RV-HATE: Reinforced Multi-Module Voting for Implicit Hate Speech Detection

Yejin Lee, Hyeseon An, Yo-Sub Han[*]
Yonsei University, Seoul, Republic of Korea,
`{ssgyejin,hsan,emmous}@yonsei.ac.kr`

Abstract

Hate speech remains prevalent in human society and continues to evolve in its forms and expressions. Modern advancements in the internet and online anonymity accelerate its rapid spread and complicate its detection. However, hate speech datasets exhibit diverse characteristics primarily because they are constructed from different sources and platforms, each reflecting different linguistic styles and social contexts. Despite this diversity, prior studies on hate speech detection often rely on fixed methodologies without adapting to data-specific features. We introduce RV-HATE, a detection framework designed to account for the dataset-specific characteristics of each hate speech dataset. RV-HATE consists of multiple specialized modules, where each module focuses on distinct linguistic or contextual features of hate speech. The framework employs reinforcement learning to optimize weights that determine the contribution of each module for a given dataset. A voting mechanism then aggregates the module outputs to produce the final decision. RV-HATE offers two primary advantages: (1) it improves detection accuracy by tailoring the detection process to dataset-specific attributes, and (2) it also provides interpretable insights into the distinctive features of each dataset. Consequently, our approach effectively addresses implicit hate speech and achieves superior performance compared to conventional static methods. Our code is available at `https://github.com/leeyejin1231/RV-HATE`.

1 Introduction

Warning: *this paper contains content that may be offensive and upsetting.*

Online platforms continue to grow rapidly and this growth increases the prevalence of hate speech (Madriaza et al., 2025). Hate speech refers to language that promotes hatred, discrimination, or violence toward a specific group or community—gender, race, religion, nationality, or other identities (Poletto et al., 2021). It is typically categorized into two types: explicit and implicit. While explicit hate speech can be easily identified via explicit abusive expressions or lexicons (Waseem et al., 2017; Caselli et al., 2020; Ocampo et al., 2023), implicit hate speech remains challenging due to its subtle and context-dependent nature. For these reasons, detecting hate speech has become a critical task. Additionally, hate speech datasets exhibit diverse linguistic and contextual variation, primarily due to their construction from diverse sources and platforms that reflect different language conventions and social dynamics. As a result, these datasets vary in linguistic style, degree of implicitness, and annotation criteria. Therefore, robust hate speech detection methods must account for the dataset-specific characteristics of individual datasets.

Recent research has focused on developing methods for hate speech detection. For instance, Ahn et al. (2024) employed clustering in the sentence embedding space to identify representative samples for contrastive learning, and Kim et al. (2024) employed an additional queue for hard negative samples beyond the batch-level. Although prior studies have made progress in implicit hate speech detection, they often overlook dataset-specific features and distinctive labeling criteria. We must design methods that account for the distinct characteristics of each dataset to effectively address these limitations.

We propose a reinforced multi-module voting method for implicit hate speech detection (RV-HATE). RV-HATE consists of multiple modules designed to capture the unique properties of each dataset and employs a reinforced voting mechanism that adaptively optimizes their contributions based on dataset-specific characteristics. RV-HATE consists of four modules ($\mathtt{M}_0 - \mathtt{M}_3$): The $\mathtt{M}_0$ module serves as the base module, capturing the context of

---

[*] Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 45364–45383
July 2-7, 2026

hate speech by using the cosine similarity during clustering-based contrastive learning. The $\mathtt{M}_1$ module tags the hate targets, thereby enabling more precise discrimination of hate speech. The $\mathtt{M}_2$ module removes the outliers in clusters of the dataset to guarantee the quality of data and the $\mathtt{M}_3$ module utilizes hard negative samples during the contrastive learning to provide a clear decision boundary. Each module is constructed by augmenting the base module $\mathtt{M}_0$ with its corresponding functionality. We finetune the four classifier models with each module, and employ these classifiers in a voting process, where reinforcement learning dynamically assigns dataset-specific weights to each classifier.

Through this voting mechanism, RV-HATE not only outperforms previous state-of-the-art (SOTA) models by an average of 1.8%p, but also provides interpretability by revealing how each module contributes under varying dataset characteristics. While an improvement of 1–2% may appear incremental, such gains are particularly significant in hate speech detection, where performance typically plateaus around the 80% range (Ahn et al., 2024; Kim et al., 2024). Our approach highlights the importance of dataset-specific strategy in hate speech detection and demonstrates that employing multi-modules can both improve detection performance and provide explainable insights into the characteristics of implicit hate speech datasets.

Our main contributions are as follows.

- We develop four specialized modules designed to capture diverse dataset-specific characteristics of implicit hate speech.
- We present a reinforcement learning-based voting mechanism that assigns dataset-specific weights to each module, enabling adaptive combination of module predictions.
- We improve the detection performance across multiple benchmarks and provide interpretable insights into how different modules contribute under the dataset characteristics.

2 Related Work

2.1 Implicit Hate Speech Detection

Hate speech detection plays a crucial role in mitigating online toxicity and preventing the spread of harmful communication (Gandhi et al., 2024; Lee et al., 2018). Implicit hate speech expresses hateful or discriminatory intent indirectly, often relying on context instead of explicit slurs or offensive sentences (Weber et al., 2020). This subtle aspect makes implicit hate speech more challenging to detect than explicit hate speech. There are a few datasets for implicit hate speech detection. ElSherief et al. (2021) introduced the Implicit Hate Corpus (IHC) that captures subtle and indirect hate speech relying on contextual cues and implicit stereotypes. The dataset contains implications for each hateful sentence, which provide explanations of their implied meanings. Hartvigsen et al. (2022) focused on adversarial non-toxic counterfactuals to evaluate model generalization in hate speech detection and incorporated LLM-generated neutral sentences. Lee et al. (2024) constructed a cross-cultural English hate speech dataset and analyzed how cultural background influences hate speech annotations across different countries. In addition to dataset development, researchers tackled the implicit hate speech problem by investigating its linguistic structures and the difficulties posed by context-dependent or culturally grounded expressions (Fortuna et al., 2021; Davani et al., 2023; Ocampo et al., 2023).

Huang et al. (2023) assessed the capability of LLMs to detect implicit hate speech by generating concise natural language explanations. Park et al. (2024) introduced a multi-agent-based debate simulation framework that generates diverse perspectives on implicit hate speech. These approaches demonstrate the significance of improving the robustness of implicit hate speech detection models. Kim et al. (2025) presented a CONELA, a data refinement strategy that utilizes human agreement and training dynamics to improve generalization in implicit hate speech detection. Lee et al. (2025) proposed a target-aware attention framework that models interactions between explicit and implicit targets and their context to improve the ability to detect implicit hate speech.

2.2 Hate Speech Detection with Contrastive Learning

Researchers notice that contrastive learning is effective for detecting implicit hate speech by distinguishing the subtle semantic nuances and context-dependent cues that characterize such language. Kim et al. (2022) applied contrastive learning using implication data to detect implicit hate speech from neutral text. Kim et al. (2023) proposed an augmentation approach that utilizes machine-generated data to enhance implicit hate speech detection.

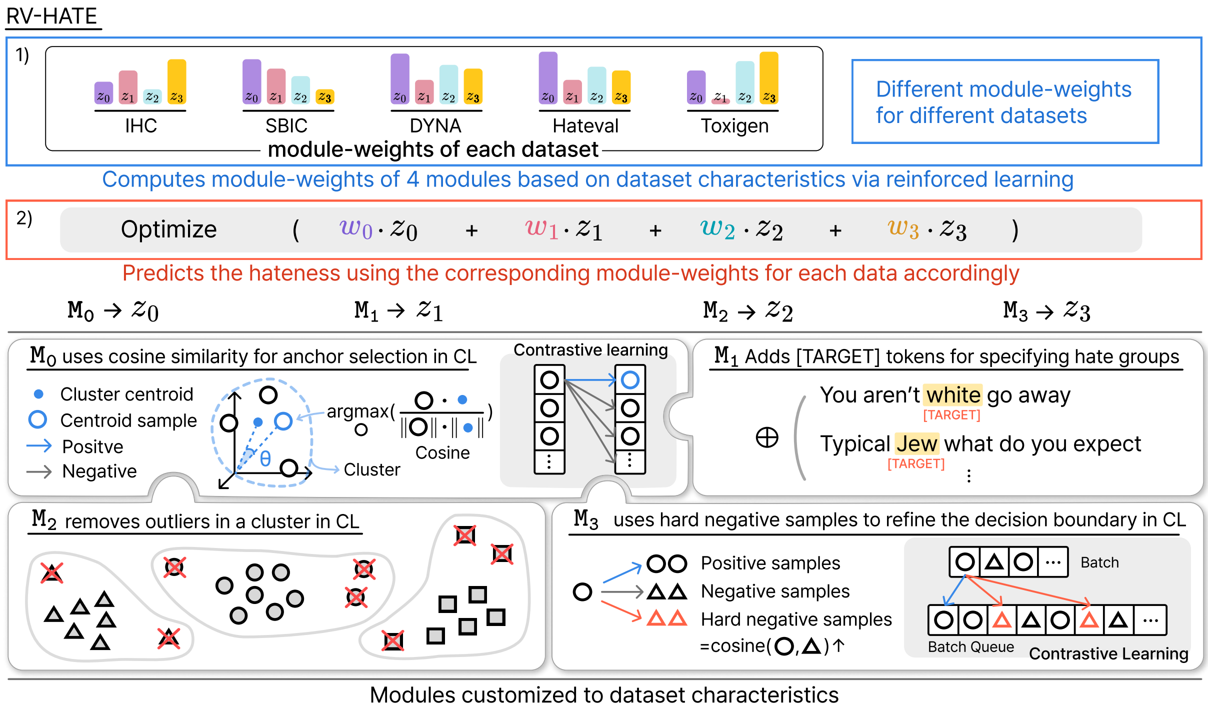


Figure 1: Overall workflow of RV-HATE. The method processes implicit hate speech data through four modules $\mathtt{M}_0$ (Sec. 3.1), $\mathtt{M}_1$ (Sec. 3.2), $\mathtt{M}_2$ (Sec. 3.3), and $\mathtt{M}_3$ (Sec. 3.4). Reinforcement learning is employed to determine the optimal weights for these modules in the voting process.

Huang and Usbeck (2024) employed a revisiting supervised contrastive learning method for subtle hate speech detection. Kim et al. (2024) proposed a negative sampling strategy using momentum contrastive learning with an additional queue. Jiang (2025) proposed an approach that applies the causal inference method to refine contrastive learning.

Ahn et al. (2024) proposed SharedCon, a SOTA model for implicit hate speech detection. The model employs clustering-based contrastive learning to improve contextual representation and model robustness. In contrastive learning, the model learns to bring similar samples closer in the representation space using an anchor. SharedCon selects the sample closest to each cluster center as an anchor, allowing the model to effectively learn shared semantic patterns across data.

3 RV-HATE

RV-HATE consists of four modules ($\mathtt{M}_0$–$\mathtt{M}_3$) designed to capture dataset-specific characteristics. The three modules $\mathtt{M}_1, \mathtt{M}_2, \mathtt{M}_3$ extend $\mathtt{M}_0$ by incorporating their specialized functionality for hate speech detection. We fine-tune each module using contrastive learning and then present a voting mechanism that combines the outputs of all modules to reflect the diverse characteristics of each dataset. We further enhance the performance via reinforcement learning (Fu et al., 2025; Birman et al., 2022) that assigns optimized and dataset-specific weights for these modules during the voting process. The overall framework is illustrated in Figure 1.

3.1 Clustering-based Contrastive Learning ($\mathtt{M}_0$)

We adopt the SharedCon approach of Ahn et al. (2024) and refine the model by modifying the criterion for anchor selection based on the intuition that cosine similarity better reflects semantic alignment in high-dimensional embedding space. Accordingly, we use cosine similarity instead of Euclidean distance when measuring the margin between cluster centers and their closest representatives. Unlike Euclidean distance, which captures absolute magnitude difference, cosine similarity focuses on vector direction. This helps to identify semantically similar data in contrastive learning better. The example sentences of cosine similarity and Euclidean distance are shown in Appendix A.4.

3.2 Adding [TARGET] Tokens ($\mathtt{M}_1$)

Hate speech is generally defined as expressions conveying hatred or discrimination toward a specific target, and it is considered a subset of abusive text (Poletto et al., 2021). Abusive (offensive) text refers to any expression intended to insult, humiliate, threaten, or harass another person. The primary distinction between hate speech and abusive text lies in the presence of a **specific target**. Therefore, a sentence that expresses insults without clear targets is considered abusive text, not hate speech. However, we observe that this boundary is often blurred in widely used hate speech datasets (Appendix A.1) where offensive and hateful language are often hard to distinguish. We therefore design $\mathtt{M}_1$ that tags tokens referring to specific groups or institutions in hate-labeled data to address this issue. We use spaCy (Honnibal, 2017) for Named Entity Recognition (NER) tagging and gpt-4o as a supplementary tagger when spaCy fails to tag certain entities. Tagging with spaCy covers approximately 18.75% of the data on average, and supplementary tagging with gpt-4o increases this coverage to almost 50.88%. Following the previous study (Khurana et al., 2025), we specifically focused on [ORG] (organization), [NORP] (nationalities), and [GPE] (country) entities as target tokens for RV-HATE. We augment the training dataset with target-tagged hate-labeled data, allowing the model to better understand context through target entities and distinguish between offensive language and hate speech.

3.3 Outlier Removal within Clusters ($\mathtt{M}_2$)

Hate speech datasets often contain broken sentences, because they are primarily collected through web crawling. A broken sentence refers to an incomplete or fragmented sentence that lacks essential grammatical components. We analyze 500 randomly sampled data from each dataset using gpt-4.1 to quantify the presence of such sentences. The analysis reveals that all datasets contain broken sentences, with an average proportion of 30.28%. Previous studies have shown that broken sentences hinder semantic understanding and may interfere with model learning (Ahn et al., 2024). In our analysis, we observe that broken sentences often exhibit abnormal representations in embedding space and tend to lie farther from the cluster center. Examples of broken sentences are provided in Appendix A.2. This suggests that they behave similarly to outliers in clusters.

During training, RV-HATE computes the center of each cluster to select anchors. However, if clusters contain broken sentences (*e.g.*, those with typos or incomplete masking), they may act as outliers and degrade the quality of anchor selection. To address this issue, $\mathtt{M}_2$ applies the InterQuartile Range (IQR) method, a well-established statistical approach for outlier detection (Mramba et al., 2024). This method computes the upper-bound threshold based on the IQR of the distances from the cluster center (Appendix G). Data points exceeding this threshold are removed, and the cluster center is recalculated. This outlier removal process reduces the influence of broken sentences and improves the overall quality of the datasets. The proportion of removed outliers appears in Appendix C.3.

3.4 Using Hard Negative Samples ($\mathtt{M}_3$)

Hate speech datasets inherently contain a high level of noise due to the subjective nature of human annotation. Annotators often interpret hate speech differently because of personal bias, background knowledge or contextual perception (Khurana et al., 2025).

We randomly select 500 samples from each dataset and use gpt-4.1 to quantify the proportion of mislabeled instances. On average, 18.24% of the data across datasets is identified as mislabeled (Appendix B.2). Such labeling inconsistencies negatively affect model training, by introducing ambiguous decision boundaries and decreasing classification performance (Ahn et al., 2024). Therefore, we propose $\mathtt{M}_3$ to identify and leverage hard negatives near the decision boundary for better discrimination. Hard negative samples include data points with high cosine similarity to the anchor yet belong to a different class. Additionally, false positive samples with high model confidence serve as hard negatives. Unlike standard contrastive learning, where only in-batch negative samples are selected, we employ a queue to store hard negative samples from multiple batches (Kim et al., 2024). This allows the model to capture challenging negatives beyond the current batch and extends the selection process across a broader range of data. Consequently, the model learns more refined decision boundaries and enhances classification performance.

3.5 Reinforcement Learning-Guided Soft Voting

We propose a reinforcement learning-guided soft voting mechanism for effective detection of implicit hate speech across datasets. This approach enables an adaptive ensemble strategy by dynamically learning the weight to assign to each module depending on dataset-specific characteristics. We independently train the four classifiers f_k ($k \in$

$\{0, 1, 2, 3\}$) based on each module designed to capture complementary aspects of implicit hate speech. Each classifier f_k outputs a logit vector

$$z_{k,i} = [z_{k,i}^{(0)}, z_{k,i}^{(1)}],$$

where i indexes an input and $(0), (1)$ denote non-hate and hate classes, respectively.

Soft Voting. We aggregate the predictions of the four classifiers by computing a weighted average of their logits. The ensemble logit for each class $h \in \{0, 1\}$ is given by

$$Z_i^{(h)} = \sum_{k=0}^{3} w_k \cdot z_{k,i}^{(h)},$$

where w_k denotes the reinforcement learning-optimized vector (module-weights). We predict the final label by selecting the class with the highest ensemble logit.

$$\hat{y}_i = \arg \max_{h \in \{0,1\}} Z_i^{(h)}.$$

This approach enables the ensemble to focus on models that perform more reliably under the specific characteristics of the input hate speech data.

Reinforcement Learning. We formulate the model weights assignment problem as a sequential decision-making task, where a policy network $\pi_\theta(\mathbf{w}|s)$ generates the weight vector $\mathbf{w} = [w_0, w_1, w_2, w_3]$ conditioned on the current state s. After sampling $\mathbf{w}$, we apply soft voting and evaluate the resulting prediction $\hat{y}$ on a validation set. The F1-based reward r guides policy optimization. We adopt Proximal Policy Optimization (PPO) (Schulman et al., 2017) to train the policy network by maximizing the following objective:

$$r_t(\theta) = \frac{\pi_\theta(\mathbf{w}_t|s_t)}{\pi_{\theta_{old}}(\mathbf{w}_t|s_t)},$$

$$L(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)],$$

where $r_t(\theta)$ is the probability ratio between the new and old policies at timestep t, and $\hat{A}_t$ is the estimated advantage, measuring how much better the selected action $\mathbf{w}_t$ performs than a baseline. The expectation $\mathbb{E}_t$ is taken over timesteps in a batch of episodes. Each timestep corresponds to a forward pass where the policy samples a weight vector $\mathbf{w}_t$ and receives a reward. PPO optimizes $L(\theta)$ over these steps, with clipping applied to the probability ratio $r_t(\theta)$ to prevent large policy updates when $r_t(\theta)$ deviates from 1, ensuring stable convergence.

ϵ is a clipping parameter that limits the policy update step. We initialize all weights $w_k = 0.25$ and constrain them during training to remain positive and sum to 1, while the policy network learns dataset-specific weight configurations that maximize ensemble performance.

4 Experimental Results

4.1 Datasets

We conduct experiments on five hate speech datasets—IHC, SBIC, DYNA, Hateval, and Toxigen—that cover a broad spectrum of characteristics for a comprehensive evaluation. The detailed settings and dataset explanations are provided in Appendix C.1, C.2.

4.2 Baselines

We compare our approach with four baseline methods: **CE** is a general approach for hate speech detection based on cross-entropy loss. **SCL** (Khosla et al., 2020) is a supervised contrastive learning that uses labels to bring representations of the same class closer and push apart representations of different classes. **SharedCon** (Ahn et al., 2024) is the current SOTA method in implicit hate speech detection. This method uses the data closest to the center of each cluster as its anchor instead of explicit implications. **LAHN** (Kim et al., 2024) uses hard negative samples in contrastive learning. Hard negatives are data samples that are close to an anchor but have different labels. LAHN illustrates the importance of hard negative samples.

4.3 Implementation Details

For our experiments, we use a pre-trained language model BERT-base-uncased (110M) (Devlin et al., 2019) as the base model and Sim-CSE[1] (Gao et al., 2021) as a text embedding model. We train the models on each of the five datasets for 6 epochs with NVIDIA RTX 4090. For hyperparameter setting, we select the learning rate from {2e-5, 3e-5}, the temperature τ from {0.3}, λ from {0.5, 0.75}, the number of clusters from {20, 75, 125}. We conduct 10,000 steps for reinforcement learning. All experiments are executed with three different random seeds. We report the average score of macro-F1 because it is more appropriate for evaluating performance on imbalanced hate speech datasets.

---

[1] princeton-nlp/unsup-simcse-bert-base-uncased

Models	Datasets IHC	 SBIC	 DYNA	 Hateval	 Toxigen	Average
CE	77.70	83.80	78.80	81.11	90.06	82.29
SCL (Khosla et al., 2020)	77.81	82.92	80.39	81.28	90.75	82.63
SharedCon (Ahn et al., 2024)	78.50	84.30	79.10	80.24	91.21	82.67
LAHN (Kim et al., 2024)	78.40	83.98	79.64	80.42	90.42	82.57
RV-HATE (*Ours*)	**79.07**	**84.62**	**81.82**	**83.44**	**93.41**	**84.47**

Table 1: Performance comparison with four baseline methods. We report the macro-F1 scores averaged over three runs with different random seeds. The **bold** text indicates the best performance. RV-HATE shows the best performance across all datasets.

	IHC	SBIC	DYNA	Hateval	Toxigen	Average
RV-HATE (*combined modules*)	77.32 ±0.51	81.31 ±1.26	76.50 ±4.95	81.26 ±0.86	92.02 ±0.62	81.64 ±1.64
RV-HATE (*equal weights*)	78.58 ±0.58	84.06 ±0.10	81.07 ±0.29	82.52 ±0.25	92.69 ±0.44	83.78 ±0.33
RV-HATE (ℓ_2)	78.90 ±0.35	82.95 ±0.25	81.64 ±0.47	83.19 ±0.49	93.36 ±0.28	84.01 ±0.37
RV-HATE (*ours*)	**79.07** ±0.15	**84.62** ±0.23	**81.82** ±0.22	**83.44** ±0.10	**93.41** ±0.21	**84.47** ±0.18

Table 2: Performance comparison of RV-HATE and its variants. The *combined modules* (Sec. 5.1) refer to a single model trained using all modules without any voting mechanism, *equal weights* (Sec. 5.2) indicates the voting performance when each module is assigned an equal weight of 0.25, and ℓ_2 denotes the base model using the Euclidean distance instead of cosine similarity (Sec. 5.3). We report the macro-F1 scores averaged over three runs with different random seeds. The **bold** text indicates the best performance.

4.4 Experimental Results

On the Hateval dataset (Table 1), contrastive learning methods such as SharedCon and LAHN underperform compared to the cross-entropy (CE) baseline (81.11%). In contrast, RV-HATE achieves 83.44%, outperforming the CE baseline by 2.33%p. On Toxigen, RV-HATE outperforms SharedCon by 2.2%p. RV-HATE achieves SOTA performance across diverse conditions, outperforming the prior leading model, SharedCon, by an average of 1.8%. These results demonstrate that the proposed modules and reinforcement learning-based weighting effectively address dataset-specific characteristics. Section 5.4 explains in detail how each module contributes to performance improvements on individual datasets and Appendix D reports the weight values assigned to each module and dataset.

5 Analysis

5.1 The Integration of all Modules

We train a single model that integrates all four modules ($\mathtt{M}_0$, $\mathtt{M}_1$, $\mathtt{M}_2$ and $\mathtt{M}_3$) to examine the impact of the voting mechanism, which is called 'combined modules'. As shown in Table 2, RV-HATE (*combined modules*) consistently exhibit worse performance compared to RV-HATE (*ours*)—average decrease of 2.83%p. This performance drop indicates that jointly training on all modules leads to a loss of specialization, reducing the ability of the model to adapt to data-specific characteristics. In contrast, RV-HATE (*ours*) retains its specialization, allowing the voting mechanism to leverage diverse perspectives. These results highlight that preserving modular specialization and leveraging their complementary views is more effective than combining them into a unified model.

5.2 The Use of Reinforced Voting Mechanism

We analyze the impact of reinforcement learning on the voting mechanism in RV-HATE. We evaluate its effectiveness by conducting an experiment using fixed weights $[0.25, 0.25, 0.25, 0.25]$ without reinforcement learning-based weights (RV-HATE (*equal weights*)). In contrast, RV-HATE (*ours*) learns dataset-specific optimal weights through reinforcement learning, enabling the voting mechanism to reflect the contribution of each module for a given dataset. When comparing the two settings in Table 2, we observe that the approach using optimized weights (RV-HATE (*ours*)) achieves an average improvement of 0.68%p. The weights are optimized according to the characteristics of each dataset, presented in Appendix D. This result demonstrates that reinforcement learning effectively identifies the optimal combination of module contributions optimized to each dataset. However,

	IHC	SBIC	DYNA	Hateval	Toxigen	Average
$\texttt{M}_0$	77.26	83.36	80.52	81.02	91.25	82.68
$\texttt{M}_1$	77.53	82.94	79.51	81.63	90.55	82.43
$\texttt{M}_2$	77.64	83.11	80.26	81.45	92.01	82.89
$\texttt{M}_3$	77.42	83.28	79.87	81.78	92.63	83.00
RV-HATE	**79.07**	**84.62**	**81.82**	**83.44**	**93.41**	**84.47**
- $\texttt{M}_0$	79.04 (-0.03)	84.28 (-0.34)	81.15 (-0.67)	83.20 (-0.24)	93.16 (-0.25)	84.17
- $\texttt{M}_1$	78.37 (-0.70)	84.50 (-0.12)	81.56 (-0.26)	83.04 (-0.40)	92.99 (-0.42)	84.09
- $\texttt{M}_2$	78.60 (-0.47)	84.36 (-0.26)	81.66 (-0.16)	83.01 (-0.43)	93.16 (-0.25)	84.15
- $\texttt{M}_3$	78.79 (-0.28)	84.24 (-0.38)	81.17 (-0.65)	82.88 (-0.56)	92.87 (-0.54)	83.99

Table 3: Ablation study results for RV-HATE. The table shows the performance of each module and the impact of excluding each module. Each result represents the average macro-F1 score of three runs with different random seeds. The value in parentheses indicates the performance change compared to RV-HATE, and **bold** text indicates the best performance. The model achieves strong performance when utilizing all modules jointly.

Type	IHC	SBIC	DYNA	Hateval	Toxigen
entity-tagged ($\texttt{M}_1$)	67.83	65.04	27.56	48.44	48.53
outlier-removed ($\texttt{M}_2$)	0.59	0.44	0.38	0.69	0.31

Table 4: Dataset statistics for $\texttt{M}_1$ and $\texttt{M}_2$. The table shows the ratio of entity-tagged data and outlier-removed data in each dataset after applying the $\texttt{M}_1$, $\texttt{M}_2$ modules.

simple voting with equal weights fails to capture dataset-specific features. By adaptively balancing the contributions of specialized modules, reinforcement learning enables the voting mechanism to leverage dataset-specific expertise and achieve improved performance.

5.3 The Impact of Cosine Similarity

We conduct an experiment to examine the impact of using cosine similarity in module training. Specifically, we compare RV-HATE (*ours*) that uses cosine similarity for the modules and RV-HATE (ℓ_2) that uses Euclidean distance. As shown in Table 2, RV-HATE (*ours*) achieves an average improvement of 0.46%p and shows a lower standard deviation across datasets. These results validate the effectiveness of employing cosine similarity over Euclidean distance for training the modules.

5.4 Ablation Study

We analyze the performance of RV-HATE and its individual modules. Table 3 presents the performances of each module. Notably, some modules perform worse than the baseline ($\texttt{M}_0$). This suggests that individual modules may be biased as they consider dataset-specific characteristics. In contrast, RV-HATE achieves a higher performance than $\texttt{M}_0$ alone, as the voting mechanism mitigates such biases and balances the variance across modules. By incorporating diverse dataset-specific characteristics, RV-HATE can achieve enhanced overall performance.

Module 1 ($\texttt{M}_1$). Excluding $\texttt{M}_1$ from IHC causes the largest performance drop (-0.7%p). Table 4 supports this finding by showing that IHC contains the highest proportion of NER-tagged data. It indicates that implicit hate speech often relies on subtle target references that help distinguish it from merely offensive content. SBIC also has a relatively high proportion of tagged data. However, since SBIC is an offensive dataset, excluding $\texttt{M}_1$ results in only a marginal performance drop (-0.12%p). These results support our hypothesis by demonstrating that datasets with ambiguous boundaries between offensive and hate speech rely on $\texttt{M}_1$ to capture target-specific cues. $\texttt{M}_1$ effectively distinguishes offensive expressions from genuine hate speech and improves the overall performance. Additionally, we analyze the proportion of implicit hate speech in each dataset. As shown in Figure 7 of Appendix I, IHC exhibits the highest proportion of implicit hate speech data, which aligns with its reliance on $\texttt{M}_1$.

Module 2 ($\texttt{M}_2$). Excluding $\texttt{M}_2$ from IHC and Hateval results in a performance drop of 0.47%p, 0.43%p, respectively. Consistently, Table 4 shows

Model	Number of Parameters	Training Time	Inference Latency
SharedCon	110M	1h	0.5-1.5ms
LAHN	110M	1h	0.5-1.5ms
RV-HATE	110M * 4 + 4,085 (PPO)	1h * 4 + 5-10m (PPO)	0.5-1.5ms * 4

Table 5: Comparison of model complexity, training time, and inference latency for RV-HATE and encoder-based baselines. All modules use BERT-base encoders. Training time is measured on the IHC dataset.

that the IHC and Hateval have the highest proportions of removed data. This performance decline indicates that $\mathtt{M}_2$ plays a crucial role in mitigating the impact of outliers. These findings demonstrate the effectiveness of $\mathtt{M}_2$ in handling noisy data and improving detection in datasets with high levels of textual noise. Additionally, we analyze the distribution of broken sentences. As shown in Figure 7 of Appendix I, IHC exhibits the highest proportion of broken sentences, whereas DYNA shows the lowest.

Module 3 ($\mathtt{M}_3$). Removing $\mathtt{M}_3$ from RV-HATE has the greatest impact across all datasets. Moreover, the standalone averaged performance of the $\mathtt{M}_3$ module surpasses that of the other modules. We examined the embedding representations of each dataset using t-SNE (Figure 6 in Appendix F) to validate whether $\mathtt{M}_3$ indeed clarifies the decision boundary. The visualizations show that the decision boundary becomes more clearly separated compared to the baseline module. More detailed results are presented in Appendix F. These findings confirm that $\mathtt{M}_3$ effectively enhances the separability of hard negatives near the decision boundary, thereby improving the ability to detect subtle and ambiguous instances.

5.5 Computational Efficiency

We analyze the computational cost of RV-HATE and compare it with existing encoder-based baselines in Table 5. All modules use BERT-base encoders with 100M parameters. RV-HATE includes four modules, and each module is trained independently. This allows parallel training across multiple GPUs and keeps the training time comparable to the baseline.

The reinforcement learning component introduces only a small additional cost. The total number of parameters for PPO-based reinforcement learning is 4,805, which is negligible compared to the encoder models. The reinforcement learning stage is applied once during training and takes about 5 to 10 minutes. Detailed parameter configurations are provided in Appendix H.

At inference time, RV-HATE requires four forward passes of BERT-base. This leads to a linear increase in latency compared to single-encoder models. However, the method does not use additional decoding or iterative reasoning, and the computation remains a simple forward-pass ensemble. RV-HATE increases computation in proportion to the number of modules, but maintains practical efficiency through parallel training and minimal reinforcement learning overhead.

6 Error Analysis

We analyze the false positive or false negative samples from each dataset to better understand why the RV-HATE fails to detect them correctly. We categorize these error samples into three different types: `type1` refers to broken instances that contain grammatical errors, typographical errors or special characters; `type2` denotes ambiguous instances that are difficult to clearly classify as hate or not-hate; `type3` corresponds to mislabeled instances; and `type4` includes all remaining cases that cannot be assigned to any of the aforementioned categories. Two experts annotate the error samples according to these three types.

Table 6 reports the distribution of error types for each dataset. IHC shows a high proportion of `type2` errors (45.15%), as the dataset consists of implicit and ambiguous instances. Error samples from IHC also exhibit a high proportion of ambiguous instances. SBIC exhibits a comparable proportion of `type2` (33.38%) and `type3` errors (23.35%). Since SBIC is an offensive language dataset, offensive expressions differ in definitions from hate speech or are subject to annotator bias, resulting in both ambiguous `type2` and mislabeled `type3` errors. In the case of DYNA, the majority of errors belong to `type2` (33.20%) and `type3` (44.76%), with `type3` accounting for the largest proportion. This result suggests that annotation inconsistencies play a significant role in DYNA errors. In particular, a high proportion of

Dataset	Error Type	Ratio	Example	label
IHC	`type1`	23.85%	“this is so unislamic – no compulsion in religion”	non-hate
	`type2`	45.15%	whine all you want davey. white id politics is here to stay.	non-hate
	`type3`	24.80%	explicitly huwhite	non-hate
	`type4`	6.20%		n/a
SBIC	`type1`	15.90%	# health... yo-ho-ho and a bottle of # rum!	non-hate
	`type2`	33.38%	any mens rights groups / people here in sf?	hate
	`type3`	23.35%	these days bitches be just letting anyone hit raw!!	not-hate
	`type4`	27.36%		n/a
DYNA	`type1`	5.31%	wooooooooooman = trrassssshhh	hate
	`type2`	33.20%	i see no difference between toffs and entitled children	not-hate
	`type3`	44.76%	men are such a fucking stupid knob heads!	not-hate
	`type4`	16.73%		n/a
Hateval	`type1`	25.35%	correction - hugh if true. # makemeasandwich	hate
	`type2`	27.70%	when you join a relationship women leave this part out # yesallmen	hate
	`type3`	41.78%	u a bitch	hate
	`type4`	5.16%		n/a
Toxigen	`type2`	50.00%	go back to china.	hate
	`type3`	21.05%	“it is not ok for me to be homophobic, but it is ok for me to be homophobic”	hate
	`type4`	28.95%		n/a

Table 6: Error analysis of RV-HATE, where `type1` is the case of broken sentences, `type2` is the case of ambiguous sentences that are difficult to classify even by human experts, `type3` is the case of clearly mislabeled, `type4` is the case of the rest. The ‘label’ column shows the ground-truth label. Note that there is no `type1` error in Toxigen.

mislabeled and ambiguous instances indicates that DYNA likely contains a variety of dynamic and context-dependent expressions. Hateval shows the greatest degree of annotator noise `type3`(41.78%), indicating a large number of mislabeled instances. Manual inspection further confirms that the dataset contains substantial label noise. In contrast, as a machine-generated dataset, Toxigen exhibits no typographical errors or broken sentences. However, `type2` errors account for 50% of the error samples, indicating that Toxigen includes many semantically ambiguous instances despite its clean text. Further analyses are provided in Appendix E.

This analysis highlights that ambiguous and mislabeled instances constitute a major source of performance degradation across datasets.

7 Conclusions

We have proposed RV-HATE, a reinforcement learning-based voting method for implicit hate speech detection that can capture the dataset-specific characteristics through the multi-module design. Each module is designed to address a particular aspect of the datasets. $\mathtt{M}_0$ focuses on improving contextual understanding through cosine similarity, $\mathtt{M}_1$ enhances the detection of implicit hate speech by incorporating hate target tagging, $\mathtt{M}_2$ removes outliers during training to preserve reliable data samples, and $\mathtt{M}_3$ provides a clear decision boundary. Our approach employs reinforcement learning to assign an adaptive weight to each module. By analyzing how each module contributes across different datasets, we have a better understanding of their characteristics. Thus, through the voting strategy, RV-HATE is able to adapt dataset-specific features better and achieves SOTA performance on multiple benchmarks.

Limitations

RV-HATE effectively captures the characteristics of implicit hate speech datasets and can decide the best module combination with the reinforced voting mechanism. In our experiments, $\mathtt{M}_1$ the target-tagging module does not consistently provide the same improvements on machine-generated samples, potentially due to style and distribution differences. Although these results do not diminish the overall utility of RV-HATE, they highlight an opportunity to explore more specialized strategies for artificially generated data. We believe that with further adaptation and refinement, RV-HATE's modular design could be extended to manage artificially generated text effectively and broaden its applicability in future work.

Ethical Consideration

Minimizing Exposure Risks Existing methods rely on annotations that human annotators directly label and explain the meaning of hateful sentences. On the other hand, our approach allows the model to learn from representative samples without requiring manually annotated implications. As a result, our method is expected to reduce the mental load on annotators and contribute to a more ethical data collection process.

Dataset-aware Hate Speech Detection Focusing solely on improving model generalization can overlook important dataset-specific characteristics. Our approach introduces a voting methodology that employs the unique hate speech patterns of each dataset to make optimal decisions. By capturing diverse forms of hate speech while respecting the contextual nuances of individual datasets, our method contributes to the development of a more reliable and context-aware hate speech detection system.

Risks and Potential Misuse Our detection capability is designed to identify hate speech effectively; however, there remains a possibility that it could be leveraged in unintended ways, such as generating new forms of hate speech. Addressing these risks requires careful monitoring of how the model is used and a critical assessment of its impact.

Acknowledgments

This research was supported by the KISA grant (RS-2025-02222626) and the AI Graduate School Program (RS-2020-II201361) funded by the Korean government.

References

Hyeseon Ahn, Youngwook Kim, Jungin Kim, and Yo-Sub Han. 2024. Sharedcon: Implicit hate speech detection using shared semantics. In *Findings of the Association for Computational Linguistics, ACL*, pages 10444–10455.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *2019 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics, NAACL*, pages 54–63.

Yoni Birman, Shaked Hindi, Gilad Katz, and Asaf Shabtai. 2022. Cost-effective ensemble models selection using deep reinforcement learning. *Information Fusion*, pages 133–148.

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, Inga Kartoziya, and Michael Granitzer. 2020. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In *International Conference on Language Resources and Evaluation, LREC*, pages 6193–6202.

Aida Mostafazadeh Davani, Mohammad Atari, Brendan Kennedy, and Morteza Dehghani. 2023. Hate speech classifiers learn normative social stereotypes. *The 61st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 300–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 345–363.

Paula Fortuna, Juan Soler-Company, and Leo Wanner. 2021. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, page 102524.

Yuqian Fu, Yuanheng Zhu, Jiajun Chai, Guojun Yin, Wei Lin, Qichao Zhang, and Dongbin Zhao. 2025. Rlae: Reinforcement learning-assisted ensemble for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 13463–13477.

Ankita Gandhi, Param Ahir, Kinjal Adhvaryu, Pooja Shah, Ritika Lohiya, Erik Cambria, Soujanya Poria, and Amir Hussain. 2024. Hate speech detection: A comprehensive review of recent works. *Expert Systems*, page e13562.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 6894–6910.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. 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, ACL*, pages 3309–3326.

Matthew Honnibal. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023, WWW*, pages 294–297.

Junbo Huang and Ricardo Usbeck. 2024. Revisiting supervised contrastive learning for microblog classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 15644–15653.

Tianming Jiang. 2025. Learn from failure: Causality-guided contrastive learning for generalizable implicit hate speech detection. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING*, pages 8858–8867.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *The Fortieth Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 18661–18673.

Urja Khurana, Eric T. Nalisnick, and Antske Fokkens. 2025. Defverify: Do hate speech models reflect their dataset's definition? In *Proceedings of the 31st International Conference on Computational Linguistics, COLING*, pages 4341–4358.

Do-Kyung Kim, Hyeseon Ahn, Youngwook Kim, and Yo-Sub Han. 2025. Analyzing offensive language dataset insights from training dynamics and human agreement level. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING*, pages 9780–9792.

Jaehoon Kim, Seungwan Jin, Sohyun Park, Someen Park, and Kyungsik Han. 2024. Label-aware hard negative sampling strategies with momentum contrastive learning for implicit hate speech detection. In *Findings of the Association for Computational Linguistics, ACL*, pages 16177–16188.

Youngwook Kim, Shinwoo Park, and Yo-Sub Han. 2022. Generalizable implicit hate speech detection using contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING*, pages 6667–6679.

Youngwook Kim, Shinwoo Park, Youngsoo Namgoong, and Yo-Sub Han. 2023. Conprompt: Pre-training a language model with machine-generated data for implicit hate speech detection. In *Findings of the Association for Computational Linguistics, EMNLP*, pages 10964–10980.

Ho-Suk Lee, Hong-Rae Lee, Jun-U Park, and Yo-Sub Han. 2018. An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems*, pages 22–31.

Nayeon Lee, Chani Jung, Junho Myung, Jiho Jin, José Camacho-Collados, Juho Kim, and Alice Oh. 2024. Exploring cross-cultural differences in english hate speech annotations: From dataset construction to analysis. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL*, pages 4205–4224.

Yejin Lee, Joonghyuk Hahn, Hyeseon Ahn, and Yo-Sub Han. 2025. AmpleHate: Amplifying the attention for versatile implicit hate detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 28862–28874.

Pablo Madriaza, Ghayda Hassan, Sébastien Brouillette-Alarie, Aoudou Njingouo Mounchingam, Loïc Durocher-Corfa, Eugene Borokhovski, David Pickup, and Sabrina Paillé. 2025. Exposure to hate in online and traditional media: A systematic review and meta-analysis of the impact of this exposure on individuals and communities. *Campbell Systematic Reviews*, page e70018.

Lazarus K Mramba, Xiang Liu, Kristian F Lynch, Jimin Yang, Carin Andrén Aronsson, Sandra Hummel, Jill M Norris, Suvi M Virtanen, Leena Hakola, Ulla M Uusitalo, et al. 2024. Detecting potential outliers in longitudinal data with time-dependent covariates. *European Journal of Clinical Nutrition*, pages 344–350.

Nicolás Benjamín Ocampo, Ekaterina Sviridova, Elena Cabrio, and Serena Villata. 2023. An in-depth analysis of implicit and subtle hate speech messages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, pages 1989–2005.

Someen Park, Jaehoon Kim, Seungwan Jin, Sohyun Park, and Kyungsik Han. 2024. Predict: Multi-agent-based debate simulation for generalized hate speech detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 20963–20987.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 477–523.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 5477–5490.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347*.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 1667–1682.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 78–84.

Mathias Weber, Christina Viehmann, Marc Ziegele, and Christian Schemer. 2020. Online hate does not stay online - how implicit and explicit attitudes mediate the effect of civil negativity and hate in user comments on prosocial behavior. *Computers in human behavior*, page 106192.

A Sentence Examples

A.1 Examples of Hate without Target

Table 7 shows some examples of misannotated sentences in hate speech datasets. While hate speech is generally defined by the presence of an explicit hate target, these examples lack such a target and are appropriately categorized as offensive language rather than hate speech. Such inconsistencies in annotation may lead to confusion during model training and potentially degrade classification performance.

A.2 Examples of Broken Sentences

A broken sentence refers to a sentence which lacks essential grammatical components. Table 8 represents examples of broken sentences from the SBIC dataset. In the SBIC dataset, numerous instances exhibit such issues, further compounded by the presence of non-standard characters and special symbols, which collectively hinder accurate interpretation and analysis.

A.3 Examples of Mislabeled Samples

Table 9 provides examples of mislabeled instances across datasets. These instances clearly constitute hate speech, but they are incorrectly labeled as not-hate. Such mislabeled data can negatively impact model training by introducing noise into the learning process.

A.4 Difference Between Euclidean Distance and Cosine Similarity

The followings are examples from all dataset to illustrate the differences between Euclidean distance and cosine similarity in measuring sentence relationships. For each anchor post, we provide the sample with the smallest Euclidean distance and the sample with the highest cosine similarity. The results demonstrate that the sample with the highest cosine similarity better preserves semantic information, making it a more suitable candidate as an anchor sentence.

- **IHC:** " rt "" : stop the white genocide in south africa ! (video) via ##whiteresister "" "" """
    - **Highest Cosine Similarity 1:** the genocide of white america : 1980-2010 | read more : via," "" : the white genocide continues : white race now minority in london via "" "" """"" rt "" : stop the white genocide in south africa ! (video) via ##whiteresister "" "" """,the genocide of white america : 1980-2010 | read more : via
    - **Smallest Euclidean Distance 1:** " "" : the white genocide continues : white race now minority in london via "" "" """
- **SBIC:** I'm strongly opposed to having ‘Harriet Tubman‘ on the $20 bill. What use would I have for a piece of paper worth $12?
    - **Highest Cosine Similarity 2:** @JStac825 @ThatAngryNigga these nig**s must be deaf. When's the last time you heard a Roots song on the radio? Quest don't get bit**es!
    - **Smallest Euclidean Distance 2:** Roast beef sandwiches are ruined for me I saw someone eating a roast beef sandwich today and all I could think about was a stretched out vag**al flaps destroyed by the c**k carousel.

Dataset	Sentence
IHC	when you drab the swamp in sacramento!
	according to one person i communicated with-why cry over as bunch of cells, would you cry over a tumor
	one monkey don't stop no show
Hateval	When it's my birthday but you're the one that's getting spanked
	bitch I was fuckn up till 4:30 but your hoe ass didn't text me back.
	He can choke and so does his bitch hoe
DYNA	Rape should be legal
	Something tells me this woman doesnt have a stable and successful life.
	our country is broken. It's basically free of opportunity now, it's overrun by animals and we've lost all control of our institutions. I don't want to stay but I cant exaxtly leave,
Toxigen	there is no difference in the way they act, they are all the same, they all act the same way, they are all the same
	no matter how much money you give them, they'll never be as good as us and so on and so on and so on
	i dont care if you are in pain or not, i dont want to hear about it

Table 7: Examples of sentences that are classified as hate speech but are actually offensive language. These samples are drawn from the IHC, Hateval, DYNA, and Toxigen datasets. The SBIC dataset is excluded as it is basically an offensive language dataset.

Class	Sentence
Offensive	S/O to this hoe tonight ol hating ass http://t.co/HIt4475tyZ
	RT @gioisblack: Goodbye you faggot http://t.co/oJKZjkgVXL
	RT @TyrinFlyGuy: Niggas fufu & bitches is too
Not Offensive	##redneck tow truck. ##johndeere ##4455 ##america https://t.co/iFInqwVRad
	RT @k_ree2: &##8220;@Trentdamanz: These hoes are loyal&##8221; plot twist
	Yasss brownies & ice cream &##128523; http://t.co/8qLa3JsglG

Table 8: Examples of broken sentences from the SBIC dataset, categorized by class. These sentences are often found as cluster outliers, which can negatively impact model performance. Mitigating this issue, we employ $\mathtt{M}_2$ to remove such outliers.

Class	Sentence	Expected Class
Hate	i'm not interested in helping you work your sickness out, you need a shrink and a pile of meds, still probably wouldn't help you. get help	Not Hate
	Hungary's New laws target Muslim migrants "to preserve Christian culture of Europe" - Free Speech Time -	
	Women are perfect. Now repeat that until you believe it.	
	It is a total disgrace when tax-payer dollars can go to promoting such degenerate filth.	
	i dont care if they are here legally or not.	
Not Hate	Can you finally see it now? Its not American vs American. Never was and still isn't! This is Americans vs Transplant-Sleeper-Cell-Capitalists ("Illegal Fucking Corporations')	Hate
	who love illegal aliens. You all have more blood on your hands. You have betrayed America. ##ILLEGALimmigrants	
	we need 2 open a window of real knowledge for born muslims to re-learn & convert . at least go secular	
	Hanging out with guys does not make you a hoe	

Table 9: Examples of mislabeled sentences across different classes, sourced from the IHC, SBIC, Hateval, DYNA and Toxigen datasets. These mislabeled instances can confuse the model, highlighting the importance of effectively leveraging hard negatives, which we address using M4.

- **DYNA:** Those black women look like eggplants
  - **Highest Cosine Similarity 3:** What is being presented here is a photograph of a species of "African-aero-ape". Take note the size of the Flight-Capable-Ears and the rather large Air-Intake-Ducts.
  - **Smallest Euclidean Distance 3:** Haha, we'd just watch her whining and screaming in combat. She won't ever make it. It should be left to the men. Women display too much ego nowadays. They think

that it is their right to replace men. Never gonna happen.

- **Hateval:** a never go respect a who post nothing but have naked pics on here or showing there ass you go be looked at as a rat h**|Im gonna treat women like a hole
    - **Highest Cosine Similarity 3:** Im gonna treat women like a hole
    - **Smallest Euclidean Distance 3:** ban muslims banislam sendthemback to their shitholes closetheborders
- **Toxigen:** "ha i post this picture of a black guy on my facebook commenting on how ugly he was he looked like an ape and the
    - **Highest Cosine Similarity 3:** no matter how much money you make, you'll never be as rich as a white person, and youll never be as smart as a white person
    - **Smallest Euclidean Distance 3:** hurr if ur chinese change ur name to a white onen my chinese friends are trying to stick up for their stereotype so hard right

B Prompts Used for ChatGPT

B.1 Prompt for Identifying Hate Speech Without Given Targets

We view the distinction between *hate speech* (which explicitly targets an individual or group) and *offensive speech* (which may be abusive but lacks a specific target) as a key challenge in hate speech detection. While many datasets define hate speech based on the presence of a clear target, real-world cases are often more ambiguous. Some hateful statements do not explicitly reference a group or individual, making it difficult for models to differentiate between hate speech and general offensive language.

Therefore, we use a determination prompt (Figure 2) that determines whether a given hate speech sentence includes an explicit target to analyze the prevalence of ambiguous cases in each dataset.

This analysis provides insight into the limitations of existing models and datasets in handling implicit hate speech. By identifying cases where models fail to make this distinction, we highlight the need for methods that better capture the nuances of hateful expression.

B.2 Prompt for Identifying Mislabeled Data

Labels in hate speech datasets often contain annotation errors due to the subjective nature of the task and inconsistencies in human judgments. Some samples may be incorrectly labeled as hate speech despite lacking explicit hateful intent, while others may be misclassified as non-hate despite containing discriminatory content. We design a verification prompt (Figure 3) that assesses whether a given post's label correctly reflects its content to address this issue.

Identifying mislabeled instances allows us to analyze patterns in label inconsistencies and provides insights into common annotation biases, which can inform future dataset construction and model training strategies.

B.3 NER Tagging

We use a Named Entity Recognition (NER) tagging approach to systematically analyze the target entities in hate speech. The objective of this process is to identify words in a given sentence that explicitly denote a specific group or organization targeted by hate speech. Since implicit hate speech often relies on subtle cues, correctly recognizing these entities is crucial for understanding the nature of hateful expressions. Figure 4 presents the prompt used to guide this tagging process. A single tag [TARGET] is used throughout. If no explicit target exists in the sentence, the model returns the sentence without modification.

By applying our NER-based tagging mechanism, we aim to make latent target information explicit. This allows for a clearer identification of the specific groups or organizations referenced in hate speech.

C Dataset Statistics

C.1 Used Dataset

We used five datasets. **IHC** (ElSherief et al., 2021) is a benchmark for implicit hate speech detection. It includes 22,584 tweets labeled with corresponding natural language explanations (*i.e.*, implications). **SBIC** (Sap et al., 2020) provides over 150,000 structured annotations of social media posts that capture the implications of social biases and stereotypes, including their offensiveness, intent, and targeted groups. **DYNA** (Vidgen et al., 2021) is a hate speech dataset created through a human-and-model-in-the-loop process, incorporating adversarial perturbations to improve the robustness of hate speech

You are a hate speech target verification GPT. When given a post, follow the instructions below:
1. The provided posts are guaranteed to be labeled as hate speech.
2. Hate speech is a form of abusive language that specifically targets individuals or groups based on characteristics such as race, gender, religion, or ethnicity.
3. Your task is to determine whether the post contains a target-a specific individual or group being attacked.
4. If the post includes a target (e.g., a group or individual being insulted, threatened, or dehumanized), classify it as "target O".
5. If the post is abusive but lacks a specific target, classify it as "target X".
6. When in doubt, classify as "target O".

Example 1
Given post: "Immigrants are ruining this country. They should all go back."
Your response: target O

Example 2
Given post: "This is absolutely disgusting. People like this have no place here."
Your response: target X

Remember: Your response should consist of only a single phrase: either "target O" or "target X". No additional explanation or extra words are allowed.

Post: `[SENTENCE]`

Figure 2: Prompt used for identifying whether a hate speech post contains an explicit target.

detection models. **Hateval** (Basile et al., 2019) contains 13,000 English and 6,600 Spanish tweets annotated for hate speech targeting immigrants and women. We use only the English portion in our experiments. **Toxigen** (Hartvigsen et al., 2022) is a machine-generated dataset of toxic and benign statements about 13 minority groups, designed to improve implicit hate speech detection.

C.2 Dataset Split Overview

We split the dataset into train, validation, and test sets in an 8:1:1 ratio. The augmented train set was generated by adding target tags only to hate-labeled data, and it only used in $\mathtt{M}_2$.

C.3 Outlier Removal Ratios

Table 11 summarizes the number and proportion of samples removed by $\mathtt{M}_2$ in each dataset. Hateval show the highest removal rate, whereas Toxigen has the lowest. Although the absolute count of discarded samples is modest, this outlier-removal step impacts RV-Hate's performance by mitigating the influence of noisy data.

D Weights

Table 12 gives the average reinforcement learning-based voting weights for modules $\mathtt{M}_0$–$\mathtt{M}_3$ across three random seeds. For IHC, although $\mathtt{M}_0$ retains a minor share of the vote, the model leans most heavily on hard negative sampling, then entity tagging, with the least emphasis on removing outliers. SBIC, Hateval and DYNA concentrate their weight on $\mathtt{M}_0$, with other modules sharing the rest of the weight moderately. On the other hand, Toxigen prioritizes hard negative sampling while almost disregarding entity tagging.

E Error Analysis

We provide error examples in Table 6 and confusion matrices in Figure 5. Note that Toxigen does not contain any instances categorized as `type1` error. Figure 5 presents a comparison between the confusion matrices of SharedCon, which serves as both one of our baselines and the previous SOTA model, and our proposed method RV-HATE. In Hateval, the majority of misclassified examples by RV-HATE fall under `type3` errors. Upon manual

You are a hate speech verification GPT. When given a post and its associated label, follow the instructions below:
1. The provided posts are samples from a hate speech dataset.
2. Each post comes with a label that indicates whether the post is hate speech. 1 indicates hate, 0 indicates non-hate.
3. Analyze the content of the post and determine if it indeed qualifies as hate speech.
4. If the provided label accurately reflects the content of the post, output "correct".
5. If the label does not match the content of the post, output "incorrect".

Example 1
Given post: "I can't stand those immigrants who take our jobs."
Label: hate speech
Your response: correct

Example 2
Given post: "I love everyone regardless of their background."
Label: hate speech
Your response: incorrect

Remember: Your response should consist of only a single word, either "correct" or "incorrect". No additional explanation or extra words are allowed.

Post: `[SENTENCE]`
Label: `[LABEL]`

Figure 3: Prompt for verifying labels of the datasets we used.

You are a hate speech token tagging GPT. When given a sentence, follow the instructions below:
1. The provided sentences are examples of hate speech.
2. To qualify as hate speech, the sentence must insult or discriminate against a specific group or organization.
3. Identify the words in the sentence that represent the specific group or organization.
4. Tag only the key words that determine the hate speech with the appropriate category.

Example
Given sentence: immigrants wouldn't ask that question if his family was murdered by savages
Your response: [TARGET] immigrants wouldn't ask that question if his family was murdered by savages

You have to use only the '[TARGET]' tag (if there exist other tags, you replace the tag to '[TARGET]'). As shown in the example, no additional explanation is necessary in your response. There may be multiple key words representing the hate speech target. If there is no specific target, output the original sentence as is.

Post: `[Sentence]`

Figure 4: Prompt for NER tagging

inspection, we observe several cases with highly similar contexts where one instance is labeled as hate and another as non-hate, suggesting the presence of annotation noise in the dataset. For Toxigen,

Dataset	Train set	Augmented Train Set	Valid set	Test set
IHC	14,932	18,796	1,867	1,867
SBIC	35,504	45,290	4,673	4,698
DYNA	33,004	44,427	4,125	4,126
Hateval	10,384	13,319	1,298	1,298
Toxigen	5,420	6,704	678	678

Table 10: The statistical information of five datasets in our experiments.

	IHC	SBIC	DYNA	Hateval	Toxigen
Outlier	66	156	127	72	17
Total	11,199	35,504	33,004	10,384	5,420
Ratio (%)	0.59	0.44	0.38	0.69	0.31

Table 11: The ratio of the removed outlier data

	$\mathbf{M}_0$	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_4$
IHC	0.191	0.258	0.167	0.357
SBIC	0.357	0.252	0.210	0.181
DYNA	0.330	0.179	0.265	0.225
Hateval	0.327	0.187	0.248	0.238
Toxigen	0.227	0.001	0.314	0.459

Table 12: The average reinforced voting weights for each dataset across three random seeds. The values are rounded to four decimal places.

the proportion of broken or masked sentences was notably low, likely due to its synthetic, machine-generated nature. Nevertheless, a substantial number of examples remained ambiguous; over 50% of the misclassified samples are categorized as `type2` errors.

F Embeddings

Figure 6 visualizes the impact of the hard negative sampling module (`M3`) on the embedding space via t-SNE projections for the SBIC, Hateval and Toxigen datasets. In the absence of `M3` (top row), embeddings form local clusters but fail to exhibit a clear separation between 'non-hate' class and 'hate' class instances. Once `M3` is applied (bottom row), these clusters persist and the two classes become distinctly partitioned, demonstrating that hard negative sampling sharpens the decision boundary in the representation space.

G IQR

The IQR is defined as the difference between the third quartile (Q_3) and the first quartile (Q_1). The upper bound threshold is calculated by adding 1.5 times the IQR to the Q_3.

$$IQR = Q_3 - Q_1. \quad (1)$$

$$upper\_bound = Q_3 + 1.5 \times IQR. \quad (2)$$

The resulting set of data points after removing outliers X includes only The resulting set of data points after removing outliers includes only points whose distances from the cluster center are less than the upper bound.

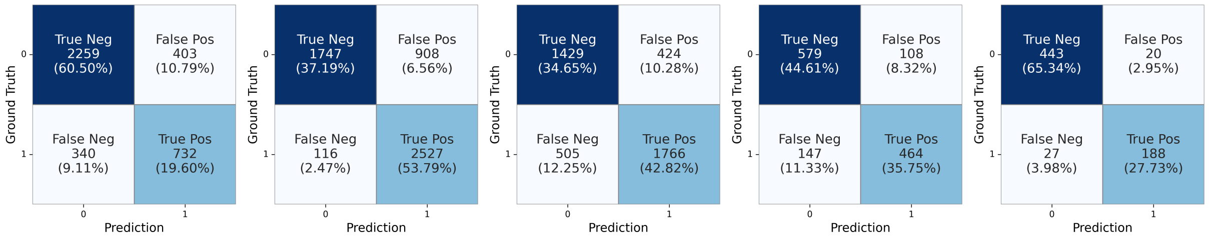
$$X = \{x | x < upper\_bound\}. \quad (3)$$

H Policy Network Architecture

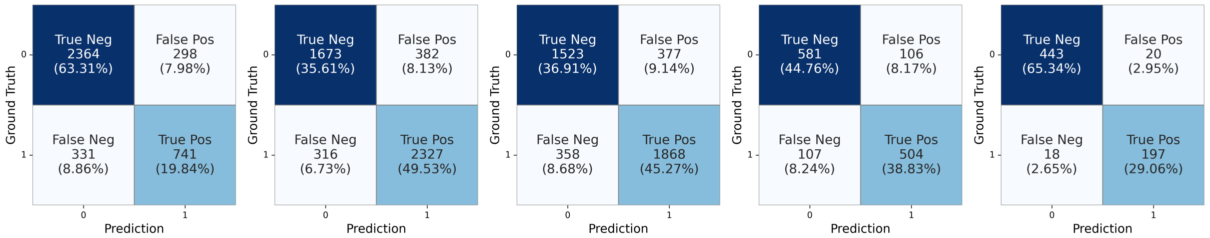
We describe the architecture of the policy network used for reinforcement learning in RV-HATE. The policy network takes a four-dimensional state as input and outputs a weight vector $w = [w_0, w_1, w_2, w_3]$, where each weight corresponds to a module in the voting process. The network consists of a shred feature extractor followed by separate actor and critic heads.

The shared feature extractor is implemented as a row-layer feedforward network with hidden dimension 64. Give an input state $s \in \mathbb{R}^4$, the hidden representation is computed through tow linear layers with nonlinear activation. This component contains 4,480 parameters.

The actor head maps the shared representation to a four-dimensional action space and contains 260 parameters. The critic head outputs a scalar value estimate and contains 65 parameters. In total, the policy network contains 4,805 parameters, which is negligible compared to the BERT-base encoders used in each module.

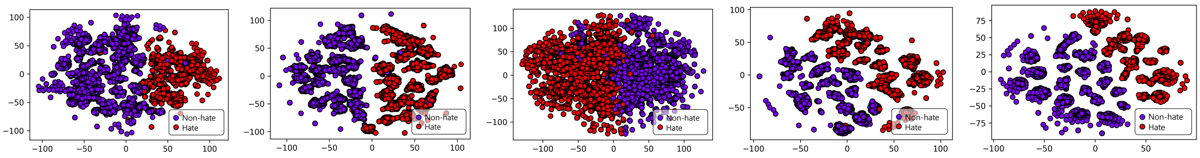


(a) IHC (SharedCon) (b) SBIC (SharedCon) (c) DYNA (SharedCon) (d) Hateval (SharedCon) (e) Toxigen (SharedCon)

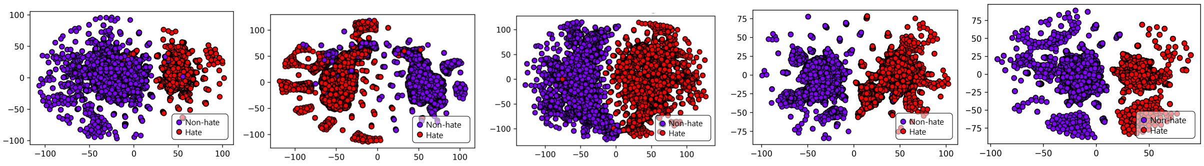


(e) IHC (RV-HATE) (f) SBIC (RV-HATE) (g) DYNA (RV-HATE) (h) Hateval (RV-HATE) (e) Toxigen (RV-HATE)

Figure 5: Confusion matrices of SharedCon (top row) and RV-HATE (bottom row) on the five hate-speech datasets (IHC, SBIC, DYNA, Hateval, and Toxigen). Each cell reports both the absolute count and the percentage of examples for true negatives, false positives, false negatives and true positives.



(a) IHC ($\mathtt{M}_0$) (b) SBIC ($\mathtt{M}_0$) (c) DYNA ($\mathtt{M}_0$) (d) Hateval ($\mathtt{M}_0$) (e) Toxigen ($\mathtt{M}_0$)

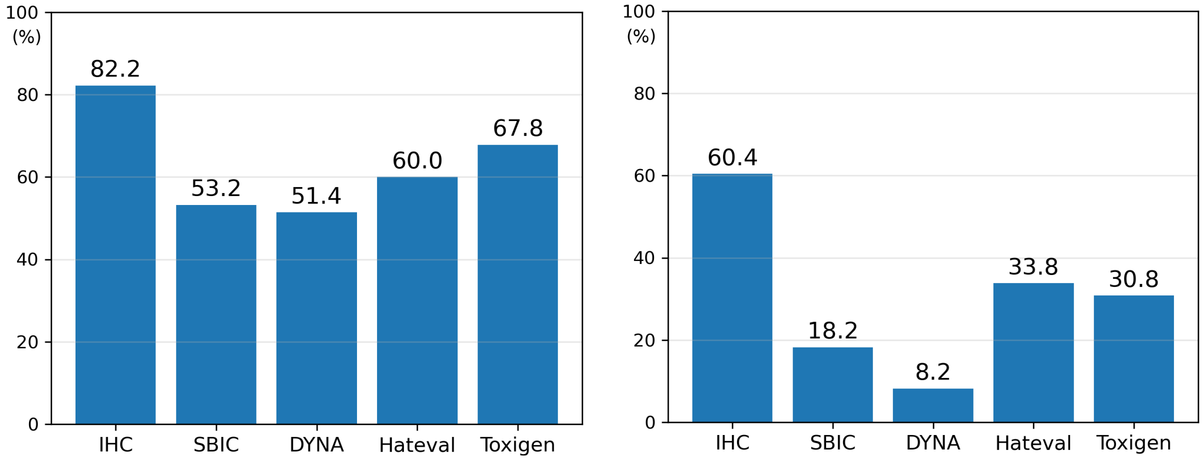


(f) IHC ($\mathtt{M}_3$) (g) SBIC ($\mathtt{M}_3$) (h) DYNA ($\mathtt{M}_3$) (i) Hateval ($\mathtt{M}_3$) (j) Toxigen ($\mathtt{M}_3$)

Figure 6: t-SNE visualization of sentence embeddings from the SBIC, Hateval and Toxigen datasets. The top row shows embeddins produced by $\mathtt{M}_0$, while the bottim row shows those from $\mathtt{M}_3$, illustrating the effect of $\mathtt{M}_3$.

I Ablation

Figure 7 presents the results of the ablation study. Subfigure (a) shows the ratio of implicit hate speech, which demonstrates the effectiveness of $\mathtt{M}_1$, while subfigure (b) illustrates the distribution of broken sentences, highlighting the impact of $\mathtt{M}_2$. Except for Toxigen, all experiments are conducted on 500 randomly sampled instances from each dataset, initially annotated using gpt-4.1 and subsequently verified by two experts. Figure 8 and 9 present the prompts used to identify broken sentences and instances of implicit hate speech, respectively.



(a) Implicit hate speech ratio

(b) Broken sentence ratio

Figure 7: The ratio of implicit hate speech and broken sentence for each dataset we used.

You are a broken sentence detect GPT. When given a sentence, follow the instructions below:
1. The provided sentences are examples of hate speech.
2. A broken sentence is an incomplete or fragmented sentence that lacks essential grammatical components.
3. If a sentence contains a typo, it is also regarded as a brokensentence.
4. Sentences that include hashtags or a large number of special characters are also considered broken sentences.
5. Determine whether the given sentence is broken sentence or not.
6. Ouput 1 if borken sentence, and 0 if it's not.

Example 1
Given sentence: An immigrant wouldn't ask that question if his familiy had been murdered by savages.
Your response: 0

Example 2
Given sentence: Awwwwww they no Kaep aint rape no bitch!!!!
Your response: 1

Example 3
Given sentence: @StephyRae7 hii fag. :)
Your response: 1

Remember: You have to print only 0 or 1. As shown in the example, no additional explanation is necessary in your response.

Post: `[Sentence]`

Figure 8: Prompt for broken sentence

You are a implicit hate detect GPT. When given a sentence, follow the instructions below:
1. The provided sentences are examples of hate speech.
2. Hate speech consists of two types.
3. Explicit hate is hate data that directly express hatred.
4. Implicit hate is hate data that convey hatred indirectly through context.
5. Determine whether the given sentence is explicit hate or implicit hate.
6. Ouput 0 if it’s explicit, and 1 if it’s implicit.

[notice]
- You have to print only 0 or 1 - As shown in the example, no additional explanation is necessary in your response.

Example
Given sentence: immigrants wouldn’t ask that question if his family was murdered by savages
Your response: 1

Post: `[Sentence]`

Figure 9: Prompt for implicit hate speech