Improving Hate Speech Detection by Fusing Textual and User Interaction Representations in Online Communities

Xu Gao[1,2,4], **Dong Jing**[1,3†], **Kee-hung Lai**[3]

[1]Soochow University, [2]Harbin Institute of Technology,
[3]Hong Kong Polytechnic University, [4]Zano Inc.

xugao_hit@stu.hit.edu.cn, djing@suda.edu.cn, mike.lai@polyu.edu.hk

Abstract

Detecting hate speech in online communities is increasingly challenging due to the implicit and context-dependent nature of toxic expressions. While text-only models often struggle with such ambiguity, incorporating user interaction signals offers critical pragmatic context for disambiguation. However, research in this direction is hindered by the scarcity of datasets that align textual content with comprehensive user behavioral graphs. To bridge this gap, we present a new dataset collected from a real-world community, featuring labeled hate speech enriched with fine-grained interaction histories. We further propose a novel user-aware hate speech detection framework that effectively fuses textual semantics with social interaction representations. Experiments demonstrate that our approach consistently outperforms strong text-only baselines by over 3.6%, validating the critical role of social context in enhancing detection accuracy. Furthermore, to mitigate real-world adversarial risks such as graph spoofing and spam, we introduce a contrastive graph augmentation strategy, ensuring model robustness against unreliable community behaviors.

1 Introduction

Effective hate speech detection is fundamental to the sustainability of online communities, as toxic content not only stifles user engagement but also inflicts significant psychological harm on individuals (Spence et al., 2023; Jiménez Durán et al., 2024). Despite its importance, identifying such content remains a formidable challenge due to the implicit and context-dependent nature of hate speech. Malicious users frequently employ lexical obfuscations, such as homophones and intentional misspellings, to disguise toxic content, leaving text-only models unable to identify the true semantic intent obscured by surface-level linguistic variations.

To overcome the limitations of pure textual analysis, incorporating user interaction representations provides essential pragmatic context. Prior research suggests that abusive users often exhibit distinct community behavioral patterns (Ribeiro et al., 2018), and that user embeddings can significantly enhance moderation efficacy by supplying missing social signals (Jing et al., 2025). However, progress in this direction is currently hindered by the scarcity of datasets that align textual content with rich social interactions. To bridge this gap, we present a new dataset collected in collaboration with community administrators, which uniquely integrates labeled hate speech with detailed user attributes and diverse interaction types, including likes, dislikes, and comments.

To effectively exploit these rich social signals, we propose **FUSH**[1] (Fusing User Signals for Hate speech detection). This framework features a dual-branch architecture that simultaneously models textual semantics and heterogeneous user interaction relationships. By fusing these complementary representations, our approach effectively disambiguates content intent. Experiments demonstrate that FUSH consistently outperforms strong text-only baselines, validating the critical role of social context in hate speech detection.

However, relying on user interactions exposes the system to adversarial vulnerabilities. Jing et al. (2025) highlight that graph-based methods are particularly susceptible to graph spoofing, where malicious actors actively distort social network structures through spamming or farming fake likes. Such manipulations compromise user embeddings and degrade performance. To mitigate this, we introduce a contrastive graph augmentation strategy that compels the model to learn invariant representations despite structural perturbations, ensuring

†Corresponding author.

[1]The code and dataset are publicly available at https://github.com/hyacinthxu99/FUSH-Fusing-User-Signals-for-Hate-speech-detection.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 76–87
July 2-7, 2026 ©2026 Association for Computational Linguistics

robustness in adversarial environments.

Our contributions are summarized as follows:

- To the best of our knowledge, we release the first hate speech detection dataset that incorporates both detailed user attributes and heterogeneous interaction behaviors, filling a critical resource gap.
- We propose FUSH, a unified user-aware detection framework that fuses textual semantics with social interaction representations. By modeling heterogeneous interaction dynamics, FUSH effectively leverages pragmatic context to disambiguate toxic intent and consistently outperforms strong text-only baselines.
- We introduce a contrastive graph augmentation strategy tailored to realistic adversarial scenarios such as graph spoofing. This strategy encourages invariant user representations under structural perturbations, improving robustness against manipulative behaviors including spam comments and artificially inflated likes.

2 Related Work

Hate speech detection has traditionally been treated as a text classification task, evolving from CNNs to pre-trained models like BERT and recent LLMs (Yuan et al., 2023; Wadud et al., 2023; Shi et al., 2023; Kolla et al., 2024). While LLMs demonstrate strong semantic understanding, recent findings suggest that smaller models like RoBERTa can achieve comparable performance at a fraction of the computational cost (Yu et al., 2023), offering a more viable solution for industrial-scale deployment. However, these text-centric approaches often struggle with implicit toxicity due to the lack of pragmatic context.

To address this, researchers have begun incorporating user-centric signals, such as follower networks (Mishra et al., 2018) and historical behavioral patterns (Rehman et al., 2023). The most relevant prior work, GEA (Jing et al., 2025), advances this by using Node2Vec to derive unsupervised user embeddings from interaction graphs. However, GEA faces two critical limitations for real-world application: (1) it cannot model heterogeneous interactions (e.g., distinguishing between likes and dislikes), and (2) it lacks defense mechanisms against graph spoofing, making it vulnerable to adversarial manipulation. Our work addresses these specific gaps by fusing heterogeneous interaction semantics and introducing a robust graph augmentation strategy.

3 Proposed Method

3.1 Problem Definition.

Let $\mathcal{U}$ denote the set of users and $\mathcal{T}$ the set of text posts. Each post $x_i \in \mathcal{T}$ is associated with a sender u_s and a receiver u_r, and labeled with $y_i \in \{0, 1\}$ indicating whether it contains hate speech. The objective is to learn a model

$$f : \mathcal{T} \times \mathcal{U}_s \times \mathcal{U}_r \rightarrow \{0, 1\}$$

that accurately predicts y_i by leveraging both the textual content of x_i and the user interaction representations of the sender u_s and receiver u_r.

3.2 Overview Architecture

The core idea of FUSH is to complement textual features with social context. It comprises two main components: a text feature module for semantic extraction, and a user embedding module for social interaction modeling. Notably, the user module constructs representations by integrating individual attributes with interpersonal interactions, thereby capturing both the user's intrinsic characteristics and the social influence from others. Finally, these features are then fused to predict toxic intent.

Text Feature Module. The text feature module is designed to generate textual representations, with RoBERTa (Liu et al., 2019) adopted as the backbone encoder. Prior work (Yu et al., 2023) has shown that although RoBERTa has significantly fewer parameters than recent LLMs, it still achieves comparable and sometimes even better performance on text classification tasks.

Given a text sequence T, we first tokenize it and obtain the input embeddings $\mathbf{X}_T \in \mathbb{R}^{n \times d_h}$, where n denotes the sequence length and d_h is the hidden size. RoBERTa encodes the input using multiple layers of self-attention and produces contextualized representations for all tokens:

$$\mathbf{H}_T = \text{RoBERTa}(\mathbf{X}_T) \in \mathbb{R}^{n \times d_t} \tag{1}$$

Following standard practice, we extract the representation of the initial [CLS] token to serve as the sentence-level embedding:

$$\mathbf{z}_{\text{text}} = \mathbf{H}_T[0] \in \mathbb{R}^{d_t} \tag{2}$$

This sentence-level representation $\mathbf{z}_{\text{text}}$ is then fused with user interaction features for hate speech detection.

Attributes of User B	
Attribute	**Value**
Active Days	5
Likes Received	3
Likes Given	2
Dislikes Received	9
Dislike Given	2
Comments Received	5
Comments Posted	7
Reports Received	4
Reports Filed	1

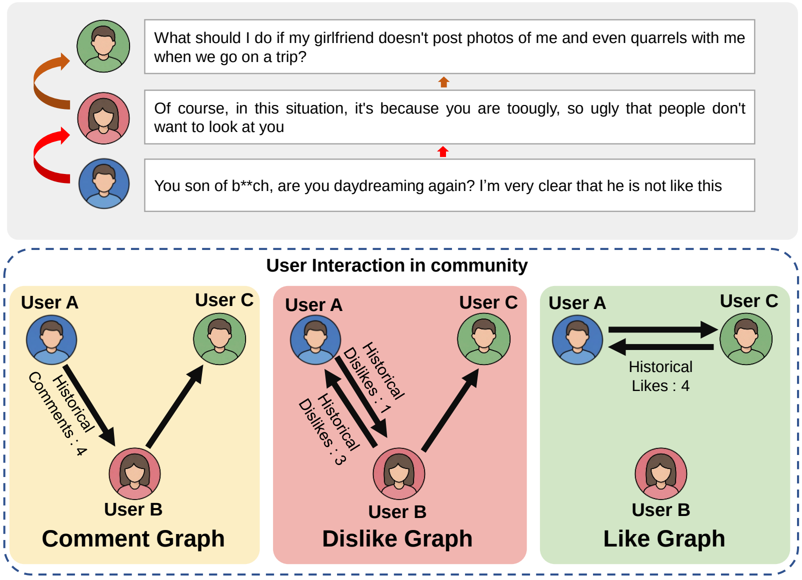

Figure 1: An illustrative example from our dataset. It includes user-generated texts, three types of interactions (likes, dislikes, comments), and user attributes. In this case, User A and B have a history of mutual dislikes, suggesting potential conflict, while A and C have exchanged likes, indicating a friendly tie. After B posted a hateful comment toward C, A responded with a hateful comment toward B. Historical interactions provide important context for understanding hate speech behavior.

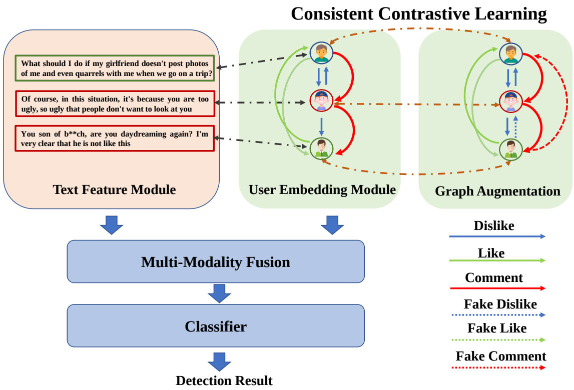

Figure 2: An overview of the FUSH.

User Embedding Module. The user embedding module is designed to extract effective user representations by integrating user attributes and user interaction behaviors. We first construct a directed heterogeneous interaction graph $G = (V, E, \mathcal{R})$, where each node $v \in V$ corresponds to a user and is associated with an attribute feature vector encoding user-specific statistics, such as activity levels and engagement history in the community. Each edge $e \in E$ represents a user interaction (e.g., like, dislike, or comment) and is associated with a relation type $\phi(e) \in \mathcal{R}$.

We then adopt a Heterogeneous Graph Transformer (Hu et al., 2020) to compute user representations from the interaction graph G. Compared to conventional heterogeneous graph neural networks, HGT employs a relation-dependent attention mechanism that automatically learns the semantic importance of diverse interaction types (e.g., distinguishing the varying impact of a "dislike" versus a "like") without relying on manually crafted meta-paths. Given a target user node v and the heterogeneous graph G, HGT aggregates relation-specific neighborhood information across multiple interaction types and produces a user embedding $\mathbf{z}_v^{\text{user}} \in \mathbb{R}^{d_u}$ that captures both topological structure and behavioral heterogeneity within the community.

Multi-Modality Fusion. To effectively fuse textual representation and user interaction representation for hate speech detection, we use a bilinear fusion mechanism. Specifically, given the textual representation $\mathbf{z}_{\text{text}} \in \mathbb{R}^{d_t}$, the embedding of the commenter (sender) $\mathbf{z}_s^{user} \in \mathbb{R}^{d_u}$, and the embedding of the replied user (receiver) $\mathbf{z}_r^{user} \in \mathbb{R}^{d_u}$, we perform fusion via a bilinear interaction:

$$\mathbf{z}_{\text{fused}} = \mathbf{W}_b(\mathbf{z}_{\text{text}} \otimes [\mathbf{z}_s^{user} \parallel \mathbf{z}_r^{user}]) + \mathbf{b} \qquad (3)$$

Here, $[\cdot \parallel \cdot]$ denotes vector concatenation, and $\otimes$ represents the Kronecker product, which explicitly models all pairwise interactions between textual and user features without explicitly constructing the high-dimensional outer product. $\mathbf{W}_b \in$

$\mathbb{R}^{d_{out}\times(d_t\cdot 2d_u)}$ serves as the learnable projection matrix, and $\mathbf{b}\in\mathbb{R}^{d_{out}}$ is the bias vector. The fused representation is subsequently fed into a linear classification layer.

To jointly optimize the detection task and improve the robustness of user representations, we adopt a hybrid loss function. Specifically, we combine the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ with a contrastive loss $\mathcal{L}^{\text{CL}}_{\text{total}}$, which encourages consistency of user embeddings under graph perturbations. The training objective is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}^{\text{CL}}_{\text{total}} \tag{4}$$

where λ is a hyperparameter balancing the two losses. The formulation and motivation of the contrastive loss will be detailed in the Section 3.3.

3.3 Contrastive Graph Augmentation

To improve the practical applicability of our model and enhance its robustness against graph spoofing scenarios, we propose a contrastive graph augmentation strategy.

Concretely, for each interaction type $r\in\mathcal{R}$ (e.g., Like, Comment, Dislike), we simulate real-world attacks such as spamming or fake likes. We construct a perturbed graph $\widetilde{G}$ by targeting a random subset of user nodes and adding fake edges or increasing interaction weights. Let G denote the original heterogeneous graph and $\widetilde{G}$ the perturbed version. We then compute user embeddings from both graphs, denoted as:

$$\mathbf{z}_u = \text{HGT}(G,u), \widetilde{\mathbf{z}}_u = \text{HGT}(\widetilde{G},u) \tag{5}$$

To encourage consistency between the original and perturbed representations of the same user, we adopt a contrastive loss. For a given user node u, the embedding $\mathbf{z}_u$ and its augmented counterpart $\widetilde{\mathbf{z}}_u$ form a positive pair, while embeddings of other users $\{\mathbf{z}_v \mid v\neq u\}$ serve as negative samples. We use the InfoNCE loss:

$$s_{\text{pos}} = f(\mathbf{z}_u,\widetilde{\mathbf{z}}_u), \quad s_{\text{neg}} = \sum_{v\neq u} f(\mathbf{z}_u,\widetilde{\mathbf{z}}_v) \tag{6}$$

$$\mathcal{L}_{\text{CL}}(u) = -\log\left(\frac{s_{\text{pos}}}{s_{\text{pos}}+s_{\text{neg}}}\right) \tag{7}$$

Here, $f(\mathbf{x},\mathbf{y}) = \exp(\text{sim}(\mathbf{x},\mathbf{y})/\tau)$ represents the exponentiated similarity score, where $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity and τ is a temperature hyperparameter. The total contrastive loss is calculated by averaging over all user nodes:

$$\mathcal{L}^{\text{CL}}_{\text{total}} = \frac{1}{|V|}\sum_{u\in V}\mathcal{L}_{\text{CL}}(u) \tag{8}$$

This contrastive objective is jointly optimized with the classification loss to guide the model in learning robust user representations that are invariant to perturbations in the interaction graph.

4 Data Sets

Data Sources. The dataset used and released in this study is sourced from a major online community platform that primarily serves Chinese users, with a focus on youth population. Upon acceptance of the paper, we will release detailed information about the platform. Similar in nature to platforms such as Reddit and Quora, this platform facilitates a wide range of discussions and interactions.

Data Collection. We extracted behavioral statistics and interaction logs for 132,755 users from the platform database between May 1, 2024, and May 1, 2025, ensuring strict anonymization of all user identifiers. From a randomly retained subset of the raw logs, we manually annotated 11,825 instances, resulting in a dataset with an approximate 1:2 ratio of hate speech to non-hate speech. Further annotation details are provided in Appendix A.

Data Composition. Each instance in the dataset includes the replied content, the comment, the anonymized user IDs of both the original author and the commenter, and a binary label indicating whether the comment contains hate speech. Beyond the textual data, the dataset also incorporates user statistical attributes and three user interaction graphs as visualized in Fig. 1. We construct three interaction graphs (like, comment, and dislike), where each directed edge denotes an interaction from a sender to a receiver and is weighted by the corresponding interaction frequency.

5 Experiment

We conduct a series of experiments using F1-score and Accuracy as the primary evaluation metrics under deployment-oriented evaluation settings. It is important to note that standard hate speech benchmarks, such as HateXplain (Mathew et al., 2021) and Davidson et al. (2017), are limited to isolated textual content and lack the user interaction graphs essential for our FUSH framework. Consequently, we validate our method exclusively on the dataset introduced in this work, as it is the unique resource providing the necessary social interaction signals. We provide details of the experimental environment and hyperparameters in the Appendix C.

5.1 RQ1. Accuracy

To ensure a rigorous evaluation that mirrors real-world deployment, we adopt a time-decoupled experimental protocol. The dataset is partitioned chronologically, with the test set strictly succeeding the training period. During evaluation, the user attributes and interaction graph are constructed solely from training data to prevent temporal information leakage. Under this setting, we compare FUSH against three categories of baselines: (1) fine-tuned PLMs represented by RoBERTa; (2) LLMs including the fine-tuned Qwen-3 (Yang et al., 2025) (7B) and prompting-based ChatGPT (gpt-5.1, with prompts detailed in Appendix F) in both zero-shot and few-shot settings; (3) specialized architectures designed for implicit toxicity and obfuscation, represented by the method CoSyn (Ghosh et al., 2023); and (4) the user-aware model GEA. Since GEA acts on single graphs, we compare it with FUSH on each interaction type separately.

Method	F1 (%)	Acc (%)
Text-Only Baselines		
RoBERTa	89.05	92.33
Qwen-3 (7B)	89.37	92.30
CoSyn	88.89	91.83
ChatGPT (Zero-shot)	77.96	86.58
ChatGPT (Few-shot)	81.24	87.76
Comparison on Single Graph		
w/ Like Graph		
GEA	89.81	92.76
FUSH (Ours)	**90.33**	**93.04**
w/ Comment Graph		
GEA	89.94	92.84
FUSH (Ours)	**91.66**	**93.94**
w/ Dislike Graph		
GEA	90.19	93.07
FUSH (Ours)	**91.77**	**94.19**
Multi-Graph Fusion (Full Model)		
FUSH (Text + All)	**92.94**	**95.04**

Table 1: Performance comparison. FUSH consistently outperforms baselines across varying interaction settings.

Table 1 reports the average performance over 10 independent runs. Overall, FUSH achieves state-of-the-art performance across all metrics, recording an F1-score of 92.94% and an accuracy of 95.04%. It surpasses the strongest fine-tuned text baselines, RoBERTa and Qwen-3, by margins of 3.89% and 3.57% in F1-score, respectively. In contrast, ChatGPT performs poorly, lagging significantly behind all fine-tuned models. This disparity suggests that general-purpose prompting is insufficient for hate speech detection, a task characterized by complex pragmatics and subtle offensiveness, thereby confirming the necessity of domain-specific fine-tuning and user context integration.

When benchmarking against the user-aware baseline GEA, FUSH consistently yields higher F1-scores and accuracy across all individual interaction graphs. This superiority validates the effectiveness of our HGT module in capturing fine-grained user signals. Furthermore, analyzing the impact of different interaction types reveals that the semantic nature of social edges plays a critical role. Dislike-based graphs provide the most significant performance boost, likely because explicit negative feedback directly correlates with user antagonism. Comment interactions offer moderate gains by capturing pragmatic intent through direct replies, while like-based graphs contribute the least due to the inherent ambiguity of likes. These findings underscore the importance of fusing diverse and semantically distinct interactions to maximize detection accuracy.

5.2 RQ2. Case Study

To investigate how FUSH leverages user features to outperform text-only models, we analyze hate speech instances that are misclassified by RoBERTa but correctly identified by FUSH. For these cases, we present the user attributes and interactions of both the commenter and the recipient to illustrate the contribution of the user modality.

Fig. 3 presents a real case from the test set. User 56893 posted a hate speech toward user 39812. Based solely on the textual content, the comment is ambiguous in terms of hatefulness, and the RoBERTa model misclassified it as non-hateful. In contrast, FUSH correctly identified it as hate speech by incorporating user attributes and historical interaction context.

Specifically, in terms of user attributes, the receiver (user 39812) had posted fewer comments but received significantly more dislikes and reports, suggesting they were generally less favored in the community. The interaction history shows that the sender disliked the receiver's content three times, posted 17 comments, and liked only once, indicating a predominantly negative attitude. These

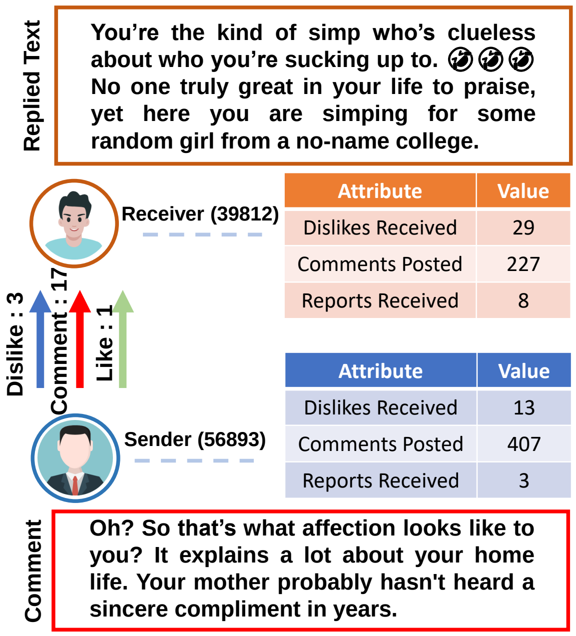

Figure 3: A case study of FUSH.

behavioral signals, combined with the receiver's community attributes, suggest that the sender was likely motivated by accumulated dissatisfaction when posting the comment. By capturing user attitudes and integrating them with textual features, FUSH outperforms the text-only model in accurately detecting hate speech.

5.3 RQ3. Robustness Analysis

To evaluate the resilience of our method, we conduct experiments under two challenging real-world conditions: graph spoofing and user cold-start scenarios. The results are reported in Table 2.

First, to evaluate robustness against graph spoofing, we simulate adversarial behaviors such as like farming, comment spamming, and dislike flooding. Reflecting real-world settings where manipulative activities are typically concentrated on a small subset of users rather than uniformly distributed, we inject adversarial noise by randomly selecting 10% of user nodes as attack targets. For each selected user, we intensify social noise by either doubling existing edge weights or injecting an equivalent number of spurious edges. Under this targeted attack, FUSH maintains strong performance with an F1-score of 92.83%. In contrast, removing the contrastive graph augmentation strategy leads to a notable degradation, with the F1-score dropping to 91.10%. These results indicate that contrastive alignment effectively improves robustness to localized structural perturbations, enabling stable detection under adversarial interaction patterns.

Second, we evaluate the performance of FUSH under cold-start scenarios. To simulate a surge of newly registered users, we randomly replace the sender IDs of 20% of test comments with pseudo-identities that have no historical interaction records or profile attributes, while retaining the original receiver IDs. Although such a high proportion of zero-profile users is uncommon in mature communities, we adopt this extreme setting as a rigorous stress test. In this setting, sender-side social signals are missing as new users have not yet generated sufficient interaction histories, and the model must rely on textual content and receiver-side context. While performance degrades as expected, FUSH still outperforms text-only baselines, since receiver representations remain accessible and can provide useful social cues for intent disambiguation. Further analyses of inference efficiency and ablation results are reported in Appendix D and Appendix E, respectively.

Scenarios	**F1 (%)**	**Acc (%)**
Scenario 1: Graph Spoofing Attack		
FUSH (Full Model)	**92.83**	**94.87**
w/o Graph Augmentation	91.10	93.94
Scenario 2: Cold Start		
FUSH (New Users)	90.53	93.12

Table 2: Performance evaluation under challenging scenarios, including robustness against graph spoofing attacks and effectiveness in user cold-start settings.

6 Conclusion

In this work, we address a critical resource gap by introducing the first hate speech detection dataset that incorporates both fine-grained user attributes and heterogeneous interaction behaviors. To exploit this multi-modal context, we propose FUSH, a unified framework designed to fuse textual semantics with social signals. By integrating a contrastive graph augmentation strategy, our approach not only captures the pragmatic context essential for accurate detection but also ensures resilience against adversarial graph spoofing. Extensive experiments under deployment-oriented evaluation protocols show that FUSH consistently outperforms strong baselines and remains stable under adversarial and cold-start scenarios, demonstrating the practical value of integrating user interaction signals into real-world hate speech detection systems.

Limitations

While our work demonstrates the effectiveness of fusing textual and social interaction signals, there are several limitations to consider. First, our dataset is sourced from a specific online community primarily serving university students. The demographic homogeneity (e.g., age, education level) and specific community norms may result in interaction patterns that differ from broader, open social media platforms like Twitter or Facebook. Consequently, the generalizability of our findings to other cultural contexts or platforms with different user dynamics requires further verification.

In addition, while our contrastive graph augmentation improves robustness against spoofing, our current approach models user interactions within static time windows. It does not fully capture the temporal evolution of user behaviors or the dynamic shift of community topics over long periods, which remains a direction for future work.

Acknowledgments

The authors would like to thank anonymous reviewers for their insightful comments and valuable suggestions. The Hong Kong Polytechnic University funds this research under project code P0045777.

Ethical considerations

Data Privacy and Compliance. We prioritize the privacy and safety of user data. The dataset used in this study was collected from an online community platform in strict adherence to its data usage policies and terms of service. Before analysis, all personally identifiable information (PII)—including user IDs, usernames, and specific profile details—was rigorously anonymized to prevent user re-identification. We emphasize that the user attributes and interaction graphs included in the dataset are used solely for the purpose of modeling community dynamics to detect hate speech and do not target specific individuals.

Annotator Well-being. Given the toxic nature of hate speech, we implemented a comprehensive protocol to safeguard the psychological well-being of our annotators. All annotators were informed of the potentially offensive nature of the content and provided written informed consent prior to participation. To minimize psychological impact, we strictly limited the daily evaluation volume and duration to prevent fatigue and desensitization. Annotators were empowered to pause or cease work immediately if they experienced discomfort, and regular check-ins were conducted to ensure a supportive working environment.

Intended Use and Disclaimer. This dataset and the proposed FUSH framework are intended exclusively for academic research to enhance content moderation systems. We explicitly condemn the use of this technology for user surveillance or profiling beyond the scope of safety maintenance. The opinions and offensive content contained in the dataset samples reflect the raw data from the platform and do not represent the views of the authors or our affiliations.

References

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.

Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4):1–30.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Sreyan Ghosh, Manan Suri, Purva Chiniya, Utkarsh Tyagi, Sonal Kumar, and Dinesh Manocha. 2023. CoSyn: Detecting implicit hate speech in online conversations using a context synergized hyperbolic network. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6159–6173, Singapore. Association for Computational Linguistics.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, WWW '20, page 2704–2710, New York, NY, USA. Association for Computing Machinery.

Rafael Jiménez Durán, Karsten Müller, and Carlo Schwarz. 2024. The effect of content moderation on online and offline hate: Evidence from germany's netzdg. *Available at SSRN 4230296*.

Dong Jing, Xu Gao, and Kee-Hung Lai. 2025. Text moderation in online communities: Integrating user attributes and interaction graph embedding. *Big Data Mining and Analytics*.

Mahi Kolla, Siddharth Salunkhe, Eshwar Chandrasekharan, and Koustuv Saha. 2024. Llm-mod: Can large language models assist content moderation? In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–8.

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*.

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.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th international conference on computational linguistics*, pages 1088–1098.

Mohammad Zia Ur Rehman, Somya Mehta, Kuldeep Singh, Kunal Kaushik, and Nagendra Kumar. 2023. User-aware multilingual abusive content detection in social media. *Information Processing & Management*, 60(5):103450.

Manoel Ribeiro, Pedro Calais, Yuri Santos, Virgílio Almeida, and Wagner Meira, Jr. 2018. Characterizing and detecting hateful users on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Xiaohou Shi, Jiahao Liu, and Yaqi Song. 2023. Bert and llm-based multivariate hate speech detection on twitter: comparative analysis and superior performance. In *International Artificial Intelligence Conference*, pages 85–97. Springer.

Ruth Spence, Antonia Bifulco, Paula Bradbury, Elena Martellozzo, and Jeffrey DeMarco. 2023. The psychological impacts of content moderation on content moderators: A qualitative study. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 17(4).

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

Md Anwar Hussen Wadud, Muhammad Firoz Mridha, Jungpil Shin, Kamruddin Nur, and Aloke Kumar Saha. 2023. Deep-bert: Transfer learning for classifying multilingual offensive texts on social media. *Computer Systems Science & Engineering*, 44(2).

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

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

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, closed, or small language models for text classification? *arXiv preprint arXiv:2308.10092*.

Lanqin Yuan, Tianyu Wang, Gabriela Ferraro, Hanna Suominen, and Marian-Andrei Rizoiu. 2023. Transfer learning for hate speech detection in social media. *Journal of Computational Social Science*, 6(2):1081–1101.

A Data Annotation

Following established literature (Davidson et al., 2017; Fortuna and Nunes, 2018), we define hate speech as language that expresses hatred, incites violence, or promotes discrimination against individuals or groups based on attributes such as race, gender, religion, or sexual identity. Our annotation guidelines are further informed by platform policies and prior hate speech detection datasets (Founta et al., 2018; Vidgen and Derczynski, 2020; Mathew et al., 2021; De Gibert et al., 2018). The annotation results yielded a Fleiss' Kappa of 0.8301 and a Krippendorff's Alpha of 0.8302, indicating a high level of inter-annotator agreement among the human annotators.

Annotators were instructed to focus on intent, target, and identity-based hostility, and to rely on contextual signals when the surface text alone was ambiguous. To ensure labeling consistency and accuracy, each sample was independently annotated by three trained annotators. Annotators were instructed to consider both textual content and contextual cues when making their decisions. For each sample, we adopted the majority vote strategy to determine the final label, thereby reducing individual bias and enhancing the reliability of the annotations. The data has been reviewed and approved by the ethics review board.

B Dataset Statistics

We partitioned the dataset into training and testing sets using a 7:3 ratio. To ensure no temporal overlap, the test set chronologically follows the training set. Specifically, the test set only utilizes historical user attributes and interactions observed during the training period, thereby preventing data leakage (i.e., look-ahead bias).

B.1 Text Description

To provide a comprehensive overview of the dataset, we present key statistics of the textual content in Table 3. The dataset contains 23,650 single comments, which are organized into 11,825 comment-reply pairs, reflecting that a portion of the replied content is shared across multiple samples. After removing punctuation and Chinese stopwords, the average length of a single comment is approximately 9.28 words, with lengths ranging from 1 to 278. The relatively high standard deviation of 18.28 indicates substantial variation in comment length, reflecting the natural heterogeneity of user-generated content in online communities. Additionally, the vocabulary size reaches 27,514, demonstrating a high degree of lexical diversity, which presents additional challenges for accurate and robust hate speech detection.

Table 3: Descriptive statistics of texts in dataset.

Statistic	Value
Total comments	23,650
Average word length	9.28
Minimum word length	1
Maximum word length	278
Standard deviation	18.28
Vocabulary size	27,514

B.2 User Attribute Description

We present the descriptive statistics of key user attributes in Table 4. The dataset exhibits wide variability in user engagement and interaction behaviors. For instance, the number of active days ranges from 0 to 1,936 with an average of approximately 963 days, indicating diverse participation lengths across users. Interaction metrics such as likes received and given show large disparities, with maximum values exceeding 30,000 but with average values around 15, highlighting a skewed distribution where a minority of users receive or give most interactions. Dislikes and reports, both received and filed, are relatively rare, as reflected by their low means and high zero ratios, indicating most users neither attract nor issue many negative feedbacks or reports. The zero ratio column further reveals that a substantial portion of users did not engage in certain interactions, such as reporting or disliking, underscoring the sparsity typical of user-generated interaction data. These statistics provide essential insights into the community dynamics and serve as valuable context for modeling user behavior in hate speech detection.

B.3 User Interaction Description

We present the statistical metrics of the three interaction graphs in Table 5. The like graph contains 29,457 nodes and over 1.26 million edges, exhibiting a relatively high average clustering coefficient (ACC) of 0.2275 and transitivity of 0.1522, indicating strong community structures and frequent mutual interactions among users who engage positively. The comment graph, with 28,545 nodes and approximately one million edges, has a lower

Table 4: Descriptive statistics of user attributes in dataset.

Attribute	min	max	mean	std	zero ratio
Active Days	0	1936	962.932492	543.998353	0.000241
Likes Received	0	32984	15.381288	199.084182	0.839222
Likes Given	0	31617	15.068818	201.765793	0.810169
Dislikes Received	0	3294	0.434966	12.061238	0.950201
Dislikes Given	0	1335	0.426040	6.713381	0.930044
Comments Received	0	13109	10.701193	85.753117	0.802659
Comments Posted	0	9512	10.938458	82.887392	0.795020
Reports Received	0	684	0.092960	2.578913	0.979661
Reports Filed	0	195	0.095514	1.656734	0.974833

Graph Type	Nodes	Edges	ACC	Transitivity
Like	29457	1261892	0.2275	0.1522
Comment	28545	1003691	0.0988	0.0808
Dislike	12187	44105	0.0269	0.0186

Table 5: **User interaction graph.**

ACC of 0.0988 and transitivity of 0.0808, suggesting a more dispersed structure where users are less tightly clustered, possibly due to the broader or more topic-driven nature of commenting behavior. In contrast, the dislike graph is much sparser, comprising only 12,187 nodes and 44,105 edges, with very low clustering (ACC of 0.0269) and transitivity (0.0186), reflecting that negative interactions are both rarer and less likely to form cohesive structures within the community. We have included the dataset in the supplementary materials to provide reviewers with additional information.

C Experimental Environment and Settings

Our experiments were conducted on a system equipped with an NVIDIA RTX 4090 GPU featuring 24 GB of VRAM. The computing environment utilized PyTorch version 2.3.0 with Python 3.12 running on Ubuntu 22.04. The CPU consists of 16 cores from an Intel Xeon Platinum 8352V.

Table 6 summarizes the hyperparameter settings used in our experiments. We use a learning rate of 1×10^{-5}, a batch size of 8, and train for 15 epochs. The maximum input sequence length is set to 256, and a dropout rate of 0.2 is applied to prevent overfitting. The dimensionality of the user embedding z^{user} is set to 256.

Table 6: Hyperparameter settings used in our experiments.

Hyperparameter	Value
Learning rate	1e-5
Batch size	8
Epochs	15
Max length	256
Dropout rate	0.2
z^{user}	256
d^t	1024
d^{out}	1536
λ	1
τ	0.5

D Inference Efficiency Analysis

While FUSH integrates a Heterogeneous Graph Transformer and a multi-modality fusion module alongside the RoBERTa backbone, its deployment cost remains highly competitive. It is acknowledged that incorporating the HGT module increases computational overhead and memory usage during the *training phase*. However, for real-world deployment, the inference process is optimized through a decoupled architecture.

Specifically, since user interaction patterns are relatively stable over short periods, the user embeddings generated by the HGT module can be precomputed and stored (e.g., in a high-performance Key-Value store). Consequently, during the inference phase, the costly graph propagation is eliminated. The system only needs to perform a low-latency retrieval of the cached user embeddings. Furthermore, the fusion module, designed as a lightweight bilinear projection, introduces negligible parameter overhead.

Our internal benchmarks indicate that compared to the standalone RoBERTa baseline, the end-to-end inference latency of FUSH increases by only 5% to 10%. Given the significant improvements in detection accuracy and robustness, this marginal increase in computational cost is highly acceptable for industrial hate speech detection systems.

E Ablation Analysis

To investigate the individual contributions of the proposed components, we conduct an ablation study focusing on two dimensions: the architecture of the user interaction encoder and the impact of the contrastive graph augmentation strategy. The results are summarized in Table 7.

Method Variant	F1-Score	Accuracy
FUSH **(Ours)**	**92.94**	**95.04**
Replacement of Graph Encoder		
w/ HAN	91.57	93.88
w/ RGCN	91.46	94.36
Removal of Training Strategy		
w/o Graph Augmentation	92.37	94.59

Table 7: Ablation study investigating the impact of different graph encoders and the contrastive augmentation strategy.

First, we evaluate the HGT backbone (Hu et al., 2020) employed by FUSH against established heterogeneous baselines, specifically HAN (Wang et al., 2019) and RGCN (Schlichtkrull et al., 2018), to assess general modeling capabilities. As observed, FUSH consistently outperforms all variants. This superiority is primarily attributed to the capability of HGT to incorporate interaction intensity via continuous edge weights. Unlike standard HAN and RGCN implementations that often focus solely on structural connectivity, HGT explicitly models the frequency of interactions to capture fine-grained behavioral patterns which are critical for characterizing the social dynamics of abusive users.

Second, we evaluate the robustness strategy by removing the contrastive graph augmentation. The results reveal a clear performance degradation when this strategy is omitted. This decline demonstrates that contrastive alignment is critical for learning invariant user representations. By enforcing consistency despite structural perturbations, the strategy effectively fortifies the model against social noise and potential graph spoofing, ensuring stable performance in realistic environments.

F Prompts of ChatGPT

We report the prompts used for ChatGPT in zero-shot and few-shot settings in Figure 4.

Zero-Shot

You are an expert Content Moderator for an online social platform. Your task is to detect uncivil language, toxicity, or hate speech in user comments.
Task Description: You will be presented with a text pair consisting of a "Replied Content" (context) and a "Reply" (comment).
Content wrapped in <CONTENT>...</CONTENT> is the original post or message being replied to.
Content wrapped in <REPLY>...</REPLY> is the user's comment that needs to be evaluated.
Evaluation Criteria: Determine if the text inside <REPLY> constitutes uncivil language, hate speech, harassment, or toxic behavior.
Label 1: The reply contains insults, aggression, hate speech, or severe incivility.
Label 0: The reply is safe, neutral, friendly, or merely purely disagreement without toxicity.
Instructions (Chain of Thought):
1. Analyze the Context: Read the <CONTENT> to understand the situation or topic.
2. Analyze the Intent: Read the <REPLY> to determine the user's tone and intent. Is it aggressive, mocking, or hateful? Does it attack the person rather than the topic?
Make a Decision: Based on the analysis, assign the correct label.
Output Format: Provide your reasoning briefly, separate the reasoning and the result with a semicolon (;), and then output the final numerical label (0 or 1). Do not add any text after the number.
Current Input: <CONTENT>{replied_content}</CONTENT> <REPLY>{content}</REPLY>
Your Response:

Few-Shot

You are an expert Content Moderator for an online social platform. Your task is to detect uncivil language, toxicity, or hate speech in user comments.
Task Description: You will be presented with a text pair consisting of a "Replied Content" (context) and a "Reply" (comment).
Content wrapped in <CONTENT>...</CONTENT> is the original post or message being replied to.
Content wrapped in <REPLY>...</REPLY> is the user's comment that needs to be evaluated.
Evaluation Criteria: Determine if the text inside <REPLY> constitutes uncivil language, hate speech, harassment, or toxic behavior.
Label 1: The reply contains insults, aggression, hate speech, or severe incivility.
Label 0: The reply is safe, neutral, friendly, or merely purely disagreement without toxicity.
Instructions (Chain of Thought):
1. Analyze the Context: Read the <CONTENT> to understand the situation or topic.
2. Analyze the Intent: Read the <REPLY> to determine the user's tone and intent. Is it aggressive, mocking, or hateful? Does it attack the person rather than the topic?
Make a Decision: Based on the analysis, assign the correct label.
Output Format: Provide your reasoning briefly, separate the reasoning and the result with a semicolon (;), and then output the final numerical label (0 or 1). Do not add any text after the number.
Few-Shot Examples:
Example 1: Input: <CONTENT>My girlfriend is unhappy. I bought her some snacks, but she rejected them and said she doesn't like them. What should I do? 🥺</CONTENT> <REPLY>You give her food and she gives you attitude? If you gave it to me, I'd call you daddy.</REPLY> Response: Reasoning: The reply uses aggressive language ("gives you attitude") and a bizarre, potentially mocking or inappropriate comparison ("I'd call you daddy"). While it might be interpreted as a joke in some circles, in a moderation context, it is dismissive and uses uncivil phrasing to criticize the girlfriend. It leans towards toxicity; 1
Example 2: Input: <CONTENT>Bro, the shirt in your first photo is unique. I think I saw you on the street a couple of days ago.</CONTENT> <REPLY>Thanks! I also think the design is really cool (· ∀ ·)</REPLY> Response: Reasoning: The reply expresses gratitude ("Thanks!") and agreement in a friendly tone with an emoticon. There is no toxicity; 0
Current Input: <CONTENT>{replied_content}</CONTENT> <REPLY>{content}</REPLY>
Your Response:

Figure 4: Prompt of ChatGPT (few-shot)