COAT: An Orthogonal Benchmark for Diagnosing Chinese Adversarial Toxicity Detection Failures

Jiaqi Hu[1,†], **Mingshuai Liu**[1,†], **Haobo Wu**[1], **Xin Li**[1,2,*]
[1]School of Information and Cyber Security
[2]Public Security Big Data Strategy Research Center
People's Public Security University of China, Beijing, China
{202121430024,liumingshuai}@stu.ppsuc.edu.cn
1328004773@qq.com, lixin@ppsuc.edu.cn

Abstract

Character-level perturbations are widely used to evade Chinese toxicity detection, yet existing benchmarks either cover only a limited range of perturbations or expand perturbation coverage without balanced construction across toxic and non-toxic samples, leaving these two requirements unmet within a single benchmark. We introduce **COAT** (**C**hinese **O**rthogonal **A**dversarial **T**oxicity), a 41K-sample benchmark spanning eight perturbation types that enforces perturbation orthogonality: all perturbations are applied with equal probability to both toxic and non-toxic texts, ensuring $P(\text{toxic} \mid \text{perturbed}) = 0.5$. As a core diagnostic layer, we annotate Natural Language Variant Ambiguity (NLVA): 21.6% of clean samples naturally contain surface features overlapping with perturbation artifacts, enabling controlled analysis of model sensitivity to perturbation-like features and classification difficulty. Evaluating five encoder classifiers, three proprietary LLMs, and Llama3-8B with chain-of-thought reasoning, we find that (1) models remain systematically sensitive to perturbation-like features even under balanced training, and (2) across Llama3-8B variants, overcorrection varies systematically with the linguistic ambiguity introduced by each perturbation type. Step-level diagnosis localizes the shared bottleneck to semantic restoration, with targeted preference optimization yielding reliable gains on the most ambiguous types.

1 Introduction

Online platforms face a persistent arms race against adversarial toxicity evasion. Users increasingly deploy character-level perturbations such as homophone substitution, pinyin encoding, emoji insertion, and visual similarity replacement to bypass automated content moderation (Xiao et al., 2024).

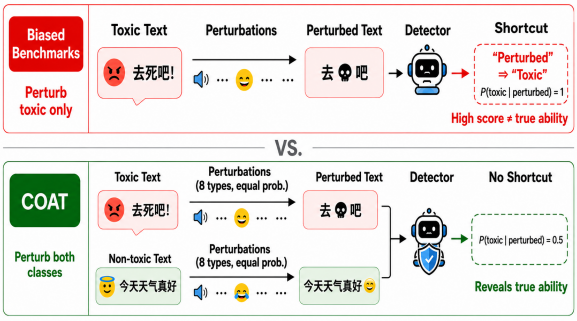


Figure 1: Evaluation bias vs. COAT's orthogonal design. COAT removes the shortcut $P(\text{toxic} \mid \text{perturbed}) = 1$ by perturbing both toxic and non-toxic texts.

These strategies are particularly effective in Chinese, where the logographic writing system offers a rich space of glyph-based, phonetic, and structural manipulations. Reliable evaluation benchmarks are therefore essential: without them, progress in adversarial toxicity detection cannot be meaningfully measured.

However, existing Chinese adversarial toxicity benchmarks suffer from structural flaws. ToxiCloakCN (Xiao et al., 2024) covers only two perturbation types; Yang et al. (2025) extend coverage to eight perturbation types but apply perturbations exclusively to toxic texts, creating $P(\text{toxic} \mid \text{perturbed}) = 1$. Under this design, a model can achieve high accuracy by learning the trivial shortcut "perturbed ⇒ toxic" without semantic discrimination (Geirhos et al., 2020). Benchmark scores obtained under such conditions conflate detection capability with shortcut exploitation, making it impossible to assess whether a model truly understands toxicity or merely recognizes perturbation artifacts.

We introduce **COAT** (**C**hinese **O**rthogonal **A**dversarial **T**oxicity), a 41K-sample benchmark applying eight perturbation types to both toxic and non-toxic texts with equal probability, ensuring $P(\text{toxic} \mid \text{perturbed}) = 0.5$. This eliminates the

[†]These authors contributed equally to this work.
[*]Corresponding author.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3453–3465
November 6-10, 2026 ©2026 Association for Computational Linguistics

shortcut confound and enables diagnosis of overcorrection (incorrectly flagging perturbed non-toxic texts). We additionally annotate Natural Language Variant Ambiguity (NLVA): 21.6% of clean texts naturally contain perturbation-like surface features, serving as natural controls for measuring model sensitivity to perturbation-like surface features.

Evaluating five encoder classifiers, three proprietary LLMs (GPT-5.4, DeepSeek-V3, Claude Sonnet 4.6), and Llama3-8B with CoT fine-tuning and step-level DPO, we find two systematic failure patterns. First, even under our balanced design, models retain a residual tendency to associate perturbation features with toxicity. Second, in the Llama3-8B per-type analyses, overcorrection varies systematically across perturbation types, with higher-ambiguity perturbations generally associated with higher FPR. Step-level DPO localizes the bottleneck to semantic restoration: only the restoration step yields reliable improvement, and this gain concentrates on high-ambiguity types.

Our contributions are:

- **COAT**, to our knowledge the first Chinese adversarial toxicity benchmark combining 8 perturbation types with strict orthogonality ($P(\text{toxic} \mid \text{perturbed}) = 0.5$), totaling 41K samples with NLVA annotation, constituting an integrated diagnostic framework.
- We reveal persistent model sensitivity to perturbation-like features even under balanced training, with NLVA-based controls showing that this pattern extends beyond perturbation-induced information loss.
- We identify a systematic link between NLVA and overcorrection, enabling fine-grained separation of perturbation-processing and ambiguity-related errors.
- A step-level diagnostic framework localizing the shared bottleneck to semantic restoration, with targeted diagnostic gains concentrated on high-ambiguity types.

2 Related Work

Automated detection of hate speech remains actively studied (Davidson et al., 2017; Hee et al., 2024). For Chinese, COLD (Deng et al., 2022), ToxiCN (Lu et al., 2023), and STATE ToxiCN (Bai et al., 2025) provide increasingly fine-grained benchmarks. Implicit hate speech (ElSherief et al., 2021; Lee et al., 2025) shares a conceptual link with NLVA, as surface features alone are insufficient for classification.

	ToxiCloakCN	CNTP	Ours
# Perturb types	2	8	**8**
Perturb target	All	Toxic only	**All**
Pert $\perp$ Toxicity	✓	✗	✓
Train/Test split	✗	✓	✓
Shortcut analysis	✗	✗	✓
NLVA annotation	✗	✗	✓
Total samples	∼18K	∼24K	**41K**

Table 1: Comparison with existing Chinese adversarial toxicity benchmarks. ToxiCloakCN (Xiao et al., 2024) achieves orthogonality but covers only two perturbation types, limiting diagnostic breadth; CNTP (Yang et al., 2025) extends to eight types but perturbs only toxic texts, enabling shortcut learning. COAT uniquely combines broad perturbation coverage, orthogonal design, and NLVA annotation for fine-grained error attribution.

Character-level attacks (Ebrahimi et al., 2018; Wallace et al., 2019), adversarial NLU benchmarks (Wang et al., 2021), and adversarial test suites for hate speech (Hartvigsen et al., 2022; Kirk et al., 2022; Hoeken et al., 2024) have exposed model brittleness (Zhang et al., 2024), while cross-lingual retrieval methods offer data-efficient alternatives (Ghorbanpour et al., 2025). For Chinese, ToxiCloakCN (Xiao et al., 2024) evaluates two perturbation types under a balanced toxic/non-toxic design, whereas CNTP (Yang et al., 2025) expands coverage to eight perturbation types but applies them only to toxic texts. Table 1 summarizes this gap: existing resources do not combine broad perturbation coverage, orthogonal construction, and fine-grained error attribution.

LLMs have been deployed as safety classifiers (Inan et al., 2023; Zeng et al., 2024) and toxic prompt detectors (Liu et al., 2024), with CoT reasoning (Liu et al., 2025) and pragmatic inference chains for implicit toxicity (Chen and Wang, 2025), and DPO (Rafailov et al., 2023) for preference-based training. Despite these advances, overcorrection has received limited systematic study. Behavioral testing frameworks (CheckList (Ribeiro et al., 2020), HateCheck (Röttger et al., 2021)) provide diagnostic probes but do not decompose error sources specific to adversarial toxicity. Our work addresses this gap.

3 COAT: Benchmark Construction

Figure 2 summarizes COAT's overall diagnostic workflow.

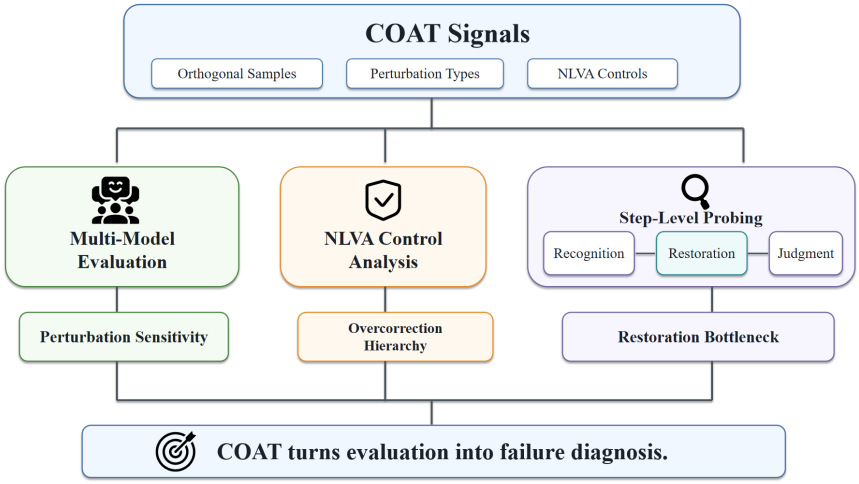


Figure 2: Overview of COAT's diagnostic workflow. Orthogonal samples, perturbation types, and NLVA controls turn benchmark evaluation into failure diagnosis.

3.1 Motivation and Design Principles

Existing Chinese benchmarks exhibit complementary limitations: ToxiCloakCN (Xiao et al., 2024) adopts a balanced toxic/non-toxic design but covers only two perturbation types, whereas CNTP (Yang et al., 2025) extends coverage to eight types but applies perturbations only to toxic texts. To combine broad perturbation coverage with balanced benchmark construction, we design COAT around the following principle:

> **Perturbation Orthogonality**: Perturbation is applied to both toxic and non-toxic texts with equal probability, ensuring $P(\text{toxic} \mid \text{perturbed}) = P(\text{toxic}) = 0.5$.

This eliminates the shortcut confound so that benchmark scores reflect content-based detection capability. Crucially, it also enables the diagnosis of *overcorrection*: models incorrectly flagging perturbed non-toxic texts as harmful. Because toxic and non-toxic texts receive the same perturbations, elevated FPR on perturbed non-toxic samples is no longer confounded by the label–perturbation imbalance targeted by conventional benchmarks, providing a cleaner signal of model sensitivity to perturbation-like features.

3.2 Perturbation Types

We apply eight character-level perturbation methods to Chinese text, covering the major adversarial strategies observed in real-world toxic content evasion (Xiao et al., 2024). Table 2 summarizes these methods.

Type	Category	Description
Emoji	Insertion	Insert emojis between characters
Homo	Phonetic	Replace with homophone characters
PY_Init	Phonetic	Replace with pinyin initials
PY_Full	Phonetic	Replace with full pinyin
Trad	Glyph	Convert to traditional characters
Shuff	Structural	Shuffle character order in words
VSim	Glyph	Replace with visually similar chars
Split	Structural	Decompose characters into radicals

Table 2: Eight perturbation types in COAT, categorized by manipulation strategy.

All eight types are applied to both toxic and non-toxic texts. Each perturbation operates at the character level with controlled intensity to maintain readability while introducing sufficient adversarial noise. For example, under emoji insertion decorative emojis appear between characters; py_init reduces each character to its pinyin initial (e.g., a four-character phrase becomes a four-letter abbreviation); vsim substitutes characters with visually similar glyphs preserving the overall outline. Detailed algorithms are provided in Appendix A.

These eight types span three major manipulation axes—phonetic (homo, py_init, py_full), glyph-based (trad, vsim), and structural (emoji, shuff, split)—covering the principal evasion strategies documented in Chinese online platforms (Xiao et al., 2024; Yang et al., 2025).

3.3 NLVA Annotation

Certain clean (unperturbed) texts naturally contain surface features that overlap with perturbation artifacts. For example, naturally occurring emojis are indistinguishable from emoji-insertion perturbation, and internet abbreviations overlap with pinyin-initial encoding. We refer to this overlap between naturally occurring surface forms and perturbation artifacts as Natural Language Variant Ambiguity (NLVA), a text-level property annotated independently of model behavior. Such overlap provides a natural diagnostic setting for examining model sensitivity to perturbation-like features.

Two trained annotators (graduate students in computational linguistics with prior hate speech annotation experience) independently label each clean sample for NLVA presence across four feature categories: emojis, Latin/pinyin characters, mixed-script patterns, and internet abbreviations. A third expert (senior researcher) adjudicates disagreements. Inter-annotator agreement reaches $\kappa = 0.83$. The resulting 21.6% NLVA-positive samples serve as natural controls for measuring model sensitivity to perturbation-like surface features.

3.4 Dataset Statistics

We source 4,586 base texts from ToxiCN (Lu et al., 2023), balanced between toxic (2,293) and non-toxic (2,293) samples. Each text undergoes all eight perturbation types plus a clean (unperturbed) condition, yielding $4{,}586 \times 9 = 41{,}274$ total samples.

The non-toxic subset is drawn from the same online forum sources to ensure domain consistency. Average text length is 28.3 characters (toxic: 24.7, non-toxic: 31.9).

We partition by base text ID to prevent cross-split leakage: 1,000 base texts for training (9K samples), 200 for development (1.8K), and 3,386 for testing (30.5K). The large test set ensures statistical reliability across all perturbation types.

Quality Assurance. Automated checks verify that every base text produces exactly nine variants with balanced labels and no cross-split leakage; 5% of outputs per perturbation type are additionally spot-checked for readability (see Appendix A).

COAT is, to our knowledge, the first Chinese adversarial toxicity benchmark combining comprehensive perturbation coverage (8 types), strict orthogonality ($P(\text{toxic} \mid \text{perturbed}) = 0.5$), and explicit NLVA annotation.

4 Experiments and Analysis

4.1 Experimental Setup

We evaluate four categories of models on COAT: (1) **Encoder classifiers**: BERT-base-chinese (Devlin et al., 2019), RoBERTa-wwm-ext and MacBERT (Cui et al., 2020), ERNIE (Sun et al., 2019), and ELECTRA (Clark et al., 2020), fine-tuned on the COAT training set (3 random seeds each); (2) **Proprietary LLMs**: GPT-5.4, DeepSeek-V3, and Claude Sonnet 4.6, evaluated zero-shot via API using the Simple Prompt shown in Appendix E; (3) **Open-source LLM**: Llama3-8B with label-only SFT (direct classification) and three-step Chain-of-Thought SFT (perturbation recognition $\rightarrow$ semantic restoration $\rightarrow$ toxicity judgment); (4) **Step-level DPO**: preference optimization targeting individual CoT reasoning steps. For each step, we construct 300 preference pairs from predictions of the CoT SFT model on training data.

We report Macro-F1 as the primary metric, supplemented by Recall (true positive rate; TPR) on toxic samples and False Positive Rate (FPR) on non-toxic samples. The Recall–FPR pair captures the detection-overcorrection trade-off central to adversarial robustness evaluation.

Encoder models are evaluated on the full test set (30,474 samples) across 3 seeds. All LLM variants (proprietary and open-source) are run on the full test set; metrics are computed over successfully parsed binary predictions, with invalid outputs excluded from metric calculation.

4.2 Multi-Model Benchmarking

Table 3 presents the overall results across all model families.

Rather than ranking models by absolute performance, we highlight three systematic patterns across all model families.

Detection–Overcorrection Trade-off. Models diverge markedly along the detection–overcorrection axis. DeepSeek-V3 shows high Recall (88.9%) together with high FPR (40.0%), whereas ERNIE and Claude show lower FPR and lower Recall. No model simultaneously achieves high detection and low overcorrection. This trade-off is observed across the evaluated models and perturbation types, suggesting that balancing

	Overall (%)			Per-perturbation Macro-F1 (%)								
Model	**F1**	**Recall↑**	**FPR↓**	**clean**	**emoji**	**homo**	**trad**	**shuff**	**py_full**	**py_init**	**vsim**	**split**
Encoder Classifiers (fine-tuned, 3-seed avg.)												
BERT-base	74.6	71.4	23.0	78.5	75.7	75.4	78.3	77.5	71.9	71.7	70.9	71.6
RoBERTa-wwm	75.4	73.7	23.9	79.2	76.6	76.1	79.0	77.5	73.2	72.3	72.1	72.3
ERNIE	75.4	67.8	17.4	78.1	78.2	74.6	78.1	78.7	73.4	72.3	72.9	72.3
ELECTRA	74.3	69.8	21.8	77.4	73.9	74.3	77.8	76.9	72.6	72.7	71.4	71.6
MacBERT	70.0	62.1	22.2	72.9	69.7	70.5	73.0	73.2	67.5	67.8	68.0	67.4
Proprietary LLMs (zero-shot)												
GPT-5.4	73.7	66.0	18.4	75.9	74.9	74.7	74.4	75.0	77.1	68.7	72.2	68.6
DeepSeek-V3	73.9	**88.9**	40.0	80.1	76.9	77.2	78.8	79.4	73.5	66.7	68.8	60.1
Claude 4.6	73.5	63.6	16.0	74.8	76.0	73.5	73.8	78.4	75.6	67.5	70.8	66.2
Open-source LLM (Llama3-8B)												
Simple Prompt	62.9	40.9	10.8	63.2	62.1	59.0	63.9	61.9	66.9	66.8	60.4	59.8
Label-only SFT	76.8	77.6	23.9	81.5	80.0	76.5	80.6	78.0	75.1	73.7	72.6	**73.3**
CoT SFT	78.9	78.3	20.5	**83.8**	81.8	**80.0**	**84.1**	81.2	79.0	75.0	73.1	72.3
Step1-DPO	78.7	77.2	19.9	83.4	**82.0**	79.8	**84.1**	80.7	78.5	74.7	72.9	72.0
Step2-DPO	**79.0**	78.2	20.2	83.6	81.9	79.8	**84.1**	**81.3**	79.1	**75.5**	**73.4**	72.6
Step3-DPO	78.9	77.0	19.3	83.4	**82.0**	79.8	**84.1**	**81.3**	**79.2**	74.7	72.7	72.5

Table 3: Main results on COAT. **Recall** = true positive rate (TPR↑); **FPR** = false positive rate (↓). Per-perturbation columns are ordered by increasing linguistic ambiguity (left = low, right = high). **Bold** = best in each column (FPR excluded due to the detection–overcorrection trade-off). Encoder results are 3-seed averages (max std < 0.8 pp across all cells).

detection sensitivity and overcorrection remains challenging under perturbation.

Consistent Vulnerability Profile. Across the trained Llama variants and encoder classifiers, vsim, split, and py_init generally fall among the more challenging perturbation types, although the exact ordering varies by model (see Table 3). This recurrent vulnerability profile is consistent with semantic restoration being a shared bottleneck. The consistency of this broader difficulty pattern across models suggests that the linguistic properties of the perturbations make a substantial contribution to the observed performance differences.

Selective Advantage of CoT Reasoning. CoT SFT achieves its largest gains on perturbation types amenable to explicit reasoning (e.g., trad: 80.6→84.1 F1 from Label-only to CoT; shuff: 78.0→81.2), while yielding limited improvement on types where semantic restoration is hardest (vsim: 72.6→73.1; split: 73.3→72.3). This selectivity suggests that CoT benefits primarily the recognition step—once a perturbation is correctly identified, traditional characters and shuffled text are straightforward to restore, whereas visually similar glyphs and radical decompositions require character-structure knowledge beyond what chain-of-thought alone provides.

Cross-Paradigm Comparison. Encoder classifiers and LLMs exhibit complementary failure profiles. Encoders show relatively uniform degradation across perturbation types, reflecting their reliance on subword token distributions that are disrupted by any character manipulation. In contrast, proprietary LLMs show higher variance: GPT-5.4 handles py_full well (77.1%) but struggles with py_init (68.7%), while DeepSeek-V3 achieves high detection at the cost of severe overcorrection across all types. The open-source Llama3-8B with CoT training reaches 78.9% F1, outperforming the evaluated proprietary LLMs on this benchmark and suggesting that explicit reasoning over perturbation structure can improve robustness under these evaluation conditions.

4.3 Finding 1: Evaluation Bias and Shortcut Learning Risk

The Structural Bias in Existing Benchmarks. In conventional adversarial benchmarks (Yang et al., 2025), perturbations are applied exclusively to toxic texts, establishing $P(\text{toxic} \mid \text{perturbed}) = 1$. A model trained on such data can achieve near-perfect detection via the shortcut $\hat{y} = \mathbf{1}[\text{contains perturbation features}]$ (Geirhos et al., 2020).

Our benchmark, where $P(\text{toxic} \mid \text{perturbed}) = 0.5$, provides a diagnostic lens. If models had learned content semantics, their False Positive Rate

Model	Clean FPR	Pert. FPR	Δ
BERT-base	18.9	23.5	+4.6
RoBERTa	20.8	24.3	+3.5
ERNIE	21.9	16.8	−5.1
Label-only SFT	18.9	24.6	+5.7
CoT SFT	16.2	21.0	+4.8

Table 4: False Positive Rate (FPR, %) on clean vs. perturbed non-toxic samples.

(FPR) on perturbed non-toxic samples should be comparable to that on clean non-toxic samples. Table 4 examines this.

Analysis. Four of five models exhibit $\Delta > 0$, revealing residual sensitivity to perturbation features even under COAT's balanced design. The sole exception, ERNIE ($\Delta = -5.1$), combines the lowest Recall (67.8%) with the lowest perturbed-sample FPR (16.8%), reflecting a more conservative prediction profile. Notably, the positive gap also remains for Llama3-8B with CoT trained on balanced data ($\Delta = +4.8$), showing that perturbation-feature sensitivity remains measurable even without label–perturbation imbalance (the conventional $P(\text{toxic} \mid \text{perturbed}) = 1$ setting).

Separating Feature Sensitivity from Perturbation Difficulty. Perturbation-induced information loss offers another account of the elevated FPR ($\Delta > 0$): distorted inputs may simply be harder to classify. NLVA provides a natural control for this possibility (§4.4). NLVA-positive clean samples are unperturbed, yet they still show consistently higher FPR (ΔFPR $= +1.3$ to $+5.9$). This parallel pattern links overcorrection to perturbation-like surface features themselves, beyond the difficulty introduced by explicit perturbation.

4.4 Finding 2: Overcorrection Patterns Reveal Perturbation-Induced Ambiguity

Having established that models retain a residual perturbation-toxicity association, we now investigate whether overcorrection variation across perturbation types is systematic, and specifically whether it correlates with the degree of linguistic ambiguity each type introduces.

We compute the False Positive Rate (FPR) on non-toxic samples for each perturbation type across all Llama3-8B variants. Table 5 reveals a general tendency for high-ambiguity perturbations to occupy the higher-FPR end.

Type	Simple	Label	CoT	S1	S2	S3	Amb.
homo	5.9	18.4	15.5	15.6	15.0	14.5	Low
emoji	6.5	20.2	15.7	15.2	15.4	14.6	Low
trad	7.2	20.6	14.1	13.2	14.0	13.0	Low
shuff	7.1	25.3	17.7	18.6	17.3	16.9	Med
py_full	16.6	26.2	22.4	20.8	21.5	19.9	Med
py_init	22.3	27.4	25.3	24.5	23.8	24.1	High
vsim	12.5	29.0	26.8	25.8	25.2	26.2	High
split	12.7	29.4	32.2	30.3	30.8	30.1	High

Table 5: False Positive Rate (FPR, %) by perturbation type. Amb. = linguistic ambiguity level, operationalized by the degree to which the perturbed surface form overlaps natural Chinese offensive expressions: **High** = output systematically coincides with profanity or slurs (split, vsim, py_init); **Med** = partial overlap with offensive patterns (shuff, py_full); **Low** = produces distinctive artifacts rarely confused with natural text (emoji, homo, trad).

Key Findings. First, across the trained Llama variants, high-ambiguity perturbations generally occupy the higher-FPR end. Second, the three high-ambiguity types produce surface forms that overlap with naturally occurring variants captured by the NLVA annotation: split and vsim can resemble naturally occurring mixed-script or character-form patterns, while py_init can resemble naturally occurring Latin/pinyin abbreviations. Third, training changes the FPR profile non-uniformly. The Simple Prompt combines low FPR with low Recall (40.9%); after CoT SFT, Recall rises to 78.3% and FPR increases on low-ambiguity types (e.g., emoji: 6.5→15.7). High-ambiguity types remain at the high-FPR end after SFT (e.g., py_init: 22.3→25.3).

We partition clean samples by NLVA status (758/3,386 = 22.4% NLVA-positive in test). If predictions are sensitive to perturbation-like surface features, NLVA-positive samples should exhibit higher FPR. Table 6 shows this pattern consistently.

All nine evaluated models exhibit ΔFPR > 0 (Table 6), showing that the pattern appears consistently across the evaluated model families. Encoders show $\Delta = +4.0$ to $+5.2$; among LLMs, the gap widens from $+1.3$ (Simple Prompt) to $+5.9$ (CoT SFT). Step2-DPO narrows the gap to $+3.9$, the largest reduction among the step-level variants. These results decompose overcorrection into two categories:

1. **Perturbation-processing errors** (low-ambiguity types): the model fails to handle the perturbation itself, addressable through improved training and CoT reasoning.
2. **Ambiguity-related errors** (high-ambiguity

Model	NLVA+	NLVA−	Δ	n_+
Encoder classifiers (3-seed avg.)				
BERT	22.5	17.9	+4.6	379
RoBERTa	24.8	19.6	+5.2	379
ERNIE	25.0	21.0	+4.0	379
ELECTRA	23.1	18.5	+4.6	379
MacBERT	26.2	21.3	+4.9	379
LLM variants (Llama3-8B)				
Simple Prompt	7.7	6.4	+1.3	323
Label-only SFT	23.1	17.9	+5.2	324
CoT SFT	19.6	13.7	+5.9	317
Step2-DPO	18.6	14.7	+3.9	317

Table 6: FPR (%) on clean non-toxic samples by NLVA status. NLVA+/NLVA− = FPR on NLVA-positive/negative subsets; n_+ = number of NLVA-positive clean non-toxic predictions included in the FPR denominator. All models exhibit $\Delta > 0$.

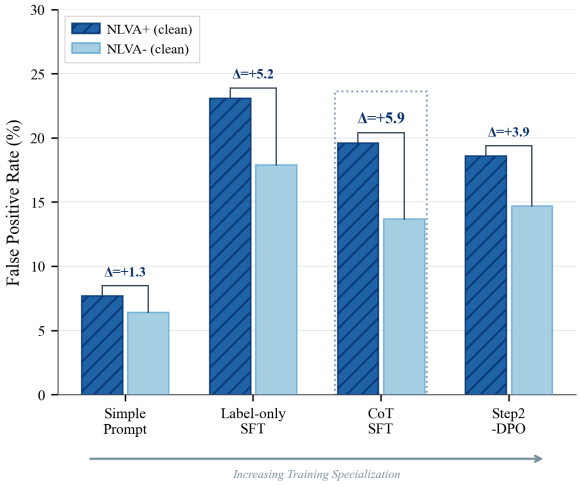


Figure 3: FPR on clean non-toxic samples by NLVA status. NLVA-positive texts consistently show higher FPR than NLVA-negative texts. Step2-DPO narrows the gap.

types and NLVA-positive samples): the perturbed surface form objectively shares features with offensive language, posing a disambiguation challenge that current detection paradigms do not resolve.

5 Discussion and Open Challenges

The preceding experiments reveal two systematic failure patterns—perturbation-feature sensitivity (§4.3) and ambiguity-related overcorrection (§4.4)—and motivate a stage-level analysis of where targeted intervention is most effective. We now conduct a supplementary diagnostic experiment: applying step-level preference optimization to each CoT reasoning stage independently, using the *pattern* of improvement across perturbation types to localize the bottleneck.

Type	Step1	Step2	Step3
Overall	−0.2	**+0.1**	±0.0
py_init	−0.3	**+0.5**	−0.3
vsim	−0.2	**+0.3**	−0.4
split	−0.3	**+0.2**	+0.2
py_full	−0.5	**+0.1**	+0.3
shuff	−0.5	+0.1	+0.1
emoji	+0.2	+0.1	+0.2
homo	−0.2	−0.2	−0.2
trad	±0.0	±0.0	±0.0

Table 7: ΔMacro-F1 (pp) of each step-level DPO variant vs. CoT SFT. Step2 (semantic restoration) yields reliable improvement concentrated on high-ambiguity types.

5.1 Step-Level Error Decomposition

Our CoT SFT model decomposes detection into three explicit reasoning stages: (1) perturbation recognition, identifying whether and how the input has been perturbed; (2) semantic restoration, reconstructing the intended meaning from the distorted surface form; and (3) toxicity judgment, classifying the restored content. We apply DPO (Rafailov et al., 2023) to each stage independently (Step1-, Step2-, Step3-DPO), using 300 preference pairs per stage constructed from errors of the CoT SFT model on training data. We emphasize that this step-level DPO serves as a *diagnostic probe* rather than a production-grade training method. The small budget and single-step intervention reveal which reasoning stage responds to targeted improvement, with the informative signal being the *pattern* of where improvement concentrates, not the magnitude.

Table 7 summarizes the per-perturbation F1 change (Δ) for each step-level DPO variant relative to the CoT SFT baseline.

The critical signal is not the magnitude of improvement (which is modest by design given the 300-pair budget) but its *selectivity* across steps and perturbation types. First, semantic restoration is the core bottleneck: Step2-DPO is the only variant that yields positive gains consistently across all high-ambiguity types—py_init (+0.5), vsim (+0.3), and split (+0.2)—while Step1-DPO (recognition) yields marginal degradation and Step3-DPO (judgment) produces mixed results.

Second, Step2-DPO gains concentrate on high-ambiguity perturbations. Its largest gains occur on py_init, vsim, and split, while improvement is

negligible on low-ambiguity types (emoji, trad).

Third, NLVA disambiguation improves with restoration. Step2-DPO reduces the FPR gap between NLVA-positive and NLVA-negative samples from +5.9 (CoT SFT) to +3.9 (34% relative reduction), compared with only 7% relative reduction for Step1-DPO and 14% for Step3-DPO.

With 300 preference pairs per step, base-text-level bootstrap resampling (1,000 iterations, resampling by base_id to respect the non-independence structure) confirms that Step2-DPO's per-type gains on py_init, vsim, and split are statistically reliable ($p < 0.05$). Cohen's d effect sizes: Step2-DPO yields $d = 0.42$ on py_init, $d = 0.31$ on vsim, and $d = 0.22$ on split (small-to-medium effects), whereas Step1-DPO and Step3-DPO produce $|d| < 0.10$ on the same types.

5.2 Implications

Our findings suggest three research directions:

(1) **Semantic restoration as a pre-processing module.** Dedicated perturbation-type-aware restoration modules leveraging Chinese character structure (radical decomposition, phonetic mapping, glyph similarity) could yield substantial robustness gains. For example, a vsim-aware module could use stroke-order embeddings to map perturbed glyphs back to intended characters before classification. COAT provides a ready-made evaluation framework for such defenses as well as for adversarial training and input normalization approaches.

(2) **Intent-aware NLVA disambiguation.** Automatic NLVA disambiguation requires reasoning about communicative intent rather than surface form. When a pinyin abbreviation coincides with profanity, the model must infer from context whether the author intends the offensive reading or the benign internet slang usage—a pragmatic inference task that current classifiers do not explicitly model.

(3) **Orthogonality as an evaluation standard.** Perturbation orthogonality, $P(\text{toxic} \mid \text{perturbed}) = P(\text{toxic})$, should be a minimum standard for *diagnostic* adversarial robustness benchmarks across languages. We encourage future benchmark designers across different languages to enforce analogous orthogonality constraints when the goal is to isolate detection capability from distributional artifacts.

Scope of the Orthogonality Principle. We distinguish diagnostic from deployment benchmarks. COAT enforces $P(\text{toxic} \mid \text{perturbed}) = 0.5$ to isolate model capability from distributional shortcuts. In deployment, $P(\text{toxic} \mid \text{perturbed})$ may naturally exceed 0.5 because adversarial perturbations are predominantly used for evasion. Our claim is not that all evaluation should be orthogonal, but that at least one diagnostic benchmark with this property is necessary to distinguish genuine detection from shortcut exploitation; the two paradigms are complementary.

6 Conclusion

We present COAT, to our knowledge the first Chinese adversarial toxicity benchmark enforcing perturbation orthogonality across eight perturbation types and 41K samples, with explicit NLVA annotation for fine-grained error attribution. Through systematic evaluation of five encoder classifiers, three proprietary LLMs, and Llama3-8B with CoT reasoning and step-level DPO, we identify two systematic failure patterns: (1) perturbation-feature sensitivity remains evident under balanced training, with four of the five models in the clean-vs.-perturbed analysis showing a 3.5–5.7 percentage-point FPR increase, while ERNIE shows the opposite pattern; (2) in the Llama3-8B per-type analyses, overcorrection varies systematically with perturbation ambiguity, with high-ambiguity types generally associated with higher FPRs across training variants. Step-level diagnosis localizes the shared bottleneck to semantic restoration: only Step2-DPO (restoration) yields consistent improvement on high-ambiguity types ($d = 0.22$–0.42), while recognition and judgment steps produce negligible effects. These findings establish a concrete research agenda: dedicated restoration modules that leverage character-structure knowledge, intent-aware NLVA disambiguation, and perturbation orthogonality as a minimum diagnostic evaluation standard. We present COAT as a diagnostic benchmark for systematic evaluation and analysis of robust Chinese toxicity detection.

Limitations

COAT is limited to Chinese; the specific perturbation types exploit Chinese writing system properties and may not transfer directly. Our eight types are applied individually; real-world attackers may combine multiple perturbations simultaneously, which COAT does not model. All 41K samples derive from 4,586 base texts (each gener-

ating 9 variants), so the effective textual diversity is limited by the source corpus size. Annotation involved only two annotators ($\kappa = 0.83$) with expert adjudication.

Base texts are sourced from ToxiCN (Lu et al., 2023), inheriting its topic distribution and potential biases. Our shortcut learning evidence (§4.3) is diagnostic rather than causal; a direct experiment training on biased data and testing on COAT would strengthen this claim and is left to future work. Step-level DPO uses only 300 pairs per stage (deliberately small for diagnostics); larger-scale optimization may yield different patterns.

Ethics Statement

This work constructs and evaluates a toxicity detection benchmark containing offensive language. Toxic texts are sourced from the publicly released ToxiCN dataset (Lu et al., 2023), which was collected and released with ethical review; we do not generate novel toxic content, collect new user data, or add personally identifying information. NLVA annotation was conducted by two trained annotators who provided informed consent and were fairly compensated; the task labels surface-form ambiguity rather than asking annotators to produce harmful content.

COAT is intended only for research on improving content moderation. We acknowledge the dual-use risk that perturbation knowledge could aid evasion, but these perturbation types are already documented in prior work (Xiao et al., 2024) and used in the wild.

References

Zewen Bai, Liang Yang, Shengdi Yin, Junyu Lu, Jingjie Zeng, Haohao Zhu, Yuanyuan Sun, and Hongfei Lin. 2025. State toxicn: A benchmark for span-level target-aware toxicity extraction in chinese hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10206–10219.

Xi Chen and Shuo Wang. 2025. Pragmatic inference chain (PIC) improving LLMs' reasoning of authentic implicit toxic language. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5826–5841, Suzhou, China. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

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

Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. 2022. Cold: A benchmark for chinese offensive language detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11580–11599.

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, volume 1 (long and short papers)*, pages 4171–4186.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 345–363.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Faeze Ghorbanpour, Daryna Dementieva, and Alexander Fraser. 2025. Data-efficient hate speech detection via cross-lingual nearest neighbor retrieval with limited labeled data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29674–29692, Suzhou, China. Association for Computational Linguistics.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3309–3326.

Ming Shan Hee, Shivam Sharma, Rui Cao, Palash Nandi, Preslav Nakov, Tanmoy Chakraborty, and Roy Ka-Wei Lee. 2024. Recent advances in online hate speech moderation: Multimodality and the role of

large models. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4407–4419.

Sanne Hoeken, Sina Zarrieß, and Özge Alacam. 2024. Hateful word in context classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 172–186.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *Preprint*, arXiv:2312.06674.

Hannah Kirk, Bertie Vidgen, Paul Röttger, Tristan Thrush, and Scott A Hale. 2022. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368.

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

Hanbing Liu, Haoyang Li, Xiaokang Zhang, Ruotong Chen, Haiyong Xu, Tian Tian, Qi Qi, and Jing Zhang. 2025. Uncovering the impact of chain-of-thought reasoning for direct preference optimization: Lessons from text-to-sql. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21223–21261.

Yi Liu, Junzhe Yu, Huijia Sun, Ling Shi, Gelei Deng, Yuqi Chen, and Yang Liu. 2024. Efficient detection of toxic prompts in large language models. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 455–467.

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)*, pages 16235–16250.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Talat, Helen Margetts, and Janet Pierrehumbert. 2021. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 41–58.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2153–2162.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

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.

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.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024. Shieldgemma: Generative ai content moderation based on gemma. *Preprint*, arXiv:2407.21772.

Min Zhang, Jianfeng He, Taoran Ji, and Chang-Tien Lu. 2024. Don't go to extremes: Revealing the excessive sensitivity and calibration limitations of llms in implicit hate speech detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12073–12086.

A Perturbation Generation Algorithms

For each perturbation type, we apply character-level transformations with controlled intensity. Below we summarize the key parameters:

- **Emoji insertion**: Random emojis inserted between 30–50% of character pairs.
- **Homophone**: Target characters replaced with same-tone homophones from a curated dictionary of 2,847 entries.
- **PY_Init / PY_Full**: Characters converted to pinyin initials or full pinyin using the `pypinyin` library.
- **Traditional**: Simplified-to-traditional conversion via the `opencc` library.
- **Shuffle**: Intra-word character order randomized (words segmented by `jieba`).
- **VSim**: Characters replaced with visually similar alternatives from a manually curated glyph-similarity dictionary (1,203 pairs, cosine similarity > 0.85 in stroke-order embeddings).
- **Split**: Characters decomposed into radical/component sequences using the IDS (Ideographic Description Sequence) database.

Unless otherwise specified, all preprocessing packages are used with their default settings; perturbation intensity and dictionary sizes are reported above.

Quality verification: A 5% random sample from each perturbation type is manually spot-checked to ensure readability and correct perturbation application. Samples failing readability checks are regenerated.

B Hyperparameter Details

Encoder Fine-tuning. Learning rate: 2×10^{-5}; batch size: 32; epochs: 5; optimizer: AdamW with weight decay 0.01. Best checkpoint selected by dev-set Macro-F1.

Llama3-8B SFT. Learning rate: 1×10^{-5}; batch size: 8 (gradient accumulation 4); epochs: 3; LoRA rank: 16, α: 32. Training on COAT train split (9K samples).

Step-level DPO. $\beta = 0.1$; learning rate: 5×10^{-7}; 300 preference pairs per step; 1 epoch; initialized from CoT SFT checkpoint. Pairs constructed from CoT SFT errors: correct chains as chosen, naturally occurring step-specific errors as rejected.

Computing Infrastructure. Open-source LLM experiments were conducted on NVIDIA A100 GPUs. The open-source model is Llama3-8B. Encoder classifiers use standard base-size Chinese encoders. Llama3-8B SFT was the dominant computational cost; each 300-pair DPO run required substantially less compute than SFT. Proprietary LLMs were evaluated through API calls and did not require local training compute.

C NLVA Annotation Guidelines

Annotators receive a detailed guideline document defining NLVA and the four feature categories. For each clean (unperturbed) sample, annotators independently answer: "Does this text naturally contain surface features that could be confused with perturbation artifacts?" The decision is based on the following category definitions:

Category 1: Emoji Features. The text contains naturally occurring emojis used for emotional expression that overlap with patterns produced by emoji-insertion perturbation. Decorative emojis between characters trigger NLVA; emojis at sentence boundaries do not.

Category 2: Latin/Pinyin Characters. The text contains naturally occurring Latin characters or pinyin expressions common in internet Chinese (e.g., "yyds", "nb", "666", brand names like "iPhone") that overlap with pinyin-initial or full-pinyin perturbation patterns.

Category 3: Mixed-Script Patterns. The text mixes Chinese characters with other scripts in ways that resemble character-splitting or visual-similarity perturbation (e.g., using Japanese katakana, traditional-simplified mixing in proper nouns, or radical-like sequences that coincidentally form common patterns).

Category 4: Internet Abbreviations. The text uses abbreviation conventions (e.g., homophonic number substitutions like "748" for profanity, alphabetic abbreviations) that overlap with homophone or pinyin perturbation outputs.

Decision Rule. A sample is labeled NLVA-positive if *any* of the four categories is triggered. Most disagreements were concentrated in Categories 2 and 4, where the boundary between "natural internet language" and "perturbation-like" is inherently ambiguous.

Simple Prompt

```
请判断以下中文文本是否属于有害/仇恨言论。
文本：{text}
只输出最终标签，不要解释。
Label: 0 表示无害，Label: 1 表示有害。
答案格式：Label: 0 或 Label: 1
```

Cot Prompt

```
文本：{text}

输出格式要求：
1. 在 <think> 标签内，严格按以下三个步骤输出：
  - 步骤一 [干扰识别]：判断文本使用了哪种干扰（无干扰/表情/谐音/拼音首字母/拼音全拼/繁体/字序打乱/形近字/拆字）。
  - 步骤二 [原句还原]：将文本还原为通顺的中文原句。
  - 步骤三 [有害性分析]：简要分析还原后的语义是否包含歧视、侮辱、攻击或煽动敌意等有害内容。
2. 在 <think> 标签结束后，输出最终标签 "Label: 0"（无害）或 "Label: 1"（有害）。

示例：
<think>
- 步骤一 [干扰识别]：表情
- 步骤二 [原句还原]：你去死吧
- 步骤三 [有害性分析]：包含死亡诅咒，具有明显攻击性。
</think>
Label: 1
```

Figure 4: Prompt templates used in the LLM experiments (shown in original Chinese). The Simple Prompt is shared by the proprietary zero-shot LLMs, zero-shot Llama3-8B, and Label-only SFT. The CoT Prompt requires three-step reasoning (perturbation identification → semantic restoration → toxicity judgment) inside `<think>` tags.

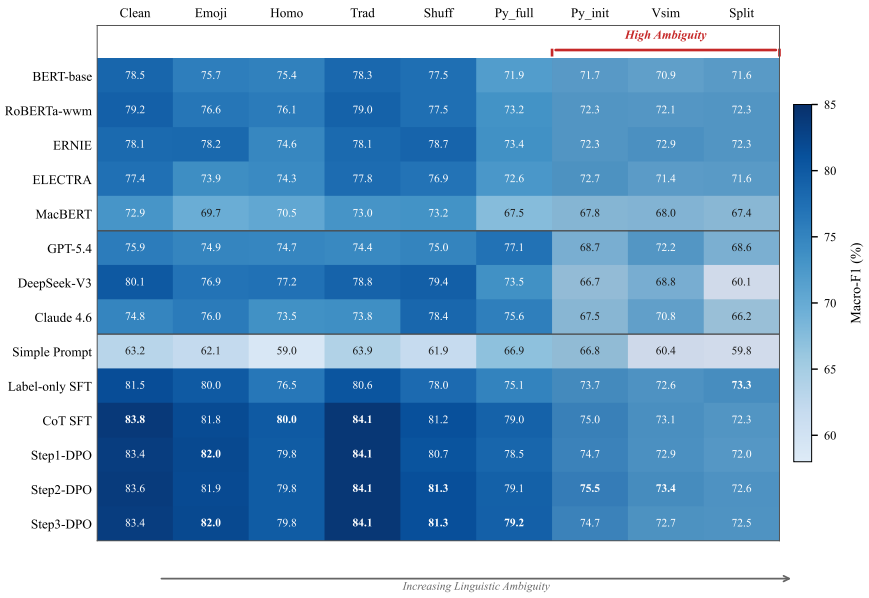


Figure 5: Per-perturbation Macro-F1 (%) across all model families on COAT. Columns are ordered from lower to higher linguistic ambiguity (left to right). Darker cells indicate higher Macro-F1. The low Macro-F1 on py_init, vsim, and split highlights a shared vulnerability to high-ambiguity perturbations.

D Full-Chain DPO Ablation

In addition to the step-level DPO variants reported in §5.1, we also train a full-chain DPO model that optimizes all three reasoning steps simultaneously. Table 8 compares all variants.

Per-Perturbation Breakdown for Full-Chain DPO. Full-chain DPO yields results comparable to the CoT SFT baseline, consistent with the interpretation that undirected optimization across all steps dilutes the targeted diagnostic signal. Only Step2-DPO (semantic restoration) produces a net

Model	F1	Recall	FPR	ΔF1
CoT SFT (baseline)	78.9	78.3	20.5	—
Step1-DPO	78.7	77.2	19.9	−0.2
Step2-DPO	**79.0**	78.2	20.2	+0.1
Step3-DPO	78.9	77.0	19.3	±0.0
Full-Chain DPO	78.8	77.5	19.7	−0.1

Table 8: Comparison of step-level and full-chain DPO variants.

positive ΔF1. The full-chain model shows no systematic improvement on any perturbation type. On high-ambiguity types (py_init: −0.1, vsim: ±0.0, split: −0.2), it fails to capture the gains achieved by Step2-DPO alone (+0.5, +0.3, +0.2 respectively). This supports the need for targeted step-level diagnosis.

E Prompt Templates

Figure 4 presents the exact Chinese prompts used in the LLM experiments. The Simple Prompt is shared by the proprietary zero-shot LLMs, the zero-shot Llama3-8B evaluation, and Label-only SFT. The CoT Prompt specifies a three-step reasoning format inside `<think>` tags with an in-context example.

F Per-Perturbation Performance Heatmap

Figure 5 visualizes the full per-perturbation Macro-F1 matrix, complementing Table 3 by making the degradation pattern on high-ambiguity perturbations directly visible.