Reinforcement Learning–Guided Adaptive Tuning for Out-of-Distribution Harmful Text Detection

Mengyu Xiang[1,2*] **Tinghao Chen**[1,2*] **Boxu Han**[1,2] **Qiudan Li**[1†] **Shu Wu**[1,2†]
Daniel Dajun Zeng[1,2]

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS)
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences

xiangmengyu2024@ia.ac.cn, tinghaochen2026@ia.ac.cn, hanboxu2025@ia.ac.cn,
qiudan.li@ia.ac.cn, shu.wu@nlpr.ia.ac.cn, zengdaniel@outlook.com

Abstract

As social media grows, harmful information spreads rapidly across platforms and evolves over time, showing cross-platform and cross-temporal variations. Existing methods rely on fixed model parameters during training, which fail to handle substantial semantic discrepancies, leading to Out-Of-Distribution (OOD) problems. While test-time tuning enables dynamic parameter adjustment, it may lead to excessive adaptation to individual samples. The key challenge is how to adapt to semantic variations during testing while preventing overfitting from continuous tuning. To tackle this issue, this paper proposes RLAT, a reinforcement learning (RL)–guided adaptive tuning method for harmful text detection. First, a tuning joint optimization module is designed to update parameters and adapt to semantic variations during testing. It tunes the model by optimizing consistency loss and applying word-level attention constraints to reduce over-reliance on local words and learn a more robust global representation. Then, to mitigate overfitting caused by continuous tuning, a RL–guided adaptive decision model is introduced to direct the tuning process. It reduces the influence of local samples by selecting data and controlling parameter updates, thereby improving overall test performance. Experimental results show that the RLAT outperforms state-of-the-art baselines in cross-platform and cross-temporal scenarios across multiple public datasets.

1 Introduction

With the rapid growth of social media, harmful content in the form of subtle satire and metaphors spreads across various platforms, evolving and iterating continuously over time (Fortuna and Nunes, 2018). As shown in Figure 1, users on platforms A and B respectively focus on using terms like

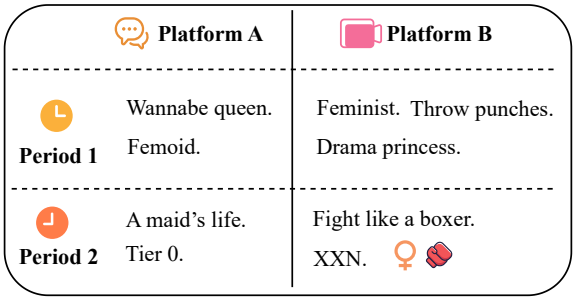


Figure 1: Examples of cross-platform and cross-temporal variations in harmful content.

"wannabe queen" and "feminist" to express gender-based antagonism, which then evolve into more covert and confrontational forms such as "Tier 0" and "xxn." "Tier 0" refers to radical feminists and "xxn" means "little fairy," both of which are used as derogatory terms against women in hostile or offensive contexts. These cross-platform and temporal differences lead to distribution shifts between the training and testing domains. Therefore, adaptively calibrating semantic representations during the testing phase to eliminate distribution shifts and achieve early detection of harmful text is crucial for purifying the online environment.

Existing methods for harmful text detection are primarily centered around contrastive learning, causal modeling, and knowledge-enhanced approaches. Contrastive learning and causal modeling improve generalization by extracting platform-independent features and true causal relationships within harmful semantics (Khondaker et al., 2023; Jiang, 2025). Knowledge-enhanced methods leverage dynamic knowledge graphs to enable the model to incorporate external semantic information beyond the training distribution (Xiang et al., 2025). Moreover, test-time tuning has emerged as an online adaptation method that allows model parameters to be updated during the testing phase (Liang et al., 2025). Recent studies generate high-

* These authors contributed to the work equally.
† To whom correspondence should be addressed.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35158–35174

confidence pseudo-labels from test samples and perform semi-supervised optimization to enhance the model's generalization ability on unseen topics (Gu et al., 2025).

However, above methods still struggle to adapt to distribution shifts at test time. *First*, most existing harmful text detection methods rely on fixed model parameters during training, which limits their ability to adapt to evolving data distributions and emerging harmful patterns. *Second*, although test-time tuning allows online parameter adaptation, current approaches typically apply a uniform strategy across all test samples. Continuous parameter tuning can lead to overfitting on individual samples, weakening the model's robustness to the overall test data. Thus, the key challenge is how to dynamically adjust the model during testing and reduce the negative impact of continuous tuning.

To resolve the issues, this paper proposes RLAT, a RL–guided adaptive tuning method for harmful text detection. It optimizes model parameters during the testing phase and introduces a decision model to guide the tuning process, thus improving the model's adaptability to OOD data. *First*, a tuning joint optimization module is designed to update model parameters and adapt to semantic variations during testing. It tunes the model by optimizing consistency loss and applying word-level attention constraints to reduce over-reliance on local words and learn a more robust global representation. *Second*, to mitigate overfitting caused by continuous tuning, a RL–guided adaptive decision model is introduced to adaptively direct the tuning process. This model selects samples for tuning and controls parameter updates based on text semantics and emotional features, thereby improving overall test performance. Experimental results show that the RLAT outperforms current state-of-the-art baselines on both cross-platform and cross-temporal datasets.

In summary, the contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to introduce a test-time tuning mechanism into harmful text detection.
- A RL–guided adaptive decision model is introduced to dynamically control the tuning process. By selecting samples for tuning and controlling parameter updates, it mitigates the performance degradation caused by continuous tuning.
- We construct multiple OOD harmful text detection scenarios based on multiple public datasets, including cross-platform and cross-temporal scenarios. Extensive experiments demonstrate that the proposed method achieves state-of-the-art performance, with improvements of up to 8.94% and 5.66% in cross-platform and cross-temporal settings.

2 Related Work

Existing methods for detecting harmful text focus on three main aspects: feature-based approaches, pre-trained model approaches, and Large Language Model (LLM) approaches (Zeng et al., 2025; Qiu et al., 2025). **Feature-based methods** extract harmful information by jointly modeling lexical, syntactic, semantic features, and user behavior patterns (Gitari et al., 2015; Chen et al., 2012). **Pre-trained model methods** involve further training and adaptation of models like BERT to improve detection performance in specific task scenarios (Caselli et al., 2021; Sarkar et al., 2021). **LLM-based methods** primarily focus on the study of harmful semantic reasoning. Yao et al. (2025) systematically explored the reasoning nature of harmful detection for the first time, by designing and comparing step-by-step and non-step-by-step reasoning methods. Yang et al. (2025) modeled harmful text into three reasoning dimensions: language, context, and emotion, and introduced a multi-dimensional reward model to optimize the model's understanding of harmful information. The above methods achieve strong performance in specific task settings. However, as the expression and semantics of harmful text evolve across platforms and over time, they often struggle with the resulting distribution shifts and perform poorly in OOD scenarios.

To improve generalization, contrastive learning (Kim et al., 2022), causal modeling (Zhang et al., 2023; Sheth et al.), and knowledge enhanced (Lu et al., 2024; Sridhar and Yang, 2022) have been adopted to mitigate distribution shifts. **Contrastive learning methods** improves the cross-platform detection ability of harmful text by aligning samples in the embedding space. Khondaker et al. (2023) proposed a meta-learning-based domain generalization method using gradient alignment to learn platform-invariant features. Ahn et al. (2024) shared semantics from training data as positive samples for contrastive learning, improving generalization while reducing annotation

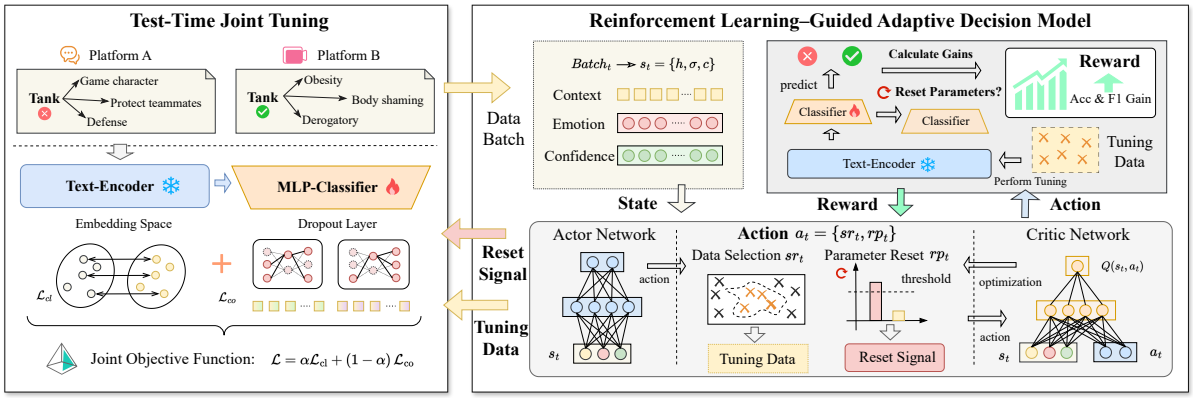


Figure 2: Architecture of the proposed method, comprising two components: (i) Test-Time Joint Tuning, which tunes model parameters in cross-platform scenarios through a joint optimization method; (ii) Reinforcement Learning–Guided Adaptive Decision Model, which mitigates performance degradation caused by continuous tuning by performing data selection and controlling parameter updates during tuning.

costs. **Causal modeling methods** mainly uncover the true causal relationships in harmful text. Jiang (2025) constructed positive sample sets from training prediction errors to mitigate spurious correlations. Sheth et al. (2023) introduced causal cues of emotional offensiveness to guide representation learning. **Knowledge-enhanced methods** improve generalization by incorporating external knowledge. Xiang et al. (2025) utilized a dynamic knowledge graph to reduce knowledge noise and conflicts, enabling fine-grained. Zhao et al. (2025) constructed a meta-knowledge graph for harmful content, injecting domain knowledge into LLMs through graph retrieval and ranking mechanisms.

The above methods rely on fixed parameters during training, which limits the model's adaptability to the test data. Existing research has introduced a test-time tuning mechanism that adjusts the classifier using high-quality supervised signals (Gu et al., 2025). However, this continuous parameter updating method may overfit to specific samples. This paper proposes a RL–guided adaptive tuning to mitigate overfitting resulting from continuous tuning.

3 Preliminary

OOD harmful text detection aims to address data shift across platforms or time periods, improving the model's early detection capabilities. Given source data $\mathcal{N}_P$ and target data $\mathcal{N}_T$, where each text has a label $y_i \in 0, 1$ (1 for harmful, 0 for appropriate), a model is trained on $\mathcal{N}_P$ and evaluated on $\mathcal{N}_T$. This paper proposes a RL–guided adaptive tuning method, which dynamically adjusts model parameters during testing to improve adaptability to unknown distributions.

4 Methodology

4.1 Test-Time Joint Tuning

Different platforms and time periods exhibit considerable variations in topic content and emotional expression. As shown in Figure 2, Platform A focuses on topic-oriented discussions, whereas Platform B centers on emotionally aggressive language, causing semantic shifts across platforms. For example, "tank" denotes a game character on Platform A but is used to demean physical appearance on Platform B, leading to cross-platform detection errors. Therefore, a local dependency suppression tuning method is introduced to mitigate over-reliance on local lexical features for semantic shift adaptation. It optimizes the consistency loss and applies word-level attention constraints to reduce the model's sensitivity to high-attention words, resulting in more robust global sentence representations.

First, high-attention clauses are extracted by segmenting the input text into clauses and computing word-level attention weights using the encoder. Clauses containing words with high attention scores are selected as high-attention clauses and used as negative samples. The model is then regularized to push the representations of these high-attention clauses away from the global sentence representation (Liu et al., 2025). The loss function is calculated as follows:

$$L_{\text{cl}} = \frac{1}{|N|} \sum_{i \in N} \max\left(0, \cos\left(\mathbf{h}_i^{\text{att}}, \mathbf{h}_i^{s}\right) - m\right), \quad (1)$$

where h_i^{att} and h_i^{s} are the representations of the high-attention clause and global sentence, $\cos(\cdot)$ is

the cosine similarity, m is a distance hyperparameter, and L_{cl} is the word-level attention loss.

Then, we introduce a consistency loss function to reduce the model's sensitivity to local perturbations in sentences. By minimizing the model's representations across different input perturbations, the model can produce consistent predictions when exposed to various input disturbances. The calculation is as follows:

$$\mathcal{L}_{\text{co}} = \frac{1}{|N|} \sum_{i=1}^{|N|} \|\mathbf{h}_i^1 - \mathbf{h}_i^2\|_2^2, \tag{2}$$

where h_i^1 and h_i^2 are the sentence representations under different dropout perturbations. $\mathcal{L}_{\text{co}}$ is the consistency loss. Finally, the word-level attention constraint L_{cl} and the consistency loss $\mathcal{L}_{\text{co}}$ are jointly optimized:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{cl}} + (1 - \alpha)\, \mathcal{L}_{\text{co}}, \tag{3}$$

where α is the weighting factor, and $\mathcal{L}$ is the joint optimization objective.

4.2 RL-Guided Adaptive Decision Model

Existing test-time tuning methods apply a uniform strategy for all test samples, which may fail to handle noisy or anomalous data and may impair model performance. To address this, we propose an adaptive test-time tuning strategy that dynamically selects the samples involved in tuning and adjusts the parameter update strategy via a decision model. Specifically, the decision model is trained on the validation set using RL, and then dynamically controls the tuning process at test time. It leverages semantic and emotional features to select samples and adaptively adjusts selection and update strategies based on tuning feedback.

As shown in Figure 2, the tuning process is formulated as a reinforcement learning problem, where each batch of samples is treated as an independent decision-making task. At each time step t, the decision model selects an action a_t according to the current state s_t and obtains a reward r_t following the tuning step.

State. For state representation, three components are considered: the pooled embedding of the samples $\boldsymbol{h}_t$, the standard deviation of sentiment scores $\boldsymbol{\sigma}_t$, and the prediction confidence $\boldsymbol{c}_t$. The pooled embedding $\boldsymbol{h}_t$ captures the contextual semantics of the samples within the batch, the standard deviation of sentiment scores $\boldsymbol{\sigma}_t$ reflects emotional fluctuation and textual uncertainty, and the prediction confidence $\boldsymbol{c}_t$ measures the certainty of the current model predictions. These features capture the multi-dimensional characteristics of the text and facilitate dynamic sample selection. Specifically, the state s_t is defined as follows:

$$s_t = \{\mathbf{h}_t, \boldsymbol{\sigma}_t, \mathbf{c}_t\}, \tag{4}$$

$$\boldsymbol{\sigma}_j = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (\mathbf{e}_k - \mu)^2}, \tag{5}$$

where σ_j is the j-th text in a batch, K is the number of words, e_k the sentiment score of the k-th word, and μ the mean word-level sentiment score.

Action. The action a_t consists of the sample selection ratio sr_t and the parameter reset probability rp_t. Specifically, the decision model randomly selects text from the current batch B_t based on the sample selection ratio, and then feeds the data into the prediction model for parameter updates and prediction (Sec. 4.1). Finally, the model determines whether to roll back the prediction model's parameters to their state prior to the current batch based on the reset probability rp_t.

$$a_t = \{sr_t, rp_t\}, \quad sr_t, rp_t \in [0, 1]. \tag{6}$$

Reward. The reward r_t is measured by the performance difference before and after tuning. This encourages the decision model to adopt tuning strategies that lead to performance improvements. The calculation is as follows:

$$r_t = \lambda_1 \Delta Acc(\theta_t^+, \theta_t^-) + \lambda_2 \Delta F1(\theta_t^+, \theta_t^-), \tag{7}$$

where λ_1 and λ_2 are weighting factors, and θ_t^+ and θ_t^- denote the parameters after and before tuning.

Objective Function. The policy network is optimized using a proximal algorithm with a clipped objective for stability (Schulman et al., 2017).

$$\mathcal{L}_w = \mathbb{E}\big[\min(\rho \hat{A}, \text{clip}(\rho, 1 - \epsilon, 1 + \epsilon)\hat{A})\big], \tag{8}$$

where ρ is the probability ratio between current and old policies, $\hat{A}$ the advantage function from generalized advantage estimation, and ϵ the clipping hyperparameter.

5 Experiments

5.1 Datasets

To evaluate performance in OOD scenarios, this paper constructed two types of evaluation datasets: cross-platform and cross-temporal. As shown in

	Dataset	Total	#Harm	%Harm	Avg.L
Cross Platform	Bilibili	4543	2294	50.50	41.22
	Tieba	4107	2359	57.44	37.26
	Weibo	8769	2943	33.56	70.69
	Zhihu	9716	3038	31.27	51.08
Cross Temporal	2012–2017	697	199	28.55	67.60
	2018	1695	384	22.65	64.00
	2019	1592	556	34.92	74.12
	2020	4785	1804	37.70	72.38

Table 1: OOD dataset statistics: Total samples, #Harm (number of harmful texts), %Harm (percentage of harmful texts), and Avg.L (average text length).

Table 1, the cross-platform dataset includes four social media platforms: Tieba, Zhihu, Weibo, and Bilibili, all derived from four publicly available datasets: ToxiCN (Lu et al., 2023), CDIAL-BIAS (Zhou et al., 2022), SWSR (Jiang et al., 2022), and TE-Dataset (Zhou et al., 2025). The cross-temporal dataset, based on Weibo post dates, is divided into four periods: 2012–2017, 2018, 2019, and 2020. In addition, the A-Distance value was calculated to quantify the differences between data from different sources (Ben-David et al., 2006). Detailed results are presented in Appendix A.2, revealing substantial discrepancies in cross-platform and cross-temporal data.

5.2 Baselines

Four types of methods were compared: pre-trained language models, causal and contrastive learning, test-time tuning, and LLM-based methods. The specific methods are detailed below:

Pre-trained language models

- BERT (Devlin et al., 2019): It is pre-trained via bidirectional encoding with masked language modeling and next-sentence prediction.
- RoBERTa (Liu et al., 2019): It improves BERT's training with dynamic masking, larger datasets, and longer training.

Causal and contrastive learning methods

- CCL (Jiang, 2025): It constructs positive samples from training prediction errors and aligns them in the representation space to mitigate reliance on spurious features.
- SCL-Fish (Khondaker et al., 2023): It uses gradient matching to extract invariant features, while supervised contrastive learning enhances task-specific representations.

Test-time tuning methods

- ConDA-TTT (Gu et al., 2025): It generates high-quality pseudo-labels at test time and updates classifier parameters to improve adaptability in unseen domains.
- ATTA-HC (Liang and Chen, 2025): It partitions parameters into shared and specific sets, updating only the specific ones at test time.

LLM-based methods

- LLM Few-Shot: This paper evaluates various LLMs in harmful text detection using a few-shot approach, detailed in Appendix A.7. The specific LLMs include GPT-4o (OpenAI, 2025), LLaMA3-8B (Meta, 2024), DeepSeek-V3.2 (DeepSeek-AI et al., 2025), GLM-4.6 (ChatGLM, 2025), and Qwen3-Max (Tongyi, 2025).
- MetaTox (Zhao et al., 2025): It builds a harmful content knowledge graph and injects it during reasoning.
- Sarcasm-R1 (Yang et al., 2025): It decomposes harmful text understanding into language, context, and emotion, and applies different reward models for multi-step reasoning.

5.3 Implementation Details

In our implementation, we adopt BERT as the backbone encoder to obtain sentence-level representations, which are then passed to a multilayer perceptron (MLP) for final classification.

The proposed method constructs sentiment-aware features by first assigning sentiment scores to each sentence using a general-purpose sentiment lexicon, and then computing the standard deviation of these scores within each sentence as a statistical indicator of sentiment fluctuation. This feature construction process does not involve any classifier training or target-domain parameter adaptation, and does not rely on test labels. The sentiment lexicon is a general resource independent of the dataset, and the computation is label-agnostic, without introducing test set information leakage or additional data advantage over baselines.

During training and evaluation, all parameters of the MLP are optimized. The dropout rate of the MLP is set to 0.5, while the weight factor α is set to 0.4. The balancing coefficients λ_1 and λ_2 are set to 0.4 and 0.6, respectively. The reset threshold is set to 0.6, and the clipping parameter ϵ is set to

Training Set	Tieba						Zhihu					
Test Set	Zhihu		Weibo		Bilibili		Tieba		Weibo		Bilibili	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT	62.76	56.18	64.37	61.29	87.13	87.62	72.87	77.01	51.65	56.44	73.76	77.45
RoBERTa	61.07	55.59	66.82	61.70	88.34	88.64	72.91	78.04	52.77	55.63	76.81	78.85
CCL	67.87	55.65	67.35	60.27	91.78	91.80	72.04	77.65	55.20	56.89	75.37	78.08
SCL-Fish	63.33	56.10	66.87	**62.78**	91.08	91.47	71.15	78.10	54.90	57.69	75.64	77.90
ConDA-TTT	64.35	57.15	64.92	62.58	92.01	92.16	73.22	77.94	60.22	57.94	74.63	76.80
ATTA-HC	57.39	43.12	66.95	60.22	91.30	91.68	72.66	77.90	54.45	57.26	76.49	79.11
RLAT	**69.22**	**58.01**	**69.25**	62.06	**92.37**	**92.50**	**74.25**	**78.88**	**61.02**	**58.20**	**77.61**	**79.71**
Improve	1.99	1.50	2.82	-1.15	0.39	0.37	1.41	1.00	1.33	0.45	1.04	0.76

Training Set	Weibo						Bilibili					
Test Set	Tieba		Zhihu		Bilibili		Tieba		Zhihu		Weibo	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT	60.58	50.95	64.72	48.88	67.93	58.75	85.89	86.77	64.35	57.25	66.14	62.40
RoBERTa	61.63	53.88	66.13	49.14	67.70	59.22	86.16	87.37	65.37	56.91	67.41	63.44
CCL	62.98	56.40	68.59	48.87	68.96	60.68	88.78	89.57	67.28	57.44	69.46	62.77
SCL-Fish	63.21	57.85	67.14	49.98	68.05	58.60	88.46	89.27	66.35	58.04	68.11	62.50
ConDA-TTT	61.49	53.77	66.53	48.39	70.48	64.53	89.54	90.42	67.55	57.90	69.49	64.01
ATTA-HC	61.50	53.66	**68.68**	47.80	68.94	58.04	88.98	89.81	59.72	43.49	67.92	62.82
RLAT	**64.83**	**60.16**	68.09	**50.45**	**71.42**	**66.20**	**90.05**	**91.11**	**73.59**	**58.88**	**71.53**	**64.78**
Improve	2.56	3.99	-0.86	0.94	1.33	2.59	0.57	0.76	8.94	1.45	2.94	1.20

Table 2: Comparison of cross-platform harmful text detection performance in terms of Accuracy(%) and F1 score(%). Bold values indicate the best performance, underlined values indicate the second-best performance. "Improve" represents the relative improvement of the proposed method over the second-best method.

0.1. More complete training details are provided in Appendix A.4.

For LLM-based evaluation, we uniformly sample 1,000 instances from each platform, with further details provided in Appendix A.3. Classification performance is reported in terms of accuracy and binary F1 score, averaged over five runs with different random seeds.

5.4 Cross-Platform Comparative Analysis

Table 2 shows the comparison results of the proposed method with pre-trained models, causal and contrastive learning, and test-time tuning methods in cross-platform scenarios. The proposed method achieves the best performance in the vast majority of experiments. Notably, a substantial improvement is observed when transferring from Bilibili to Zhihu. The accuracy and F1 score reach 73.59% and 58.88%, representing a relative increase of 8.94% and 1.45% compared to the second-best method's 67.55% and 58.04%. Meanwhile, the A-Distance score for both platforms is 1.48, reflecting a substantial difference in distribution. From the perspective of textual characteristics, Bilibili posts are shorter, more emotional, context-dependent, and aggressive. In contrast, Zhihu posts use standard vocabulary, show lower emotional intensity, and focus on discussion and opinion expression. The proposed method achieves relative gains of 8.94% and 1.45% in accuracy and F1 score. This improvement demonstrates that adaptive data selection and parameter updates via the decision model enable performance-driven tuning strategies, further enhancing model's cross-platform adaptability. More example analyses are in Appendix A.1. Moreover, ConDA-TTT outperformed all baselines in five scenarios, further demonstrating the effectiveness of test-time tuning methods.

5.5 Cross-temporal Comparative Analysis

Table 3 shows the comparative results of the proposed method with pre-trained models, causal and

Training Set	2012-2017						2018				2019	
Test Set	2018		2019		2020		2019		2020		2020	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT	76.67	56.40	73.03	65.68	71.97	66.19	75.78	69.44	74.13	69.77	75.77	70.62
RoBERTa	80.94	60.38	77.76	70.01	73.57	66.67	75.88	65.86	74.52	67.60	76.63	70.59
CCL	81.96	57.52	78.82	67.78	74.60	66.29	77.68	64.43	75.48	64.39	76.79	69.50
SCL-Fish	79.02	56.76	78.13	68.44	75.04	68.77	76.33	68.74	75.80	68.86	75.75	71.18
ConDA-TTT	81.75	55.67	78.20	70.16	75.80	68.32	76.57	66.81	75.90	69.56	75.80	70.50
ATTA-HC	81.36	58.01	78.56	68.46	75.92	68.19	75.13	67.41	76.80	69.67	76.53	69.22
RLAT	**82.60**	**63.80**	**80.00**	**73.58**	**76.28**	**70.23**	**78.93**	**72.43**	**77.74**	**70.04**	**77.43**	**71.65**
Improve	0.78	5.66	1.50	4.87	0.47	2.12	1.61	4.31	1.22	0.39	0.83	0.66

Table 3: Comparison of cross-temporal harmful text detection performance in terms of Accuracy(%) and F1 score(%). Bold indicates the best performance, underlined indicates the second-best. "Improve" denotes the relative gain of the proposed method over the second-best method.

contrastive learning, and test-time tuning methods across time scenarios. The proposed method consistently achieved the best performance across all experiments. From 2012–2017 to 2018, the proposed method's F1 score increased from 58.01% to 63.80%, achieving the largest relative improvement of 5.66% over the second-best method. Meanwhile, the A-Distance value between the 2012-2017 and 2018 data was 0.5882, representing the largest difference across time scenarios. In terms of topic content, posts from 2012-2017 were relatively scattered, primarily addressing equality and rights. In contrast, posts from 2018 centered on sexual violence and power, characterized by strong emotional language, explicit stances, and pronounced conflict. The improved performance demonstrates that the RLAT effectively captures the temporal evolution of harmful text under substantial cross-temporal distribution shifts, enhancing model generalization.

5.6 Comparison with LLM-Based Methods

The LLM-based method performs exceptionally well in detecting harmful text in OOD data. Table 4 shows the performance of different types of LLM methods across platforms. The proposed method consistently outperforms all baselines in the majority of comparative experiments. In the Bilibili to Tieba test, the accuracy and F1 score of the proposed method reached 90.05% and 91.11%. Compared to the second-best LLM method, the accuracy and F1 score are improved by 15.15% and 10.46%. This indicates that the proposed method effectively identifies harmful texts in OOD scenarios. In certain experiments with Weibo as the source platform, the LLM-based method slightly outperforms the proposed method. These results can be attributed to differences in method type and model parameter size. First, the LLM-based method employs a model with a large number of parameters parameters and strong representation capacity. It provides rich semantic understanding and achieves strong performance in harmful text detection. In contrast, the proposed method, built on a lightweight BERT+MLP architecture, offers notable advantages in computational efficiency and inference cost. Moreover, it outperforms LLMs in the majority of experiments while maintaining low resource consumption.

To demonstrate the efficiency advantages of RLAT in detail, we compare the inference time, peak memory usage, and parameter size of different models. On the same test set, the average inference time per sample for GPT-4o, DeepSeek-V3.2, and Qwen3-Max is 850 ± 383 ms, 754 ± 169 ms, and 529 ± 273 ms, respectively. In contrast, RLAT achieves an average inference time of only 5.92 ± 0.08 ms per sample, with an average fine-tuning time of 11.63 ms and an average policy training time of 219 ms, demonstrating significantly lower overall inference overhead compared to large language model methods. Regarding peak memory usage, LLaMA3-8B requires 15.13 GB during inference, while RLAT consumes 7.10 GB during policy training and only 3.29 GB during inference, resulting in a substantial reduction in memory consumption. In terms of parameter size, LLaMA3-8B exceeds 14.98 GB, whereas RLAT requires only 0.38 GB, further illustrating its clear advantage in

Training Set	Tieba						Zhihu					
Test Set	Zhihu		Weibo		Bilibili		Tieba		Weibo		Bilibili	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GPT-4o	57.24	**59.39**	58.65	61.86	68.00	73.33	65.00	70.59	55.50	56.59	69.50	72.15
Llama3-8b	49.50	54.55	42.50	52.91	67.50	69.77	64.50	69.79	52.20	55.82	58.51	69.94
Deepseek-v3.2	61.12	55.65	52.71	56.93	68.44	72.21	72.30	79.47	56.50	55.38	65.64	66.10
GLM-4.6	54.50	49.72	49.40	55.69	56.70	63.82	63.50	75.06	52.00	55.14	59.10	64.95
Qwen3-Max	49.00	55.26	51.50	55.70	62.00	70.99	70.50	80.40	54.80	56.70	62.50	70.36
Sarcasm R1	59.40	55.48	63.50	60.11	84.64	86.80	73.97	**81.46**	59.50	52.63	69.32	65.61
MetaTox	62.50	55.62	58.50	56.99	70.60	75.08	71.70	76.86	60.23	54.81	68.40	74.67
RLAT	**69.22**	58.01	**69.25**	**62.06**	**92.37**	**92.50**	**74.25**	78.88	**61.02**	**58.20**	**77.61**	**79.71**

Training Set	Weibo						Bilibili					
Test Set	Tieba		Zhihu		Bilibili		Tieba		Zhihu		Weibo	
Method	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GPT-4o	68.50	73.86	63.20	**57.89**	71.00	**74.11**	68.50	73.86	63.00	56.98	59.00	57.73
Llama3-8b	75.80	81.33	49.00	52.78	58.53	69.95	63.50	75.06	48.70	53.99	57.70	52.20
Deepseek-v3.2	**76.20**	**81.80**	60.50	57.75	70.47	72.86	68.88	78.67	64.20	57.18	51.15	56.01
GLM-4.6	63.50	75.06	57.50	53.55	60.60	65.68	68.30	75.78	65.00	54.30	54.40	56.24
Qwen3-Max	74.00	79.84	57.00	57.00	64.00	71.88	71.00	79.29	51.50	54.30	52.00	61.60
Sarcasm R1	71.05	74.21	**68.34**	56.81	69.62	69.66	72.70	79.64	68.50	50.62	67.05	57.05
MetaTox	69.00	76.19	63.00	56.98	69.80	73.42	78.20	82.48	70.40	61.05	59.50	51.50
RLAT	64.83	60.16	68.09	50.45	**71.42**	66.20	**90.05**	**91.11**	**73.59**	**58.88**	**71.53**	**64.78**

Table 4: Comparison of Accuracy(%) and F1(%) score between the proposed method and LLM-based methods. Bold values indicate the best performance.

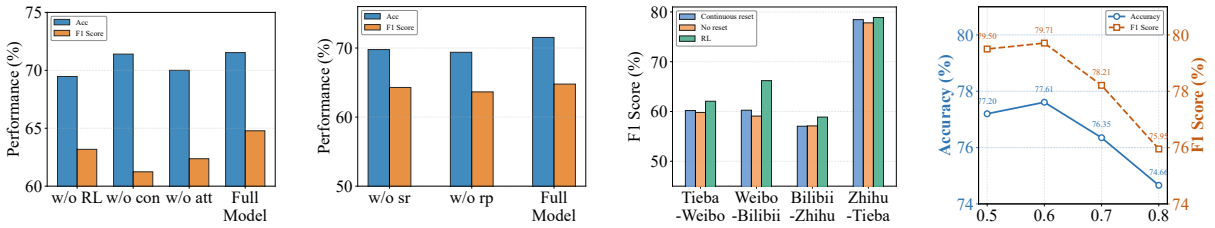
computational efficiency and resource overhead.

5.7 Ablation and Sensitivity Analysis

Figures 3a shows the ablation analysis results of the proposed method on Bilibili to Weibo. "w/o RL," "w/o con," and "w/o att" denote the removal of the decision model, the consistency loss, and the word-level attention constraint, respectively. The results show that all components are important in the proposed method. The removal of the decision model resulted in a 2.06% reduction in accuracy and a 1.60% reduction in F1 score. This indicates that selectively updating parameters based on data stabilizes the model's performance and mitigates the negative impact of continuous tuning.

Figure 3b illustrates the ablation results for the adaptive decision model. "w/o sr" and "w/o rp" represent the removal of data selection and parameter reset in the action space. Notably, removing the parameter reset rp led to the largest drops in accuracy and F1 score, 2.15% and 1.13%. This suggests that selectively applying parameter updates during tuning effectively mitigates the adverse effects of continuous updates, enhancing the stability of the model's adaptation at test time.

Figure 3c presents a comparison of the performance of various tuning strategies across different cross-platform scenarios. "No reset" indicates continuous tuning without parameter resets, "Continuous reset" denotes resetting parameters after each batch, and "RL" represents the decision model-based tuning strategy in the proposed method. The results show that the RL-based adaptive parameter reset strategy achieves better performance in the four cross-platform scenarios. Specifically, except for the Bilibili to Zhihu, where the "No reset" strategy slightly outperforms "Continuous reset," the latter achieves superior performance in the remaining three scenarios. This suggests that resetting parameters after each batch mitigates the negative effects of local data updates, thereby enhancing overall detection performance. In addition, the RL–guided adaptive decision model can autonomously determine whether to reset parameters



(a) Ablation analysis from Bilibili to Weibo.

(b) Ablation analysis of sr and rp from Bilibili to Weibo.

(c) Comparison of tuning strategies

(d) Sensitivity analysis of resetting threshold.

Figure 3: Ablation studies and parameter sensitivity analysis. (a) and (b) show cross-platform ablation results. (c) compares the F1 of different tuning strategies. (d) compares the accuracy and F1 of different reset thresholds.

during tuning, maximizing the benefits of parameter updates.

Figure 3d illustrates the sensitivity analysis of the hyperparameter reset threshold in the Zhihu to Bilibili scenario. Experimental results show that when the reset threshold is 0.6, the proposed method achieves the highest accuracy and F1 score of 77.61% and 79.61%. As the reset threshold increases beyond 0.6, the model performance gradually decreases, approaching the performance without parameter reset. This indicates that continuous parameter updates during testing make the model susceptible to noise, leading to skew. However, exploring reset action probability through reinforcement learning offers a more stable tuning strategy, effectively mitigating interference.

6 Conclusion

This paper proposes RLAT, a RL–guided adaptive tuning for harmful text detection. First, it designs a local dependency suppression loss to alleviate the model's tendency to over-reliance on local words, achieved by optimizing a consistency loss and enforcing word-level attention constraints. Then, a RL–guided adaptive decision model is introduced to dynamically control data selection and parameter updates. Finally, cross-platform and cross-temporal experiments demonstrate the effectiveness of the RLAT in harmful text detection.

Limitations

While the proposed method is highly effective in harmful text detection, it still has certain limitations. Current methods primarily focus on binary classification for harmful content detection. Future work could extend the RLAT method to multi-class detection, enabling fine-grained OOD detection of various types of harmful content, such as malicious attacks, misinformation, and discriminatory speech. In addition, it could also explore modality-level shifts, thereby facilitating more generalizable approaches.

Ethical Considerations

Our work primarily provides a harmful text detection method that allows for adaptive model adjustments during the testing phase. The data used in our experiments are all from publicly available datasets to ensure transparency and reproducibility. Our use of the data is consistent with the scientific research intent of the original paper. In addition, all data is anonymized, and no personal information is disclosed. It is important to emphasize that our goal is to improve the detection of harmful text, not to exacerbate its harm.

Acknowledgments

This research was partially funded by the National Natural Science Foundation of China under Grant Nos. 72293575 and 62372454.

References

Hyeseon Ahn, Youngwook Kim, Jungin Kim, and Yo-Sub Han. 2024. Sharedcon: Implicit hate speech detection using shared semantics. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 10444–10455.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 137–144.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In

Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), pages 17–25, Online.

ChatGLM. 2025. Glm-4.5.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, and 2012 International Confernece on Social Computing, SocialCom 2012, Amsterdam, Netherlands, September 3-5, 2012*, pages 71–80.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4):85:1–85:30.

Njagi Dennis Gitari, Zuping Zhang, Zuping Zhang, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. In *International Conference on Multimedia and Ubiquitous Engineering*.

Yimeng Gu, Mengqi Zhang, Ignacio Castro, Shu Wu, and Gareth Tyson. 2025. Contrastive domain adaptation with test-time training for out-of-context news detection. *Pattern Recognit.*, 164:111530.

Aiqi Jiang, Xiaohan Yang, Yang Liu, and Arkaitz Zubiaga. 2022. SWSR: A chinese dataset and lexicon for online sexism detection. *Online Soc. Networks Media*, 27:100182.

Tianming Jiang. 2025. Learn from failure: Causality-guided contrastive learning for generalizable implicit hate speech detection. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 8858–8867.

Md Tawkat Islam Khondaker, Muhammad Abdul-mageed, and Laks Lakshmanan, V.s. 2023. Cross-platform and cross-domain abusive language detection with supervised contrastive learning. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 96–112.

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

Jian Liang, Ran He, and Tieniu Tan. 2025. A comprehensive survey on test-time adaptation under distribution shifts. *Int. J. Comput. Vis.*, 133(1):31–64.

Yunsheng Liang and Kai Chen. 2025. Automatic test-time adaptation for heterogeneous contexts in meta-learning. *Neural Comput. Appl.*, 37(18):12631–12652.

Qiang Liu, Xinlong Chen, Yue Ding, Bowen Song, Weiqiang Wang, Shu Wu, and Liang Wang. 2025. Attention-guided self-reflection for zero-shot hallucination detection in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21016–21032.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023. Facilitating fine-grained detection of chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16235–16250.

Junyu Lu, Bo Xu, Xiaokun Zhang, Hongbo Wang, Haohao Zhu, Dongyu Zhang, Liang Yang, and Hongfei Lin. 2024. Towards comprehensive detection of chinese harmful memes. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Meta. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

OpenAI. 2025. Gpt-4o.

Ziqi Qiu, Jianxing Yu, Yufeng Zhang, Hanjiang Lai, Yanghui Rao, Qinliang Su, and Jian Yin. 2025. Detecting emotional incongruity of sarcasm by commonsense reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 9062–9073. Association for Computational Linguistics.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander G. Ororbia II. 2021. fbert: A neural transformer for identifying offensive content. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana,*

Dominican Republic, 16-20 November, 2021, pages 1792–1798.

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

Paras Sheth, Tharindu Kumarage, Raha Moraffah, Aman Chadha, and Huan Liu. 2023. PEACE: cross-platform hate speech detection - A causality-guided framework. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part I*, volume 14169 of *Lecture Notes in Computer Science*, pages 559–575.

Paras Sheth, Raha Moraffah, Tharindu S. Kumarage, Aman Chadha, and Huan Liu. Causality guided disentanglement for cross-platform hate speech detection. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pages 626–635.

Rohit Sridhar and Diyi Yang. 2022. Explaining toxic text via knowledge enhanced text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–826, Seattle, United States. Association for Computational Linguistics.

Tongyi. 2025. Qwen-max.

Mengyu Xiang, Yuxuan Song, Qiudan Li, Shu Wu, and Daniel Dajun Zeng. 2025. Dynamic detection of sarcasm topic-target pairs via llm-based knowledge alignment. In *Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025*, pages 1422–1425.

Qi Yang, Jingjie Zeng, Liang Yang, Kai Ma, and Hongfei Lin. 2025. Sarcasm-r1: Enhancing sarcasm detection through focused reasoning. *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2025. Is sarcasm detection a step-by-step reasoning process in large language models? In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 25651–25659.

Jingjie Zeng, Liang Yang, Zekun Wang, Yuanyuan Sun, and Hongfei Lin. 2025. Sheep's skin, wolf's deeds: Are llms ready for metaphorical implicit hate speech? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 16657–16677.

Zhehao Zhang, Jiaao Chen, and Diyi Yang. 2023. Mitigating biases in hate speech detection from A causal perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6610–6625.

Yibo Zhao, Jiapeng Zhu, Can Xu, Yao Liu, and Xiang Li. 2025. Enhancing llm-based hatred and toxicity detection with meta-toxic knowledge graph. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24747–24760.

Gang Zhou, Haizhou Wang, Di Jin, Wenxian Wang, Shuyu Jiang, Rui Tang, and Xingshu Chen. 2025. A toxic euphemism detection framework for online social network based on semantic contrastive learning and dual channel knowledge augmentation. *Inf. Process. Manag.*, 62(5):104143.

Jingyan Zhou, Jiawen Deng, Fei Mi, Yitong Li, Yasheng Wang, Minlie Huang, Xin Jiang, Qun Liu, and Helen Meng. 2022. Towards identifying social bias in dialog systems: Framework, dataset, and benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3576–3591.

A Appendix

A.1 Case Study

To further analyze the adaptive tuning method, we conducted a case study using a cross-platform example. We compared the prediction results of "No Tuning," "No-strategy Tuning," and "Adaptive Strategy Tuning" on specific samples. "No Tuning" refers to a model without tuning, "No-strategy Tuning" refers to a method that updates parameters using the same strategy across all samples, and "Adaptive Strategy Tuning" refers to the tuning strategy of the proposed method.

As shown in figure 6, the training platform includes many texts that use the term "little fairy" to attack and defame. The prediction model learns to associate "little fairy" with harmful labels. However, the test platform contains new expressions, such as "a mediocre woman with overconfidence", which were not seen during training. The "No Tuning" approach fails to adapt to these changes, resulting in incorrect predictions for all samples except T2 and T6. The "No-strategy Tuning" approach, by reducing over-reliance on local features, successfully identifies harmful semantics similar to "little fairy," predicting T1 as harmful. At the same time, tuning on data similar to T3 and T4 allows the model to classify marriage-related semantics as harmless. However, the model misclassifies T5 and T6, which contain defamatory content about marriage. The "Adaptive Strategy Tuning" method mitigates overfitting and reduces the risk of incorrect predictions. It retains the model's ability to identify harmful texts in training samples similar to S-n, which uses the "marriage donkey" for defamation, improving overall detection performance.

A.2 A-Distance Calculation

To quantify the differences between data distributions, A-distance is employed, which measures distributional divergence based on the error rate of a domain classifier. Specifically, let $h^* \in \mathcal{H}$ denote the optimal domain classifier that minimizes the classification error when distinguishing samples drawn from the source distribution P_S and the target distribution P_T. The A-distance is then defined as

$$d_{\mathcal{A}}(P_S, P_T) = 2\left(1 - 2\epsilon(h^*)\right), \tag{9}$$

where $\epsilon(h^*)$ is the minimum achievable error rate of the domain classifier.

As shown in Figure 4 and 5, the values range from 0.1400 to 1.5067 for cross-platform datasets and from 0.1647 to 0.5882 for cross-temporal datasets. This indicates that there are substantial differences in topic content and language style across different platforms, with text on each platform also evolving over time.

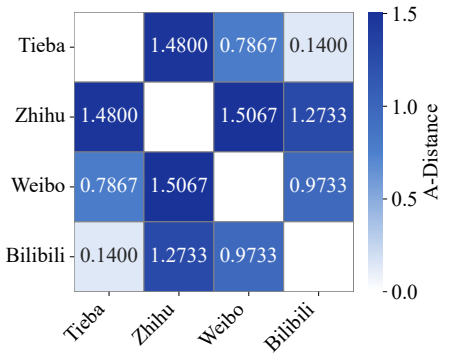


Figure 4: Cross-platform A-Distance scores.

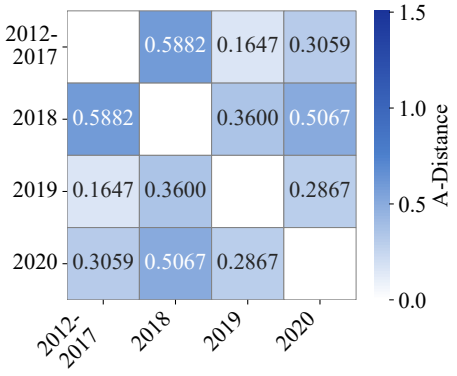


Figure 5: Cross-temporal A-Distance scores.

A.3 LLM Evaluation Dataset

To evaluate the performance of LLMs in cross-platform scenarios, an evaluation dataset was constructed, with its statistics summarized in Table 5. For each platform, 1,000 samples were uniformly sampled with respect to text length and label distribution to form the test set, ensuring that it remains representative of the original dataset.

Dataset	Bilibili	Tieba	Weibo	Zhihu
#Harmful	505	575	335	313
#Harmless	495	425	665	687
Avg.L	41.11	37.39	70.12	51.22

Table 5: LLM evaluation dataset statistical metrics. #Harmful represents the number of harmful texts, #Harmless represents the number of harmless texts, and Avg.L represents the average text length.

Cross-platform Case Study

Training Platform

S-1: Forget the facts, does the 'little fairy' of the new era even have the strength to find someone to raise her kids? Label: Harmful.

S-2: Why should the 'little fairy' reflect on herself when she can always find someone else to blame? Label: Harmful.

S-3: This wine is so precious, it should be the 'little fairy' who drinks it first. Label: Harmful.

⋮

S-n: MARRIAGE DONKEY is just inferior!!!!!!! Label: Harmful.

Testing Platform

T-1: I remember this is a list of conditions written by a mediocre woman with overconfidence about men, hurry up and pin it to shame her! Label: Harmful.

T-2: The 'little fairy' must've lived like this in her past life. Label: Harmful.

T-3: No way, is anyone really getting married? What could you possibly lack? Money? Nothing to worry about. Label: Harmless.

T-4: That's why marriage tests are so important. Label: Harmless.

T-5: Dream on, who hasn't been cheated on before marriage? Where would you even file a lawsuit? Label: Harmful.

T-6: Of course, if you put yourself in the shoes of a marriage donkey, you'd think you were being insulted. Haha. Label: Harmful.

Comparison of Prediction Results

Method	T1	T2	T3	T4	T5	T6
No Tuning	✗	✓	✗	✗	✗	✓
No-strategy Tuning	✓	✓	✓	✓	✗	✗
Adaptive Strategy Tuning	✓	✓	✓	✓	✓	✓

Figure 6: Cross-platform case study. It includes three methods: "No Tuning," "No-strategy Tuning," and "Adaptive Strategy Tuning."

A.4 RL Training Details and Analysis

We provide additional details on the RL training process, model configuration, and generalization behavior. The policy network is optimized using the PPO framework. In each RL round, the model performs multiple interaction steps on the validation set to collect trajectories, compute rewards, and update the policy. The reward is defined based on the performance improvement of the prediction model, measured by accuracy (Acc) and F1 score on the validation set.

The policy network takes a 770-dimensional input, consisting of the 768-dimensional BERT [CLS] representation, a sentiment standard deviation, and a confidence score. It uses a hidden layer of size 128 and outputs a 2-dimensional action space corresponding to the sample selection ratio and reset probability. The value network adopts the same input and hidden dimensions, but outputs a scalar value for state evaluation. Due to the use of BERT-based state encoding, the memory usage consists of both the encoder and the RL model, with a peak of 7.10GB during training and only 3.29GB during inference. The average policy training time is 219ms.

The detailed hyperparameters are summarized in Table 6. Briefly, batch size controls the number of samples per batch; epochs_pred is the number of training epochs for the source-domain prediction model (BERT+MLP); max_len defines the maximum input length; mlp_hidden_dim is the hidden size of the classifier; margin and alpha are loss coefficients; lr_policy and lr_value are the learning rates of the policy and value networks; clip_epsilon is the PPO clipping parameter; ppo_epochs is the number of optimization epochs per trajectory; lambda1 and lambda2 are the weights of Acc and F1 in the reward; and rl_episodes and rl_steps_per_episode control the RL training schedule.

We provide more detailed information about rl training and discuss the risk of overfitting in the method. The RL policy network is trained on the source-domain validation set, with rewards ΔAcc and ΔF1 computed from its true labels. At test time in the target domain, the network outputs actions solely from state features, without access to any test labels, fully complying with the label-free test-time setting. Regarding the risk of overfitting in the policy network, firstly, the policy input is a statistical state rather than specific sample semantics, reducing dependence on a specific distribution. Secondly, the samples used for tuning are randomly selected based on the policy network's actions. This random mechanism avoids the model repeatedly over-optimizing on the same class of high-confidence or high-uncertainty samples, thereby reducing dependence on local noise. Finally, we tested the model in 12 cross-platform and 6 cross-time experiments, covering diverse distribution scenarios, as shown in Tables 2, 3, and 4. The results show that performance gains are stable across most tasks, indicating that the policy learns general tuning and reset rules rather than overfitting to the validation domain.

Hyperparameter	Value
batch_size	16
epochs_pred	3
max_len	128
mlp_hidden_dim	256
margin	0.5
alpha	0.4
lr_policy	1×10^{-4}
lr_value	1×10^{-3}
clip_epsilon	0.1
ppo_epochs	4
lambda1	0.4
lambda2	0.6
rl_episodes	100
rl_steps_per_episode	10

Table 6: Hyperparameter settings for RL training and prediction models in RLAT.

A.5 Comparison and Analysis of Adaptation Strategies

We further provide three groups of supplementary experiments, including comparisons with standard test-time adaptation (TTA) methods, tuning variants with different parameter configurations, and sample-level RL policies, as shown in Table 7.

For TTA comparison, ConDA-TTT and ATTA-HC have already been evaluated in Tables 2 and 3, which are representative TTA variants and their effectiveness has been verified in prior work. Our results show that RLAT consistently outperforms these methods in both cross-platform and cross-temporal OOD settings. We further conduct two additional cross-platform experiments under standard TTA settings: (1) direct test-time tuning using pseudo-labels; (2) combining the proposed RL

Training Set	Tieba		Zhihu		Bilibili	
Test Set	Zhihu	Bilibili	Tieba	Bilibili	Tieba	Zhihu
Standard TTA	55.31	89.40	77.83	78.94	87.68	56.91
RL+Standard TTA	56.97	91.93	78.31	78.56	90.14	57.24
Full Tuning	56.16	90.90	78.43	79.33	89.58	58.35
MLP+BERT(Last 4)	55.97	91.45	78.69	79.37	90.02	57.57
Sample-Level	57.27	91.80	77.38	78.08	90.58	58.55
RLAT	**58.01**	**92.50**	**78.88**	**79.71**	**91.11**	**58.88**

Table 7: Comparison of adaptation strategies, including standard test-time adaptation (TTA) methods, tuning variants with different parameter configurations, and sample-level RL policies, in terms of F1 score (%). Bold values indicate the best performance.

adaptive decision framework with pseudo-label-based TTA. As shown in the table, RLAT achieves the best performance in all settings, while standard TTA methods benefit when integrated with the RL framework, demonstrating the effectiveness of both the tuning mechanism and the adaptive policy.

For tuning variants with different parameter configurations, we compare two additional settings: (1) jointly fine-tuning BERT and MLP; (2) fine-tuning the last four layers of MLP+BERT. Both approaches update encoder-level semantic representations during test-time adaptation. Results show that fine-tuning only the MLP achieves better and more stable performance. This suggests that updating the encoder may disrupt pre-trained semantic structures, amplify noise under distribution shifts, and increase overfitting risk. In contrast, freezing the encoder and updating only the classification head preserves the semantic space and leads to more robust OOD performance.

To further analyze adaptive decision-making, we compare batch-level and sample-level RL strategies in terms of performance, efficiency, and statistical properties. Six cross-platform experiments show that the batch-level policy consistently outperforms the sample-level policy in F1 score, while also being more efficient. Specifically, the sample-level policy requires 337.53 ms and 6.63 GB memory on average, whereas the batch-level policy only requires 219.95 ms and 6.26 GB memory. This is because sample-level policies make decisions for each instance independently, leading to computational costs that scale with batch size, while batch-level policies operate on global batch statistics with fixed complexity.

Finally, we analyze the correlation between sr_t and rp_t using 608 batches. The results show $sr_t = 0.5653 \pm 0.4318$ and $rp_t = 0.5800 \pm 0.4245$, with a Pearson correlation coefficient of 0.1059 (p = 0.0089) and $R^2 = 0.0112$, indicating very weak linear correlation. This suggests that the policy does not directly couple sample ratio and rollback probability, but instead assigns them independent roles in the policy space.

A.6 Additional LLM Comparisons

We further add two sets of performance comparison experiments, including fine-tuned LLMs and prompt-based LLMs, to better verify the effectiveness of RLAT.

For fine-tuned LLMs, we conduct six cross-platform experiments on Tieba, Zhihu, and Bilibili based on Qwen3-8B and LLaMA3-8B with LoRA fine-tuning. As shown in Table 8, the performance of fine-tuned LLMs improves on the Tieba and Zhihu datasets, while it decreases on the Bilibili dataset. This is mainly due to overfitting caused by cross-platform distribution shifts. Taking the transfer from Bilibili to Zhihu as an example, Figure 4 shows that the A-distance reaches 1.27, indicating a significant platform gap. Specifically, Bilibili text is more colloquial, consisting of short bullet-screen style phrases with strong emotional expressions, while Zhihu content is relatively longer, more argumentative, and structurally organized. Fine-tuned LLMs tend to capture platform-specific patterns, thereby weakening their cross-platform generalization ability. Overall, RLAT achieves consistent improvements across all test settings and effectively mitigates overfitting via adaptive decision-making.

For prompt-based LLMs, we conduct additional experiments on six cross-platform scenarios using

Training Set	Tieba		Zhihu		Bilibili	
Test Set	Zhihu	Bilibili	Tieba	Bilibili	Tieba	Zhihu
Qwen3-8b	53.78	70.77	64.90	70.66	79.06	56.32
Qwen3-8b-LoRA	58.71	88.11	70.37	66.09	74.49	53.35
Llama3-8b	54.55	69.77	69.79	69.94	75.06	53.99
Llama3-8b-LoRA	54.50	75.02	79.47	74.62	79.64	53.94
Qwen3-Max	55.26	70.99	80.40	70.36	79.29	54.30
Qwen3-Max-CoT	**60.42**	73.42	80.43	77.55	75.68	58.13
Qwen3-Max-SC	58.82	73.02	**81.53**	65.56	75.76	56.07
GPT-4o	59.39	73.33	70.59	72.15	73.86	56.98
GPT-4o-CoT	50.00	74.63	73.74	59.50	80.75	53.21
GPT-4o-SC	59.74	73.08	73.04	68.57	80.95	57.14
RLAT	58.01	**92.50**	78.88	**79.71**	**91.11**	**58.88**

Table 8: Comparison of cross-platform harmful text detection performance in terms of F1 score (%). SC denotes the self-check prompting mechanism, and CoT denotes the chain-of-thought prompting strategy. Bold values indicate the best performance.

Qwen3-Max and GPT-4o with chain-of-thought and self-check prompting strategies. As shown in Table 8, RLAT achieves the best performance in most settings except the Tieba-to-Zhihu and Zhihu-to-Tieba scenarios. This indicates that RLAT's adaptive tuning mechanism is more effective than prompt engineering strategies. Overall, prompt-based improvements remain limited for harmful text detection, while RLAT consistently provides stronger and more stable gains across domains.

A.7 Prompt

Harmful Text Detection Prompt

Please determine whether the following text contains harmful content. If YES, output 1; if NO, output 0.

Task Description: Harmful content refers to any text that may cause harm to individuals or society, including but not limited to: incitement of violence, hate speech, discriminatory content, extremist viewpoints, malicious personal attacks, harassment, threats, and toxic behavior. Carefully analyze the text's intent, potential impact, linguistic style, and social media context.

Examples:

Text: If you can't control your wife, you reap what you sow. Since you support women's independence, you should accept it when women have momentary lapses in judgment.
Label: 1
Text: Looks like both of you are on the same side.
Label: 0
Text: Ordinary confident women are terrifying. Let's go cyberbully her, bros.
Label: 1
Text: Even honest officials find it hard to settle family disputes; every family has its own problems.
Label: 0
Text: Disgusting homosexuals, that's the precise term.
Label: 1
Text: It's better to clarify things. You might have other thoughts about her now, and other women might use the same reasoning against you in the future.
Label: 0
Based on the above examples, determine whether the following text contains harmful content. Output ONLY 1 or 0, without any additional explanation:

Text: [Text to be detected]
Label: