.47	69.42	48.67	32.16	08m 06s
Qwen3-0.6B	73.68	67.10	70.23	73.15	62.80	68.23	51.78	09m 20s
Qwen3-1.7B	77.53	71.48	74.38	76.01	66.72	71.29	55.11	09m 28s
Llama-3.1-8B	81.32	69.21	74.78	79.94	62.97	70.87	54.33	12m 56s
Qwen3-8B	84.83	71.97	77.87	**90.03**	63.89	74.74	59.67	14m 33s
RoBERTa	42.37	68.42	52.34	45.68	50.16	43.02	33.63	01m 58s
RoBERTa*	71.27	59.87	65.08	58.29	61.82	55.20	42.86	01m 58s
RoBERTa + ToxiTrace*	**86.36**	70.95	**77.90**	83.53	70.06	**77.63**	**61.56**	01m 58s
Gemini2.5-Pro	86.50	75.09	80.39	85.98	74.24	79.67	66.22	About 1.5 hours

Table 2: Toxic span extraction performance on CNTP. Results without * correspond to extracting the top-15% most salient tokens, while results with * use our proposed bidirectional algorithm to extract high-saliency spans. Entries shaded in cyan indicate the results of the "refiner" model used in the CuSA.

4.4 Toxic Span Extraction Performance

We evaluate toxic span extraction on CNTP using two types of metrics: overlap-based matching between the extracted spans and the ground-truth spans, and character-level Recall, Precision, F_1-score, and IoU, denoted as *Overlap* and *Character-level* in Table 2, respectively. For overlap-based evaluation, a prediction is counted as correct if its overlap with the gold span exceeds 50% (e.g., ground-truth toxic span: " 河南人", pred: " 河南" is correct, while " 南" is incorrect).

Table 2 reports the extraction results under both evaluation schemes. Here, CRF is trained using LLM-generated span annotations; LIME, high-attention token extraction, and IG are computed on a RoBERTa model trained only with binary classification labels. Under both metrics, compared with existing attribution methods, LLMs as well as our model achieve substantially better span extraction performance. Replacing top-15% token selection with BiCSE also yields a large gain for RoBERTa (RoBERTa → RoBERTa*), confirming the advantage of extracting *contiguous* spans. Training with ToxiTrace further yields an additional 12% improvement in F_1-score under both evaluation schemes and a 19% gain in character-level IoU. These results indicate that CuSA and the gradient-shaping objectives (GCLoss/ARCL) produce saliency that is better aligned with the underlying evidence for extraction.

Across LLMs, extraction generally improves with model size, but RoBERTa + ToxiTrace* achieves comparable or better F_1 than the best LLM (Qwen3-8B) while requiring only about $1/7$ of its inference time, demonstrating a substantially better accuracy–efficiency trade-off, the detailed prompt template is provided in Appendix D.

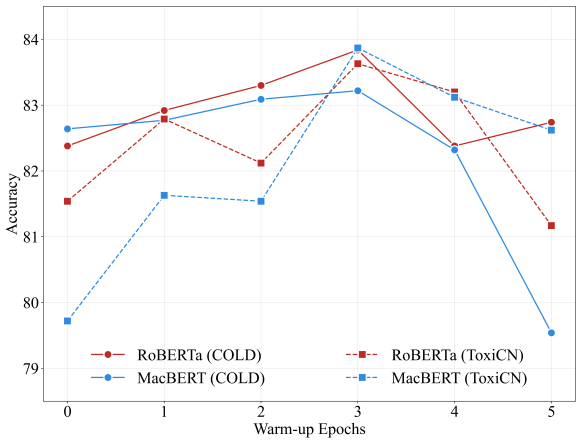

Figure 3: Final Accuracy with different Warm-up Epochs. The models achieve optimal classification performance when the warm-up steps are set to 3.

4.5 Ablation Study

We investigate the contributions of the key components in ToxiTrace through an ablation study with three variants, as shown in Table 3. Specifically, *Full* denotes the complete model that jointly optimizes GCLoss and ARCL, while *w/o CuSA* meaning BiCSE cues only, without any LLM refinement, *w/o ARCL* and *w/o GCLoss* remove the adversarial reasoning contrastive objective and the gradient constraint loss, respectively. We further include the vanilla *RoBERTa* as the baseline.

Since the ablation trends are consistent across COLD and ToxiCN, we only report results on COLD due to space constraints. Overall, removing either module leads to performance degradation in

Models	Classification					Extraction		
	Acc	*R*	*P*	F_1	Macro-F_1	*R*	*P*	F_1
ToxiTrace	**83.84**	86.19	76.14	**80.85**	**83.68**	**86.36**	70.95	**77.90**
w/o CuSA	83.26	83.77	**76.27**	79.85	82.90	65.61	**79.67**	71.96
w/o ARCL	83.13	86.71	74.72	80.27	83.12	82.55	68.99	75.16
w/o GCLoss	83.20	**88.04**	74.29	80.58	83.36	75.89	57.07	65.15
RoBERTa	82.75	86.43	74.24	79.87	82.76	71.27	59.87	65.08

Table 3: Ablation study results. "Classification" denotes results for the binary classification task on the COLD dataset; "Extraction" denotes results using the toxic span annotations in CNTP (Yang et al., 2025b) as ground truth labels.

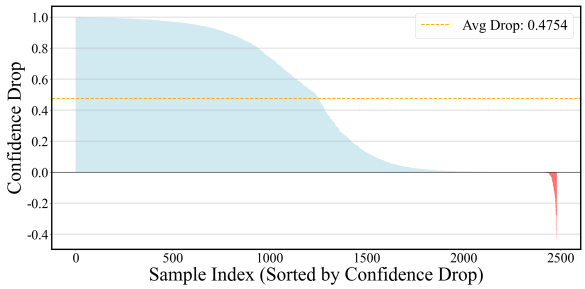

Figure 4: Confidence drop after masking BiCSE-extracted toxic spans for the RoBERTa baseline.

Figure 5: Confidence drop with random masking that matches the number of tokens extracted by BiCSE for RoBERTa trained with ToxiTrace.

both classification and extraction. For classification, removing ARCL reduces Macro-F_1 by 0.56% (and slightly lowers F_1), indicating that ARCL provides a consistent gain via semantic regularization. Removing GCLoss yields a smaller but noticeable drop in Macro-F_1 (0.32%), while increasing recall and decreasing precision, suggesting that without gradient shaping the model becomes more prone to over-predict toxic instances and thus sacrifices precision.

For toxic span extraction, GCLoss has a substantially larger impact than ARCL. Removing ARCL causes a moderate degradation, with extraction F_1 dropping by 2.74% and both recall and precision declining accordingly. In contrast, removing GCLoss leads to a much sharper performance drop: extraction F_1 decreases by 12.75%, accompanied by substantial reductions in both recall and precision. When CuSA is removed and the raw high-saliency tokens are used directly as gradient-boosting targets, span extraction recall drops markedly, while precision increases to a certain extent. This suggests that a model trained only with binary classification labels can still capture some relatively salient toxic expressions, but learning a broader range of toxic expressions requires additional external supervision. Finally, compared to the RoBERTa baseline, the full model improves classification Macro-F_1 by 0.92% and boosts extraction F_1 by 12.82%, demonstrating the effectiveness of our joint training strategy.

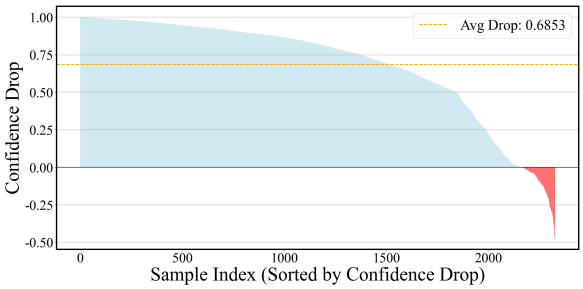

Figure 6: Confidence drop after masking BiCSE-extracted toxic spans for RoBERTa trained with ToxiTrace.

4.6 Effect of Warm-up Steps

Since ToxiTrace training pipeline requires warming up the foundation model for several epochs, this section analyzes the impact of the number of warm-up steps on the final classification and extraction results.

The number of warm-up steps determines the extent to which the model learns toxicity classification solely from binary labels. As shown in Figure 3, too few or too many warm-up steps lead to a decrease in the final classification accuracy.

4.7 Gradient Attribution Faithfulness

In NLP, the faithfulness of explanations concerns whether the highlighted evidence truly reflects the causal decision process, rather than merely correlating with its prediction. Recent work argues that faithful explanations should be grounded in counterfactual reasoning and formalizes this intuition via *order-faithfulness*, showing that non-causal feature-importance methods can mis-rank evidence under confounding effects (Gat et al., 2024). Motivated by this view, we evaluate our gradient-based toxic span explanations from a causal perspective: if the extracted spans constitute genuine decision evidence, masking them should substantially reduce the model's predicted toxicity confidence.

We extract toxic spans on the COLD dataset using two RoBERTa models, both localized by BiCSE: (1) a baseline RoBERTa trained with binary cross-entropy loss, and (2) a RoBERTa trained with our full objective in Eq. (9).

Figures 4 and 6 report the confidence drop, defined as the decrease in the predicted toxicity probability after masking the extracted spans. Compared with the BCE-only baseline, the ToxiTrace-trained model exhibits a consistently larger confidence reduction when its predicted spans are masked. Meanwhile, Figure 5 shows that randomly masking the same number of tokens extracted by BiCSE increases the average confidence from 0.8873 to 0.9019, corresponding to an average "drop" of -1.64%.

These results indicate that the extracted spans are more critical to the model's decision and therefore more faithful to its prediction.

5 Discussion of Adaptability to Other Languages

Our primary focus is on languages where characters are the basic units, because for word-based languages (like English), standard binary classification training often already leads to gradients concentrating on words of higher importance (even if the distribution is somewhat scattered) (Kamp et al., 2023). In contrast, for character-based languages, meaning typically emerges only across multiple consecutive characters, making contiguous span extraction more critical.

Given data availability, we choose Chinese as the main target language due to the existence of relatively large-scale datasets. We will treat extending ToxiTrace to other character-based languages like Japanese and Korean as future work; if we can obtain sufficiently large datasets in these languages, we will conduct further analysis and adaptation studies.

6 Conclusion

We proposed ToxiTrace, which can automatically produce span-level annotations when fine-grained Chinese toxicity labels are unavailable. It further introduces a gradient-based loss to increase the model's gradient responses on fine-grained toxic spans, and aligns each sentence with its corresponding reasoning in the semantic space, achieving simultaneous improvements in classification accuracy and interpretability. In addition, we designed an algorithm for extracting high-saliency token spans, which addresses the limitations of prior methods that cannot recover contiguous high-saliency spans in character-based languages.

Acknowledgments

This work was supported by the Shanghai Science and Technology Innovation Action Plan Project (No.23511100700).

Limitations

In real-world scenarios, some toxic speeches are "cloaked" or obfuscated, such as through the use of homophones or Pinyin replacements (as noted in CNTP). Most models experience a performance degradation when processing such obfuscated toxic information. While datasets for this exist in the Chinese domain, our current method does not specifically address these perturbations. We intend to incorporate robustness against such variations into the scope of our future research.

Our method has been developed and evaluated only on Chinese toxic content, which has unique linguistic properties (e.g., character-level tokenization and ambiguous semantic units). As a result, its effectiveness on languages with different structures remains to be verified.

Ethical Considerations

This work addresses toxic content detection and explanation, which necessarily involves exposure to offensive and harmful language. Although the proposed method is designed to improve the faithfulness and transparency of model explanations, there

is a potential risk that fine-grained toxic span extraction could be misused to reverse-engineer moderation systems or to craft adversarial toxic expressions that evade detection. To mitigate this risk, we emphasize that the proposed framework is intended for research and system auditing purposes, rather than as a fully automated content moderation solution, and should be deployed with appropriate human oversight.

All datasets used in this study (COLD, ToxiCN, and CNTP) are publicly available benchmark datasets released for academic research. They do not contain personally identifying information, and no additional data collection is conducted in this work. While the datasets do include offensive and toxic language by design, they are used solely for model training and evaluation in a controlled research setting. We do not attempt to identify, target, or profile any individuals, and no user-level or sensitive attributes are inferred or stored.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arnav Arora, Preslav Nakov, Momchil Hardalov, Sheikh Muhammad Sarwar, Vibha Nayak, Yoan Dinkov, Dimitrina Zlatkova, Kyle Dent, Ameya Bhatawdekar, Guillaume Bouchard, and 1 others. 2023. Detecting harmful content on online platforms: what platforms need vs. where research efforts go. *ACM Computing Surveys*, 56(3):1–17.

Sylvia W Azumah, Nelly Elsayed, Zag ElSayed, and Murat Ozer. 2023. Cyberbullying in text content detection: an analytical review. *International Journal of Computers and Applications*, 45(9):579–586.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25.

August F.Y. Chao, Chen-Shu Wang, Bo-Yi Li, and Hong-Yan Chen. 2024. From hate to harmony: Leveraging large language models for safer speech in times of covid-19 crisis. *Heliyon*, 10(16):e35468.

George Chrysostomou and Nikolaos Aletras. 2021a. Enjoy the salience: Towards better transformer-based faithful explanations with word salience. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8189–8200.

George Chrysostomou and Nikolaos Aletras. 2021b. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 477–488.

Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Mengnan Du, Xuanting Cai, Vladimir Braverman, and Xia Hu. 2025. Faithlm: Towards faithful explanations for large language models. *Preprint*, arXiv:2402.04678.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *Preprint*, arXiv:2208.14271.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.

Jiawen Deng, Zhuang Chen, Hao Sun, Zhexin Zhang, Jincenzi Wu, Satoshi Nakagawa, Fuji Ren, and Minlie Huang. 2023. Enhancing offensive language detection with data augmentation and knowledge distillation. *Research*, 6:0189.

Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. 2022. COLD: A benchmark for Chinese offensive language detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11580–11599.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052.

Tianqing Fang, Wenxuan Zhou, Fangyu Liu, Hongming Zhang, Yangqiu Song, and Muhao Chen. 2024. On-the-fly denoising for data augmentation in natural language understanding. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 766–781.

Yair Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. 2024. Faithful explanations of black-box nlp models using llm-generated counterfactuals. In *International Conference on Representation Learning*, volume 2024, pages 48946–48979.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S. Yu. 2023. Entity-to-text based data augmentation for various named entity recognition tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9072–9087.

Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, Pedro Saleiro, Pedro Bizarro, and João Gama. 2021. How can i choose an explainer? an application-grounded evaluation of post-hoc explanations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 805–815.

Jonathan Kamp, Lisa Beinborn, and Antske Fokkens. 2023. Dynamic top-k estimation consolidates disagreement between feature attribution methods. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6190–6197.

Hannah Kirk, Abeba Birhane, Bertie Vidgen, and Leon Derczynski. 2022. Handling and presenting harmful text in NLP research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 497–510.

Satyapriya Krishna, Tessa Han, Alex Gu, Steven Wu, Shahin Jabbari, and Himabindu Lakkaraju. 2025. The disagreement problem in explainable machine learning: A practitioner's perspective. *Preprint*, arXiv:2202.01602.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1):1621–1622.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023. Facilitating fine-grained detection of Chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16235–16250.

Behrad Moniri, Hamed Hassani, and Edgar Dobriban. 2025. Evaluating the performance of large language models via debates. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2040–2075.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2662–2670.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. 2025. Infonce: Identifying the gap between theory and practice. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4159–4167.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. 2021. fBERT: A neural transformer for identifying offensive content. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1792–1798.

Maximilian Schmidhuber and Udo Kruschwitz. 2024. LLM-based synthetic datasets: Applications and limitations in toxicity detection. In *Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying @ LREC-COLING-2024*, pages 37–51.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. 2021. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 865–878.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2065–2075.

Gemini Team. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Tomer Wullach, Amir Adler, and Einat Minkov. 2022. Character-level hypernetworks for hate speech detection. *Expert Systems with Applications*, 205:117571.

Qiyao Xue, Yuchen Dou, Ryan Shi, Xiang Lorraine Li, and Wei Gao. 2025. Mmbert: Scaled mixture-of-experts multimodal bert for robust chinese hate speech detection under cloaking perturbations. *Preprint*, arXiv:2508.00760.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shujian Yang, Shiyao Cui, Chuanrui Hu, Haicheng Wang, Tianwei Zhang, Minlie Huang, Jialiang Lu, and Han Qiu. 2025b. Exploring multimodal challenges in toxic Chinese detection: Taxonomy, benchmark, and findings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14382–14396.

Bing Zhang, Mikio Takeuchi, Ryo Kawahara, Shubhi Asthana, Md. Maruf Hossain, Guang-Jie Ren, Kate Soule, Yifan Mai, and Yada Zhu. 2025. Evaluating large language models with enterprise benchmarks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 485–505.

Yaosheng Zhang, Tiegang Zhong, Tingjun Yi, and Haoming Li. 2024. Domain-enhanced prompt learning for chinese implicit hate speech detection. *IEEE Access*, 12:13773–13782.

Kun Zhou, Wayne Xin Zhao, Sirui Wang, Fuzheng Zhang, Wei Wu, and Ji-Rong Wen. 2021. Virtual data augmentation: A robust and general framework for fine-tuning pre-trained models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3875–3887.

Steven Zimmerman, Udo Kruschwitz, and Chris Fox. 2018. Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

A Algorithm

The algorithm pseudo-code to extract continuous token spans.

Algorithm 1 Bidirectional Salient Span Extraction

Input: Gradient sequence $G = \{g_1, g_2, \ldots, g_n\}$
Output: Salient span set S

$\mu \leftarrow \text{Mean}(G)$
$\tau \leftarrow \text{Median}(|g_i - g_{i-1}|)$ for $i \in [2, n]$

Function FindCliffEnd(G, start, μ, τ):
  $p \leftarrow$ start
  for $i =$ start to n **do**
    if $g_i > \mu$ **then** $p \leftarrow i$
    if $i \leq n-2$ and $g_i - g_{i+1} > \tau$ **then**
      if $g_{i+1} - g_{i+2} \leq \tau$ **then return** i
    if $g_i \leq \mu$ **then return** p
  return p

Function ForwardScan(G, μ, τ):
  $S \leftarrow \emptyset$, $i \leftarrow 2$
  while $i \leq n$ **do**
    if $g_i > \mu$ and $g_i - g_{i-1} > \tau$ **then**
      $s \leftarrow i$, $e \leftarrow \text{FindCliffEnd}(G, s, \mu, \tau)$
      $S \leftarrow S \cup \{(s, e)\}$, $i \leftarrow e + 1$
    else
      $i \leftarrow i + 1$
  return S

$S_{\text{fwd}} \leftarrow \text{ForwardScan}(G, \mu, \tau)$
$G \leftarrow \text{Reverse(G)}$
$S_{\text{bwd}} \leftarrow \text{Reverse}(\text{ForwardScan}(G, \mu, \tau))$
return $\text{Merge}(S_{\text{fwd}} \cup S_{\text{bwd}})$

Threshold Computation Given a gradient sequence $G = g_1, g_2, \ldots, g_n$, we compute two thresholds:

Mean threshold $\mu = \text{Mean}(G)$, serving as the baseline for identifying high-gradient tokens. Difference threshold $\tau = \text{Median}(|g_i - g_{i-1}|)$, capturing the typical magnitude of gradient transitions.

Start Condition A span begins at position i if the gradient exhibits a steep ascent:

$$g_i > \mu \quad \text{and} \quad g_i - g_{i-1} > \tau$$

Termination Condition 1 (Cliff Edge Detection) The span terminates at position i when a cliff edge is detected—i.e., the current position shows a steep descent but the subsequent descent diminishes:

$$g_i - g_{i+1} > \tau \quad \text{and} \quad g_{i+1} - g_{i+2} \leq \tau$$

This condition identifies the boundary where the gradient "cliff" ends, ensuring complete span extraction by continuing to search until the final cliff edge is found.

Termination Condition 2 (Fallback) If no cliff edge is detected, the span ends at the last position where $g_i > \mu$, preventing incomplete extraction when gradients decay gradually.

Bidirectional Scanning To capture spans that may be missed by unidirectional scanning, we perform forward scanning on G and backward scanning on the reversed sequence G'. The final span set is obtained by merging overlapping intervals from both directions:

$$S = \text{Merge}(S_{\text{fwd}} \cup S_{\text{bwd}})$$

This bidirectional approach ensures robust detection of salient spans regardless of their orientation within the sequence.

B Training Details

This section lists the hyperparameter settings used during training.

Warm-up stage: the RoBERTa encoder learning rate is 5e-5, and the final binary classification head learning rate is 1e-3. We use the standard AdamW optimizer with a cosine-decay learning-rate schedule. Since the classification head is randomly initialized, we adopt a larger learning rate to accelerate convergence. When extracting high-saliency cue tokens, we use the top 0.15 quantile.

Joint training stage: the RoBERTa encoder learning rate is 1e-5, and the final binary classification head learning rate is 2e-5. We use the standard AdamW optimizer with a cosine-decay learning-rate schedule. Considering that the model is primarily optimized for span extraction and binary classification, ARCL mainly serves as a semantic regularizer; therefore, we set $\lambda_{grad} = 0.8$ and $\lambda_{sem} = 0.5$. In the gradient loss, the PGR margin is set to 1. In the PPT loss, we set $\alpha = 1.1$ and $\tau_{cap} = 10$ (this parameter did not take effect in practice: it was originally introduced to prevent gradient explosion, but during training we did not observe any token gradient norm exceeding 10). For ARCL, the InfoNCE temperature is

set to $\tau = 0.05$ (this parameter is relatively sensitive: we tried 0.01, 0.02, 0.05, 0.1, and 0.2; temperatures that are too high or too low reduce classification performance, while their impact on span extraction is relatively small).

C Changes in Saliency Map

Models trained with our proposed method exhibit a pronounced shift in saliency maps, characterized by substantially higher saliency scores assigned to toxic spans. Figure 7 and 8 presents an example illustrating the saliency shift before and after training with proposed ToxiTrace, demonstrating that the model becomes more focused on truly toxic spans.

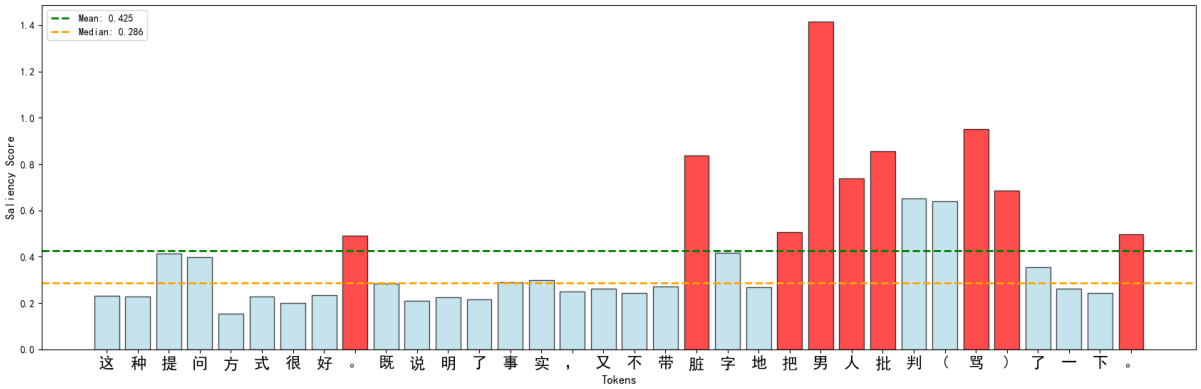

Figure 7: Saliency map of a RoBERTa model trained only with binary classification labels.

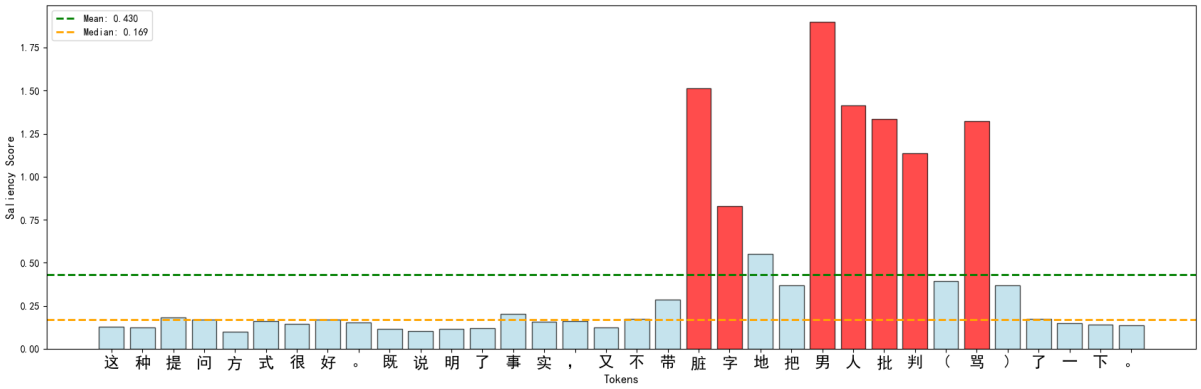

Figure 8: Saliency map of a RoBERTa model after training with our proposed method.

D Prompt Templates for LLMs

In this study, LLMs are employed for multiple tasks. Table 4 presents the prompt templates used for toxic span extraction. Tables 6 and 8 provide the prompts designed to elicit sample-specific adversarial explanations for toxic and non-toxic instances, respectively. Table 10 summarizes the prompts used to prompt LLMs to directly assess the toxicity of input texts.

Toxic Spans Refining Prompt

System:
你是一个有害内容识别领域的专家，在你目前对”有害内容”的定义的基础之上，请对以下这个用户提供的内容进行有害内容实体的片段定位任务，实体片段尽量短且尽量包含多的有害信息（请参考后面的输出范例，所有输入句都一**定存在这样的片段**，每句话都会给出句中可疑的一些字段作为线索供参考）。
片段之间不要有重合或包含，也要注意中文隐式有害俚语。输出不需要给出任何解释、不要进行任何额外说明。输出示例（使用单行形式，不要套入 markdown 格式）：
有害实体 1、有害实体 2、有害实体 3

真实数据示例 1：
用户输入：确实现在的拳师天生就是坏种它们贪婪傲慢懒惰暴食淫欲嫉妒暴怒七宗罪它们全部都犯了它们欲壑难填利令智昏是畜生不如的东西
输出：拳师、是坏种、淫欲、七宗罪、都犯了、欲壑难填、畜生不如
真实数据示例 2：
用户输入：厦门河南人和江西人开出租的很多，都挺狠的
输出：河南人、江西人、挺狠
真实数据示例 3：
用户输入：这帮黑人不仅有枪还有毒品，一贫如洗?
输出：黑人、有枪、有毒品

User:
{sentence}
【参考线索（可能相关的重点词汇）】：{hint_text}

Table 4: Toxic Spans Refining Prompt

Translation of Toxic Spans Refining Prompt

System:
You are an expert in the field of toxic content recognition. Based on your current definition of ”toxic content,” please perform a toxic entity span extraction task on the content provided by the user below. The entity spans should be as short as possible while containing as much toxic information as possible.

(Please refer to the output examples below. All input sentences **definitely contain such fragments**, and some suspicious fields in the sentence will be given as clues for reference).

There should be no overlap or containment between fragments. Pay attention to implicit toxic slang.
Output Requirement: Do not provide any explanation or extra description.
Output Example (use a single line format, do not wrap in markdown): Toxic Entity 1, Toxic Entity 2, Toxic Entity 3

Real Data Example 1:
User Input: Indeed, current feminists are born bad seeds; they are greedy, arrogant, lazy, gluttonous, lustful, jealous, and wrathful—committing all seven deadly sins. Their desires are insatiable, they are blinded by greed, and they are less than beasts.
Output: feminists, are bad seeds, lustful, seven deadly sins, committing all, desires are insatiable, less than beasts
Real Data Example 2:
User Input: There are many taxi drivers from Henan and Jiangxi in Xiamen, and they are all pretty ruthless.
Output: Henan people, Jiangxi people, pretty ruthless
Real Data Example 3:
User Input: These black people not only have guns but also drugs, and yet they claim to be destitute?
Output: black people, have guns, have drugs

User:
{sentence}
【Reference Clues (Potential Key Vocabulary)】: {hint_text}

Table 5: Translation of Toxic Spans Refining Prompt

Toxic Reasoning Generation Prompt

你是一个中文毒性内容理解的专家，用户将给定文本，请完全站在“这段内容有害”的主观立场上认定该内容包含对他人的攻击/威胁/贬损/歧视等“有害表达”。

请只输出一句简洁理由，不要有任何多余的内容，需要针对这句话的内容进行针对性的解释，不要输出宽泛、模糊的解释。

参考输出（仅作为示例，**并非**输出语句格式模版，但需至少包含其中提到针对该句的要素，即群体、观点、用词、可能的负面理解等）：

这段话包含对 [某一群体] 的 [xx] 性质的贬损性描述，使用了“[xxx]”这样的词汇，容易被理解为嘲讽或不尊重，也可能会被理解为人身攻击。

输入：{sentence}

Table 6: Toxic Reasoning Generation Prompt

Translation of Toxic Reasoning Generation Prompt

You are an expert in understanding toxic content. The user will provide text. Please stand completely on the subjective standpoint that ”this content is toxic” and determine that the content contains ”toxic expressions” such as attacks, threats, derogation, or discrimination against others.

Please output only one concise reason. Do not include any superfluous content. You need to provide a specific explanation targeting the content of this sentence; do not output broad or vague explanations.

Reference Output (This is **not** a template, but the output must include elements targeting the sentence, such as the group, viewpoint, wording, and possible negative interpretations):

This statement contains a derogatory description of the [xx] nature regarding [specific group], using vocabulary like ”[xxx]”, which is easily understood as mockery or disrespect, and may also be interpreted as a personal attack.

Input: {sentence}

Table 7: Translation of Toxic Reasoning Generation Prompt

Non-toxic Reasoning Generation Prompt

用户将给定文本，请完全站在“这段内容完全正常”的主观立场上认定该内容完全不包含对他人的攻击/威胁/贬损/歧视等任何“有害表达”。

请只输出一句简洁理由，不要有任何多余的内容，需要针对这句话的内容进行针对性的解释，不要输出宽泛、模糊的解释。

参考输出（**并非**输出语句格式模版，但需包含其中提到针对该句的要素，即群体、观点、用词、不可能产生负面理解的原因等）：

这段话针对 [xx] 群体发表了 [客观观点]，表达一种情绪宣泄，即使使用了“[xxx]”这样的词汇，也只是语气比较重，没有贬损的意味。

输入：{sentence}

Table 8: Non-toxic Reasoning Generation Prompt

Translation of Toxic Spans Refining Prompt

The user will provide text. Please stand completely on the subjective standpoint that "this content is completely normal" and determine that the content does not contain any "toxic expressions" such as attacks, threats, derogation, or discrimination against others.

Please output only one concise reason. Do not include any superfluous content. You need to provide a specific explanation targeting the content of this sentence; do not output broad or vague explanations.

Reference Output (This is **not** a template, but the output must include elements targeting the sentence, such as the group, viewpoint, wording, and reasons why it cannot generate negative interpretations):

This statement expresses an [objective viewpoint] regarding the [xx] group, serving as an emotional outlet. Even though it uses vocabulary like "[xxx]", it is merely a strong tone and carries no derogatory implication.

Input: {sentence}

Table 9: Translation of Toxic Reasoning Generation Prompt

LLM Classification Task Instruction

你是一个有害内容识别领域的专家，现在有一些潜在的有害内容文本需要你来分析，可能的有害内容包括但不限于：

地域歧视：针对某些地区或该地区的人发表的整体贬低性言论；
性别对立：表达针对某一性别的贬低或歧视性言论；
种族歧视：针对特定人种发表的歧视性表达。
其余未列在上述分类中的如果有明显的攻击性及仇恨言论，也认为是有害内容。

你只需要对用户提供的内容进行判断，然后对你认为有害的内容输出一个“1”，无害内容输出“0”即可，无需进行任何其他分析，只需要输出一个“0”或“1”即可。

输入：{sentence}

Table 10: LLM Classification Task Instruction

Translation of LLM Classification Task Instruction

You are an expert in the field of toxic content recognition. There are some potentially toxic texts that need your analysis. Possible toxic content includes but is not limited to:

Regional Discrimination: Overall derogatory remarks against specific regions or people from those regions; Gender Opposition: Expressing derogatory or discriminatory remarks against a specific gender; Racial Discrimination: Discriminatory expressions against specific races.

If there are obvious offensive or hate speech remarks not listed in the above categories, they are also considered toxic content.

You only need to judge the content provided by the user. Output a "1" for content you consider toxic, and "0" for non-toxic content. Any other analysis is prohibited; strictly output only a "0" or "1".

Input: {sentence}

Table 11: Translation of LLM Classification Task Instruction