

Article

Fine-Grained Detection of Implicit Hate Speech in Chinese Based on Contrastive Learning and Retrieval-Augmented Adjudication

Xiao Han ¹, Kangbo Hu ^{2,*}, Chichen Lin ³, Yijie Jin ² and Huaijin Xie ²

¹ School of Information and Communication, Communication University of China, Beijing 100024, China; hanxiao@cuc.edu.cn

² School of Computer and Cyber Sciences, Communication University of China, Beijing 100024, China; 202312033012@mails.cuc.edu.cn (Y.J.); 202312063032@cuc.edu.cn (H.X.)

³ State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China; cuc_lcc@cuc.edu.cn

* Correspondence: hkbyyds@cuc.edu.cn

Featured Application

FICHS can support Chinese social media moderation systems by improving the detection of subtle, culturally contextualized, and implicitly hateful expressions.

Abstract

Implicit hate speech is difficult to detect because hostile intent is often conveyed through metaphor, irony, coded expressions, stereotypes, or culturally situated allusions rather than direct insults. We first construct a fine-grained Chinese implicit hate speech dataset containing 19,939 samples from Zhihu and Baidu Tieba, covering nine target categories and three expression labels. Based on this dataset, we propose FICHS, a fine-grained and implicit Chinese hate speech detection framework with three synergistic modules built on a RoBERTa-based base detector: an Explicit–Implicit Supervised Contrastive Module that learns a discriminative semantic space between explicit and implicit hate speech, a Semantic Opacity-Aware Routing and Adjudication module that selectively routes low-confidence but high-implicitness samples to an LLM for secondary judgment, and a Pattern-Guided Retrieval-Augmented Recovery Module that retrieves socio-cultural risk patterns to support second-round review and false-negative recovery. Experimental results demonstrate that FICHS effectively improves Chinese implicit hate speech detection, achieving a weighted F1 score of 0.8391, with precision of 0.8428 and recall of 0.8396, outperforming both the SOTA detector and standalone LLM baselines.

Keywords: Chinese hate speech detection; implicit hate speech; contrastive learning; retrieval-augmented generation



Academic Editor: Douglas O'Shaughnessy

Received: 10 June 2026

Revised: 18 July 2026

Accepted: 22 July 2026

Published: 27 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Online hate speech harms affected communities, distorts public discourse, and increases the burden of platform governance [1,2]. In Chinese social media, the problem is particularly challenging because hostile intent is often embedded in high-context discourse, local memes, homophones, stereotypes, irony, or topic-specific allusions. A sentence may avoid direct slurs while still conveying exclusion, contempt, or dehumanization toward a protected group. Such implicit hate speech is therefore harder to detect than explicit hate speech that contains direct insults, slurs, or violence-related language.

Existing hate speech and offensive language detection research has provided increasingly fine-grained datasets and neural models. English resources such as Latent Hatred and ToxiGen emphasize that implicit hate frequently requires pragmatic inference and background knowledge [3,4]. For Chinese, COLDataset, ToxiCN, and STATE ToxiCN have advanced offensive and toxic language detection by introducing real social media data, hierarchical labels, and span-level target-aware resources [5–7]. Recent work on Chinese phonetic cloaking further shows that offensive intent can be deliberately hidden through pronunciation-based replacement [8]. However, several limitations remain. First, target coverage in Chinese toxic language resources is still relatively narrow compared with the diversity of contemporary online conflicts. Second, conventional classifiers often rely on lexical or surface-level cues, which weakens their ability to capture semantically opaque hatred. Third, LLMs are promising adjudicators for subtle cases, and recent studies show that they can improve hate speech detection and explanation generation [9–11]. At the same time, LLMs can be overly sensitive or poorly calibrated in implicit hate speech detection [12]. This suggests that LLMs should not simply replace task-specific classifiers; instead, they should be used selectively where their reasoning ability is most useful.

To address these issues, we propose FICHS (fine-grained and implicit Chinese hate speech detection), a Chinese implicit hate speech detection framework based on contrastive learning and retrieval-augmented adjudication. FICHS is built around three ideas. First, we expand the original ToxiCN-style data space by adding new target groups and removing an ambiguous reporting-expression category, resulting in a 19,939-sample dataset with nine target categories. Second, we use supervised contrastive learning to construct an explicit-implicit semantic space in which explicit and implicit hate expressions become more separable. Third, we combine the base model with selective LLM adjudication and pattern-guided retrieval augmentation, so that high-risk implicit cases can be reviewed with additional socio-cultural cues.

The main contributions are as follows:

- We construct an expanded Chinese hate speech dataset containing 19,939 samples from Zhihu and Baidu Tieba, annotated with toxicity, hate intensity, expression type, and nine protected or socially salient target categories.
- We introduce an Explicit-Implicit Supervised Contrastive Module (EI-SCM) that uses expression labels to construct a semantic opacity representation and derive a continuous implicitness score for routing difficult cases.
- We propose Semantic Opacity-Aware Routing and Adjudication (SOAR), a confidence- and implicitness-aware mechanism that selectively invokes an LLM for samples likely to be missed by the base detector.
- We design a Pattern-Guided Retrieval-Augmented Recovery Module (PGRM), which uses training-derived high-frequency 2-gram and 3-gram risk patterns with human-written notes to support second-round false-negative recovery.

The remainder of this paper is organized as follows. Section 2 reviews related work on hate speech detection, implicit hate speech, contrastive learning, LLM-based adjudication, and retrieval-augmented reasoning. Section 3 presents the FICHS framework and describes its three main modules. Section 4 introduces the dataset, annotation scheme, and experimental settings. Section 5 presents the main experimental results and ablation studies. Section 6 discusses the implications and limitations of the proposed framework, and Section 7 concludes the paper.

2. Related Work

2.1. Hate Speech and Offensive Language Detection

Hate speech detection has developed from binary offensive language classification toward more fine-grained modeling of target groups, intent, severity, and implicitness. Early work demonstrated the importance of distinguishing hate speech from general profanity and offensive speech [13,14]. This distinction remains central to current research, since many false positives come from confusing vulgar but non-targeted language with hate speech, while many false negatives arise when hostile intent is expressed without explicit slurs.

Survey studies further show that dataset design, target definitions, and annotation guidelines strongly influence model behavior [15,16]. A model trained on narrow target categories or platform-specific expressions may perform well on an in-domain benchmark but fail to generalize to new communities, new coded terms, or new forms of implicit hostility. Recent multilingual and domain-specific studies also highlight the importance of language-specific annotation and model design, including Turkish ensemble transformers, Arabic dialectal multi-task learning, Nigerian Twitter data, Spanish multi-task hate speech corpora, and LLM-generated hate campaign benchmarks [17–21]. These studies show that hate speech detection is shaped by both linguistic structure and local social context.

For Chinese, COLDataset provides a benchmark for offensive language detection over race, gender, and region topics [5]. ToxiCN further introduces a hierarchical toxic language taxonomy and a Toxic Knowledge Enhancement (TKE) benchmark that incorporates lexical features for fine-grained Chinese toxic language detection [6]. STATE ToxiCN extends this line of work by emphasizing span-level target-aware toxicity extraction [7]. Recent work on phonetic cloaking also shows that offensive intent in Chinese can be deliberately hidden through pronunciation-based replacement [8]. These resources are valuable, but Chinese implicit hate speech still requires broader target coverage and stronger modeling of indirect, culturally situated hostility.

2.2. Implicit Hate and Semantic Opacity

Implicit hate speech is characterized by a mismatch between surface form and hostile intent. It may depend on metaphors, stereotypes, dog whistles, coded phrases, or socially shared assumptions. ElSherief et al. introduced Latent Hatred to model implicit hate through fine-grained categories and free-text explanations [3]. ToxiGen further showed that implicit toxicity can be generated and evaluated at large scale, with a focus on subtle statements about minority groups [4].

Recent work also shows that implicit hate may require broader context. Conversation-level benchmarks indicate that hostile meaning can depend on preceding turns, community conventions, or the interaction between speakers [22]. Multimodal studies further demonstrate that hateful intent may emerge from the interaction between text, image, and video signals rather than from isolated text alone [23,24]. These findings suggest that implicit hate detection should not be treated as ordinary text classification over surface tokens.

In Chinese, implicitness is amplified by homophones, local slang, forum-specific memes, and high-context expressions. A sentence may mention a seemingly ordinary social topic while implicitly invoking stereotypes about gender, region, religion, nationality, education, or social hierarchy. This motivates models that go beyond lexical matching and represent expression style as a meaningful semantic dimension. In FICHHS, this idea is operationalized through an explicit–implicit contrastive representation space, which provides a continuous implicitness signal for difficult-case routing.

2.3. Contrastive Learning and Retrieval-Augmented Reasoning

Supervised contrastive learning pulls samples with the same label together and separates samples with different labels through normalized batch-level competition [25]. It has been widely adopted for representation learning because it can improve class structure beyond conventional cross-entropy supervision. This is useful when the model needs to capture subtle distinctions such as explicit versus implicit expression. Retrieval-guided contrastive learning has also been explored for hateful meme detection, indicating that retrieval cues can complement contrastive representations in hate-related tasks [26].

Retrieval-augmented generation (RAG) combines parametric models with retrieved external memory to improve knowledge-intensive tasks [27]. In hate speech detection, retrieval, explanations, and external knowledge are especially useful when a text requires social, cultural, or contextual interpretation [28,29]. Structured explanations can help human moderators make faster and more consistent decisions, and recent evaluation work has highlighted the importance of reasoning quality in hate speech explanations. These studies suggest that external knowledge should be compact, interpretable, and closely tied to the moderation decision.

FICHS adapts these ideas to Chinese implicit hate speech by using contrastive learning for explicit–implicit representation and lightweight pattern retrieval for false-negative recovery. Specifically, EI-SCM uses supervised contrastive learning to model the semantic relation between explicit and implicit hate speech. SOAR uses the resulting implicitness signal to decide when LLM adjudication is needed. PGRM then retrieves lightweight pattern–note pairs derived from the training set to support second-round false-negative recovery. Unlike open-domain RAG, this retrieval component is designed as a targeted and interpretable knowledge aid for recurring socio-cultural risk patterns.

3. Methods

Figure 1 illustrates the overall architecture of FICHS. The framework first uses a RoBERTa-based base detector enhanced with toxic knowledge features to obtain an initial hate speech prediction and classifier confidence. In parallel, the Explicit–Implicit Supervised Contrastive Module (EI-SCM) learns an expression-aware representation space, where explicit and implicit hateful expressions are optimized through intra-mode cohesion and inter-mode separation, producing a semantic opacity signal for each sample.

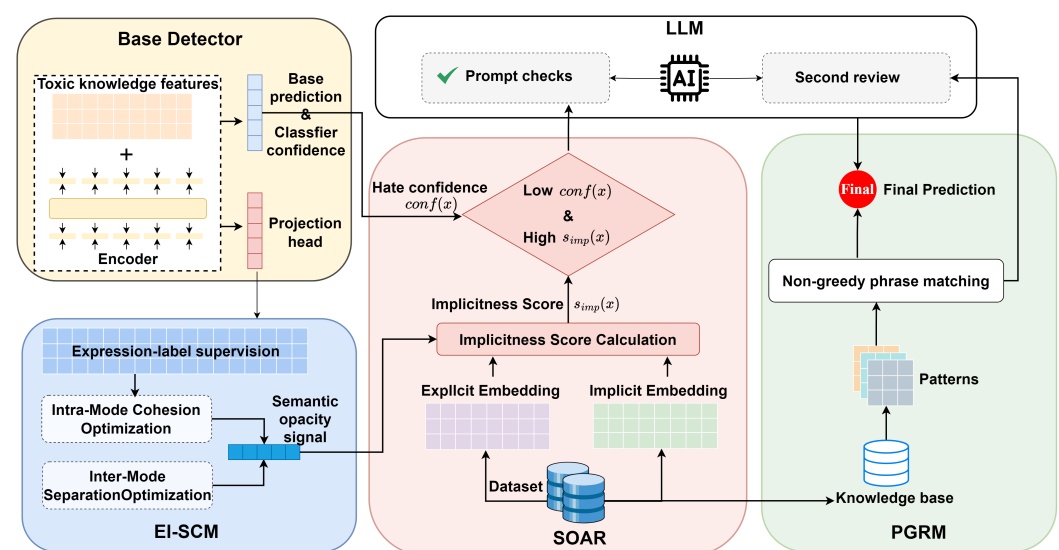


Figure 1. Architecture of FICHS.

The Semantic Opacity-Aware Routing and Adjudication module (SOAR) then integrates the classifier confidence from the base detector and the implicitness score derived from the contrastive representation space. Samples with low hate confidence but high implicitness are regarded as difficult implicit cases and are selectively routed to the LLM for prompt-based adjudication, while easier cases retain the base prediction. To further reduce false negatives, the Pattern-Guided Retrieval-Augmented Recovery Module (PGRM) retrieves matched risk patterns from a lightweight knowledge base through non-greedy phrase matching and triggers a second LLM review when necessary. The final prediction is therefore produced by combining the base detector, contrastive semantic opacity modeling, selective LLM reasoning, and pattern-guided recovery.

3.1. Base Detector

The base detector in FICHS follows the RoBERTa+TKE setting of ToxiCN [6]. The encoder is hf1/chinese-roberta-wwm-ext, a Chinese RoBERTa-style pretrained model with whole-word masking [30–32]. A two-layer feed-forward classifier predicts the binary toxicity label. The training objective for the base detector is binary cross-entropy over the one-hot toxicity target:

$$\mathbf{o}_i = q_\psi(f_\theta(x_i)) = [o_{i,0}, o_{i,1}], \quad (1)$$

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{NC} \sum_{i=1}^N \sum_{c=0}^{C-1} [y_{i,c} \log \sigma(o_{i,c}) + (1 - y_{i,c}) \log(1 - \sigma(o_{i,c}))]. \quad (2)$$

$$\hat{y}_{\text{base}}(x_i) = \arg \max_{c \in \{0,1\}} o_{i,c}. \quad (3)$$

Here, x_i is the i -th input text; $f_\theta(\cdot)$ is the RoBERTa encoder with parameters θ ; $q_\psi(\cdot)$ is the two-layer feed-forward classification head with parameters ψ ; $\mathbf{o}_i$ is the raw logit vector; $o_{i,0}$ and $o_{i,1}$ are the logits for the non-hate and hate classes; $y_{i,c} \in \{0, 1\}$ is the one-hot ground-truth value for class c ; N is the batch size; $C = 2$ is the number of classes; and $\sigma(\cdot)$ is the sigmoid function. Equation (2) corresponds to the BCEWithLogitsLoss used in the implementation, while Equation (3) corresponds to the argmax decision rule used during evaluation.

The model was trained for five epochs using AdamW with a learning rate of 1×10^{-5} , a batch size 32, a dropout rate of 0.5, a maximum sequence length of 80, and a random seed of 1.

3.2. Explicit–Implicit Supervised Contrastive Module

The EI-SCM module uses the expression label as supervision to model the semantic distinction between explicit and implicit hate speech. In the implementation, only samples with expression labels 1 and 2 are retained for contrastive learning, where explicit hate is mapped to label 0 and implicit hate is mapped to label 1. The contrastive encoder uses the [CLS] representation of hf1/chinese-roberta-wwm-ext, and the projection head is a linear layer followed by tanh. Let x_i denote an input text, $e_i \in \{0, 1\}$ its contrastive expression label, $f_\theta^{\text{con}}(\cdot)$ the contrastive encoder, and $g_\phi(\cdot)$ the projection head. The hidden representation and normalized contrastive embedding are defined as

$$\mathbf{h}_i = f_\theta^{\text{con}}(x_i)_{[\text{CLS}]}, \quad (4)$$

$$\mathbf{z}_i = \frac{g_\phi(\mathbf{h}_i)}{\|g_\phi(\mathbf{h}_i)\|_2}. \quad (5)$$

Here, $\mathbf{h}_i$ is the [CLS] hidden vector, $g_\phi(\mathbf{h}_i) = \tanh(\mathbf{W}\mathbf{h}_i + \mathbf{b})$ is the projected vector, $\mathbf{W}$ and $\mathbf{b}$ are projection parameters, and $\mathbf{z}_i$ is the L_2 -normalized embedding used for contrastive learning. The similarity between two samples is computed as

$$\text{sim}_{ij} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\tau}, \quad (6)$$

where τ is the temperature parameter. For a mini-batch B , the implementation constructs a positive mask from expression labels and removes self-comparisons from the exponential similarity matrix. The positive set and candidate set for sample i are

$$P(i) = \{j \in B \mid j \neq i, e_j = e_i\}, \quad A(i) = \{j \in B \mid j \neq i\}. \quad (7)$$

The supervised contrastive loss used in the code is

$$\mathcal{L}_{\text{SupCon}} = -\frac{1}{|V|} \sum_{i \in V} \log \frac{\sum_{p \in P(i)} \exp(\text{sim}_{ip}) + \epsilon}{\sum_{a \in A(i)} \exp(\text{sim}_{ia}) + \epsilon}. \quad (8)$$

In Equation (8), $V = \{i \in B \mid |P(i)| > 0\}$ is the set of valid anchors that have at least one positive sample in the same batch, and $\epsilon = 10^{-12}$ is a small constant used for numerical stability. This formula follows the implementation more closely than the standard averaged-positive SupCon expression: the numerator is the sum of all positive exponential similarities, and the denominator is the sum over all non-self candidates in the batch. The best-performing setting in our experiments uses a projection dimension of 256, a batch size of 64, a maximum sequence length of 128, a learning rate of 3×10^{-5} , five epochs, and a temperature of $\tau = 0.07$. Although the contrastive module is trained separately from the base classifier, it provides an embedding space for implicitness scoring and routing.

3.3. Semantic Opacity-Aware Routing and Adjudication

SOAR is designed to identify samples that are likely to be missed by the base detector because they appear non-hateful on the surface but are close to implicit hate in the contrastive space. Given the base model logits $\mathbf{o}_i = [o_{i,0}, o_{i,1}]$, the hate confidence is computed using temperature-scaled softmax:

$$\text{conf}(x_i) = p_{i,1} = \frac{\exp(o_{i,1}/T)}{\exp(o_{i,0}/T) + \exp(o_{i,1}/T)}. \quad (9)$$

Here, T is the confidence temperature, $p_{i,1}$ is the calibrated probability assigned to the hate class, and $o_{i,0}$ and $o_{i,1}$ are the same logits used by the base detector. This confidence score is used only for routing; the base classifier itself still predicts by arg max over raw logits.

Let D_{exp} and D_{imp} denote explicit and implicit hate samples in the contrastive training set. Their prototype centers are

$$\mathbf{c}_{\text{exp}} = \frac{1}{|D_{\text{exp}}|} \sum_{x \in D_{\text{exp}}} \mathbf{z}_x, \quad \mathbf{c}_{\text{imp}} = \frac{1}{|D_{\text{imp}}|} \sum_{x \in D_{\text{imp}}} \mathbf{z}_x. \quad (10)$$

For a sample x , its distances to the two prototypes are

$$d_{\text{exp}}(x) = \|\mathbf{z}_x - \mathbf{c}_{\text{exp}}\|_2, \quad d_{\text{imp}}(x) = \|\mathbf{z}_x - \mathbf{c}_{\text{imp}}\|_2. \quad (11)$$

The implicitness score is then defined as

$$s_{\text{imp}}(x) = \frac{d_{\text{exp}}(x)}{d_{\text{exp}}(x) + d_{\text{imp}}(x) + \epsilon}. \quad (12)$$

A larger $s_{\text{imp}}(x)$ indicates that the sample is relatively closer to the implicit prototype than to the explicit prototype. In Equations (10)–(12), D_{exp} and D_{imp} are the explicit and implicit training subsets, c_{exp} and c_{imp} are their prototype centers, d_{exp} and d_{imp} are Euclidean distances, and ϵ prevents division by zero. A sample is routed to the LLM adjudicator if

$$r(x) = \mathbb{I}(\gamma_{\text{low}} < \text{conf}(x) < \gamma_{\text{high}}) \cdot \mathbb{I}(s_{\text{imp}}(x) > \eta) = 1. \quad (13)$$

In Equation (13), $\mathbb{I}(\cdot)$ is the indicator function, γ_{low} and γ_{high} are the lower and upper confidence thresholds, and η is the implicitness threshold. Following the experimental configuration, $\gamma_{\text{low}} = 0$, $\gamma_{\text{high}} = 0.5$, and $\eta = 0.15$. Thus, the routed samples are those that the base model tends to regard as non-hate, but whose representations are sufficiently close to the implicit hate prototype.

The final SOAR prediction is

$$\hat{y}(x) = \begin{cases} \hat{y}_{\text{base}}(x), & r(x) = 0, \\ \hat{y}_{\text{llm}}(x), & r(x) = 1. \end{cases} \quad (14)$$

In Equation (14), $\hat{y}_{\text{base}}(x)$ is the base detector prediction and $\hat{y}_{\text{llm}}(x) \in \{0, 1\}$ is the LLM adjudication result. The LLM adjudicator used in the experiments is doubao-1-5-pro-256k-250115. The prompt instructs the model to perform protected-group identification, explicit attack detection, implicit attack detection, exclusion checking for neutral contexts, and conservative binary decision making.

3.4. Pattern-Guided Retrieval-Augmented Recovery

PGRM is introduced because the LLM still misses some implicit hate samples. The module builds a lightweight pattern knowledge base $\mathcal{K} = \{(p_k, n_k)\}_{k=1}^M$, where p_k is a high-frequency 2-gram or 3-gram pattern and n_k is a human-written note describing its possible hateful implication. To avoid information leakage, all pattern–note pairs used in the reported development and test experiments are constructed only from the training set. No development-set or test-set samples, labels, prediction errors, or LLM outputs are used to construct, select, or revise the PGRM knowledge base.

For a pattern $p_k = (w_{k,1}, w_{k,2}, \dots, w_{k,m})$, PGRM compiles an order-preserving flexible matching rule:

$$R(p_k) = \text{esc}(w_{k,1}) \cdot *? \text{esc}(w_{k,2}) \cdot *? \dots \cdot *? \text{esc}(w_{k,m}), \quad (15)$$

where $w_{k,t}$ is the t -th token in pattern p_k , m is the number of tokens in the pattern, $\text{esc}(\cdot)$ denotes regular-expression escaping, and $*?$ is the non-greedy wildcard used in the code to allow arbitrary inserted characters while preserving token order. The hit set is

$$H_K(x) = \text{TopK}(\{(p_k, n_k) \in \mathcal{K} \mid R(p_k) \text{ matches } x\}, K). \quad (16)$$

In Equation (16), $H_K(x)$ is the retrieved hit list and K is the maximum number of returned pattern–note pairs. The implementation uses $K = 5$. If the first LLM prediction is non-hate, the system performs a second LLM review. When $H_K(x)$ is non-empty, the matched pattern–note pairs are appended to the user prompt; when no pattern is matched, the original text is reviewed again without retrieved notes. Examples of knowledge entries include “region discrimination”, “artificial uterus”, “Lanzhou noodles”, “Chinese women”, and “Chinese education”, each paired with a note that warns about potential stereotypes, gender hostility, religious prejudice, or national–cultural bias.

The final PGRM decision is

$$\hat{y}(x) = \begin{cases} \hat{y}_{\text{llm}}(x), & \hat{y}_{\text{llm}}(x) = 1, \\ \hat{y}_{\text{rag}}(x; H_K(x)), & \hat{y}_{\text{llm}}(x) = 0. \end{cases} \quad (17)$$

Here, $\hat{y}_{\text{rag}}(x; H_K(x))$ denotes the second-round LLM prediction conditioned on the original text and the retrieved hit list. If $H_K(x) = \emptyset$, the conditioning text contains only the original content. Thus, PGRM follows the code-level behavior and acts as a targeted false-negative recovery mechanism rather than a full open-domain RAG pipeline.

4. Materials and Metrics

4.1. Dataset Construction

The dataset was constructed from two sources. The first source is the original Chinese toxic language data released by ToxiCN [6]. We retained the original ToxiCN schema and removed samples belonging to the ambiguous general-reporting expression category, resulting in 11,098 cleaned samples. The second source consists of 8841 newly collected samples from Zhihu and Baidu Tieba in March 2025. These samples were collected to expand the target coverage of the original ToxiCN-style data space, especially for religion, nation, physical condition, education, and hierarchy. The final dataset contains 19,939 samples and is split into training, validation, and test sets in an 8:1:1 ratio.

Each sample contains the text content, platform, target category, binary toxicity label, hate intensity label, expression label, target one-hot vector, and text length. The binary toxicity label is defined as non-hate or hate. Hate intensity has three levels: non-hate, offensive, and hate. Expression type has three labels: non-hate, explicit hate, and implicit hate. The target categories are LGBT, region, gender, race, religion, nation, physical condition, education, and hierarchy. The detailed distribution of these categories and labels is summarized in Table 1.

Table 1. Distribution of the final dataset.

Field	Category	Count
Platform	Zhihu	11,423
Platform	Baidu Tieba	8516
Target	Region	2370
Target	Hierarchy	1971
Target	Nation	2160
Target	Race	4774
Target	Gender	2875
Target	Education	1038
Target	LGBT	1997
Target	Physical condition	708
Target	Religion	2046
Toxicity	Non-hate	10,832
Toxicity	Hate	9107
Hate intensity	Non-hate	10,832
Hate intensity	Offensive	1188
Hate intensity	Hate	7919
Expression	Non-hate	12,019
Expression	Explicit	4741
Expression	Implicit	3179

The newly added target categories are religion, nation, physical condition, education, and hierarchy. These categories reflect frequent Chinese social media topics where implicit hostility is often embedded in jokes, coded words, or apparently neutral social commentary. The average text length of the final dataset is 36.10 Chinese characters.

For the newly collected portion, we followed a target-oriented sampling strategy. Candidate posts were retrieved using topic terms related to the five added categories, including religion-related discussions, country and nationality terms, physical-condition and health-related expressions, education-related discussions, and hierarchy- or social-status-related terms. We removed empty posts, clearly irrelevant posts, duplicate or near-duplicate entries, and samples that could not be consistently mapped to the adopted toxicity, intensity, expression, and target schema.

4.2. Annotation Guidelines

The annotation schema and procedure were adapted from the hierarchical toxic-language annotation framework of ToxiCN [6]. For the inherited ToxiCN portion, we retained the original annotations after removing samples belonging to the ambiguous general-reporting expression category. For the newly collected portion, we applied the same hierarchical annotation logic and normalized all labels into the same schema, including binary toxicity, hate intensity, target category, and expression type.

Each newly collected sample was annotated by at least three annotators. Following the ToxiCN annotation procedure, the annotators first determined whether the text was hateful and then assigned its hate intensity, expression type, and target category. The final annotation was determined by majority voting. Before formal annotation, the annotators were familiarized with the ToxiCN annotation framework and the operational definitions adopted in this study.

A sample was marked as hateful if it attacked, demeaned, excluded, or expressed hostility toward a protected or socially salient group rather than merely criticizing an individual. Explicit hate was defined as direct hostility using insults, stigmatizing labels, dehumanization, or exclusionary language. Implicit hate was defined as indirect hostility conveyed through stereotypes, microaggressions, irony, metaphors, coded references, or seemingly humorous expressions. Neutral academic, legal, historical, or news discussions, as well as condemnation of hate speech, were marked as non-hate when no hostile intent was expressed.

4.3. Evaluation Metrics

We report the weighted F1 score, precision, recall, class-wise F1 scores, and loss. Weighted F1 score is used as the primary metric because the dataset is moderately imbalanced between non-hate and hate classes. To analyze the value of LLM adjudication, we additionally compute the net correction effect:

$$\Delta_{\text{llm}} = N(\text{base wrong, llm correct}) - N(\text{base correct, llm wrong}). \quad (18)$$

In Equation (18), $N(\text{base wrong, llm correct})$ is the number of disagreement cases in which the LLM corrects a base-model error, while $N(\text{base correct, llm wrong})$ is the number of disagreement cases in which the LLM introduces an error. A positive Δ_{llm} indicates that LLM adjudication is beneficial under the routing condition.

5. Results

5.1. Base Detector Performance

Table 2 reports the performance of the RoBERTa+TKE base detector on the development and test sets. The base detector achieves a development weighted F1 score of 0.8288

and a test weighted F1 score of 0.8025. The test result is close to the best result reported for the original baseline setting, but it remains limited on implicit and boundary cases.

Table 2. Performance of the RoBERTa+TKE base detector.

Split	Weighted F1	Class-Wise F1	Precision	Recall
Development	0.8288	[0.8482, 0.8025]	0.8302	0.8281
Test	0.8025	[0.8240, 0.7740]	0.8034	0.8021

The class-wise F1 scores show that the hate class is harder to classify than the non-hate class. On the test set, the non-hate F1 score is 0.8240, whereas the hate F1 score is 0.7740. This 5.00-point gap is consistent with the task motivation: many hateful samples do not contain direct lexical cues and are therefore more likely to be missed by a surface-sensitive classifier. The similar values of weighted F1 score, precision, and recall also indicate that the base detector is not strongly biased toward either precision or recall; its main weakness lies in the semantic difficulty of the positive class.

5.2. Contrastive Representation Analysis

The contrastive module is not used as a standalone toxicity classifier. Instead, it provides an explicit–implicit semantic space for implicitness estimation. With a projection dimension of 256, a temperature of 0.07, and a batch size of 64, the module reaches a test kNN accuracy of 0.6816 and a kNN ROC AUC of 0.6738 for explicit versus implicit expression classification. The silhouette score is low and is therefore interpreted cautiously. More importantly, the prototype-based implicitness scores show a clear distributional difference between explicit and implicit samples, as shown in Table 3.

These results indicate that, although explicit and implicit hate speech are not perfectly separable, the contrastive embedding space provides a useful continuous signal for detecting semantically opaque samples.

Figure 2 visualizes the mean implicitness scores. The development-set gap between implicit and explicit samples is 0.212, while the test-set gap is 0.185. The gap narrows on the test set, suggesting distributional variability, but it remains large enough to support threshold-based routing. This is why we use the score as an auxiliary signal rather than as an independent classifier.

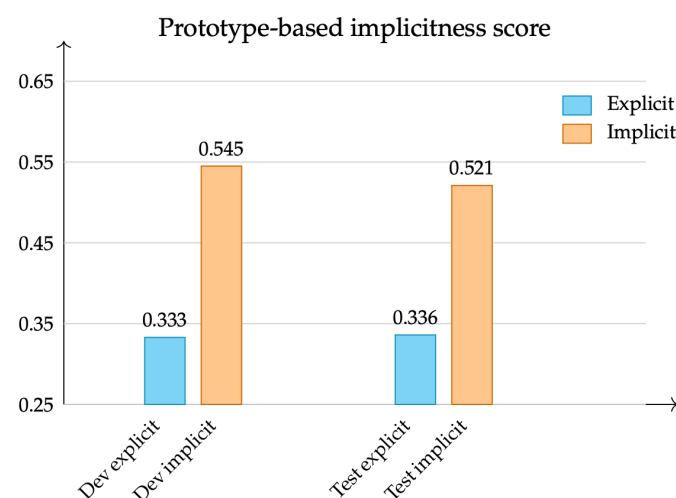


Figure 2. Mean implicitness scores for explicit and implicit hate speech on the development and test sets. The larger implicit scores for implicit samples support the use of the contrastive space as a routing signal.

Table 3. Prototype-based implicitness score distributions.

Split	Expression	Mean	Median	Std.	Min.	Max.
Development	Explicit	0.333	0.273	0.191	0.118	0.820
Development	Implicit	0.545	0.578	0.201	0.145	0.834
Test	Explicit	0.336	0.268	0.197	0.120	0.800
Test	Implicit	0.521	0.543	0.207	0.127	0.827

5.3. Effect of SOAR

Table 4 reports the performance on the development and test sets after adding SOAR. Adding SOAR improves the weighted F1 score from 0.8288 to 0.8504 on the development set and from 0.8025 to 0.8271 on the test set. Precision and recall also improve simultaneously, suggesting that selective LLM adjudication does not merely trade one type of error for another.

Table 4. Effect of Semantic Opacity-Aware Routing and Adjudication.

Split	Weighted F1	Class-Wise F1	Precision	Recall
Development	0.8504	[0.8606, 0.8394]	0.8524	0.8508
Test	0.8271	[0.8389, 0.8146]	0.8290	0.8276

The disagreement analysis further supports the value of selective adjudication. On the development set, among base-model errors, 80 samples are corrected by the LLM, while 36 originally correct base predictions are changed incorrectly, yielding a net correction of +44. On the test set, 109 samples show disagreement between the base model and LLM; 80 are cases where the LLM is correct and 29 are cases where the base model is correct, yielding a net correction of +51. This shows that, when disagreement occurs under the SOAR routing condition, trusting the LLM is beneficial overall.

Table 5 clarifies why SOAR improves both precision and recall. Among the routed/error cases, the LLM corrected 80 base-model errors, introduced 29 reversals where the base model was correct, and failed on 132 cases where both systems produced incorrect predictions. The LLM does introduce errors, but the number of corrected base-model errors is substantially larger than the number of newly introduced errors. In addition, the “both wrong” column indicates that the remaining difficult cases are not merely classifier-boundary errors. They often require additional socio-cultural cues, which motivates the PGRM stage.

Table 5. Disagreement analysis for LLM adjudication.

Split	Base Wrong, LLM Correct	Base Correct, LLM Wrong	Both Wrong	Net Correction
Development	80	36	109	+44
Test	80	29	132	+51

5.4. Effect of PGRM

PGRM further improves performance after SOAR by recovering additional false negatives. The development weighted F1 increases from 0.8504 to 0.8636, and the test weighted F1 score increases from 0.8271 to 0.8391. Table 6 reports the final results.

The gain from PGRM indicates that some difficult samples are not only semantically implicit but also tied to recurring social topics or coded patterns. When the prompt includes matched pattern-level notes, the LLM is better able to reconsider texts that initially appear neutral.

Table 6. Performance after Pattern-Guided Retrieval-Augmented Recovery.

Split	Weighted F1	Class-Wise F1	Precision	Recall
Development	0.8636	[0.8714, 0.8558]	0.8684	0.8638
Test	0.8391	[0.8484, 0.8297]	0.8428	0.8396

Table 7 reports all standalone LLM baselines under the same setting. Under this unified evaluation protocol, FICHS outperforms the strongest standalone LLM baseline by 4.43 percentage points in terms of weighted F1 score. The improvement is also reflected in the hate-class F1 score, where FICHS achieves 0.8297 compared with 0.7576 for Qwen3-235B-A22B. These results suggest that directly replacing the detector with a general-purpose LLM is not sufficient for this task. FICHS provides a more effective alternative by combining a task-specific detector with selective LLM adjudication.

Table 7. Comparison of standalone LLM baselines and FICHS on the test set.

Method	Best Weighted F1	Class-Wise F1	Precision	Recall
GPT-4o	0.7527	0.8146, 0.6758	0.7893	0.7641
DeepSeek-V3	0.7400	0.7849, 0.6843	0.7463	0.7441
Qwen3-32B	0.7836	0.7893, 0.7766	0.7932	0.7831
Doubao-1.5-Pro	0.7558	0.8190, 0.6773	0.7981	0.7681
Qwen3-235B-A22B	0.7948	0.8248, 0.7576	0.7977	0.7966
FICHS	0.8391	0.8484, 0.8297	0.8428	0.8396

5.5. Ablation Summary

Table 8 summarizes the incremental contribution of each FICHS component. Compared with the base detector, FICHS improves the weighted F1 score on the test set by 3.66 percentage points, precision by 3.94 percentage points, and recall by 3.75 percentage points.

Table 8. Ablation comparison on development and test sets.

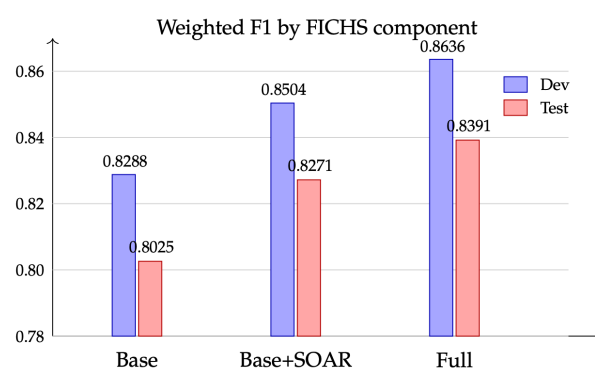
Split	Method	Weighted F1	Precision	Recall
Development	Base detector	0.8288	0.8302	0.8281
Development	Base + SOAR	0.8504	0.8524	0.8508
Development	Base + SOAR + PGRM	0.8636	0.8684	0.8638
Test	Base detector	0.8025	0.8034	0.8021
Test	Base + SOAR	0.8271	0.8290	0.8276
Test	Base + SOAR + PGRM	0.8391	0.8428	0.8396

Table 9 shows that SOAR contributes the larger single-stage gain, while PGRM provides a smaller but consistent second-stage improvement. The test-set gain from SOAR is slightly larger than its development-set gain, suggesting that semantic-opacity routing generalizes beyond the development set. PGRM also improves precision and recall together, which indicates that retrieval notes are not simply making the system more aggressive; rather, they help the LLM recover specific false negatives without causing a large precision collapse.

Figure 3 makes the additive pattern of FICHS visible. The base detector provides the initial decision boundary, SOAR supplies the largest jump by using LLM reasoning only on routed difficult cases, and PGRM adds a second but stable gain by correcting remaining LLM false negatives. The same trend appears on both the development and test sets, which supports the robustness of the overall FICHS pipeline.

Table 9. Incremental metric gains from each module.

Split	Increment	$\Delta F1$	$\Delta Precision$	$\Delta Recall$
Development	SOAR over Base	+0.0215	+0.0222	+0.0226
Development	PGRM over SOAR	+0.0133	+0.0160	+0.0131
Development	Full over Base	+0.0348	+0.0382	+0.0357
Test	SOAR over Base	+0.0246	+0.0256	+0.0255
Test	PGRM over SOAR	+0.0120	+0.0138	+0.0120
Test	Full over Base	+0.0366	+0.0394	+0.0375

**Figure 3.** Weighted F1 score comparison across FICHS components. Bar heights are scaled from 0.78 to emphasize incremental gains.

6. Discussion

The results show that implicit hate speech detection benefits from a division of labor between a compact task-specific detector, representation-level implicitness estimation, and selective LLM-based reasoning. The base detector handles most samples efficiently, while EI-SCM supplies a semantic opacity signal that is not directly available from classifier logits. SOAR then uses this signal to identify low-confidence and high-implicitness samples for LLM adjudication. This is important because LLMs are expensive and not always reliable as full replacement classifiers; selective routing concentrates their reasoning ability on samples where the base model is most vulnerable.

PGRM provides another layer of improvement by making socio-cultural risk patterns explicit. Unlike conventional RAG, which retrieves long documents from an external corpus, PGRM retrieves compact pattern–note pairs that are directly tied to Chinese implicit hate contexts. This design is lightweight, interpretable, and easy to update when new coded expressions emerge. It also aligns with practical moderation needs: reviewers need not only a label but also a clue about why a seemingly ordinary topic may carry hateful intent.

Several limitations remain. First, although the dataset supports richer tasks such as hate intensity, target classification, and expression classification, the current pipeline is evaluated primarily as a binary hate detector. This setting is useful for comparison with existing moderation-oriented benchmarks, but future work should extend FICHS to multi-task prediction over toxicity, intensity, target group, and expression type. Second, the LLM adjudicator is treated as a black-box component, and its behavior may change across model versions, decoding settings, or safety policies. This affects reproducibility and long-term deployment, so future applications should record model versions, prompts, inference parameters, and safety settings, and should periodically re-evaluate the adjudicator after model updates. Third, because the data are drawn from Zhihu and Baidu Tieba, additional validation is needed before applying the system to other Chinese platforms, short-video comments, microblogs, or multimodal content. In particular, future work should evaluate FICHS on additional Chinese social media platforms and explore

automatic or semi-automatic expansion of the pattern knowledge base to improve cross-platform external validity. Finally, the pattern knowledge base relies on human-written notes, which improves interpretability but may introduce annotator bias when culturally situated or ambiguous expressions are interpreted. Future versions of PGRM should incorporate multi-annotator note construction, adjudication, bias auditing, and periodic revision to reduce this subjectivity.

7. Conclusions

This paper proposed FICHS, a fine-grained and implicit Chinese hate speech detection framework based on contrastive learning and retrieval-augmented adjudication. We expanded the dataset to 19,939 samples with nine target categories and introduced three complementary modules: EI-SCM for explicit–implicit semantic representation, SOAR for low-confidence high-implicitness routing to an LLM, and PGRM for pattern-guided false-negative recovery. Experiments demonstrate that FICHS consistently improves development and test performance over the RoBERTa+TKE base detector, increasing the weighted F1 score on the test set from 0.8025 to 0.8391. The findings suggest that implicit hate speech detection should combine learned semantic geometry with selective reasoning and explicit socio-cultural knowledge, rather than relying on either small classifiers or LLMs alone. Future research could explore multimodal extensions and cross-lingual transfer of FICHS, with appropriate language- and culture-specific adaptation.

Author Contributions: Conceptualization, X.H.; methodology, K.H. and X.H.; software, K.H.; validation, Y.J.; investigation, C.L.; data curation, H.X.; writing—original draft preparation, X.H.; writing—review and editing, X.H. and K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fundamental Research Funds for the Central Universities, project number CUC26TD02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: This work was supported by Public Computing Cloud, CUC.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FICHS	Fine-Grained and Implicit Chinese Hate Speech Detection
EI-SCM	Explicit–Implicit Supervised Contrastive Module
PGRM	Pattern-Guided Retrieval-Augmented Recovery Module
RAG	Retrieval-Augmented Generation
SOAR	Semantic Opacity-Aware Routing and Adjudication
TKE	Toxic Knowledge Enhancement

References

1. Geetanjali; Kumar, M. Exploring Hate Speech Detection: Challenges, Resources, Current Research and Future Directions. *Multimed. Tools Appl.* **2025**, *84*, 38423–38459. [CrossRef]
2. Ghenai, A.; Noorian, Z.; Moradisani, H.; Abadeh, P.; Erentzen, C.; Zarrinkalam, F. Exploring Hate Speech Dynamics: The Emotional, Linguistic, and Thematic Impact on Social Media Users. *Inf. Process. Manag.* **2025**, *62*, 104079. [CrossRef]

3. ElSherief, M.; Ziems, C.; Muchlinski, D.; Anupindi, V.; Seybolt, J.; De Choudhury, M.; Yang, D. Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 345–363.
4. Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; Kamar, E. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 3309–3326.
5. Deng, J.; Zhou, J.; Sun, H.; Zheng, C.; Mi, F.; Meng, H.; Huang, M. COLD: A Benchmark for Chinese Offensive Language Detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 11580–11599.
6. Lu, J.; Xu, B.; Zhang, X.; Min, C.; Yang, L.; Lin, H. 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*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 16235–16250. [\[CrossRef\]](#)
7. Bai, Z.; Yin, S.; Lu, J.; Zeng, J.; Zhu, H.; Sun, Y.; Yang, L.; Lin, H. STATE ToxiCN: A Benchmark for Span-Level Target-Aware Toxicity Extraction in Chinese Hate Speech Detection. *arXiv* **2025**, arXiv:2501.15451. [\[CrossRef\]](#)
8. Guo, H.; He, J.; Ma, J.; Na, H.; Wang, Z.; Zhang, H.; Chen, Q.; Wang, W.; Shi, Z.; Shen, T.; et al. Lost in Pronunciation: Detecting Chinese Offensive Language Disguised by Phonetic Cloaking Replacement. *arXiv* **2025**, arXiv:2507.07640. [\[CrossRef\]](#)
9. Albladi, A.; Islam, M.; Das, A.; Bigonah, M.; Zhang, Z.; Jamshidi, F.; Rahgouy, M.; Raychawdhary, N.; Marghitu, D.; Seals, C. Hate Speech Detection Using Large Language Models: A Comprehensive Review. *IEEE Access* **2025**, *13*, 20871–20892. [\[CrossRef\]](#)
10. Umansky, N.; Kubli, M.; Kotarcic, A.; Bronner, L.; Kurer, S.; Grech, P.; Hangartner, D.; Gilardi, F.; Donnay, K. Improving Hate Speech Detection with Large Language Models. *Eur. J. Polit. Res.* **2026**, 1–12. [\[CrossRef\]](#)
11. Rehman, M.Z.U.; Kasu, S.K.R.; Koppula, S.R.; Chirra, S.R.R.; Singh, S.S.; Kumar, N. X-MuTeST: A Multilingual Benchmark for Explainable Hate Speech Detection and A Novel LLM-Consulted Explanation Framework. *arXiv* **2026**, arXiv:2601.03194. [\[CrossRef\]](#)
12. Zhang, M.; He, J.; Ji, T.; Lu, C.-T. Don't Go To Extremes: Revealing the Excessive Sensitivity and Calibration Limitations of LLMs in Implicit Hate Speech Detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 12073–12086. [\[CrossRef\]](#)
13. Waseem, Z.; Hovy, D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 88–93.
14. Davidson, T.; Warmley, D.; Macy, M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the International AAAI Conference on Web and Social Media*; AAAI Press: Washington, DC, USA, 2017; Volume 11, pp. 512–515.
15. Fortuna, P.; Nunes, S. A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.* **2018**, *51*, 1–30. [\[CrossRef\]](#)
16. Poletto, F.; Basile, V.; Sanguinetti, M.; Bosco, C.; Patti, V. Resources and Benchmark Corpora for Hate Speech Detection: A Systematic Review. *Lang. Resour. Eval.* **2021**, *55*, 477–523. [\[CrossRef\]](#)
17. Aksoy, C.; Demirezen, M.U.; Sagioglu, S. Hate Speech Detection in Turkish: An Ensemble Transformer-Based Deep Learning Approach. *Eng. Appl. Artif. Intell.* **2026**, *164*, 113147. [\[CrossRef\]](#)
18. Abdelsamie, M.M.; Azab, S.S.; Hefny, H.A. The Dialects Gap: A Multi-Task Learning Approach for Enhancing Hate Speech Detection in Arabic Dialects. *Expert Syst. Appl.* **2026**, *295*, 128584. [\[CrossRef\]](#)
19. Tonneau, M.; Quinta De Castro, P.; Lasri, K.; Farouq, I.; Subramanian, L.; Orozco-Olvera, V.; Fraiberger, S. NaijaHate: Evaluating Hate Speech Detection on Nigerian Twitter Using Representative Data. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 9020–9040. [\[CrossRef\]](#)
20. Pan, R.; García-Díaz, J.A.; Valencia-García, R. Spanish MTLHateCorpus 2023: Multi-Task Learning for Hate Speech Detection to Identify Speech Type, Target, Target Group and Intensity. *Comput. Stand. Interfaces* **2025**, *94*, 103990. [\[CrossRef\]](#)
21. Shen, X.; Wu, Y.; Qu, Y.; Backes, M.; Zannettou, S.; Zhang, Y. HateBench: Benchmarking Hate Speech Detectors on LLM-Generated Content and Hate Campaigns. In *Proceedings of the 2025 ACM Internet Measurement Conference*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 221–240.
22. Ghosh, S.; Suri, M.; Chiniya, P.; Tyagi, U.; Kumar, S.; Manocha, D. 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*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 6159–6173. [\[CrossRef\]](#)
23. Rehman, M.Z.U.; Bhatnagar, A.; Kabde, O.; Bansal, S.; Kumar, N. ImpliHateVid: A Benchmark Dataset and Two-Stage Contrastive Learning Framework for Implicit Hate Speech Detection in Videos. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 17209–17221. [\[CrossRef\]](#)
24. Kapil, P.; Ekbal, A. A Transformer Based Multi Task Learning Approach to Multimodal Hate Speech Detection. *Nat. Lang. Process. J.* **2025**, *11*, 100133. [\[CrossRef\]](#)

25. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 18661–18673.
26. Mei, J.; Chen, J.; Lin, W.; Byrne, B.; Tomalin, M. Improving Hateful Meme Detection through Retrieval-Guided Contrastive Learning. *arXiv* **2024**, arXiv:2311.08110. [[CrossRef](#)]
27. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Kuttler, H.; Lewis, M.; Yih, W.-t.; Rocktaschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 9459–9474.
28. Calabrese, A.; Neves, L.; Shah, N.; Bos, M.; Ross, B.; Lapata, M.; Barbieri, F. Explainability and Hate Speech: Structured Explanations Make Social Media Moderators Faster. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 398–408. [[CrossRef](#)]
29. Hu, Y.; Lee, R.K.-W. HateXScore: A Metric Suite for Evaluating Reasoning Quality in Hate Speech Explanations. *arXiv* **2026**, arXiv:2601.13547. [[CrossRef](#)]
30. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692. [[CrossRef](#)]
31. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Pre-Training with Whole Word Masking for Chinese BERT. *IEEE/ACM Trans. Audio Speech Lang. Proc.* **2021**, 29, 3504–3514. [[CrossRef](#)]
32. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 657–668.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.