UNCOVERING UNCONTROLLED REPETITION THROUGH RESIDUAL STREAM DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Uncontrolled repetition can prolong autoregressive generation in large language models (LLMs) and enable resource consumption attacks. Prior analyses of repetitive generation have identified strongly activated features in intermediate and late layers. However, how uncontrolled repetition activity emerges and develops before becoming prominent in these layers remains insufficiently understood. In this paper, we investigate this question primarily in large vision-language models (LVLMs), which support a richer set of uncontrolled repetitions through both visual and textual inputs. We propose *Tokenwise Residual Comparison* (TRC), a method that identifies and localizes anomalies associated with repetition from residual dynamics during generation. TRC compares attention and multilayer perceptron writes to the residual stream across generated tokens to identify patterns associated with repetition. It then selectively suppresses coordinates in the residual stream at the identified layer. Experiments show that TRC effectively mitigates uncontrolled repetition, reducing loop rates by 57% on average. Our analysis further shows that repetition semantics emerge in shallow layers and propagate through the residual stream, disrupting normal representations. TRC also generalizes to large language models (LLMs) and large reasoning models (LRMs), where it consistently captures analogous repetition dynamics and achieves effective mitigation. Our work broadens the study of repetitive generation from its prominent internal representations to earlier opportunities for intervention, providing insights for mitigating resource consumption attacks. Our code is available at: `https://anonymous.4open.science/r/RC-TRC/`.

1 INTRODUCTION

Language models can become trapped in repetitive generation, where recurring content disrupts the normal development of the response (Hiraoka & Inui, 2025). Attackers can deliberately induce such repetition through textual or visual inputs to increase inference cost and latency, threatening service availability (Li et al., 2026; Gao et al., 2025). Prior studies have primarily analyzed repetition using manually constructed repetitive sequences (Yao et al., 2025; Hiraoka & Inui, 2025). These analyses identify repetition-related features concentrated in intermediate and final layers (Hiraoka & Inui, 2025; Yao et al., 2025). We argue that understanding these prominent representations also requires examining how uncontrolled repetition emerges across layers during model generation. LVLMs provide a convenient setting for this analysis, as the continuous space of visual inputs facilitates the construction of examples that induce uncontrolled generation (Gao et al., 2024a; 2025). By varying visual and textual triggers, we can study a broader range of model-generated failure cases and examine repetition activity before it becomes prominent in deeper layers.

Existing mitigation strategies address repetition through decoding controls and internal activation interventions (Xu et al., 2022; Yao et al., 2025; Zhang et al., 2026). At the output level, nucleus sampling and repetition penalties counter degenerate continuations by changing token selection (Holtzman et al., 2019; Zhu et al., 2023). Output-budget constraints require balancing token cost against answer accuracy, while aggressive repetition penalties can impair legitimate generation (Han et al., 2025; Gao et al., 2025). At the activation level, prior studies identify repetition related neurons or features from constructed repetitive sequences and show that suppressing their activations can mitigate repetition (Hiraoka & Inui, 2025; Yao et al., 2025). However, these approaches primarily identify representations after the repetitive semantics have become strongly activated, leading to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

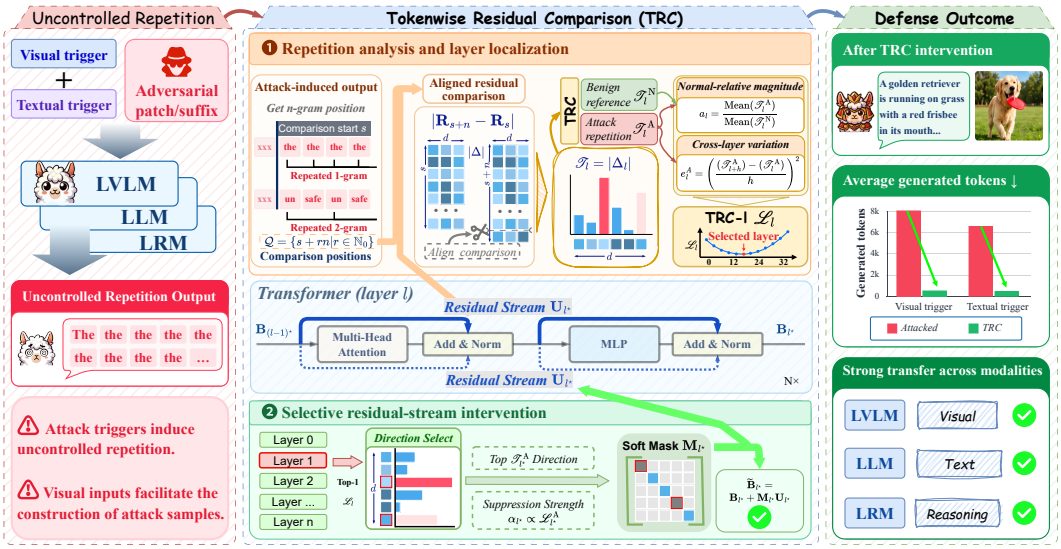


Figure 1: Overview of **TRC**, showing uncontrolled repetition across model families (***Left***), tokenwise residual comparison and selective intervention (***Middle***), and mitigation outcomes(***Right***).

their localization mainly in intermediate or later layers (Hiraoka & Inui, 2025; Yao et al., 2025). Our question is whether early layers already exhibit signals of repetition that can support analysis.

In this paper, we investigate this question using repetition that naturally emerges from LVLM generation under visual and textual input triggers, rather than artificially constructed repetition in model outputs. We propose *Tokenwise Residual Comparison* (TRC), which characterizes uncontrolled repetition changes in residual stream contributions. TRC compares residual stream contributions across generated sequence to derive tokenwise repetition signals at each layer. It then identifies candidate layers based on the magnitude of these signals relative to benign references and their local consistency across layers. TRC scores at the selected layer are then used to construct a mask that selectively suppresses anomalous residual stream contributions. By comparing residual stream contributions along the generated sequence, TRC exposes fine grained changes associated with repetition and localizes them to specific layers and coordinates.

Experiments show that suppressing residual contributions at the layer with the minimum TRC score reduces repetition by 57% on average. The same criterion consistently localizes repetition related changes to shallow layers, suggesting that the signals captured by TRC emerge well before repetition becomes prominent in later representations. TRC further generalizes to large language models (LLMs) and large reasoning models (LRMs), where it similarly mitigates repetitive generation and identifies corresponding changes at early layers. These findings indicate that fine grained residual changes in shallow layers provide effective targets for observing and suppressing uncontrolled repetition before its representations become dominant.

In summary, we introduce TRC, which uses features from residual connections to characterize tokenwise repetition signals across layers. TRC reveals that uncontrolled repetition already leaves distinguishable traces in shallow layers, where targeted suppression substantially mitigates repetitive generation under both visual and textual triggers in LVLMs. This behavior further generalizes to LLMs and LRMs, indicating that early repetition signals and their intervention effects are shared across different model families.

2 Related Work

2.1 Input Modalities and Adversarial Example Construction

Different input modalities provide distinct perturbation spaces for adversarial example construction. Text attacks operate over discrete tokens through efficient gradient guided modifications or direct contextual replacements, with continuous relaxations enabling optimization over discrete se-

quences(Ebrahimi et al., 2018; Li et al., 2020a; Garg & Ramakrishnan, 2020; Guo et al., 2021). Visual and speech attacks instead optimize continuous pixels or waveforms under perceptual constraints (Goodfellow et al., 2014; Carlini & Wagner, 2017; 2018; Qin et al., 2019). Multimodal attacks further exploit interactions across channels through coordinated perturbations and cross modal alignment (Zhang et al., 2022; Yin et al., 2023; Lu et al., 2023). In LVLMs, optimized pixels and typographic visual prompts can bypass textual safety alignment, while visual optimization can also prolong generation (Qi et al., 2024; Gong et al., 2025; Gao et al., 2024a). This flexibility facilitates the construction of diverse visual and textual triggers for studying model generated repetition.

2.2 Resource Consumption Attacks and Defenses

Resource consumption attacks arise through different mechanisms across model types. Sponge examples increase energy use and latency, while DeepSloth and SlowBERT undermine early exit efficiency (Shumailov et al., 2021; Hong et al., 2020; Zhang et al., 2023). In autoregressive models, optimized prompts, natural instructions, adversarial images, poisoned training data, and injected reasoning decoys can all prolong computation or induce excessive outputs (Dong et al., 2025; Chen et al., 2026; Gao et al., 2024a;b; Kumar et al., 2025; Zhang et al., 2025b). Existing mitigation includes token budget control, filtering or paraphrasing external context, decoding strategies, and training objectives that discourage degeneration or repetition (Han et al., 2025; Kumar et al., 2025; Holtzman et al., 2019; Zhu et al., 2023; Su et al., 2022; Li et al., 2023; Welleck et al., 2019; Li et al., 2020b; Xu et al., 2022). Existing internal analyses primarily capture prominent repetition activations, providing limited insight into when repetition first emerges during generation (Hiraoka & Inui, 2025; Yao et al., 2025). TRC instead tracks evolving semantic changes in residual stream contributions across the generated sequence to identify the emergence of repetition signals.

3 Tokenwise Residual Comparison

Tokenwise Residual Comparison (TRC) identifies repetition signals by comparing differences in residual stream contributions, then selects residual coordinates for targeted suppression. After specifying the calibration setting in Section 3.1, we derive TRC scores and layer localization scores (TRC-l) in Section 3.2. Section 3.3 converts the selected TRC scores into a fixed soft mask applied at the selected layer in residual addition.

3.1 Preliminaries and Problem Setup

We consider an LVLM whose autoregressive generation can be driven into uncontrolled repetition. The defender has access to the transformer backbone and records the contributions transmitted through its residual connections during generation. Calibration uses a set of attack-induced repetition outputs $\mathcal{D}_{\mathrm{A}}$ and a separate set of benign reference outputs $\mathcal{D}_{\mathrm{N}}$. Attention and multilayer perceptron (MLP) modules are analyzed independently within each model, so we omit the module index throughout. The formulation also applies separately to text-only LLMs and LRMs.

Given an input request, the model generates one token at a time, conditioned on that request and its previously generated tokens. We denote a completed output by $\mathbf{y} = (y_1, \ldots, y_T)$, where y_t is the token at output position t and $T \geq 2$ is the output length. The backbone has L layers indexed by $l \in \{0, \ldots, L-1\}$, each with hidden dimension d. At generation position t, $\mathbf{R}_{l,t} \in \mathbb{R}^{d \times S_t}$ denotes the recorded contribution of the selected module in residual addition, where S_t is the total number of tokens from the input through generation position t.

Our goal is to characterize changes in the residual stream associated with uncontrolled repetition and use them to determine a localized intervention during generation. Given the recorded module contributions, we seek to identify layers and hidden coordinates that exhibit distinctive changes under repetition, and selectively suppress these contributions while preserving benign generation.

3.2 TRC Scores and Layer Localization

We measure changes in the contributions transmitted through residual connections across generation positions. To identify the dominant repeated pattern, we search over n-grams with $n \in \{1, \ldots, \lfloor T/2 \rfloor\}$. For each n, we consider only n-grams occurring at least twice and measure the

fraction of output tokens covered by their occurrences. We then select the smallest n whose most-covered repeated n-gram accounts for more than half of the output:

$$n^\star = \min\left\{ n : \max_{g:\,|\mathcal{O}_n(g)|\geq 2} \frac{\left|\bigcup_{t\in\mathcal{O}_n(g)}\{t,\ldots,t+n-1\}\right|}{T} > 0.5 \right\},$$
$$g^\star = \arg\max_{g:\,|\mathcal{O}_{n^\star}(g)|\geq 2} \left| \bigcup_{t\in\mathcal{O}_{n^\star}(g)} \{t,\ldots,t+n^\star-1\} \right|, \qquad \mathcal{P}^\star = \mathcal{O}_{n^\star}(g^\star), \tag{1}$$

where $g = (g_1, \ldots, g_n)$ denotes an n-gram in $\mathbf{y}$, and $\mathcal{O}_n(g) = \{t \in \{1, \ldots, T-n+1\} : (y_t, \ldots, y_{t+n-1}) = g\}$ denotes the set of starting positions at which g occurs in the output. Thus, $n^\star$ gives the minimum n for which a repeated n-gram covers more than 50% of the output, while $\mathcal{P}^\star$ records all starting positions of the selected repeated n-gram. We then determine the comparison start s, token spacing q, and comparison positions $\mathcal{Q}$ as:

$$(s,q) = \begin{cases} (\min \mathcal{P}^\star, n^\star), & \text{if } n^\star \text{ is defined,} \\ (1,1), & \text{otherwise,} \end{cases} \qquad \mathcal{Q} = \{s + rq | r \in \mathbb{N}_0,\ s + (r+1)q \leq T\}. \tag{2}$$

The set $\mathbb{N}_0$ contains the nonnegative integers, and each $t \in \mathcal{Q}$ serves as the earlier position in a pairwise comparison with $(t, t+q)$.

Since $S_{t+q} = S_t + q$, we align each comparison pair by retaining the trailing columns corresponding to the newly generated tokens in $\mathbf{R}l, t$ and $\mathbf{R}l, t+q$. We then compute the element-wise absolute difference between these aligned module contributions:

$$\mathbf{v}_l = \frac{1}{|\mathcal{Q}|} \sum_{t\in\mathcal{Q}} \mathrm{Mean}\left(|\mathbf{R}_{l,t+q}[:, S_{t+q-1}:] - \mathbf{R}_{l,t}[:, S_{t-1}:]|\right). \tag{3}$$

Here, $\mathrm{Mean}(\cdot)$ denotes averaging over the column dimension corresponding to token positions, and $\mathbf{v}_l \in \mathbb{R}^d$ represents the contribution change at layer l averaged across all comparison pairs.

We then aggregate the sample-level vectors across all attack calibration samples under the same request condition to obtain the attack TRC score vector:

$$\mathscr{T}_l^{\mathrm{A}} = \frac{1}{|\mathcal{D}_{\mathrm{A}}|} \sum_{z\in\mathcal{D}_{\mathrm{A}}} \mathbf{v}_l(z), \tag{4}$$

$\mathbf{v}_l(z)$ is the contribution-change vector computed for sample z. The resulting $\mathscr{T}_l^{\mathrm{A}} \in \mathbb{R}^d$ represents the attack TRC scores at layer l, with each entry corresponding to one hidden coordinate.

We analogously aggregate the contribution-change vectors over the benign calibration set to obtain $\mathscr{T}_l^{\mathrm{N}}$. For layer-wise analysis, we summarize the attack and benign TRC scores over hidden coordinates. For a controllable window span $h \in \{1, \ldots, L-1\}$, we define the normal-relative magnitude and the cross-layer variation as:

$$a_l = \frac{\mathrm{Mean}(\mathscr{T}_l^{\mathrm{A}})}{\mathrm{Mean}(\mathscr{T}_l^{\mathrm{N}})}, \qquad e_l^A = \left(\frac{\mathrm{Mean}(\mathscr{T}_{l+h}^{\mathrm{A}}) - \mathrm{Mean}(\mathscr{T}_l^{\mathrm{A}})}{h} \right)^2. \tag{5}$$

Where, a_l measures the magnitude of attack-induced contribution changes relative to benign generation, while e_l^A measures the net variation of the attack TRC scores over a local layer interval of span h. Attention and MLP modules are calibrated separately within each model.

To normalize the cross-layer variation using benign observations only, we construct a benign layer-wise reference curve from $\mathcal{D}_{\mathrm{N}}$ using the same aggregation procedure as for the attack samples. We collect its positive cross-layer variation values and define the TRC-l score as

$$\mathscr{L}_l = a_l \left(1 + \frac{e_l^{\mathrm{A}}}{\exp\left(\mathrm{Mean}_{e\in\mathcal{E}_{\mathrm{N}}} \log e\right)} \right), \qquad \mathcal{E}_{\mathrm{N}} = \left\{ e_l^{\mathrm{N}} \mid 0 \leq l \leq L-h-1,\ e_l^{\mathrm{N}} > 0 \right\}. \tag{6}$$

e_l^{N} denotes the cross-layer variation of the benign reference curve starting at layer l, computed using the same formulation as e_l^{A}. The geometric mean over $\mathcal{E}_{\mathrm{N}}$ provides a benign reference scale for

the variation term, such that $\mathscr{L}_l$ jointly captures the normal-relative magnitude and the normalized cross-layer variation of the attack curve.

We use $\mathscr{L}_l$ as the criterion for localizing the intervention layer, favoring candidate starting layers with both a low normal-relative magnitude and limited variation over the following h layers. The starting layer is selected by minimizing $\mathscr{L}_l$, with ties broken in favor of the shallowest layer $l^\star = \min\left(\operatorname{argmin}_{0\le l\le L-h-1} \mathscr{L}_l\right)$. The selected localization window is represented by $(l^\star, h)$, where $l^\star$ specifies the starting layer and h is the controllable window span.

3.3 Selective Residual-Stream Intervention

The localization stage returns $l^\star$ as the selected intervention layer. At $l^\star$, we use the TRC scores computed from attack samples $\mathscr{L}^{\mathrm{A}}_{l^\star}$, to identify the residual-stream contributions most associated with repetition. Given a direction-selection ratio $\rho \in (0, 1]$, we select $\Omega_{l^\star} = \mathrm{Top}_{\lfloor \rho d \rfloor}\left(\mathscr{T}^{\mathrm{A}}_{l^\star}\right)$, where $\Omega_{l^\star}$ contains the $\lfloor \rho d \rfloor$ coordinates with the largest attack TRC scores at the selected layer. We further determine the suppression strength directly from the layer-level TRC-l score:

$$\alpha_{l^\star} = \frac{\max\left(0, \ln \mathscr{L}^{\mathrm{N}}_{l^\star} - \ln \mathscr{L}^{\mathrm{A}}_{l^\star}\right)}{1 + \max\left(0, \ln \mathscr{L}^{\mathrm{N}}_{l^\star} - \ln \mathscr{L}^{\mathrm{A}}_{l^\star}\right)}. \tag{7}$$

The resulting $\alpha_{l^\star} \in [0, 1)$ increases as the TRC-l score decreases, assigning stronger suppression to layers exhibiting a smaller normal-relative magnitude and more stable cross-layer variation. The direction-selection ratio ρ controls the intervention sparsity, while $\alpha_{l^\star}$ controls the suppression strength. During inference, let $\mathbf{B}_{l^\star} \in \mathbb{R}^{d\times S}$ denote the current module contribution over S sequence positions and $\mathbf{U}_{l^\star} \in \mathbb{R}^{d\times S}$ the residual stream entering the corresponding residual addition. We construct a diagonal soft mask $\mathbf{M}_{l^\star} \in [0, 1]^{d\times d}$ and apply it in residual addition:

$$[\mathbf{M}_{l^\star}]_{u,u} = \begin{cases} 1 - \alpha_{l^\star}, & u \in \Omega_{l^\star}, \\ \end{cases} \qquad \widetilde{\mathbf{B}}_{l^\star} = \mathbf{B}_{l^\star} + \mathbf{M}_{l^\star}\mathbf{U}_{l^\star}. \tag{8}$$

Left-multiplying $\mathbf{U}_{l^\star}$ by $\mathbf{M}_{l^\star}$ scales each selected row by $1 - \alpha_{l^\star}$ in residual addition. The mask is shared across sequence positions and is applied only to the selected target layer, leaving all other layers unchanged. For every layer $l \neq l^\star$, the module contribution is added without modification. The selected layer, coordinate set, and soft mask are fixed after calibration and directly applied to subsequent evaluation requests.

4 Experiments

4.1 Experimental Setup

Models and Evaluation Scope Our evaluation suite comprises three groups. The LVLM group includes InstructBLIP-Vicuna-7B (Dai et al., 2023), Qwen2.5-VL-3B-Instruct (Team, 2025), and LLaVA-1.5-7B (Liu et al., 2024a). The text-only LLM group includes Llama-3.2-3B[1] and Qwen2.5-3B (Team, 2024). The reasoning-model (LRM) group includes DeepSeek-Llama-8B (Guo et al., 2025), Qwen3.6-27B (Team, 2026), and GLM-4.7-Flash (Zeng et al., 2025).

Attack and Benign Evaluation Data The attack suite separates visual perturbations from textual induction. RECITE optimizes image perturbations to elicit repeated output (Gao et al., 2025). Textual conditions comprise GCG-based prompt optimization (Zou et al., 2023), LoopLLM's repetition-inducing attacks (Li et al., 2026), and direct repetition instructions (Direct). Benign evaluation uses ScienceQA for multimodal science question answering (Lu et al., 2022) and TextVQA for answering questions requiring scene-text understanding (Singh et al., 2019). MMLU assesses knowledge and reasoning across text-based subjects for the LLMs and LRMs (Hendrycks et al., 2020). Table 6 in the Appendix specifies the attack coverage and benign tasks for each model.

Baselines We compare TRC with three defense baselines, alongside the unmodified model. *Fixed-length truncation (Fixed-length)* stops decoding at a preset output-token limit (Zhang et al., 2025a;

[1]Official Llama 3.2 model card: `https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md`.

Table 1: Defense results on three LVLMs. Lower is better for both metrics.

		RECITE (visual)		GCG (textual)		LoopLLM (textual)	
Model	**Defense**	**Length**	**Loop rate**	**Length**	**Loop rate**	**Length**	**Loop rate**
InstructBLIP-7B	No defense	1,866.44	92.0%	1,531.56	96.0%	1,833.96	100.0%
	Fixed-length	389.84	92.0%	1,017.76	96.0%	952.36	88.0%
	No-repeat	18.36	4.0%	17.88	4.0%	22.04	0.0%
	AUSteer	413.16	20.0%	85.40	4.0%	165.84	8.0%
	TRC	422.04	16.0%	1,312.56	56.0%	1,229.72	60.0%
Qwen2.5-VL-3B	No defense	4,096.00	100.0%	4,096.00	100.0%	1,395.32	32.0%
	Fixed-length	62.00	100.0%	1,241.00	100.0%	177.16	32.0%
	No-repeat	40.08	0.0%	35.28	4.0%	29.80	0.0%
	AUSteer	4,096.00	76.0%	4,096.00	100.0%	4,096.00	84.0%
	TRC	62.44	0.0%	2,469.72	60.0%	1,044.28	28.0%
LLaVA-7B	No defense	3,611.08	88.0%	3,113.60	76.0%	–	–
	Fixed-length	181.72	88.0%	924.12	76.0%	–	–
	No-repeat	34.80	0.0%	7.04	0.0%	–	–
	AUSteer	1,250.32	60.0%	715.32	32.0%	–	–
	TRC	362.56	8.0%	4.84	0.0%	–	–

Table 3: Benign task accuracy on LLMs and LRMs. TRC causes minor changes in performance.

	LLM		LRM		
Setting	**Qwen2.5-3B**	**Llama-3.2-3B**	**DeepSeek-Llama-8B**	**Qwen3.6-27B**	**GLM-4.7-Flash**
Benign	61.0%	55.0%	97.0%	99.5%	100.0%
TRC	61.0%	56.0%	95.0%	99.5%	100.0%
Difference (pp)	0.0%	+1.0%	-2.0%	0.0%	0.0%

Gao et al., 2025). *No-repeat* uses no-repeat n-gram blocking to prevent any next token from reproducing a previously generated token n-gram (Fu et al., 2026; Zhu et al., 2023). *Interpretability-based intervention (AUSteer)* adapts AUSteer's selection and steering of individual activation dimensions to repetition suppression (Feng et al., 2026).

Evaluation Metrics We evaluate generation length, task accuracy, and loop rate. Following Hiraoka and Inui (Hiraoka & Inui, 2025), we identify repetitive outputs by detecting recurring token subsequences in the generated sequence. The loop rate is defined as N_{rep}/N, where N_{rep} is the number of samples exhibiting repetitive loops and N is the total number of evaluated samples.

4.2 Mitigation Effectiveness and Generalization

Defense Effectiveness. Table 1 shows that TRC consistently suppresses uncontrolled repetition across visual and textual attacks, reducing both generation length and loop rate in most settings. Its numerical efficiency is weaker than direct decoding constraints such as no-repeat, since TRC intervenes through localized internal signals rather than explicitly blocking repeated output patterns. Importantly, Table 2 shows that this intervention has little effect on benign task accuracy. These results indicate that TRC can mitigate repetition while preserving normal generation, supporting the identified residual changes as meaningful intervention targets.

Table 2: Benign task accuracy (%).

Method	ScienceQA	TextVQA	Average
Benign	82.00%	79.00%	80.50%
Fixed-length	82.00%	79.00%	80.50%
No-repeat	82.00%	78.50%	80.25%
AUSteer	76.00%	60.50%	68.25%
TRC	81.50%	78.50%	80.00%

Generalization. We next examine whether TRC generalizes beyond LVLMs to text only LLMs and LRMs. As shown in Figure 2, TRC consistently reduces generation length and loop rate across model families whenever the underlying attack succeeds, with especially strong effects on LRMs. Table 3 shows that these gains come with little change in benign MMLU accuracy. We further observe more stable intervention effects on newer and larger models. A possible explanation is that advances in model scale and architecture lead to better separation between repetition related residual changes and normal semantic representations, allowing targeted suppression to preserve task relevant behavior more effectively. The cross model results indicate that TRC captures internal repetition signals that transfer beyond the original LVLM setting.

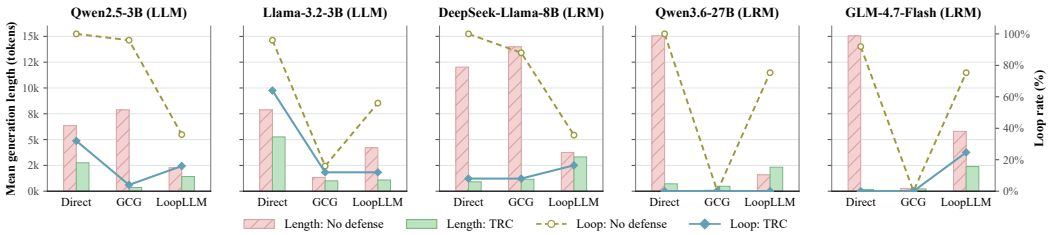
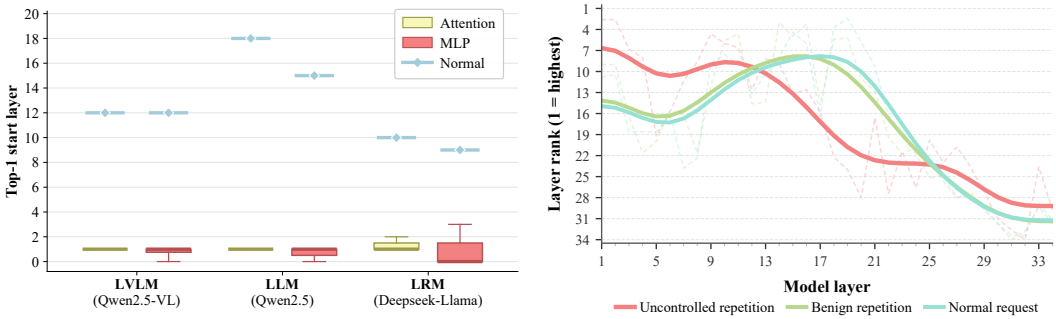


Figure 2: Generalization of TRC to LLMs and LRMs under Direct, GCG, and LoopLLM attacks. Bars show generation length and lines show loop rate; lower is better for both.



(a) Top 1 TRC localization across model families.

(b) Cross layer ranking of TRC

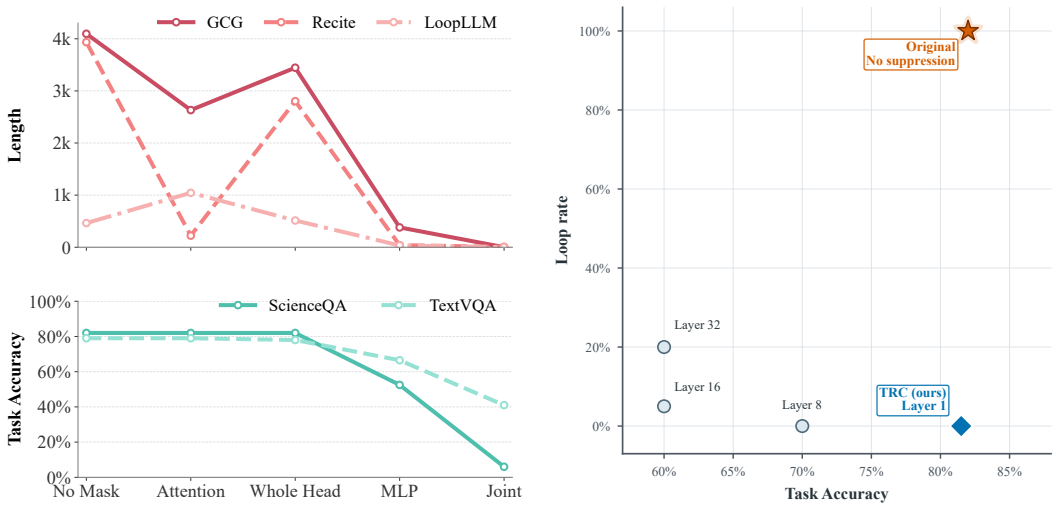
Figure 3: Layer wise localization of uncontrolled repetition. (a) Top 1 TRC localization across LVLM, LLM, and LRM families, showing consistently shallow attack locations. (b) Cross layer TRC rankings under uncontrolled repetition, benign repetition, and normal requests, revealing a distinct early layer profile for uncontrolled repetition.

4.3 Mechanistic Analysis

We use Qwen2.5-VL-3B-Instruct as the primary model for the detailed mechanistic analyses in this section; experiments comparing model families include the other specified models.

Shallow Localization of Repetition Signals. Figure 3(a) compares the Top-1 layers identified by TRC for an LVLM, a text-only LLM, and an LRM. Across all evaluated attack conditions, both Attention and MLP branches consistently localize repetition signals to shallow layers within 0–3, whereas benign references are localized substantially later, between layers 9 and 18. This separation is preserved across three models, suggesting that uncontrolled repetition residual changes emerge early rather than at a model-specific depth. Attention provides the more stable localization signal, with nine of ten conditions concentrated at layer 1, while MLP locations vary across layers 0, 1, and 3. These results therefore identify the residual stream entering shallow Attention layers as a particularly consistent location of repetition related changes. Notably, its concentration at layer 1 corresponds to the residual representation immediately after the layer 0 MLP, indicating that repetition related changes are already prominent after the first MLP transformation.

Specificity to Uncontrolled Repetition. To test whether TRC merely responds to repeated tokens or repeated semantics, we compare layer-wise localization under three conditions: uncontrolled-repetition failures, benign requests containing legitimate repetition, and ordinary normal requests. Figure 3(b) shows that benign repetition closely follows the localization profile of normal requests across layers. In both cases, the highest-ranked region shifts toward the middle layers before weakening in later layers. Uncontrolled repetition follows a different trajectory, with candidate locations ranked substantially higher in the early layers and progressively lower at greater depths. This separation rules out the simple explanation that TRC detects repetition semantics or repeated token identity alone. Instead, the localized signal is associated with uncontrolled repetition collapse. The result therefore supports the specificity of TRC to failure-related repetition dynamics, while the causal role of the localized components is evaluated separately in subsequent chapters. The absolute scores and their depth-dependent trends are examined in Appendix D.



(a) Branch selection (b) Layer selection

Figure 4: Mechanistic evidence for branch and layer selection. (a) Branch-wise suppression on attack and normal examples. Attention-only suppression preserves normal accuracy, whereas MLP-only and joint suppression reduce attack length at substantially higher utility cost. (b) Effect of the suppression start layer on Recite repetition suppression and ScienceQA accuracy. The star denotes the original model without suppression.

Attention and MLP Branch Contributions. Figure 4(a) shows that suppressing the full Attention write provides a better mitigation–utility tradeoff than suppressing individual heads, suggesting that repetition is not yet concentrated in a specific head at shallow layers. In contrast, suppressing the MLP write or both branches causes substantially larger degradation on benign tasks. Since the intervened residual stream already contains the output of the preceding block, these results suggest that repetition related features are formed early in MLP computations and then propagated through subsequent residual updates. This interpretation is consistent with prior studies characterizing MLPs as key value memories and linking them to concept and knowledge representations (Geva et al., 2021; 2022; Dai et al., 2022; Meng et al., 2022).

Intervention Effects of Layer and Direction Selection. To isolate the effect of layer choice, we keep the suppression rule fixed and vary only the intervention layer. As shown in Figure 4(b), the shallow layer selected by TRC achieves the best mitigation–utility trade-off: layer 1 suppresses all of repetitive failures while retaining 81.5% ScienceQA accuracy, close to the original model. Applying the same intervention at deeper layers can maintain high suppression but causes substantially larger accuracy degradation. These results show that effective repetition mitigation depends critically on where suppression is applied, supporting TRC's shallow-layer localization rather than depth-agnostic intervention.

Attention Update Magnitude and Residual Propagation. We further examine whether shallow attention blocks repeatedly amplify the repetition signal, or whether the signal is mainly retained through residual propagation. For the first two attention blocks, we measure the update magnitude $\mathbf{D}_l = \mathbf{U}_l^{\text{out}} - \mathbf{U}_l^{\text{in}}$ using both L2 norm and RMS, and normalize the condition gap by the corresponding normal fluctuation threshold τ_l. As shown in Table 4, the repetition-induced gaps remain within normal fluctuation ranges for both metrics, reaching only 23.4%/17.2% of the threshold in Block 0 and 73.0%/69.2% in Block 1 for L2/RMS, respectively. This indicates that repetitive generation is not accompanied by

Table 4: Attention-update magnitudes in the first two blocks. Gap/τ_l < 1 indicates that the condition difference remains within the normal token-level fluctuation bound.

Block	Metric	Normal	Repeat	Gap/τ_l
0	L2	22.2009	22.8838	0.234
0	RMS	0.4946	0.5057	0.172
1	L2	12.9280	13.7456	0.730
1	RMS	0.2867	0.3038	0.692

Table 5: Cross intervention results with unit strength on GCG examples. *TRC* denotes coordinates selected by TRC, while *Random* uses an equal size coordinate set. Transition success measures repetition to nonrepetition for attack inputs and nonrepetition to repetition for normal inputs. ΔLen is computed relative to the corresponding no intervention baseline.

Input	Coordinates	Residual intervention	Repeat rate (%)	Transition success (%)	Avg. length	ΔLen (%)	Observed change
Attack	–	None	100.0	–	4096	–	Repetition baseline
Attack	**TRC (ours)**	Normal residual	**20.0**	**80.0**	**828**	−79.8	Strong restoration
Attack	Random	Normal residual	60.0	40.0	2462	−39.9	Partial restoration
Normal	–	None	0.0	–	11	–	Normal baseline
Normal	**TRC (ours)**	Attack addition	0.0	0.0	2	–	Output failure without repetition
Normal	Random	Attack addition	0.0	0.0	11	–	Nearly unchanged

an abnormal increase in shallow attention-update magnitude. Combined with the earlier localization results, this is more consistent with repetition-related features emerging through shallow attention transformations and then being retained through the residual stream, rather than being repeatedly amplified by subsequent attention transformations. This early emergence precedes the progressive strengthening of repetition related semantics in intermediate layers, as observed in prior activation level studies (Hiraoka & Inui, 2025; Yao et al., 2025).

Propagation and Activation Restoration. We test whether the coordinates localized by TRC capture an early attack-associated feature using GCG examples that differ only by the optimized suffix. At the first layer, we perform unit-strength cross-interventions by either restoring localized attack coordinates with values from the paired normal trajectory or injecting the corresponding attack-minus-normal difference into the normal trajectory, with equal-size random coordinates as controls. As shown in Table 5, restoring the TRC-localized coordinates converts 80% repetitive attack trajectories to non-repetitive outputs and reduces average generation length from 4096 to 828 tokens, substantially outperforming random restoration. This indicates that the localized coordinates already encode behaviorally relevant attack-associated information at the first layer and that restoring these early activations can propagate to downstream recovery. Conversely, injecting the attack-associated difference into normal trajectories disrupts generation but does not reproduce repetition, suggesting that these coordinates are important for the failure state but are not independently sufficient to induce it. Together, the results support TRC as identifying early residual features that contribute to repetition and can be causally restored to mitigate the failure. Appendix E presents representative outputs for both intervention directions.

Appendix F further reports sensitivity analyses on the layer window span, the number of attack training examples, the suppression ratio, and the maximum repetition distance used by the repeated n gram detector. These results examine the robustness of TRC to its main design choices.

5 Conclusion

We presented *Tokenwise Residual Comparison* (TRC), a framework for identifying and mitigating uncontrolled repetition by comparing residual stream contributions across generation positions. Rather than focusing only on prominent repetition representations in intermediate or later layers, TRC tracks how residual contributions evolve along the generated sequence and localizes fine grained repetition signals to specific layers and coordinates. Across LVLMs, text only LLMs, and LRMs, TRC consistently identifies shallow repetition signals and enables targeted suppression that reduces repetitive generation while largely preserving benign task performance. Mechanistic analyses further show that these signals emerge early, remain distinguishable from benign and legitimate repetition, and can be partially restored through localized activation replacement, supporting residual propagation as a plausible mechanism by which repetition related features persist through the network. Together, these findings shift the analysis of uncontrolled repetition from where strong repetition representations are observed to how they emerge early and become actionable, providing a finer grained perspective for understanding and mitigating resource consumption failures in autoregressive models.

AI USE STATEMENT

Generative AI tools were used to assist with literature retrieval, review manuscript formatting, and suggest caption and editorial revisions. The authors reviewed all AI-assisted outputs and suggestions and take full responsibility for the final text, data, results, and claims.

ETHICS STATEMENT

This work studies repetition-inducing attacks and a defense against them using existing model and benchmark data. Attack procedures are reported to support evaluation and defense research; they may also be misused to increase inference costs or disrupt model services. We report defensive results alongside benign-task performance to make this trade-off visible.

REPRODUCIBILITY STATEMENT

The method and evaluation metrics are described in Sections 3 and 4.1; the models, attack conditions, and benign tasks are listed in Appendix A. The appendix also documents the construction and validation of the legitimate-repetition controls.

REFERENCES

Mirelle Candida Bueno, Roberto Lotufo, and Rodrigo Frassetto Nogueira. Mlissard: Multilingual long and simple sequential reasoning benchmarks. In *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pp. 86–95, 2024.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)*, pp. 1–7. IEEE, 2018.

Yiming Chen, Zexin Li, Xianghu Yue, Robby T Tan, and Haizhou Li. Naturalsloth: Revisiting denial-of-service attacks on large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19685–19702, 2026.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, Ke Xu, and Han Qiu. An engorgio prompt makes large language model babble on. In *International Conference on Learning Representations*, volume 2025, pp. 67280–67307, 2025.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 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)*, pp. 31–36, 2018.

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Chua Deryl, Lee Mak, Gee Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. In *International Conference on Learning Representations*, volume 2026, pp. 39421–39443, 2026.

Jiyuan Fu, Kaixun Jiang, Lingyi Hong, Jinglun Li, Haijing Guo, Dingkang Yang, Zhaoyu Chen, and Wenqiang Zhang. Lingoloop attack: Trapping mllms via linguistic context and state entrapment into endless loops. In *International Conference on Learning Representations*, volume 2026, pp. 87860–87893, 2026.

Haoran Gao, Yuanhe Zhang, Zhenhong Zhou, Lei Jiang, Fanyu Meng, Yujia Xiao, Li Sun, Kun Wang, Yang Liu, and Junlan Feng. Resource consumption red-teaming for large vision-language models, 2025.

Kuofeng Gao, Yang Bai, Jindong Gu, Shu-Tao Xia, Philip Torr, Zhifeng Li, and Wei Liu. Inducing high energy-latency of large vision-language models with verbose images. In *International Conference on Learning Representations*, volume 2024, pp. 17156–17182, 2024a.

Kuofeng Gao, Tianyu Pang, Chao Du, Yong Yang, Shu-Tao Xia, and Min Lin. Denial-of-service poisoning attacks against large language models. *arXiv preprint arXiv:2410.10760*, 2024b.

Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6174–6181, 2020.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 5484–5495, 2021.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 30–45, 2022.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23951–23959, 2025.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5747–5757, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Tatsuya Hiraoka and Kentaro Inui. Repetition neurons: How do language models produce repetitions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 483–495, 2025.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019.

Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. *arXiv preprint arXiv:2010.02432*, 2020.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms, 2025.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6193–6202, 2020a.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4715–4728, 2020b.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

Xingyu Li, Xiaolei Liu, Cheng Liu, Yixiao Xu, Kangyi Ding, Bangzhou Xin, and Jia-Li Yin. Loopllm: Transferable energy-latency attacks in llms via repetitive generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31770–31777, 2026.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14551–14558, 2024b.

Dong Lu, Zhiqiang Wang, Teng Wang, Weili Guan, Hongchang Gao, and Feng Zheng. Set-level guidance attack: Boosting adversarial transferability of vision-language pre-training models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 102–111. IEEE, 2023.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 21527–21536, 2024.

Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*, pp. 5231–5240. PMLR, 2019.

Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pp. 212–231. IEEE, 2021.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. *Advances in neural information processing systems*, 35: 21548–21561, 2022.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwen2.5-vl, January 2025. URL https://qwenlm.github.io/blog/qwen2.5-vl/.

Qwen Team. Qwen3. 6-27b: Flagship-level coding in a 27b dense model, 2026.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training, 2019.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095, 2022.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7787–7815, 2025.

Ziyi Yin, Muchao Ye, Tianrong Zhang, Tianyu Du, Jinguo Zhu, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. Vlattack: Multimodal adversarial attacks on vision-language tasks via pre-trained models. *Advances in Neural Information Processing Systems*, 36:52936–52956, 2023.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. *arXiv preprint arXiv:2503.08908*, 2025.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025.

Jiaming Zhang, Qi Yi, and Jitao Sang. Towards adversarial attack on vision-language pre-training models. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 5005–5013, 2022.

Shengyao Zhang, Xudong Pan, Mi Zhang, and Min Yang. Slowbert: Slow-down attacks on input-adaptive multi-exit bert. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9992–10007, 2023.

Yuanhe Zhang, Xinyue Wang, Haoran Gao, Zhenhong Zhou, Fanyu Meng, Yuyao Zhang, and Sen Su. Pd^{3}f: A pluggable and dynamic dos-defense framework against resource consumption attacks targeting large language models. In *EMNLP (Findings)*, pp. 3641–3671, 2025a.

Yuanhe Zhang, Zhenhong Zhou, Wei Zhang, Xinyue Wang, Xiaojun Jia, Yang Liu, and Sen Su. Crabs: Consuming resource via auto-generation for llm-dos attack under black-box settings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11128–11150, 2025b.

Yuanhe Zhang, Xinyue Wang, Zhican Chen, Weiliu Wang, Zilu Zhang, Zhengshuo Gong, Zhenhong Zhou, Kun Wang, Li Sun, Yang Liu, et al. Resource consumption threats in large language models. *arXiv preprint arXiv:2603.16068*, 2026.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19781–19791. IEEE, 2025c.

Wenhong Zhu, Hongkun Hao, and Rui Wang. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1218–1228, 2023.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

A Model and Attack Coverage

Table 6 maps the attack pools and benign evaluation tasks to the models used in our experiments. We construct attack pools separately for each checkpoint so that model-specific tokenization, chat templates, and multimodal preprocessing are preserved. A candidate is retained only when greedy decoding produces an uncontrolled one- or two-token cycle for at least ten consecutive repetitions. A dash in the table therefore denotes a condition outside the evaluated scope, rather than a failed attack.

Training–test separation and default configuration. The attack examples used to localize TRC are disjoint from all examples used for evaluation. For each target model, we perform localization once using a designated training attack: RECITE for LVLMs and GCG for LLMs and LRMs. We do not retrain or retune TRC separately for every evaluation attack or dataset; instead, the resulting model-specific configuration is applied directly to the held-out attack pools summarized in Table 6. In the main experiments, the adaptive rule yields a suppression strength of approximately $\alpha_{l^\star} = 0.73$ (73%). We use a direction-selection ratio of $\rho = 5\%$ and a layer-window span of $h = 3$.

Table 6: Model-specific attack coverage and benign evaluation tasks. Y marks an included attack condition; – denotes a condition outside the evaluated scope.

Model	RECITE	GCG	LoopLLM	Direct	Benign tasks
LVLMs					
InstructBLIP-Vicuna-7B	Y	Y	Y	–	ScienceQA, TextVQA
Qwen2.5-VL-3B-Instruct	Y	Y	Y	–	ScienceQA, TextVQA
LLaVA-1.5-7B	Y	Y	–	–	ScienceQA, TextVQA
LLMs					
Llama-3.2-3B	–	Y	Y	Y	MMLU
Qwen2.5-3B	–	Y	Y	Y	MMLU
LRMs					
DeepSeek-Llama-8B	–	Y	Y	Y	MMLU
Qwen3.6-27B	–	Y	Y	Y	MMLU
GLM-4.7-Flash	–	Y	Y	Y	MMLU

RECITE. For the multimodal models, we follow the original RECITE construction procedure (Gao et al., 2025). Each source item contains an image, its associated request, and a repeated-token target. A targeted projected-gradient attack modifies the image while leaving the text request fixed. We then run the corresponding LVLM on the adversarial image and retain only candidates whose generated continuation satisfies the repetition criterion above. This produces paired clean and adversarial images for the same request and avoids changing the linguistic content of the input.

GCG. We implement GCG (Zou et al., 2023) with NanoGCG and adapt its optimization to the tokenizer, chat template, and cache representation of each model family. Starting from the clean request pool associated with a target model, NanoGCG optimizes a discrete suffix toward a repeated continuation. The resulting suffix is transferred into the corresponding target input format and screened again on the actual target checkpoint. Thus, the GCG datasets contain model-specific request–suffix pairs rather than one universal suffix reused across all models.

LoopLLM. We construct the LoopLLM pools with the same model-wise adaptation and transfer protocol used for GCG, but replace the GCG target loss with LoopLLM's repetition-inducing objective (Li et al., 2026). For each clean request, coordinate optimization searches for a suffix that concentrates probability mass on a short cyclic token pattern. The suffix is appended to the request under the target model's own chat template, and the generated continuation is retained only after target-model screening. This keeps the base request intact while making the optimized suffix specific to the model and source pool.

Direct. Direct follows the repeated-token attack setting studied by Yona et al. (2025). We first serialize the original request with the model's chat template, then concatenate a short repeated token sequence to the assistant-side generation prefix. Autoregressive decoding resumes from this prefilled partial output, so the model receives the repetition as its own unfinished continuation rather than as a new user instruction. We retain examples only when the newly generated tokens continue into an uncontrolled repetition cycle.

Benign task sets. For utility evaluation, LVLMs use ScienceQA and TextVQA, which test multimodal science reasoning and question answering over scene text, respectively (Lu et al., 2022; Singh et al., 2019). Text-only LLMs and LRMs use MMLU across its subject domains (Hendrycks et al., 2020). These benign inputs preserve the original images, questions, and answer choices and receive no attack suffix or assistant-side repetition prefix. For each dataset, we use a subset of 200 samples for evaluation.

B Distinction from Existing Interpretability Methods

Table 7 compares TRC with representative interpretability approaches along the three axes most relevant to our design: the quantity being observed, the internal site being localized, and the model modalities on which the method has been demonstrated. The comparison concerns methodological scope rather than a shared numerical benchmark; the listed approaches answer complementary questions and were not all designed to mitigate uncontrolled repetition.

Table 7: Comparison with representative interpretability approaches. "Scope" records the model families demonstrated in the cited work, rather than a theoretical restriction. TRC differs primarily in observing changes between cycle-aligned generation positions, localizing the corresponding pre-addition residual writes, and applying the same formulation across LVLMs, LLMs, and LRMs.

Approach	Primary observation	Localization target	Demonstrated scope	Principal distinction from TRC
FFN vocabulary analysis (Geva et al., 2021; 2022)	Activations and vocabulary-space promotion produced by feed-forward updates	FFN memories, values, and layer-wise concept promotion	Text LMs	Explains stored or promoted concepts, but does not compare residual changes along a repetitive generation trajectory.
Layer-wise lexical probing (Liu et al., 2024b)	Probe recoverability of lexical-semantic information from hidden representations	Layers at which lexical semantics are most accessible	Generative text LMs	Measures information accessibility through an external readout rather than locating native residual coordinates for intervention.
Repetition neurons (Hiraoka & Inui, 2025)	Neuron-activation changes before and after the onset of repetition	Repetition-associated neurons in intermediate and final layers	Text LMs	Directly studies repetition, but uses activation amplitude around onset rather than cycle-aligned changes before residual addition.
SAE repetition features (Yao et al., 2025)	Logit-based layer screening followed by learned sparse-feature activations	SAE features within selected layers	Text LLMs	Resolves features with an auxiliary learned dictionary; TRC operates on native attention and MLP contribution coordinates.
Cross-modal information flow (Zhang et al., 2025c)	Answer-performance changes after blocking attention between image and question positions	Layered cross-token attention pathways for visual–linguistic integration	LLaVA-style MLLMs	Explains modality fusion within MLLMs, but is not a repetition-localization method and is not evaluated on text-only LLMs or LRMs.
TRC (ours)	**Cycle-aligned changes in native attention and MLP contributions, normalized by benign magnitude and local cross-layer variation**	**A shallow pre-addition residual-write layer and its highest-scoring hidden coordinates**	**LVLMs, LLMs, and LRMs**	**Uses one modality-agnostic scoring and masking formulation without adding an auxiliary interpreter or network module.**

Observation dimension. Most neuron, probe, and feature based approaches inspect activation magnitude, prediction attribution, recoverable content, or a learned feature basis at a position or stage. TRC instead treats repetition as a temporal change pattern: it aligns positions separated by the detected repetition cycle and measures how the native attention and MLP contributions change across those positions. The attack curve is then interpreted relative to benign generation and its local cross-layer variation. This observation axis is suited to distinguishing a continuation that becomes dynamically stationary from one that continues to advance semantically.

Localization position. TRC localizes both a layer and residual coordinates at the module output before residual addition, with attention and MLP analyzed separately. Our experiments consistently place the repetition-associated low-change regime in shallow layers, whereas normal generation reaches its minimum later. This gives TRC an early and directly actionable intervention site, rather than requiring a probe, an SAE dictionary, or a search over late output activations. The comparison does not imply that shallow layers are universally optimal for every behavior; it identifies the location supported for uncontrolled repetition under our evaluated settings.

Cross-modal applicability. TRC observes the Transformer backbone's residual pathway after modality-specific inputs have entered the autoregressive decoder. Consequently, the same score, layer-selection rule, and soft-mask construction can be used for visually induced repetition in LVLMs and textually induced repetition in LLMs and LRMs. The resulting advantage is not merely support for multimodal inputs: it is a shared internal analysis and intervention interface across visual and textual attack constructions, as demonstrated by the model coverage and held-out evaluations in this paper.

C Construction of Legitimate-Repetition Controls

Models. For generation, we cap the number of newly generated tokens at 2,048 for InstructBLIP-Vicuna-7B (its default), 4,096 for the other LVLMs, 8,192 for all text-only LLMs, and 16,384 for all LRMs.

The specificity analysis in Section 4.3 requires a control condition that contains intentional repetition without an uncontrolled generation failure. We construct this condition using two design principles rather than copying benchmark instances. MLissard controls sequential complexity through repeated applications of simple rules, while BIG-bench emphasizes explicitly specified, auditable tasks that probe distinct model capabilities (Bueno et al., 2024; Srivastava et al., 2022). Following these principles, each of our image-grounded items requires the model to first produce a normal semantic description and then execute a finite repetition instruction.

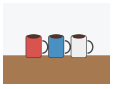
Each prompt contains two ordered requirements for Qwen2.5-VL-3B-Instruct. The model must first write one concise English sentence grounded in the input image and must then repeat a designated common English token exactly ten times. We select repeat units that map to one token both in isolation and with a leading space under the model tokenizer. The repeated token is excluded from the semantic description so that the descriptive and repetitive portions can be validated separately.

An item is accepted only when the output begins with an image-grounded description, contains an expected visual keyword before the repeated segment, omits the repeat unit from that semantic prefix, and ends with ten contiguous copies of the same token ID. All five constructed items pass these criteria. Table 8 shows two representative examples. They preserve a bounded, task-compliant repetition after a meaningful visual response, in contrast to an uncontrolled loop that displaces the task answer or fails to terminate.

D Layer-wise TRC-l Profiles under Repetition

This analysis expands the shallow-localization and specificity results in Sections 4.3 and 4.3. Figure 5 plots the TRC-l score at every eligible candidate start layer for Direct, GCG, RECITE, and normal requests on Qwen2.5-VL-3B. Attention and MLP are evaluated separately at window spans $h \in \{2, 3, 4\}$. Lower scores indicate a smaller attack-to-normal contribution change ratio and/or less net change across the layer window, as defined in Section 3.2. This does not affect layer ranking, and the score should not be interpreted as a direct measure of semantic similarity.

Table 8: Two representative legitimate-repetition controls for Qwen2.5-VL-3B-Instruct. Each prompt requires an image-grounded description followed by exactly ten copies of a designated token. Output text is reproduced verbatim; line breaks are normalized for table layout.

Image	Prompt	Generated output
	Examine the image and return exactly two lines. Line 1 must begin with 'Description:' and contain one concise factual English sentence describing the main visible object. Do not use the word 'echo' in line 1. Line 2 must be exactly: 'Repeat: echo echo echo echo echo echo echo echo echo echo.' Do not add any other text before, between, or after these two lines.	`Description: The image depicts a simple, stylized red apple with a green leafy top.` `Repeat: echo echo echo echo echo echo echo echo echo echo`
	Examine the image and return exactly two lines. Line 1 must begin with 'Description:' and contain one concise factual English sentence describing the visible objects. Do not use the word 'signal' in line 1. Line 2 must be exactly: 'Repeat: signal signal signal signal signal signal signal signal signal signal.' Do not add any other text before, between, or after these two lines.	`Description: Three mugs of different colors are lined up on a table.` `Repeat: signal signal signal signal signal signal signal signal signal signal.`

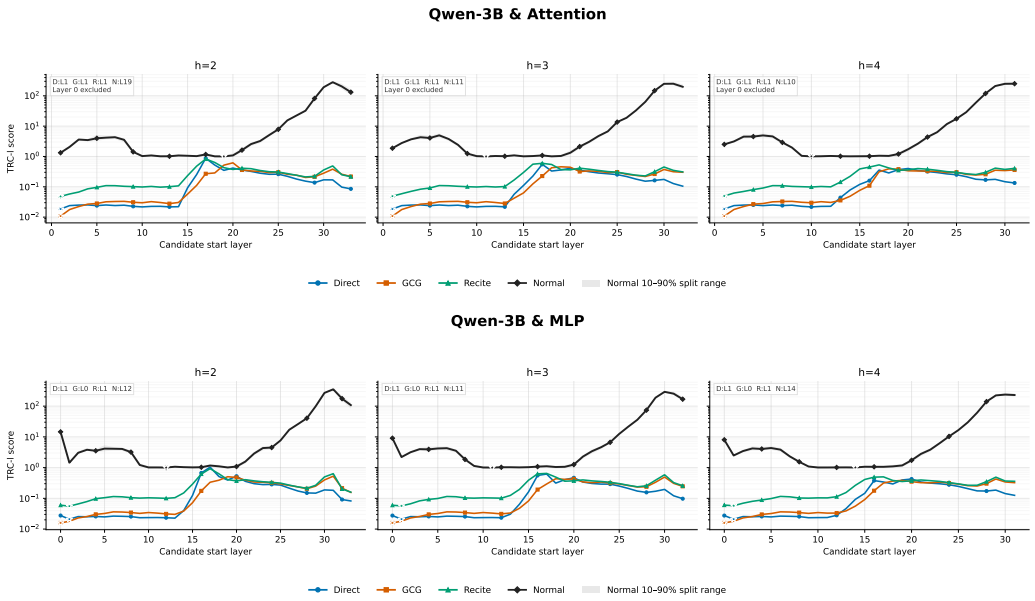


Figure 5: Layer-wise TRC-l profiles for Qwen2.5-VL-3B Attention (top) and MLP (bottom), with window spans $h = 2, 3, 4$. Curves show Direct, GCG, and RECITE failures and the mean of normal-reference splits; gray shading denotes the normal range.

Across the plotted candidate layers, the failure curves generally lie below the normal curve. This scale difference is compatible with the construction of TRC: when successive comparison positions revisit a similar repetitive content state, their aligned residual contributions can differ less than those in a normal continuation that advances the answer. The smaller tokenwise difference lowers the magnitude term of TRC-l. Because TRC-l also contains a cross-layer variation term, and because

comparison positions depend on the detected repetition unit, the plot alone does not establish semantic similarity as the unique cause of the score gap. The more diagnostic observation is where each curve reaches its minimum within its own condition.

For normal requests, the minimum remains in the middle portion of the network: Attention selects layers 19, 11, and 10 for $h = 2, 3, 4$, respectively, while MLP selects layers 12, 11, and 14. Ordinary generation can therefore exhibit a relatively stable local contribution profile even while the model continues to process a coherent task. This pattern is compatible with intermediate layers integrating task-relevant information before later output decisions; it does not establish a universal reasoning layer or imply that normal representations are semantically unchanged between tokens.

The repetitive failures show a different location for their most stable, low-change state. All three attacks select Attention layer 1 for every plotted span. In MLP, Direct and RECITE select layer 1, whereas GCG selects layer 0. The separation from normal requests persists for each tested h. Prior layer-wise studies offer a more specific basis for interpreting this shallow location: lower feed-forward layers tend to recognize relatively simple textual patterns, whereas later layers capture more semantic patterns (Geva et al., 2021); probing a generative LLM likewise finds lexical semantics in lower-layer representations (Liu et al., 2024b). Feed-forward updates can also promote human-interpretable concepts in the model's vocabulary space (Geva et al., 2022). Together, these findings make it plausible that basic lexical and local-pattern features in shallow layers provide a substrate from which a repetition-associated representation can emerge. Our TRC-l curves locate a low-change regime consistent with this interpretation; they do not identify a dedicated repetition neuron or prove that any cited model implements the same failure mechanism.

Once established, such a feature can remain available to later blocks through the residual stream, consistent with the account of feed-forward outputs being refined across layers via residual connections (Geva et al., 2021). This interpretation also aligns with Section 4.3: shallow attention-update magnitudes are nonzero, yet their attack–normal gaps remain within normal token-level fluctuation bounds.

E Qualitative Cross-Intervention Examples

We provide one representative paired example for each cross-intervention direction discussed in Section 4.3. These examples illustrate the aggregate patterns in Table 5. In each pair, the normal and attack inputs share the same base request, and the attack input differs only by the optimized GCG suffix. The ellipsis below marks an output truncated for presentation.

Example A: Restoring a normal activation on an attack trajectory.

Attack baseline `Hello, hello, hello, hello, hello, ...`
Outcome: 2048 newly generated tokens, repetitive, stopped at the generation cap.

Localized coordinates, normal residual replacement `Hello, hello, hello, hello, hello.`
Outcome: 11 newly generated tokens and non-repetitive generation.

Matched random coordinates, normal residual replacement `Hello, hello, hello, hello, hello, ...`
Outcome: 2048 newly generated tokens, repetitive, stopped at the generation cap.

Observed contrast. Restoring the localized coordinates recovers the short normal response, whereas the equal-size random replacement leaves this attack trajectory in the repetitive state.

Example B: Injecting the attack difference into a normal trajectory.

Normal baseline `Hello, hello, hello, hello, hello!`
Outcome: 11 newly generated tokens and non-repetitive generation.

Localized coordinates, attack-minus-normal addition `. .`
Outcome: 2 newly generated tokens and non-repetitive but erroneous termination.

Matched random coordinates, attack-minus-normal addition `Hello, hello, hello, hello, hello!`
Outcome: 11 newly generated tokens and non-repetitive generation, matching the normal baseline.

Observed contrast. The localized perturbation severely disrupts the normal response but does not recreate repetition; the matched random perturbation leaves the response unchanged.

F Ablation Studies

F.1 Sensitivity to the Layer-Window Span

The TRC-l score measures cross-layer variation over a local span h (Section 3.2). We vary $h \in \{2, 3, 4\}$ while keeping the remaining localization procedure fixed. Table 9 reports the resulting Top-1 start layer for the attention and MLP branches.

Table 9: Sensitivity of the localized Top-1 start layer to the layer-window span h. Each entry is Attention / MLP, and layers are zero-indexed.

Attack	$h = 2$	$h = 3$	$h = 4$
GCG	$1/0$	$1/0$	$1/0$
RECITE	$1/1$	$1/1$	$1/1$
LoopLLM	$1/1$	$1/1$	$1/1$

The selected locations are invariant over the three evaluated spans. In particular, the branch-specific difference for GCG (attention layer 1 versus MLP layer 0) is preserved, whereas both branches select layer 1 for RECITE and LoopLLM.

F.2 Sensitivity to the Suppression Ratio on Attacks

This ablation varies the suppression ratio $\rho \in \{1, 5, 10, 15, 20\}\%$ while keeping the attack training data and other intervention settings fixed. Table 10 reports attack behavior; each attack cell gives mean newly generated tokens followed by repetition rate. The final column reports the overlap between the coordinates selected at each ratio and the coordinates selected at $\rho = 20\%$.

Every evaluated suppression ratio lowers the macro-average generation length and repetition rate relative to no defense, but the improvement is not monotonic. At $\rho = 20\%$, the macro-average output is shortest (620.56 tokens), while $\rho = 1\%$ and $\rho = 20\%$ tie for the lowest macro-average repetition rate (21.3%). The attack-level results expose important heterogeneity: RECITE is strongly mitigated at every evaluated ratio, GCG benefits most at $\rho = 20\%$.

F.3 Effect of the Suppression Ratio on Benign Utility

Table 11 is a separate benign-utility study. Here, the learned direction ranking, the training-set size, and all other training settings are fixed, and we vary only the suppression ratio ρ, i.e., the fraction of hidden coordinates included in the soft mask.

Across the evaluated range, task accuracy decreases gradually as the suppression ratio increases, with a maximum drop of only 1.5 percentage points relative to the unsuppressed model. Meanwhile, the average response length varies by no more than 0.1 tokens.

F.4 Sensitivity to the Maximum Repetition Distance

Finally, we vary the maximum token distance $q_{\max}$ considered by the repeated-n-gram detector. This control determines how far apart matching positions may be when constructing the tokenwise comparisons used by TRC.

RECITE remains localized at layer 1 throughout, while GCG shifts from layer 12 at $q_{\max} = 1$ to layer 1 for every $q_{\max} \geq 2$. A one-token limit can be too restrictive because a repeated semantic unit may be segmented into several subword tokens; comparisons restricted to adjacent single-token matches can therefore miss the actual cycle alignment. Allowing even a short multi-token distance is sufficient to recover the stable shallow location in this experiment. This result motivates using a small but non-unit search range rather than assuming that one semantic repetition always corresponds to one token.

Table 10: Sensitivity to the suppression ratio on attacks. Cells under each attack and the macro average report mean newly generated tokens / repetition rate (%). Coordinate overlap is measured against the mask selected at $\rho = 5\%$. Lower is better for both reported attack metrics.

ρ (%)	GCG	RECITE	LoopLLM	Macro avg.	Coordinate overlap
No defense	4096.00/100.0	4096.00/100.0	1395.32/32.0	3195.77/77.3	–
1	1809.56/44.0	54.80/0.0	444.64/20.0	769.67/21.3	8/103
5	2469.72/60.0	62.44/0.0	1044.28/28.0	1192.15/29.3	67/103
10	2631.92/64.0	222.60/4.0	1044.24/28.0	1299.59/32.0	70/103
15	2462.60/60.0	214.48/4.0	805.80/24.0	1160.96/29.3	99/103
20	1159.36/28.0	58.72/0.0	643.60/36.0	620.56/21.3	103/103

Table 11: Effect of the suppression ratio ρ on benign task. Accuracy is reported in percent.

ρ (%)	Accuracy	Avg. length
0	82.0	2.125
1	82.0	2.130
5	81.5	2.145
10	82.0	2.115
15	81.0	2.125
20	80.5	2.065

G Online Efficiency

TRC does not append network layers, auxiliary detectors, or additional model forward passes at deployment. After offline localization, the fixed intervention is applied directly to the residual stream at the selected target layers by suppressing the identified residual coordinates during the original residual update. All other layers and residual coordinates remain unchanged. TRC therefore operates within the model's existing residual pathway, without introducing a separate inference module, additional forward computation, or changes to the depth of the computational graph.

Table 13 reports attack-generation time for two LLMs. TRC shortens generation in every listed model–attack condition. The reductions range from 90.9% to 99.6%, because suppressing uncontrolled repetition allows generation to terminate much earlier instead of continuing the attack-induced loop.

Table 14 separately summarizes mean throughput. Across the two LLMs, throughput changes only slightly after intervention, decreasing by 3.00 tokens/s for Qwen2.5-3B and increasing by 1.05 tokens/s for Llama-3.2-3B. These small variations indicate that the intervention has negligible impact on generation throughput.

Table 12: Top-1 start layer under different maximum repetition distances $q_{\max}$. Layers are zero-indexed.

Attack	$q_{\max} = 1$	$q_{\max} = 2$	$q_{\max} = 3$	$q_{\max} = 4$	$q_{\max} = 5$
GCG	12	1	1	1	1
RECITE	1	1	1	1	1

Table 13: Attack-generation time on two LLMs. Reduction is computed as $(T_{\text{orig}} - T_{\text{TRC}})/T_{\text{orig}}$. Lower is better.

Model	Attack	Original (s)	TRC (s)	Reduction
Qwen2.5-3B	Direct	278.92	9.33	96.7%
Qwen2.5-3B	GCG	402.52	1.73	99.6%
Qwen2.5-3B	LoopLLM	91.72	8.36	90.9%
Llama-3.2-3B	Direct	340.19	5.38	98.4%
Llama-3.2-3B	GCG	64.73	1.67	97.4%
Llama-3.2-3B	LoopLLM	143.44	4.68	96.7%

Table 14: Mean throughput on two LLMs. Difference is TRC minus Original. Throughput is measured in tokens per second.

Model	Original	TRC	Difference
Qwen2.5-3B	22.60	19.60	-3.00
Llama-3.2-3B	25.76	26.80	$+1.05$