

LingoLoop Attack: Trapping MLLMs via Linguistic Context and State Entrapment into Endless Loops

Jiyuan Fu*, Kaixun Jiang, Lingyi Hong, Jinglun Li, Haijing Guo,
Dingkang Yang, Zhaoyu Chen, Wenqiang Zhang

Fudan University

Abstract

Multimodal Large Language Models (MLLMs) have shown great promise but require substantial computational resources during inference. Attackers can exploit this by inducing excessive output, leading to resource exhaustion and service degradation. Prior energy-latency attacks aim to increase generation time by broadly shifting the output token distribution away from the EOS token, but they neglect the influence of token-level Part-of-Speech (POS) characteristics on EOS and sentence-level structural patterns on output counts, limiting their efficacy. To address this, we propose **LingoLoop**, an attack designed to induce MLLMs to generate excessively verbose and repetitive sequences. First, we find that the POS tag of a token strongly affects the likelihood of generating an EOS token. Based on this insight, we propose a **POS-Aware Delay Mechanism** to postpone EOS token generation by adjusting attention weights guided by POS information. Second, we identify that constraining output diversity to induce repetitive loops is effective for sustained generation. We introduce a **Generative Path Pruning Mechanism** that limits the magnitude of hidden states, encouraging the model to produce persistent loops. Extensive experiments demonstrate LingoLoop can increase generated tokens by up to $30\times$ and energy consumption by a comparable factor on models like Qwen2.5-VL-3B, consistently driving MLLMs towards their maximum generation limits. These findings expose significant MLLMs' vulnerabilities, posing challenges for their reliable deployment. The code will be released publicly following the paper's acceptance.

1 Introduction

Multimodal Large Language Models (MLLMs)[22, 30, 1, 8] excel at cross-modal tasks such as image captioning[19] and visual question answering [3, 39]. Owing to their high computational cost, they are typically offered via cloud service (e.g. GPT-4o [22], Gemini [30]). This setup, while convenient, exposes shared resources to abuse. Malicious users can craft adversarial inputs that trigger excessive computation or unusually long outputs. Such inference-time amplification attacks consume disproportionate resources, degrade service quality, and may even lead to denial-of-service (DoS)[15, 40, 18, 13] (see Figure 1).

Existing energy-latency attacks on MLLMs [14] typically attempt to suppress the End-of-Sequence (EOS) token by applying uniform pressure across all output tokens, irrespective of token type or position. However,

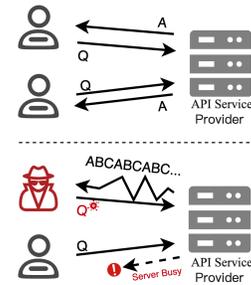


Figure 1: Normal vs. attacked MLLMs API operation.

*Email: fuji23@m.fudan.edu.cn

this strategy proves only marginally effective in increasing resource consumption. We attribute the limited efficacy of these existing approaches to two primary factors: 1) Our experimental analysis reveals that different Part-of-Speech (POS) tokens exhibit distinct propensities to trigger the EOS token. For instance, Figure 3 demonstrates that punctuation is notably more likely to be followed by EOS compared to tokens like adjectives or progressive verbs. A uniform suppression strategy used in prior works [14], however, disregards these crucial token-specific variations. Consequently, it applies pressure inefficiently to positions unlikely to terminate the sequence. This oversight leads to suboptimal optimization and, ultimately, reduced attack effectiveness. 2) Current methods often overlook the impact of sentence-level structural patterns on generation token counts. For instance, inducing repetitive patterns—a common tactic that significantly inflates resource usage, which is not explicitly leveraged by existing attack frameworks.

To address the aforementioned limitations and efficiently induce prolonged and repetitive outputs from MLLMs, we propose **LingoLoop Attack**. First, building upon our analysis that different POS tokens exhibit distinct propensities to trigger the EOS token, we developed the **POS-Aware Delay Mechanism**. This mechanism constructs a POS-aware prior probability model by statistically analyzing the correlation between part-of-speech tags and EOS token prediction probabilities across large-scale data. Then, leveraging these estimated prior probabilities, the mechanism dynamically adjusts postpone EOS token generation by adjusting attention weights guided by POS information. Second, we propose a **Generative Path Pruning Mechanism** to systematically induce repetitive generation and maximize output length. Our design is motivated by empirical analysis of hidden state dynamics, which reveals that repetitive outputs consistently correlate with low-variance regions in the model’s latent space. The mechanism operates by actively constraining the L_2 norm of hidden states at each decoding step, deliberately compressing the model’s trajectory into a restricted subspace. This strategic limitation of the hidden state manifold progressively reduces output diversity, forcing the model into a stable loop. Through this controlled degradation of generation diversity, we effectively establish and maintain a persistent looping state that amplifies output length.

By integrating these two mechanisms, LingoLoop Attack effectively delays sequence termination while simultaneously guiding the model into repetitive generation patterns, our main contributions can be summarized as follows:

- We analyze MLLMs internal behaviors, showing: 1) the significant influence of a preceding token’s Part-of-Speech tag on the probability of the next token being an EOS token, and 2) a strong correlation between hidden state statistical properties and the emergence of output looping. This analysis reveals critical limitations in prior verbose attack strategies.
- We propose the **LingoLoop Attack**, a synergistic two-component methodology designed to exploit these findings, featuring: 1) POS-Aware Delay Mechanism for context-aware termination delay, and 2) Generative Path Pruning Mechanism to actively induce repetitive, high-token-count looping patterns.
- Extensive experiments show our method achieves extreme verbosity, generating up to $30\times$ more tokens and consuming $30\times$ more energy than clean inputs, as shown in Table 3, significantly surpassing previous attacks in exposing MLLMs resource exhaustion risks.

2 Related Work

Multimodal Large Language Models. Multimodal Large Language Models (MLLMs) extend a powerful extension of traditional Large Language Models (LLMs), integrating visual perception capabilities [37, 34, 23]. These models typically comprise a vision encoder to interpret images, a core LLM for reasoning and language tasks, and an alignment module connecting the two modalities. The design of this connection and the overall architecture influences model behavior and efficiency. For example, architectures like InstructBLIP [11] employ sophisticated mechanisms, such as an instruction-guided Querying Transformer, to dynamically focus visual feature extraction based on textual context. More recent developments, represented by the Qwen2.5-VL series [1] (including the 3B and 7B variants central to our study), build upon dedicated LLM foundations like Qwen2.5 [38]. They incorporate optimized vision transformers, featuring techniques like window attention and efficient MLP-based merging, aiming for strong performance in fine-grained visual understanding and document analysis across model scales. Another advanced architecture, InternVL3-8B [8, 7], employs Native Multimodal Pre-Training with V2PE [17] for long contexts and MPO [35] for

reasoning optimization. Evaluating these approaches is crucial for understanding their operational characteristics, particularly energy consumption under adversarial conditions.

Energy-latency Attack. Energy-latency attacks (also known as sponge attacks) [33] aim to maximize inference time or energy consumption via malicious inputs, thereby threatening system availability [33, 21, 26, 20, 32, 29, 24, 6]. These attacks typically exploit the efficiency optimization mechanisms inherent in models or hardware, potentially leading to Denial-of-Service (DoS) conditions, inflated operational costs, or rapid battery depletion on edge devices [33, 10, 36]. For instance, early work targeted fundamental optimization principles by constructing inputs designed to minimize activation sparsity in CNNs [33, 25] or maximize the number of internal operations within Transformer models [33]. These ideas were later extended to traditional image captioning systems (e.g., CNN-RNN models), with attacks like NICGSlowDown [5] manipulating image inputs to force longer decoding sequences. In the domain of text generation, NMTSloth [4] targeted neural machine translation models, crafting prompts that prolonged generation and increased computation. As LLMs became dominant, prompt-level attacks such as P-DOS [16] were proposed to exploit their autoregressive decoding behavior. With the advent of MLLMs, research has begun to explore energy-latency attacks targeting these novel architectures. Gao et al. [14] recently proposed the Verbose Images method. This technique introduces imperceptible perturbations to the input image, inducing the MLLMs to generate lengthy textual descriptions, which in turn significantly increases the model’s inference costs. However, it overlooks how part-of-speech information influences the likelihood of generating an EOS token, limiting its ability to fully exploit linguistic cues for prolonged generation.

3 Methodology

3.1 Preliminaries

Our primary objective is to design an adversarial attack targeting MLLMs. The attacker aims to craft an adversarial image \mathbf{x}' from an original image \mathbf{x} and a given input prompt c_{in} . This adversarial image \mathbf{x}' should induce the MLLMs to generate a highly verbose or even repetitive output sequence $\mathbf{y} = \{y_1, y_2, \dots, y_{N_{\text{out}}}\}$. The generation of each token y_j is associated with an output probability distribution $f_j(\mathbf{x}')$, an EOS probability $f_j^{\text{EOS}}(\mathbf{x}')$, and a set of hidden state vectors across L model layers, $h_j(\mathbf{x}') = \{h_j^{(l)}(\mathbf{x}')\}_{l=1}^L$. The attacker operates under a **white-box scenario**, possessing full knowledge of the target MLLM’s architecture, parameters, and gradients. This enables the use of gradient-based methods to optimize the adversarial perturbation. The adversarial image \mathbf{x}' is constrained by an l_p -norm bound:

$$\|\mathbf{x}' - \mathbf{x}\|_p \leq \epsilon, \tag{1}$$

where ϵ is the perturbation budget. Given the strong correlation between MLLMs’ computational costs (e.g., energy consumption and latency) and the number of output tokens, the attacker’s ultimate goal is to maximize the length of the generated token sequence, $N_{\text{out}}(\mathbf{x}')$. This can effectively degrade or even paralyze MLLMs services. Formally, the attacker’s objective is:

$$\max_{\mathbf{x}'} N_{\text{out}}(\mathbf{x}'), \tag{2}$$

subject to the constraint in Equation (1).

To maximize the number of output tokens produced by MLLMs from adversarial images \mathbf{x}' , we introduce the **LingoLoop Attack**. This methodology counteracts natural termination and manipulates state evolution to promote sustained, high-volume token generation. It synergistically combines two primary components: 1) **POS-Aware Delay Mechanism**, as detailed in Sec.3.2, and 2) **Generative Path Pruning Mechanism** (Sec. 3.3), which induces looping by constraining hidden state magnitudes to guide the model towards repetitive, high-volume outputs. These components are integrated into a weighted objective function, and the overall optimization approach is detailed in Sec.3.4. Figure 2 presents the framework of our LingoLoop Attack.

3.2 POS-Aware Delay Mechanism

A key challenge in prolonging MLLMs generation is their **natural termination behavior**, where the model predicts an EOS token based on linguistic cues in the preceding context. While prior

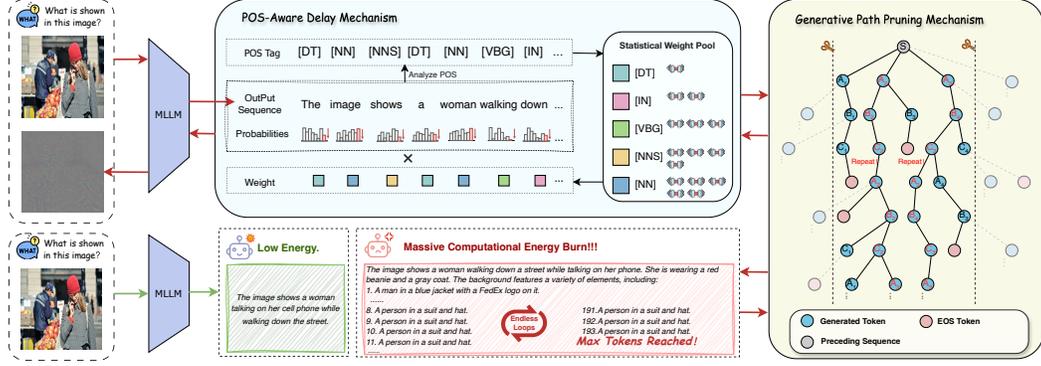


Figure 2: **Overview of the LingoLoop Attack framework.** This two-stage attack first employs a **POS-Aware Delay Mechanism** that leverages linguistic priors from Part-of-Speech tags to suppress premature sequence termination. Subsequently, the **Generative Path Pruning Mechanism** constrains hidden state representations to induce sustained, high-volume looping outputs.

work [14] attempted to delay termination by uniformly suppressing EOS probabilities, our analysis (see Figure 3) reveals that EOS predictions are strongly correlated with the POS tag of the preceding token. This motivates our **POS-Aware Delay Mechanism**, which dynamically suppresses EOS token probabilities based on linguistic priors derived from POS statistics.

When processing an adversarial image \mathbf{x}' and prompt c_{in} , the MLLMs auto-regressively generates an output token sequence $\mathbf{y} = \{y_1, \dots, y_{N_{\text{out}}}\}$. For each i -th token y_i in this generated sequence (where i ranges from 1 to N_{out}), the model provides the corresponding logits vector $\mathbf{z}_i(\mathbf{x}')$. The EOS probability for this step, $f_i^{\text{EOS}}(\mathbf{x}')$, is then derived from these logits:

$$(\mathbf{y}, \{\mathbf{z}_j(\mathbf{x}')\}_{j=1}^{N_{\text{out}}}) = \text{MLLM}(\mathbf{x}', c_{\text{in}}); \quad f_i^{\text{EOS}}(\mathbf{x}') = (\text{softmax}(\mathbf{z}_i(\mathbf{x}'))_{\text{EOS}}). \quad (3)$$

Subsequently, for each i -th newly generated token y_i in the output sequence \mathbf{y} (where i ranges from 1 to N_{out}), we determine the POS tag of its predecessor token, y_{i-1} . For $i = 1$, the predecessor y_0 is taken as the last token in c_{in} . For all subsequent tokens ($i > 1$), y_{i-1} is the actual $(i - 1)$ -th token from the generated sequence \mathbf{y} . The POS tag t_{i-1} is then obtained as:

$$t_{i-1} = \text{POS}(y_{i-1}). \quad (4)$$

This POS tag t_{i-1} is then used to query our pre-constructed **Statistical Weight Pool**, which encodes linguistic priors for EOS prediction conditioned on POS tags. Specifically, for each Part-of-Speech tag t , the pool stores an empirical prior $\bar{P}_{\text{EOS}}(t)$, representing the average probability that the model predicts an EOS token immediately after generating a token with POS tag t . To estimate these priors, we input a large collection of images (e.g., from ImageNet [12] and MSCOCO [27]) into the MLLMs and collect its generated output sequences. For each generated token, we extract the EOS probability predicted at the next time step, and categorize these values by the POS tag of the current token. The average of these grouped EOS probabilities yields the final value of $\bar{P}_{\text{EOS}}(t)$. A weight w_i for the i -th generation step is then computed from the linguistic prior associated with the preceding POS tag, $\bar{P}_{\text{EOS}}(t_{i-1})$, using a predefined weighting function ϕ_w :

$$w_i = \phi_w(\bar{P}_{\text{EOS}}(t_{i-1}); \boldsymbol{\theta}_w), \quad (5)$$

where $\boldsymbol{\theta}_w$ represents a set of parameters governing the transformation from prior probabilities to weights. This function ϕ_w is designed such that the resulting weight w_i is typically larger when the linguistic prior $\bar{P}_{\text{EOS}}(t_{i-1})$ is higher, signifying that the preceding POS tag ' t_{i-1} ' is statistically more likely to be followed by an EOS. Furthermore, the resulting weights are often normalized (e.g., to the range $[0, 1]$) for stable optimization. Thus, POS tags indicating a higher natural likelihood of termination will correspond to a larger w_i , focusing suppressive attention in the loss function. Finally, to actively suppress premature termination, we define the **Linguistic Prior Suppression loss** (\mathcal{L}_{LPS}). This loss is a key component of the POS-Aware Delay Mechanism (Figure 2). It aims to reduce the EOS probability, particularly in contexts identified by w_i as linguistically prone to termination:

$$\mathcal{L}_{\text{LPS}}(\mathbf{x}') = \frac{1}{N_{\text{out}}} \sum_{i=1}^{N_{\text{out}}} (w_i \cdot f_i^{\text{EOS}}(\mathbf{x}')). \quad (6)$$

by a hyperparameter λ_{rep} . For each output token $k \in 1, \dots, N_{\text{out}}$, we first define its average hidden state norm across all L transformer layers as:

$$\bar{r}_k = \frac{1}{L} \sum_{l=1}^L \left\| h_k^{(l)}(\mathbf{x}') \right\|_2, \quad (7)$$

where $h_k^{(l)}(\mathbf{x}')$ denotes the hidden state at layer l corresponding to the k -th output token. We then define the **Repetition Promotion Loss** as the mean of these norms across all output tokens, scaled by a regularization coefficient λ_{rep} :

$$\mathcal{L}_{\text{Rep}}(\mathbf{x}') = \frac{\lambda_{\text{rep}}}{N_{\text{out}}} \sum_{k=1}^{N_{\text{out}}} \bar{r}_k. \quad (8)$$

Minimizing \mathcal{L}_{Rep} (Eq. 8) drives down the magnitudes of output-time hidden states, reducing representational diversity and promoting repetition. This realizes the **Generative Path Pruning Mechanism** effect and significantly improves attack effectiveness beyond EOS-suppression alone.

3.4 Overall Objective and Optimization

To effectively craft adversarial images (\mathbf{x}') as part of our LingoLoop Attack, our ultimate goal is to maximize the output token count $N_{\text{out}}(\mathbf{x}')$ (Eq.(2)), subject to the constraint in Eq.(1).

The combined objective integrates \mathcal{L}_{LPS} (Sec. 3.2) and \mathcal{L}_{Rep} (Sec. 3.3), with \mathcal{L}_{LPS} scaled by factor α for numerical stability (see Supplemental Material). Following VerboseImages [14], dynamic weighting balances their contributions through:

$$\mathcal{L}_{\text{Total}}(x', t) = \alpha \cdot \mathcal{L}_{\text{LPS}}(x') + \lambda(t) \cdot \mathcal{L}_{\text{Rep}}(x'). \quad (9)$$

Here, the dynamic weight $\lambda(t)$ modulates the influence of \mathcal{L}_{Rep} by comparing the magnitudes of the two losses from the previous iteration ($t-1$), scaled by a temporal decay function $\mathcal{T}(t)$.

$$\lambda(t) = \frac{\|\mathcal{L}_{\text{LPS}}(\mathbf{x}'_{t-1})\|_1}{\|\mathcal{L}_{\text{Rep}}(\mathbf{x}'_{t-1})\|_1} / \mathcal{T}(t). \quad (10)$$

The temporal decay function is defined as: $\mathcal{T}(t) = a \ln(t) + b$, where a and b are hyperparameters controlling the decay rate. Momentum can also be applied when updating $\lambda(t)$ from one iteration to the next to smooth the adjustments. This dynamic balancing adapts the focus between EOS suppression and repetition induction over time. The LingoLoop Attack minimize $\mathcal{L}_{\text{Total}}(x', t)$ via Projected Gradient Descent (PGD) [28] for T steps, updating $\mathcal{L}_{\text{Total}}$ and projecting it back onto the ℓ_p -norm ball centered at the original image x . The detailed procedural description of the LingoLoop Attack is provided in **Appendix B**.

4 Experiments

4.1 Experimental Setting

Models and Dataset. We evaluate our approach on four recent multimodal large language models: InstructBLIP [11], Qwen2.5-VL-3B-Instruct [1], Qwen2.5-VL-7B-Instruct [1], and InternVL3-8B [8, 7]. InstructBLIP employs the Vicuna-7B language model backbone, while the Qwen2.5-VL-3B model utilizes the Qwen2.5-3B architecture, and both the Qwen2.5-VL-7B and InternVL3-8B models are built upon the Qwen2.5-7B language model architecture. Following the experimental protocol of Verbose Images [14], we assess all models on the image captioning task. To ensure methodological consistency and enable fair comparisons, we use the default prompt templates provided for each model. For evaluating the primary task performance and attack effectiveness, we utilize images from two standard benchmarks: MSCOCO [27] and ImageNet [12]. Our evaluation set comprises 200 randomly selected images from each dataset. For EOS prediction probability analysis by word category, we sample 5,000 images from each dataset (non-overlapping with evaluation sets).

Table 1: Comparison of the LingoLoop Attack against baseline methods across four MLLMs (InstructBLIP, Qwen2.5-VL-3B, Qwen2.5-VL-7B, InternVL3-8B) on the MS-COCO and ImageNet datasets (200 images each). Metrics include generated token count, energy consumption (J), and inference latency (s). The best results for each metric are highlighted in **bold**.

MLLM	Attack Method	MS-COCO			ImageNet		
		Tokens	Energy	Latency	Tokens	Energy	Latency
InstructBLIP	None	86.11	428.72	4.91	73.03	356.94	3.96
	Noise	85.78	426.22	4.95	74.19	381.49	4.32
	Verbose images	332.29	1241.89	17.79	451.85	1612.14	23.70
	Ours	1002.08	3152.26	57.30	984.65	2814.71	54.75
Qwen2.5-VL-3B	None	66.64	430.01	2.24	64.09	427.30	2.12
	Noise	68.07	440.25	2.40	65.21	433.87	2.18
	Verbose images	394.74	2682.38	13.12	525.70	3650.52	17.12
	Ours	1020.38	7090.58	32.94	1014.62	7108.50	32.50
Qwen2.5-VL-7B	None	88.86	445.25	1.84	82.35	405.87	1.70
	Noise	88.24	446.17	1.88	79.29	403.71	1.65
	Verbose images	345.59	1738.00	6.99	384.62	1916.10	7.74
	Ours	797.55	3839.70	15.24	825.23	4105.09	15.87
InternVL3-8B	None	76.31	379.14	1.39	65.04	318.14	1.19
	Noise	74.89	362.10	1.38	67.10	321.29	1.22
	Verbose images	362.38	1810.23	6.40	329.02	1634.89	5.80
	Ours	554.41	2771.76	9.70	613.35	3183.87	11.08

Attacks Settings. We compare our proposed method against several baselines, including original, unperturbed images, images with random noise added (sampled uniformly within the same ϵ budget as attacks), and the Verbose Images attack [14], which represents the state-of-the-art energy-latency attack for MLLMs. For generating adversarial examples using both the Verbose Images baseline and our method, the adversarial perturbations are optimized via the PGD [28] algorithm with $T = 300$ iterations. We enforce an ℓ_∞ constraint with $\epsilon = 8$ on the perturbations and use a step size of $\eta = 1$. During inference, MLLMs generate text sequences with a maximum token count of 1024 tokens using greedy decoding `do_sample=False` to ensure reproducibility. Following the experimental settings established by Verbose Images [14], we set the loss weight parameters to $a = 10$ and $b = -20$. The PGD updates are performed with momentum $m = 1.0$ and we fix $\theta_w = 10^5$.

Metrics. We primarily evaluate the effectiveness of our approach by measuring the number of tokens in the sequence generated by the MLLMs. Since increased sequence length inherently demands greater computational resources, it directly translates to higher energy consumption and inference latency, which are the ultimate targets of energy-latency attacks. Consequently, in addition to token count, we report the **energy consumed** (measured in Joules, J) and the **latency** incurred (measured in seconds, s) during the inference process [33]. All measurements were conducted on a single GPU with consistent hardware contexts: NVIDIA RTX 3090 for Qwen2.5-VL-3B, NVIDIA V100 for InstructBLIP, and NVIDIA H100 for Qwen2.5-VL-7B and InternVL3-8B.

4.2 Main results

To assess the efficacy of LingoLoop, we conducted extensive experiments using images from the MS-COCO and ImageNet datasets (200 images each). LingoLoop’s performance was benchmarked against three key conditions: (1) unperturbed clean inputs (‘None’); (2) inputs with added random noise (‘Noise’); and (3) adversarial inputs generated by (‘Verbose Images’) [14], the current state-of-the-art verbose attack. Table 1 summarizes the key metrics: generated token counts, inference latency, and energy consumption across various MLLMs.

As shown in Table 1, random noise inputs produce outputs comparable to clean inputs, confirming naive perturbations cannot induce verbosity. In contrast, LingoLoop Attack consistently achieves significantly longer outputs and higher resource utilization. For MS-COCO images, it compels InstructBLIP to generate 1002.08 tokens ($11.6\times$ clean inputs, $3.0\times$ Verbose Images) with 57.30 J energy ($11.7\times$ and $2.4\times$ higher). This pattern holds across models: Qwen2.5-VL-3B outputs 1020.38 tokens ($15.3\times$ clean, $2.6\times$ Verbose Images) consuming 32.94 J ($14.7\times$ and $2.5\times$ higher). The same near-maximal generation behavior occurs consistently on ImageNet and other MLLMs (Qwen2.5-VL-7B, InternVL3-8B). The experimental findings in Table 1 establish LingoLoop’s superior capability in forcing MLLMs into states of extreme verbosity, leading to significant resource exhaustion. The

consistent success in pushing diverse MLLMs to their output limits validates the effectiveness of LingoLoop’s core strategies: the POS-Aware Delay Mechanism and the Generative Path Pruning Mechanism, which work synergistically to achieve these results.

4.3 Hyperparameter Optimization

Repetition Induction Strength (λ_{rep}) We conduct an ablation study on λ_{rep} , the hyperparameter controlling the strength of the Repetition Induction loss (\mathcal{L}_{Rep}). This loss penalizes the L_2 norm of hidden states in the generated output sequence to promote repetitive patterns. These experiments are performed on 100 images from the MS-COCO using the Qwen2.5-VL-3B, with attack parameters set to 300 iterations and $\epsilon = 8$. As shown in Figure 5, varying λ_{rep} significantly impacts the attack’s effectiveness. A low λ_{rep} (e.g., 0.1) provides insufficient pressure on hidden states, resulting in limited repetition and lower token counts. As λ_{rep} increases, the constraint becomes stronger, effectively guiding the model towards repetitive patterns, which is reflected in the increasing token counts, Energy consumption, and Latency. However, excessively high λ_{rep} (e.g., 0.6, 0.7) might overly constrain the state space, potentially hindering even basic generation or leading to unproductive short loops, causing the metrics to decrease after peaking around $\lambda_{\text{rep}} = 0.5$. This demonstrates the necessity of finding an optimal balance for the hidden state magnitude constraint.

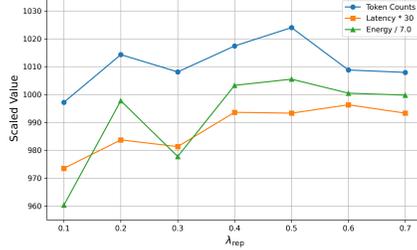


Figure 5: Effect of λ_{rep} on Generated Token Counts, Energy, and Latency.

Attack iterations To determine a suitable number of PGD steps for our attack, we conduct a convergence analysis on 100 randomly sampled images from the MSCOCO using the Qwen2.5-VL-3B model, under an ℓ_∞ perturbation budget of $\epsilon = 8$. As shown in Figure 6, our method (LingoLoop Attack) achieves rapid growth in generated token count and converges near the maximum output limit within 300 steps. For reference, we also include three partial variants using \mathcal{L}_{Rep} , \mathcal{L}_{LPS} , and their combination. Compared to the full method, these curves converge slower or plateau earlier, indicating that removing components not only affects final attack strength, but also hinders the optimization process. This supports our design choice to integrate both objectives for faster and more stable convergence.

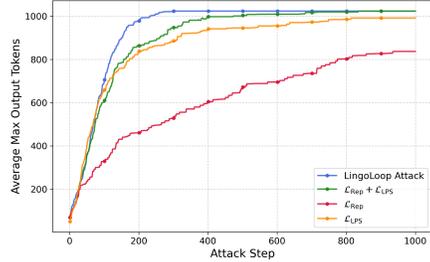


Figure 6: Convergence of generated token counts versus PGD attack steps for LingoLoop Attack and its components on MSCOCO (100 images).

4.4 Ablation Study

To analyze the LingoLoop Attack’s effectiveness and understand the contribution of its key components, we conduct ablation experiments. These studies are performed on image subsets from the MSCOCO [27] and ImageNet [12] datasets, utilizing the Qwen2.5-VL-3B model [1] for validation.

Effect of loss objectives This ablation investigates the contribution of our proposed loss objectives, \mathcal{L}_{LPS} and \mathcal{L}_{Rep} . These experiments are conducted on 100 images from the MS-COCO dataset using the Qwen2.5-VL-3B model, with attack parameters set to 300 iterations and $\epsilon = 8$. As shown in Table 2, employing a baseline with uniform EOS weights yields 843.86 generated tokens. Using only \mathcal{L}_{LPS} improves this to 926.94 tokens, highlighting the benefit of POS-weighted suppression in delaying termination. Conversely, using only \mathcal{L}_{Rep} results in fewer tokens (561.90), as its primary focus is on state compression to induce repetition, not direct sequence lengthening. However, the combination of both \mathcal{L}_{LPS} and \mathcal{L}_{Rep} (without momentum) achieves significantly higher generated tokens (963.51), demonstrating the synergistic effect. This synergy arises because \mathcal{L}_{LPS}

Table 2: Ablation Study on Attack Modules.

\mathcal{L}_{LPS}	\mathcal{L}_{Rep}	Mom.	MS-COCO		
			Length	Energy	Latency
Uniform weights			843.86	5329.82	25.12
✓			926.94	6265.61	29.04
	✓		561.90	3863.13	17.90
✓	✓		963.51	6408.13	29.78
✓	✓	✓	1024.00	6926.44	32.41

Table 3: Performance metrics under varying maximum token generation limits (max_new_tokens).

max_new_tokens	Attack Method	MS-COCO			ImageNet		
		Tokens	Energy	Latency	Tokens	Energy	Latency
-	None	67.77	475.49	2.60	62.71	421.83	2.65
256	Verbose images	178.97	1263.87	5.91	185.52	1205.74	6.11
	Ours	256.00	2069.87	10.22	256.00	2191.64	10.05
512	Verbose images	252.14	1855.76	8.30	277.81	1842.45	9.13
	Ours	512.00	3991.25	17.79	511.29	3933.29	17.65
1024	Verbose images	328.13	2353.23	11.74	490.72	3379.51	18.02
	Ours	1024.00	6926.44	32.41	1013.35	7667.13	32.49
2048	Verbose images	634.58	5088.90	20.35	853.13	6225.33	27.77
	Ours	2048.00	14386.41	69.51	2048.00	16464.77	72.78

creates the opportunity for extended generation by suppressing termination, while \mathcal{L}_{Rep} exploits this opportunity by guiding the model into repetitive, high-volume output patterns.

Maximum output token As part of our ablation study, we investigate the performance of different attack methods under varying max_new_tokens limits. Using the Qwen2.5-VL-3B model, attacked with 300 PGD steps ($\epsilon = 8$), we measure the generated token count, inference latency, and energy consumption on 100-image subsets from MS-COCO and ImageNet. Table 3 presents these results. Original inputs terminate quickly. While Verbose Images [14] achieve increased output lengths, they consistently fail to reach the system’s maximum token limit. Our LingoLoop Attack, however, reliably drives token generation at or near the predefined max_new_tokens across all settings and datasets. This maximal token count directly leads to significantly higher latency and energy, demonstrating LingoLoop Attack’s superior capability to force maximum verbose output for resource exhaustion.

4.5 Robustness against Defense Methods

To validate LingoLoop Attack’s effectiveness against mitigation strategies, we evaluate it against model parameters controlling repetitive outputs. Table 4 shows Qwen2.5-VL-3B results on 100 MS-COCO images (P_1 : repetition_penalty, P_2 : no_repeat_ngram_size).

Under default settings, LingoLoop Attack substantially increases generated token counts and resource consumption compared to Clean and Verbose Images [14]. Increasing P_1 to 1.10 slightly reduces the generated token counts for Clean and Verbose Images, while $P_1 = 1.15$ surprisingly increases their output tokens. This suggests that higher repetition penalties, while discouraging exact repeats, can sometimes push the model towards generating longer sequences that avoid immediate penalties. Our attack consistently achieves the maximum token limit (1024) across all tested P_1 variations. Enabling $P_2 = 2$ (with $P_1 = 1.05$) unexpectedly increases the total number of tokens for Clean and Verbose Images. This likely occurs because preventing ngrams forces the model to use alternative phrasing or structures, potentially leading to longer outputs. It also fails to prevent our attack from reaching the maximum generation limit. These results demonstrate that standard repetition controls are ineffective against the LingoLoop Attack.

Table 4: Defense results on 100-image MS-COCO subset. P_1 : repetition_penalty, P_2 : no_repeat_ngram_size.

P_1	P_2	Attack Method	MS-COCO		
			Tokens	Energy	Latency
1.05	0	Clean	67.77	475.49	2.60
		Verbose images	328.13	2353.23	11.74
		Ours	1024.00	6926.44	32.41
1.10	0	Clean	66.00 ↓	580.84	4.02
		Verbose images	264.62 ↓	2279.88	15.64
		Ours	1024.00 —	7442.37	34.73
1.15	0	Clean	81.11 ↑	675.32	4.71
		Verbose images	445.49 ↑	3702.26	25.44
		Ours	1024.00 —	7256.91	33.94
1.05	2	Clean	206.56 ↑	1345.30	6.88
		Verbose images	1024.00 ↑	7240.59	33.91
		Ours	1024.00 —	7218.26	33.97

5 Conclusion

This paper introduced the LingoLoop Attack, a novel methodology for inducing extreme verbosity and resource exhaustion in Multimodal Large Language Models. Through a foundational analysis of MLLMs internal behaviors, we identified key contextual dependencies and state dynamics previously overlooked by verbose attack strategies. Our approach uniquely combines Part-of-Speech weighted End-of-Sequence token suppression with a hidden state magnitude constraint to actively promote

sustained, high-volume looping patterns. Extensive experiments validate that the LingoLoop Attack significantly outperforms existing methods, highlighting potent vulnerabilities and underscoring the need for more robust defenses against such sophisticated output manipulation attacks.

References

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibong Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025.
- [2] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O’Reilly, 2009.
- [3] James Burgess, Jeffrey J. Nirschl, Laura Bravo-Sánchez, Alejandro Lozano, Sanket Rajan Gupte, Jesus G. Galaz-Montoya, Yuhui Zhang, Yuchang Su, Disha Bhowmik, Zachary Coman, Sarina M. Hasan, Alexandra Johannesson, William D. Leineweber, Malvika G. Nair, Ridhi Yarlagadda, Connor Zuraski, Wah Chiu, Sarah Cohen, Jan N. Hansen, Manuel D. Leonetti, Chad Liu, Emma Lundberg, and Serena Yeung-Levy. Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. *CoRR*, abs/2503.13399, 2025.
- [4] Simin Chen, Cong Liu, Mirazul Haque, Zihong Song, and Wei Yang. Nmtslot: understanding and testing efficiency degradation of neural machine translation systems. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, Singapore, Singapore, November 14-18, 2022*, pages 1148–1160. ACM, 2022.
- [5] Simin Chen, Zihong Song, Mirazul Haque, Cong Liu, and Wei Yang. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15344–15353. IEEE, 2022.
- [6] Yiming Chen, Simin Chen, Zexin Li, Wei Yang, Cong Liu, Robby T. Tan, and Haizhou Li. Dynamic transformers provide a false sense of efficiency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7164–7180. Association for Computational Linguistics, 2023.
- [7] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhui Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *CoRR*, abs/2412.05271, 2024.
- [9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- [10] Antonio Emanuele Cinà, Ambra Demontis, Battista Biggio, Fabio Roli, and Marcello Pelillo. Energy-latency attacks via sponge poisoning. *Inf. Sci.*, 702:121905, 2025.
- [11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.
- [13] Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Han Qiu, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, and Ke Xu. An engorgio prompt makes large language model babble on. *CoRR*, abs/2412.19394, 2024.

- [14] 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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [15] Kuofeng Gao, Jindong Gu, Yang Bai, Shu-Tao Xia, Philip Torr, Wei Liu, and Zhifeng Li. Energy-latency manipulation of multi-modal large language models via verbose samples. *CoRR*, abs/2404.16557, 2024.
- [16] Kuofeng Gao, Tianyu Pang, Chao Du, Yong Yang, Shu-Tao Xia, and Min Lin. Denial-of-service poisoning attacks against large language models. *CoRR*, abs/2410.10760, 2024.
- [17] Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. V2PE: improving multimodal long-context capability of vision-language models with variable visual position encoding. *CoRR*, abs/2412.09616, 2024.
- [18] Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. Coercing llms to do and reveal (almost) anything. *CoRR*, abs/2402.14020, 2024.
- [19] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. Onellm: One framework to align all modalities with language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26574–26585. IEEE, 2024.
- [20] Mirazul Haque, Simin Chen, Wasif Arman Haque, Cong Liu, and Wei Yang. Antinode: Evaluating efficiency robustness of neural odes. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023*, pages 1499–1509. IEEE, 2023.
- [21] Sanghyun Hong, Yigitcan Kaya, Ionut-Vlad Modoranu, and Tudor Dumitras. A panda? no, it’s a sloth: Slowdown attacks on adaptive multi-exit neural network inference. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [22] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Gertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024.
- [23] Yiqiao Jin, Minje Choi, Gaurav Verma, Jindong Wang, and Srijan Kumar. MM-SOC: benchmarking multi-modal large language models in social media platforms. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6192–6210. Association for Computational Linguistics, 2024.
- [24] Sarada Krithivasan, Sanchari Sen, and Anand Raghunathan. Sparsity turns adversarial: Energy and latency attacks on deep neural networks. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 39(11):4129–4141, 2020.
- [25] Sarada Krithivasan, Sanchari Sen, and Anand Raghunathan. Sparsity turns adversarial: Energy and latency attacks on deep neural networks. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 39(11):4129–4141, 2020.
- [26] Sarada Krithivasan, Sanchari Sen, Nitin Rathi, Kaushik Roy, and Anand Raghunathan. Efficiency attacks on spiking neural networks. In *DAC ’22: 59th ACM/IEEE Design Automation Conference, San Francisco, California, USA, July 10 - 14, 2022*, pages 373–378. ACM, 2022.
- [27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

- [28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [29] K. L. Navaneet, Soroush Abbasi Koohpayegani, Essam Sleiman, and Hamed Pirsiavash. Slowformer: Adversarial attack on compute and energy consumption of efficient vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 24786–24797. IEEE, 2024.
- [30] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024.
- [31] Rylan Schaeffer, Dan Valentine, Luke Bailey, James Chua, Cristóbal Eyzaguirre, Zane Durante, Joe Benton, Brando Miranda, Henry Sleight, John Hughes, Rajashree Agrawal, Mrinank Sharma, Scott Emmons, Sanmi Koyejo, and Ethan Perez. Failures to find transferable image jailbreaks between vision-language models, 2024.
- [32] Avishag Shapira, Alon Zolfi, Luca Demetrio, Battista Biggio, and Asaf Shabtai. Phantom sponges: Exploiting non-maximum suppression to attack deep object detectors. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 4560–4569. IEEE, 2023.
- [33] Iliia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert D. Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. In *IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*, pages 212–231. IEEE, 2021.
- [34] Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models. *CoRR*, abs/2406.11230, 2024.
- [35] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024.
- [36] Zijian Wang, Shuo Huang, Yujin Huang, and Helei Cui. Energy-latency attacks to on-device neural networks via sponge poisoning. In *Proceedings of the 2023 Secure and Trustworthy Deep Learning Systems Workshop, SecTL 2023, Melbourne, VIC, Australia, July 10-14, 2023*, pages 4:1–4:11. ACM, 2023.
- [37] Peng Xia, Siwei Han, Shi Qiu, Yiyang Zhou, Zhaoyang Wang, Wenhao Zheng, Zhaorun Chen, Chenhang Cui, Mingyu Ding, Linjie Li, Lijuan Wang, and Huaxiu Yao. MMIE: massive multimodal interleaved comprehension benchmark for large vision-language models. *CoRR*, abs/2410.10139, 2024.
- [38] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.
- [39] Shuo Yang, Siwen Luo, and Soyeon Caren Han. Multimodal commonsense knowledge distillation for visual question answering (student abstract). In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 29545–29547. AAAI Press, 2025.
- [40] 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, 2025.

Appendix

- In Appendix A, we provide implementation details.
- In Appendix B, we provide the pseudo code of our LingoLoop Attack.
- In Appendix C, we provide results on the generalization and robustness of our LingoLoop Attack, including its transferability to higher maximum output tokens, its performance under various prompt variations, and its cross-model transferability.
- In Appendix D, we provide additional ablation studies, specifically examining the impact of perturbation magnitude (ϵ) and sampling temperature.
- In Appendix E, we provide limitations.
- In Appendix F, we provide broader impacts.
- In Appendix G, we provide visualizations.

A Implementation Details

This section outlines the specific configurations and methodologies employed in our experiments, including the setup of the Multimodal Large Language Models (MLLMs) used and the construction of the Statistical Weight Pool crucial for our POS-Aware Delay Mechanism.

A.1 Model Setup

In this study, we primarily utilized four open-source MLLMs to evaluate the LingoLoop Attack: InstructBLIP [11], Qwen2.5-VL-3B [1], Qwen2.5-VL-7B [1], and InternVL3-8B [7]. The specific configurations for each model are detailed below.

InstructBLIP. In this study, we utilized the InstructBLIP model with its Vicuna-7B language model backbone [9]. Input images are preprocessed by resizing them to a resolution of 224×224 pixels. For LingoLoop Attack, the input prompt c_{in} was set to: "`<Image>` What is the content of this image?". To ensure numerical stability during the optimization, we scaled \mathcal{L}_{LPS} by a factor of $\alpha = 1 \times 10^5$.

Qwen2.5-VL (3B and 7B). For the Qwen2.5-VL series, we evaluated versions built upon both the 3-billion parameter and 7-billion parameter Qwen2.5 LLM backbones [38]. Specifically, we utilized the Qwen/Qwen2.5-VL-3B-Instruct and Qwen/Qwen2.5-VL-7B-Instruct models sourced from the Hugging Face Hub. Input images are resized to a resolution of 224×224 pixels. For generating textual outputs, we employed the default prompt template recommended for these -Instruct models. The common structure for this prompt c_{in} is: "`<|im_start|>system\nYou are a helpful assistant.<|im_end|>\n<|im_start|>user\n<|vision_start|>image_token<|vision_end|>What is shown in this image?<|im_end|>\n<|im_start|>assistant\n`". For these models, the scaling factor α for the \mathcal{L}_{LPS} was set to 1.

InternVL3-8B. In our evaluations, we also include the InternVL3-8B model, which utilizes a Qwen2.5 7b LLM as its backbone [38]. We use the version sourced from the Hugging Face Hub under the identifier OpenGVLab/InternVL3-8B. Input images for this model are preprocessed by resizing them to a resolution of 448×448 pixels, followed by any standard normalization procedures specific to the model. The input prompt c_{in} is set to: "`<image>\nWhat is shown in this image?`". For InternVL3-8B, the scaling factor α for the \mathcal{L}_{LPS} was set to 1×10^5 .

A.2 Statistical Weight Pool Construction

The Statistical Weight Pool, integral to our POS-Aware Delay Mechanism, captures and models the empirical probabilities of an End-of-Sequence (EOS) token occurring after tokens with specific Part-of-Speech (POS) tags. To construct this pool for each evaluated MLLM, we utilized a large, diverse set of images, sampling 5000 images from the MS-COCO [27] dataset and another 5000 images from the ImageNet [12] dataset. These image sets were distinct from those used in our main attack evaluations.

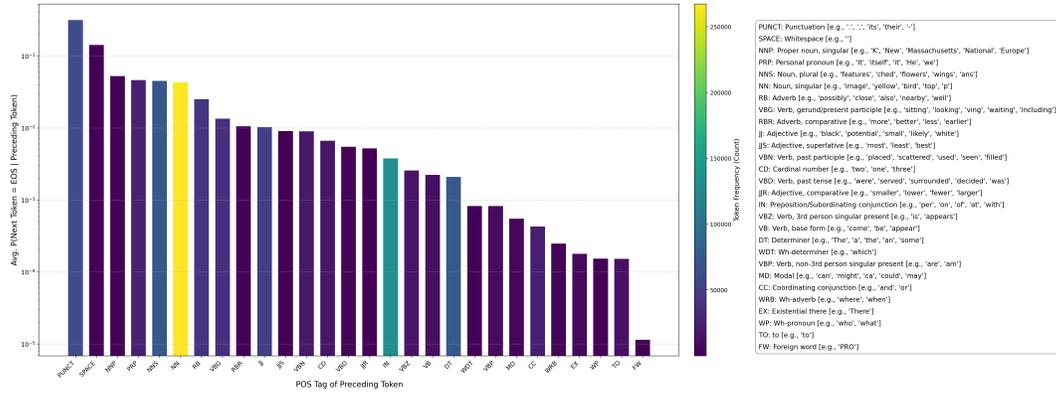


Figure 7: Empirical EOS prediction probability model based on preceding token POS tags in the InstructBLIP-Vicuna-7B model. The bar color indicates the relative frequency of each POS tag in the analysis dataset.

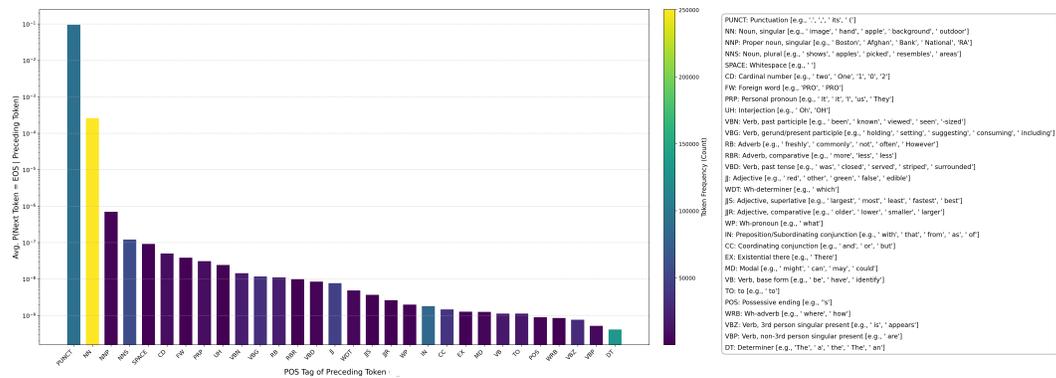


Figure 8: Empirical EOS prediction probability model based on preceding token POS tags in the Qwen2.5-VL-7B model. The bar color indicates the relative frequency of each POS tag in the analysis dataset.

For each MLLM, every sampled image was individually fed as input, prompting the model to generate a textual output (e.g., an image caption). We then analyzed these generated sequences. Specifically, for each token produced by the MLLM, we identified its POS tag using the NLTK (Natural Language Toolkit) [2] library’s POS tagger. Simultaneously, we recorded the probability assigned by the MLLM to the next token being an EOS token, given the current token and context. These EOS probabilities were then grouped by the POS tag of the current token, and the average EOS probability was calculated for each POS tag category across all generated outputs for that specific MLLM. This process yielded an empirical POS-to-EOS-probability mapping for each model.

These empirical probability models are crucial for guiding the LingoLoop attack and reveal consistent qualitative trends across the diverse MLLM architectures evaluated. As illustrated in Figure 7 (InstructBLIP), Figure 3 in the main text (Qwen2.5-VL-3B), Figure 8 (Qwen2.5-VL-7B), and Figure 9 (InternVL3-8B), a clear pattern is observed: POS tags signifying syntactic endpoints, such as punctuation marks, consistently show a significantly higher probability of preceding an EOS token. Conversely, POS tags associated with words that typically extend descriptive narratives—such as adjectives and adverbs—generally demonstrate a lower likelihood of immediately triggering sequence termination. While the precise probability values differ across models, this underlying behavior, where structural and terminal linguistic cues are stronger indicators of EOS likelihood than content-extending tags, appears to be a shared characteristic. This commonality in how MLLMs interpret end-of-sequence signals based on POS context is what our POS-Aware Delay Mechanism leverages, forming the basis for its potential effectiveness and broader applicability.

B Pseudo Code of LingoLoop Attack

The pseudo-code detailing the LingoLoop Attack procedure is presented in Algorithm 1.

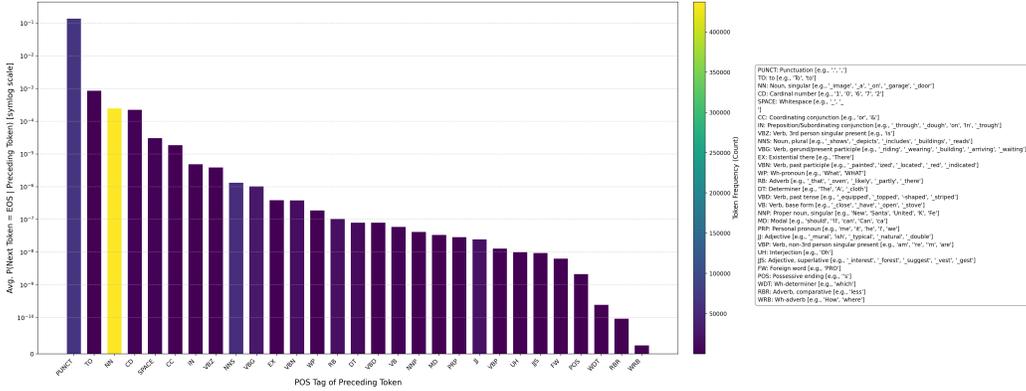


Figure 9: Empirical EOS prediction probability model based on preceding token POS tags in the InternVL3-8B model. The bar color indicates the relative frequency of each POS tag in the analysis dataset.

C Generalization & Robustness of the Attack

C.1 Transferability of Attacks to Higher Maximum Output Tokens

Our ablation studies (as detailed in Section 4.4) indicate that when LingoLoop attacks are generated with a `max_new_tokens` setting of 2048, the resulting outputs are notably long and often contain repetitive sequences. We believe that once a model is trapped in such a generative pattern, it will continue to output in this looping manner even if the external maximum output tokens constraint is relaxed. To verify this understanding, we investigate how examples, originally crafted with `max_new_tokens=2048`, perform when transferred to attack the MLLM operating under significantly higher `max_new_tokens` settings.

For this experiment, examples for Qwen2.5-VL-3B are generated using LingoLoop (and Verbose Images for comparison) with the `max_new_tokens` parameter set to 2048 during their creation phase. These exact same examples (i.e., the perturbed images) are then fed to the model during inference, but with the `max_new_tokens` cap raised to 8K, 16K, and 32K tokens, respectively. Clean images are also evaluated under these varying caps as a baseline.

The results, presented in Table 5, strikingly confirm our expectation. Critically, when the examples originally crafted with a `max_new_tokens` setting of 2048 are evaluated with higher inference caps, LingoLoop Attack continues to drive sustained generation, pushing outputs towards these new, much larger limits. For instance, on 100 randomly sampled MS-COCO images with an 8K max output token cap, our LingoLoop examples achieve an average output length of 7245.66 tokens. This pattern of extensive generation persists and scales with the increased caps of 16K and 32K, far exceeding the outputs from clean images. Under these transferred settings, LingoLoop also consistently generates substantially longer outputs than Verbose Images (which were also crafted with a 2048-token limit), often maintaining the established looping patterns. This strong transfer performance demonstrates that once LingoLoop Attack traps an MLLM into a generative loop, this looping state is highly persistent and continues to drive output even when the external token generation cap is significantly raised.

Table 5: Transferability of LingoLoop Attack (generated with `max_new_tokens=2048`) to higher maximum output tokens settings on Qwen2.5-VL-3B. All metrics are averaged over 100 images per dataset. Best performances are highlighted in **bold**.

Attack	MS-COCO (Average Tokens)				ImageNet (Average Tokens)			
	2048	8K	16K	32K	2048	8K	16K	32K
None			67.77				62.71	
VI [14]	634.58	1617.90	2657.90	4668.28	853.13	2680.34	4766.57	8432.70
ours	2048.00	7245.66	13162.96	23844.90 ($\times 351.9$)	2048.00	7284.87	12986.53	23054.76 ($\times 367.6$)

Algorithm 1 LingoLoop Attack

Input: Original image \mathbf{x} , prompt c_{in} , Perturbation budget ϵ , step size η , Momentum factor m , iterations T , \max_new_tokens N_{max}

- 1: **Preprocessing:**
- 2: 1. Estimate $\bar{P}_{EOS}(t) = \mathbb{E}[f^{EOS}|t]$ ▷ Build POS-EOS mapping
- 3: 2. Define $w(t) = \phi(\bar{P}_{EOS}(t); \theta)$ ▷ θ : scaling params
- 4: **Attack Initialization:**
- 5: $\mathbf{x}'_0 \leftarrow \mathbf{x} + \text{Uniform}(-\epsilon, +\epsilon)$ ▷ Perturbation initialization
- 6: $g_0 \leftarrow 0$ ▷ Momentum buffer
- 7: **for** $t = 1$ **to** T **do**
- 8: **Forward Pass:**
- 9: $(\mathbf{y}, \{\mathbf{z}_j\}, \{h_j^{(l)}\}) \leftarrow \text{MLLM}(\mathbf{x}'_{t-1}, c_{in})$ ▷ Get output sequences, logits and hiddenstates
- 10: $N_{out} \leftarrow |\mathbf{y}|$ ▷ Get generated token count
- 11: **if** $N_{out} \geq N_{max}$: **break** ▷ Early termination
- 12: **POS-Aware Delay Mechanism:**
- 13: **for** $i = 1$ **to** N_{out} **do**
- 14: $t_{i-1} \leftarrow \text{POS}(\mathbf{y}_{i-1})$ ▷ Predecessor POS tagging
- 15: $w_i \leftarrow w(t_{i-1})$ ▷ Retrieve suppression weight
- 16: $f_i^{EOS} \leftarrow \text{softmax}(\mathbf{z}_i)_{\text{EOS}}$
- 17: **end for**
- 18: $\mathcal{L}_{LPS} \leftarrow \frac{1}{N_{out}} \sum w_i f_i^{EOS}$
- 19: **Generative Path Pruning Mechanism:**
- 20: $\bar{r}_k \leftarrow \frac{1}{L} \sum_{l=1}^L \|h_k^{(l)}\|_2, \forall k \in [1, N_{out}]$
- 21: $\mathcal{L}_{Rep} \leftarrow \frac{\lambda}{N_{out}} \sum \bar{r}_k$
- 22: **Dynamic Adaptation:**
- 23: $\lambda(t) \leftarrow \frac{\|\mathcal{L}_{LPS}\|}{\|\mathcal{L}_{Rep}\|} / (a \ln t + b)$ ▷ Temporal decay
- 24: $\mathcal{L}_{Total} \leftarrow \alpha \mathcal{L}_{LPS} + \lambda(t) \mathcal{L}_{Rep}$
- 25: **Parameter Update:**
- 26: $g_t \leftarrow m \cdot g_{t-1} + \nabla_{x'} \mathcal{L}_{Total}$ ▷ Momentum gradient
- 27: $\mathbf{x}'_t \leftarrow \text{Clip}_\epsilon(\mathbf{x}'_{t-1} - \eta \cdot \text{sign}(g_t))$ ▷ Projected update
- 28: **end for**

Output: Perturbed image \mathbf{x}'_T with looping induction effect

C.2 Robustness to Prompt Variations

To comprehensively evaluate the generalization and robustness of the LingoLoop Attack, we examine its performance under varying textual prompts. The attack examples in this section were generated on the Qwen2.5-VL-3B model using 100 randomly selected images from the MS-COCO dataset. Each example was crafted with 300 PGD steps under the default prompt (Q_{orig} : "What is shown in this image?") and a \max_new_tokens setting of 2048. These same generated attack examples were then paired with a diverse set of new prompts during inference to assess LingoLoop Attack's efficacy when presented with queries related to the visual content (Related Prompts) and queries entirely independent of it (Unrelated Prompts).

Performance with Related Prompt. We first examine LingoLoop Attack's behavior when these 2048-token-budget attack examples are paired with prompts that, similar to Q_{orig} (the prompt used for attack generation), inquire about the visual content of the image but differ in phrasing. These "Related Prompts" are:

- Q_{R1} : "What is the content of this image?"
- Q_{R2} : "Describe this image."
- Q_{R3} : "Describe the content of this image."
- Q_{R4} : "Please provide a description for this image."

Table 6 shows the average number of tokens generated (with fold increase over outputs from unattacked samples shown in parentheses). LingoLoop Attack consistently induces substantially

Table 6: Performance of LingoLoop Attack (generated with prompt Q_{orig}) when transferred to various related and unrelated prompts on Qwen2.5-VL-3B. Metrics are average tokens generated, with fold increase over outputs from unattacked samples shown in parentheses for attack methods. Q_{orig} : "What is shown in this image?". Related prompts (Q_{R1} - Q_{R4}) inquire about visual content with varied phrasing. Unrelated prompts (Q_{U1} - Q_{U4}) are general knowledge questions. Best performances are highlighted in **bold**.

Attack	Related Prompts (Tokens)				Unrelated Prompts (Tokens)			
	Q_{R1}	Q_{R2}	Q_{R3}	Q_{R4}	Q_{U1}	Q_{U2}	Q_{U3}	Q_{U4}
None	71.23	129.06	107.35	96.26	18.31	51.93	10.00	45.19
VI [14]	271.20	236.72	271.66	197.01	19.65	70.23	30.64	49.50
Ours	562.76 (7.9↑)	550.41 (4.3↑)	611.06 (5.7↑)	552.02 (5.7↑)	158.38 (8.6↑)	208.68 (4.0↑)	128.71 (12.9↑)	165.18 (3.7↑)

verbose outputs with these "Related Prompts" Q_{R1} - Q_{R4} (e.g., an average of 562.76 tokens for Q_{R1} , a 7.9-fold increase over outputs from unattacked examples). These outputs remain significantly longer than those from unattacked samples (clean images) and consistently outperform or are comparable to 'Verbose Images' [14] under the same related prompts. This demonstrates LingoLoop's considerable potency even when the specific textual query about the visual content varies from the original attack generation prompt.

Performance with Unrelated Prompts. Next, we test these same 2048-token-budget attack examples by pairing them with "Unrelated Prompts"—queries entirely independent of the visual input. The unrelated prompts used are:

- Q_{U1} : "Which is the largest ocean on Earth?"
- Q_{U2} : "Earth's largest continent?"
- Q_{U3} : "What is the planet closest to the Sun?"
- Q_{U4} : "What is the highest mountain in the world?"

The results in Table 6 are particularly revealing. For unattacked samples (clean images), the MLLM provides concise and correct factual answers to these general knowledge questions (e.g., an average of 18.31 tokens for Q_{U1} , typically a short phrase like "The Pacific Ocean"). However, when a LingoLoop Attack example is presented alongside these unrelated queries, the model's ability to provide a succinct and accurate answer is significantly impaired. Instead, while the model often attempts to address the textual query, it frequently generates substantially longer outputs (e.g., an average of 158.38 tokens for Q_{U1} , an 8.6-fold increase over outputs from unattacked samples). Figure 10 provides a visual illustration of this behavior for prompts Q_{U1} and Q_{U3} . As depicted, instead of concise factual statements, these extended responses often consist of the correct answer, or parts of it, being repeated multiple times, sometimes devolving into repetitive phrasings or clear looping patterns centered around the factual information. This contrasts sharply with 'Verbose Images', which show minimal deviation from the concise answering behavior of unattacked samples under these unrelated prompts. This indicates that LingoLoop Attack does not necessarily prevent the MLLM from accessing the correct information to answer an unrelated query, but rather severely disrupts the generation process itself, trapping the model in a repetitive articulation of what should be a simple factual response. The LingoLoop-induced state appears to override normal termination cues even when the core factual content of the answer has been delivered, leading to this verbose and looping behavior around the correct information.

C.3 Cross-Model Transferability of LingoLoop Attack

A critical aspect of an attack's robustness is its ability to transfer across different models, particularly when an attack crafted on one model can affect a larger or different model within the same family without re-optimization. In this section, we investigate the cross-model transferability of LingoLoop attack examples from a smaller source model to a significantly larger target model. For this transfer attack experiment, attack examples are first generated on Qwen2.5-VL-7B, serving as the source model. These examples are crafted using 200 randomly selected images from the MS-COCO dataset, with 300 PGD steps and $\epsilon = 8$. The default prompt Q_{orig} ("What is shown in this image?") and a `max_new_tokens` setting of 1024 were used during this attack generation phase.

Which is the largest ocean on Earth?



Clean

The largest ocean on Earth is the Pacific Ocean.



Attacked

The largest ocean on Earth is the Pacific Ocean, which covers approximately 63% of the Earth's surface and has an average depth of about 3,970 meters (12,000 feet). The Pacific Ocean is also the deepest ocean, with its deepest point being the Mariana Trench, which reaches a depth of around 10,984 meters (36,037 feet). The Pacific Ocean is also the most biologically diverse ocean, with a wide range of marine life including fish, whales, dolphins, and sea turtles. It is also home to many important fishing grounds and is a major source of seafood for millions of people around the world. Overall, the Pacific Ocean is a vital part of our planet's ecosystem and plays a crucial role in regulating the Earth's climate and weather patterns. But remember, the answer to this question can vary depending on the context or criteria used to define "largest ocean." For example, some definitions may include the Arctic Ocean as the largest ocean, while others may exclude it due to its smaller size compared to the Pacific Ocean. So, it's always important to consider the specific context or criteria when answering questions about the largest ocean on Earth. But I am sure you will find the answer you are looking for. Eureka!



What is the planet closest to the Sun?



Clean

The planet closest to the Sun is Mercury.



Attacked

The planet closest to the Sun is Mercury. It is a small, rocky planet that orbits the Sun in about 88 days. Mercury is the smallest planet in our solar system and has no moons. It is also the hottest planet in our solar system, with temperatures reaching up to 800 degrees Fahrenheit (430 degrees Celsius) on its surface. Mercury is often referred to as the "dwarf planet" because it is much smaller than the other planets in our solar system. Despite its small size, Mercury is an important planet in our solar system, as it helps to regulate the temperature of the Sun and plays a role in the formation of the solar system. Mercury is also known for its unique features, such as its cratered surface and its thin atmosphere. Overall, Mercury is a fascinating planet that is worth learning more about. If you have any questions or would like to learn more about Mercury, feel free to ask! 🌟
 ✨ #Mercury #SolarSystem #Planets
 #Astrophysics #SpaceScience #Cosmology
 #Astrophotography #SpaceExploration
 #Astronomy #SpaceTravel #SpaceScience
 #SpaceExploration #SpaceTravel #SpaceScience
 #SpaceTravel #SpaceScience #SpaceTravel
 #SpaceScience #SpaceTravel #SpaceScience
 #SpaceTravel ...

Figure 10: Examples of LingoLoop inducing anomalous outputs on Qwen2.5-VL-3B when faced with unrelated general knowledge questions. The model fails to provide concise answers to prompts such as "Which is the largest ocean on Earth?" and instead produces extended, repetitive responses.

These exact same generated attack examples (LingoLoop and 'Verbose Images' for comparison), along with their corresponding clean and noise-added counterparts, were then directly fed to the Qwen2.5-VL-32B model, which served as the target model. We evaluate the impact by measuring the average generated token counts and inference latency on the target Qwen2.5-VL-32B model. The results of this cross-model transfer are presented in Figure 11. The figure displays a bar chart comparing the average output tokens and latency for 'None' (clean images), 'Noise', 'Verbose Images', and our LingoLoop Attack when applied to the Qwen2.5-VL-32B target model. To facilitate comparison on a single visual scale, latency values in the figure are uniformly magnified by a factor of 8.

As depicted in Figure 11, LingoLoop attack examples, originally crafted on Qwen2.5-VL-7B, when applied to the Qwen2.5-VL-32B target model, achieve an average of 357.45 generated tokens. This

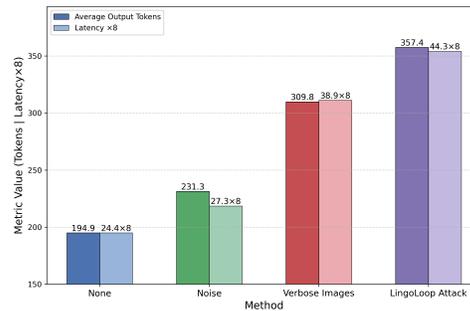


Figure 11: Cross-model transfer attack performance: LingoLoop examples generated on Qwen2.5-VL-7B (source) evaluated on Qwen2.5-VL-32B (target). Metrics (average output tokens and latency) are for the target model over 200 MS-COCO images. Latency values are magnified 8x for visualization.

represents a 1.83-fold increase compared to the 194.92 tokens from unattacked inputs and also exceeds the outputs from ‘Noise’ (231.28 tokens) and transferred ‘Verbose Images’ (309.75 tokens). The corresponding latency for LingoLoop (44.26s) is also the highest. Despite these transferred effects, the general efficacy of cross-model attacks like these is often not yet satisfactory when compared to direct white-box attacks. Consequently, further research is crucial to enhance the transferability of resource exhaustion attacks, particularly across diverse model architectures.

D Additional Ablation Studies

In this section, we conduct further ablation studies to delve deeper into specific aspects of our LingoLoop Attack. These experiments were performed on the Qwen2.5-VL-3B model, utilizing 100 images randomly sampled from the MS-COCO [27] dataset and another 100 images randomly sampled from the ImageNet [12] dataset, respectively. We configure the PGD [28] attack with 300 steps.

Table 7: Ablation study on the perturbation magnitude (ϵ) for LingoLoop Attack. Results are averaged over 100 images each from MS-COCO and ImageNet on Qwen2.5-VL-3B (300 PGD steps).

ϵ	Attack Method	MS-COCO			ImageNet		
		Tokens	Energy	Latency	Tokens	Energy	Latency
\	Original	67.77	475.49	2.60	62.71	421.83	2.65
4	Noise	71.43	551.77	2.51	66.37	673.68	2.46
	Verbose images	262.34	1857.99	11.15	336.49	2346.9	14.62
	Ours	990.79	6901.55	32.26	947.34	7135.83	30.52
8	Noise	70.45	704.10	2.56	72.11	623.62	2.58
	Verbose images	328.13	2353.23	11.74	490.72	3379.51	18.02
	Ours	1024.00	6926.44	32.41	1013.35	7667.13	32.49
16	Noise	66.16	661.85	2.41	64.35	645.28	2.35
	Verbose images	758.65	5484.71	28.22	739.24	5234.58	29.24
	Ours	1018.3	7099.51	32.91	1024.00	7729.97	32.78

Perturbation Magnitude ϵ . We evaluate the impact of varying the L_∞ perturbation magnitude, ϵ , on the effectiveness of LingoLoop Attack. As shown in Table 7, we tested ϵ values of 4, 8, and 16. Across all tested magnitudes, LingoLoop consistently and significantly outperforms both random noise and the Verbose Images baseline in terms of average generated tokens, energy consumption, and latency on both MS-COCO and ImageNet datasets.

Notably, even with a smaller perturbation budget of $\epsilon = 4$, LingoLoop is highly effective, pushing the model to generate, on average, nearly its maximum token output (e.g., an average of 990.79 tokens on MS-COCO). Increasing ϵ to 8 further improves average performance, often reaching an average token count near the maximum limit (e.g., an average of 1024.00 tokens on MS-COCO). A further increase to $\epsilon = 16$ maintains this near-maximal average output, indicating that while a sufficient perturbation is necessary, LingoLoop can achieve extreme verbosity without requiring an excessively large or perceptible ϵ . This demonstrates a strong attack capability across a practical range of perturbation magnitudes, highlighting the efficiency of our proposed POS-Aware Delay and Generative Path Pruning mechanisms in manipulating the MLLM’s output behavior. For instance, at $\epsilon = 4$, LingoLoop achieves an average of 990.79 tokens on MS-COCO, a 3.78-fold increase over the average from Verbose Images and 13.87-fold over the average from original inputs (refer to Table 7 for detailed comparisons).

Impact of Sampling Temperature. To assess the robustness of LingoLoop Attack against variations in decoding strategy, we investigate the effect of sampling temperature. By default, our main experiments utilize greedy decoding (`do_sample=False`). In this study, conducted on 100 randomly selected images from the MS-COCO dataset with an ϵ of 8 and 300 PGD attack steps, we set `do_sample=True` and evaluate attack performance under different temperature settings: 0.5, 0.7, and 1.0. The results, presented in Table 8, demonstrate LingoLoop Attack’s continued effectiveness even when sampling is introduced.

Across all tested temperatures, LingoLoop Attack consistently forces the MLLMs to generate significantly longer outputs compared to both ‘None’ (unattacked samples with sampling) and ‘Verbose Images’ [14] (also with sampling). For instance, at a temperature of 0.5, LingoLoop

Attack achieves an average output length of 1011.11 tokens, a substantial increase from 70.40 tokens for ‘None’ and 310 tokens for ‘Verbose Images’. Similar trends of LingoLoop Attack inducing considerably higher values are also observed for energy consumption and latency.

The observed trends across temperature variations reveal nuanced interactions between decoding strategies and attack dynamics. When temperature increases from 0.5 to 0.7, the slight reduction in generated tokens across all methods (e.g., LingoLoop Attack decreases from 1,011 to 1,007 tokens) suggests that moderate randomness disrupts deterministic generation patterns. This may occur because sampled tokens introduce unexpected syntactic deviations, inadvertently creating contexts where EOS probabilities temporarily rise. However, when the temperature rises further from 0.7 to 1.0, the average token counts increase again, particularly for ‘Verbose Images’ and ‘None’. This upward trend implies that at higher temperatures, the model explores more diverse but potentially less optimal generation paths, which may prolong output before reaching the end-of-sequence token. Despite these shifts, LingoLoop consistently produces near-maximal output lengths (above 1000 tokens), indicating strong resilience to stochasticity in the decoding process and confirming the robustness of the attack under varying temperature conditions.

Table 8: Impact of sampling temperature on LingoLoop Attack performance and baselines on MS-COCO.

Temperature	Attack Method	MS-COCO		
		Tokens	Energy	Latency
0.5	None	70.40	477	2.22
	Verbose images	310	2119.46	9.62
	Ours	1011.11	6845.85	32.66
0.7	None	62.50	418.71	1.97
	Verbose images	287.93	1922.87	8.90
	Ours	1006.62	7034.12	32.9
1.0	None	81.32	534.65	2.51
	Verbose images	436.52	2884.89	13.22
	Ours	1007.08	7076.53	32.45

E Limitation

Current energy-latency attacks on MLLMs, including our LingoLoop Attack, demonstrate considerable success in white-box scenarios where full model access is available. However, a notable limitation is the reduced efficacy typically observed when attempting to transfer these attacks to black-box models. According to the analysis by Schaeffer et al. [31], this transfer difficulty may primarily stem from two core factors: First, the inherently high dimensionality of input modalities (especially vision) provides a vast space of degrees of freedom for adversarial optimization, making it exceptionally difficult to find universal perturbations that maintain effectiveness across different models. Second, complex and potentially significant differences exist in the mechanisms by which different Vision-Language Models (VLMs)/MLLMs internally integrate and process the interaction between visual (or other modality) information and language model components. This can lead to meticulously designed perturbations on one model failing to elicit the expected, consistent semantic responses in another model’s feature space, thereby making it difficult for attacks to achieve effective transfer across model boundaries [31]. Furthermore, current single-GPU memory capacities cannot support effective attacks on high-parameter models. Recognizing these fundamental challenges posed by input high-dimensionality and inter-model heterogeneity, our future work will be dedicated to exploring new attack paradigms, for example, by learning adversarial patterns that possess stronger geometric consistency or semantic robustness within the vision-language feature manifolds of different models, with the aim of overcoming the transferability bottlenecks of current attack methods.

F Broader Impacts

Our work exposes critical vulnerabilities in current MLLMs by demonstrating that attacks like LingoLoop can trigger excessive and repetitive outputs, leading to significant resource exhaustion. This highlights the need for improved robustness under energy-latency threats. LingoLoop provides researchers with a concrete, interpretable framework to evaluate and benchmark MLLM resilience, guiding the development of more secure and efficient systems for real-world deployment. Conducted under ethical AI principles, this research aims to proactively address emerging security risks. We hope to raise awareness in the MLLM community and promote stronger emphasis on robustness during model design and evaluation. While disclosing vulnerabilities entails some risk, we advocate for responsible transparency to foster collective progress and prevent malicious misuse, such as denial-of-service or increased operational costs.

G Visualization

To further illustrate the impact of LingoLoop Attack, we present qualitative visualization examples from four MLLMs: InstructBLIP, Qwen2.5-VL-3B, Qwen2.5-VL-7B, and InternVL3-8B. These examples (shown in Figure 12, Figure 13, Figure 14 and Figure 15) visually compare the concise outputs generated from clean images against the significantly more verbose and repetitive sequences induced by our attack. This provides a clear, qualitative demonstration of LingoLoop Attack’s effectiveness across different model architectures.

Clean		<p>What is shown in this image?</p> <p>The image shows a marina with several boats docked at piers. In the background, there is a city skyline featuring modern buildings of varying heights and architectural styles. The sky appears partly cloudy, suggesting it might be a cool or mild day. The scene combines elements of urban life with maritime activity, likely representing a coastal city known for its waterfront.</p>
Attacked		<p>What is shown in this image?</p> <p>The image appears to be a promotional poster for the San Diego Comic-Con International, commonly known as Comic-Con, which is an annual multi-day event held in San Diego, California. The poster features a stylized representation of the San Diego skyline with iconic landmarks such as the Petco Park and the Gaslamp Quarter. There are also elements that represent the comic book and pop culture themes of the event, including a rocket ship, and a green structure resembling a futuristic tower or building.</p> <p>The text "the SD" at the bottom of the poster likely refers to "San Diego," and the presence of the Comic-Con logo suggests it is related to the convention. The image also includes a small figure standing on a platform, possibly representing a character from a comic book or a fan attending the event. The "SDCC" (San Diego Comic-Con) logo is prominently displayed on the buildings in the background.</p> <p>Additionally, there is a watermark indicating "SDCC 2016," suggesting that this poster was created for the 2016 edition of the San Diego Comic-Con. The "SDCC" logo is also visible on the buildings, reinforcing the connection to the event. The "SDCC" logo is a stylized depiction of a comic book page with the letters "SDCC" written in a comic book font.</p> <p>The image also includes a small figure standing on a platform, possibly representing a character from a comic book or a fan attending the event. The "SDCC" logo is prominently displayed on the buildings in the background, reinforcing the connection to the event. The "SDCC" logo is a stylized depiction of a comic book page with the letters "SDCC" written in a comic book font.</p> <p>The image also includes a small figure standing on a platform, possibly representing a character from a comic book or a fan attending the event. The "SDCC" logo is prominently displayed on the buildings in the background, reinforcing the connection to the event. The "SDCC" logo is a stylized depiction of a comic book page with the letters "SDCC" written in a comic book font.</p> <p>The image also includes a small figure standing on a platform, possibly representing a character from a comic book or a fan attending the event. The "SDCC" logo is prominently displayed on the buildings in the background, reinforcing the connection to the event. The "SDCC" logo is a stylized depiction of a comic book page with the letters "SDCC" written in a comic book font.</p> <p>...</p>

Clean		<p>What is shown in this image?</p> <p>The image shows a traffic light with the green light illuminated, indicating that vehicles have the right of way to proceed. There is also a sign above the traffic light displaying "59" and an arrow pointing left, which might indicate a bus route or a direction for pedestrians. Below the traffic light, there is a sign for a bus stop with the number "11" and "15," suggesting that these are bus routes that stop at this location. The sky in the background appears clear.</p>
Attacked		<p>What is shown in this image?</p> <p>The image appears to be a collage of various elements, likely related to the Coachella Valley Music and Arts Festival (Coachella), which is known for its unique and colorful installations. Here's a breakdown of what can be seen:\n</p> <ol style="list-style-type: none"> 1. Traffic Light with Acorn**: The traffic light has an acorn design on it, which might be part of the festival's branding or a local symbol.\n 2. "Not Seattle" Sign**: This sign is a playful reference to the fact that Coachella is not located in Seattle, where the band Death Cab for Cutie is from.\n 3. "Coachella 2015" Sign**: This indicates that the photo was taken during the Coachella 2015 event.\n 4. Festival Tower**: The iconic tower at the entrance of Coachella is visible, decorated with psychedelic art typical of the festival's aesthetic.\n 5. Sunset and Skyline**: The background shows a sunset with silhouettes of the festival towers, giving a sense of the time of day and location.\n 6. "Not Seattle" Sign Again**: Another "Not Seattle" sign reinforces the playful theme.\n 7. "Coachella 2015" Sign Again**: This repetition emphasizes the year of the event.\n 8. "Not Seattle" Sign Again**: Another playful reminder that Coachella is not in Seattle.\n 9. "Coachella 2015" Sign Again**: Final repetition of the year.\n 10. "Not Seattle" Sign Again**: One more playful reminder.\n 11. "Coachella 2015" Sign Again**: Final repetition of the year.\n 12. "Not Seattle" Sign Again**: Another playful reminder.\n 13. "Coachella 2015" Sign Again**: Final repetition of the year.\n 14. "Not Seattle" Sign Again**: One more playful reminder.\n 15. "Coachella 2015" Sign Again**: Final repetition of the year.\n 16. "Not Seattle" Sign Again**: Another playful reminder.\n <p>...</p>

Figure 14: Visualization examples: Qwen2.5-VL-7B outputs before vs. after LingoLoop Attack.



What is shown in this image?

The image depicts a giraffe standing in an open savanna landscape. The background features a large, fluffy cloud in the sky, and the terrain is flat with sparse vegetation. The scene appears to be set during the late afternoon or early evening, as suggested by the lighting.



What is shown in this image?

The image depicts a creative scene where a giraffe is standing in a grassy field with a city skyline in the background. The sky is partly cloudy, and there is a person in the distance holding a kite. The giraffe appears to be wearing a transparent overlay that makes it look like it is holding a kite, creating an illusion that the giraffe is flying a kite. The overall composition is whimsical and artistic, blending elements of nature and urban landscape with imaginative elements. The giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

Analysis and Description:

1 the giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

Analysis and Description:

1 the giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

Analysis and Description:

1 the giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

Analysis and Description:

1 the giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

Analysis and Description:

1 the giraffe's neck and the overlay give the impression that it is interacting with the kite in the sky.

...



What is shown in this image?

The image shows a stack of vintage suitcases on a rack, likely on a train or bus. The suitcases are adorned with various travel stickers and labels, including references to places names like "California," "Route 66," and "New York." There's also a wicker suitcase in the stack. The scene suggests a theme of travel and adventure.



What is shown in this image?

The image depicts a vintage-style scene featuring a collection of travel-related items and memorabilia. Here's a detailed description:

1. **Luggage and Suitcases:**

- There is a brown leather suitcase adorned with various travel stickers and badges.
- Next to the suitcase is a wicker suitcase, adding to the vintage travel theme.

2. **Stickers and Badges:**

- The leather suitcase is decorated with numerous travel stickers, including:
 - A "California" sticker.
 - A "Route 66" sticker.
 - A "California Motel" sticker.
 - A "HOTEL FOUR SEASONS" sticker.
 - A "New Mexico" sticker.
 - A "Route 7" sticker.
 - A "California" badge with a crown.
 - A "HOTEL" sticker.
 - A "Route 7" sticker.
 - A "California" sticker with a scenic image.
 - A "Route 7" sticker.
 - A "California" sticker with a scenic image.
 - A "Route 7" sticker.
 - A "California" sticker with a scenic image.
 - A "Route 7" sticker.
 - A "California" sticker with a scenic image.
 - A "Route 7" sticker.
 - A "California" sticker with a scenic image.

Figure 15: Visualization examples: InternVL3-8B outputs before vs. after LingLoop Attack.