Overflip: Repetition-Induced Label Flips in Guardrail Models

Anonymous ACL submission

Abstract

Guardrail models are classifiers deployed to screen malicious prompts and responses in LLM-based services. To meet latency constraints, many lightweight guardrails adopt compact Transformer backbones (e.g., DeBERTa) that are trained with short context windows (typically around 512 tokens) and rely on bucketed relative positional encodings to operate on longer inputs. Prior evaluations largely assume that a guardrail's decision is stable as the input is lengthened. We show that this assumption can fail. We identify *Overflip*, a repetition-induced instability where simply repeating a prompt causes the guardrail's prediction to flip as the sequence grows, enabling a practical overflip-style bypass (MAL→BEN) without semantic manipulation. Across 9 widely used lightweight guardrail models, 5 exhibit MAL→BEN flips on a benchmark of 100 prompts, with confidence margins shrinking steadily with repetition. Across 9 widely used lightweight guardrail models, 5 exhibit MAL→BEN flips on a benchmark of 100 prompts; among vulnerable models, flip rates range from 8% to 87%, with first flips occurring at roughly 2.6k–9.4k tokens. Our analysis suggests Overflip is not explained by traditional attention-dilution baselines (e.g., benign padding or shuffling): despite preserving content, repetition can homogenize token-level attention over repeated structure and induce a distinct, more gradual attention-dispersion trajectory than padding. Because the bypassed prompt remains semantically intact and is still readily understood by downstream business LLMs, it can transmit malicious intent after passing the guardrail. These findings expose input length as an attack surface for safety filters and motivate length-robust evaluation and mitigation for lightweight guardrail deployments.

1 Introduction

As large language models (LLMs) become increasingly deployed in production systems, specialized *guardrail models* have emerged as a critical safety component (Wang et al., 2025), which are designed to detect malicious prompts and potentially harmful responses before they reach the business model and end users. To maintain low detection latency, guardrail models could be lightweight classifiers (Zheng et al., 2024), typically employing compact architectures (such as Deberta (He et al., 2020; ProtectAI.com, 2023) and MordenBert (Warner et al., 2024; Zheng et al., 2024)) with limited context windows (commonly around 512 tokens).

The limited context window is clearly mismatched with modern deployment scenarios that routinely accumulate long prompts (e.g., multi-turn dialogues and retrieval-augmented inputs) and may reach far beyond 512 tokens (Paulsen, 2025). In practice, many lightweight guardrails therefore rely on relative positional encodings to operate on longer sequences when needed (Shaw et al., 2018a). Despite this, length robustness is rarely treated as a first-class requirement when deploying low-latency safety filters.

Prior evaluations of guardrail models have focused primarily on classification accuracy, robustness to adversarial perturbations (DAIR.AI, 2025), and detection of specific attack patterns such as jailbreaks and prompt injections (Chao et al., 2024). However, these assessments implicitly assume that a model's decision remains stable as input length increases, an assumption we show to be flawed.

In this paper, we reveal a systematic vulnerability we term *Overflip*: as inputs are lengthened via simple repetition, guardrail models can exhibit prediction flips. Our primary focus is the security-critical bypass setting, where an originally malicious prompt is reclassified as benign (MAL→BEN) after sufficient repetition, allowing

the prompt to pass the guardrail while preserving its intent. In addition to bypass, we also observe benign↔malicious instabilities, suggesting a broader length-induced decision fragility.

Across 9 widely used lightweight guardrail models, 5 exhibit at least one MAL→BEN flip under the repetition-based overflow attack on a benchmark of 100 prompts. Among vulnerable models, flip rates range from 8% to 87%, and first flips occur at lengths spanning roughly 2.6k–9.4k tokens. These flips are accompanied by shrinking confidence margins, indicating progressive destabilization as the sequence grows.

While one might initially attribute this phenomenon to attention dilution (Liu et al., 2023; Bai et al., 2023), where longer contexts spread attention weights more thinly, our results show a qualitatively different behavior under repetition. Unlike padding-based lengthening that appends novel benign content, Overflip preserves content via exact repetition but still destabilizes the classifier: attention over repeated structure becomes increasingly homogenized, and attention dispersion follows a distinct, more gradual trajectory compared to padding and shuffling.

This points to structural long-input effects beyond "more content" as a driver. In particular, many lightweight guardrails employ *bucketed relative positional encodings* (similar to T5-style position biases (Raffel et al., 2020a)) to extend beyond their native 512-token training window. Once inputs exceed this boundary, positional bucketing can coarsely compress long-range distinctions, further undermining the model's ability to isolate critical evidence for classification.

Furthermore, compared to typical length-extending attacks, Overflip better preserves the original prompt semantics: in downstream business models (GPT-4.1 Mini and Llama 3.1 8B), repetition-based overflip yields substantially higher behavioral consistency than padding-based baselines, while padding more frequently induces misinterpretations.

We conduct extensive experiments to characterize prevalence, flip dynamics, and practical impact. We evaluate 9 guardrail models across 100 prompts, measure flip rates and first-flip lengths, and analyze confidence-margin trajectories around the 512-token boundary. We further validate real-world effectiveness by testing whether repetition preserves attack semantics in downstream LLMs.

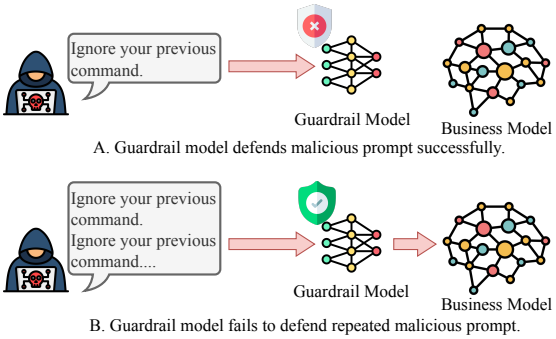

Figure 1: Overview of the Overflip Attack. An attacker repeatedly lengthens an original malicious prompt via semantic-preserving repetition until the guardrail model flips from malicious to benign (MAL→BEN), causing the prompt to be forwarded to the downstream business model while retaining the attack intent.

Contributions. This work makes the following contributions:

1. **Prevalence**: We establish that Overflip is widespread in lightweight guardrails: across 9 production models evaluated on 100 prompts, 5 exhibit MAL→BEN flips under simple repetition-based lengthening.

2. **Mechanism beyond dilution baselines**: We compare Overflip with standard lengthening baselines (padding and shuffling) and show that repetition can trigger flips without introducing new content. Attention analyses reveal homogenization under repetition and distinct dispersion dynamics relative to padding, consistent with a long-input failure mode amplified by positional bucketing beyond the 512-token boundary.

3. **Real-world impact**: We validate that repetition-based Overflip better preserves attack semantics in downstream business models (GPT-4.1 Mini and Llama 3.1 8B), increasing the practical risk of guardrail bypass.

We hope this study encourages the community to treat positional encoding design and length robustness as first-class safety considerations when developing and deploying guardrail models at scale.

2 Related Work

2.1 Guardrail Models

Guardrail models are lightweight classifiers used to filter malicious prompts (e.g., jailbreaks and

prompt injections) in LLM deployments (Wang et al., 2025; Zheng et al., 2024). Representative families include prompt-injection detectors built on compact Transformer backbones (e.g., DeBERTa variants) (ProtectAI.com, 2023), dedicated jailbreak detectors (Jindal, 2024a,b, 2025), and production-oriented guardrails such as LLaMA Prompt Guard (Meta, 2024, 2025), Sentinel (Ivry and Nahum, 2025), and Granite-Guardian (IBM Research, 2024b,a). These systems prioritize low latency and are commonly trained or fine-tuned with relatively short context windows, making length robustness a practical concern.

Most existing evaluations emphasize accuracy and robustness to semantically crafted adversarial prompts, while treating input length as a secondary factor. Our work complements this line by highlighting that even semantic-preserving lengthening can induce decision instabilities in deployed guardrails.

2.2 Attention Dilution

"Attention dilution" describes long-context degradation where evidence becomes harder to use as attention is distributed across many tokens (Vaswani et al., 2017; Liu et al., 2023). It motivates long-context architectures (e.g., sparse/local attention and recurrence) (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Dai et al., 2019) and also appears in prompt-level attacks that saturate context and create instruction competition (e.g., many-shot jailbreaking and prompt injection) (Anil et al., 2024; Liu et al., 2024; Li et al., 2024; Yi et al., 2023; Wallace et al., 2024).

Our findings are related but distinct: Overflip uses *semantic-preserving repetition* and still induces flips, with attention dynamics that differ from padding-based dilution baselines. In particular, padding introduces novel benign content and quickly increases attention dispersion, while repetition creates many identical occurrences and leads to attention homogenization over repeated structure. This points to a long-input failure mode that cannot be reduced to "more competing content" alone.

2.3 Positional Encodings in Transformers

Transformer models inject order via positional encodings. Absolute encodings (Shaw et al., 2018a) attach position-specific embeddings, while relative encodings (Shaw et al., 2018b) add position-dependent attention biases (including variants that couple content and position more tightly (Huang and Xu, 2020)). To scale beyond a fixed window, many efficient classifiers adopt T5-style *bucketed* relative positions (Raffel et al., 2020b), and modern alternatives such as ALiBi (Press et al., 2022) and RoPE (Su et al., 2021) also support extrapolation to long contexts. Our work highlights that positional-design choices can affect *classification stability* under lengthening, complementing prior work that primarily evaluates long-context generation and understanding.

In lightweight safety classifiers, positional extrapolation is often adopted as an engineering mechanism to accept long inputs even when training is dominated by short sequences. Our results suggest that this mismatch can surface as label instability under repetition, motivating length-aware stress testing when deploying guardrails.

3 Problem Setup and Threat Model

3.1 Task Definition

We model a guardrail as a binary classifier $f : \mathcal{X} \to \{0, 1\}$ that maps an input prompt $x \in \mathcal{X}$ to a label $y \in \{0, 1\}$, where 0 denotes benign and 1 denotes malicious (e.g., prompt injection or jailbreak). The model outputs class probabilities $p_{\text{mal}}(x)$ and $p_{\text{safe}}(x)$ with $p_{\text{mal}}(x) + p_{\text{safe}}(x) = 1$, and predicts via thresholding:

$$f(x) = \mathbb{1}[p_{\text{mal}}(x) > 0.5] \tag{1}$$

Given an input x with base length L tokens and prediction $y(L) = f(x)$, an **Overflip** occurs if there exists a lengthened version x' with length $L' > L$ such that x' is produced through repetition and $y(L') = f(x') \neq y(L)$.

Unless otherwise stated, we focus on the security-critical **MAL→BEN** case, where a malicious prompt is misclassified as benign and forwarded downstream.

Attack objective. As shown in Fig. 1, we consider a deployment pipeline where an input prompt is first screened by a guardrail model f; if classified as benign, it is forwarded to a downstream business model. The attacker's goal is to *bypass* the guardrail while preserving as much malicious intent as possible in the prompt that reaches the business model. Concretely, given an original malicious prompt x with $f(x) = 1$, the attacker applies a *semantic-preserving* lengthening operator based on repetition,

$$\text{Rep}_n(x) = \underbrace{x \oplus x \oplus \cdots \oplus x}_{n \text{ times}}, \quad n \geq 2, \tag{2}$$

and seeks an n such that $f(\mathrm{Rep}_n(x)) = 0$. Because repetition keeps the original instructions intact (and even amplifies them), the resulting prompt retains the attack intent when it is forwarded to the business model, while potentially being misclassified as benign by the guardrail.

3.2 Threat Model

We consider attackers who exploit length-dependent behavior without crafting semantically novel adversarial content. The core insight is that a guardrail's decision may become unstable under semantic-preserving lengthening, enabling misclassification without changing the underlying intent.

Attacker capabilities. The attacker can submit prompts to the deployment pipeline and apply semantic-preserving lengthening (e.g., repetition). They may estimate token counts via the production tokenizer and observe predictions in a black-box setting (or confidence scores in a gray-box setting). We assume no access to model parameters or training data.

The attacker is constrained to *text-level* manipulations (no character-level obfuscation or semantic rewriting), keeping the attack realistic for production pipelines.

Lengthening strategies. We consider three lengthening strategies that preserve the original semantic intent:

- **Exact repetition** concatenates the prompt n times: $x' = x \oplus x \oplus \cdots \oplus x$.
- **Benign padding** appends neutral text after the prompt: $x' = x \oplus b_1 \oplus b_2 \oplus \cdots \oplus b_k$, where each b_i is drawn from a benign pool.
- **Sentence shuffling** perturbs word order while preserving critical keywords, testing sensitivity to surface patterns rather than semantics.

Among them, **Exact repetition** is the attack approach of Overflip, while **Benign padding** and **Sentence shuffling** are common attention dilution approaches, which we use for baseline comparison.

3.3 Research Questions

We structure our study around three research questions:

RQ1 (Prevalence). *How prevalent is Overflip across widely deployed lightweight guardrail models?* We evaluate repetition-based lengthening on a diverse benchmark of prompts and measure model-level flip occurrence, flip rates, and first-flip lengths.

RQ2 (Mechanism vs. attention dilution). *How does Overflip differ from traditional attention-dilution strategies, and what causes the flips?* We compare exact repetition with padding and shuffling baselines and analyze attention behavior under lengthening, including token-level attention homogenization and attention-dispersion statistics (e.g., normalized CLS-attention entropy) as length grows and at the flip point.

RQ3 (Real-world impact). *Does Overflip preserve attack semantics when the prompt reaches downstream business LLMs?* We test manipulated prompts on representative backend models and quantify behavioral consistency to assess whether repetition-based lengthening better retains the original attack intent than attention-dilution baselines.

4 Experimental Setup

Guardrail Models. We evaluate 9 lightweight guardrail models spanning multiple architectures and training objectives. Models are listed in Table 1 with key specifications.

The suite includes widely deployed DeBERTa-based classifiers (e.g., Llama Prompt Guard and ProtectAI prompt-injection detectors), as well as a DistilBERT jailbreak detector and more recent backbones such as ModernBERT and Qwen-based sentinels. Model sizes range from 22M to 596M parameters, covering both small on-device guardrails and larger but still low-latency filters. Across vendors, these models target different threat categories: jailbreak detection, prompt-injection detection, or both. In our experiments, several models exhibit at least one MAL→BEN flip under the repetition-based overflow attack (Table 1, **Flip Occurrence**), highlighting that length-induced instability is not confined to a single architecture family.

Datasets. We evaluate the attack on a curated set of 100 validated malicious prompts. Prompts are collected from two Hugging Face sources (`jayavibhav/prompt-injection-safety` and `ahsanayub/malicious-prompts`), deduplicated, filtered to be English-only, and constrained to short inputs (<500 characters). We further verify that all prompts are detected as unsafe by `meta-llama/Llama-Prompt-Guard-2-86M` (100% verification), and retain a diverse mix of prompt styles including instruction override,

Model Vendor	Model Name	Mode Arch.	Para. Size	Detection Type	Flip Occurrence
meta-llama	Llama-Prompt-Guard-2-22M (LPG-2-22M)	deberta-v2	22M	Both	✓
	Llama-Prompt-Guard-2-86M (LPG-2-86M)	deberta-v2	86M	Both	✓
protectai	deberta-v3-base-prompt-injection (dvbpi)	deberta-v3	140M	Prompt Injection	✓
	deberta-v3-small-prompt-injection-v2 (dvspiv2)	deberta-v3	140M	Prompt Injection	–
madhurjindal	Jailbreak-Detector (V1) (JD-V1)	distilbert	66M	Jailbreak	–
	Jailbreak-Detector-Large (JD-L)	deberta-v2	279M	Jailbreak	–
	Jailbreak-Detector–2-XL (JD-2XL)	qwen2	500M	Jailbreak	–
qualifire	prompt-injection-sentinel (PIS)	modernbert	395M	Jailbreak	✓
	prompt-injection-jailbreak-sentinel-v2 (PIJS-v2)	qwen3	596M	Both	✓

Table 1: Guardrail models evaluated in this paper. We report each model's vendor, backbone architecture, parameter size, and intended detection type (jailbreak vs. prompt injection vs. both). **Flip Occurrence** indicates whether the model exhibits at least one prediction flip under our SafeGuardrail Overflip Attack on any prompt in the evaluation set.

Model Name	Flip Rate	Flip Round	Flip Length
LPG-2-22M	15%	45.7	2753
LPG-2-86M	87%	61.1	4960
dvbpi	8%	85.0	5376
PIS	65%	131.3	9424
PIJS-v2	33%	39.2	2599

Table 2: Overall attack outcomes across models. **Flip Rate** is the fraction of prompts that trigger at least one prediction flip. **Flip Round** is the average iteration index when the first flip occurs. **Flip Length** is the average input length (in tokens) at the first flip.

harmful requests, and role-playing jailbreak patterns.

All base prompts are kept short (<512 tokens after tokenization) so that the initial prediction is made within each model's native context window.

Downstream Business Models (RQ3). To assess real-world impact, we test whether lengthening strategies preserve attack semantics after a successful guardrail bypass. We run bypassed prompts on two backend LLMs: GPT-4.1 Mini and Llama 3.1 8B. Concretely, we start from the 100 malicious prompts above and use LPG-2-86M as the target guardrail. We retain 92 prompts that are initially flagged as malicious, apply three lengthening strategies (Overflip repetition, benign padding, and sentence shuffling), and keep the successful bypassed prompts, yielding 172 attacked prompts in total (Overflip: 71; Padding: 92; Shuffling: 9). For each backend model, we query both the original and attacked prompts and categorize responses into acceptance (*ac*), rejection (*rej*), or misinterpretation (*mis*) to quantify semantic preservation.

Hardware. Experiments are conducted on Apple Silicon (MPS), GPU (NVIDIA A100), and CPU backends. Batch size is set to 1.

5 Results

5.1 Prevalence Across Models (RQ1)

Overflip is *not* universal, but it is widespread across lightweight guardrails: in the MAL→BEN setting (Section 3), 5 of 9 evaluated models exhibit at least one flip when inputs are lengthened via exact repetition (Table 1, **Flip Occurrence**). Table 2 summarizes outcomes for these vulnerable models.

Flips are not observed in every model, but they appear in multiple vendors and backbone families (DeBERTa, ModernBERT, and Qwen), indicating the issue is not tied to a single implementation. Among models that flip, the flip rate ranges from 8% to 87% (Table 2). First flips can require substantial lengthening, with average first-flip lengths spanning 2.6k–9.4k tokens and occurring after dozens of repetition rounds (Table 2).

Overall, these results show that Overflip is model-dependent but not isolated: it appears across multiple architectures and vendors.

LLaMA Prompt Guard. Both variants exhibit MAL→BEN flips, with markedly different prevalence: LPG-2-22M flips on 15% of prompts, while LPG-2-86M flips on 87% (Table 2). This suggests that increased model capacity does not necessarily translate to improved length robustness.

Prompt-injection detectors (ProtectAI). The DeBERTa-v3-base prompt-injection detector (dvbpi) shows a low but non-zero flip rate (8%), whereas dvspiv2 shows no flips in our setting (Table 1–2).

Sentinel-style models. We observe flips in both a ModernBERT-based sentinel (PIS, 65%) and a Qwen-based model (PIJS-v2, 33%), with PIS flipping at particularly long inputs on average (9,424 tokens; Table 2).

Jailbreak detectors. In contrast, the three jailbreak

detectors in Table 1 do not exhibit flips under our evaluation protocol (**Flip Occurrence**: –), suggesting that length-induced bypass is model-dependent even within the same detection category.

We next analyze why repetition induces these flips and how the resulting dynamics differ from attention-dilution baselines.

5.2 Mechanism (RQ2)

Repetition triggers flips via a distinct long-input failure mode: attention becomes increasingly homogenized over repeated structure, destabilizing the decision boundary without introducing new content. We support this with two complementary analyses on a representative guardrail classifier: (i) token-level attention under repetition (Figure 2), and (ii) attention-dispersion dynamics across lengthening strategies (Figure 3).

Token-level homogenization under repetition. We extract the last-layer CLS-to-token attention weights while repeating a fixed base prompt k times. To compare patterns across lengths, we *collapse* the repeated sequence back onto a single copy of the base prompt: tokens aligned to the same base position across the k repetitions are aggregated (summed), yielding a length-normalized attention vector over base tokens. Figure 2 shows that as k increases the aggregated distributions become increasingly similar: high-attention tokens preserve their relative ordering and per-token weights converge. This *attention homogenization* suggests that repetition can smooth evidence allocation, reducing the model's ability to isolate a single salient occurrence.

Attention dispersion across strategies. To compare Overflip with attention-dilution baselines (padding and shuffling), we quantify how dispersed the CLS attention becomes as length grows. **Attention Entropy Metric.** To quantify the concentration of the classifier's (llama86M) attention across three categories of attacks, we compute the Shannon entropy of the CLS token's attention distribution over the input sequence. For a transformer-based classifier with H attention heads in the final layer, we first extract the attention weights from each head h:

$$\mathbf{a}^{(h)} = \text{Attention}^{(h)}_{[\text{CLS}],:} \in \mathbb{R}^n \tag{3}$$

where n is the sequence length and $\mathbf{a}_i^{(h)}$ represents the attention weight assigned to the i-th token by

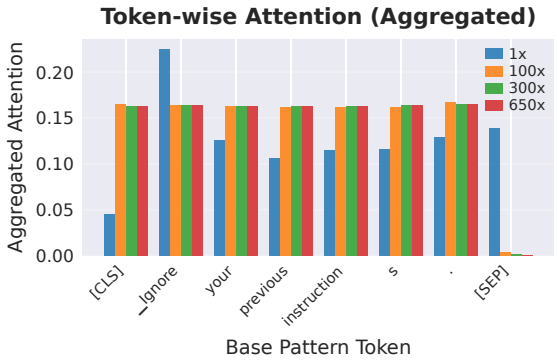

Figure 2: Aggregated CLS-to-token attention under repetition. For each repetition count k, we sum attention weights over tokens aligned to the same base-prompt position across the k copies, producing a length-normalized attention profile over the base tokens. As k grows, the profiles converge, illustrating attention homogenization under repetition.

head h. The entropy for each head is computed as:

$$H^{(h)} = -\sum_{i=1}^{n} a_i^{(h)} \log a_i^{(h)} \tag{4}$$

To enable comparison across sequences of varying lengths, we normalize by the maximum possible entropy (achieved under uniform attention):

$$\hat{H}^{(h)} = \frac{H^{(h)}}{\log n} \tag{5}$$

The final normalized entropy is the average across all heads:

$$\hat{H} = \frac{1}{H} \sum_{h=1}^{H} \hat{H}^{(h)} \in [0, 1] \tag{6}$$

A low entropy ($\hat{H} \to 0$) indicates concentrated attention on a small subset of tokens, while high entropy ($\hat{H} \to 1$) indicates diffuse attention across the sequence.

Uniqueness of Overflip Attack. Figure 3 shows distinct entropy dynamics across strategies. Padding increases entropy most rapidly and yields the highest flip rate: appended benign sentences introduce novel content that quickly disperses attention across semantically diverse regions, but such content is often contextually incoherent and thus easier to detect. Shuffling produces only minor entropy changes (e.g., $\Delta\hat{H} < 0.1$) and a low flip rate (9.8%), consistent with a classifier that is more sensitive to *keyword presence* than to local word order.

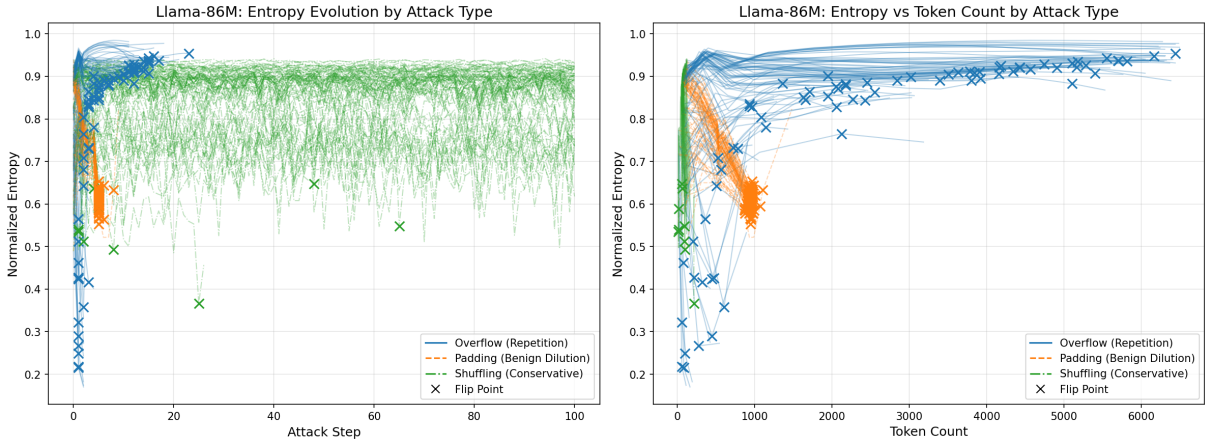

Figure 3: Normalized CLS-attention entropy under three lengthening strategies (overflip repetition, benign padding, and sentence shuffling). Padding disperses attention most aggressively and achieves the highest flip rate, while overflip follows a distinct, more gradual trajectory despite preserving semantic content.

Overflip exhibits a more gradual (and more variable) entropy trajectory while preserving semantic content. This matches the homogenization effect in Figure 2: when the same token pattern appears repeatedly, softmax-normalized attention must be distributed across multiple identical occurrences, progressively diffusing evidence over the repeated segments. Taken together, these results show that repetition can destabilize decisions without introducing new content, and it behaves differently from padding-based attention dilution.

This distinction also holds when conditioning on the *flip event* itself. Appendix Figure 4 summarizes outcomes at the first flip point: padding flips almost always and at relatively short lengths, whereas Overflip flips at substantially longer token counts and with more variable attention dispersion (entropy) at the flip. These complementary statistics reinforce that Overflip is not simply "more dilution"—it induces flips under a different long-input regime while keeping content unchanged.

Finally, we ask whether this content-preserving bypass translates into *semantic preservation* once the prompt reaches downstream business LLMs.

5.3 Real-world Impact (RQ3)

For completeness, Appendix Table 4 reports the full transition breakdown (including cross-category transitions such as ac→rej and ac→mis) for each backend model and perturbation.

Overflip is operationally relevant because it tends to *preserve* attack semantics after bypassing the guardrail: when the bypassed prompt reaches downstream business LLMs, the response category often matches that of the original prompt. We measure this practical impact by comparing backend-model behavior on original prompts versus bypassed prompts (see Section 4).

Setup. We evaluate semantic preservation on two backend LLMs (GPT-4.1 Mini and Llama 3.1 8B) using the business-model evaluation protocol described in Section 4. We compare outputs for the original prompts and their bypassed counterparts under Overflip, padding, and shuffling, and label each response as acceptance (*ac*), rejection (*rej*), or misinterpretation (*mis*).

Transition patterns. We identify several transition patterns that characterize model behaviors under Overflip, padding, and shuffling.

LLM Behavioral Persistence. Both models generally preserve their initial response category after being attacked. GPT-4.1 Mini shows 69.7% stability (accept→accept: 42.4%; reject→reject: 24.4%; misinterpret→misinterpret: 2.9%), while Llama 3.1 8B shows 63.3% stability (reject→reject: 45.9%; accept→accept: 11.0%; misinterpret→misinterpret: 6.4%). This consistency in behavior indicates that both models possess defenses against all three attack types. It also suggests that our semantic robustness evaluation is reliable and does not favor any particular attack strategy.

Attack-Specific Semantic Robustness. Overflip best preserves original behavior: GPT-4.1 Mini keeps 87.3% of responses in the same category (accept→accept, reject→reject, misinterpret→misinterpret), while Llama 3.1 8B preserves 69.0%. Padding is the most disruptive, producing

GPT-4.1 Mini

Overflip ($n = 71$)		
Original → Attacked	Count	%
ac → ac	41	57.7%
rej → rej	19	26.8%
mis → mis	2	2.8%

Padding ($n = 92$)		
Original → Attacked	Count	%
ac → ac	28	30.4%
rej → rej	19	20.7%
mis → mis	2	2.2%

Shuffling ($n = 9$)		
Original → Attacked	Count	%
ac → ac	4	44.4%
rej → rej	4	44.4%
mis → mis	1	11.1%

Llama 3.1 8B

Overflip ($n = 71$)		
Original → Attacked	Count	%
rej → rej	33	46.5%
ac → ac	13	18.3%
mis → mis	3	4.2%

Padding ($n = 92$)		
Original → Attacked	Count	%
rej → rej	43	46.7%
mis → mis	7	7.6%
ac → ac	6	6.5%

Shuffling ($n = 9$)		
Original → Attacked	Count	%
rej → rej	3	33.3%
mis → mis	1	11.1%

Table 3: Semantic robustness (Original → Attacked) under three strategies (Overflip/Padding/Shuffling). Outputs are categorized as acceptance (*ac*), rejection (*rej*), or misinterpretation (*mis*). Overflip best preserves semantics, while padding induces misinterpretations most often across both models.

the highest rates of accept→misinterpret transitions (25.0% for GPT-4.1 Mini and 13.0% for Llama 3.1 8B). Shuffling leads to almost complete category stability for GPT-4.1 Mini but increases refusal rates for Llama 3.1 8B.

Taken together, these results indicate that Overflip is both *prevalent* in lightweight guardrails and *operationally relevant*: repetition can bypass the guardrail while keeping the downstream model's behavior largely consistent with the original attack intent.

6 Discussion

Length as an attack surface. Our findings suggest that input length itself is an attack surface for lightweight guardrail models. Unlike many prompt-based attacks that rely on carefully crafted paraphrases or obfuscation, Overflip can be triggered by simple repetition and therefore requires little adversarial expertise. The phenomenon is also not tied to a single vendor or backbone, and it may arise naturally in long-context pipelines (e.g., retrieval augmentation or multi-turn agent traces) where prompts grow over time.

Implications for safety evaluation and deployment. Overflip challenges a common implicit assumption in safety evaluation: that decisions remain stable as inputs are lengthened, and that "more context" yields monotonic protection. In practice, guardrails are increasingly composed with systems that concatenate retrieved documents, tool outputs, and conversation histories. This makes length-robustness a deployment concern, not merely an academic edge case. We therefore recommend that guardrail benchmarks and regressions include length stress tests and report stability-oriented measurements (e.g., flip rates, first-flip length, and attention-dispersion/entropy dynamics under lengthening) alongside accuracy.

7 Limitations

Dataset scope. Our evaluation covers 100 prompts across diverse domains, but larger-scale testing on production traffic distributions may reveal additional edge cases.

Model documentation gaps. Not all models publicly document their positional encoding schemes. We infer bucketed relative encodings based on observed behavior and architecture similarities, but cannot confirm exact implementations.

Mitigation evaluation. We focus on characterizing Overflip and its practical impact rather than exhaustively evaluating defenses. Future work should validate mitigation strategies under realistic deployment constraints and across a broader range of guardrails and long-input regimes.

8 Conclusion

We reveal *Overflip*: lengthening a prompt by simple repetition can flip lightweight guardrail predictions and enable MAL→BEN bypass. Across 9 guardrail models on 100 prompts, 5 exhibit such flips, and the instability is consistent with structural effects in long-input handling around the 512-token regime (e.g., positional bucketing). Overall, input length should be treated as an attack surface, motivating routine length-robust stress tests for guardrail deployment.

9 Ethical Considerations

Dual-use concerns. Overflip could be exploited to evade guardrail models, enabling adversarial actors to bypass content filters. We mitigate this risk by:

1. Coordinating responsible disclosure with model providers
2. Releasing defensive implementations (segmentation, calibration) alongside attack descriptions
3. Emphasizing that the vulnerability arises from architectural choices, not zero-day exploits

Societal impact. Improved guardrail robustness benefits all LLM users by reducing false negatives (missed adversarial prompts) and false positives (legitimate prompts incorrectly flagged). Our work aims to strengthen the safety ecosystem rather than undermine it.

Open science. We commit to open-sourcing evaluation code, datasets, and mitigation tools upon publication to accelerate community-wide adoption of length-robust guardrails.

10 Acknowledgements

LLMs were used for editorial purposes in this manuscript, and all outputs were inspected by the authors to ensure accuracy and originality. LLMs are not a core component of our methodology. As our experiments involve the validation and querying of LLMs, reproducibility may be constrained by their uncontrollability.

References

Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J. Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J. Hubinger, and 2 others. 2024. Many-shot jailbreaking. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. *arXiv preprint arXiv:2308.14508*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *ArXiv*, abs/2404.01318.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

DAIR.AI. 2025. Adversarial prompting in llms: Jailbreaking. Accessed: 2025-12-30.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

Zhiheng Huang and Peng Xu. 2020. Improve Transformer Models with Better Relative Position Embeddings. *arXiv preprint arXiv:2009.13658*.

IBM Research. 2024a. Granite-Guardian-HAP-125m toxicity classifier. Release Date: September 6th, 2024.

IBM Research. 2024b. Granite-Guardian-HAP-38m lightweight toxicity classifier. Release Date: September 6th, 2024.

Dror Ivry and Oran Nahum. 2025. Sentinel: Sota model to protect against prompt injections. *Preprint*, arXiv:2506.05446.

Madhur Jindal. 2024a. Jailbreak detector: Advanced ai security model.

Madhur Jindal. 2024b. Jailbreak detector large: Advanced ai security model.

Madhur Jindal. 2025. Jailbreak-detector-2-xl: Qwen2.5 chat adapter for ai security.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. 2024. Evaluating the instruction-following robustness of large language models to prompt injection. In *Proceedings of EMNLP*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024. Formalizing and benchmarking prompt injection attacks and defenses. In *USENIX Security Symposium*.

Meta. 2024. Llama-Prompt-Guard-2-22M classifier for prompt attacks.

Meta. 2025. Llama-Prompt-Guard-2-86M classifier for prompt attacks.

Norman Paulsen. 2025. Context is what you need: The maximum effective context window for real world limits of llms. *ArXiv*, abs/2509.21361.

Ofir Press, Noah A Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation.

ProtectAI.com. 2023. Fine-tuned deberta-v3 for prompt injection detection.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel and 1 others. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018a. Self-attention with relative position representations.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018b. Self-attention with relative position representations. In *NAACL*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. In *arXiv preprint arXiv:2104.09864*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. 2024. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*.

Xunguang Wang, Zhenlan Ji, Wenxuan Wang, Zongjie Li, Daoyuan Wu, and Shuai Wang. 2025. Sok: Evaluating jailbreak guardrails for large language models. *arXiv preprint arXiv:2506.10597*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, and et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. arXiv preprint arXiv:2412.13663. *arXiv preprint*.

Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2023. Benchmarking and defending against indirect prompt injection attacks on large language models. *arXiv preprint arXiv:2312.14197*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Aaron Zheng, Mansi Rana, and Andreas Stolcke. 2024. Lightweight safety guardrails using fine-tuned bert embeddings. *arXiv preprint arXiv:2411.14398*.

A Additional Results

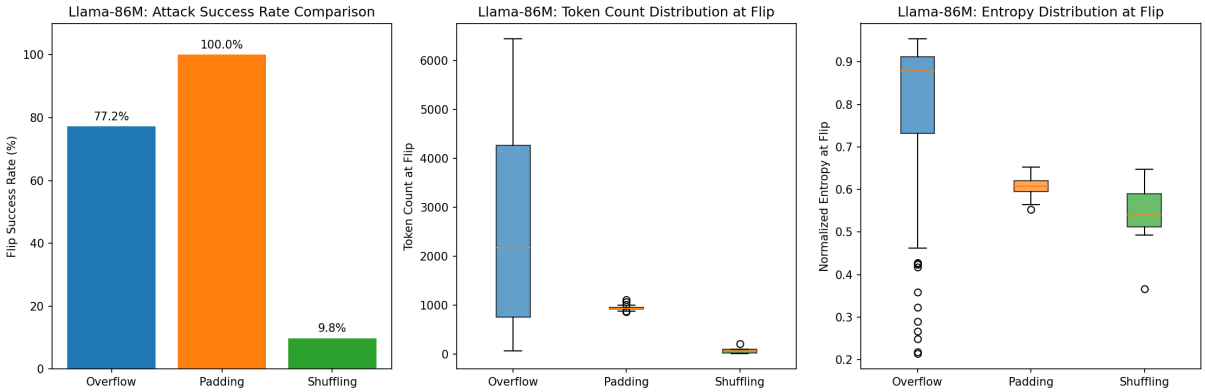

Figure 4: Comparison of three lengthening strategies on the LLaMA Prompt Guard 2 (86M) classifier. **Left:** MAL→BEN flip success rate for repetition-based Overflip (labeled "Overflow" in the plot), benign padding, and sentence shuffling. **Middle:** distribution of the token count at which the first flip occurs (boxplots over flipped prompts). **Right:** distribution of normalized CLS-attention entropy at the flip point, where higher values indicate more diffuse attention. Padding achieves the highest flip rate with flips occurring near ∼1k tokens, while Overflip flips occur at substantially longer lengths with more variable attention dispersion; shuffling rarely flips.

GPT-4.1 Mini

overflip ($n = 71$)		
Original → Attacked	Count	%
ac → ac	41	57.7%
rej → rej	19	26.8%
ac → rej	3	4.2%
ac → mis	2	2.8%
mis → mis	2	2.8%
rej → ac	2	2.8%
mis → ac	1	1.4%
rej → mis	1	1.4%

Padding ($n = 92$)		
Original → Attacked	Count	%
ac → ac	28	30.4%
ac → mis	23	25.0%
rej → rej	19	20.7%
ac → rej	12	13.0%
rej → mis	5	5.4%
mis → mis	2	2.2%
rej → ac	2	2.2%
mis → rej	1	1.1%

Shuffling ($n = 9$)		
Original → Attacked	Count	%
ac → ac	4	44.4%
rej → rej	4	44.4%
mis → mis	1	11.1%

Llama 3.1 8B

overflip ($n = 71$)		
Original → Attacked	Count	%
rej → rej	33	46.5%
ac → ac	13	18.3%
ac → rej	7	9.9%
mis → rej	5	7.0%
ac → mis	3	4.2%
mis → mis	3	4.2%
rej → ac	3	4.2%
mis → ac	2	2.8%
rej → mis	2	2.8%

Padding ($n = 92$)		
Original → Attacked	Count	%
rej → rej	43	46.7%
ac → mis	12	13.0%
rej → mis	11	12.0%
ac → rej	8	8.7%
mis → mis	7	7.6%
ac → ac	6	6.5%
mis → rej	3	3.3%
rej → ac	2	2.2%

Shuffling ($n = 9$)		
Original → Attacked	Count	%
rej → rej	3	33.3%
ac → rej	2	22.2%
rej → mis	2	22.2%
ac → mis	1	11.1%
mis → mis	1	11.1%

Table 4: Outcome transition counts (Original → Attacked) under three perturbations. Each block sums to 100% within a perturbation (overflip/Padding/Shuffling). For each type of attack, we observe three types of behaviors: acceptance (*ac*), rejection (*rej*), or misinterpretation (*mis*) the instructions in prompts. Note that the behavior of accaptance here does not necessarily mean that model execute the instructions explicitly, but mean that model implicitly accept the prompt and execute the part of the instructions. From this table, we can find that: overflip attack is the most effective one to retain the semantics; padding attack is the one that causes "misinterpretation" responses in both models to the most.