ToxiTwitch: Toward Emote-Aware Hybrid Moderation for Live Streaming Platforms

Baktash Ansari*
baktash@uw.edu
University of Washington
Bothell, WA, USA

Elias Martin
University of Washington
Bothell, WA, USA

Afra Mashhadi
mashhadi@uw.edu
University of Washington
Bothell, WA, USA

Abstract

The rapid growth of live-streaming platforms such as Twitch has introduced complex challenges in moderating toxic behavior. Traditional moderation approaches, such as human annotation and keyword-based filtering, have demonstrated utility, but human moderators on Twitch constantly struggle to scale effectively in the fast-paced, high-volume, and context-rich chat environment of the platform while also facing harassment themselves. Recent advances in large language models (LLMs), such as DeepSeek-R1-Distill and Llama-3-8B-Instruct, offer new opportunities for toxicity detection, especially in understanding nuanced, multimodal communication involving emotes. In this work, we present an exploratory comparison of toxicity detection approaches tailored to Twitch. Our analysis reveals that incorporating emotes substantially improves the detection of toxic behavior. To this end, we introduce ToxiTwitch, a hybrid model that combines LLM-generated embeddings of text and emotes with traditional machine learning classifiers, including Random Forest and SVM. In our case study, the proposed hybrid approach reaches up to 80% accuracy under channel-specific training (with 13% improvement over BERT and F1-score of 76%). This work is an exploratory study intended to surface challenges and limits of emote-aware toxicity detection on Twitch.

CCS Concepts

• **Computing methodologies** → *Machine learning approaches*; • **General and reference** → *Empirical studies*.

Keywords

Toxicity Detection, Large Language Models

Live-streaming platforms such as Twitch have become major arenas for real-time social interaction, where users express themselves through a mix of text, slang, and platform-specific visual symbols called emotes. While these emotes enrich communication, Twitch's largely unmoderated chat environment has made both streamers and moderators vulnerable to toxic behaviors, including harassment, hate speech, and community-driven hostility [14, 33]. Each streamer relies on volunteer moderators who manually review thousands of fast-paced comments [44], a form of invisible labor that carries emotional and psychological burdens [37, 45]. Twitch users collectively produce over 29 billion chat messages annually [1], generating more than 80 GB of daily data—making comprehensive human moderation infeasible. From a Responsible AI standpoint, moderation on Twitch poses distinctive challenges. Unlike text-only platforms, toxicity on Twitch is multimodal and culturally situated: meanings emerge not only from words but from

*Corresponding author.

visual symbols (emotes) that carry community-specific and evolving semantics [22, 28]. The same emote may signify humor in one context and harassment in another, depending on the streamer, audience, and conversational flow. Automated systems that treat toxicity as static or universal risk misinterpreting cultural symbols, reinforcing biases, or over-censoring marginalized voices. Developing AI systems that are both context-sensitive and transparent about their limitations is therefore critical to ensuring safe and equitable participation in live-streaming communities. Prior studies have begun to analyze Twitch toxicity using Natural Language Processing (NLP) methods. Kim et al. [22] demonstrated that emotes embedded within words can evade detection by standard text classifiers such as HateSonar [9]. Moosavi et al. [28] later showed that channel-specific emotes are disproportionately present in toxic interactions, yet assumed a fixed relationship between an emote and its toxicity label. These approaches, while valuable, do not address the contextual fluidity and cultural drift that define Twitch communication. Effective moderation requires models capable of reasoning about emotes' meanings within their social and temporal contexts, and doing so responsibly, without overgeneralizing across communities. Recent advances in Large Language Models (LLMs) and Vision–Language Models (VLMs)—such as GPT-4, Llama 3-8B-Instruct [38], and Deepseek-R1 [11]—offer new opportunities to capture this nuance by integrating textual and visual reasoning. Yet their performance in culturally embedded environments like Twitch remains underexplored. How well do these models interpret the social meaning of emotes? Can they generalize across communities with distinct linguistic norms? And how might their reasoning be augmented to mitigate bias and improve transparency? This study investigates these questions through three research directions:

- **RQ1 (LLM Reasoning):** How well do reasoning-enabled LLMs perform in zero-shot toxicity classification across different Twitch communities?
- **RQ2 (Emote-Enabled Prompting):** Can enriching prompts with contextual information about emotes enhance interpretability and reduce misclassification?
- **RQ3 (Hybrid Model Efficacy):** Can combining LLM-generated embeddings with lightweight machine learning classifiers yield an efficient, interpretable, and latency-aware framework for real-time moderation?

To explore these questions, we focus on two distinct Twitch channels—HasanAbi, a "Just Chatting" channel where political discourse often sparks controversy, and LolTyler1, a gaming channel known for high-tempo interactions and gender-based toxicity. We collect 500 comments per channel and obtain human annotations from three coders, capturing generational and subjective perspectives on toxicity. Our contributions are twofold:

- We investigate how contextual features extracted from emotes can enhance LLM reasoning for toxicity detection, showing that structured emote-aware cues reduce false positives by clarifying community-specific slang and symbolism.
- We introduce ToxiTwitch, a hybrid architecture that integrates text and emote embeddings from LLMs with lightweight classifiers (Random Forest, SVM), achieving near 80% accuracy with a 60 ms inference time per message—suitable for real-time deployment.

While our study is exploratory in scale, it provides insight into how multimodal reasoning and hybrid modeling can inform more responsible and context-aware AI moderation on dynamic social platforms.

1 Related Work

Toxicity on live streaming platforms like Twitch remains a growing concern as they expand beyond gaming [12]. Studies show topics like racism and politics attract more toxic comments than science or arts [12, 35]. The anonymity and pace of live chats reduce accountability, fueling toxic behavior through disinhibition [35], with consequences like mental health risks and hostile communities [17]. Solutions include better moderation tools and community guidelines [12]. Kim et al. [22] reverse-engineered toxic visual chats (e.g., emotes replacing letters), uncovering 196,448 new toxic comments via neural classifiers. Dreier and Pirker [12] found that male-hosted streams, larger audiences, and multiplayer/shooter games correlate with higher toxicity. To this end, we review previous research on detecting toxicity on Twitch and highlight the contributions of our paper based on the identified research gaps.

1.1 Towards Automated Toxicity Detection on Live Streaming Platforms

Twitch's community guidelines allow streamers to appoint human moderators who manually filter chat messages, enforcing rules by blocking or suspending violators [39]. However, moderators face significant challenges, including processing large volumes of messages in real-time and enduring harassment while mediating conflicts [21, 36, 44]. To alleviate these burdens, researchers have proposed integrating automated moderation tools to assist human moderators in maintaining a less toxic environment. Moon et al. [27] focuses on detecting norm violations, including toxic language, in live-streaming platforms like Twitch. It uses automated methods, specifically training models with contextual information, to improve moderation performance. They conclude that models trained with existing data sets perform poorly in detecting toxic messages and motivate the development of specialized approaches for Twitch. Recent research has explored various computational approaches to detect and mitigate toxicity in live streaming platforms. We organize prior work into three key themes: (1) toxicity detection methods, and (2) platform-specific challenges. Efforts to improve real-time detection include model efficiency and cross-platform adaptation. Oikawa et al. [32] proposes a stacking-based method for Japanese live chats, prioritizing inference speed for offensive language. Similarly, Gao et al. [13] leverages transfer learning from Twitter to detect offensive content on Twitch, addressing data scarcity.

However, most of these solutions lack the incorporation of Twitch-specific features. We take inspiration from [28], which uses toxicity detection on Twitch by creating an emote embedding space. They use a RoBERTa-based toxicity classifier to identify seed emotes and then discover other toxic emotes through embedding association. The paper also discusses the limitations of NLP models in detecting toxicity in Twitch chat and proposes an active learning moderating system.

A core limitation is the difficulty in interpreting complex linguistic features such as sarcasm, metaphors, plus the addition of platform-specific emotes, often leading to both false positives and missed detections [19, 31]. Even sophisticated tools like the PERSPECTIVE API struggle with interpretability, as their toxicity scores reflect prediction confidence rather than severity, limiting their practical utility [43]. Additionally, there is an over-reliance on lexicon-based methods and models trained on data from other platforms, which fail to capture Twitch's unique contextual dynamics [12, 34]. While lexicon-based approaches can identify known toxic patterns, they are ineffective against novel or evolving forms of toxicity [3]. The scarcity of labeled data poses another major constraint, particularly for token-level toxicity detection, where transformer models require extensive fine-tuning datasets that are often unavailable [20]. Current methods also grapple with false positives and negatives, especially in real-world scenarios where toxic instances are rare. Even low false positive rates become problematic at scale, generating excessive false alarms [19].

While LLMs show promise for toxicity detection, they introduce new challenges. These include sensitivity to prompt quality, high computational costs, and latency issues when processing large volumes of text [30]. Advanced models like ToxBuster acknowledge they cannot fully replace human moderators and are unsuitable for fully automated moderation [46]. These limitations demonstrate that no single automated solution can perfectly address toxicity detection, especially in dynamic environments like live streaming platforms. This underscores the necessity for hybrid approaches and continuous methodological improvements [2, 47].

1.2 Emotes and Toxicity Detection Challenges

Emotes in Twitch have become an integral part of online communication and research has shown that they play a significant role in toxic communication in more nuanced and covert ways, making them a crucial clue for comprehensive toxicity detection systems [22, 28]. The visual nature of emotes allows users to convey complex emotions and intentions that may *not* be easily captured by traditional text-based methods. In this vein, studies have demonstrated that certain emotes are frequently associated with toxic behavior, and their usage patterns can provide valuable insights into the overall toxicity of a conversation [22]. However, the context-dependent nature of emotes adds complexity to toxicity detection, as the same emote may have different connotations depending on the surrounding text and conversation [8]. To address this challenge, Moosavi et al. [28] mapped an embedding space of channel and global emotes, which could be used to discover similarity of emotes across communities and thus aid with knowledge transfer of toxicity detection models across communities. Kim et al. [22]

developed labeled datasets and neural network classifiers specifically designed to identify toxic chats that would be missed by traditional methods. They demonstrated that analyzing emotes in chat conversations can catch an additional 1.3% of toxic instances out of 15 million utterances on Twitch. This finding underscores the need for more sophisticated detection methods, as traditional approaches may miss visual forms of toxicity. While prior work such as Kim et al. [22] and Moosavi et al. [28] provide important insights into emote usage, their experimental setups differ in scope and dataset composition, making direct comparison infeasible. Instead, we situate our work, ToxiTwitch, as complementary: our contribution lies in demonstrating the latency–accuracy tradeoff of hybrid architectures.

2 Dataset and Annotation

This section outlines our data collection methods and the human annotation technique.

2.1 Twitch Comments

In curating a dataset and selecting Twitch streams suitable for this task, we identified two topics of just-chatting and gaming. Within channels related to these topics, we then selected two channels that are consistently known as the top 10 popular but also controversial channels on Twitch. In the gaming category, this resulted in a channel dedicated to the League Of Legends game named *LolTyler1*[1]. In just chatting category, we selected the *HasanAbi*[2] channel, which is known for its coverage of controversial discussions.

For HasanAbi, data was recorded for November 17th 2022, and captured the entirety of the channel's stream chat logs. Similarly, for LolTyler1, data was captured during one of his streams on November 11, 2022. Overall, we captured 500 comments from each of the two channels.

These comments are in length typically 10-15 words for HasanAbi and 0-5 words for LolTyler1, indicating that the majority of comments on Twitch are relatively short. This makes content moderation a unique challenge as the majority of comments lack the length and context that more traditional forms of moderation might be trained to perform on platforms with lengthier comments, such as Twitter or Wikipedia.

Each HasanAbi comment contains on average 1.40 emotes, showing a noticeable use of emotes in communication. Among the emotes used, a large majority are channel-specific emotes, making up over 80% of all emote usage across all emote count ranges. The rest are global emotes, which appear more often in comments with fewer total emotes. This shows that viewers in HasanAbi's stream mostly use emotes that are unique to the channel, which usually require a subscription to unlock.

Similarly, the LolTyler1 dataset also shows emote usage but at a lower rate. On average, each comment includes 0.72 emotes. Like HasanAbi's data, most of the emotes used are channel-specific, with over 80% of all emotes being tied to the channel. As the total number of emotes in a comment increases, the share of channel emotes also increases, reaching nearly 100% in comments with more than 5 emotes.

Figure 1 shows the percentage of global and channel-specific emotes used in Twitch comments, illustrating patterns in emote popularity. These results show the strong presence of emote culture in Twitch chats, especially through the use of channel-specific emotes. This makes it harder for NLP models to understand the content, since many of the most common emotes are not available outside the specific channel and often carry unique meanings within that community.

2.2 LLM's Knowledge of Emotes

Given the popular nature of global emotes and their widespread presence in public data, it is highly likely that these LLMs encountered them during training, enabling them to incorporate knowledge of these emotes into their language models. We investigate the extent to which LLMs have acquired knowledge of Twitch's global and channel emotes, focusing on their ability to interpret and explain the meaning of an emote.

Global Emotes: To test LLM's knowledge of the global emotes, we randomly selected 100 global emotes and performed the following analysis. We prompted Llama-3-8B to explain the meaning of each emote. We then manually inspected its answers. Out of 100 global emotes Llama described 93 of those emotes correctly. Through our analysis we found that LLMs could accurately interpret global emotes in both isolated and contextually rich environments, simulating real-world Twitch conversations. We observed similar results with Deepseek.

Channel-specific Emotes: As channel-specific emotes are smaller sets, we ran the entire set of HasanAbi channel emotes and Loltyler emotes and prompted the LLMs whether they knew these emotes. Out of the 318 Loltyler channel-emotes Llama-3-8B could describe only 72 emotes (23%). Out of 736 HasanAbi channel-emotes, Llama-3-8B only could provide description of 145 emotes (20%).

For Deepseek, we observe that the model tries to convince itself through reasoning steps that it knows the emote. It often relies on contextual clues, such as the emote text and the name of the channel, to generate an explanation. The final answer is usually given with a high degree of uncertainty, so it is unclear whether the model truly understands the emote or is just hallucinating a response.

2.3 Ground-Truth Human Annotation

2.3.1 Annotation Guideline. In order to assess the performance of various models, we require a human-annotated validation set. Toxicity labeling is a challenging annotation task due to the inherent subjectivity and ambiguity involved. To address this, we developed a detailed annotation code-book [3] and recruited three annotators. While all annotators currently reside in the United States, they brought diverse cultural perspectives. Annotators were asked to classify comments as toxic or non-toxic, and for toxic comments, to assign one or more relevant toxicity categories (e.g., obscene, threat, insult, identity attack, sexually explicit) as in [6]. Annotators were told to look up the meanings of emotes to assist in their decisions. To this end and to minimize misinterpretations, annotators consulted emote dictionaries or platforms such as BTTV and FFZ, particularly for channel-specific emotes that may not be widely

---

[1]https://www.twitch.tv/loltyler1
[2]https://www.twitch.tv/hasanabi

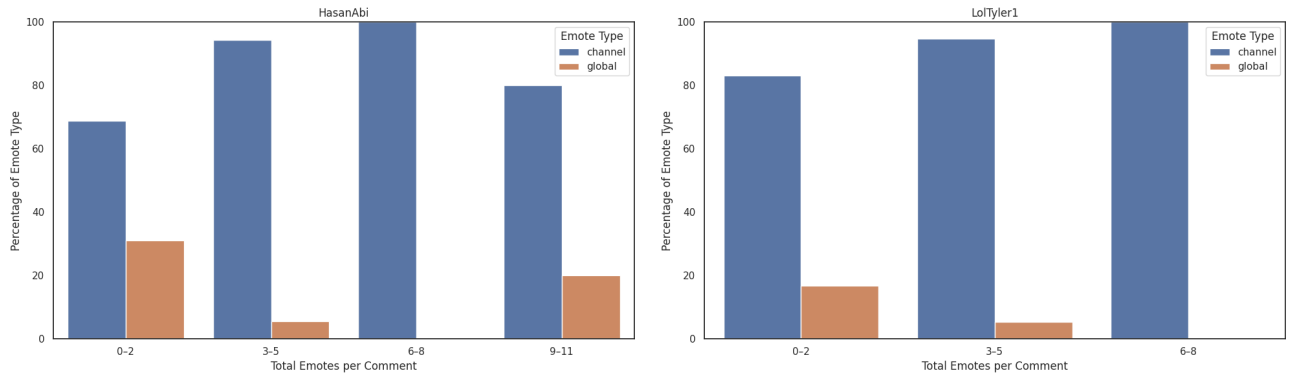[3]https://anonymous.4open.science/r/toxitwitch-C5A7/codebook.pdf

Figure 1: Percentage of Global vs. Channel Emotes by Number of Emotes per Comment

recognized outside certain Twitch communities. This additional step was critical to ensure the correct assessment.

2.3.2 Annotated Dataset. In order to balance the dataset and to increase the chances of toxic comments in the annotation pool, we first applied a toxicity detection model to our datasets described above. We chose a DistilBERT model trained on ToxiGen Dataset [16] to perform the initial pass and classify the comments into toxic/non-toxic. ToxiGen [16] is a toxicity dataset generated using an LLM to analyze and address toxicity in language. It is designed to capture diverse forms of toxic content, often organized into categories representing different types of harmful speech. ToxiGen stands out due to its focus on the subtle, contextual, and diverse aspects of toxic language.

Out of the comments that were predicted as toxic by Distilbert-Toxigen, we selected 500 comments that higher prevalence of emotes in them for each of channels. All three annotators labeled each data point and the label with the majority votes was chosen.

3 Experimental Setting

3.1 Large Language Models

LLMs have emerged as powerful tools for toxicity detection, leveraging their deep contextual understanding and vast knowledge to identify harmful content across multiple languages and domains. We leverage two open-source implementations of LLMs, namely LLaMA 3-8B-Instruct [38] and Deepseek-R1-Distill [11], for our analysis. Both models are comparable in scale, with LLaMA 3-8 B-Instruct and Deepseek-R1-Distill containing approximately 8 billion parameters each. For all our experiments, we used hyperparameters temperature = 0.5 and a probability threshold for sampling tokens (top_p) = 0.9 for consistency.

3.1.1 LLaMA 3-8B-Instruct [38]: Llama-based models, developed by Meta, are highly effective for toxicity detection due to their advanced language understanding and training on large, diverse datasets [26, 29]. These models can identify harmful language patterns, including hate speech and cyberbullying, even in subtle contexts. Compared to traditional models like RoBERTa, Llama's generative capabilities make it adaptable to evolving toxic behaviors, particularly in real-time environments like social media [26].

3.1.2 Deepseek-R1 [11]: We chose Deepseek as the second model for comparison in our study. DeepSeek-R1-Distill is optimized for high-throughput, low-latency environments where both speed and precision are essential for real-time applications like content moderation. While larger models (e.g., GPT-4, Claude) achieve state-of-the-art results on broad NLP benchmarks, DeepSeek-R1-Distill strikes a practical balance between model efficiency, inference cost, and task-specific accuracy, while remaining open-weight and deployable on commodity hardware. Its design prioritizes scalability for live platforms, making it a compelling candidate for comparison against both heavyweight LLMs (e.g., LLaMA 3-8B-Instruct) and compact task-specific models (e.g., DistilBERT). By evaluating DeepSeek-R1-Distill alongside these alternatives, we assess for real-world viability, particularly in scenarios demanding low-resource operation without sacrificing robustness.

3.2 Prompt Baselines

Prompts are text fragments passed into LLMs. In the past years, research has shown that designing prompts is notoriously challenging, and even small differences in the prompt template can significantly impact the answer of the LLMs [25]. In this work, we focus on soft prompting, that is the content of the comment is integrated into the prompt. We also focus on *zero-shot* prompting, that is, the prompt given to the LLM does not include any examples or demonstration [23].

There are several approaches for zero-shot prompting pre-trained LLMs that have been shown to help LLMs in implementing step-by-step reasoning. One such approach is Chain-of-thought (CoT) [42]. CoT is a robust method for reasoning, particularly effective in addressing open-ended questions. It works by breaking down complex problems into a series of logical, coherent steps, facilitating clearer and more structured reasoning. Previous research on LLMs' reasoning on social media data has shown that COT outperforms vanilla prompting by far [4, 18, 41]. By employing COT prompting, we believe that the LLMs would be able to reason on the *global* emotes embedded in comments and detect toxicity more accurately. The COT prompt that we adapted to this study is as follows:

```
"If you were a moderator on the Twitch streaming
platform, imagine encountering a comment with the following
content. Your objective is to flag the comment as
<toxic> or <non-toxic> based on the following toxicity
definitions: <obscene, threat, insult, identity attack,
sexually explicit>. Consider this reasoning step by step.
Return a JSON object with four fields: "Is it toxic,"
which is either "yes" or "no" depending on whether you
would remove this comment from the stream; which toxicity
category it falls into; and "explanation," which provides
a reason for your decision; and whether your explanation
included any emotes present in the message and whether the
emote conveyed toxicity."
```

4 Our Proposed Method

We propose a modular, emote-aware AI moderation agent designed to operate in real-time, high-volume social platforms such as Twitch. Unlike static classifiers, our model, ToxiTwitch, functions as a contextually grounded reasoning agent that continuously interprets user-generated chat streams through three tasks:

Perceiving multimodal inputs (text and emotes) through lightweight preprocessing and embedding generation;

Reasoning about the contextual intent of emotes using prompt-augmented LLMs, which act as the agent's inference engine;

Acting via a decision-making module—realized through traditional classifiers—that flags toxic content with low latency and high precision.

We describe perceiving and reasoning task as part of a combined stage referred to as Emote-Enabled-Prompting, and the acting task of the agent in the ToxiTwitch model.

4.1 Emote-Enabled-Prompting

Previous research has shown that providing extra context as part of prompt can enhance LLM's social reasoning abilities [4]. To this end, we incorporate additional information related to channel emotes into the LLMs in two ways:
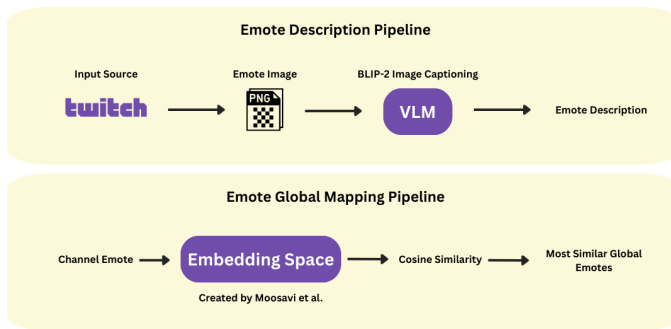

Figure 2: Emote Description Generation and Global Mapping Pipelines

4.1.1 Emote-Description (ED):. For each channel emote, we construct a description of the emote. The system relies on a pipeline that maps each channel-specific emote to a description of what the emote represents using multimedia processing. Figure 2 illustrates our process for generating these emote descriptions. We used the

BLIP-2 model [24] to generate descriptions for each emote image. Finally, we created a dictionary for each emote along with its description. We modify the COT prompt template and append the emote name followed by its description for every emote that was seen in the comment:

```
"Consider that <channel-emote> in this comment is described
as <emote description>. Perform step-by-step reasoning."
```

4.1.2 Emote-Global-Mapping (EGM):. Alternatively and given that LLMs have a good knowledge of global emotes (as shown in Section 3.3), we find the top most similar global emotes for each channel emotes. To do this, we used the embedding space created by Moosavi et al. [28] which was based on the top 2379 English-speaking channels in 2022. Unlike other available Twitch embedding spaces, this dataset includes mapping of global and, importantly *channel emotes*, enabling us to calculate and retrieve the similarity of emotes across different channels and communities. As the embedding space was created using Word-2-Vec, we applied cosine similarity to measure relations between channel to global emotes. To provide LLMs with contextual information about the *channel* emotes, we use this embedding space to map the top three closest **global** emotes to any given channel emote. Figure 2 illustrates our process for extracting the most similar global emotes to our channel emote. We *append* the extracted top global emotes to the zero-shot CoT prompt template:

```
"Consider that <channel-emote> in this comment is closest
to Global Emotes:(GE1,GE2,GE3). Perform step-by-step
reasoning."
```

4.2 ToxiTwitch Model

Our framework processes Twitch comments in two stages: (1) contextual embedding extraction using Large Language Models (LLMs), and (2) toxicity classification via simple machine learning models. The pipeline is designed to handle the unique characteristics of platform-specific emotes, fast-paced interactions, and contextual dependencies.

4.2.1 Embedding Generation. To generate semantically meaningful representations of chat messages, we leverage LLMs to obtain dense embeddings. Given a sequence of chat messages $X = \{x_1, x_2, \ldots, x_n\}$, where each x_i denotes a single user message, our objective is to encode each message into a fixed-size vector capturing both lexical and contextual semantics. The tokenized input is then passed through the LlaMA transformer encoder, which produces token-level contextual embeddings. For LLaMA 3-8B-Instruct, the embedding for message x_i is:

$$H_i = \text{Llama}(x_i) \in \mathbb{R}^{L \times d} \qquad (1)$$

where L is the number of tokens in the message after tokenization, and $d = 4096$ is the hidden dimensionality of the model. The output matrix $H_i = [h_{i1}, h_{i2}, \ldots, h_{iL}]$ consists of token embeddings $h_{ij} \in \mathbb{R}^d$ for each token j in message i. We extract these embeddings from the final layer of the transformer to capture the highest level of semantic abstraction.

To obtain a message-level representation, we apply mean pooling over all token embeddings:

$$e_i = \frac{1}{L} \sum_{j=1}^{L} h_{ij}, \quad e_i \in \mathbb{R}^d \tag{2}$$

This operation yields a single vector e_i representing the entire message content. For Deepseek-R1, we follow an analogous procedure but with $d = 2048$. These message-level embeddings $\{e_1, \ldots, e_n\}$ serve as input features for the downstream task of toxicity classification. To account for the contextual nature of conversational toxicity, we also explored incorporating emotes into the text embeddings.

4.2.2 Emote Handling Strategies. A critical challenge in processing Twitch chat is the prevalence of platform-specific emotes that carry semantic meaning but may be tokenized sub-optimally by general-purpose language models. We leverages the same two distinct strategies for emote handling as mentioned in Section 4.1. For each strategy, we compute separate embedding sets and evaluate their downstream classification performance.

4.2.3 Toxicity Classification. The generated embeddings serve as feature vectors for toxicity classification models. We use traditional machine learning models, Random Forest [5] and Linear SVMs [7] for the classification stage due to their efficiency and simplicity in compared to LLMs. We hypothesize that by using simple traditional machine learning models, we can achieve performance comparable to zero-shot experiments with the added advantage of lower computational cost. In subsequent sections, we evaluate the effectiveness of multiple embedding strategies, raw text embeddings, emote description augmentation, and emote replacement across different classification architectures. We use repeated stratified cross-validation with separate held-out test sets per fold; we retain the original description to reflect the exploratory setup.

5 Results

In this section, we present the results of our study in addressing each of the research questions.

5.1 Baseline Performance of LLMs

We first compare the models in terms of their performance to understand how well they perform when classifying toxicity on Twitch in different communities. Table 1 presents the results of this comparison. As can be seen, both LLMs perform comparatively to each other and across two different communities. Although the models exhibit high recall, they suffer in terms of low precision, that is, suffering from a high number of false positives. These results confirm our hypothesis that LLMs lack precision and due to their reasoning become over sensitive in flagging toxicity when they lack information about community nuances.

Table 1: Model Performance Comparison Across Datasets

Dataset	Model	Acc.	Precision	Recall	F1
HasanAbi	Llama	0.84	0.30	0.81	0.39
	DeepSeek	0.78	0.25	0.90	0.39
LolTyler1	Llama	0.63	0.37	0.90	0.52
	DeepSeek	0.53	0.32	0.87	0.47

5.2 Impact of Emote-Enabled Prompting

Table 2 presents the result of our emote-enabled prompting technique for both ED and EGM across models and communities. As we can see by including information on emotes, in all cases the models improve in their reasoning about toxicity in terms of F1 Score. While we do not observe a drastic increase in precision, we can see that in some cases such as Llama-ED on Hasanabi, the description of the channel emotes has guided the model to achieve higher recall, while maintaining precision. It is worth noting that as we described in Figure 1 there is a relatively low usage of emotes in our sample, which explains the lack of significant changes in the emote-enabled prompting compared to the baseline.

Table 2: Emote-Enabled-Prompting Performance Comparison Across Datasets. Underlined values denote the highest performance for the channel, and stars denote p-value with significant improvement of F1 score in comparison to baseline results of Table 1. p-values < 0.05 are indicated with *.**

Dataset	Model	Acc.	Precision	Recall	F1
HasanAbi	Llama-ED	<u>0.84</u>	<u>0.31</u>	<u>0.87</u>	<u>0.46</u> ***
	Llama-EGM	0.83	0.30	0.84	0.44 ***
	DeepSeek-ED	0.79	0.26	0.88	0.45 ***
	DeepSeek-EGM	0.77	0.25	0.84	0.38
LolTyler1	Llama-ED	0.64	0.38	0.85	0.53 ***
	Llama-EGM	<u>0.61</u>	<u>0.37</u>	<u>0.92</u>	<u>0.53</u> ***
	DeepSeek-ED	0.63	0.36	0.80	0.50 ***
	DeepSeek-EGM	0.62	0.36	0.85	0.51 ***

5.3 Evaluating ToxiTwitch: Hybrid Model for Toxicity Detection

5.3.1 ToxiTwitch Performance. We present the performance of various embedding strategies on two streamers—HasanAbi and LolTyler—using Random Forest and Linear SVM classifiers in Tables 3 and 4 respectively. Our findings reveal that the addition of EGM significantly outperforms all other embedding configurations. For HasanAbi, Textual Embedding + EGM achieves the highest F1 score of 0.79 (RF) and 0.79 (SVM), along with substantial gains in accuracy (0.86 and 0.83, respectively). Similarly, Textual Embedding + EGM attains the strongest performance for LolTyler1, with F1 scores of 0.78 (RF) and 0.78 (SVM). Providing emote descriptions also enhances the base embeddings. When ED is added to Llama or Deepseek embeddings, performance improves consistently across all metrics. For instance, in the HasanAbi dataset. However, the magnitude of improvement from ED is less than that of EGM, indicating that replacing channel emotes with global emotes offers richer context than descriptions (ED) alone. We also find that Llama-based embeddings generally outperform Deepseek-based ones across both datasets. For example, Textual Embedding + EGM achieves an F1 of 0.79 (RF) for HasanAbi, whereas Textual Embedding + EGM scores 0.75. This trend is consistent in the LolTyler1 dataset, indicating that Llama embeddings provide more semantically rich representations for downstream classification tasks in this domain. Lastly, Random Forest slightly outperforms Linear SVM in most configurations, particularly in Recall and Accuracy, suggesting that

ensemble-based models may be better suited to capture the subtle variations in toxic content.

5.3.2 Benchmark Comparison. To assess the efficacy of our proposed hybrid model for toxicity detection on Twitch, we perform a comparative evaluation against three publicly available transformer-based baselines: Detoxify [40], HateSonar [10], and DistilBERT-ToxiGEN [15] and our model ToxiTwitch. For ToxiTwitch, we selected the best performing setting based on previous observation (Section 6.3.1) which corresponds to Random Forest on Llama Text + EGM. To this end, we collected two **different** days from our original training data of HasanAbi and LolTyler1 that were previously not seen by ToxiTwitch. For each streamer, we inferred toxicity labels in real time until 200 toxic and 200 non-toxic messages were identified by ToxiTwitch. We then manually annotated all 800 messages (400 per streamer) to establish human ground truth labels following the same procedure stated in Section 3.3.2.

In addition, we measured inference latency (in milliseconds) on a single-core CPU Thermal Design Power (TDP) of 29 Watts and using PyTorch, simulating a constrained but realistic deployment setting for live moderation systems. Our latency metric reflects the time required to process a single average-length Twitch message (16–24 tokens).

ToxiTwitch achieves competitive performance in terms of F1-score and precision, demonstrating its ability to reduce false positives while retaining high recall (Tables 5) while demonstrating the viability of hybrid emote-aware models. Notably, our model achieves an F1 of 0.73 and 0.71 on HasanAbi and LolTyler datasets, respectively, compared to 0.68/0.65 for HateSonar and substantially lower scores for Detoxify and ToxiGEN. These improvements can be attributed to the hybrid architecture's capacity to leverage both linguistic semantics and emote-based contextual cues common in Twitch discourse. While HateSonar achieves slightly higher recall, its precision is lower, suggesting more false positive detections.

Regarding latency, ToxiTwitch remains competitive, with an average inference time of 60 ms per message—within the threshold for near-real-time moderation and only marginally higher than Detoxify (45 ms), which is optimized for speed but shows reduced classification accuracy.

6 Discussion

In this section, we discuss the key implications of our findings based on our three overarching research questions.

RQ1 (LLM Reasoning): How well do LLMs with reasoning capabilities perform in zero-shot classification to detect toxicity in Twitch chat communities? We found that LLMs with reasoning capabilities achieved baseline F1 scores ranging from 0.39 to 0.52. In particular, this was over an initial sample of 500 messages per channel, where the Toxigen model labeled the messages as toxic. However, only 7.8% of these messages for HasanAbi and 23% for LolTyler1 were actually toxic according to human annotators.

RQ2 (Emote-Enabled Prompting): Can augmenting prompts with contextual information about emotes enhance LLM reasoning and improve toxicity detection by reducing misclassification? Our focus was not on exhaustively benchmarking all possible moderation models (e.g., OpenAI API), but rather on demonstrating how

emote-aware prompting and hybrid architectures can contribute to the space. Our results from the agentic perceive-and-reason implementation—which included various methods for injecting emote context into LLM prompts—demonstrated slight improvements in F1 score, even with a small sample of channel-specific emotes included. However, precision remained low, indicating that while contextual emote information helped to some extent, it did not significantly improve overall classification correctness.

RQ3 (Hybrid Model Efficacy): Can we develop a novel hybrid architecture that combines LLM-generated embeddings with lightweight classical machine learning models to outperform state-of-the-art baselines in both accuracy and latency? We showed that integrating different agentic components (i.e., perceive-reason-act) of ToxiTwitch and leveraging machine learning models trained on labeled data enabled us to construct a low-latency system with high accuracy and precision. This hybrid model effectively mitigated the limitations of LLM-only approaches and demonstrated improved performance across key metrics.

6.1 Limitation

We acknowledge that our dataset is limited to two channels and 1,000 comments. While this scale restricts generalizability, our goal was to provide a proof-of-concept framework rather than a comprehensive benchmark. Future work should validate ToxiTwitch across broader channel diversity and larger datasets. Another limitation of our current approach focusing on EGM is the assumption that the learned representations in embedding space that enable us map channel emotes to global emotes remain consistent overtime. Indeed, Moosavi et al. [28] showed that both channel and global emotes meaning changes overtime and thus the similarity association between them may evolve too. These changes mean that for EGM approach to work efficiently we require the embedding space to remain up-to-date. In this study, we did not face this issue as we purposely collected Twitch chat messages that corresponded to the time-frame that the embedding space was created. However, we discuss approaches to overcome these two limitation next.

6.2 Future Work

Future research could explore addressing some of the observed limitations of our work in two threads. Firstly, we believe that there is a great opportunity to expand our annotation process. Currently, majority of works in LLMs rely on small binary annotation that assist the classifiers in binary prediction tasks. We believe future work could focus on focusing on LLMs reasoning by extracting reasoning syntax tree from LLMs and integrate Human in the loop to annotate the structural reasoning of the task as opposed to its binary outcome. Finally, we intend to investigate the use of continual learning techniques to update the model incrementally without retraining from scratch, ensuring responsiveness to platform dynamics while preserving past knowledge. Although a full ablation of ED vs. EGM vs. text-only embeddings would strengthen the analysis, we see our results as an initial demonstration that both strategies provide measurable gains. A deeper per-category and ablation study is a valuable avenue for future extensions.

Table 3: Performance comparison of embedding techniques with highest values of each metric being underlined. (HasanAbi)

Embedding Technique	Random Forest				Linear SVM			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Llama Text Embedding	0.64	0.79	0.71	0.75	0.61	0.74	0.67	0.70
Llama Text + ED	0.68	0.81	0.74	0.76	0.68	0.79	0.73	0.74
Llama Text + EGM	0.74	0.86	0.79	0.86	0.76	0.83	0.79	0.83
Deepseek Text Embedding	0.62	0.70	0.66	0.77	0.61	0.70	0.65	0.77
Deepseek Text + ED	0.70	0.75	0.72	0.78	0.70	0.72	0.71	0.72
Deepseek Text + EGM	0.71	0.80	0.75	0.78	0.73	0.76	0.75	0.78

Table 4: Performance comparison of embedding techniques with highest values of each metric being underlined (LolTyler1)

Embedding Technique	Random Forest				Linear SVM			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Llama Text Embedding	0.61	0.80	0.69	0.71	0.58	0.75	0.66	0.67
Llama Text + ED	0.65	0.82	0.73	0.73	0.65	0.80	0.72	0.73
Llama Text + EGM	0.70	0.87	0.78	0.79	0.72	0.84	0.78	0.78
Deepseek Text Embedding	0.59	0.71	0.64	0.65	0.58	0.72	0.64	0.65
Deepseek Text + ED	0.66	0.76	0.71	0.71	0.67	0.73	0.70	0.70
Deepseek Text + EGM	0.67	0.82	0.74	0.74	0.70	0.77	0.73	0.73

Table 5: Benchmark Comparison of SOTA models. The underline shows the highest performance metric/ lowest latency.

Model	Prec.	Rec.	F1	Acc.	Latency (ms)
HasanAbi					
Detoxify [40]	0.45	0.69	0.54	0.67	45
HateSonar [10]	0.55	0.88	0.68	0.77	70
DistilBERT-ToxiGEN [15]	0.24	0.72	0.36	0.43	55
ToxiTwitch (ours)	0.63	0.87	0.73	0.80	60
LolTyler1					
Detoxify [40]	0.38	0.75	0.50	0.63	45
HateSonar [10]	0.47	0.93	0.62	0.72	70
DistilBERT-ToxiGEN [15]	0.25	0.79	0.38	0.44	55
ToxiTwitch (ours)	0.61	0.88	0.72	0.79	60

7 Conclusion

The findings from ToxiTwitch highlight the complex interplay between technical design and sociocultural context in automated moderation systems. Emote-based communication on Twitch reflects community-specific and evolving visual languages that LLMs are not yet equipped to interpret fairly or transparently. Our results reveal how toxicity detection can fail when meaning is detached from its local social setting, underscoring the need for **Responsible AI** approaches that integrate community knowledge and cultural nuance into model development. By combining visual and textual reasoning with human-grounded annotations, ToxiTwitch illustrates a hybrid path toward moderation that is context-aware, latency-efficient, and interpretable. We argue that future Responsible AI frameworks for online moderation should prioritize not only accuracy but also semantic transparency, annotator diversity, and participatory evaluation, ensuring that AI systems respect the cultural norms and expressive practices of the communities they aim to serve.

References

[1] 2024. Twitch Statistics 2024: Revenue, Users, Viewership. https://electroiq.com/stats/twitch-statistics/. Accessed: 2025-05-20.

[2] Noura Aldahoul et al. 2024. Hybrid human-AI approaches for content moderation. *Nature Machine Intelligence* (2024).

[3] Shiza Ali, Jeremy Blackburn, and Gianluca Stringhini. 2025. Evolving Hate Speech Online: An Adaptive Framework for Detection and Mitigation. *arXiv preprint arXiv:2502.10921* (2025).

[4] Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Can LLMs Reason Like Humans? Assessing Theory of Mind Reasoning in LLMs for Open-Ended Questions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 34–44. doi:10.1145/3627673.3679832

[5] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[6] cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic Comment Classification Challenge. https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge. Kaggle.

[7] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[8] Pratik Davange, Pratik Chaudhari, ST Patil, and Arwa Bhojawala. 2024. Toxic Chat Detection Using Deep Learning. *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)* 11, 1 (January 2024), 12.

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

[10] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. https://github.com/nikbearbrown/HateSonar.

[11] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[12] Lukas Dreier and Johanna Pirker. 2023. Toxicity in Twitch Live Stream Chats: Towards Understanding the Impact of Gender, Size of Community and Game Genre. In *2023 IEEE Conference on Games (CoG)*. IEEE, 1–4.

[13] Zhiwei Gao, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2020. Offensive Language Detection on Video Live Streaming Chat. In *International Conference on Computational Linguistics*.

[14] Catherine Han, Joseph Seering, Deepak Kumar, Jeffrey T Hancock, and Zakir Durumeric. 2023. Hate raids on twitch: Echoes of the past, new modalities, and implications for platform governance. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–28.

[15] Tom Hartvigsen and et al. 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)*.

[16] 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. arXiv:2203.09509 [cs.CL] https://arxiv.org/abs/2203.09509

[17] Hello Partner. 2022. The Dangerous Downfall of a Toxic Twitch Audience. (November 2022). https://hellopartner.com/2022/11/08/the-dangerous-downfall-of-a-toxic-twitch-audience/ Accessed: 2024-10-01.

[18] Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large Language Models Are Reasoning Teachers. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:254877399

[19] Zhanhao Hu, Julien Piet, Geng Zhao, and David Wagner. 2024. Toxicity Detection for Free. *Advances in Neural Information Processing Systems* (2024).

[20] Saahil Jain et al. 2021. Token-level toxicity detection in online conversations. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–30.

[21] Jialun Aaron Jiang, Charles Kiene, Skyler Middler, Jed R. Brubaker, and Casey Fiesler. 2019. Moderation Challenges in Voice-based Online Communities on Discord. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 55 (Nov. 2019), 23 pages. doi:10.1145/3359157

[22] Jaeheon Kim, Donghee Yvette Wohn, and Meeyoung Cha. 2022. Understanding and identifying the use of emotes in toxic chat on Twitch. *Online Social Networks and Media* 27 (2022), 100180.

[23] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Boot-strapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. arXiv:2301.12597 [cs.CV] https://arxiv.org/abs/2301.12597

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[26] Tinh Luong, Thanh-Thien Le, Linh Ngo, and Thien Nguyen. 2024. Realistic Evaluation of Toxicity in Large Language Models. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 1038–1047. doi:10.18653/v1/2024.findings-acl.61

[27] Jihyung Moon, Dong-Ho Lee, Hyundong Justin Cho, Woojeong Jin, Chan Young Park, Min-Woo Kim, Jonathan May, Jay Pujara, and Sungjoon Park. 2023. Analyzing Norm Violations in Live-Stream Chat. *ArXiv* abs/2305.10731 (2023). https://api.semanticscholar.org/CorpusID:258762439

[28] Korosh Moosavi, Elias Martin, Muhammad Aurangzeb Ahmad, and Afra Mashhadi. 2024. E2T2: Emote Embedding for Twitch Toxicity Detection. In *Companion of the 2024 Computer-Supported Cooperative Work and Social Computing (CSCW Companion '24)* (San Jose, Costa Rica). ACM, New York, NY, USA, 6. doi:10.1145/3678884.3681840

[29] Thanh Thi Nguyen, Campbell Wilson, and Janis Dalins. 2023. Fine-Tuning Llama 2 Large Language Models for Detecting Online Sexual Predatory Chats and Abusive Texts. arXiv:2308.14683 [cs.CL] https://arxiv.org/abs/2308.14683

[30] Ines Njeh et al. 2025. LLM-based toxicity detection: Opportunities and challenges. *Computational Linguistics* (2025).

[31] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*. 145–153.

[32] Yuto Oikawa, Yuki Nakayama, and Koji Murakami. 2022. A Stacking-based Efficient Method for Toxic Language Detection on Live Streaming Chat. In *Conference on Empirical Methods in Natural Language Processing*.

[33] Aisha Powell and Dana Williams-Johnson. 2023. "You dumb cracker b* tch": The legitimizing of White supremacy during a Twitch ban of HasanAbi. *New Media & Society* (2023), 14614448231191776.

[34] M Poyane et al. 2018. Toxic comment detection in online discussions. *Deep Learning Indaba* (2018).

[35] Joni Salminen, Sercan Sengün, Juan Corporan, Soon-gyo Jung, and Bernard J. Jansen. 2020. Topic-driven toxicity: Exploring the relationship between online toxicity and news topics. *PLOS ONE* 15, 2 (02 2020), 1–24. doi:10.1371/journal.pone.0228723

[36] Joseph Seering and Sanjay R. Kairam. 2022. Who Moderates on Twitch and What Do They Do? Quantifying Practices in Community Moderation on Twitch. *Proc. ACM Hum.-Comput. Interact.* 7, GROUP, Article 18 (Dec. 2022), 18 pages. doi:10.1145/3567568

[37] Miriah Steiger, Timir J Bharucha, Sukrit Venkatagiri, Martin J Riedl, and Matthew Lease. 2021. The psychological well-being of content moderators: the emotional labor of commercial moderation and avenues for improving support. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–14.

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv* abs/2302.13971 (2023). https://api.semanticscholar.org/CorpusID:257219404

[39] Twitch Interactive, Inc. 2023. Twitch Community Guidelines. Twitch Safety Center. https://safety.twitch.tv/s/article/Community-Guidelines?language=en_US Last updated September 2023.

[40] Unitary AI. 2024. Detoxify. https://github.com/unitaryai/detoxify.

[41] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:254877569

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1800, 14 pages.

[43] Johannes Welbl, Amelia Glaese, Jonathan Uesato, et al. 2021. Challenges in detoxifying language models. *arXiv preprint arXiv:2109.07445* (2021).

[44] Donghee Yvette Wohn. 2019. Volunteer Moderators in Twitch Micro Communities: How They Get Involved, the Roles They Play, and the Emotional Labor They Experience. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300390

[45] Donghee Yvette Wohn. 2019. Volunteer moderators in twitch micro communities: How they get involved, the roles they play, and the emotional labor they experience. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[46] Jie Yang et al. 2023. ToxBuster: A hybrid LLM-human approach to toxicity detection. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023).

[47] Hao Zhu et al. 2024. Advances in real-time toxicity detection. *Comput. Surveys* (2024).

A Random Forest & SVM Details

Table 6 summarizes the classifier configurations and evaluation protocol. We use repeated stratified cross-validation to ensure robust performance estimates across class imbalances. All random seeds are fixed for reproducibility.

Table 6: Classifier configurations and evaluation protocol for ToxiTwitch hybrid model.

Component	Configuration
Random Forest	
n_estimators	100
class_weight	balanced
random_state	42
Linear SVM	
class_weight	balanced
max_iter	5000
dual	auto
random_state	42
Evaluation Protocol	
cross-validation	RepeatedStratifiedKFold
n_splits	5
n_repeats	3
total evaluations	15
random_state	42