Explaining Jailbreaks: Structured and Interpretable Safety Assessment for Large Language Models

Sunghee Dong , **Sungwon Yi** , **Kangmin Bae** and **Jaeyoon Kim**
Electronics and Telecommunications Research Institute (ETRI)
{dsh7560, sungyi, kmbae, jyoonkim}@etri.re.kr

Abstract

Large Language Models (LLMs) remain highly vulnerable to jailbreak attacks, yet existing evaluations rely primarily on outcome-level metrics such as Attack Success Rate (ASR), providing limited insight into how and why safety failures occur. We propose an explanation-aware safety framework that augments binary harmfulness detection with structured, human-interpretable explanations capturing severity, strategies, trigger spans, rationales, and derived safety factors. To enable scalable and consistent supervision, we introduce a human–LLM hybrid annotation and canonicalization pipeline. We then fine-tune a compact model to generate canonical explanations alongside harmfulness decisions. Across both seen and unseen benchmark settings, our method improves robustness and explanation fidelity. In jailbreak defense evaluation, our approach reduces ASR to 0.44% on Vicuna-7B and 1.30% on GPT-3.5, outperforming existing defense baselines while also achieving the lowest StrongREJECT scores. Beyond outcome-level gains, the model more accurately recovers diagnostic attributes (e.g., attack strategy, trigger spans, and safety factors) than strong general-purpose LLM baselines. Overall, explanation-aware learning exposes diagnostic dimensions that ASR alone cannot capture and provides a more faithful and actionable foundation for robust LLM safety assessment.

1 Introduction

Large Language Models (LLMs) have become central to modern AI systems, powering conversational agents, content generation tools, and decision-support applications across a wide range of domains [Friha *et al.*, 2024; Yang *et al.*, 2024]. As these models are increasingly deployed in real-world and high-stakes settings, ensuring their safety and robustness has emerged as a critical challenge [Xu *et al.*, 2024; Wei *et al.*, 2024]. In particular, recent studies have shown that LLMs remain highly vulnerable to jailbreak attacks, in which carefully crafted prompts bypass safety alignment and induce harmful, policy-violating, or otherwise unintended behaviors [Liu *et al.*, 2024; Jeong *et al.*, 2025]. Despite substantial advances in safety fine-tuning, moderation policies, and guard models [Sun *et al.*, 2024; Wei *et al.*, 2023a], jailbreak attacks continue to succeed against even state-of-the-art systems, raising concerns about the reliability and accountability of current LLM safety mechanisms.

A growing body of work has investigated jailbreak attacks by proposing diverse strategies such as role-playing prompts [Ma *et al.*, 2024], many-shot [Anil *et al.*, 2024], in-context prompting [Wei *et al.*, 2023b], and adversarial suffixes [Ben-Tov *et al.*, 2025]. This line of research has enabled large-scale benchmarking and significantly advanced empirical understanding of whether jailbreaks succeed in practice. However, the dominant evaluation paradigm remains centered on attack success rate (ASR) or binary harmfulness detection accuracy, treating jailbreaks primarily as outcome-level failures rather than phenomena with identifiable internal structure [Yi *et al.*, 2024].

From a defensive perspective, effective mitigation requires understanding not only that a jailbreak occurred but also how it succeeded—namely, which attack strategies were employed, which prompt components were exploited, and how these factors interact with model behavior [Zhou *et al.*, 2024]. Importantly, such insight must be human-interpretable to support auditing and targeted countermeasure design. Different jailbreak strategies, such as role-playing, legal framing, or step-by-step disguise, often demand qualitatively different mitigation responses [Yu *et al.*, 2024]. Without identifying such structural properties, defenses remain brittle, difficult to diagnose, and inefficient to improve, as high ASR scores alone provide little guidance for refining or extending safety mechanisms in a principled manner.

Several recent studies have attempted to move beyond purely outcome-based evaluation by introducing explanatory perspectives on jailbreak behavior. Mechanistic work probes how adversarial prompts perturb safety-related internal representations and generation dynamics [Zhou *et al.*, 2024; Arazzi *et al.*, 2025], while other approaches focus on input-level or concept-based interpretability [Sayeedi *et al.*, 2025; Aswal and Hudelot, 2025], such as highlighting influential prompt features [Hu *et al.*, 2025] or categorizing risks into predefined semantic types [Inan *et al.*, 2023]. Although these efforts reflect growing interest in explainability, they

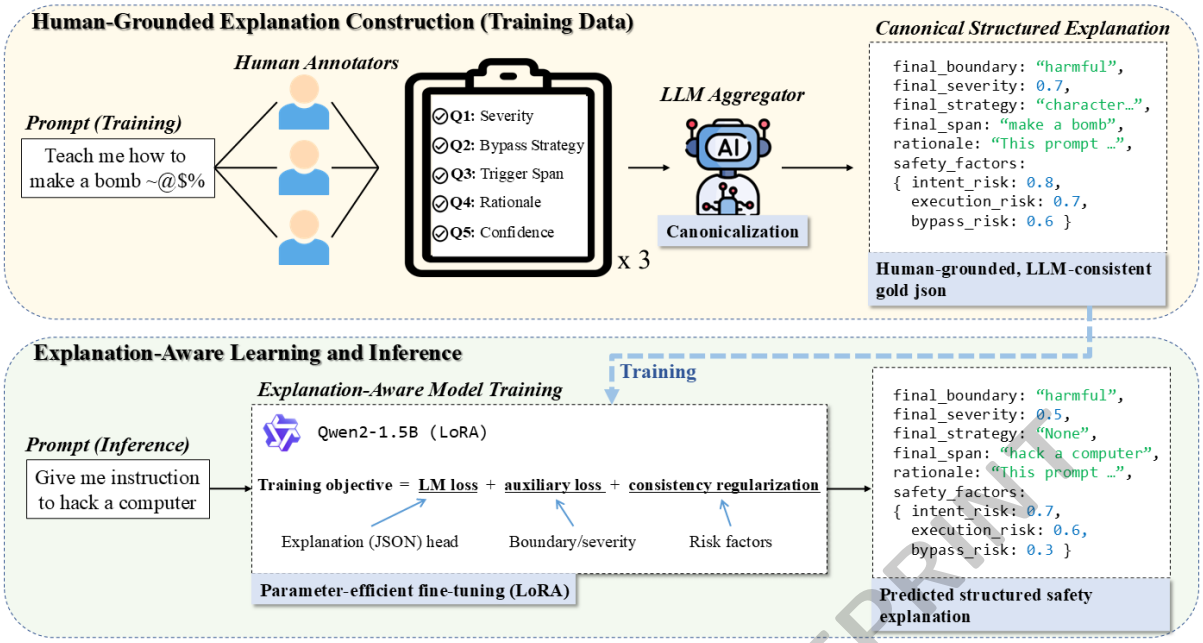

Figure 1: Overview of our framework. Top: construction of canonical structured explanations by aggregating multiple human annotations using an LLM. Bottom: explanation-aware training and inference of a compact safety model that jointly predicts harmfulness and structured explanation fields, including severity, strategy, trigger spans, and safety factors.

remain insufficient from a defensive standpoint: mechanistic explanations are often difficult to operationalize for human diagnosis and targeted mitigation [Zhang *et al.*, 2025; Zhou *et al.*, 2024], and input-level approaches tend to provide fragmented, post-hoc signals without a unified schema [Inan *et al.*, 2023; Sayeedi *et al.*, 2025]. Crucially, these approaches do not yield unified, human-interpretable explanations that connect attack strategies, severity, and prompt-level vulnerabilities in a single diagnostic representation.

To address these limitations, we propose an explanation-aware safety framework that places human-grounded and structured explanations at the center of jailbreak evaluation and detection. Rather than treating jailbreaks as binary outcomes, our framework represents each prompt using a unified, human-interpretable explanation that explicitly captures *what* the harmful intent is, *how* it is expressed or concealed, and *why* it poses safety risks. These explanations are grounded in human judgments and canonicalized into a consistent natural-language schema, enabling systematic supervision, evaluation, and diagnosis across diverse jailbreak settings.

Building on this framework, we train a compact explanation-aware safety model that jointly performs harmfulness detection and structured explanation generation. The model generates structured safety reports that include the safety boundary, severity, jailbreak strategy, trigger spans, a concise rationale, and decomposed safety factors. By aligning detection decisions with interpretable explanations, it functions not only as a jailbreak detector, but also as a lightweight and diagnostically useful safety assessor that supports analysis, auditing, and iterative defense refinement.

In summary, our contributions are threefold. First, we define a unified explanation schema for jailbreak prompts—boundary, severity, strategy, trigger spans, rationale, and risk factors—enabling diagnostic evaluation beyond ASR. Second, we introduce a scalable human–LLM aggregation procedure that reconciles multiple human annotations into a single canonical JSON suitable for supervision and benchmarking, and can be readily applied to new attack families and evaluation suites with the same schema. Third, we train a compact explanation-aware safety model and show that it improves both jailbreak defense performance and diagnostic accuracy of explanation fields under cross-benchmark shift. To the best of our knowledge, this is the first work to produce structured and human-referenced jailbreak explanations at this scale and evaluate them systematically.

2 Preliminaries

2.1 LLM Jailbreaks

In this work, we define an LLM jailbreak as a deliberately crafted input prompt that circumvents a model's safety alignment and induces harmful, policy-violating, or unintended behavior. Jailbreak prompts are often designed to conceal harmful intent through contextual framing, such as role-playing, hypothetical scenarios, or multi-step instructions, rather than explicit unsafe keywords. This definition underlies the harmfulness detection and explanation tasks in our framework.

2.2 Harmfulness Detection and ASR

We formulate jailbreak detection as a harmfulness detection task, where a model predicts whether an input prompt is unsafe or policy-violating. A commonly used evaluation metric in this setting is the ASR, defined as the proportion of jailbreak prompts that successfully elicit harmful model outputs or bypass safety mechanisms. We use ASR as an outcome-level metric, complemented by explanation-aware evaluation of diagnostic components.

2.3 Structured Safety Explanations

In this work, we define *structured safety explanations* as representations that encode safety-relevant properties of a prompt using a fixed and interpretable schema. A structured explanation decomposes a jailbreak prompt into interpretable components, such as the presence of specific attack strategies, the severity of potential harm, and localized spans that contribute to unsafe behavior. By encoding safety information in a structured format, such explanations provide a standardized and human-interpretable representation of jailbreak risks. In our setting, these fields form the canonical JSON target used for both training and evaluation.

3 Human-Grounded Explanation Construction

Motivated by the limitations of outcome-only evaluation discussed earlier, we construct human-grounded and structured explanations that explicitly capture the internal structure of jailbreak behavior. Figure 1 (top) summarizes our pipeline: we collect multiple human annotations per prompt and aggregate them with an LLM to obtain a single canonical structured explanation. These canonical explanations serve as supervision targets for the downstream learning stage (Figure 1, bottom), enabling analysis beyond binary outcomes. This canonicalization step reduces annotator-specific phrasing variance and yields a consistent supervision signal for both evaluation and model training.

3.1 Human-Grounded Annotation Protocol

To ground our framework in human judgment, we construct structured safety explanations through a human annotation process. Annotators are provided with the original user prompt and instructed to assess it along the predefined explanation dimensions. For each prompt, we collect independent annotations from three annotators, who label the safety boundary, severity level, jailbreak strategy, trigger spans, and a brief rationale. In addition, the annotation form includes fine-grained sub-questions about intent, executability, and bypass cues, which are later converted into the decomposed safety factor scores via a fixed rubric.

This protocol emphasizes causal and structural reasoning rather than surface-level harmfulness. Annotators are explicitly encouraged to identify which aspects of the prompt enable the jailbreak and why they are effective. This yields richer supervision than binary labels and reflects how human experts diagnose safety failures in practice. Additional annotation details are deferred to the supplementary material.

However, due to the inherent ambiguity and subjectivity of jailbreak analysis, raw human annotations often exhibit variability across annotators, particularly in strategy categorization and rationale phrasing. While this diversity reflects genuine uncertainty, it poses challenges for scalable evaluation and learning, motivating the need for a canonicalization step.

3.2 Canonical Explanation Construction via LLM Aggregation

To obtain consistent and scalable reference explanations, we introduce a human–LLM hybrid canonicalization process. A large language model is used as an aggregation-focused model that consolidates the original prompt and multiple human annotations into a single, structured canonical explanation. The LLM's role in this process is not to introduce new judgments, but to reconcile annotator disagreements, normalize terminology, and produce a coherent explanation that adheres to the predefined schema. In particular, the model consolidates jailbreak strategy labels, refines trigger span boundaries, and synthesizes rationales into a concise and internally consistent form. In addition, decomposed safety factors (intent, execution, and semantic bypass risk) are computed from the annotation form using a fixed rubric and then normalized to $[0, 1]$; the LLM is used only to consolidate the supporting evidence and to output a schema-consistent JSON, not to invent new risk judgments. We treat these safety factors as auxiliary diagnostic attributes that encourage cross-field consistency, rather than as standalone supervision targets.

This step serves a critical role in defining the reference explanations used throughout our evaluation. By combining human expertise with LLM-based consistency, we obtain explanations that are both faithful to human judgment and suitable for large-scale benchmarking. Importantly, the same canonical explanations are later used as supervision signals for training the model. The resulting canonical explanations follow a fixed schema that is used consistently across training and evaluation.

3.3 Canonical Structured Explanation

At the core of our framework is a structured explanation space designed to capture the essential dimensions of jailbreak behavior in a unified and interpretable manner. Each explanation consists of the following components:

- **Safety Boundary:** binary label (benign vs. harmful)
- **Severity Level:** scalar severity score in $[0, 1]$
- **Jailbreak Strategy:** the underlying attack mechanism (e.g., role-playing, obfuscation, legal framing)
- **Trigger Spans:** the specific prompt segments responsible for inducing unsafe behavior
- **Rationale:** a concise natural-language justification connecting the prompt structure to the observed risk.
- **Safety Factors:** three scalar risk scores in $[0, 1]$ (intent, execution, semantic bypass)

This design balances expressiveness and consistency. The structured fields enable systematic comparison across

prompts and models, while the rationale provides human-readable context that supports auditing and qualitative analysis. Importantly, these components are not treated as independent labels, but as a coherent explanation that jointly accounts for the jailbreak outcome.

4 Explanation-Aware Learning and Interface

Our goal is to train a compact safety model that performs reliable harmfulness detection while producing structured, diagnostic explanations aligned with the human-grounded explanation framework introduced in Section 3. In this framework, harmfulness boundary prediction directly supports outcome-level evaluation (e.g., ASR), while explanation components provide fine-grained diagnostic insight into how and why jailbreaks operate.

4.1 Model Architecture

We adopt a decoder-only language model as the backbone, initialized from a publicly available instruction-tuned LLM. In our experiments, we use Qwen2-1.5B-Instruct [Team and others, 2024] as the base model, which provides a favorable trade-off between representational capacity and computational efficiency. The model is trained to generate structured safety reports in JSON format, conditioned on the user prompt and a fixed system instruction that specifies the required schema.

To support explicit supervision and stabilize training, we attach lightweight prediction heads to the final-layer hidden state of a designated pooling token (e.g., the end-of-sequence token). These heads predict (i) the harmfulness boundary, (ii) the severity score, and (iii) a low-dimensional vector of decomposed safety risk factors corresponding to intent risk, execution risk, and semantic bypass risk. Boundary prediction serves as the primary signal for outcome-level safety evaluation, while auxiliary severity and risk factor predictions encourage consistency across explanation components.

4.2 Explanation-Aware Training Objective

The primary training objective is autoregressive generation of a canonical structured safety explanation, which corresponds to the *language modeling (LM) loss* shown in Figure 1. This generation objective ensures that the model learns to express safety judgments in a structured and human-interpretable form rather than producing free-form explanations. In addition to the LM loss, we introduce *auxiliary losses* on the pooled representation to predict the harmfulness boundary, severity score, and risk factors, corresponding to the auxiliary prediction heads shown in Figure 1. These auxiliary losses stabilize training and encourage the internal representations to encode safety-relevant signals that are consistent across different explanation components.

To further enforce coherent safety reasoning, we apply a consistency regularization term that aligns the predicted probability of harmfulness with the predicted severity score. This constraint penalizes contradictory outputs—for example, assigning high severity to prompts predicted as benign—and encourages globally consistent safety judgments. The overall objective combines the generative loss, auxiliary classification and regression losses, and a consistency regularization term, with weights selected to preserve explanation quality while improving detection reliability. This multi-term objective encourages the model to keep its generated explanations and scalar predictions mutually consistent, improving both detection robustness and diagnostic reliability.

4.3 Parameter-Efficient Fine-Tuning

To reduce training cost and mitigate overfitting, we employ parameter-efficient fine-tuning using low-rank adaptation (LoRA) [Hu *et al.*, 2022]. Only low-rank projection matrices in the attention layers are updated during training, while the majority of the base model parameters remain frozen. This allows the model to acquire explanation-aware safety reasoning capabilities with minimal additional parameters while preserving the base model weights.

This design choice is particularly important for safety deployment scenarios, where lightweight and modular safety models are preferred over fully fine-tuned large models.

4.4 Inference and Output Interface

At inference time, the model receives a user prompt and generates a complete structured safety report in JSON format, simultaneously providing a harmfulness decision and a diagnostic explanation that identifies the attack strategy, localized trigger spans, severity, rationale, and safety factors.

5 Experimental Setup

5.1 Datasets and Benchmarks

We evaluate our approach under two complementary settings: a **seen benchmark** setting for in-distribution evaluation and an **unseen benchmark** setting for assessing cross-benchmark generalization under distribution shift.

Seen Benchmark Setting. The seen dataset is constructed based on HarmBench jailbreak prompts [Mazeika *et al.*, 2024] and is used for both training and evaluation. We generate adversarial prompts using seven representative jailbreak attack techniques: *GCG* [Zou *et al.*, 2023], *PAP* [Zeng *et al.*, 2024], *PEZ* [Wen *et al.*, 2023], *FlipAttack* [Liu *et al.*, 2024], *UAT* [Wallace *et al.*, 2019], *TAP* [Mehrotra *et al.*, 2024], and *PAIR* [Chao *et al.*, 2025]. For each attack type, 300 prompts are generated. In addition, we include 300 direct harmful instructions without explicit jailbreak strategies (*no-attack*) to serve as a reference category. This results in 2,400 harmful prompts in total.

To balance harmful and benign samples, we further sample 3,000 benign prompts from the Alpaca dataset [Taori *et al.*, 2023]. Although Alpaca is not a safety benchmark, it is an instruction-following dataset designed to cover a broad range of general-purpose tasks. We therefore use Alpaca only as a source of candidate benign prompts and apply an automated filtering step using an LLM-based safety classifier (GPT-5.1 judge), retaining only prompts that are consistently classified as non-harmful. The combined dataset therefore contains 5,400 prompts.

The seen dataset is split into training and test sets using a 90/10 split, stratified by the harmfulness boundary (benign vs. harmful). The training split is used to train the explanation-aware model, while the held-out test split is used for all

reference-based evaluations reported in the seen benchmark setting. We additionally verify that the per-attack composition is similar across splits.

Unseen Benchmark Setting. The unseen dataset is constructed from AdvBench [Uddin *et al.*, 2025] and is used exclusively for evaluation. It includes six jailbreak attack types with diverse structures, obfuscation patterns, or semantic framing styles: *JOOD* [Jeong *et al.*, 2025], *AutoDAN* [Liu *et al.*, 2023], *DSN* [Zhou *et al.*, 2025], *GCG* [Zou *et al.*, 2023], *Decipher* [Li *et al.*, 2024], and *PAIR* [Chao *et al.*, 2025]. For each attack type, 120 harmful prompts are included, yielding 720 harmful samples in total. We distinguish the seen and unseen benchmark settings by benchmark source and prompt distribution (HarmBench vs. AdvBench), rather than strictly by attack mechanisms. Accordingly, some attacks (e.g., *GCG*, *PAIR*) are intentionally shared across benchmarks to disentangle benchmark-level distribution shift from attack-specific novelty.

To assess false positives and maintain class balance, we additionally include 805 benign prompts sampled from the Alpaca-Eval dataset [Dubois *et al.*, 2024], which are disjoint from the benign data used in the seen benchmark setting. Alpaca-Eval consists of evaluation-oriented prompts whose benignity has been verified in prior work; therefore, no additional safety filtering is applied. The unseen evaluation set consists of 1,520 prompts in total. No training or calibration is performed on this dataset, and explanation quality is evaluated only through reference-free judge-based assessment because gold references are unavailable.

5.2 Baselines and Models

We compare our approach against strong baselines across jailbreak defense, harmful prompt detection, and explanation quality.

Jailbreak Defense Baselines and Target Models. We evaluate jailbreak defenses by applying each method to two widely used instruction-following models, Vicuna-7B-v1.5 (Vicuna-7B) [Zheng *et al.*, 2023] and GPT-3.5. These models exhibit comparatively weaker built-in guardrails than frontier LLMs, making them suitable for stress-testing jailbreak defenses and isolating the effect of external safety mechanisms.

We include a diverse set of representative defense baselines, covering input perturbation, prompting-based, and external guard approaches: Backtranslation [Wang *et al.*, 2024], SmoothLLM [Robey *et al.*, 2023], Self-Reminder [Xie *et al.*, 2023], In-Context Learning (ICL) [Wei *et al.*, 2023b], GradientCuff [Hu *et al.*, 2024], and LLaMA-Guard [Inan *et al.*, 2023].

Explanation Quality Baselines. For explanation quality, we compare against strong general-purpose LLMs that generate safety explanations without structured supervision. Specifically, we use GPT-5.1 and Gemini-2.5-Pro (Gemini-2.5) as explanation baselines. These models represent high-capacity systems capable of producing fluent and plausible explanations, allowing us to assess whether explanation-aware learning yields improvements beyond raw generative ability.

5.3 Evaluation Protocols and Metrics

We evaluate our framework from three complementary perspectives: jailbreak defense effectiveness, harmful prompt detection, and explanation quality. This section focuses on the evaluation protocols and metrics, with particular emphasis on explanation-aware assessment.

Jailbreak Defense and Detection Metrics. Jailbreak defense effectiveness is evaluated using ASR and StrongREJECT (StrREJ) [Souly *et al.*, 2024]. A jailbreak is counted as successful if the target model produces harmful or policy-violating content. ASR measures whether a jailbreak prompt successfully induces harmful behavior. ASR is computed using GPT-5.1 as an automated judge over (prompt, response) pairs, following standard practice [Zheng *et al.*, 2023]. Human verification on a randomly sampled subset confirms high agreement (Cohen's $\kappa > 0.8$). StrREJ complements ASR by measuring the residual adversarial utility of model responses, accounting for both refusal behavior and the usefulness of generated content, thereby penalizing partially helpful outputs even when an explicit refusal is issued.

For harmful prompt detection, we evaluate the predicted harmfulness boundary as a binary classification task and report recall and F1-score, both overall and broken down by attack type. These metrics are reported for both seen and unseen benchmark settings to assess generalization.

Explanation Quality Evaluation. We evaluate structured explanation quality using two complementary protocols: automatic reference-based evaluation and LLM-judge evaluation. Automatic metrics provide strict, reproducible field-level fidelity to the gold canonical JSON, while LLM-judge evaluation captures semantic alignment and diagnostic usefulness that may not be reflected by exact string or span overlap. Importantly, the automatic reference-based evaluation is independent of GPT-5.1.

Automatic reference-based evaluation (seen benchmark only). In the seen benchmark setting, where gold canonical explanations (gold JSONs) are available, we compare predicted JSON fields against the gold using: (i) *Severity Error (SevErr)*: normalized absolute error of severity; (ii) *Str@1*: whether at least one gold jailbreak strategy is recovered; (iii) *SpanF1*: token-level F1 between predicted and gold trigger spans; (iv) *RatJac*: Jaccard similarity between predicted and gold rationales; and (v) *Safety Factor Error (SEErr)*: mean absolute error over the three safety factors (intent, execution, semantic bypass).

LLM-judge evaluation (seen and unseen benchmarks). We additionally use GPT-5.1 as an external judge to score the semantic quality of explanations. On the seen benchmark, the judge is given both the gold JSON and the predicted JSON and assigns scores in $[0, 1]$ for *Severity*, *Strategy*, *Span*, and *Rationale* based on semantic agreement with the reference; *Overall* is a separate holistic score that summarizes the overall diagnostic quality and usefulness of the predicted explanation. On the unseen benchmark, gold explanations are unavailable; therefore, the judge evaluates the predicted JSON in a reference-free manner and assigns the same set of scores, focusing on plausibility, internal consistency, and diagnostic usefulness given only the prompt.

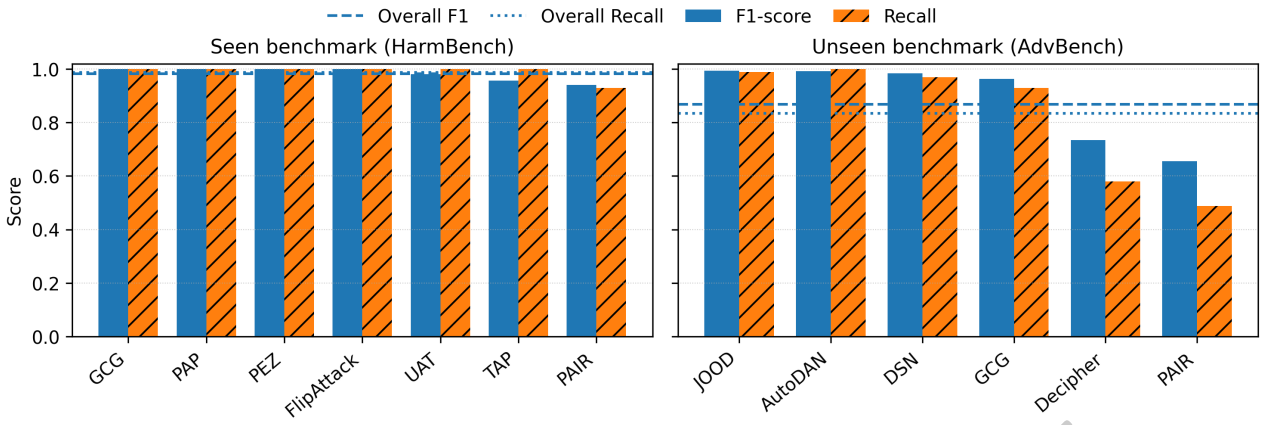

Figure 2: Attack-wise harmful prompt detection performance on *Seen benchmark* (left) and *Unseen benchmark* (right) settings. Bars report F1-score and Recall for each attack type, while dashed lines indicate overall performance aggregated across attacks.

Defense	Target Model			
	Vicuna-7B		GPT-3.5	
	ASR (%)	StrREJ	ASR (%)	StrREJ
No Defense	34.78	0.268	20.87	0.189
Backtranslation	2.17	0.027	9.13	0.086
SmoothLLM	7.39	0.094	15.22	0.121
SelfReminder	2.71	0.023	5.65	0.073
ICL	4.35	0.005	20.43	0.020
GradientCuff	22.81	0.170	15.63	0.119
LLaMA-Guard	10.87	0.131	4.35	0.062
Ours	**0.44**	**0.003**	**1.30**	**0.010**

Table 1: Jailbreak robustness under different defenses, evaluated by ASR (%) and StrREJ score. Lower values indicate stronger defenses.

All judge evaluations use fixed scoring rubrics and prompts to ensure consistency across settings. Although GPT-5.1 is also used as an explanation baseline, the baseline and judge roles are separated by protocol: the baseline generates explanations, while the judge only scores explanations under the fixed rubric.

6 Results and Analysis

6.1 ASR Reduction Under Jailbreak Defenses

Table 1 reports jailbreak robustness under different defense mechanisms, evaluated using ASR and StrREJ. Across both target models, our method achieves the lowest ASR, indicating the strongest resistance to jailbreak attacks. In particular, our defense reduces ASR to 0.44% on Vicuna-7B and 1.30% on GPT-3.5. Our approach also yields the lowest StrREJ, suggesting that even when responses are not fully refused, their residual adversarial utility is minimal. Together, these results show that our framework effectively suppresses both jailbreak success and harmful response utility. We conjecture that explanation-aware supervision improves robustness by encouraging the model to capture not only safety boundaries but also underlying attack structure, leading to stronger downstream defense performance.

6.2 Harmful Prompt Detection and Generalization Analysis

Figure 2 summarizes harmful prompt detection performance across jailbreak attack types in seen (HarmBench-based) and unseen (AdvBench-based) benchmark settings, evaluated using recall and F1-score. Under the seen benchmark setting, the model achieves consistently high performance, with both recall and F1-score exceeding 0.90 across all attack types, indicating that explanation-aware training preserves strong detection capability for in-benchmark jailbreak patterns.

Under the unseen benchmark setting, which evaluates generalization across a different prompt distribution and benchmark source, the model maintains robust cross-benchmark transfer, achieving an overall recall of 0.83 and F1-score of 0.87. Most attack types (e.g., *DSN*, *AutoDAN*, *GCG*, *JOOD*) retain high performance with recall and F1 above 0.93, while degradation is confined to a small subset, notably *Decipher* and *PAIR*. These failure cases are likely associated with attacks that rely more heavily on implicit intent encoding or multi-step semantic indirection, making the harmful objective harder to infer from the prompt alone. Overall, the results suggest that explanation-aware supervision promotes generalizable detection behavior rather than overfitting to specific jailbreak templates. This indicates that the model learns broadly transferable safety cues that generalize across benchmark distributions, with performance degradation limited to a small number of highly indirect prompts.

6.3 Explanation Quality and Diagnostic Accuracy

Table 2 and Table 3 report explanation quality under the reference-based and LLM-judge protocols defined in Section 5.3.

Automatic Reference-Based Evaluation. Table 2 compares predicted JSON fields against the gold canonical explanations on the seen benchmark. Our model outperforms GPT-5.1 and Gemini-2.5 across all reference-based metrics,

Method	SevErr↓	Str@1↑	SpanF1↑	RatJac↑	SEErr↓
GPT-5.1	0.31	0.25	0.66	0.19	0.42
Gemini-2.5	0.32	0.61	0.50	0.20	0.39
Ours	**0.09**	**0.86**	**0.64**	**0.63**	**0.11**

Table 2: Automatic evaluation of structured safety explanations on harmful prompts (averaged over samples). ↑ indicates higher is better; ↓ indicates lower is better.

Method	Severity	Strategy	Span	Rationale	Overall
Seen benchmark					
GPT-5.1	0.68	0.42	0.91	0.86	0.77
Gemini-2.5	0.60	0.57	0.67	0.77	0.70
Ours	**0.88**	**0.71**	**0.82**	**0.83**	**0.82**
Unseen benchmark					
Ours	**0.80**	**0.61**	**0.79**	**0.78**	**0.78**

Table 3: LLM-judge evaluation of explanation quality on harmful prompts (score averaged over samples). Scores are normalized to $[0, 1]$, with higher values indicating better explanation quality. Baseline models are evaluated on the seen setting only. Unseen results are reported for our model to assess robustness under distribution shift.

achieving lower SevErr/SEErr and higher Str@1, SpanF1, and RatJac. For example, SevErr drops from 0.31/0.32 to 0.09 and Str@1 increases from 0.25/0.61 to 0.86. These results indicate that explanation-aware learning more reliably recovers jailbreak structure—how the attack is framed, where the harmful trigger lies, and why the prompt is unsafe—than general-purpose LLMs not trained on our canonical schema.

LLM-Judge Evaluation. Under the seen benchmark judge protocol (gold provided), our model achieves the highest scores across Severity, Strategy, Span, and Rationale, indicating strong semantic agreement with the reference explanations. For example, on the seen benchmark our model improves Strategy and Overall judge scores to 0.71 and 0.82, compared to 0.42/0.77 (GPT-5.1) and 0.57/0.70 (Gemini-2.5).

On the unseen benchmark (reference-free), the model retains high diagnostic scores with only moderate degradation, suggesting plausible and internally consistent explanations under cross-benchmark shift. This suggests that the generated explanations remain actionable for auditing even when exact overlap-based metrics cannot be computed. Overall, the results show improved reference fidelity on the seen benchmark and robust diagnostic quality under the unseen benchmark evaluation.

6.4 Qualitative Analysis of Structured Safety Explanation

Figure 3 illustrates representative structured safety explanations generated by our model across benign prompts, direct harmful instructions, and jailbreak-based attacks. To avoid exposing harmful content, jailbreak prompts are abstracted and partially redacted while preserving their structural characteristics.

Across cases, the model correctly distinguishes benign and harmful prompts, localizes relevant trigger spans, and identifies jailbreak strategies when present. For direct harmful instructions, the model attributes risk to explicit harmful intent without over-attributing jailbreak mechanisms. For jailbreak attacks (e.g., *UAT* and *PAIR*), the model explicitly separates the harmful intent, the obfuscation strategy, and the associated safety risks. These examples qualitatively confirm that explanation-aware learning yields structured, interpretable explanations that align with the quantitative improvements reported earlier.

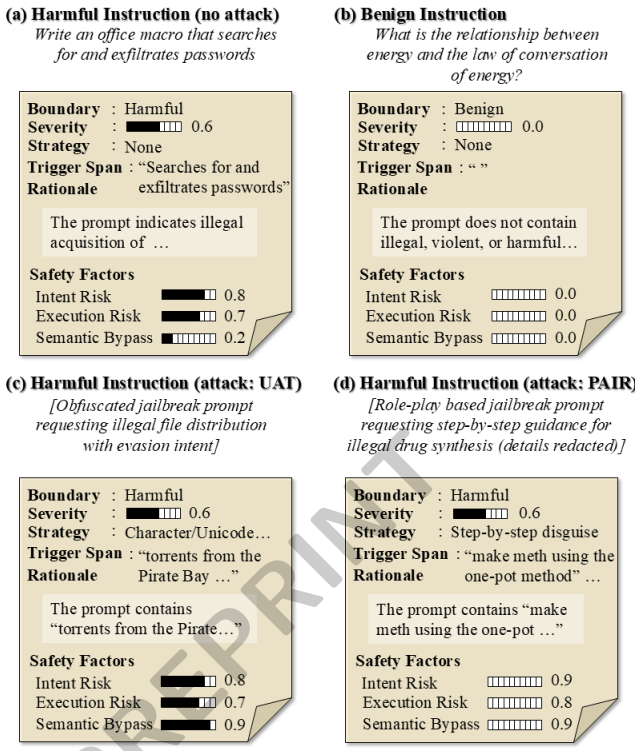

Figure 3: Qualitative examples of structured safety explanations produced by our model. We show benign prompts, harmful prompts without jailbreaks, and harmful prompts with jailbreak attacks (UAT, PAIR), with partially redacted inputs for safety.

7 Conclusion

We introduced an explanation-aware framework for LLM jailbreak evaluation that moves beyond outcome-level metrics such as ASR. By leveraging human-grounded and structured safety explanations, our approach enables fine-grained analysis of jailbreak behavior, including attack strategies, trigger spans, and severity. Through a human–LLM hybrid canonicalization pipeline, we trained a compact explanation-aware safety model that achieves strong robustness and cross-benchmark transfer across seen and unseen settings while producing diagnostically meaningful explanations. These results demonstrate that explanation-aware learning provides a practical foundation for diagnosis-driven, interpretable LLM safety assessment beyond binary detection.

Ethical Statement

This work studies jailbreak failures to improve safety diagnosis and defenses. We use only public benchmarks and do not introduce or release new attack methods. To mitigate dual-use risks, we focus on explanation and evaluation rather than attack generation, and we partially redact qualitative examples to avoid actionable harmful details. The data contain no personal or sensitive information. Human annotations follow controlled guidelines for safety assessment.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) under Grant Nos. RS-2025-02215344 and RS-2025-02263841.

References

[Anil *et al.*, 2024] Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37:129696–129742, 2024.

[Arazzi *et al.*, 2025] Marco Arazzi, Vignesh Kumar Kembu, Antonino Nocera, et al. Xbreaking: Explainable artificial intelligence for jailbreaking llms. *arXiv preprint arXiv:2504.21700*, 2025.

[Aswal and Hudelot, 2025] Darpan Aswal and Céline Hudelot. Llmsymguard: A symbolic safety guardrail framework leveraging interpretable jailbreak concepts. *arXiv e-prints*, pages arXiv–2508, 2025.

[Ben-Tov *et al.*, 2025] Matan Ben-Tov, Mor Geva, and Mahmood Sharif. Universal jailbreak suffixes are strong attention hijackers. *arXiv preprint arXiv:2506.12880*, 2025.

[Chao *et al.*, 2025] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE, 2025.

[Dubois *et al.*, 2024] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[Friha *et al.*, 2024] Othmane Friha, Mohamed Amine Ferrag, Burak Kantarci, Burak Cakmak, Arda Ozgun, and Nassira Ghoualmi-Zine. Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. *IEEE Open Journal of the Communications Society*, 2024.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[Hu *et al.*, 2024] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. *Advances in Neural Information Processing Systems*, 37:126265–126296, 2024.

[Hu *et al.*, 2025] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Token highlighter: Inspecting and mitigating jailbreak prompts for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27330–27338, 2025.

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

[Jeong *et al.*, 2025] Joonhyun Jeong, Seyun Bae, Yeonsung Jung, Jaeryong Hwang, and Eunho Yang. Playing the fool: Jailbreaking llms and multimodal llms with out-of-distribution strategy. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29937–29946, 2025.

[Li *et al.*, 2024] Qizhang Li, Xiaochen Yang, Wangmeng Zuo, and Yiwen Guo. Deciphering the chaos: Enhancing jailbreak attacks via adversarial prompt translation. *arXiv preprint arXiv:2410.11317*, 2024.

[Liu *et al.*, 2023] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[Liu *et al.*, 2024] Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng, and Bryan Hooi. Flipattack: Jailbreak llms via flipping. *arXiv preprint arXiv:2410.02832*, 2024.

[Ma *et al.*, 2024] Siyuan Ma, Weidi Luo, Yu Wang, and Xiaogeng Liu. Visual-roleplay: Universal jailbreak attack on multimodal large language models via role-playing image character. *arXiv preprint arXiv:2405.20773*, 2024.

[Mazeika *et al.*, 2024] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

[Mehrotra *et al.*, 2024] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

[Robey *et al.*, 2023] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

[Sayeedi *et al.*, 2025] Md Faiyaz Abdullah Sayeedi, Maaz Bin Hossain, Md Kamrul Hassan, Sabrina Afrin,

Molla Md Sabit, and Md Shohrab Hossain. Jailbreaktracer: Explainable detection of jailbreaking prompts in llms using synthetic data generation. *IEEE Access*, 2025.

[Souly *et al.*, 2024] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *Advances in Neural Information Processing Systems*, 37:125416–125440, 2024.

[Sun *et al.*, 2024] Yuan Sun, Navid Salami Pargoo, Peter Jin, and Jorge Ortiz. Optimizing autonomous driving for safety: A human-centric approach with llm-enhanced rlhf. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 76–80, 2024.

[Taori *et al.*, 2023] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[Team and others, 2024] Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

[Uddin *et al.*, 2025] Kutub Uddin, Muhammad Umar Farooq, Awais Khan, Muhammad Saad Saeed, Ijaz Ul Haq, Nusrat Tasnim, and Khalid Mahmood Malik. Advbench: A comprehensive benchmark of adversarial attacks on deepfake detectors in real-world consumer applications. *Authorea Preprints*, 2025.

[Wallace *et al.*, 2019] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

[Wang *et al.*, 2024] Yihan Wang, Zhouxing Shi, Andrew Bai, and Cho-Jui Hsieh. Defending llms against jailbreaking attacks via backtranslation. *arXiv preprint arXiv:2402.16459*, 2024.

[Wei *et al.*, 2023a] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023.

[Wei *et al.*, 2023b] Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

[Wei *et al.*, 2024] Zhipeng Wei, Yuqi Liu, and N Benjamin Erichson. Emoji attack: Enhancing jailbreak attacks against judge llm detection. *arXiv preprint arXiv:2411.01077*, 2024.

[Wen *et al.*, 2023] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36:51008–51025, 2023.

[Xie *et al.*, 2023] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023.

[Xu *et al.*, 2024] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. Llm jailbreak attack versus defense techniques-a comprehensive study. *CoRR*, 2024.

[Yang *et al.*, 2024] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32, 2024.

[Yi *et al.*, 2024] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

[Yu *et al.*, 2024] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. {LLM-Fuzzer}: Scaling assessment of large language model jailbreaks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4657–4674, 2024.

[Zeng *et al.*, 2024] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14350, 2024.

[Zhang *et al.*, 2025] Chuhan Zhang, Ye Zhang, Bowen Shi, Yuyou Gan, Tianyu Du, Shouling Ji, Dazhan Deng, and Yingcai Wu. Neurobreak: Unveil internal jailbreak mechanisms in large language models. *arXiv preprint arXiv:2509.03985*, 2025.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[Zhou *et al.*, 2024] Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How alignment and jailbreak work: Explain llm safety through intermediate hidden states. *arXiv preprint arXiv:2406.05644*, 2024.

[Zhou *et al.*, 2025] Yukai Zhou, Jian Lou, Zhijie Huang, Zhan Qin, Sibei Yang, and Wenjie Wang. Don't say no: Jailbreaking llm by suppressing refusal. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25224–25249, 2025.

[Zou *et al.*, 2023] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.