ReCAST: Restoration-aware Cascaded Stage-wise Training for Obfuscated SMS Risk Classification

Jieyun Huang, Yi Shen*, Kaikai Zhao, Jiangze Yan, Wenjing Zhang, Ping Chen, Ning Wang, Zhaoxiang Liu, Kai Wang, Shiguo Lian*
Unicom Data Intelligence, China Unicom
Data Science & Artificial Intelligence Research Institute, China Unicom
{huangjy115, sheny73, zhaokk3, yanjz17, liansg}@chinaunicom.cn

Abstract

Fraudulent messages sent via Short Message Service (SMS) are increasingly obfuscated to evade cost-conscious classifiers in production systems. In Chinese SMS, attackers can exploit a wide range of carefully crafted obfuscation strategies to hide risk-bearing phrases while preserving human readability, making direct classification brittle under real-world latency and throughput constraints. We propose ReCAST, a Restoration-aware Cascaded Stage-wise Training framework for robust obfuscated Chinese SMS classification. ReCAST distills a large teacher model's de-obfuscation ability into a smaller deployable student model by supervising obfuscated span detection, obfuscation type prediction, and text restoration, and then uses the restoration-aware student for downstream risk classification. Experiments on an internally constructed real-world Chinese SMS benchmark show that ReCAST substantially improves classification performance over directly trained baselines under obfuscation. The results suggest that restoration-aware distillation offers a practical path toward robust SMS risk classification with smaller deployable models under production-oriented constraints.

Figure 1: Representative obfuscation strategies in Chinese SMS. Attackers hide risk-bearing phrases while preserving human readability with these strategies, making direct classification brittle.

1 Introduction

Short message service (SMS) remains widely used, but is also abused for fraud, gambling, illegal promotion, and other risky activities. Industrial SMS risk detection systems must classify massive streams under strict latency, throughput, and cost constraints, making cost-conscious classifiers common for online inference. However, these classifiers are increasingly challenged by intentionally obfuscated SMS designed to evade detection.

Chinese SMS obfuscation is especially challenging because attackers can exploit the characteristics of the Chinese writing system and user reading habits. As shown in Figure 1, risk-bearing phrases can be hidden through carefully designed language-specific transformations, often combined with generic noisy perturbations, while remaining understandable to human recipients. [1] These transformations preserve the intended meaning but remove, fragment, or distort the surface forms that classifiers, keyword rules, and tokenizers rely on, making the original risk indicators no longer directly observable in the input sequence.

This creates a practical robustness gap in deployed SMS risk detection systems. Rule-based methods and keyword matching are easily by-

*Corresponding authors.

[1] Please refer to Appendix F for the complete Obfuscation Taxonomy.

EMNLP 2026 Industry Track, pages 3915–3931
October 24-29, 2026 ©2026 Association for Computational Linguistics

passed by surface-level modifications, while directly trained cost-conscious classifiers often fail when key risk indicators are transformed or fragmented. Large language models (LLMs) can often infer the intended meaning behind obfuscated text and restore normalized expressions, but invoking them for every incoming SMS is impractical in production due to latency, cost, and concurrency requirements. The central challenge is therefore how to transfer the de-obfuscation ability of LLMs into smaller models intended for online deployment.

To address this challenge, we propose ReCAST, a Restoration-aware Cascaded Stage-wise Training framework for robust classification of obfuscated Chinese SMS. ReCAST distills the de-obfuscation ability of a large teacher model into smaller deployable models without using the teacher during online inference. In the first stage, a student model learns structured restoration-oriented prediction, including obfuscated span detection, obfuscation type, and text restoration. In the second stage, the restoration-aware student is further trained for downstream risk classification. This design enables smaller deployable models to internalize de-obfuscation signals while maintaining single-pass inference efficiency in our serving setup.

We evaluate ReCAST on Chinese SMS datasets covering multiple risk categories. Compared with directly trained classification baselines, ReCAST achieves substantial improvements under obfuscation. Ablation studies confirm the effectiveness of restoration supervision and stage-wise training, while deployment-oriented evaluation shows that ReCAST preserves efficiency close to same-backbone online classification baselines. These results suggest that restoration-aware distillation provides a practical way to improve robustness against diverse real-world SMS obfuscation patterns.

Our contributions are summarized as follows:

- We characterize the industrial challenge of obfuscated Chinese SMS risk detection and discuss how various obfuscation strategies can make classifiers brittle in real-world settings.
- We propose **ReCAST**, a restoration-aware Stage-wise distillation framework that transfers de-obfuscation ability from a large teacher model to cost-conscious classifiers.
- We demonstrate that ReCAST improves robustness under obfuscation while maintaining latency close to same-backbone single-pass classification baselines in a controlled serving setup.

2 Related Work

SMS abuse detection under adaptive obfuscation. SMS abuse detection is commonly studied as a text-classification task, with recent approaches adopting BERT-style classifiers and LLM-based detectors (Liu et al., 2021; Oswald et al., 2022; Salman et al., 2025). For industrial risk-control systems processing massive SMS streams under stringent latency, throughput, and cost constraints, lightweight LLM-based classifiers are more practical than directly serving large LLMs (Zhou et al., 2024). However, adaptive obfuscation challenges their robustness: malicious senders rewrite risk-bearing expressions into readable variants to evade detection (Hosseinpour and Das, 2025). This issue is particularly salient in Chinese SMS, where studies on Chinese spam detection show that homophonic, glyph-level, and sub-character variants can preserve readability while concealing risky content (Jiang et al., 2020; Yao et al., 2022; Lai et al., 2022), and studies on Chinese offensive-language detection report similar cloaking effects (Xiao et al., 2024; Wu et al., 2025; Guo et al., 2025).

Restoration and intermediate supervision. Under such obfuscation, training with only final category labels forces the classifier to implicitly locate evasive spans, infer canonical risk expressions, and associate the restored meaning with the risk label, making it difficult to maintain robustness as camouflage patterns evolve. One natural defense is to normalize or restore the input before prediction (Bitton et al., 2022). However, this two-stage pipeline introduces an additional inference-time step, which may increase latency and propagate restoration errors. To avoid this extra inference-time step, a classifier can instead internalize obfuscation-handling capability through richer training supervision, following the broader paradigms of knowledge distillation (Hinton et al., 2015), learning with privileged information (Vapnik and Vashist, 2009), annotator rationales (Zaidan et al., 2007), and reasoning-step distillation (Hsieh et al., 2023; Feng et al., 2024). ReCAST follows this intermediate supervision view, but replaces open-ended rationales with task-specific de-obfuscation supervision: obfuscated spans, obfuscation types, and restored text. Because these signals are used only during training, deployment remains single-pass: the model directly maps raw SMS messages to risk labels while benefiting from the de-obfuscation capability learned during training.

3 Methodology

3.1 Task Formulation

Given an input Chinese SMS message x, the goal is to predict its risk label $y \in \mathcal{Y}$, where $\mathcal{Y}$ includes fraud, gambling, pornography, and benign. In real-world settings, x may contain obfuscated expressions that hide risk-bearing phrases while preserving human readability.

Besides the final label y, ReCAST introduces restoration-oriented intermediate supervision during training. For an obfuscated message, we define a set of obfuscated spans $S = \{s_i\}_{i=1}^{m}$, $T = \{t_i\}_{i=1}^{m}$, where t_i denotes the obfuscation type associated with span s_i, and the restored text r. These annotations describe where the message is obfuscated, how it is obfuscated, and what the normalized message should be. They are used only during training; at inference time, the model directly predicts y from the original input x.

3.2 Overview of ReCAST

We propose ReCAST, a Restoration-aware Cascaded Stage-wise Training framework for robust obfuscated SMS classification. ReCAST uses a large teacher model only during offline training, and distills its ability to interpret and restore obfuscated text into a smaller deployable student model for online deployment.

As shown in Figure 2, ReCAST consists of two cascaded stages. Stage 1 trains the student with restoration-oriented supervision, including obfuscated spans, obfuscation types, restored text, and risk labels. Stage 2 further adapts the restoration-aware student to the downstream SMS risk classification task. During online inference, ReCAST takes the original SMS as input and directly outputs the risk label, without invoking the teacher model or generating the full restoration output.

A key design of ReCAST is a unified task-conditioned prompt interface. All tasks share the same system instruction and differ only in a task tag: `<TASK=FULL>` outputs structured restoration supervision and the risk label; `<TASK=CLS>` outputs only the final risk category; and `<TASK=NORM>` outputs only the restored text for explicit restoration baselines and diagnostic comparison. This unified interface makes Stage 1 and Stage 2 prompt-compatible, and allows Stage 2 to mix `FULL`-format rehearsal examples with `CLS`-format classification examples without changing the model interface. The complete prompt template is provided in Appendix E.

This design uses restoration as a training-time supervision signal rather than an inference-time preprocessing step, allowing the deployable classifier to benefit from de-obfuscation ability while preserving the efficiency required by production SMS risk detection systems.

3.3 Stage 1: Restoration-oriented Distillation

The first stage aims to teach the student model to interpret obfuscated SMS before optimizing it for final classification. Given an obfuscated SMS x, we use the large teacher model to produce restoration-oriented supervision. As shown in Figure 2, this structured target is denoted as (S, T, r, y), where S represents obfuscated spans, T represents obfuscation types, r represents the restored SMS, and y represents the risk category.

The student model is trained under the unified prompt interface with the `<TASK=FULL>` tag to generate the full structured output from the original obfuscated SMS. Unlike direct classification, this stage provides intermediate evidence that links obfuscated surface forms to their normalized risk-bearing expressions and corresponding obfuscation types. Such dense supervision encourages the student to learn mappings between obfuscated forms, restored expressions, and risk semantics.

In our implementation, structured labels are obtained from a combination of human annotations and LLM-assisted generation, followed by automatic checking and manual correction for flagged cases. The student is optimized with the standard sequence generation loss over the structured target output. After this stage, the resulting restoration-aware student has internalized preliminary de-obfuscation ability and is used to initialize the second-stage classifier. Complete examples of the structured output format are provided in Figure 4 in Appendix E.

3.4 Stage 2: Classification-oriented Training

In the second stage, we further train the restoration-aware student for SMS risk classification. The model initialized from Stage 1 has learned to associate obfuscated surface forms with their normalized expressions and risk semantics. Stage 2 transfers this knowledge from structured restoration generation to the final classification objective, so that the model can make robust label predictions without explicitly producing restoration outputs.

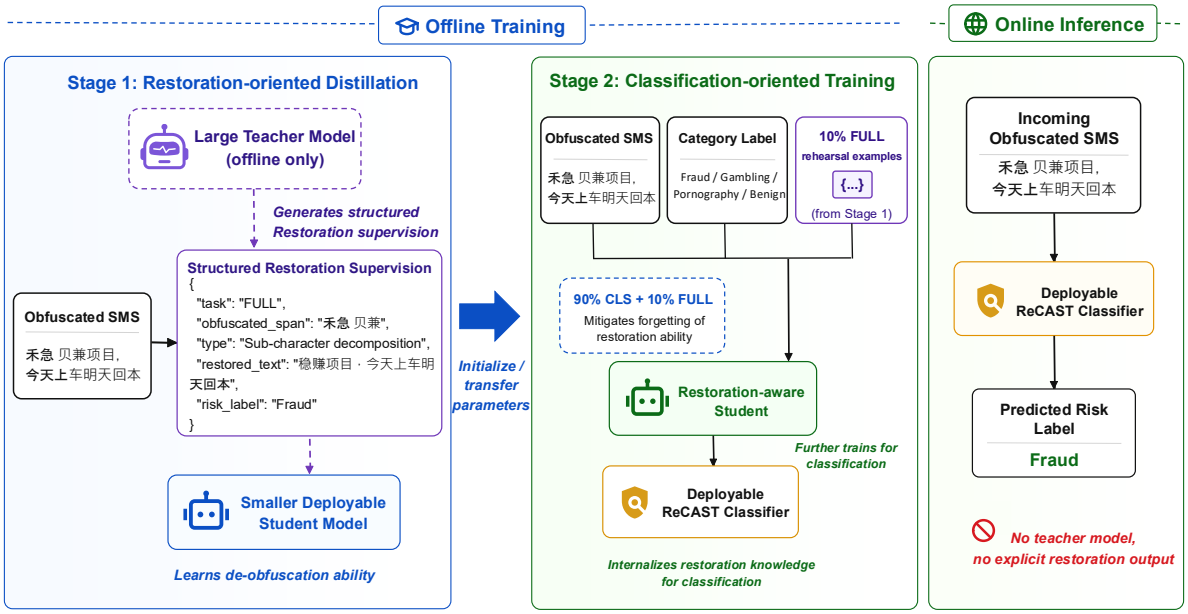

Figure 2: ReCAST distills restoration-oriented supervision from a large teacher into a smaller deployable student, then further trains it for robust SMS risk classification.

The input remains the original obfuscated SMS rather than the restored text, keeping the training setting consistent with online deployment, where ground-truth restorations are unavailable. We switch the task tag from <TASK=FULL> to <TASK=CLS> under the same unified prompt interface, and supervise the model to output only the final risk label y. This encourages the model to convert the explicit restoration behavior learned in Stage 1 into implicit decision evidence for classification, rather than relying on an external normalization step.

Because CLS and FULL examples share the same system prompt and differ only in the task tag and output schema, Stage 2 can naturally mix the two formats. Specifically, 90% of the Stage 2 instances use the <TASK=CLS> format, while the remaining 10% use the <TASK=FULL> format as rehearsal data. This lightweight replay strategy mitigates catastrophic forgetting of the restoration ability acquired in Stage 1, while allowing the training process to focus primarily on classification.

This design differs from an explicit restoration-then-classification pipeline. ReCAST does not first generate a restored SMS and then pass it to a separate classifier. Instead, restoration is used only as training-time supervision, and the resulting knowledge is internalized in the model parameters. After Stage 2, the model serves as the deployable ReCAST classifier and directly maps an incoming SMS to a risk category with the <TASK=CLS> tag.

3.5 Inference

During online inference, ReCAST uses only the deployable student classifier obtained after Stage 2. Given an incoming SMS, the model uses the same unified prompt interface with the <TASK=CLS> tag and directly predicts the risk label. It does not invoke the large teacher model, call a separate restoration module, or generate the full FULL-format output. This keeps inference as efficient as a single-pass cost-conscious classifier while still benefiting from the de-obfuscation knowledge distilled during training.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We use real-world Chinese SMS data for evaluation and a mixture of human-labeled and LLM-assisted data for training. The validation and test sets contain 500 and 1,000 production SMS messages, respectively, collected from different months to reduce temporal leakage. All evaluation samples are manually annotated with one of four labels: fraud, gambling, pornography, and benign. In these sets, non-benign messages include both obfuscated cases and a small number of unobfuscated cases. Benign messages do not contain intentional obfuscation, but may still include naturally occurring textual noise, such as typos, variant characters, or traditional Chinese forms, reflecting real-world

SMS traffic. To prevent data leakage, validation and test messages are strictly excluded from all LLM-assisted training data generation steps, including seed selection, prompting, reference examples, and hard-example mining. All messages are anonymized: Personally identifiable information, phone numbers, account identifiers, and platform-specific signatures are removed or replaced with placeholders. The test set is approximately balanced across fraud, gambling, pornography, and benign categories, with each category accounting for about one quarter of the samples.

The training set contains 30K category-balanced examples constructed through a human-in-the-loop LLM-assisted pipeline. A subset is manually annotated with category labels and restored texts. For the rest, DeepSeek-V4-Pro(DeepSeek-AI, 2026) generates diverse clean SMS messages from seed examples and creates obfuscated variants using a small set of human-labeled obfuscation examples, covering both single and mixed strategies with at most two variants per clean message. We also mine hard examples by manually annotating a subset of LLM-misclassified obfuscated messages. Clean–obfuscated pairs are then converted into structured restoration-training instances with restored text, obfuscated spans, obfuscation types, and category labels. All generated instances are LLM-checked, with flagged samples manually reviewed and corrected. Details are provided in Appendix A.

4.1.2 Baselines

We compare ReCAST with four baselines covering direct classification, data augmentation, explicit restoration pipelines, and prompt-based inference. Unless otherwise specified, all trainable baselines use the same backbone as ReCAST.

Direct-CLS trains the backbone on the 30K training set using the original SMS as input and the risk label as output, without any restoration-oriented supervision.

Aug-CLS tests whether the gains come merely from more obfuscated data. It extends Direct-CLS with 20K additional synthetic obfuscated SMS messages, yielding 50K training examples, but still uses only classification labels.

Pipeline-CLS evaluates an explicit restoration-then-classification pipeline. We train a separate restoration model using the ReCAST training data, but only keep the original obfuscated SMS and its restored text as supervision. The model is trained with a **<TASK=NORM>** tag to normalize obfuscated SMS into restored text. The restored output is then fed into Direct-CLS for risk classification. This baseline contrasts with ReCAST, which uses restoration as training-time intermediate supervision and internalizes the restoration ability into a single classifier.

Prompt-CLS employs Qwen3.5-9B(Qwen Team, 2026) without task-specific fine-tuning, using only a classification prompt. It serves as a zero-shot reference for the benefit of supervised distillation and fine-tuning.

Published baselines. To validate effectiveness beyond same-backbone controls, we additionally evaluate two published methods under the same SMS test protocol: **RoCBert** (Su et al., 2022), a robust Chinese encoder (RoCBert-base, $\approx$0.1B params) that incorporates semantic, phonetic, and visual information, fine-tuned for our four-way task; and an **adapted CA-CoT** (Yang et al., 2025), which explicitly identifies and restores linguistic perturbations before classification, using the same Qwen3.5-9B backbone as ReCAST while retaining CA-CoT's original three-stage reasoning procedure. We also report **DS-CLS**, direct inference of the DeepSeek-V4-Pro teacher with the classification prompt, as a non-deployable offline reference.

For Direct-CLS, Aug-CLS and Prompt-CLS, we use a direct classification prompt that asks the model to output only one risk category. The English translation of the prompt template is provided in Appendix E, Figure 5.

4.1.3 Evaluation Metrics

We evaluate both classification effectiveness and serving efficiency. For effectiveness, we report Accuracy (ACC) for overall four-way classification and Risk Recall (RR) for recall over all non-benign categories, i.e., the proportion of risky messages not predicted as benign. RR is particularly important in production SMS risk detection, where missed risky messages may cause downstream harm. For efficiency, we report Average Latency (Avg Lat.), P95 Latency (P95 Lat.), and Test Time (TT) under the same vLLM(Kwon et al., 2023) serving configuration. Avg Lat. and P95 Lat. measure the mean and 95th-percentile end-to-end request latency, respectively, while TT measures the total wall-clock time for processing the 1,000-message test set. All efficiency metrics are averaged over multiple runs with identical hardware, concurrency, batch-size, decoding, and prefix-caching settings.

Method	ACC (%)	RR (%)	Avg (s)	P95 (s)	TT (s)
Prompt-CLS	62.8	69.5	0.65	1.15	2.94
Direct-CLS	75.8 ± 0.30	79.2 ± 0.36	0.67	1.18	2.98
Aug-CLS	80.6 ± 0.28	85.1 ± 0.31	0.68	1.17	3.00
Pipeline-CLS	83.4 ± 0.53	86.2 ± 0.47	1.42	2.52	6.66
ReCAST	$\mathbf{86.6 \pm 0.17}$	$\mathbf{89.5 \pm 0.15}$	0.71	1.20	3.03

Table 1: Main results on internal real-world Chinese SMS test set. Trainable methods report mean $\pm$ std over 4 random seeds; Prompt-CLS uses deterministic constrained decoding and is a single value. Avg/P95/TT are average latency, P95 latency, and total test time under 256 concurrent requests with the same vLLM serving configuration and prefix caching.

Method	Backbone	Params	ACC (%)	RR (%)
RoCBert	RoCBert-base	$\approx$0.1B	79.1	82.7
Adapted CA-CoT	Qwen3.5-9B	9B	81.5	84.3
ReCAST	Qwen3.5-9B	9B	**86.6**	**89.5**
DS-CLS	DeepSeek-V4-Pro	1.6T	92.2	94.1

Table 2: Comparison with published methods and the teacher. RoCBert (Su et al., 2022) and CA-CoT (Yang et al., 2025) are evaluated under the same SMS test protocol; ReCAST outperforms both, including the same-backbone CA-CoT. ReCAST values are the 4-seed mean from Table 1 (std omitted for compactness). DS-CLS is direct teacher inference, a non-deployable offline reference; ReCAST recovers about 94% of its accuracy.

4.1.4 Implementation

We use Qwen3.5-9B as the deployable student backbone for ReCAST and all same-backbone trainable baselines in Table 1, which is substantially smaller than the teacher model and avoids online teacher inference. All methods are trained under comparable configurations. Classification outputs are generated with constrained label decoding over the predefined label set. Detailed training hyperparameters and serving configurations are provided in Appendix B.

4.2 Main Results

Table 1 reports the main results on the real-world obfuscated Chinese SMS test set. ReCAST achieves the best classification performance, with $86.6 \pm 0.17\%$ Accuracy and $89.5 \pm 0.15\%$ Risk Recall. All gaps below are computed on the 4-seed mean values. Compared with Direct-CLS, it improves Accuracy and Risk Recall by 10.8 and 10.3 points, respectively, showing that training only with final category labels is insufficient for robust classification under intentional obfuscation. Aug-CLS improves over Direct-CLS by adding synthetic obfuscated examples, but still lags behind ReCAST by 6.0 points in Accuracy and 4.4 points in Risk Recall. This indicates that ReCAST's gains are not merely due to exposure to more obfuscated data, but also come from restoration-aware intermediate supervision. Against the strongest same-backbone baseline, Pipeline-CLS, ReCAST improves mean ACC and RR by 3.2 and 3.3 points. Cross-seed variation is small for all trainable methods, the mean-based ranking is unchanged, and ReCAST attains the lowest standard deviation (0.17/0.15), confirming that these conclusions are robust to training randomness. Pipeline-CLS further confirms the value of restoration, but its explicit restoration step leads to much higher serving cost: 1.42s average latency, 2.52s P95 latency, and 6.66s total test time.

Beyond same-backbone controls, Table 2 compares ReCAST with two published methods—RoCBert (Su et al., 2022) and an adapted CA-CoT (Yang et al., 2025) that explicitly restores perturbations before classification on the Qwen3.5-9B backbone—and with DS-CLS, direct teacher inference. ReCAST outperforms the published methods, improving over the same-backbone CA-CoT by 5.1 ACC and 5.2 RR points, which supports internalizing restoration-oriented supervision rather than performing explicit restoration at inference time; it recovers about 94% of the teacher's accuracy (86.6 vs. 92.2) while using a substantially smaller 9B student instead of the 1.6T-parameter teacher.

In contrast, ReCAST preserves the efficiency of single-pass classifiers. Its average latency, P95 latency, and total test time are 0.71s, 1.20s, and 3.03s, respectively, close to Direct-CLS and Aug-CLS under the same vLLM serving configuration. These results show that ReCAST internalizes restoration ability during training and achieves a favorable robustness-efficiency trade-off without explicit restoration at inference time. Under our target serving configuration, the observed latency is compatible with our offline deployment-oriented evaluation target, supporting the practical feasibility of ReCAST for subsequent shadow deployment and online validation.

Additional benign false positive analysis in Appendix C further shows that ReCAST ties Pipeline-CLS for the lowest Benign FPR among the student methods reported in Table 4, indicating that the improvement of ReCAST does not come from simply over-predicting risky categories. Class-wise results and the confusion matrix (Appendix D) show that

Fraud is the most challenging category (F1 82.6%), with residual errors dominated by risky-to-benign misclassification rather than confusion among the risky categories.

4.3 Ablation Study

Variant	ACC↑	RR ↑
ReCAST full	**86.4**	**89.7**
Joint-Shuffled	84.1	86.8
w/o FULL replay in Stage 2	84.5	87.4
w/o CLS label in Stage 1	85.3	87.8
w/o Stage 2	82.9	85.6
- w/o span&type in Stage 1	73.8	72.1
- w/o restored text in Stage 1	78.6	80.4

Table 3: Ablation of ReCAST. The upper block evaluates variants with Stage 2 classification training. The lower block removes Stage 2 and uses the CLS label generated in Stage 1 for classification, where each indented variant further removes one Stage 1 restoration signal. span&type denotes joint supervision of obfuscated spans and their types. Ablations use a single representative checkpoint—the same one underlying the class-wise, restoration, and perturbation analyses—whose ACC/RR fall within 0.2pp of the 4-seed mean in Table 1. Because the smallest ablation drop (1.1pp ACC / 1.9pp RR) exceeds the largest cross-seed std in Table 1 (0.53pp for ACC, 0.47pp for RR) by over 2×, suggesting that the observed ablation effects are larger than the typical cross-seed variation.

Table 3 summarizes the contribution of each ReCAST component. Removing FULL-format replay in Stage 2 reduces Accuracy from 86.4% to 84.5%, suggesting that replaying a small amount of restoration-format data helps preserve the de-obfuscation ability learned in Stage 1. Removing the CLS label from Stage 1 also hurts performance, showing that task-aware label supervision helps align restoration with downstream risk semantics.

To isolate the effect of stage-wise supervision ordering, we further compare ReCAST with Joint-Shuffled, which uses the same initialization, total FULL/CLS exposure, and computational budget but randomly mixes the two formats throughout training. Joint-Shuffled achieves 84.1% ACC and 86.8% RR, 2.3 and 2.9 points below ReCAST, respectively. This result indicates that the gain is not explained solely by the amount of restoration and classification supervision; organizing the transition from structured restoration-oriented learning to classification-oriented adaptation also contributes to the final performance.

Without Stage 2, Accuracy drops to 82.9%, indicating that restoration-oriented training still needs classification adaptation. Further removing span&type supervision causes the largest drop, to 73.8% Accuracy and 72.1% Risk Recall, highlighting the importance of localized obfuscation supervision. Removing restored text supervision substantially degrades performance, confirming the value of recovering normalized risk-bearing expressions.

4.4 Restoration Analysis

Because restoration-oriented supervision is central to Stage 1, we directly evaluate whether the intermediate restoration capability is learned and retained after classification-oriented adaptation. We randomly sample 200 obfuscated risky messages from the test set and manually annotate their obfuscated spans, restored forms, and obfuscation types. Using the <TASK=FULL> interface, the checkpoint immediately after Stage 1 obtains 92.0% span-set exact-match accuracy and 3.5% normalized character-level edit distance. The final ReCAST checkpoint achieves 89.0% and 4.9%, respectively. Although Stage 2 introduces a moderate degradation in explicit restoration quality, the final model retains most of the restoration capability while optimizing for direct classification, providing direct evidence that the restoration knowledge learned in Stage 1 is not fully forgotten.

5 Conclusion

We presented ReCAST, a restoration-aware cascaded stage-wise training framework for robust obfuscated Chinese SMS classification. ReCAST distills de-obfuscation ability from a large teacher into a smaller deployable model via structured supervision over obfuscated spans, restored text, and task-aware labels, followed by classification fine-tuning. Experiments on real-world obfuscated Chinese SMS show that ReCAST outperforms direct classification, data augmentation, and explicit restoration pipelines, while preserving single-pass inference efficiency comparable to same-backbone classifiers. Ablations further show that localized obfuscation and restored-text supervision are key to the gains. Although evaluated on Chinese SMS, ReCAST is conceptually applicable to other harmful text detection settings involving adversarial obfuscation, with broader validation left for future work.

Limitations

This work has three main limitations. First, our experiments focus on Chinese SMS risk classification using a 9B-scale student backbone. Although ReCAST is not tied to a specific language, message channel, or model size, its effectiveness and efficiency on other languages, platforms, risk domains, and smaller deployable models remain to be systematically validated. Future work will explore whether restoration-aware supervision can provide similar robustness gains for more compact models under stricter latency and resource constraints.

Second, our training-data construction and structured supervision rely on a single teacher model, DeepSeek-V4-Pro. Although the test data are independently annotated by human experts and are excluded from all LLM-assisted training-data generation, the teacher may still introduce correlated biases in the types, styles, and coverage of synthetic obfuscation patterns. This may affect the representativeness of the resulting training corpus, particularly for naturally occurring, ambiguous, or evolving obfuscation patterns, so the robustness gains may not fully generalize, and the sensitivity of ReCAST to teacher choice remains an open question. As a sanity check, ReCAST achieves a benign false positive rate of 2.8% on the held-out production test set; however, this does not establish fairness across all legitimate sender types or linguistic varieties.

Third, our deployment-oriented evaluation is conducted offline under controlled settings. It has not yet been evaluated under full production traffic with evolving templates, sender behaviors, and system-level constraints. More comprehensive shadow deployment or online monitoring is needed before high-impact operational use.

Ethics Statement

This work aims to improve the robustness of SMS risk detection systems against obfuscated messages. Since SMS data may contain sensitive content, all data used in this study was processed under internal data governance procedures for security research and risk-control. Personally identifiable information, including phone numbers, account identifiers, verification codes, URLs, names, addresses, and sensitive fields, was removed or replaced with placeholders before training and evaluation. We do not release raw SMS messages or data that could be linked to individual users.

The teacher model used for training-data construction was deployed privately using the officially released DeepSeek-V4-Pro model weights rather than accessed through a third-party or public API. The released model weights and code are distributed under the MIT License. We use only the teacher's final structured outputs as offline supervision for training the student model and do not use intermediate reasoning traces as training targets.

We recognize that false positives may affect legitimate communication, such as platform notifications, verification messages, financial reminders, marketing messages, and ordinary personal messages. ReCAST has not been deployed as an autonomous production decision system. Rather, it is designed as an assistive risk-detection component. If deployed in practice, we recommend that its predictions be combined with rule-based safeguards, trusted sender signals, threshold control, manual review for high-impact cases, and appeal or correction mechanisms where appropriate.

This work also has potential dual-use risks because it analyzes obfuscation strategies used to evade SMS risk detectors. To mitigate misuse, we only present sanitized examples and avoid disclosing production rules, operational thresholds, complete evasion dictionaries, raw abusive messages, or deployment-sensitive prompts. The purpose of this work is defensive: improving anti-abuse systems against evolving real-world obfuscation.

For human annotation and review, the data were handled under existing internal security, risk-control, and data-governance procedures. Before research use and human annotation, personally identifiable information and other sensitive fields were removed or replaced. Annotation was conducted exclusively through an access-controlled internal system by authorized personnel bound by confidentiality agreements, who were provided with task guidelines and quality-control procedures and instructed to focus on risk categories and obfuscation patterns. Raw messages were not downloaded, exported, or transferred outside this controlled environment, within which DeepSeek-V4-Pro was also privately deployed. The restored text produced by our model should be treated as an intermediate analytical signal rather than factual evidence of user intent. If used in operational settings, the system would require periodic monitoring, evaluation, and updating, and should remain part of a broader human- and policy-governed anti-abuse pipeline.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund Project U24B20179.

References

Joanna Bitton, Maya Pavlova, and Ivan Evtimov. 2022. Adversarial text normalization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 268–279, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. Technical report.

Tao Feng, Yicheng Li, Chenglin Li, Hao Chen, Fei Yu, and Yin Zhang. 2024. Teaching small language models reasoning through counterfactual distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5831–5842, Miami, Florida, USA. Association for Computational Linguistics.

Haotan Guo, Jianfei He, Jiayuan Ma, Hongbin Na, Zimu Wang, Haiyang Zhang, Qi Chen, Wei Wang, Zijing Shi, Tao Shen, and Ling Chen. 2025. Lost in pronunciation: Detecting chinese offensive language disguised by phonetic cloaking replacement. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2538–2550, Suzhou, China. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Shaghayegh Hosseinpour and Sanchari Das. 2025. Poster: A multi-signal model for detecting evasive smishing. *arXiv preprint arXiv:2505.18233*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Zhuoren Jiang, Zhe Gao, Yu Duan, Yangyang Kang, Changlong Sun, Qiong Zhang, and Xiaozhong Liu. 2020. Camouflaged chinese spam content detection with semi-supervised generative active learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3080–3085, Online. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Kaiting Lai, Yinong Long, Bowen Wu, Ying Li, and Baoxun Wang. 2022. Semorph: A morphology semantic enhanced pre-trained model for chinese spam text detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1003–1013. ACM.

Xiaoxu Liu, Haoye Lu, and Amiya Nayak. 2021. A spam transformer model for sms spam detection. *IEEE Access*, 9:80253–80263.

C. Oswald, Sona Elza Simon, and Arnab Bhattacharya. 2022. Spotspam: Intention analysis–driven sms spam detection using bert embeddings. *ACM Transactions on the Web*, 16(3):1–27.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Muhammad Salman, Muhammad Ikram, Nardine Basta, and Mohamed Ali Kaafar. 2025. Spallm-guard: Pairing sms spam detection using open-source and commercial llms. *arXiv preprint arXiv:2501.04985*.

Hui Su, Weiwei Shi, Xiaoyu Shen, Zhou Xiao, Tuo Ji, Jiarui Fang, and Jie Zhou. 2022. RoCBert: Robust Chinese bert with multimodal contrastive pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–931, Dublin, Ireland. Association for Computational Linguistics.

Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557.

Junqi Wu, Shujie Ji, Kang Zhong, Huiling Peng, Zhendongxiao, Xiongding Liu, and Wu Wei. 2025. Enhancing chinese offensive language detection with homophonic perturbation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22660–22675, Suzhou, China. Association for Computational Linguistics.

Yunze Xiao, Yujia Hu, Kenny Tsu Wei Choo, and Roy Ka-Wei Lee. 2024. Toxicloakcn: Evaluating robustness of offensive language detection in chinese with cloaking perturbations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6012–6025, Miami, Florida, USA. Association for Computational Linguistics.

Shujian Yang, Shiyao Cui, Chuanrui Hu, Haicheng Wang, Tianwei Zhang, Minlie Huang, Jialiang Lu, and Han Qiu. 2025. Exploring multimodal challenges in toxic Chinese detection: Taxonomy, benchmark, and findings. In *Findings of the Association*

for Computational Linguistics: ACL 2025, pages 14382–14396, Vienna, Austria. Association for Computational Linguistics.

Jinliang Yao, Chenrui Wang, Chuang Hu, and Xiaoxi Huang. 2022. Chinese spam detection using a hybrid bigru-cnn network with joint textual and phonetic embedding. *Electronics*, 11(15):2418.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267. Association for Computational Linguistics.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. 2024. A survey on efficient inference for Large Language Models. *Preprint*, arXiv:2404.14294.

A Dataset Details

A.1 Evaluation Data

The validation and test sets are sampled from real production SMS data and manually annotated by human experts. The validation set contains 500 messages, while the test set contains 1,000 messages. The two sets are collected from different months to reduce temporal leakage and to better evaluate robustness under temporal distribution shift. Each message is assigned one of four category labels: fraud, gambling, pornography and benign. The evaluation sets are never generated or rewritten by the LLM.

A.2 Training Data Construction

The final training set contains 30K examples with approximately balanced category distributions. It is built from both human-labeled data and LLM-assisted synthetic data. A subset of the training data is manually annotated with risk categories and restored texts. To scale up the training data, we use DeepSeek-V4-Pro in a multi-step generation pipeline.

First, we provide a set of seed examples to the LLM and ask it to generate clean SMS messages with similar topics. We explicitly require the generated messages to be category-balanced, diverse in topic, sentence pattern, and length, and free from near duplicates. These clean messages serve as the base texts for later obfuscation.

Second, we provide a small number of human-labeled obfuscation examples and ask the LLM to create obfuscated variants from the clean messages. The generated variants cover all single obfuscation strategies in our taxonomy, including homophone substitution, visually similar characters, simplified–traditional variants, digit homophones, whitespace insertion, symbol insertion, URL splitting, prefix/suffix noise, abbreviation, slang, and sub-character decomposition. We also generate mixed obfuscations by combining multiple strategies. To avoid over-representing a single clean message, each clean message is used to generate at most two obfuscated variants.

Third, we perform LLM-assisted hard example mining. We ask DeepSeek-V4-Pro to directly classify a subset of obfuscated messages and select a subset of the incorrectly predicted samples for manual annotation. Human annotators then provide corrected labels and restored texts for these hard cases. This step enriches the training set with examples that are likely to confuse direct classification models.

Fourth, we prompt the LLM to convert clean–obfuscated pairs into structured restoration-training instances. Each instance contains the obfuscated SMS, restored text, risk category, obfuscated spans, and corresponding obfuscation types in JSON format. These structured labels are used to supervise the restoration stage of ReCAST.

A.3 Quality Control

We apply both automatic and manual quality control to the generated training data. The LLM is first used to check whether each generated instance is internally consistent, including whether the restored text matches the obfuscated text, whether the labeled span corresponds to an actual obfuscated fragment, whether the obfuscation type is appropriate, and whether the category label is plausible. Instances flagged as potentially problematic are manually reviewed and corrected before being added to the final training set.

We also control data diversity during generation by requiring diverse sentence patterns, lengths, and topics, and by limiting the number of obfuscated variants generated from each clean message. This reduces duplication and prevents the model from overfitting to a small set of templates. The resulting training set provides large-scale structured supervision for restoration-aware distillation, while all validation and test results are reported on real, manually annotated production SMS data.

B Implementation Details

Model and Hardware. We adopt Qwen3.5-9B as our backbone model. Full-parameter fine-tuning is conducted using the SWIFT framework on eight NVIDIA H100 (80GB) GPUs in BF16 precision.

Hyperparameters. Across all stages, we employ the AdamW optimizer with a learning rate of 1×10^{-5}, a weight decay of 0.1, and a linear learning rate scheduler with a 0.05 warmup ratio. The training process spans 5 epochs with a maximum sequence length of 4096 tokens. With a per-device batch size of 2 and 16 gradient accumulation steps, the effective batch size scales to 256.

Inference Serving Setup. We deploy the model using the vLLM serving engine on a single NVIDIA H100 GPU (tensor parallelism degree of 1). Serving parameters are configured with a GPU memory utilization of 0.93 and a maximum of 256

concurrent sequences, with prefix caching enabled to enhance serving throughput. The maximum generation length is capped at 256 tokens, sufficient for the outputs observed in our experiments. For classification evaluation, we use constrained label decoding, where the output is restricted to the predefined risk-label set, ensuring deterministic and valid category predictions.

Teacher model and generation settings. We use a privately deployed DeepSeek-V4-Pro, a Mixture-of-Experts (MoE) model with 1.6T total parameters, as the offline teacher for synthetic data generation, hard-example mining, and restoration-oriented annotation. The teacher is substantially larger than the 9B student backbone and is used only during training-data construction. During teacher-side generation, we enable reasoning mode to improve annotation consistency for difficult obfuscation cases. We retain only the final structured outputs, including restored text, obfuscated spans, obfuscation types, and risk labels; intermediate reasoning traces are discarded and are not used as student supervision. We additionally evaluate direct teacher inference as an offline reference, but do not consider it a deployable baseline due to substantially higher serving cost. The results are shown in Table 4.

C Benign False Positive Analysis

In production SMS risk detection, improving recall on risky messages should not come at the cost of excessive false positives on benign traffic. We therefore report the benign false positive rate (Benign FPR), defined as the proportion of benign messages incorrectly predicted as fraud, gambling, or pornography. This metric complements Risk Recall by measuring whether a model tends to over-predict risky categories on normal SMS messages.

As shown in Table 4, ReCAST achieves the best overall classification performance while maintaining the lowest benign false positive rate. Compared with Prompt-CLS, ReCAST substantially reduces Benign FPR from 16.0% to 2.8%, indicating that supervised restoration-aware training greatly improves robustness to benign but risk-looking surface patterns. ReCAST also achieves a lower Benign FPR than Direct-CLS and Aug-CLS, and matches Pipeline-CLS, while substantially outperforming Pipeline-CLS in ACC and Risk Recall. This suggests that ReCAST does not improve Risk Recall by simply over-predicting risky categories.

Method	ACC↑	RR↑	Benign FPR↓
DS-CLS	92.2	94.1	2.4
Prompt-CLS	62.8	69.5	16.0
Direct-CLS	75.7	79.6	6.8
Aug-CLS	80.3	85.5	6.4
Pipeline-CLS	82.9	86.2	2.8
ReCAST	86.4	89.7	2.8

Table 4: Benign false positive analysis on the real-world Chinese SMS test set (representative ReCAST checkpoint). The benign subset contains 250 messages. Benign FPR measures the proportion of benign messages incorrectly classified as fraud, gambling, or pornography. DS-CLS directly applies DeepSeek-V4-Pro with the direct classification prompt and is included only as a non-deployable offline teacher reference.

Actual \ Pred.	Benign	Gambling	Porn.	Fraud
Benign	243	2	1	4
Gambling	22	201	5	15
Pornography	17	5	220	7
Fraud	38	12	8	200

Table 5: Confusion matrix of the ReCAST checkpoint on the 1,000-message test set (rows: actual, columns: predicted).

Instead, restoration-aware supervision helps the model better distinguish intentional obfuscation in risky messages from naturally occurring noise in benign SMS.

We further manually inspect the benign false positives made by ReCAST. Most of the remaining benign errors are normal notifications related to banking, financial, or securities services. These messages often contain URLs, account-operation terms, amounts, or stock-code-like strings, which can resemble risk-bearing cues in fraudulent SMS. This indicates that the residual false positives are concentrated in high-risk-looking but legitimate transactional scenarios. In practical deployment, trusted-sender signals, domain or sender whitelists, and lightweight business rules can be combined with ReCAST to further reduce such false positives.

D Class-wise Analysis

Table 5 reports the confusion matrix of the ReCAST checkpoint on the 1,000-message test set, and Table 6 reports the corresponding per-class precision, recall, and F1.

Pornography obtains the highest F1 (91.1%), whereas Fraud is the most challenging category (F1 82.6%): 38 of 258 fraud messages are predicted as Benign, so the dominant residual error is

Category	Precision (%)	Recall (%)	F1 (%)
Benign	75.9	97.2	85.3
Gambling	91.4	82.7	86.8
Pornography	94.0	88.4	91.1
Fraud	88.5	77.5	82.6

Table 6: Class-wise precision, recall, and F1 of the ReCAST checkpoint. Pornography obtains the highest F1, while Fraud is the most challenging category.

risky-to-benign misclassification rather than confusion among the three risky categories. For Benign messages, ReCAST achieves 97.2% recall, with only seven benign messages incorrectly predicted as risky; manual inspection shows that most of these false positives are legitimate banking, financial, or securities notifications whose URLs, account-operation terms, monetary amounts, or stock-code-like strings resemble risk-bearing cues in fraudulent SMS.

E Unified Multi-task Prompt Template

You are an anti-fraud SMS obfuscation restoration and risk classification model.

You should perform the corresponding task according to the `TASK` type specified in the input.

The model supports three TASK types:
1. `<TASK=FULL>`
   Output the obfuscated spans, the restored text, and the risk category.
   The output must be a strict JSON object in the following format:
   `{"sp":[["original_span","restored_span","type"]],"nt":"restored_text","y":"risk_category"}`

2. `<TASK=NORM>`
   Output only the restored text. Do not output JSON or the risk category.

3. `<TASK=CLS>`
   Output only the risk category. Do not output JSON or the restored text.

The risk category `y` must be one of the following four labels:
* gambling-related risk
* fraud-related risk
* pornography-related risk
* benign

The obfuscation type must be selected from the following set:
* none
* character decomposition
* homophonic substitution
* visually similar substitution
* traditional / simplified Chinese variant
* digit-letter-symbol mixing
* symbol insertion
* whitespace insertion
* URL splitting
* noisy prefix or suffix
* abbreviation
* slang or coded expression
* multiple obfuscation types
* unknown

Task rules:
1. For `<TASK=FULL>`, first identify the obfuscated spans, then generate the restored text, and finally determine the risk category.
2. Each element in `sp` must be a 3-tuple: `["original_span","restored_span","type"]`.
3. The `original_span` in `sp` must come directly from the input SMS. Do not use the restored expression as the original span.
4. If there is no obfuscation, set `sp=[]`.
5. `nt` is the conservatively restored text derived from the original SMS and `sp`.
6. Restore only high-confidence obfuscations. If uncertain, keep the original text unchanged.
7. Do not hallucinate or add any amount, phone number, URL, platform, organization, group number, or business action that does not appear in the original SMS.
8. A benign SMS should not be classified as risky solely because it contains a verification code, URL, amount, group number, or pickup code.
9. If the text contains clear gambling, fraud, or pornography-related semantics, choose the corresponding risk category; otherwise, label it as `benign`.
10. For `<TASK=FULL>`, output only JSON. Do not output explanations, Markdown, code blocks, or `<think>`.
11. For `<TASK=NORM>`, output exactly one line of restored text.
12. For `<TASK=CLS>`, output exactly one category label.

Definitions of obfuscation types:
……

User Prompt for Stage 1
`<TASK=FULL>

Short Message: {{raw_text}}`

User Prompt for Stage 2
`<TASK=CLS>

Short Message: {{raw_text}}`

Figure 3: English translation of the unified multi-task prompt template. Provided solely for readability; the original Chinese prompts are used for training and evaluation.

Figure 4: Few-Shot examples.

You are an anti-fraud risk classification model.

The risk category must be one of the following four labels:
* gambling-related risk
* fraud-related risk
* pornography-related risk
* benign

Task rules:
1. A benign SMS should not be classified as risky solely because it contains a verification code, URL, amount, group number, or pickup code.
2. If the text contains clear gambling, fraud, or pornography-related semantics, choose the corresponding risk category; otherwise, label it as `benign`.
3. Output exactly one category label.

Figure 5: This is an English translation of the direct classification prompt template for the Prompt-CLS, Direct-CLS, and Aug-CLS baselines; it is provided for readability only, while the original Chinese prompt is used in all experiments.

F Obfuscation Type List

Table 7: Taxonomy of obfuscation types used for structured de-obfuscation supervision.

Type	English Name	Chinese Explanation	English Explanation	Examples
拆字	Character Decomposition	把一个字或词拆成多个部件、偏旁或近似结构表达，使模型难以直接匹配原词。	Splitting a Chinese character or word into components, radicals, or visually related parts to obscure the original expression.	“禾急贝兼” → “稳赚”
同音	Homophonic Substitution	用读音相同或相近的字、词、拼音或谐音表达替换原始风险词。	Replacing the original expression with characters, words, or phonetic forms that have the same or similar pronunciation.	“薇信” → “微信”; “稳砖” → “稳赚”
形近	Glyph Substitution	用字形相近、视觉上容易混淆的字符替换原字符。	Replacing characters with visually similar characters to preserve human readability while disrupting model recognition.	“很行” → “银行”; “已冻结” → “已冻结”
繁简	Traditional-Simplified Mixing	使用繁体、异体字、简繁混用或其他字符变体替换规范简体表达。	Mixing simplified, traditional, variant, or non-standard Chinese characters to disguise the canonical form.	“賺錢” → “赚钱”; “發財” → “发财”
数混	Alphanumeric & Symbol Mixing	使用数字、字母、符号与汉字混合表达, 或用形似字符替代原字符。	Mixing digits, letters, symbols, and Chinese characters, or replacing characters with visually similar alphanumeric symbols.	“V信” → “微信”; “1oo%” → “100%”; “O” → “0”
符号	Symbol Insertion	在词语内部或关键片段之间插入无意义符号, 破坏连续匹配。	Inserting irrelevant symbols within words or key phrases to break lexical matching while keeping the text understandable.	“加-微-信” → “加微信”; “赚￥钱” → “赚钱”
空格	Whitespace Insertion	在词语内部或关键片段之间插入空格、制表符或异常间隔。	Inserting spaces, tabs, or abnormal gaps within words or key phrases to disrupt tokenization and matching.	“验证码” → “验证码”; “加微信” → “加微信”
URL拆分	URL Splitting	对网址、域名、短链或联系方式进行空格、符号、换行等拆分。	Splitting URLs, domains, short links, or contact strings with spaces, symbols, or line breaks to avoid URL-based detection.	“www . abc . com” → “www.abc.com”; “t . cn / xx” → “t.cn/xx”

Continued on next page...

Table 7 – Continued from previous page

Type	English Name	Chinese Explanation	English Explanation	Examples
前后缀	Noise Prefix/Suffix	在正文前后插入无意义字符、乱码、表情、符号或干扰短语。	Adding meaningless characters, random symbols, emojis, or distracting phrases before or after the main message.	"@@@稳赚项目###" → "稳赚项目"
缩写	Abbreviation	使用拼音首字母、英文缩写、简称、符号化表达或平台黑称表示原词。	Using initials, pinyin abbreviations, shorthand, symbolic forms, or aliases to refer to the original expression.	"+V" → "加微信"; "VX" → "微信"
黑话	Coded Jargon	使用行业黑话、隐晦说法、暗语或上下文依赖表达来隐藏真实意图。	Using domain-specific slang, euphemisms, coded expressions, or context-dependent phrases to conceal the real intent.	"上车" → "参与项目/入局"
多重	Compound Obfuscation	同一片段同时包含两种或多种伪装方式,需要联合判断才能还原。	Applying multiple obfuscation strategies to the same span, requiring combined interpretation for restoration.	"加-薇-信V" → "加微信"
未知	Unknown Obfuscation	存在明显伪装痕迹,但无法可靠归入以上具体类型,或还原结果不确定。	Obfuscation is present, but its type cannot be reliably assigned to the predefined categories, or the restoration is uncertain.	火星文/异常混写→还原结果不确定