BEYOND ONE-SIZE-FITS-ALL: PERSONALIZED HARMFUL CONTENT DETECTION WITH IN-CONTEXT LEARNING

A PREPRINT

Rufan Zhang
University of Science and Technology of China

Lin Zhang
University of Science and Technology of China

Xianghang Mi
University of Science and Technology of China

The proliferation of harmful online content—e.g., toxicity, spam, and negative sentiment—demands robust and adaptable moderation systems. However, prevailing moderation systems are centralized and task-specific, offering limited transparency and neglecting diverse user preferences—an approach ill-suited for privacy-sensitive or decentralized environments. We propose a novel framework that leverages in-context learning (ICL) with foundation models to unify the detection of toxicity, spam, and negative sentiment across binary, multi-class, and multi-label settings. Crucially, our approach enables lightweight personalization, allowing users to easily block new categories, unblock existing ones, or extend detection to semantic variations through simple prompt-based interventions—all without model retraining. Extensive experiments on public benchmarks (TextDetox, UCI SMS, SST2) and a new, annotated Mastodon dataset reveal that: (i) foundation models achieve strong cross-task generalization, often matching or surpassing task-specific fine-tuned models; (ii) effective personalization is achievable with as few as one user-provided example or definition; and (iii) augmenting prompts with label definitions or rationales significantly enhances robustness to noisy, real-world data. Our work demonstrates a definitive shift beyond one-size-fits-all moderation, establishing ICL as a practical, privacy-preserving, and highly adaptable pathway for the next generation of user-centric content safety systems. To foster reproducibility and facilitate future research, we publicly release our code on GitHub[1] and the annotated Mastodon dataset on Hugging Face[2].

1 Introduction

The integrity of online ecosystems is persistently challenged by a spectrum of harmful content, from overtly toxic language and manipulative spam to subjectively negative speech. The societal impact of such content is severe, such as corroding healthy discourse, inflicting psychological harm, facilitating the spread of misinformation, and fracturing digital communities [6, 16, 21, 22, 51]. As a result, harmful content detection has become a critical task for both academic research and industry applications. Traditional supervised approaches, such as deep neural classifiers [10, 46, 49] have achieved high performance on harmful content benchmarks. However, these methods rely on large-scale annotated datasets that are domain- and task-specific. Once deployed, without costly retraining, they struggle to adapt to new categories of harmfulness, user preferences, or concept shifts.

The recent ascendancy of large language models (LLMs) offers a promising alternative through in-context learning (ICL), which adapts to new tasks dynamically using task descriptions and few-shot demonstrations [13, 15, 57]. While preliminary studies have explored ICL for tasks like toxicity detection [7], they remain confined to narrow, single-task settings on curated benchmarks. Consequently, three critical questions remain largely unaddressed: **(1) Can ICL generalize effectively across heterogeneous harmful content tasks? (2) Can it be efficiently personalized to align with individual user preferences?** and **(3) How robust is ICL when faced with the noisy, ambiguous, and multi-faceted nature of real-world "wild" data?**

In this paper, we present the first systematic study of ICL for **personalizable and cross-task harmful content detection**. We unify three subtasks—binary harmfulness detection, multi-class categorization (`toxic`, `spam`, `negative`), and multi-label classification—into a single evaluation framework, enabling us to assess cross-task generalization. Beyond these canonical settings, we introduce three personalization scenarios inspired by real-world moderation: (i) *blocking new*

[1] `https://github.com/ChaseSecurity/personalizable_harmful_content_detection`
[2] `https://huggingface.co/datasets/ChaseLabs/Harmful-Texts-On-Mastodon`

categories, (ii) *unblocking existing categories*, and (iii) *blocking perturbed variations*. To further validate robustness, we extend evaluation to *wild data* collected from Mastodon, where content distributions are noisier and more ambiguous.

Specifically, we implement our framework by systematically comparing multiple foundation model families (e.g., LLaMA-3, Mistral, Qwen2) under zero-, few-, and many-shot ICL settings, combined with diverse demonstration retrieval strategies (random, lexical retrieval, and semantic retrieval) and prompt designs (with/without definitions or decision rationales). On benchmark datasets, our unified ICL formulation achieves strong cross-task generalization: for instance, in binary harmful text classification, LLaMA-3-8B with random 128-shot prompts attains 97.1% F1 on TextDetox Toxicity, 97.0% on UCI SMS Spam, and 95.9% on SST2 sentiment, competitive with or surpassing task-specific baselines. In multi-task binary and multi-task multi-class tasks, LLaMA-3-8B with 196 shots achieves 91.8% F1 and 93.8% Weighted avg F1, respectively.

For personalization, we design prompt-level interventions that allow users to modify filtering categories with minimal effort. Adding only a few user-provided examples or definitions enables the model to reliably block or unblock categories.

Finally, we evaluate robustness in a new Mastodon dataset annotated under binary, multi-class, and multi-label schemes. Results reveal a significant gap between curated benchmarks and real-world noisy data: for instance, in multi-task binary, LLaMA-3-8B with random 48-shot prompts' F1 drops from 88.7% on benchmarks to 79.5% in wild settings. However, replacing demos with wild data and incorporating reason-augmented prompts mitigates performance loss and yields more interpretable outputs(87.5% F1).

Together, these advances move beyond the one-size-fits-all paradigm and highlight ICL as a practical and privacy-preserving solution for user-centric and cross-task harmful content detection. our main contributions are threefold:

- **Unified Multi-Task Formulation:** We introduce a comprehensive evaluation framework that unifies binary, multi-class, and multi-label detection of toxicity, spam, and negative sentiment, demonstrating strong cross-task generalization by foundation models.

- **Lightweight Personalization Mechanisms:** We design and evaluate novel prompt-based interventions for users to effortlessly block new categories, unblock existing ones, and defend against adversarial variations, requiring minimal supervision.

- **Wild Data Evaluation and Robustness Analysis:** We construct and release a new annotated dataset from Mastodon, enabling the first large-scale evaluation of personalized detection in a decentralized context. Our analysis reveals the limitations of benchmark-only evaluation and identifies rationale-augmented prompts as a key technique for improving robustness.

The remainder of this paper is organized as follows: We review preliminaries in Section 2 and detail our experimental setup in Section 3. Sections 4, 5 and 6 present our results on single-task, multi-task binary, and multi-task multi-class detection, respectively. Section 7 explores personalization scenarios, and Section 8 evaluates performance on wild data. We discuss implications and limitations in Section 9 and conclude in Section 10.

2 Preliminaries

2.1 Defining Harmful Texts

We define *harmful texts* as content that can undermine individual well-being or platform integrity, typically manifested through offensive, manipulative, or otherwise negative language.

Toxicity encompasses language involving harassment, hate speech, profanity, or offensive expressions. Its societal impact is well-documented, with benchmarks like the Wikipedia Toxic Comments dataset [17] underlining its prevalence and importance for online communities. Spam encompasses unsolicited or manipulative content, often characterized by promotional intent, repetitive links, or coordinated amplification. Classical spam detection research dates back to early work on email and web spam [12, 53], and continues to be relevant for social media platforms where adversarial actors exploit open publishing mechanisms. Negative sentiment, while not overtly abusive, captures emotionally harmful expressions such as anger, frustration, or despair. Although often excluded from traditional harmfulness definitions, negative sentiment has psychosocial consequences for online well-being [42]. Widely adopted sentiment analysis datasets such as SST-2 [52] provide a standardized basis for studying this dimension.

Together, these categories capture three major axes of online harm: abusive harms (toxicity), economic and functional disruption (spam), and psychosocial harms (negative sentiment). This tripartite scheme allows us to ground our experiments in well-established benchmarks while reflecting the heterogeneity of real-world harmful discourse.

2.2 Limitations of Prior Work

Traditional approaches to harmful text detection relied heavily on keyword-based filters or classical machine learning classifiers trained on handcrafted features [51]. While straightforward, these methods suffer from low recall, brittleness to adversarial rephrasing, and poor adaptability to domain shifts. The advent of deep learning introduced neural text classifiers trained on large annotated datasets, achieving strong performance in within-domain settings [10, 62]. However, subsequent studies revealed that such models generalize poorly across domains or platforms [9], limiting their practical reliability.

Recent advances in large language models (LLMs) have spurred interest in adapting them for harmful content detection. Fine-tuned LLMs have shown strong results in toxicity [25], spam [35] and sentiment detection [46], while prompting-based methods reduce the reliance on large labeled datasets [7, 27, 61]. Nevertheless, most existing works focus on narrow problem settings, such as binary harmful detection within a single task, while overlooking more comprehensive scenarios that span multiple tasks and involve binary, multi-class, or multi-label classification. Moreover, prior studies typically optimize for within-domain benchmarks, without explicitly testing generalization to real-world "wild data," which is noisier, more diverse, and less neatly separable into canonical categories.

2.3 In-Context Learning

In-context learning (ICL) is a technique that enables large language models to adapt to new tasks by conditioning on a task description and a small set of demonstrations, without the need for parameter updates. Mathematically, given a task description t and a set of k demonstrations $D_k = \{(x_i, y_i)\}_{i=1}^{k}$, where x_i represents the input and y_i is the corresponding label, ICL predicts the label y_{query} for a query x_{query} as follows:

$$y_{query} = M(t, D_k, x_{query}),$$

where M denotes the large language model. This approach allows for rapid task adaptation and showcases the model's ability to learn from examples provided within the context.

A growing body of research highlights that in-context learning (ICL) performance is highly sensitive to design choices across multiple dimensions. Model choice [14, 15, 57] is one important factor: instruction-tuned and larger LLMs consistently exhibit stronger generalization and more robust few-shot behavior. Retrieval strategy [41, 50] also matters, as demonstrations can be selected randomly, by lexical similarity (e.g., BM25 [48]), or by semantic similarity (e.g., Sentence-Transformers [47]), with each approach shaping the inductive bias of ICL. The number of demonstrations [39, 43] is another critical variable: while increasing the number of shots generally improves stability, studies show diminishing returns and even saturation beyond certain thresholds.

Equally important is the design of the task description and output format. Prior work demonstrates that subtle variations in prompt phrasing can drastically alter performance, and structured outputs help models adhere more reliably to task requirements [31, 60]. Finally, the addition of rationales—natural language explanations paired with labels—has been shown to enhance reasoning and reduce error propagation [36, 44, 58], providing interpretive signals that guide the model beyond surface-level pattern recognition.

Our work builds on these insights by systematically examining how these dimensions shape ICL for harmful content detection. Unlike prior research, we unify evaluation across binary, multi-class, and multi-label formulations; explore personalization strategies via three common personalization scenarios; and extend evaluation to Mastodon wild data, where domain shift, noise, and label overlap provide a realistic stress test for ICL robustness.

3 Experimental Setup: Tasks, Datasets, and In-Context Learning Configuration

Building upon the foundation laid in Section 2, we now detail our experimental setup for evaluating multi-task and personalized harmful content detection via in-context learning (ICL). This section is organized as follows: we first describe the datasets (Section 3.1), then formally define the learning tasks (Section 3.2), and finally specify the ICL configurations and parameters (Section 3.3).

3.1 Datasets

Building upon the categories of harmful content—toxicity, spam, and negative sentiment—introduced in Section 2, we have meticulously curated representative, publicly available datasets to facilitate model development and evaluation. These datasets form the cornerstone for constructing and benchmarking our detection models. For datasets with predefined train/test splits, we adhere to these standard divisions to ensure comparability with prior research. For datasets lacking official splits, we employ an 80/20 random stratified split, allocating 80% of the data for training (used to select

Table 1: Overview of datasets.

Task	Dataset	Language	Label	Train	Test
Toxicity	Textdetox	English	Toxic	2,000	500
			Non-toxic	2,000	500
Spam	UCI	English	Spam	586	161
			Ham	3,873	954
Sentiment	SST2	English	Negative	29,780	428
			Positive	37,569	444

demonstration examples for prompting) and reserving 20% for testing. This approach ensures a balanced and representative evaluation framework.

Table 1 provides a summary of the selected datasets. Next, we delve into the specifics of each dataset, highlighting their collection methodologies, quality attributes, and the rationale for their selection over alternative datasets.

TextDetox [19]. TextDetox is a meticulously curated dataset tailored for binary toxicity classification across multiple natural languages. As of this writing, it supports 15 languages. For each languages, the dataset comprises 5,000 samples per language, evenly distributed between toxic and non-toxic content. TextDetox is constructed by carefully selecting high-quality subsets from existing toxicity corpora, ensuring its utility as a reliable resource for training and evaluating toxicity detection models.

In our experiments, we utilize only the English subset of TextDetox, which is derived from the Jigsaw Toxic Comment Classification Challenge (Jigsaw [30]), a widely recognized benchmark for toxicity detection. The Jigsaw dataset contains over 200,000 comments, manually annotated by multiple human raters for six toxicity categories: toxic, severe toxic, obscene, threat, insult, and identity hate. Its large scale, multi-label structure, and real-world relevance make it an invaluable resource for developing and benchmarking toxicity detection models. Furthermore, its extensive adoption in academic research [18, 23, 33] underscores its status as a standard benchmark for toxicity classification tasks.

The decision to use TextDetox instead of directly employing Jigsaw is motivated by two key considerations. First, TextDetox enhances data quality through rigorous manual screening when selecting samples from Jigsaw, ensuring a cleaner and more reliable dataset. Second, the inclusion of non-English samples in TextDetox provides a valuable opportunity to evaluate our detection models on out-of-domain datasets, thereby enabling a more comprehensive assessment of their generalization capabilities in the future. This dual advantage makes TextDetox an ideal choice for our experiments.

UCI [4]. The UCI SMS Spam Collection is a widely recognized dataset curated for SMS spam detection research. It comprises 5,574 English SMS messages, each annotated as either spam or ham (legitimate). The dataset aggregates messages from multiple reputable sources, including the Grumbletext website (425 manually extracted spam messages from a UK forum), the NUS SMS Corpus (3,375 randomly selected legitimate messages from a large collection at the National University of Singapore), Caroline Tag's PhD thesis (450 legitimate messages), and the SMS Spam Corpus v.0.1 Big (1,002 legitimate and 322 spam messages). This diverse origin ensures that the dataset captures a wide range of real-world texting behaviors and spam types, enhancing its representativeness and utility for spam detection research.

The manual selection process employed in curating the dataset further ensures high data quality, making it a reliable benchmark for evaluating spam detection models. Since its public release in 2012, the dataset has been extensively used in academic research, serving as a standard benchmark for SMS spam detection tasks. Its adoption in numerous studies [1, 5, 8, 40, 49] underscores its significance in the field and ensures that findings based on this dataset are comparable and reproducible across different works.

SST2 [52]. The Stanford Sentiment Treebank (SST2) is a benchmark dataset widely utilized for binary sentiment classification. It comprises 70,042 movie review sentences annotated as either positive or negative, with 67,349 samples designated for training, 872 for validation, and 1,821 for testing. In our experiments, we use the original training split for generating demonstration examples and the validation split for testing. This ensures that the test set size for sentiment analysis aligns with the other two categories, thereby avoiding over-representation when calculating the performance of multi-task detection models.

The dataset originates from 11,855 movie reviews, with 215,154 phrases within their parse trees manually annotated with sentiment labels. This granular annotation enables models to capture the compositional nature of sentiment within sentence structures, making SST2 a high-quality resource for sentiment analysis. Compared to other sentiment datasets, SST2 distinguishes itself through its detailed phrase-level annotations and syntactic structure information, which facilitate

the learning of complex linguistic patterns. Furthermore, as part of the GLUE benchmark, SST2 has been extensively adopted for evaluating NLP models [2, 29, 37, 46], ensuring that research findings are both comparable and reproducible across studies.

To further assess the generalization capabilities of our models, we incorporate out-of-domain test data sourced from Mastodon, a widely recognized federated social network platform. This data corresponds to the three categories of harmful content under investigation. The selection of this dataset aims to rigorously challenge the models' robustness and adaptability when confronted with unseen data distributions. Comprehensive details regarding the dataset and the associated evaluation methodologies will be presented in Section 8.

3.2 Learning Tasks

Building upon the datasets introduced earlier, we now provide a formal definition of the learning tasks designed to enable multi-task and personalized harmful content detection using in-context learning (ICL). The evaluation of these tasks will be detailed in subsequent sections.

Let $\mathcal{X}$ denote the input space, $\mathcal{Y}$ the label space, and t a natural-language task instruction. We define $D_k^r = \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\} \subseteq D$ as a set of k demonstration examples retrieved from dataset D under retrieval strategy r. Let $x_{\text{query}} \in \mathcal{X}$ denote a new input under evaluation (i.e., the query sample), and let $\mathcal{M}$ denote the foundation model employed for in-context learning. The predicted output y_{query} is defined as:

$$y_{\text{query}} = \mathcal{M}(t,\, D_k^r,\, x_{\text{query}}).$$

This formulation represents the general ICL process. For classification tasks, we typically employ **greedy decoding** to select the most probable label from the model's vocabulary that matches a label in $\mathcal{Y}$. This can be formalized as:

$$y_{\text{query}} = \arg\max_{y \in \mathcal{Y}} P_{\mathcal{M}}\left(y \mid t,\, D_k^r,\, x_{\text{query}}\right),$$

where $P_{\mathcal{M}}$ represents the model's conditional probability distribution over its vocabulary.

Remark. In our experiments, we primarily adopt demonstrations of the form (x_i, y_i), i.e., input–label pairs. Only in the wild data experiments do we extend demonstrations with reasons (x_i, y_i, e_i), where e_i provides a textual rationale for the assigned label.

Single-Task Harmful Content Classification. Single-task harmful content classification independently detect each category of harmful content. In this approach, the label space $\mathcal{Y}$ corresponds to the specific harm category under consideration (e.g., *spam* and *ham* for spam detection), and the demonstration set D_k^r is exclusively sampled from the relevant training dataset (e.g., toxicity training datasets for toxicity detection). These independent ICL detectors are collectively referred to as single-task harmful content classifiers.

Multi-Task Harmful Content Classification. In contrast to single-task classification, multi-task harmful content classification aims to simultaneously detect multiple categories of harmful content. Here, the label space $\mathcal{Y}$ is defined as the union of all harm categories under consideration. The demonstration set D_k^r is sampled from all relevant training datasets, encompassing examples from each harm category. Depending on the structure of the label space $\mathcal{Y}$, this task can be further divided into two subcategories: multi-task *binary* ICL and multi-task *multi-class* ICL.

In multi-task binary ICL, the model distinguishes harmful content from benign content, with the label space defined as $\mathcal{Y} = \{\textit{harmful}, \textit{benign}\}$, where *harmful* represents harmful(toxic, spam, negative) content and *benign* represents non-harmful content. Conversely, in multi-task multi-class ICL, the model assigns harmful content to one of the specific harm categories, with the label space defined as $\mathcal{Y} = \{\textit{toxic}, \textit{spam}, \textit{negative}, \textit{benign}\}$.

As mentioned in preliminaries(section 2), the performance of an ICL classifier is influenced by various hyperparameters, including but not limited to the number and retrieval strategy of demonstration examples (k), the task instruction (t), and the choice of foundation model ($\mathcal{M}$). In the following section, we provide a comprehensive overview of the ICL hyperparameters to be systematically explored for each harmful content detection task.

3.3 General ICL Settings

Foundation Models. In our experiments, we evaluate three state-of-the-art open-weight large language model (LLM) families: Llama [56], Mistral [28], and Qwen [11]. The selection of specific model configurations (e.g., instruction-tuned variants, model sizes and versions) is guided by multiple criteria, including performance on diverse NLP benchmarks, support for long-context inferences, and computational efficiency on our available hardware resources.

For our study, we adopt the latest instruction-tuned versions available as of July 2024: Llama-3.1-8B-Instruct [3], Mistral-7B-Instruct-v0.3 [54], and Qwen2-7B-Instruct [55]. These models support maximum context lengths of 128K, 32K, and

128K tokens, respectively. To ensure compatibility across all tasks, we constrain the prompt length to remain within 32K tokens. This selection balances cutting-edge performance with practical considerations for efficient evaluation on our computing infrastructure.

As of July 2025, newer releases of the Llama, Mistral, and Qwen families have become available. However, these models are either not well aligned with the requirements of our current evaluation setup or show no significant performance gains on our tasks compared to the versions we adopt in this study. We therefore retain the above models as our primary evaluation suite. Further empirical details are provided in Appendix A.

Prompt Template. As illustrated in Figure 1 and discussed in Section 3.2, our prompt consists of three components: the task description t, the demonstration set D_k^r, and the query input x_{query}. Varying the formulation of t and D_k^r plays a critical role in shaping ICL performance.

Demonstration-Relevant Parameters. We systematically study demonstration-related hyperparameters. First, we vary the number of demonstrations k (e.g., $k \in \{0, 1, 2, 4, 8, 16, 32, 64, 128\}$ in single-task binary tasks), thereby covering the full spectrum from zero-shot to many-shot settings. Unless otherwise noted, demonstrations are balanced across all classes in $\mathcal{Y}$.

Second, we evaluate the effect of different retrieval strategies for selecting D_k^r. We consider five methods: **Random**, which uniformly samples an equal number of examples per label from the training set, averaging over three random seeds for robustness; **Semantic**, which retrieves semantically similar examples for each query using dense sentence embeddings from `sentence-transformers` [47], implemented with the `retriv` toolkit[3]; **Balanced Semantic**, which combines semantic similarity with label balancing; **Lexical**, which selects examples based on surface-level overlap using BM25 [48], also implemented in `retriv`; and **Balanced Lexical**, which enforces class balance on top of BM25 retrieval.

Finally, we extend demonstrations beyond input–label pairs (x_i, y_i) to **reason-augmented demonstrations** (x_i, y_i, e_i), where e_i is a natural-language explanation of why x_i belongs to class y_i. In our study, such reason-augmented demonstrations are employed only in wild data experiments, where interpretability and robustness to noisy annotations are most critical.

Task Description. We design two levels of task descriptions. **Level 1** provides a concise statement of the task and its label space, while **Level 2** augments this with detailed definitions for each label. This design allows us to evaluate whether richer task information improves ICL performance. Unless otherwise stated, Level 1 descriptions are used.

Inference Setting. For efficient deployment of foundation models, we employ *vLLM* [34], a high-performance inference engine optimized for large-scale LLMs. We adopt GuidedDecodingParams to constrain models to **structured outputs**, i.e., predictions restricted to the label space without generating irrelevant natural-language content. To ensure deterministic and reproducible results, we set the temperature to zero and disable stochastic sampling by configuring top-p to 1 and top-k to -1. Given the cumulative logits for candidate labels, we normalize them with a softmax function to obtain prediction probabilities.

```
<Task Description>
==
Query: <sentence 1>
Answer: <label>
==
Query: <sentence 2>
Answer: <label>
==
...
==
Query: <query text>
Answer:
```

Figure 1: Prompt template for in-context learning (ICL).

[3] `https://github.com/AmenRa/retriv`

```
#### Positioning
- Assistant Name: Toxic text Detector
- Primary Task: Classify whether a given text is toxic or benign.

#### Capabilities
- Message Analysis: Accurately analyze the content and characteristics of text.
- Toxicity Classification: Determine whether a text is 'toxic' or 'benign'.

#### Knowledge Base
- Message Categories:
  - toxic
  - benign

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('toxic' or 'benign'), without any additional
    explanation.
```

Figure 2: The task description of single-task ICL (Toxicity).

4 Single-Task ICL

We begin our evaluation by establishing strong single-task baselines. This allows us to assess the fundamental capability of ICL-equipped foundation models for each harm category independently, before investigating more complex multi-task and personalized settings.

4.1 Experimental Setup

Models. In single-task binary classification scenarios, we assess the performance of three instruction-tuned foundation models: *Llama-3.1-8B-Instruct*, *Mistral-7B-Instruct-v0.3*, and *Qwen2-7B-Instruct*. Formally, let $\mathcal{M}$ denote the set of models under evaluation, where $\mathcal{M} = \{$*Llama-3.1-8B-Instruct*, *Mistral-7B-Instruct-v0.3*, *Qwen2-7B-Instruct*$\}$.

Datasets. As summarized in Table 1, each dataset D consists of training data D_{train} and testing data D_{test}. Demonstrations $D_k^r \subseteq D_{\text{train}}$ are retrieved to construct prompts, while D_{test} is directly used for evaluation. Accordingly, the input space in each single-task setting is defined as $\mathcal{X} = D_{\text{test}}$.

Label Names. The label spaces are defined as follows: **Toxicity**: $\mathcal{Y} = \{$*toxic*, *benign*$\}$; **Spam**: $\mathcal{Y} = \{$*spam*, *ham*$\}$; **Sentiment**: $\mathcal{Y} = \{$*negative*, *positive*$\}$.

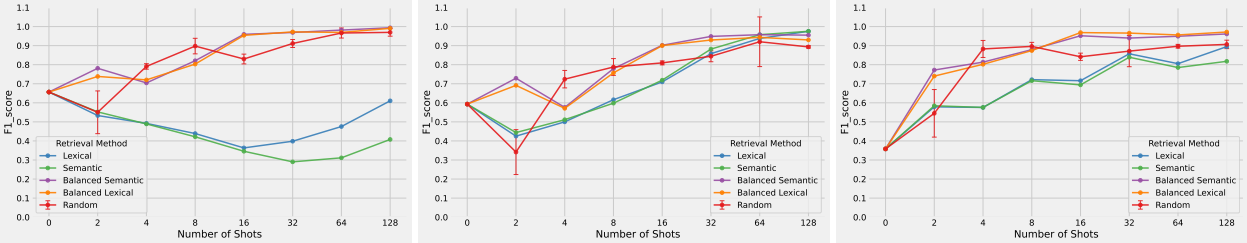
Number of Shots. We experiment with $k \in \{0, 2, 4, 8, 16, 32, 64, 128\}$ demonstrations, selected according to retrieval strategy r, to analyze the impact of shot number.

Retrieval Strategies. As described in Section 3.3, we evaluate five retrieval strategies: *Random*, *Semantic*, *Balanced Semantic*, *Lexical*, and *Balanced Lexical*.

Prompt Design. Figure 2 illustrates the task description for Toxicity. Descriptions for Spam and Sentiment follow the same format and are provided in Appendix B.

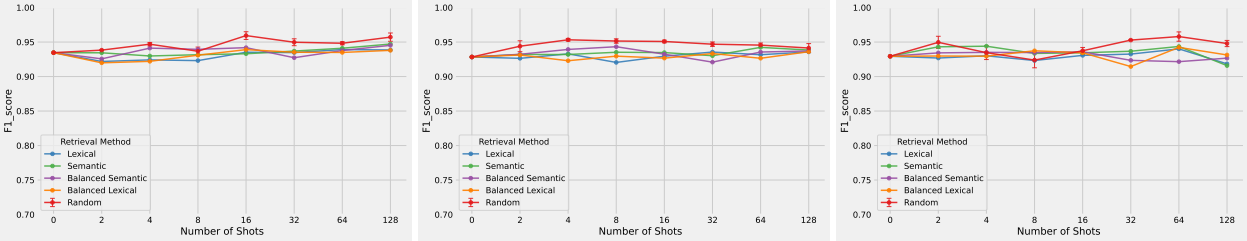
Metrics. Typical classification Metrics are adopted, including recall, precision, F1-score, and accuracy. Also, as fake alarm rate is a critical metric in content moderation, we also measure the metric of false positive rate.

Baselines. Prior studies [20, 24, 32] have demonstrated that fine-tuned BERT models remain strong baselines for a variety of NLP classification tasks. In particular, BERT-based approaches have achieved competitive or even state-of-the-art results on the datasets we use in this work: TextDetox for toxicity detection [26], SST-2 for sentiment classification [45], and the UCI SMS Spam Collection for spam detection [38]. To provide a fair and informative comparison, we adopt `BERT-base-uncased` as our supervised baseline. The model is fine-tuned separately on each dataset using its respective training set and evaluated on the same test sets as used in our ICL experiments. This allows us to directly contrast the performance of traditional fine-tuning against in-context learning under identical data splits, thereby highlighting the efficiency and adaptability of ICL relative to established supervised methods.



(a) Llama (b) Mistral (c) Qwen

Figure 3: The performance of ICL on Binary Spam Classification



(a) Llama (b) Mistral (c) Qwen

Figure 4: The performance of ICL on Binary Sentiment Analysis

4.2 Results

Spam. As shown in Figure 3, ICL performance on the UCI spam dataset generally improves with more demonstrations, though saturation occurs beyond $k = 32$. Balanced Semantic and Balanced Lexical retrieval consistently yield the best results across all models, closely followed by Random. In contrast, Lexical and Semantic retrieval without label balancing exhibit instability, especially for Llama. For model-specific results, Llama achieves the best F1-score of 0.991 (Balanced Lexical), Qwen reaches 0.971 (Balanced Lexical), and Mistral achieves 0.957 (Balanced Semantic).

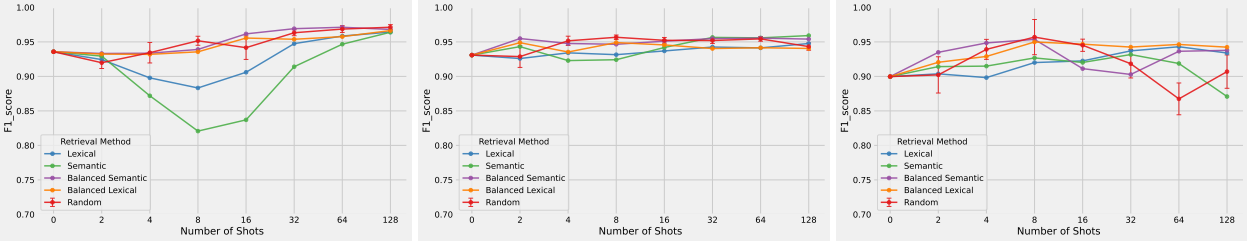
Sentiment. Figure 4 shows that all models achieve near-ceiling performance on SST-2, with F1-scores consistently above 0.9 regardless of retrieval strategy or number of shots. This suggests that sentiment classification is relatively easy for instruction-tuned models. Random retrieval shows a slight advantage as k increases, but overall differences between strategies are marginal.

Toxicity. As shown in Figure 5, performance on TextDetox improves with more demonstrations but exhibits less stability than UCI. Some retrieval strategies degrade at higher k, e.g., Mistral–Random and Qwen–Semantic. Top-performing strategies vary across models, but Random and Balanced Semantic are consistently competitive. Similar to UCI, Llama displays poor performance under Lexical and Semantic retrieval, whereas Random and Balanced Semantic achieve the strongest results, with Llama reaching 0.971 F1.

Comparison with Baselines. Table 2 summarizes the best-performing ICL classifiers for all three tasks along with a direct comparison with the baselines (fine-tuned BERT models). Key observations show that ICL exhibits strong performance across different tasks, with certain setups outperforming the BERT baseline. For instance, Llama with Balanced Semantic retrieval achieves an average F1-score of 0.970, which is higher than BERT's average of 0.959.

Specifically, in the spam classification task, most of Llama's top ICL configurations surpass BERT's performance. Similarly, for the toxicity task, Llama's best ICL results match or exceed BERT's performance. However, for the sentiment analysis task, BERT's F1-score of 0.923 is relatively lower compared to the near-ceiling performance of ICL models, which all achieve F1-scores above 0.9. This comparison indicates that ICL can be a competitive alternative to traditional fine-tuned models like BERT, showcasing its efficiency and adaptability in binary classification tasks.

Summary. Our results establish that ICL adapts effectively across diverse binary classification tasks for harmful content. Crucially, ICL models achieve performance comparable to or surpassing a fine-tuned BERT baseline across all three tasks, despite requiring no task-specific gradient updates and far less explicit supervision. This demonstrates ICL's strong potential as an efficient and highly adaptable alternative to traditional fine-tuning pipelines.



(a) Llama (b) Mistral (c) Qwen

Figure 5: The performance of ICL on binary toxicity classification

Table 2: Comparison of Best F1-score for Different Model-Retrieval Pairs Across Tasks

Model	Retrieval Method	Average	Spam	Sentiment	Toxicity
Bert(Baseline)	-	0.959	0.985	0.923	0.969
Llama	Random	0.967	0.970 (128)[1]	0.959 (16)	0.971 (128)
	Balanced Lexical	0.965	0.991 (128)	0.939 (16)	0.966 (128)
	Balanced Semantic	0.970	0.994 (128)	0.945 (128)	0.971 (64)
Qwen	Random	0.940	0.906 (128)	0.958 (64)	0.957 (8)
	Balanced Lexical	0.954	0.971 (128)	0.942 (64)	0.950 (8)
	Balanced Semantic	0.950	0.961 (128)	0.936 (16)	0.954 (8)
Mistral	Random	0.943	0.920 (64)	0.953 (4)	0.957 (8)
	Balanced Lexical	0.943	0.943 (64)	0.936 (128)	0.949 (8)
	Balanced Semantic	0.952	0.957 (64)	0.943 (8)	0.955 (64)

[1] **F1-score (Number of Shots)**: The F1-score achieved by the model-retrieval pair, followed by the number of shots used in that configuration.

5 Multi-Task Binary ICL

Having established strong single-task baselines, we now investigate the feasibility of ICL for **multi-task binary** harmful content classification. This setting tests the model's ability to learn a unified concept of "harmfulness" from heterogeneous categories (toxicity, spam, negative sentiment), a crucial step toward a general-purpose moderation system.

5.1 Experimental Setup

We only report experimental configurations that differ from the single-task setting (Section 4).

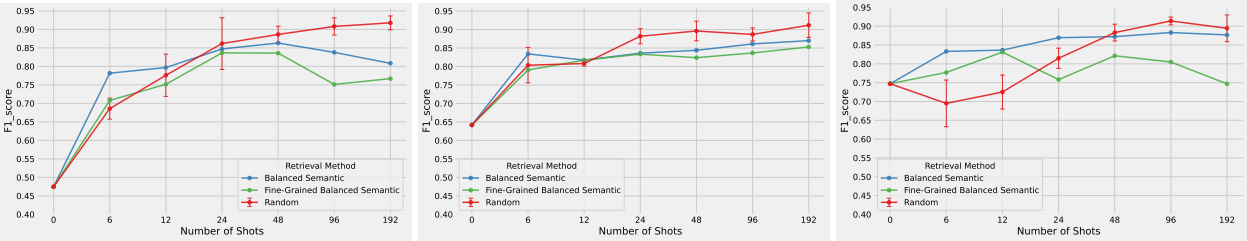
Datasets. As shown in Table 1, we merge the training data from all three binary datasets into a single training pool for retrieving demonstrations. Similarly, the testing data are consolidated into a unified evaluation set:

$$\mathcal{X} = D_{\text{test}}^{\text{TextDetox}} \cup D_{\text{test}}^{\text{UCI}} \cup D_{\text{test}}^{\text{SST2}}, \quad D_k^r \subseteq D_{\text{train}}^{\text{TextDetox}} \cup D_{\text{train}}^{\text{UCI}} \cup D_{\text{train}}^{\text{SST2}}.$$

Label Names. To construct a binary label space, we group $\{toxic, spam, negative\}$ as **harmful**, and $\{non\text{-}toxic, ham, positive\}$ as **benign**, yielding $\mathcal{Y} = \{harmful, benign\}$.

Number of Shots. To ensure balanced coverage of categories in the prompt, k is set as a multiple of the number of classes, i.e., $k \in \{0, 6, 12, 24, 48, 96, 192\}$.

Retrieval Strategies. Based on single-task results, we retain the two strongest strategies: Random and Balanced Semantic. Although Balanced Lexical also performed competitively, its mechanism closely resembles Balanced Semantic, which generally achieved better results; therefore, we omit it for efficiency. In the multi-task setting, Balanced Semantic is further extended to **Fine-grained Balanced Semantic**, which balances across six subcategories instead of the two superclasses.



(a) Llama (b) Mistral (c) Qwen

Figure 6: The performance of multi-task binary ICL, as measured by F1-score.

Table 3: Best-performing ICL setups for multi-task binary harm text classification, where each cell shows the best F1-score followed by the number of demonstration shots in parentheses.

Model	Retrieval Strategy		
	Random	Balanced Semantic	Fine-grained Balanced Semantic
Llama	**0.918(192)**	0.863(48)	0.836(24)
Mistral	0.912(192)	0.870(192)	**0.853(192)**
Qwen	0.914(96)	**0.883(96)**	0.831(12)

Task Description. Appendix B.2 shows two prompt levels: **Level 1** mirrors the single-task binary description and only specifies the binary label space, while **Level 2** additionally provides explicit definitions and subcategory examples. Unless otherwise specified, we default to Level 1 in this section.

5.2 Results

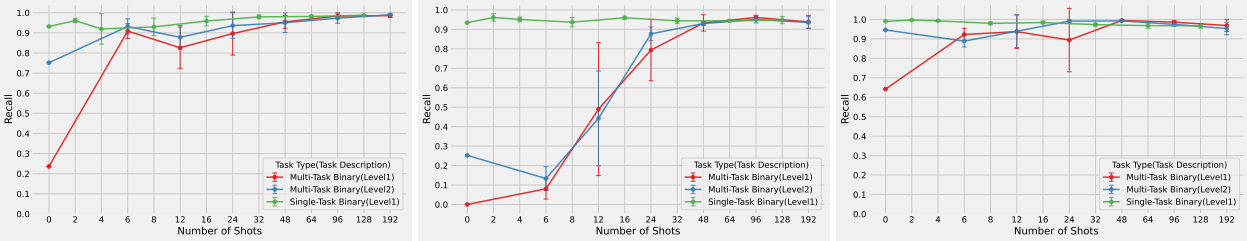
Comparison of Retrieval Strategies and Models. As shown in Figure 6, several consistent patterns emerge in different models. (1) Balanced Semantic generally outperforms Fine-grained Balanced Semantic, suggesting that enforcing balance at the superclass level is more beneficial than over-constraining to subcategories. (2) At small k, semantic-based strategies (Balanced and Fine-grained) outperform Random, but their advantage diminishes as k increases. At larger k, Random consistently surpasses semantic-based methods, indicating that diversity may be more important than semantic closeness when sufficient demonstrations are available. (3) The best overall results are achieved under the Random strategy.

Regarding performance difference among LLMs, as presented in Figure 6 and Table 3, Llama-Random achieves the strongest result with an F1-score of 0.918 at 192 shots. Qwen and Mistral follow closely, achieving best F1-scores of 0.914 and 0.912, respectively.

Performance Gap: Multi-Task vs. Single-Task. Although Llama-Random is the best-performing combination, its peak performance in multi-task ICL remains below that of single-task ICL. To investigate this discrepancy, we compare recall and false positive rates (FPR) between the two settings. As shown in Figure 7, recall is comparable or even higher in multi-task ICL, particularly on TextDetox, indicating that reduced performance is not due to missed harmful samples. Instead, Figure 8 shows that the FPR is substantially higher in multi-task ICL. This outcome is expected: when exposed to heterogeneous harmful categories, the model learns a broader notion of harmfulness, increasing the likelihood of misclassifying benign texts as harmful.

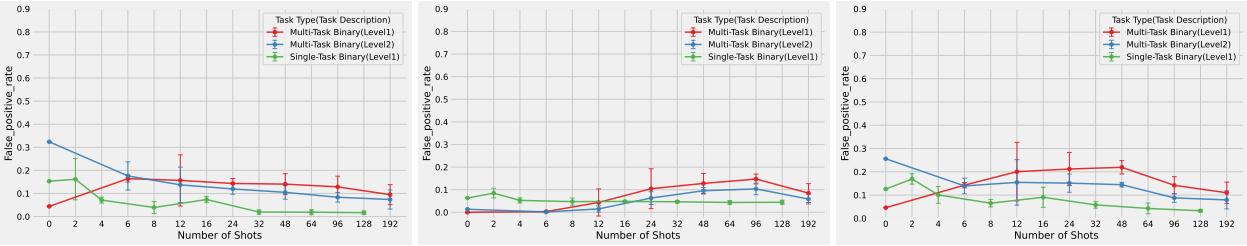
Mitigating False Positives with Enhanced Task Descriptions. To mitigate elevated FPR, we introduce Level 2 task descriptions with explicit definitions and examples of harmful and benign categories (Figure 24). As shown in Figures 7 and 8, Level 2 task descriptions significantly reduce FPR while preserving recall. Consequently, the best F1-score of Llama-Random improves from 0.918 (192 shots) to 0.934 (192 shots).

These findings demonstrate that richer task descriptions are crucial for helping models disambiguate harmfulness in multi-task contexts, significantly mitigating the false positive problem. While a performance gap with single-task models remains, this work establishes a strong foundation for unified harmful content detection.



(a) UCI (b) SST2 (c) Textdetox

Figure 7: The performance comparison between multi-task binary ICL and single-task binary ICL, as measured by recall.



(a) UCI (b) SST2 (c) Textdetox

Figure 8: The false positive rate for Llama-Random on multi-task binary ICL and single-task ICL

6 Multi-Task Multi-Class ICL

While binary classification identifies whether content is harmful, effective moderation and **personalization** often require understanding the *type* of harm. To this end, we extend our evaluation to **multi-task multi-class** ICL, where the model must distinguish between specific harmful categories (toxic, spam, negative) and benign content.

6.1 Experimental Setup

We only report the experimental configurations that differ from the Multi-Task Binary ICL (Section 5).

Label Names. We extend the binary setting into a four-class setting by keeping harmful categories distinct while merging non-toxic, ham, and positive into a single benign category. Formally, $\mathcal{Y} = \{toxic, spam, negative, benign\}$.

Retrieval Strategies. We follow the retrieval setting from Multi-Task Binary ICL. However, since this is a multi-class task with four distinct labels ($\mathcal{Y} = \{toxic, spam, negative, benign\}$), we retain the *Fine-grained Balanced Semantic* strategy, which enforces balance across all four classes, and discard the binary-level *Balanced Semantic* strategy.

Task Descriptions. Based on the findings in Multi-Task Binary ICL, where Level 2 prompts consistently improved performance, we also evaluate both Level 1 and Level 2 prompts here. To ensure efficiency, Level 2 prompts are only applied to the best-performing model-strategy pair.

Metrics. For overall performance, we report the weighted average F1-score across all categories. For per-class analysis, we adopt precision, recall, and F1-score, consistent with Single-Task Binary ICL (Section 4). We also examine confusion matrices for error analysis.

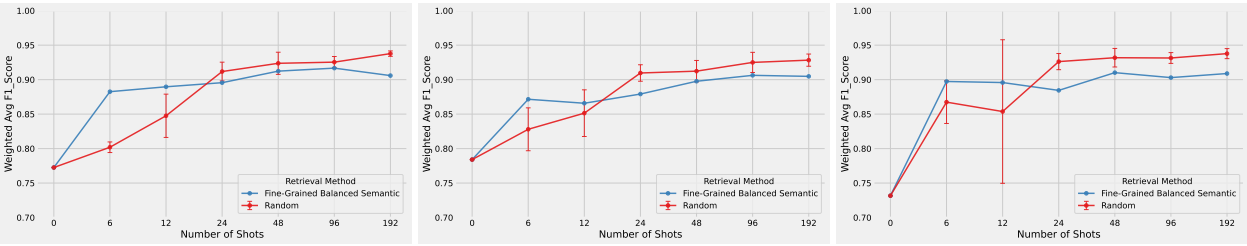
6.2 Results

Comparison between different models and retrieval strategies. As shown in Figure 9, we observe a consistent trend with the binary case: Fine-grained Balanced Semantic performs better than Random at small shot counts, while Random surpasses it as the number of shots increases and then stabilizes. Since we are primarily concerned with peak performance, we compare models under the Random strategy (Figure 10a). Qwen generally achieves slightly higher weighted F1-scores than Llama and Mistral, except at $k = 0$ and $k = 192$. Both Qwen-Random and Llama-Random reach the highest weighted F1-score 0.938 at 192 shots. Given Llama's consistently strong performance across tasks, we choose Llama-Random, rather than Qwen-Random, as the representative configuration for further analysis.

Fine-grained analysis of the best-performing ICL setup: Llama-Random. Figure 11a shows that F1-scores for all categories improve with larger shot counts and eventually stabilize. However, the *negative* category lags behind others. From Figures 11b and 11c, we attribute this to its low precision. In contrast, the benign category shows very high precision but relatively low recall. Examining the confusion matrix (Figure 12), we find that misclassified harmful texts (toxic, spam, negative) are more often predicted as benign rather than another harmful type. This indicates that the model effectively distinguishes different harmful categories once harmfulness is recognized, but it struggles more in deciding whether a text is harmful at all.

Moreover, more than half of the misclassified benign texts are predicted as negative, which explains the low precision of the negative category. This aligns with the intuition that negative sentiment is less saliently harmful compared to overt toxicity or spam, making it more semantically ambiguous and confusable with benign content. This inherent challenge underscores the difficulty of fine-grained harm classification and motivates the need for the more flexible multi-label formulation and a more strictly curated dataset we explore in Section 8.

Comparison between Level 1 and Level 2 task descriptions. Building on the success of Level 2 prompts in the binary setting, we evaluate their impact on the multi-class task. As shown in Figure 13, incorporating Level 2 task descriptions significantly boosts ICL performance when $k \geq 12$. The weighted average F1-score improves from 0.938 (192 shots) with Level 1 to 0.955 (192 shots) with Level 2. Comparing the confusion matrices in Figure 12, we find that Level 2 task descriptions notably reduce the false positive rate, at the cost of a slight increase in false negatives. Importantly, the model still maintains strong discrimination among harmful categories. This suggests that richer label definitions help the model refine its decision boundary between harmful and benign texts, thereby improving robustness in the multi-class setting.

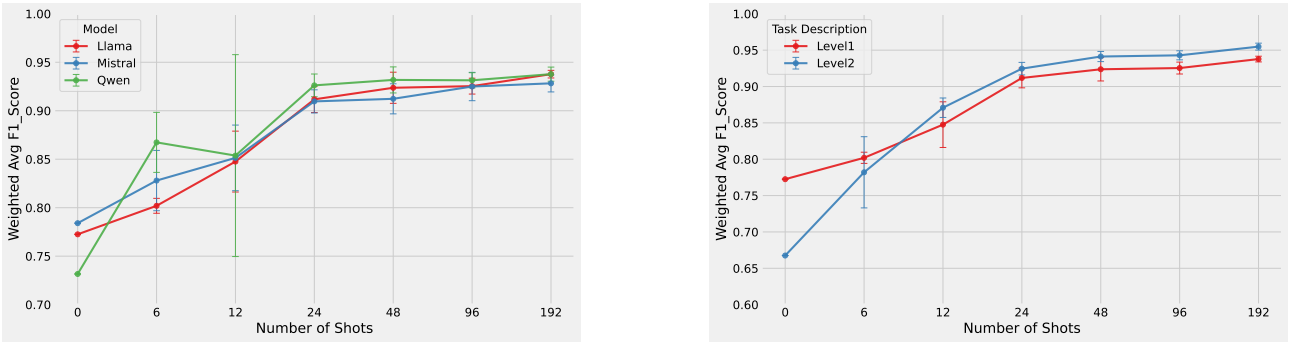


(a) Llama (b) Mistral (c) Qwen

Figure 9: The overall performance of Multi-task Multi-class ICL

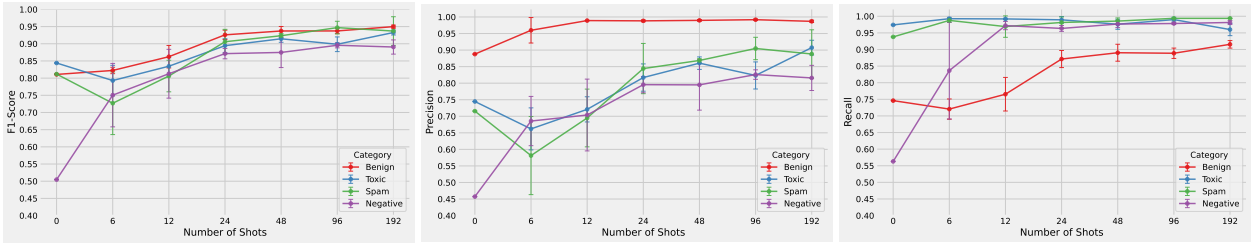
7 Personalization with ICL

Having established the strong multi-task performance of ICL for harmful content detection, we now investigate its potential for **personalization**. A key limitation of centralized moderation systems is their "one-size-fits-all" approach, which fails to accommodate diverse user preferences and contexts. We therefore address a critical question: *Can ICL be efficiently personalized to align with individual user preferences for what constitutes harmful content, without compromising its core detection capabilities on non-personalized categories?*



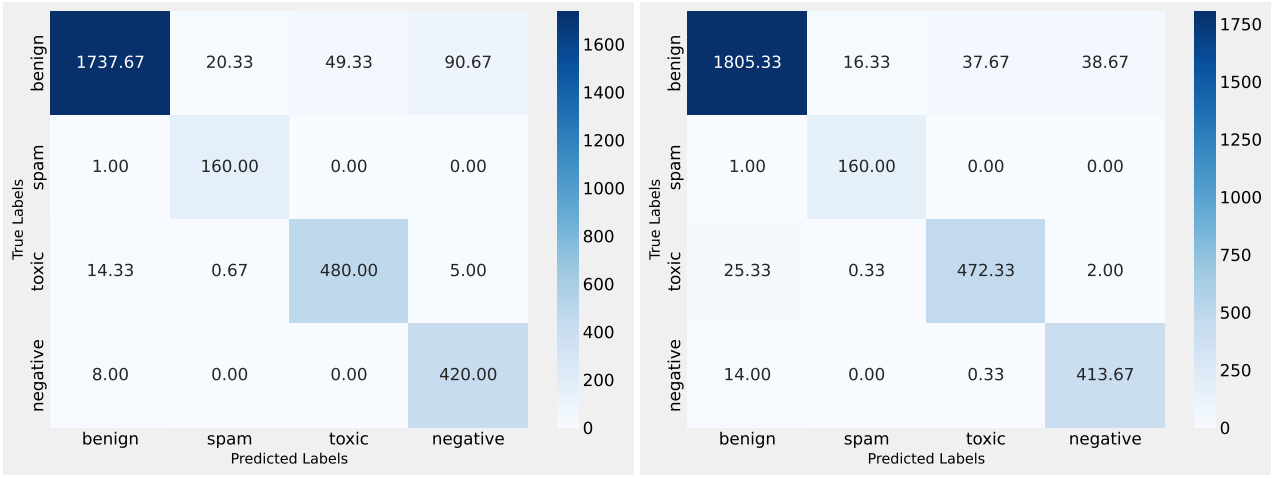
(a) Different LLMs. The retrieval strategy is set as random. (b) Level 1 and level 2 prompts with Llama and random.

Figure 10: The impact of key ICL parameters on multi-task multi-class performance.



(a) F1-score (b) Precision (c) Recall

Figure 11: The subcategory-wise performance of the best-performing ICL setup (Llama-Random) in Multi-task Multi-class ICL.



(a) Level 1 Prompt (b) Level 2 Prompt

Figure 12: The Confusion Matrix of Llama-Random (192 shots) in Multi-task Multi-class ICL

Affirmative answers would position ICL as a cornerstone for transparent, user-controlled, and on-device moderation systems. To this end, we design and evaluate three personalization scenarios that simulate real-world user interactions: blocking a new category of harmful texts (Section 7.1), unblocking a known category (Section 7.2), and blocking semantic variations of a specific harmful instance (Section 7.3).

General Evaluation Setup. In these experiments, the *Multi-Task Binary ICL* setting is adopted, which most closely reflects real deployment conditions. Based on the results in Section 5, we adopt Llama-Random as the representative configuration for all experiments. The overall setup is the same as in the multi-task binary case, with minor adjustments tailored to each personalization scenario.

7.1 Blocking a New Category of Harmful Texts

Motivation. We first consider the scenario where users encounter a new harmful category not present in the training pool. To simulate this, we remove the TextDetox dataset from the training stage and treat it as a new harmful category encountered at test time. This allows us to examine whether ICL can rapidly adapt to unseen harmful types with minimal supervision. For Level 1 prompts, we directly inject a small number of new examples ($k_2 \in \{0, 1, 2, 4, 8\}$). For Level 2 prompts, we additionally extend the task description with a definition of the new harmful type.

Experimental Setup. The experimental setup is designed to simulate a user introducing a previously unseen harmful category into an existing moderation system.

- **Datasets and Simulation:** We treat toxicity (the **TextDetox** dataset) as the *new, user-defined harmful category*. To simulate a system that has not encountered this category before, we **remove all TextDetox samples from the main demonstration pool** D_k^r. The test set $\mathcal{X}$ remains the union of all three datasets ($D_{test}^{TextDetox} \cup D_{test}^{UCI} \cup$

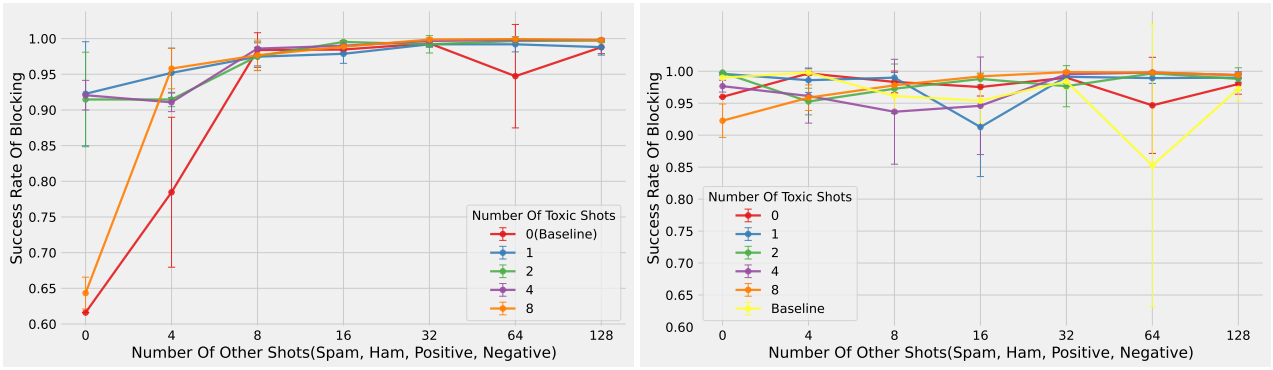
D_{test}^{SST2}). The main demonstration pool is thus $D_k^r \subseteq D_{train}^{UCI} \cup D_{train}^{SST2}$. The personalization is achieved by injecting a small number of TextDetox harmful samples into the prompt.

- **Shots:** The total number of shots is $k = k_1 + k_2$. Here, k_1 (drawn from UCI and SST2) represents the *base system knowledge* ($k_1 \in \{0, 4, 8, 16, 32, 64, 128\}$), while k_2 (drawn from TextDetox harmful samples) represents the *user-provided personalization examples* ($k_2 \in \{0, 1, 2, 4, 8\}$).

Results. As shown in Figure 13a, under Level 1 prompts, the model already achieves high blocking success (>0.98) when k_1 is sufficiently large, even without any TextDetox demonstrations. This indicates that ICL generalizes well across harmful types by exploiting shared harmfulness cues. However, success rates fluctuate more when no TextDetox shots are provided, highlighting the stabilizing role of even a single example. Adding 1–2 examples yields a sharp improvement when k_1 is small, but further increases ($k_2 = 4$ or 8) bring only marginal gains. Level 2 prompts (Figure 13b) further enhance performance when k_1 is small, since explicit definitions compensate for limited demonstrations. Once $k_1 \geq 16$, the effect of definitions diminishes, as the model already learns sufficient cross-category generalizations.

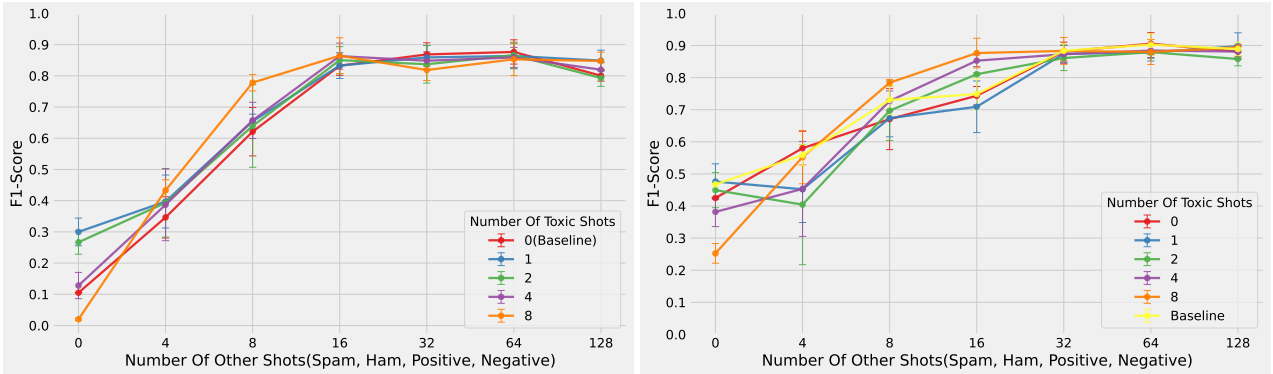
Impact on Known Categories. Figure 14 shows that performance on UCI and SST2 remains stable regardless of TextDetox additions or task description modifications, especially when $k_1 \geq 32$. This suggests that personalization to block new harmful categories does not degrade recognition of original categories.

Summary. ICL exhibits strong generalization to new harmful categories, requiring minimal supervision to achieve stable performance. Definitions in prompts are particularly useful when shot counts are small, but their impact diminishes once sufficient demonstrations are provided. Importantly, personalization does not compromise performance on existing datasets.



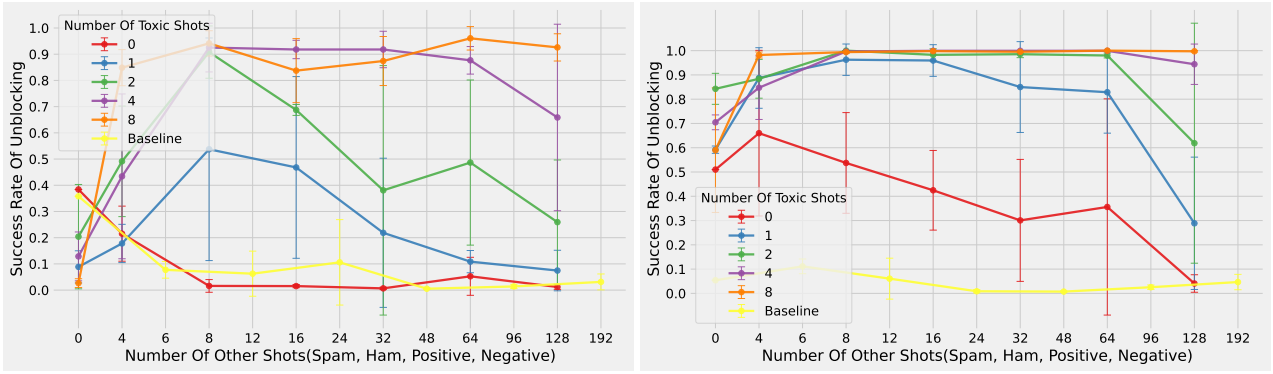
(a) Level 1 Prompt (b) Level 2 Prompt

Figure 13: **The Performance of Blocking New Harmful Texts.** Note: **Baseline** represents use original Multi-Task Binary ICL(UCI, SST2) to detect Textdetox.



(a) Level 1 Prompt (b) Level 2 Prompt

Figure 14: The performance of ICL on known categories of harmful texts, when a new category is added. Here, the known categories are set as spam and negative sentiment, while the newly added category is toxicity.



(a) Level 1 Prompt

(b) Level 2 Prompt

Figure 15: **The Performance of unblocking a known category of harmful texts.** Note: **Baseline** represents using original Multi-Task Binary ICL(Textdetox, UCI, SST2) to unblock Textdetox, wherein in-context toxic demonstrations are labeled as positive (harmful).

7.2 Unblocking a Known Category of Harmful Texts

Motivation. Different users may not fully agree on what constitutes harmful content. To simulate this, we treat TextDetox as a harmful type that certain users wish to *unblock*. For Level 1 prompts, we add a small number of TextDetox samples labeled as benign. For Level 2 prompts, we additionally redefine toxicity as benign in the task description.

Experimental Setup. Below summarizes the experimental setup:

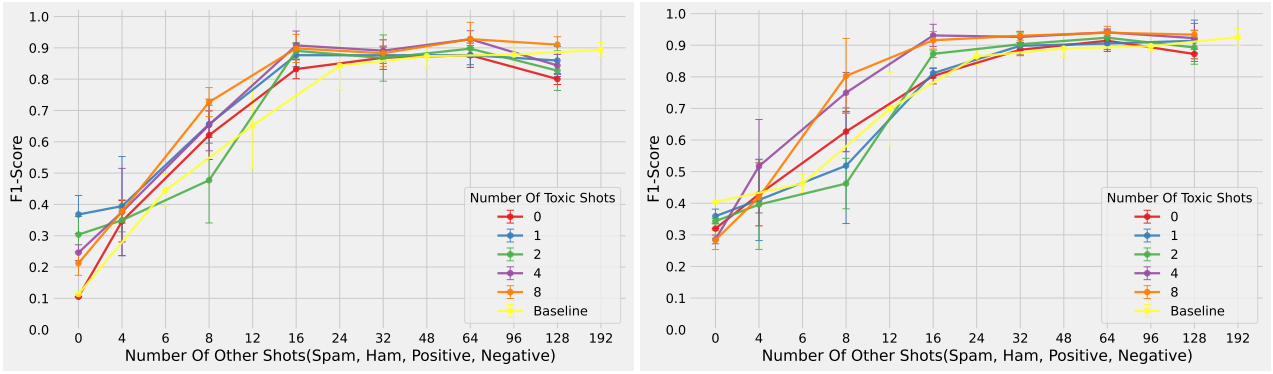
- **Datasets:** Same as before, but TextDetox is redefined as benign. Only harmful samples are used.
- **Shots:** $k = k_1 + k_2$, where $k_1 \in \{0, 4, 8, 16, 32, 64, 128\}$ (UCI, SST2) and $k_2 \in \{0, 1, 2, 4, 8\}$ (TextDetox, re-labeled as benign).

Results. As shown in Figure 15a, the model's baseline behavior (with $k_2 = 0$) is to block toxic content, correctly classifying it as harmful based on its pre-existing knowledge. Introducing a small number of re-labeled examples ($k_2 = 1$ or 2) successfully *unblocks* the category when the base knowledge k_1 is small. However, as k_1 grows, the model is exposed to more examples reinforcing the original "toxicity = harmful" association, which begins to overwhelm the sparse personalization signal, causing the unblocking success rate to drop. This indicates a **tension between general harmfulness knowledge and specific user overrides**. Crucially, with a stronger personalization signal ($k_2 \geq 4$), the model reliably unblocks the category, maintaining a high success rate (e.g., > 0.93 at $k_1 = 128$), demonstrating that persistent user input can effectively recalibrate the model's behavior.

Level 2 prompts (Figure 15b) alleviate this issue further by explicitly redefining toxicity. Specifically, Level 2 prompt performs better than level 1 prompt in the following aspects: Firstly, due to the explicit definition of toxicity and the clear indication that toxic texts should be considered benign, ICL achieves a certain level of unblocking effectiveness even without the inclusion of toxic shots—outperforming the level 1 prompt with one toxic text added. Then, when the same number of toxic texts is introduced, the success rate of unblocking is consistently higher for level 2 than for level 1. Moreover, the decline in the success rate with increasing other shots is less pronounced. Even when $k1$ reaches 128 and $k2$ = 8, the success rate reaches approximately 0.99.

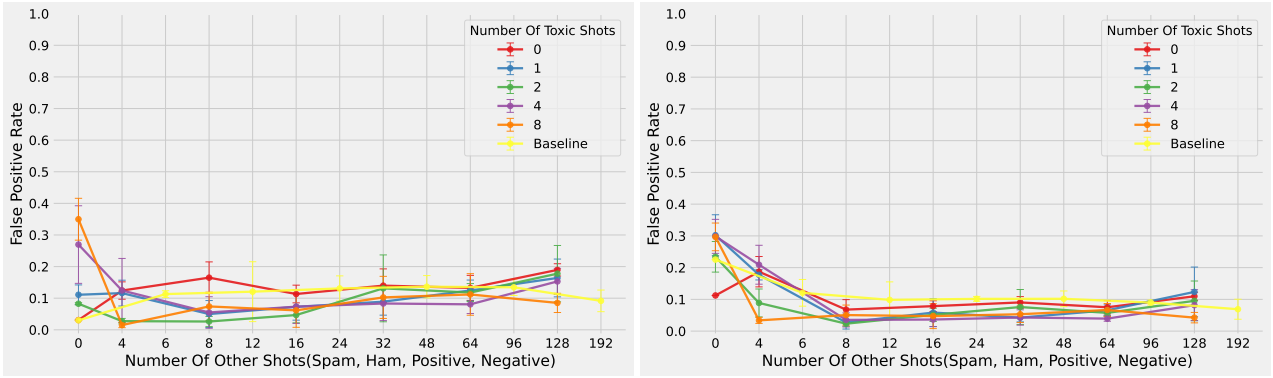
Impact on the Left Blocking Categories. As shown in Figure 16a, performance on UCI and SST2 is preserved or even slightly improved when TextDetox is re-labeled as benign. We hypothesize that this is due to the following reason: In Multi-Task Binary ICL, we previously noted that a key limitation of ICL is its high false positive rate. The addition of toxic texts labeled as benign encourages ICL to adopt a more conservative decision-making strategy for harmful classification, thereby reducing false positives. This is confirmed by Figure 17.

Summary. Without affecting the performance on the original dataset (the left blocking categories), ICL can be flexibly adapted to user preferences for unblocking harmful categories, using only a handful of re-labeled demonstrations or explicit definitions. This demonstrates a form of **contextual unlearning**, where the model suppresses a previously learned association within the specific context of a user's prompt.



(a) Level 1 Prompt (b) Level 2 Prompt

Figure 16: The F1-score Performance of ICL on the left blocking categories, when unblocking a known category of harmful texts.



(a) Level 1 Prompt (b) Level 2 Prompt

Figure 17: The false positive rate of ICL on the left blocking categories, when a known category of harmful texts is unblocked.

7.3 Blocking Variations of a Newly Annotated Harmful Text

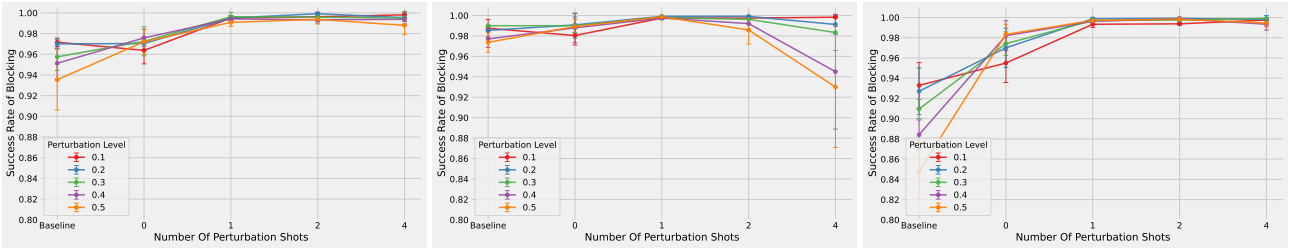
Motivation. In many cases, users cannot articulate a new harmful category but simply wish to block future texts resembling a specific harmful instance they encountered. To simulate this, we randomly select 100 harmful texts from each dataset and generate perturbed variants using EDA [59], a technique for data augmentation (via TextAttack[4]). For each original text, we generate 12 perturbed variants *per perturbation level* (0.1–0.5), resulting in a total of 60 variants across five levels. At each level, four variants are used for training (i.e., demonstration selection) and eight for testing, forming the dataset D^v. The original text itself is always included in the prompt.

Experimental Setup. Below are the experimental setup for blocking variations of a newly annotated harmful text:

- **Datasets:** $\mathcal{X} = D_{test}^{TextDetox} \cup D_{test}^{UCI} \cup D_{test}^{SST2} \cup D_{test}^{v}$, $D_k \subseteq D_{train}^{TextDetox} \cup D_{train}^{UCI} \cup D_{train}^{SST2} \cup D_{train}^{v}$.
- **Shots:** $k = k_1 + k_2 + k_3$, where $k_1 = 196$ (original datasets), $k_2 = 1$ (the original harmful text), and $k_3 \in \{0, 1, 2, 3, 4\}$ (perturbed variants).

Results. We evaluate the success rate of blocking these perturbed variants. We define two key conditions: The "**Baseline**" condition, where the prompt contains *neither* the original text nor its variants ($k_2 = 0, k_3 = 0$), tests the model's inherent ability to generalize from its base knowledge. The "**0-perturbation**" condition, where the original text is included ($k_2 = 1, k_3 = 0$), tests the utility of a single canonical example.

[4]https://textattack.readthedocs.io



(a) Toxic (b) Spam (c) Negative

Figure 18: The success rate of blocking variants of specific harmful texts. **Baseline**: The original harmful text is *not* in the prompt. **0-perturbation**: The original text is included, but no perturbed variants are.

For **toxic** texts (Figure 18a), the baseline achieves good performance. For example, when the perturbation level = 0.1, the success rate of blocking is about 0.97. However, as the perturbation level increases, the success rate begins to decrease, and when perturbation = 0.5, it drops to 0.93. But once a perturbed variant is added to the prompt, the success rate quickly rises above 0.99 across all perturbation levels, and additional variants (2–4) do not bring further gains. This indicates that toxic texts are highly amenable to instance-level personalization, requiring minimal supervision.

For **negative** texts (Figure 18c), the baseline struggles more severely with highly perturbed variants, with success rates as low as 0.85 at perturbation level 0.5. However, similar to the toxic case, including just one perturbed example dramatically improves performance across all perturbation levels, pushing success rates above 0.99. Interestingly, when perturbed examples are added, higher perturbation levels (0.4–0.5) sometimes become *easier* to block than lower ones. A possible explanation is that heavy perturbations disrupt the original semantic structure, and leaving behind distinctive lexical or surface-level cues. These cues may actually make the perturbed texts more distinguishable from benign content, thereby allowing the model to recognize and block them more reliably.

For **spam** texts (Figure 18b), the baseline performance is better than toxic and negative and achieves about a blocking success rate of 0.98. The blocking rate can be further increased when including either the original test sample or one perturbation. However, there is an abnormal phenomenon that the addition of two or more perturbed variants can actually degrade performance, especially at higher perturbation levels. For instance, with 4 perturbation shots at level 0.5, the success rate drops below 0.90. We hypothesize this is because, different perturbed spam samples often diverge significantly from each other in both semantics and lexical form, especially at higher perturbation levels, introducing inconsistent patterns that confuse the model. As a result, instead of reinforcing stable spam features, excessive perturbation examples may amplify noise and even mislead ICL.

Summary. Overall, these results demonstrate that ICL can be effectively personalized to block variations of harmful texts with minimal effort. As little as one perturbed example enables near-perfect blocking for harmful texts across all perturbation levels and more perturbed examples do not mean better results.

7.4 Takeaways for Personalization with ICL

Across the three personalization scenarios, we derive the following key insights:

- **Effortless Expansion:** ICL can seamlessly incorporate **new harmful categories** with minimal supervision—often a single example or definition suffices. This expansion occurs without catastrophic forgetting, as performance on pre-existing categories remains intact.
- **Flexible Re-calibration:** Users can **unblock or re-define** categories they find acceptable. While a tension exists between general knowledge and user override, a handful of consistent examples or a clear definition enables reliable personalization, effectively implementing **contextual unlearning**.
- **Granular Control:** Personalization operates effectively at the **instance level**. Providing a single harmful example (and optionally, its perturbed variants) allows the model to construct a robust "concept" that generalizes to future, similar instances, enabling precise, user-driven content filtering.

Taken together, these results highlight that ICL is a powerful and flexible framework for personalization. With only minimal supervision—ranging from a handful of examples to natural-language definitions—ICL can adapt to new user requirements while maintaining strong performance on established tasks. This positions ICL as a practical solution for real-world harmful content detection where personalization and customization are essential.

8 Evaluation on Wild Data

Thus far, our evaluation of ICL has focused on multi-task harmful content detection and personalization using well-established public benchmarks. While such datasets provide a controlled environment for systematic comparison, they may not fully capture the complexity and diversity of real-world scenarios.

To bridge this gap, we further examine ICL performance on *wild data*—unseen, uncurated examples collected from open platforms—which often feature noisier distributions, domain shifts, and ambiguous labeling. This section investigates the generalizability and robustness of ICL under such unconstrained, naturally occurring conditions.

We consider three task formulations: multi-task binary, multi-task multi-class, and multi-task multi-label classification. The first two mirror the settings introduced earlier, while the multi-task multi-label formulation is newly incorporated here to better reflect the compositional nature of real-world texts, where content frequently spans multiple categories (e.g., spam messages expressed in a toxic manner).

8.1 Data Source

We construct our wild dataset from Mastodon, a decentralized social media platform characterized by diverse communities and open discourse. Between December 2024 and February 2025, we collected 8,998,738 posts, of which 3,948,831 were unique English-language entries.

Since benign content dominates real-world platforms, we employ a two-stage filtering strategy to obtain a more balanced distribution. First, we randomly sample 15,000 English-language posts. Next, we apply the Llama model with 48-shot random demonstrations—identified as strong in our benchmark experiments—for a preliminary harmfulness prediction. This yields 12,616 posts predicted as benign and 2,384 as harmful. From each group, we then randomly select 1,500 posts, resulting in a balanced subset of 3,000 instances for manual annotation.

8.2 Annotation Protocol

To accommodate the multiple granularities and inherent overlaps of harmfulness in real-world data, we designed a hierarchical annotation protocol where each post received three types of labels:

1. **Binary:** A single label of either *benign* or *harmful*. This supports the evaluation of multi-task binary classification.
2. **Multi-class:** A single, mutually exclusive label from the set {*benign*, *toxic*, *spam*, *negative*}. Harmful posts are assigned one of the three harmful sub-categories. This supports multi-task multi-class classification.
3. **Multi-label:** One or more labels from the set {*benign*, *toxic*, *spam*, *negative*}. This formulation explicitly captures overlaps (e.g., *toxic* + *negative*), borderline cases (e.g., *benign* + *spam*), and composite scenarios, providing a more faithful representation of real-world content for multi-task multi-label classification.

This hierarchical scheme enables evaluation under binary, multi-class, and multi-label settings, providing a rich testbed for analyzing ICL robustness in complex, real-world contexts.

8.3 Label Distribution

Table 4 summarizes the label composition. Under the multi-class setting, 1,202 of the 3,000 samples are harmful: 755 `negative` (62.8%), 259 `toxic` (21.6%), and 188 `spam` (15.6%). This distribution aligns with expectations: Mastodon users frequently express personal emotions or frustrations, making `negative` content dominant among harmful posts.

In the multi-label setting, 980 texts (32.7%) carry two or more labels. Common overlaps include `negative + toxic` (339 texts), `benign + negative` (249 texts), and `benign + spam` (184 texts). We also observe 32 triple-label cases (e.g., `benign + negative + toxic`). Notably, some posts simultaneously bear `benign` and `toxic`/`spam` labels. These overlaps may seem counterintuitive at first glance, but closer inspection reveals their plausibility. Below we provide illustrative examples:

- **Benign + Spam.** These texts lack overtly malicious intent, but their link-heavy structure or excessive use of hashtags makes them spam-like in appearance.

```
case1: obligatory provided link monstrosity https://www.theguardian.com/[...]
case2: Depression. Written by Satan & Lee Wilson. #bookblog #readallthebooks #MSWL
#WhatToRead #Writing #LeeWilson #BookMarketing #WritingCommunity #amquerying #Reading
Block Filter - #UGhhYmFyZw https://satan-speaks.com/[...]
```

Table 4: Label distribution of Mastodon Wild Data.

(a) Multi-class Label

Label	Count
Benign	1798
Negative	755
Toxic	259
Spam	188

(b) Multi-label Label

Labels	Count	Labels	Count	Labels	Count
Benign	1437	Benign, Negative	249	Benign, Negative, Spam	11
Negative	517	Benign, Spam	184	Benign, Negative, Toxic	13
Spam	60	Benign, Toxic	8	Benign, Spam, Toxic	5
Toxic	6	Negative, Spam	10	Negative, Spam, Toxic	3
-	-	Negative, Toxic	339	-	-
-	-	Spam, Toxic	98	-	-
Sum	2020	-	948	-	32

- **Benign + Toxic.** These posts are primarily expressive or informative, yet contain profanity, offensive language, or sarcastic tone that warrant a toxic label.

```
case1:  FUCK this show is so good
case2:  LIVEBLOG: Israeli Army Kills Scores in Gaza, Assassinates Police Chief [...]
\At least 11 displaced Palestinians were killed, and others injured, after Israeli
occupation forces targeted tents in the Al-Mawasi area, west of Khan Yunis [...]"
#Press #Gaza #Palestine #Resistance #Israel #Genocide #Terrorism #WarCrimes #Hamas
#Zionism #Barbarism #BloodLust
case3:  Nothing but clap for MY jeb!sident
```

These examples demonstrate that what might initially appear contradictory under rigid single-label schemes often reflects the fuzzy and compositional nature of real-world discourse. Consequently, multi-label formulations are more faithful in capturing the nuanced semantics of wild data.

8.4 Multi-Task Binary on Wild Data

Experimental Setup Most of the experimental settings remain the same as those previously described, with only minor modifications. In this section, we detail the parts that differ.

- **Models, Selection Strategies, and Number of Shots** We adopt the best-performing model-strategy combination identified on the public benchmarks: `Llama-Random`. Regarding the number of shots, although using 192 shots yields the highest performance (F1-score: 0.918), the 48-shot setting already achieves competitive results (F1-score: 0.887). To balance effectiveness and efficiency, we fix the number of shots at 48.
- **Datasets** We use the 3,000 annotated examples introduced earlier as the test set D_{test}. To evaluate the generalization capability of ICL, we first construct demonstrations (D_k) from the training sets of public datasets, i.e., $D_k^r \subseteq D_{train}^{\text{TextDetox}} \cup D_{train}^{\text{UCI}} \cup D_{train}^{\text{SST2}}$. However, since these public datasets differ from our wild data in both source and style, we also explore a more realistic scenario: we annotate an additional 1,000 wild data samples following the same annotation protocol, and use them as D_{train} for constructing in-domain demonstrations. This allows us to assess the performance of ICL when a certain amount of real-world annotated data is available.
- **Prompt Template** Initially, we used the same prompt template as in the previous experiments. However, inspired by later empirical findings (to be discussed in detail later), we experimented with a modified prompt format: when presenting demonstration examples in the prompt, we not only include the corresponding labels but also provide a reason for why the text belongs to that label. The updated prompt template is illustrated in Figure 19. The *reason* for a demonstration is generated by using DeepSeek-V3 under the assumption that the label is known. Let $\mathcal{R}$ denote the reason space. Under this setting, $D_k^r = \{(x_1, r_1, y_1), (x_2, r_2, y_2), \ldots, (x_k, r_k, y_k)\}$.

Results As shown in Table 5, when using demonstrations sampled from public datasets, ICL achieves an accuracy of 0.829 and an F1-score of 0.795 on the wild test set. Although the recall is relatively high at 0.883, the false positive rate (FPR) is also high (0.203), resulting in a relatively low precision of 0.724. Compared with the performance on public test sets (F1-score = 0.887), there is a clear performance drop. We hypothesize that this is due to the domain gap between the public datasets and the wild data—differences in language style, expression patterns, and content distribution likely cause a mismatch, making it difficult for the model to generalize when demonstrations come from a different source.

To test this hypothesis, we replaced all demos with samples drawn from the wild dataset and repeated the experiment. This setup leverages one of the strengths of ICL: the ability to rapidly adapt to new domains given only a few labeled

```
<Description of the specific task>
==
Query:<sentence 1>
Answer:{"reason": <reason>, "label": <label>}
==
Query:<sentence 2>
Answer:{"reason": <reason>, "label": <label>}
==
...
Query:<predicted text>
Answer:
```

Figure 19: Prompt Template with Reason for ICL.

Table 5: Performance of Multi-Task Binary ICL on Wild Data with Different Demonstration Settings

Demos Source (Prompt Template)	Precision	Recall	FPR	F1-Score	Accuracy
Public Datasets (No Reason)	0.724	0.883	0.203	0.795	0.829
Wild Data (No Reason)	0.714	0.836	0.224	0.770	0.800
Wild Data (With Reason)	0.966	0.800	0.019	0.875	0.909

examples, without further training. Surprisingly, however, the performance did not improve—in fact, it deteriorated. The FPR increased from 0.203 to 0.224, and the recall dropped from 0.883 to 0.836. This counterintuitive result suggests that although the demos now match the domain of the test set, their effectiveness is compromised. A possible explanation is that the wild data is inherently noisy and exhibits uneven quality, making it difficult for the model to extract reliable decision patterns from the raw labels alone.

The failure of in-domain wild demos suggests that raw labels alone are insufficient for the model to disambiguate noisy, real-world examples. We hypothesize that providing *interpretive signals* could guide the model towards more robust decision patterns. To test this, we augment the demonstration format to include a natural-language rationale, transforming each demo from (x_i, y_i) to (x_i, r_i, y_i), where r_i is a reason explaining why x_i belongs to class y_i. These rationales are generated automatically by DeepSeek-V3, conditioned on the ground-truth label.

Experimental results confirm this intuition. After adding the rationale, although recall dropped slightly to 0.800, the FPR significantly decreased to 0.019. As a result, the precision rose dramatically to 0.966, and accuracy increased to 0.909. These results indicate that providing interpretive signals during in-context learning can greatly reduce overprediction and improve overall robustness, especially in noisy, real-world scenarios.

8.5 Multi-Task Multi-Class

After evaluating ICL under the multi-task binary setting, we now shift our focus to the more fine-grained *multi-task multi-class* setting.

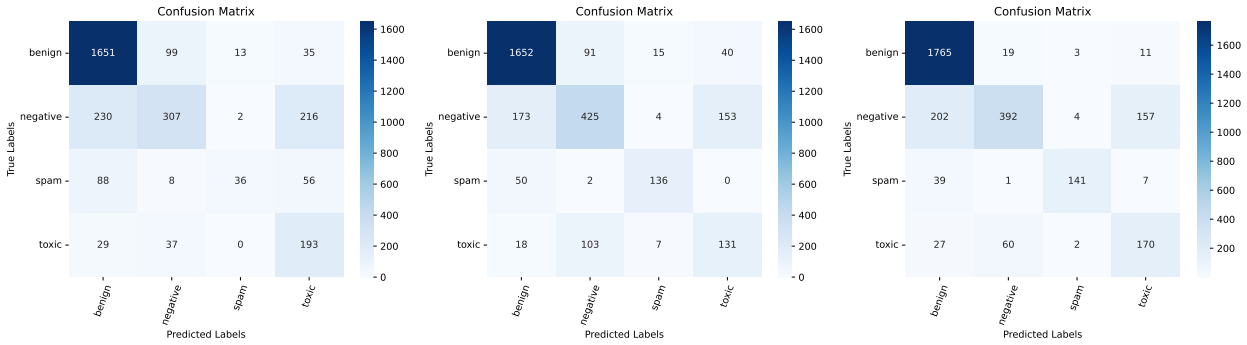
Experimental Setup We adopt the same three configurations as in the previous section to investigate the impact of demo source and prompt format: (1) using public datasets as demos without reasoning, (2) using wild data as demos without reasoning, and (3) using wild data with both labels and corresponding reasoning. Apart from the expanded label space, the experimental setup remains largely consistent with the multi-task binary setting.

Results Performance improves consistently across configurations (Table 6). Weighted F1 increases from 0.712 (public demos) to 0.814 (wild demos + reason). The largest gain occurs for `spam` (F1: 0.301 $\rightarrow$ 0.834). Despite these overall improvements, the performance on the *negative* and *toxic* categories remains suboptimal. As visualized in the confusion matrix (Figure 20), a considerable number of *negative* samples are misclassified as *benign* or *toxic*, while many *toxic* samples are predicted as *negative*. Additionally, a significant portion of *spam* instances are incorrectly labeled as *benign*.

We hypothesize that these confusions are partly due to the multi-label nature of some examples in our dataset. Based on our annotation experience, certain texts may simultaneously exhibit characteristics of multiple categories (e.g., a toxic comment that also expresses negative emotion), which makes the single-label classification setting inherently challenging for these classes.

Table 6: F1-Score of Multi-Task Multi-Class ICL on Wild Data with Different Demonstration Settings

Demos Source (Prompt Template)	Benign	Negative	Spam	Toxic	Weighted Avg
Public Datasets (No Reason)	0.870	0.509	0.301	0.509	0.712
Wild Data (No Reason)	0.895	0.618	0.777	0.449	0.779
Wild Data (With Reason)	0.921	0.639	0.834	0.563	0.814



(a) Public Datasets(No Reason) (b) Wild Data(No Reason) (c) Wild Data(Reason)

Figure 20: Confusion Matrix of Multi-Task Multi-Class ICL on Wild Data with Different Demonstration Settings.

8.6 Multi-Task Multi-Label

While the previous section focuses on multi-task multi-class classification, many real-world harmful content detection scenarios involve overlapping or co-occurring categories—that is, a single piece of content may simultaneously exhibit characteristics of multiple classes, such as being both *toxic* and *negative*. To better accommodate such complexity, we now consider the more realistic and challenging **multi-task multi-label** setting, where the model must predict one or more labels per input instance.

This formulation is particularly relevant for open-domain harmful content detection, where rigidly assigning a single label may overlook important semantic nuances. For example, a message attacking someone in a hostile and angry tone may reasonably be classified as both *negative* and *toxic*. As a result, this setting imposes higher demand on the model's ability to capture multi-faceted cues within each input and distinguish subtle overlaps between label semantics.

In this section, we evaluate how in-context learning (ICL) performs under this multi-label setup, and analyze the ICL's strengths and limitations when handling overlapping categories.

Experimental Setup Since previous experiments have demonstrated that using demonstrations from wild data along with explicit reasoning consistently yields better performance in both multi-task binary and multi-task multi-class settings, we adopt this configuration exclusively in our evaluation of the multi-task multi-label setup.

Given the multi-label nature of this task, the prompt template is slightly modified to instruct the model to output multiple labels per instance. An illustration of the updated prompt format is shown in Figure 21. All other experimental settings remain consistent with multi-task binary apart from the expanded label space.

Results The Multi-Task Multi-Label ICL model achieves promising results on wild data, with a **Subset Accuracy** of 0.707, indicating that 70.7% of samples have exactly matched label sets between prediction and ground truth. This is a relatively high score in the multi-label setting, where strict exact-match criteria are often hard to satisfy. To evaluate per-label prediction quality, we report two common metrics: **Hamming Loss**[5] and **Jaccard Score**[6]. Hamming Loss is defined as the fraction of incorrectly predicted labels over the total number of labels, and the lower the better. Our model achieves a Hamming Loss of 0.090, which means only 9.0% of all possible label predictions are incorrect, demonstrating low per-label error. Jaccard Score, also known as Intersection over Union (IoU), is calculated as the size of the intersection divided by the size of the union between predicted and true label sets, and reflects the degree of partial correctness. Our model achieves a Jaccard Score of 0.831, suggesting strong alignment between predicted and actual label combinations.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html
[6]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.jaccard_score.html

```
<Description of the specific task>
==
Query:<sentence 1>
Answer:{"reason": <reason>, "is_benign": True/False, "is_negative": True/False,
"is_toxic": True/False, "is_spam": True/False,}
==
Query:<sentence 2>
Answer:{"reason": <reason>, "is_benign": True/False, "is_negative": True/False,
"is_toxic": True/False, "is_spam": True/False,}
==
...
Query:<predicted text>
Answer:
```

Figure 21: ICL Prompt Template for Multi-Task Multi-Label Classification with Reason.

Table 7: Performance of Multi-Task Multi-Label ICL on Wild Data

Label	Count	Precision	Recall	F1-Score
Benign	1907	0.917	0.874	0.895
Negative	1202	0.810	0.916	0.859
Spam	371	0.758	0.836	0.795
Toxic	532	0.845	0.829	0.837
Weighted Avg	4012	0.861	0.877	0.867

Detailed per-class results are summarized in Table 7. The model performs particularly well on *benign* and *negative* categories. *Benign* has very high precision (0.917), while *negative* achieves an impressive recall of 0.916, showing that most truly negative samples are successfully detected. Performance on *spam* is slightly weaker, with a recall of 0.836 and F1-score of 0.795; we hypothesize this may be due to the presence of link-heavy or hashtag-laden content on platforms like Mastodon, which could inflate false positives. The *toxic* class shows a good balance between precision (0.845) and recall (0.829), leading to an F1-score of 0.837. Overall, the model attains a **weighted average F1-score** of 0.867, indicating reliable and balanced performance across categories without major bias. These results demonstrate that ICL is effective in the multi-task multi-label setting, performing well both at the instance level and the label level.

8.7 Summary

Our evaluation on Mastodon wild data highlights three key insights. First, models that achieve strong performance on clean public benchmarks exhibit clear degradation when directly applied to wild data, underscoring the challenge of domain shifts and noisy distributions. This confirms that results obtained solely on curated datasets may overestimate real-world robustness, and highlights the necessity of complementary wild data evaluation. Second, augmenting demonstrations with explicit rationales proves highly effective: this design substantially reduces false positives and improves robustness, showing that interpretive signals can compensate for label noise in real-world contexts. Third, both annotation observations and experimental results confirm that single-label formulations are often insufficient in open-domain harmful content detection. Many posts simultaneously display characteristics of multiple categories, and the multi-label setting provides a more faithful and informative modeling paradigm.

Taken together, these findings establish wild data evaluation as an essential step for validating the practical utility of ICL. Compared with public datasets, wild data not only reveals previously hidden weaknesses (e.g., domain sensitivity, high FPR), but also motivates new design choices (reason-enriched prompts, multi-label-aware formulations) that significantly improve performance. These results demonstrate that robust deployment of ICL for harmful content detection requires going beyond benchmarks to embrace real-world complexity.

9 Discussion

Cross-Task Generalization of ICL. Our results show that in-context learning (ICL) exhibits strong cross-task generalization for harmful content detection. Across binary, multi-class, and multi-label formulations, ICL consistently achieves

competitive or superior performance compared to supervised baselines, despite requiring only a small number of demonstrations. This highlights ICL's ability to flexibly adapt to diverse task formulations without retraining, a critical advantage in real-world moderation where task boundaries often shift dynamically. Importantly, the robustness of ICL across these heterogeneous setups distinguishes our work from prior studies that evaluate harmful content detection only in single-task settings.

Personalization. Beyond cross-task robustness, our study highlights the promise of ICL for personalization. By designing three personalized scenarios, we show that models can rapidly adapt to blocking or unblocking specific categories of harmful behaviors with minimal supervision. Moreover, ICL also allows for targeted defenses against adversarial modifications, ensuring that even highly altered harmful content is effectively captured. Together, these findings demonstrate that personalization in ICL is not limited to dataset-level adjustments but can extend to user- or community-specific harmfulness definitions, offering a lightweight alternative to costly retraining pipelines.

Nuances of Wild Data Evaluation. The wild data experiments underscore the importance of moving beyond curated benchmarks. Mastodon posts often display overlapping or borderline labels (e.g., `benign+spam`, `negative+toxic`), which are poorly represented in traditional datasets. Our multi-label evaluation demonstrates that ICL can capture these compositional cases, but also exposes limitations in single-label formulations that mask the inherent fuzziness of real-world content. These findings reinforce the necessity of adopting more nuanced evaluation protocols and richer datasets that account for overlap, ambiguity.

Limitations and Future Directions. Despite these contributions, several limitations remain. First, our study currently covers only three representative harmful categories (toxicity, spam, negative sentiment). Future research should incorporate a broader taxonomy, including misinformation, violent content, etc., to achieve more comprehensive coverage. Second, while our multi-class and multi-label settings already capture a range of realistic scenarios, the taxonomy itself remains relatively coarse. Future work could explore finer-grained subdivisions within each category. For example, *toxicity* can be broken down into subtypes such as threats, insults, or personal attacks; *spam* can be subdivided into promotional spam, phishing attempts, or link farming; and *negative sentiment* can be differentiated into emotions such as sadness, anger, or frustration. Such granularity would allow for more precise modeling of harmful behaviors as they manifest in diverse forms. Third, our evaluation of reason-augmented prompts focuses solely on classification outcomes without systematically assessing the quality, faithfulness, or potential biases of the generated explanations. Fourth, although Mastodon provides a valuable wild testbed, it cannot capture the full diversity of harmful content across platforms, domains, and user demographics. Beyond text, multimodal harmful content (e.g., hateful memes, doctored videos, multimodal spam) poses additional challenges. Finally, our experiments are restricted to English, leaving open questions about the effectiveness of ICL in multilingual or code-switching scenarios.

Code and Data Availability. To foster reproducibility and future research, we publicly release our codebase and the annotated Mastodon dataset. The codebase includes implementations for all ICL configurations, retrieval strategies, and evaluation scripts. The dataset comprises the 3,000 annotated posts with binary, multi-class, and multi-label labels, providing a valuable benchmark for robust harmful content detection. Access links are available in the Abstract for reference.

10 Conclusion

We have presented a comprehensive study of in-context learning for harmful content detection, systematically evaluating its capabilities across cross-task generalization, user personalization, and robustness in the wild. Our findings demonstrate that ICL, when configured with appropriate models, retrieval strategies, and prompt designs, achieves strong performance competitive with fine-tuned models across binary, multi-class, and multi-label tasks, all without task-specific training.

Crucially, we showed that ICL is not only powerful but also highly adaptable. It enables **lightweight personalization**, allowing users to block new categories, unblock existing ones, and defend against adversarial variations with minimal examples. Furthermore, our evaluation on a new Mastodon wild dataset revealed the limitations of benchmark-only evaluation and identified **rationale-augmented prompts** as a key technique for restoring robustness against noisy, real-world data and mitigating false positives.

In summary, this work moves **beyond one-size-fits-all** moderation by establishing ICL as a practical, privacy-preserving, and user-centric pathway for content safety. By bridging the gap between controlled benchmarks and the complexities of the wild, we hope to inspire future work towards more robust, explainable, and adaptable online safety systems.

References

[1] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli, "A deep learning method for automatic sms spam classification: Performance of learning algorithms on indigenous dataset," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 17, p. e6989, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6989

[2] A. Aghajanyan, A. Gupta, A. Shrivastava, X. Chen, L. Zettlemoyer, and S. Gupta, "Muppet: Massive multi-task representations with pre-finetuning," *CoRR*, vol. abs/2101.11038, 2021. [Online]. Available: https://arxiv.org/abs/2101.11038

[3] M. AI, "Llama-3.1-8b-instruct," https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024, accessed: 2025-03-25.

[4] T. Almeida and J. Hidalgo, "SMS Spam Collection," UCI Machine Learning Repository, 2011, DOI: https://doi.org/10.24432/C5CC84.

[5] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering*, ser. DocEng '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 259–262. [Online]. Available: https://doi.org/10.1145/2034691.2034742

[6] H. Almuhimedi, S. Wilson, B. Liu, N. Sadeh, and A. Acquisti, "Tweets are forever: a large-scale quantitative analysis of deleted tweets," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 897–908. [Online]. Available: https://doi.org/10.1145/2441776.2441878

[7] M. M. Amin, R. Mao, E. Cambria, and B. W. Schuller, "A wide evaluation of chatgpt on affective computing tasks," *IEEE Trans. Affect. Comput.*, vol. 15, no. 4, p. 2204–2212, Oct. 2024. [Online]. Available: https://doi.org/10.1109/TAFFC.2024.3419593

[8] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, "Sms spam message detection using term frequency-inverse document frequency and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 509–515, 2019, the Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919318617

[9] A. Arango, J. Pérez, and B. Poblete, "Hate speech detection is not as easy as you may think: A closer look at model validation (extended version)," *Information Systems*, vol. 105, p. 101584, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437920300715

[10] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 759–760. [Online]. Available: https://doi.org/10.1145/3041021.3054223

[11] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu, "Qwen technical report," 2023. [Online]. Available: https://arxiv.org/abs/2309.16609

[12] A. Bratko, B. Filipič, G. V. Cormack, T. R. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," *J. Mach. Learn. Res.*, vol. 7, p. 2673–2698, Dec. 2006.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[14] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton *et al.*, "Palm: Scaling language modeling with pathways," 2022. [Online]. Available: https://arxiv.org/abs/2204.02311

[15] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus *et al.*, "Scaling instruction-finetuned language models," *J. Mach. Learn. Res.*, vol. 25, no. 1, Jan. 2024.

[16] D. K. Citron, *Hate Crimes in Cyberspace*. Harvard University Press, 2014. [Online]. Available: http://www.jstor.org/stable/j.ctt7zsws7

[17] cjadams, J. Sorensen, J. Elliott, L. Dixon, M. McDonald, nithum, and W. Cukierski, "Toxic comment classification challenge," https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge, 2017, kaggle.

[18] D. Dale, A. Voronov, D. Dementieva, V. Logacheva, O. Kozlova, N. Semenov, and A. Panchenko, "Text detoxification using large pre-trained neural models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7979–7996. [Online]. Available: https://aclanthology.org/2021.emnlp-main.629/

[19] D. Dementieva, D. Moskovskiy, N. Babakov, A. A. Ayele, N. Rizwan, F. Schneider, X. Wang, S. M. Yimam, D. Ustalov, E. Stakovskii, A. Smirnova, A. Elnagar, A. Mukherjee, and A. Panchenko, "Overview of the multilingual text detoxification task at pan 2024," in *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, G. Faggioli, N. Ferro, P. Galuščáková, and A. G. S. de Herrera, Eds. CEUR-WS.org, 2024.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, p. 82–89, Apr. 2013. [Online]. Available: https://doi.org/10.1145/2436256.2436274

[22] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surv.*, vol. 51, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3232676

[23] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, ser. SETN '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3200947.3208069

[24] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing bert against traditional machine learning text classification," *arXiv preprint arXiv:2005.13012*, 2020.

[25] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 8342–8360. [Online]. Available: https://aclanthology.org/2020.acl-main.740/

[26] L. Hanu and Unitary team, "Detoxify," Github. https://github.com/unitaryai/detoxify, 2020.

[27] X. He, S. Zannettou, Y. Shen, and Y. Zhang, " You Only Prompt Once: On the Capabilities of Prompt Learning on Large Language Models to Tackle Toxic Content ," in *2024 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2024, pp. 770–787. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00061

[28] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[29] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, "SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 2177–2190. [Online]. Available: https://aclanthology.org/2020.acl-main.197/

[30] Jigsaw and Google, "Toxic comment classification challenge," https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge, 2018, accessed: 2025-09-14.

[31] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," 2023. [Online]. Available: https://arxiv.org/abs/2205.11916

[32] M. Koroteev, "Bert: a review of applications in natural language processing and understanding," *arXiv preprint arXiv:2103.11943*, 2021.

[33] D. Kumar, P. G. Kelley, S. Consolvo, J. Mason, E. Bursztein, Z. Durumeric, K. Thomas, and M. Bailey, "Designing toxic content classification for a diversity of perspectives," in *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, Aug. 2021, pp. 299–318. [Online]. Available: https://www.usenix.org/conference/soups2021/presentation/kumar

[34] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[35] M. Labonne and S. Moran, "Spam-t5: Benchmarking large language models for few-shot email spam detection," 2023. [Online]. Available: https://arxiv.org/abs/2304.01238

[36] A. Lampinen, I. Dasgupta, S. Chan, K. Mathewson, M. Tessler, A. Creswell, J. McClelland, J. Wang, and F. Hill, "Can language models learn from explanations in context?" in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 537–563. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.38/

[37] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[38] Y. Li, R. Zhang, W. Rong, and X. Mi, "Spamdam: Towards privacy-preserving and adversary-resistant sms spam detection," 2024. [Online]. Available: https://arxiv.org/abs/2404.09481

[39] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, E. Agirre, M. Apidianaki, and I. Vulić, Eds. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. [Online]. Available: https://aclanthology.org/2022.deelio-1.10/

[40] X. Liu, H. Lu, and A. Nayak, "A spam transformer model for sms spam detection," *IEEE Access*, vol. 9, pp. 80 253–80 263, 2021.

[41] M. Luo, X. Xu, Z. Dai, P. Pasupat, M. Kazemi, C. Baral, V. Imbrasaite, and V. Y. Zhao, "Dr.icl: Demonstration-retrieved in-context learning," 2023. [Online]. Available: https://arxiv.org/abs/2305.14128

[42] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447914000550

[43] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11 048–11 064. [Online]. Available: https://aclanthology.org/2022.emnlp-main.759/

[44] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Natural instructions: Benchmarking generalization to new tasks from natural language instructions," *CoRR*, vol. abs/2104.08773, 2021. [Online]. Available: https://arxiv.org/abs/2104.08773

[45] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using bert," 2019. [Online]. Available: https://arxiv.org/abs/1910.03474

[46] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[47] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *CoRR*, vol. abs/1908.10084, 2019. [Online]. Available: http://arxiv.org/abs/1908.10084

[48] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, p. 333–389, Apr. 2009. [Online]. Available: https://doi.org/10.1561/1500000019

[49] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter sms spam," *Future Generation Computer Systems*, vol. 102, pp. 524–533, 2020.

[50] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2655–2671. [Online]. Available: https://aclanthology.org/2022.naacl-main.191/

[51] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, L.-W. Ku and C.-T. Li,

Eds. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10. [Online]. Available: https://aclanthology.org/W17-1101/

[52] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[53] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1–9. [Online]. Available: https://doi.org/10.1145/1920261.1920263

[54] M. A. Team, "Mistral-7b-instruct-v0.3," https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3, 2024, accessed: 2025-03-25.

[55] Q. Team, "Qwen2-7b-instruct," https://huggingface.co/Qwen/Qwen2-7B-Instruct, 2024, accessed: 2025-03-25.

[56] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[57] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022, survey Certification. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD

[58] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *CoRR*, vol. abs/2201.11903, 2022. [Online]. Available: https://arxiv.org/abs/2201.11903

[59] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," 2019. [Online]. Available: https://arxiv.org/abs/1901.11196

[60] J. Ye, Z. Wu, J. Feng, T. Yu, and L. Kong, "Compositional exemplars for in-context learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[61] J. Zhang, Q. Wu, Y. Xu, C. Cao, Z. Du, and K. Psounis, "Efficient toxic content detection by bootstrapping and distilling large language models," ser. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. [Online]. Available: https://doi.org/10.1609/aaai.v38i19.30178

[62] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 745–760. [Online]. Available: https://doi.org/10.1007/978-3-319-93417-4_48

Appendix A Additional Details on Model Selection

As discussed in Section 3.3, we adopt three instruction-tuned foundation models available by July 2024: Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen2-7B-Instruct. Here we provide additional details on subsequent releases between July 2024 and July 2025, and explain why we don't update models.

Table 8 summarizes the newer models. For the Llama family, multiple versions were released, including lightweight models (1B/3B), large multimodal models (11B/90B), a single 70B text-only model, and the Llama 4 MoE series (109B/400B). These models are either too small to serve as strong baselines or too large to evaluate efficiently in our ICL setup, and thus fall outside our experimental scope.

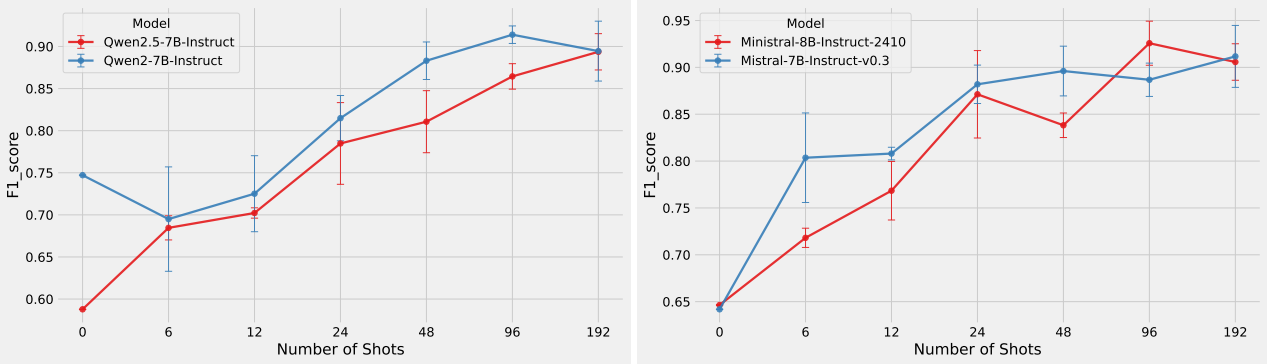
For Qwen, two new generations have been released. Qwen2.5 provides a range of dense models (0.5B–72B) and task-specific variants (Coder and Math). Qwen3 introduces both MoE and dense models, spanning from 0.6B to 235B. Among them, only Qwen2.5-7B-Instruct is comparable in scale and task orientation to our baseline. However, as shown in Figure 22a, Qwen2.5-7B-Instruct performs similarly to, or in some cases worse than, Qwen2-7B-Instruct under our evaluation.

For Mistral, many new models have appeared. The suitable variant is Ministral-8B-Instruct-2410. Yet, Figure 22b demonstrates that this model does not yield clear improvements over Mistral-7B-Instruct-v0.3 in the multi-task binary setting.

In summary, although a wide range of newer models have been released between July 2024 and July 2025, they are either mismatched in scale, specialized for unrelated domains (e.g., multimodal, programming, math), or do not provide measurable advantages over our adopted baselines. We therefore retain Llama-3.1-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7B-Instruct-v0.3 as the representative models throughout our study.

Family	Previously Used Model	New Releases (2024.7–2025.7)
Llama	Llama-3.1-8B-Instruct	Llama 3.2 (1B/3B, 11B/90B multimodal); Llama 3.3 (70B); Llama 4 (109B/400B MoE)
Qwen	Qwen2-7B-Instruct	Qwen2.5 (0.5B–72B, incl. 7B-Instruct); Qwen2.5-Coder/Math; Qwen3 (MoE + dense, up to 235B)
Mistral	Mistral-7B-Instruct-v0.3	Ministral-8B-Instruct-2410 and other variants

Table 8: Previously adopted models vs. newer releases (July 2024–July 2025).



(a) Qwen (b) Mistral

Figure 22: Performance comparison between adopted baselines and newly released models on the multi-task binary task (same configurations).

Appendix B Task Descriptions

This section provides the task descriptions used across different experimental settings.

Appendix B.1 Single-Task

The task descriptions for single-task experiments are presented below(Figure 23). All three datasets share the same task description format, differing only in their label definitions. Each description specifies the task objective and label space, ensuring consistency across toxicity detection, sentiment classification, and spam detection.

Appendix B.2 Multi-Task Binary

For the multi-task binary setting, we design two levels of task descriptions. Level 1 (Figure 24a) provides a concise overview of the task objective and label space, while Level 2 (Figure 24b) further elaborates on the meaning and criteria of each label to enhance task clarity and consistency.

Appendix B.3 Multi-Task Multi-Class

Similar to the binary setting, we also define two levels of task descriptions for the multi-task multi-class scenario (Figure 25). The two levels follow the same hierarchical design, offering both concise and detailed task formulations.

Appendix B.4 Wild Data

In the wild data experiments, each demonstration in the prompt additionally includes a *reason* explaining why the text corresponds to a given label. Accordingly, the task description is modified to incorporate this reasoning component (Figure 26). Moreover, since we conduct multi-label classification in this setting, the corresponding task description for the multi-label setup is also provided (Figure 27).

‘

```
#### Positioning
- Assistant Name: Spam Detector
- Primary Task: Classify whether a given text is spam or ham.

#### Capabilities
- Message Analysis: Accurately analyze the content and characteristics of text.
- Spam Classification: Determine whether a text is 'spam' or 'ham'.

#### Knowledge Base
- Message Categories:
  - spam
  - ham

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('spam' or 'ham'), without any additional explanation.
```

(a) Spam

```
#### Positioning
- Assistant Name: Toxic text Detector
- Primary Task: Classify whether a given text is toxic or benign.

#### Capabilities
- Message Analysis: Accurately analyze the content and characteristics of text.
- Spam Classification: Determine whether a text is 'toxic' or 'benign'.

#### Knowledge Base
- Message Categories:
  - toxic
  - benign

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('toxic' or 'benign'), without any additional
    explanation.
```

(b) Toxicity

```
#### Positioning
- Assistant Name: Negative text Detector
- Primary Task: Classify whether a given text is negative or positive.

#### Capabilities
- Message Analysis: Accurately analyze the content and characteristics of text.
- Spam Classification: Determine whether a text is 'negative' or 'positive'.

#### Knowledge Base
- Message Categories:
  - negative
  - positive

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('negative' or 'positive'), without any additional
    explanation.
```

(c) Sentiment

Figure 23: The task descriptions of single-task binary ICL.

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: Classify whether a given text is harmful or benign.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Harmfulness Classification: Determine whether a text is 'harmful' or 'benign'.

#### Knowledge Base
- Message Categories:
 - harmful
 - benign

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('harmful' or 'benign'), without any additional
    explanation.
```

(a) Level1

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: Classify whether a given text is harmful or benign.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Harmfulness Classification: Determine whether a text is 'harmful' or 'benign'.

#### Knowledge Base
- Message Categories and Definitions:
 - harmful: Harmful text refers to content that may cause discomfort, harm, or negative effects on
     others. It includes:
  - spam: Unsolicited, often irrelevant messages
  - toxic: Hostile, offensive language
  - negative emotion text: Messages expressing sadness, anger, or frustration that may affect others
      negatively
 - benign: Benign text does not contain harmful content. It includes:
  - ham: Non-spam, relevant messages
  - non-toxic text: Neutral or friendly language
  - positive emotion text: Messages expressing happiness, gratitude, or optimism

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('harmful' or 'benign'), without any additional
    explanation.
```

(b) Level2

Figure 24: The task descriptions of multi-task binary ICL.

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: Classify whether a given text is benign or a specific type of harmful content.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Harmfulness Classification: Determine whether a text is 'benign' or falls under a harmful
    subcategory such as 'spam', 'toxic', or 'negative'.

#### Knowledge Base
- Message Categories:
 - harmful:
  - spam
  - toxic
  - negative
 - benign

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('benign', 'spam', 'toxic', or 'negative'), without any
     additional explanation.
```

(a) Level1

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: Classify whether a given text is benign or a specific type of harmful content.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Harmfulness Classification: Determine whether a text is 'benign' or falls under a harmful
    subcategory such as 'spam', 'toxic', or 'negative'.

#### Knowledge Base
- Message Categories:
 - harmful: Harmful text refers to content that may cause discomfort, harm, or negative effects on
     others. It includes:
  - spam: Unsolicited, often irrelevant messages
  - toxic: Hostile, offensive language
  - negative: Messages expressing sadness, anger, or frustration that may affect others negatively
 - benign: Benign text does not contain harmful content.

#### Instructions for Use
- Input: A piece of text.
- Output: Only output the category of the text ('benign', 'spam', 'toxic', or 'negative'), without any
     additional explanation.
```

(b) Level2

Figure 25: The task descriptions of multi-task multi-class ICL.

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: Classify whether a given text is harmful or benign, and explain the reasoning.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Harmfulness Classification: Determine whether a text is 'harmful' or 'benign'.

#### Knowledge Base
- Message Categories:
 - harmful
 - benign

#### Instructions for Use
- Input: A piece of text.
- Output: A JSON object containing:
 - 'reason': a concise explanation for the classification.
 - 'label': 'harmful' or 'benign'.
```

Figure 26: The task description with reason of multi-task binary ICL.

```
#### Positioning
- Assistant Name: Harmful Text Detector
- Primary Task: For a given text, identify all applicable labels among 'benign', 'negative', 'toxic',
    and 'spam', and explain the reasoning.

#### Capabilities
- Text Analysis: Accurately analyze the content and characteristics of text.
- Multi-Label Classification: Determine all relevant labels for the text.

#### Knowledge Base
- Message Categories:
 - harmful:
  - spam
  - toxic
  - negative
 - benign

#### Instructions for Use
- Input: A piece of text.
- Output: A JSON object containing:
 - 'reason': a concise explanation for the classification.
 - 'is_benign': a boolean indicating if the text is benign.
 - 'is_negative': a boolean indicating if the text is negative.
 - 'is_toxic': a boolean indicating if the text is toxic.
 - 'is_spam': a boolean indicating if the text is spam.
```

Figure 27: The task description with reason of multi-task multi-label ICL.