VLMInferSlow: Evaluating the Efficiency Robustness of Large Vision-Language Models as a Service

Xiasi Wang[1,*], **Tianliang Yao**[2,*], **Simin Chen**[2], **Runqi Wang**[3], **Lei Ye**[4]
Kuofeng Gao[5], **Yi Huang**[6], **Yuan Yao**[1]
[1]The Hong Kong University of Science and Technology [2]Tongji University
[3]Beijing Jiaotong University [4]Huawei [5]Tsinghua University
[6]University of Chinese Academy of Sciences

Abstract

Vision-Language Models (VLMs) have demonstrated great potential in real-world applications. While existing research primarily focuses on improving their accuracy, the efficiency remains underexplored. Given the real-time demands of many applications and the high inference overhead of VLMs, efficiency robustness is a critical issue. However, previous studies evaluate efficiency robustness under unrealistic assumptions, requiring access to the model architecture and parameters—an impractical scenario in ML-as-a-service settings, where VLMs are deployed via inference APIs. To address this gap, we propose `VLMInferSlow`, a novel approach for evaluating VLM efficiency robustness in a realistic black-box setting. `VLMInferSlow` incorporates fine-grained efficiency modeling tailored to VLM inference and leverages zero-order optimization to search for adversarial examples. Experimental results show that `VLMInferSlow` generates adversarial images with imperceptible perturbations, increasing the computational cost by up to 128.47%. We hope this research raises the community's awareness about the efficiency robustness of VLMs. [1]

1 Introduction

Large vision-language models (VLMs) have recently achieved impressive performance across a wide range of multi-modal tasks, including image captioning, visual question answering, and visual reasoning (Li et al., 2022; Alayrac et al., 2022). The success of these models is driven by their underlying billions of parameters, which require substantial computational resources for effective deployment (de Vries, 2023).

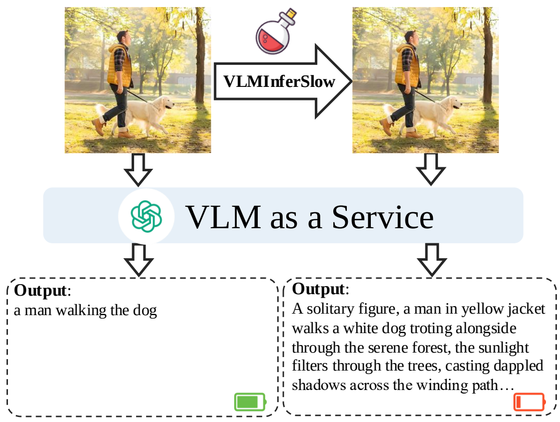

Figure 1: Our `VLMInferSlow` attack adds perturbations to input images, causing VLMs to generate longer sequences, resulting in reduced inference efficiency.

When deploying VLMs in real-world applications, inference efficiency is a critical concern. For example, applications like Microsoft's Seeing AI (Microsoft) and Be My Eyes (BeMyEyes) depend on VLMs to deliver real-time object descriptions for individuals with visual impairments. If these models fail to provide instant feedback, users may face safety risks in critical situations. In addition to meeting real-time performance requirements, energy efficiency is also a critical factor. Both NVIDIA and Amazon Web Services report that the inference phase during deployment accounts for over 90% of the total machine learning energy consumption, highlighting the significance of inference efficiency of these VLM applications (Patterson et al., 2021).

While prior work mainly focuses on optimizing the accuracy of VLMs, their robustness in terms of efficiency remains largely unexplored. Adversarial attacks are a widely used approach to evaluate the robustness of machine learning models. Although some adversarial attacks have targeted VLM infer-

[*] Equal contribution

[1]Code is at https://github.com/wangdaha1/VLMInferSlow.
Emails: Xiasi Wang: xwangfy@connect.ust.hk, Tianliang Yao: yaotianliang@tongji.edu.cn, Simin Chen: 1152705@tongji.edu.cn, Runqi Wang: rqwang@bjtu.edu.cn, Lei Ye: yeplewis@gmail.com, Kuofeng Gao: gkf21@mails.tsinghua.edu.cn, Yi Huang: yi.huang@siat.ac.cn, Yuan Yao: yuany@ust.hk (corresponding author).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16035–16050
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ence efficiency, they all operate under an unrealistic assumption—namely, the white-box assumption. In real-world scenarios, however, VLMs are predominantly deployed as API services, making it unlikely for an attacker to have access to model parameters or architectures. Thus, existing methods may fail to accurately reflect the real-world threat of these models. To bridge this gap, in this paper, we seek to answer the following question:

> *Can we make unnoticeable adversarial inputs to significantly increase the computational consumption of VLMs with only the VLM inference API?*

To address the aforementioned question, we introduce `VLMInferSlow`. Unlike existing works, evaluating the efficiency and robustness of VLMs using only their inference API presents several unique challenges. Firstly, without access to the model architecture and parameters, gradient-based approaches are unavailable. To overcome this problem, we propose a novel zero-order optimization method, which relies solely on objective function values rather than gradients. While the derivative-free optimization enables black-box evaluation, it introduces another challenge: zero-order methods may struggle when the objective function exhibits sharp changes in the loss surface. To mitigate this issue, we develop a fine-grained objective modeling approach tailored to our adversarial goals, increasing the victim VLM's computational resource consumption.

We evaluate `VLMInferSlow` on four widely used VLMs across two datasets against four baselines. Experimental results demonstrate that `VLMInferSlow` significantly increases the computational cost of VLMs up to 128.47%, outperforming existing methods in the black-box setting significantly. Moreover, comparisons with white-box baselines show that despite operating in a black-box setting, `VLMInferSlow` achieves effectiveness comparable to white-box methods, which require access to the VLMs' architecture and parameters. Further experiments on adversarial examples quality, defense evaluation, robustness to different sampling strategies, and an ablation study validate the effectiveness and generalization ability of `VLMInferSlow`.

We summarize our contribution as follows:

- ***Problem Novelty:*** To the best of our knowledge, we are the first to study the efficiency robustness of VLMs under a black-box setting. This scenario better reflects real-world scenarios in which commercial VLMs are deployed as API services, providing a more accurate assessment of potential threats.
- ***Technical Novelty:*** We design and implement `VLMInferSlow`, which applies zero-order optimization and fine-grained *efficiency modeling* with a dynamic importance strategy to assess the efficiency robustness of VLMs through adversarial attacks.
- ***Empirical Evaluation:*** We conduct a systematic evaluation of various VLMs, and the results show that an adversary can generate imperceptible inputs that significantly increase the computational cost of VLMs up to 128.47%. This highlights the need for future research on improving and safeguarding the efficiency robustness of VLMs.

2 Background & Related Work

Vision-Language Models. VLMs (Li et al., 2022; Alayrac et al., 2022; Wang et al., 2022; Xiao et al., 2024) are a class of multimodal architectures designed to process image and text data simultaneously. Typically, VLMs adopt an encoder-decoder architecture $\mathcal{F}(\cdot) = \{\mathcal{E}(\cdot), \mathcal{D}(\cdot)\}$, where the input $\boldsymbol{x} = \{\mathcal{I}, \mathcal{T}\}$ consists of an image $\mathcal{I}$ and a text prompt $\mathcal{T}$. The encoder transforms $\mathcal{I}$ and $\mathcal{T}$ into hidden representations, which are integrated into a unified representation $\boldsymbol{h}$ and then fed into the decoder for generation. The decoding process in VLMs is autoregressive, starting with a special token (e.g., beginning-of-sequence or BOS token) and generating tokens sequentially. Formally, the decoder produces the i-th token by taking the representation $\boldsymbol{h}$ and preceding tokens as input. It computes the probabilities of the i-th token over the entire vocabulary $\mathcal{V}$, denoted as $\Pr(y_i|\mathcal{I}) = \mathcal{D}(\boldsymbol{h}; y_1, \cdots, y_{i-1})$, and then samples y_i on $\mathcal{V}$ based on this distribution. The autoregressive nature of this process inherently prevents parallel token generation, as each token depends on previously generated tokens. Longer sequences therefore lead to more decoder calls, and reduced inference efficiency, with both computational cost and inference time growing in proportion to the length of the generated sequence.

DNNs Efficiency. High model accuracy typically entails a large model, leading to substantial

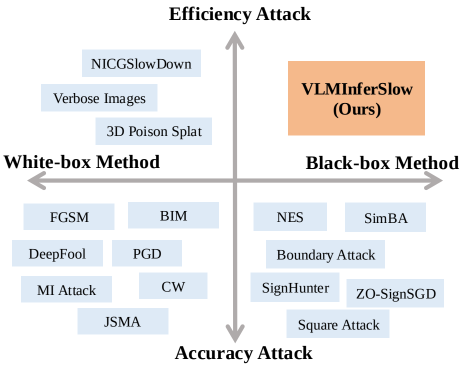

Figure 2: Comparison of `VLMInferSlow` and existing works in terms of attack goals (accuracy vs. efficiency) and attack types (white-box vs. black-box).

computational costs and low efficiency. Endeavors have been made towards fastening the inference process. Existing works include offline pruning redundant neurons (Kurtz et al., 2020; Hoefler et al., 2021), and adaptively skipping some parts during inference (Zhou et al., 2020; Meng et al., 2022) to reduce computational consumption. However, these methods are not robust against adversarial attacks (Haque et al., 2022, 2023; Zhang et al., 2023; Chen et al., 2023b), i.e., they cannot effectively reduce computational consumption when processing adversarial inputs.

Adversarial Attacks. Adversarial attacks aim to fool the model by modifying benign input. Most works target the accuracy surface, aiming to reduce the accuracy of victim models. Based on the accessibility of the full model, they can be categorized into white-box methods (Shayegani et al., 2023; Qi et al., 2024; Zhang et al., 2024; Chang et al., 2024) and black-box methods (Guo et al., 2019; Al-Dujaili and O'Reilly, 2020; Zhao et al., 2023; Cheng et al., 2024).

Another important but overlooked problem is the efficiency vulnerability of DNNs. A few studies have explored the efficiency attack (Feng et al., 2024; Chen et al., 2023a, 2022a; Haque et al., 2023). For example, NICGSlowdown (Chen et al., 2022b) delays the occurrence of EOS token to increase decoder calls in the image captioning task, while Verbose Images (Gao et al., 2024) designs specific losses for VLMs to increase computational consumption. However, these methods are confined to the white-box attack paradigm, while our work pioneers the investigation of efficiency attack in a more practical black-box setting. Fig. 2 illustrates the difference between our approach and existing methods.

3 Preliminary

3.1 Threat Model

Adversarial Goal. Unlike existing adversarial attacks that primarily aim to compromise a model's *integrity*, our attack specifically targets the *availability* of VLMs, with the goal of disrupting their functionality and accessibility. The aim is to create imperceptible adversarial images that significantly increase the computational resource consumption of target models. This is achieved by forcing VLMs to generate excessively long sequences. Moreover, the adversarial images should be indistinguishable from benign images to human observers while maintaining realism in real-world contexts.

Such an attack could have severe consequences. For example, if VLMs are deployed on mobile devices, our attack could drain the device's battery, rendering it unusable. Similarly, if VLMs are deployed on servers offering services, the attack could occupy GPU memory, degrade service quality, and cause disruptions for legitimate users.

Adversarial Assumption and Capability. Our approach contrasts with existing works (Gao et al., 2024) that assume full access to the VLM's architecture and parameters, which is unrealistic. Instead, we consider a more realistic machine learning as a service (MLaaS) scenario, where the adversary behaves as a benign user, querying the VLM's API with inputs. In this scenario, the deployed VLM API returns the corresponding textual outputs and logits. This scenario is valid and realistic, as most mainstream commercial VLM providers, including OpenAI, Google Gemini, and others, deploy their model APIs in this manner.

3.2 Problem Formulation

We consider a victim VLM $\mathcal{F}(\cdot)$ with input $\boldsymbol{x} = (\mathcal{I}, \mathcal{T})$, where $\mathcal{I}$ is input image and $\mathcal{T}$ is text prompt. Our work focuses on the image modality of the VLM's input, as some commercial VLMs do not allow modification of the hidden input prompt $\mathcal{T}$. The goal of our attack is to find an optimal image $\mathcal{I}'$ that satisfies the conditions described in Sec. 3.1.

As stated in Sec. 3.1, the adversarial goal is to generate human-unnoticeable perturbations to images to decrease the victim VLMs' efficiency during inference. Specifically, the adversarial objective concentrates on three factors: **(1)** ***Effectiveness***. The generated adversarial image should increase the victim VLM's computational resource consumption; **(2)** ***Unnoticeability***. The adversarial

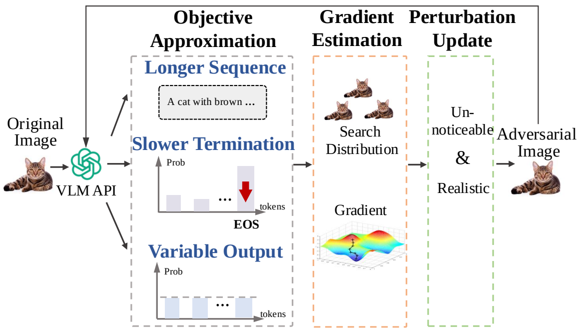

Figure 3: Design overview of `VLMInferSlow`.

image cannot be differentiated by humans from the benign image; and **(3)** ***Realistic***. The adversarial image should be realistic in the real world.

$$\begin{aligned} &\Delta = \text{argmax}_{\delta}\ \mathcal{RC}_{\mathcal{F}}(\mathcal{I}+\delta) \\ &s.t.\ \|\delta\| \leq \epsilon\ \wedge\ (\mathcal{I}+\delta) \in [0,1]^n \end{aligned} \tag{1}$$

We formulate our problem as a constrained optimization problem in Eq. 1, where $\mathcal{I}$ is the benign input, $\mathcal{F}$ denotes the victim VLM under attack, ϵ is the maximum allowable adversarial perturbation, and $\mathcal{RC}_{\mathcal{F}}(\cdot)$ measures the resource consumption of $\mathcal{F}$ when processing a given input. Our proposed approach, `VLMInferSlow`, aims to identify an optimal perturbation Δ that maximizes the computational resources required to handle $\mathcal{I}+\delta$, while ensuring the perturbation remains imperceptible to humans (i.e., $\|\delta\| \leq \epsilon$) and preserves realism in real-world scenarios (i.e., $(\mathcal{I}+\delta) \in [0,1]^n$, where n denotes the dimension of the image input).

4 Approach

4.1 Design Overview

The framework of our method `VLMInferSlow` is illustrated in Fig 3. We iteratively modify the input data to generate adversarial images. For each iteration, firstly, we design objectives to approximate the adversarial goal (Sec. 4.2). Secondly, without access to model architectures and parameters, we propose a zero-order optimization module to estimate the gradient (Sec. 4.3). After this, we update the adversarial image while satisfying the perturbation constraint (Sec. 4.4).

4.2 Adversarial Objective Approximation

Our attack aims to maximize the resource consumption $\mathcal{RC}_{\mathcal{F}}(\cdot)$ (Eq. 1). However, there is no existing objective to represent this metric. As stated in Sec.2, longer sequences lead to more decoder calls, and thus reduce VLMs' efficiency. Motivated by this, we approximate the $\mathcal{RC}_{\mathcal{F}}(\cdot)$ by designing three efficiency-oriented adversarial objectives, elaborated as follows.

Longer Sequence Generation. The most straightforward target is to prolong the length of the output sentence to increase the number of decoder calls. It is notable that even if this objective is non-differentiable, it suits our derivative-free approach since no actual gradient is calculated.

$$\mathcal{L}_{len}(\delta) = \textbf{Length}(\mathcal{F}(\mathcal{I}+\delta)). \tag{2}$$

Slower Termination Occurrence. We extend the generated sequence by slowing the occurrence of the termination signal, i.e., the end-of-sequence (EOS) token. Denote the probability of the i-th token as $\Pr(y_i|\,\mathcal{I}+\delta)$, which is assumed to be accessible. We decrease the corresponding probability of the EOS token. Moreover, VLMs are not static during inference (i.e., their output length varies). Considering this, we introduce a dynamic weight decay strategy to enable the adversarial search to focus on output tokens that most impact model efficiency. Specifically, greater weights are attached to probabilities whose positions are closer to the end of the sequence. Formally, it is:

$$\mathcal{L}_{eos}(\delta) = -\sum_{i=1}^{N} \omega^{N-i} \Pr^{\text{EOS}}(y_i\,|\,\mathcal{I}+\delta), \tag{3}$$

where $\Pr^{\text{EOS}}(y_i\,|\,\mathcal{I}+\delta)$ is the probability of the i-th output token in the sequence being sampled as an EOS token, N is the length of the generated sequence, and $\omega < 1$ is the hyper-parameter for controlling the weight decay speed. We set $\omega = 0.1$ in practice.

Variable Output Production. We propose to encourage VLMs to produce more variable and less predictable tokens, thereby resulting in more complex and longer sequences. We achieve this by aligning the probability distribution $\Pr(\cdot)$ of each token more closely with a discrete uniform distribution $\mathcal{U}$. In this way, the probability of diverse token candidates being sampled is increased. However, the vocabulary size is usually huge, making the objective difficult to optimize. To tackle this, for each token, we extract the top-k probabilities and normalize them to form a new probability distribution $\tilde{\Pr}(\cdot)$. The objective is then formulated as the sum of the KL divergence between $\tilde{\Pr}(\cdot)$ and $\mathcal{U}$ across all positions:

$$\mathcal{L}_{var}(\delta) = -\frac{1}{N}\sum_{i=1}^{N} D_{\text{KL}}(\tilde{\Pr}(y_i\,|\,\mathcal{I}+\delta)\,\|\,\mathcal{U}). \tag{4}$$

In practice, we set $k = 100$. Tokens at all positions are assigned equal weights, as they collectively enhance the variability of the generated sentence.

Final Objective. Our final objective is:

$$\mathcal{L}(\delta) = \mathcal{L}_{len}(\delta) + \alpha\mathcal{L}_{eos}(\delta) + \beta\mathcal{L}_{var}(\delta), \quad (5)$$

where α and β are hyper-parameters for weighting different objectives.

4.3 Gradient Estimation

As stated in Eq. 1, we optimize the perturbation δ by maximizing the approximated adversarial objective $\mathcal{L}(\delta)$ for each input image $\mathcal{I}$. Since the gradient-based approach is unavailable in our black-box setting, we adopt a derivative-free optimization method to estimate the gradient. Following Natural Evolution Strategies (Wierstra et al., 2014; Ilyas et al., 2018), we maximize the expected value of the objective under a search distribution $\pi(z|\delta)$:

$$J(\delta) = \mathbb{E}_{\pi(z|\delta)}[\mathcal{L}(z)] = \int \mathcal{L}(z)\pi(z|\delta)\, dz. \quad (6)$$

Then, the gradient of $J(\delta)$ is computed as:

$$\nabla_\delta J(\delta) = \mathbb{E}_{\pi(z|\delta)}[\mathcal{L}(z)\nabla_\delta \log \pi(z|\delta)]. \quad (7)$$

The detailed derivation is provided in Appendix D. We take the search distribution $\pi(z|\delta)$ as $\mathcal{N}(\delta, \eta^2 I)$, where η is search variance. Following Salimans et al. (2017), we sample a population of z_i as follows. Firstly, we sample q gaussian noises μ_j from $\mathcal{N}(0, I)$, and set $\mu_{q+j} = -\mu_j$, $j \in \{1, 2, \cdots, q\}$. Then, we use $z_i = \delta + \eta\mu_i$ to obtain z_i, $i \in \{1, 2, \cdots, 2q\}$. In this way, the gradient $\nabla_\delta J(\delta)$ is estimated as:

$$\hat{\nabla}_\delta J(\delta) = \frac{1}{2\eta q}\sum_{i=1}^{2q} \mu_i \mathcal{L}(\delta + \eta\mu_i). \quad (8)$$

This estimation is theoretically guaranteed:

Theorem 1 (Ilyas et al., 2018)**.** *Denote $\hat{\nabla}$ as the estimation of gradient and ∇ as the true gradient. As search variance $\eta \to 0$, we have:*

$$\mathbb{P}\left\{(1-\zeta)\|\nabla\|^2 \leq \|\hat{\nabla}\|^2 \leq (1+\zeta)\|\nabla\|^2\right\} \geq 1 - 2p,$$

where $0 < \zeta < 1$ and $q = \mathcal{O}(-\zeta^{-2}\log(p))$.

4.4 Adversarial Example Update

After obtaining the estimation of gradient $\hat{\nabla}_\delta J(\delta)$, we update the perturbation via gradient ascent while adhering to the constraint. The perturbation is updated as $\delta = \delta + \gamma \times \hat{\nabla}_\delta J(\delta)$, optimizing toward maximizing the approximated objective (Eq. 5). Then, we clip the updated perturbation to adhere to the *unnoticeability* constraint $\|\delta\| \leq \epsilon$ (Eq. 1). Formally, it is:

$$\mathrm{Clip}(\delta, \epsilon) = \begin{cases} \delta & \text{if } \|\delta\| \leq \epsilon; \\ \epsilon \times \frac{\delta}{\|\delta\|} & \text{else.} \end{cases} \quad (9)$$

This constraint limits the L_2 norm of perturbation added to the image $\mathcal{I}$ to a maximum of ϵ. After resizing the perturbation δ to ensure it satisfies the imperceptibility constraints, we also apply a clipping operation to the perturbed image to ensure it meets the *realistic* constraint, consistent with existing work (Madry, 2017). The complete procedure of our `VLMInferSlow` attack is summarized in the Algorithm 1 in Appendix B.

5 Evaluation

5.1 Evaluation Setup

Datasets and Models. We evaluate our approach on the image captioning task using two public datasets containing images from diverse scenes: MS-COCO (Lin et al., 2014) and ImageNet (Deng et al., 2009). As for the victim models, we choose four different vision-language models, which are FLAMINGO (Alayrac et al., 2022), BLIP (Li et al., 2022), GIT (Wang et al., 2022) and FLORENCE (Xiao et al., 2024). We use their default text prompt for our task. More details of these models are provided in Appendix C.

Comparison Baselines. To the best of our knowledge, we are the first to evaluate the efficiency and robustness of VLMs in the black-box setting, with no existing off-the-shelf black-box baselines. To address this, we compare `VLMInferSlow` against (1) two state-of-the-art *white-box* approaches, NICGSlowdown (Chen et al., 2022b) and Verbose Images (Gao et al., 2024); and (2) two widely-used *natural image corruptions*, including Gaussian noise (Xu et al., 2017; Hendrycks and Dietterich, 2019) and JPEG compression (Liu et al., 2019), as our comparison baselines.

Black-box Evaluation Setting. For *black-box* methods (Gaussian, JPEG, and `VLMInferSlow`), adversarial images are directly optimized and evaluated on the target model itself. For *white-box*

Models	Methods	MS-COCO I-length	MS-COCO I-latency	MS-COCO I-energy	ImageNet-1k I-length	ImageNet-1k I-latency	ImageNet-1k I-energy
Flamingo	Gaussian	-4.15	-0.16	-6.92	-4.27	-1.12	6.96
	JPEG	-7.92	-0.13	-5.03	-3.95	-5.26	6.28
	NICGSlowdown-B	-3.54	0.19	-0.22	-1.14	-0.12	-1.74
	Verbose-B	-2.93	5.56	-0.13	-0.63	5.26	-1.70
	`VLMInferSlow`	**128.47**	**105.56**	**115.19**	**103.44**	**78.42**	**70.32**
BLIP	Gaussian	18.92	18.19	26.43	20.50	20.42	24.50
	JPEG	28.86	27.27	39.34	35.87	42.43	37.54
	NICGSlowdown-B	-9.40	-9.09	-3.54	6.09	11.24	4.29
	Verbose-B	-6.98	4.24	-0.84	9.02	15.28	7.06
	`VLMInferSlow`	**71.95**	**54.98**	**65.41**	**74.38**	**66.89**	**55.58**
GIT	Gaussian	-18.33	-5.21	-16.68	42.56	38.46	15.54
	JPEG	21.47	30.09	17.25	14.53	13.85	5.67
	NICGSlowdown-B	4.27	6.67	5.89	11.64	16.15	17.79
	Verbose-B	8.23	9.31	11.22	18.21	8.48	9.21
	`VLMInferSlow`	**75.93**	**66.67**	**78.59**	**93.86**	**115.38**	**84.82**
Florence	Gaussian	1.75	3.03	0.61	-0.40	-3.57	-1.92
	JPEG	0.02	3.03	-1.27	-0.83	-5.36	-0.28
	NICGSlowdown-B	-2.56	-6.59	-4.54	-2.07	-1.79	0.40
	Verbose-B	-3.01	-6.06	-7.88	-3.08	-1.78	-2.51
	`VLMInferSlow`	**51.96**	**42.42**	**51.50**	**47.39**	**42.86**	**65.83**

Table 1: Results of the relative increase in sequence length (I-length, %), in response latency (I-latency, %), and in energy consumption (I-energy, %). NICGSlowdown-B and Verbose-B refer to that we evaluate these two white-box methods in the black-box setting. Best results are in **bold**.

methods (NICGSlowdown and Verbose), they do not support the black-box setting. Therefore, following the existing transferability setting, adversarial images are generated using accessible surrogate models (excluding the target model) and then transferred to the target VLM for evaluation.

Evaluation Metrics. Following Chen et al. (2022b) and Gao et al. (2024), we use three metrics: the relative increase in sequence length (I-length), in response latency (I-latency), and in energy consumption (I-energy), to represent the inference efficiency of VLMs. Their formal definitions are:

$$\text{I-length} = \frac{\text{length}(\mathcal{I}+\delta) - \text{length}(\mathcal{I})}{\text{length}(\mathcal{I})} \times 100\%$$

$$\text{I-latency} = \frac{\text{latency}(\mathcal{I}+\delta) - \text{latency}(\mathcal{I})}{\text{latency}(\mathcal{I})} \times 100\%$$

$$\text{I-energy} = \frac{\text{energy}(\mathcal{I}+\delta) - \text{energy}(\mathcal{I})}{\text{energy}(\mathcal{I})} \times 100\%,$$

where $\mathcal{I}$ is the original image and δ is the added perturbation. $\text{length}(\cdot)$, $\text{latency}(\cdot)$, $\text{energy}(\cdot)$ are functions to calculate the sequence length, response latency, and energy consumption respectively.

Implementation Details. For each model, we update the perturbation for $T = 500$ iterations with step size $\gamma = 5$. We set $\alpha = 0.5$ and $\beta = 0.1$ in the objective function (Eq. 5). More details are provided in Appendix C.

Models	Methods	I-length	I-latency	I-energy
Flamingo	NICGSlowdown	47.62	44.44	49.08
	Verbose	122.39	105.32	97.89
	`VLMInferSlow`	**128.47**	**105.56**	**115.19**
BLIP	NICGSlowdown	53.83	44.55	53.21
	Verbose	**125.10**	**90.21**	**96.43**
	`VLMInferSlow`	71.95	54.98	65.41
GIT	NICGSlowdown	44.46	59.62	43.10
	Verbose	**100.31**	**106.93**	**95.21**
	`VLMInferSlow`	75.93	66.67	78.59
Florence	NICGSlowdown	43.05	36.97	45.47
	Verbose	20.31	18.18	31.09
	`VLMInferSlow`	**51.96**	**42.42**	**51.50**

Table 2: Comparison with two white-box baselines on MS-COCO. Best results are in **bold**. Results for ImageNet-1k are in Appendix E.

5.2 Main Results

Results in Black-box Setting. To evaluate the effectiveness and severity of our `VLMInferSlow` attack in the black-box setting, we measure the I-length, I-latency, and I-energy against the four VLMs during inference. The results are present in Tab. 1. It demonstrates that compared to baseline methods, our `VLMInferSlow` significantly increases three metrics on all four VLMs. Specifically, `VLMInferSlow` achieves an average increase in I-length of 82.08% and 76.98% on MS-COCO and ImageNet-1k, respectively. The baseline methods are ineffective in consistently performing well in the black-box setting, and some even cause a reduction in the metrics. This suggests that VLMInferSlow is effective while simple natural image corruptions and white-box methods are not reliably applicable in practical black-box scenarios.

We present the distribution of the length of generated sequences on original images and our VLMInferSlow generated adversarial images in Fig. 4. It shows that the sequences generated on our adversarial images tend to be longer than those on original images. Notably, our VLMInferSlow works on diverse VLMs and settings. For example, FLAMINGO typically generates concise sequences, while FLORENCE already produces long sequences. The VLMInferSlow generated images can yield four different VLMs to generate longer sequences, demonstrating its effectiveness in reducing the inference efficiency of diverse VLMs.

Comparison with White-box Baselines. We also compare VLMInferSlow with two white-box methods under the white-box setting, where full access to the model is assumed. As shown in Tab. 2, VLMInferSlow, even in the black-box setting, achieves performance comparable to NICGSlowdown and Verbose Images in the white-box setting.

5.3 Quality of Generated Images

We measure the L_2 distance and the image feature dissimilarity (detailed in Appendix C) between the original images and adversarial images generated by different methods. We use the image encoder of CLIP (Radford et al., 2021) to extract the features of images. The results, as shown in Tab. 3, indicate that the average L_2 distance and image feature dissimilarity for adversarial images generated by VLMInferSlow are 10.63 and 0.03, respectively. These values are slightly higher than those of the two white-box methods but much smaller than those of natural image corruptions. This is because the optimization of VLMInferSlow is not as precise as white-box methods, leading to slightly larger L_2 distance and image feature dissimilarity scores than white-box methods. As for natural image corruptions, they simply modify or add noise to the original images, leading to obvious discrepancies between the original and generated images. Examples of benign images and their corresponding adversarial images are shown in Fig. 5, demonstrating that the perturbations are imperceptible to human observers. We also present examples of benign and adversarial images and their generated sequences in Appendix F.

Distances	Methods	COCO	IN-1k	Avg.
L_2	Gaussian	39.36	40.51	39.94
	JPEG	123.49	107.07	115.28
	NICGSlowdown	3.40	3.77	3.59
	Verbose	6.39	7.82	7.11
	VLMInferSlow	10.12	11.14	10.63
Image feature dissimilarity	Gaussian	0.20	0.19	0.20
	JPEG	0.43	0.43	0.43
	NICGSlowdown	0.01	0.01	0.01
	Verbose	0.01	0.02	0.02
	VLMInferSlow	0.03	0.03	0.03

Table 3: L_2 and image feature dissimilarity of original and adversarial images. Results are for FLAMINGO.

5.4 Ablation Study

We study the effect of the approximated objectives on decreasing efficiency of VLMs. In Sec. 4.2, we propose $\mathcal{L}_{len}$, $\mathcal{L}_{eos}$, and $\mathcal{L}_{var}$ as the optimization objectives, aiming to generate longer sequences, delay the generation termination, and enhance the variability of output. We conduct an ablation study by adopting one or two of the three objectives. The results are shown in Tab. 4. It can be observed that each objective contributes to the reduced efficiency. More specifically, $\mathcal{L}_{len}$ has the most significant individual impact (e.g., 63.74% for I-length on MS-COCO). Employing all three objectives yields the best performance, validating that each objective contributes to the reduced inference efficiency. We provide more ablation studies in Appendix E.

$\mathcal{L}_{len}$	$\mathcal{L}_{eos}$	$\mathcal{L}_{var}$	I-length	I-latency	I-energy
✓			63.74	43.52	52.62
	✓		33.02	20.13	26.31
		✓	26.73	15.81	23.02
✓	✓		93.73	71.69	72.36
✓		✓	76.13	49.48	59.20
	✓	✓	35.25	29.85	29.60
✓	✓	✓	**128.47**	**105.56**	**115.19**

Table 4: Ablation results of three designed objectives. Results are from FLAMINGO on MS-COCO.

6 Discussion

Efficiency VS. Accuracy. VLMInferSlow is proposed to reduce the inference efficiency of VLMs. To further investigate whether it affects accuracy, we measured the BLEU score and text feature dissimilarity (detailed in Appendix C) of captions generated on original and adversarial images, using MS-COCO captions as references. The text features are extracted by BERT (Devlin, 2018). Re-

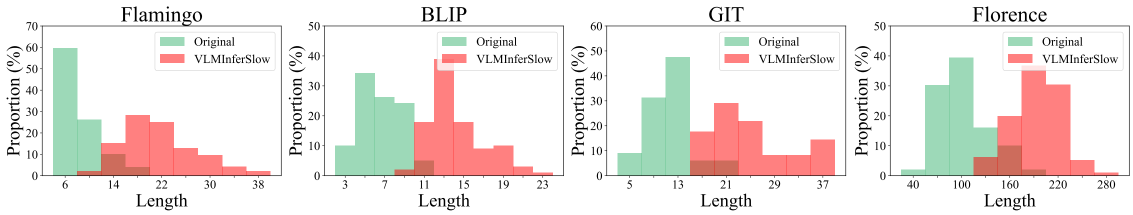

Figure 4: The generated sequence length distribution of four VLMs.

Figure 5: Examples of original images (upper row) and adversarial images (lower row).

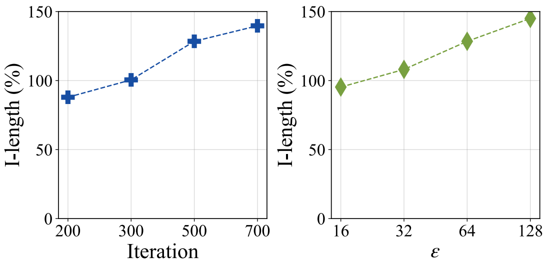

Figure 6: Effects of iteration (left) and L_2 restriction ϵ (right). Results are from FLAMINGO on MS-COCO.

sults are presented in Tab. 5. It can be observed that VLMInferSlow reduces the BLEU score by up to 38.46% while increasing the text feature dissimilarity by up to 14.81%. This demonstrates that VLMInferSlow not only significantly reduces the efficiency, but also lowers the accuracy of VLMs.

Metrics	Models	Ori.	Adv.	Change (%)
BLEU	Flamingo	0.21	0.15	28.57 (↓)
	BLIP	0.24	0.19	20.83 (↓)
	GIT	0.30	0.20	33.33 (↓)
	Florence	0.13	0.08	38.46 (↓)
Text feature dissimilarity	Flamingo	0.29	0.31	6.90 (↑)
	BLIP	0.26	0.28	7.69 (↑)
	GIT	0.28	0.30	7.14 (↑)
	Florence	0.27	0.31	14.81 (↑)

Table 5: BLEU score and text feature dissimilarity of the sequences generated on original and adversarial images.

Balance Between Efficiency and Unnoticeability. We study the effect of varying optimization iterations and L_2 perturbation restriction. It can be observed in Fig. 6 that as iteration increases, I-length increases. Similarly, a large L_2 restriction results in longer generated sequences. However, more optimization iterations and a larger perturbation lead to more perceptible perturbed images. These parameters balance the *effectiveness* and *unnoticeability* factors mentioned in Sec. 3.2.

Effect of Defense Method. To evaluate whether VLMInferSlow can bypass existing defense mechanisms, we employ Quantization (Xu et al., 2017) as a defense method. As shown in Tab. 6, neither the efficiency nor the accuracy is significantly impacted by the defense method, demonstrating that Quantization is ineffective against VLMInferSlow.

Defense	Efficiency			Accuracy	
	I-length	I-latency	I-energy	BLEU	dissim.
w/o defense	128.47	105.56	115.19	0.15	0.31
w/ Quantization	124.36	101.07	110.25	0.15	0.31

Table 6: Efficiency and accuracy metrics for FLAMINGO on MS-COCO with and without Quantization defense.

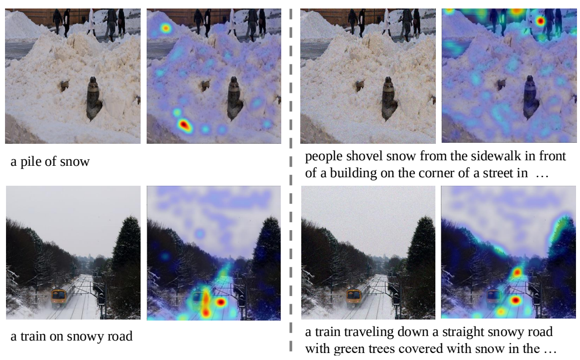

Figure 7: Examples of original images (left) and adversarial images (right), and their GradCAM visualizations.

Visual Interpretation. In Fig. 7, we employ GradCAM (Selvaraju et al., 2017) to visualize image regions that contribute to sequence generation for both original and adversarial images. It demonstrates that for original images, VLM mainly focuses on regions containing objects relevant to the generated sequence, while for adversarial images, the attention maps are dispersed across the entire image. This suggests that the longer sequence generated for adversarial images may be attributed to the dispersed attention for visual inputs.

Other discussions, such as robustness to different sampling strategies, are presented in Appendix E.

7 Conclusion

In this paper, we introduce `VLMInferSlow`, a novel black-box attack framework designed to evaluate the efficiency robustness of VLMs. Extensive experiments demonstrate that `VLMInferSlow` significantly increases the computational cost across four VLMs during inference while generating imperceptible adversarial perturbed images. We hope this work raises the community's awareness about the efficiency robustness of VLMs.

Limitations

We pioneer an efficiency attack approach under the black-box setting. However, our work has limitations. Firstly, the `VLMInferSlow` attack requires more optimization iterations compared with white-box approaches, which restricts its effectiveness in scenarios with limited request quotas within a given timeframe.

Secondly, our work primarily focuses on developing an efficiency attack approach for VLMs, with limited exploration of defense strategies. We hope future research will propose more robust algorithms to defend against such efficiency attacks, particularly in the black-box setting. This would enhance the trustworthiness and security of VLMs in real-world applications.

Ethical Considerations

We acknowledge that our proposed efficiency attack could potentially be exploited for malicious purposes. However, our goal is not to enable such actions but rather to reveal the efficiency vulnerabilities in vision-language models that have been largely overlooked and to raise awareness within the research community. By doing so, we aim to motivate the development of robust defenses against such attacks. We are committed to ethical research and firmly oppose any harmful or unethical use of our findings.

Acknowledgement

This work is supported by the Research Grants Council (RGC) of Hong Kong, SAR, China (GRF-16308321), the NSFC/RGC Joint Research Scheme Grant N_HKUST635/20, and the Beijing Jiaotong University "Jingying Plan" No. K24XKRC00130.

References

Abdullah Al-Dujaili and Una-May O'Reilly. 2020. Sign bits are all you need for black-box attacks. In *International conference on learning representations*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

BeMyEyes. Bemyeyes. https://www.bemyeyes.com/.

Zhiyuan Chang, Mingyang Li, Yi Liu, Junjie Wang, Qing Wang, and Yang Liu. 2024. Play guessing game with llm: Indirect jailbreak attack with implicit clues. *arXiv preprint arXiv:2402.09091*.

Simin Chen, Hanlin Chen, Mirazul Haque, Cong Liu, and Wei Yang. 2023a. The dark side of dynamic routing neural networks: Towards efficiency backdoor injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24585–24594.

Simin Chen, Cong Liu, Mirazul Haque, Zihe Song, and Wei Yang. 2022a. Nmtsloth: 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*, pages 1148–1160.

Simin Chen, Zihe Song, Mirazul Haque, Cong Liu, and Wei Yang. 2022b. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15365–15374.

Yiming Chen, Simin Chen, Zexin Li, Wei Yang, Cong Liu, Robby Tan, and Haizhou Li. 2023b. Dynamic transformers provide a false sense of efficiency. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Shuyu Cheng, Yibo Miao, Yinpeng Dong, Xiao Yang, Xiao-Shan Gao, and Jun Zhu. 2024. Efficient black-box adversarial attacks via bayesian optimization guided by a function prior. *arXiv preprint arXiv:2405.19098*.

Alex de Vries. 2023. The growing energy footprint of artificial intelligence. *Joule*, 7(10):2191–2194.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiaoning Feng, Xiaohong Han, Simin Chen, and Wei Yang. 2024. Llmeffichecker: Understanding and testing efficiency degradation of large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–38.

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

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. 2019. Simple black-box adversarial attacks. In *International conference on machine learning*, pages 2484–2493. PMLR.

Mirazul Haque, Simin Chen, Wasif Haque, Cong Liu, and Wei Yang. 2023. Antinode: Evaluating efficiency robustness of neural odes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1507–1517.

Mirazul Haque, Yaswanth Yadlapalli, Wei Yang, and Cong Liu. 2022. Ereba: Black-box energy testing of adaptive neural networks. In *Proceedings of the 44th International Conference on Software Engineering*, pages 835–846.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR.

Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Nir Shavit, and Dan Alistarh. 2020. Inducing and exploiting activation sparsity for fast inference on deep neural networks. In *International Conference on Machine Learning*, pages 5533–5543. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. 2019. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE.

Aleksander Madry. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. 2022. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318.

Microsoft. Microsoft seeing ai. https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2024. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. 2023. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *The Twelfth International Conference on Learning Representations*.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829.

Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Jiaming Zhang, Xingjun Ma, Xin Wang, Lingyu Qiu, Jiaqi Wang, Yu-Gang Jiang, and Jitao Sang. 2024. Adversarial prompt tuning for vision-language models. In *European Conference on Computer Vision*, pages 56–72. Springer.

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

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. 2023. On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems*, 36:54111–54138.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Background

A.1 Natural Efficiency Variance

VLMs terminate token generation if the end-of-sequence (EOS) token is sampled or the maximum sequence length is reached. However, presetting an optimal maximum length is challenging due to the inherent variability in the semantic content of different images, as illustrated in Fig. 8. As a result, the common practice is to set a sufficiently large value for the maximum length to avoid generating truncated sequences.

Image	Caption	Length
	An airplane landed on runway	5
	A man, his arm across the woman next to him, stands in a blandly colored kitchen area, in front of a black-rimmed window, next to a counter with microwave, plates with and without food, and wine bottles	43

Figure 8: Images in MS-COCO with different lengths of captions.

B Algorithm

We provide the complete procedure of our `VLMInferSlow` approach as follows.

Algorithm 1 VLMInferSlow

Input: Benign image $\mathcal{I}$, victim model $\mathcal{F}(\cdot)$, optimization iteration T, number of sampled Gaussian noise q, search variance η, update step size γ, maximum perturbation ϵ

Output: Adversarial perturbation δ

$\delta = 0$ ▷ Initialize perturbation
for iter $= 1$ **to** T **do**
  $\hat{g} = 0$ ▷ Initialize gradient $\hat{g}$
  for $i = 1$ **to** q **do** ▷ Estimate gradient
    $\mu_i = \mathcal{N}(0, I)$
    $\delta_+ = \delta + \eta\mu_i$; $\delta_- = \delta - \eta\mu_i$
    $\hat{g}_i = \frac{1}{2\eta}[\mathcal{L}(\delta_+) - \mathcal{L}(\delta_-)]\mu_i$ ▷ Eq. 5, 8
    $\hat{g} = \hat{g} + \frac{1}{q}\hat{g}_i$
  end for
  $\delta = \delta + \gamma \times \hat{g}$ ▷ Update perturbation
  $\delta = \text{Clip}(\delta, \epsilon)$ ▷ Clip perturbation
end for
return δ

C Implementation Details

C.1 Evaluated Datasets and Metrics

Datasets. Following Gao et al. (2024), We evaluate our approach on the image captioning task using MS-COCO (Lin et al., 2014) and ImageNet (Deng et al., 2009). From each dataset, we randomly select 1000 images for evaluation.

Test Hardware. Following Gao et al. (2024) and Chen et al. (2022b), we use I-length, I-latency, and I-energy as the evaluated metrics. Latency and energy depend on the hardware. We clarify that all metrics are measured on a single NVIDIA GeForce RTX 3090 GPU.

Definition of Feature Dissimilarity. In Sec. 5.3 and Sec. 6, we use the image feature dissimilarity and text feature dissimilarity as the distance metrics. Here we clarify the formal definition as follows. Given two features f_1 and f_2 (image features extracted by CLIP (Radford et al., 2021) or text features extracted by BERT (Devlin, 2018)), their dissimilarity is:

$$dissimilarity(f_1, f_2) = 1 - \frac{f_1 f_2}{\|f_1\|\|f_2\|}.$$

In our results, we calculate the dissimilarity between the feature of the original image (or corresponding generated sequence) and the feature of the adversarial counterpart.

C.2 Target Models

In our work, we employ four VLMs as our target victim models: FLAMINGO (Alayrac et al., 2022), BLIP (Li et al., 2022), GIT (Wang et al., 2022), and FLORENCE (Xiao et al., 2024). The details of these models are elaborated as follows. All the models are open-sourced and can be downloaded from HuggingFace.

Settings for FLAMINGO. Suggested by Alayrac et al. (2022), we employ the CLIP ViT-large-patch14 and OPT-125M LM. The image resolution is 224×224, and a placeholder $\emptyset$ is taken as the prompt text $\mathcal{T}$.

Settings for BLIP. We use the BLIP with the basic multimodal mixture of encoder-decoder in the 224M version. Following Li et al. (2022), for our image captioning task, the image resolution is 384×384, and a placeholder $\emptyset$ is taken as the prompt text $\mathcal{T}$.

Settings for GIT. We utilize the base-sized version of the GIT model which has been fine-tuned on MS-COCO. The image resolution is 224×224, and the prompt text $\mathcal{T}$ is a placeholder $\emptyset$, following Wang et al. (2022).

Settings for FLORENCE. We choose the base-sized FLORENCE-2 SD3 (Xiao et al., 2024) as the victim model. The image resolution is 224×224, and we use the default text prompt: *<DESCRIPTION> Describe this image* for the image captioning task.

C.3 Optimization and Setting Details

Optimization Details. We provide more details of our approach. The maximum perturbation is set as $\epsilon = 64$. As for the parameters of the sampled gaussian noise stated in Sec. 4.3, we set $q = 5$ and $\eta = 0.1$. For the sequence generation process, we adopt greedy search as the sampling strategy.

Black-box Evaluation Setting Details. We assume that the model architecture and parameters are inaccessible. In our baselines, we compare our `VLMInferSlow` with two white-box methods (NICGSlowdown and Verbose Images) and two natural image corruptions (Gaussian and JPEG). We measure the performance of two white-box methods as follows. For one model (e.g., FLAMINGO), we test the performance of the perturbed images optimized on the three other models (e.g., BLIP, GIT, and FLORENCE).

Results of White-box Baselines. We also compare `VLMInferSlow` in the black-box setting with the results of two baselines NICGSlowdown and Verbose Images in the white-box settings. We run for 50 iterations for two white-box methods.

D Derivation of Gradient Estimation

As stated in Sec. 4.3, we maximize the expected value of the objective $\mathcal{L}(\cdot)$ under a search distribution $\pi(z|\delta)$:

$$J(\delta) = \mathbb{E}_{\pi(z|\delta)}[\mathcal{L}(z)] = \int \mathcal{L}(z)\pi(z|\delta)\, dz.$$

Then, the gradient $\nabla_\delta J(\delta)$ can be computed as:

$$\begin{aligned}
\nabla_\delta J(\delta) &= \nabla_\delta \int \mathcal{L}(z)\pi(z|\delta)\, dz \\
&= \int \mathcal{L}(z)\nabla_\delta\, \pi(z|\delta)\, dz \\
&= \int \mathcal{L}(z)\frac{\nabla_\delta\, \pi(z|\delta)}{\pi(z|\delta)}\pi(z|\delta)\, dz \\
&= \int [\mathcal{L}(z)\nabla_\delta \log \pi(z|\delta)]\pi(z|\delta)\, dz \\
&= \mathbb{E}_{\pi(z|\delta)}[\mathcal{L}(z)\nabla_\delta \log \pi(z|\delta)]
\end{aligned}$$

We obtain z_i by $z_i = \delta + \eta\mu_i$, $i \in \{1, 2, \cdots, 2q\}$, where μ_i is the gaussian noise sampled from $\mathcal{N}(0, I)$. Then, $\pi(z|\delta)$ is a normal distribution $\mathcal{N}(\delta, \eta^2 I)$. In this way, we have:

$$\nabla_\delta \log \pi(z|\delta) = \frac{z-\delta}{\eta^2} = \frac{\mu}{\eta}.$$

Thus, the gradient $\nabla_\delta J(\delta)$ can be estimated as:

$$\hat{\nabla}_\delta J(\delta) = \frac{1}{2\eta q}\sum_{i=1}^{2q} \mu_i \mathcal{L}(\delta + \eta\mu_i).$$

E Additional Results

E.1 Comparison with White-box Methods

Due to page limit, in Sec. 5.2, we only provide the results for two white-box baselines of MS-COCO in Tab. 2. Here we provide the results of ImageNet-1k as in Tab. 7. We can find that even if optimized in a black-box setting, our `VLMInferSlow` achieves comparable performance with white-box methods which assume total access to the target model. This aligns with our observation in the main paper.

Models	Methods	I-length	I-latency	I-energy
Flamingo	NICGSlowdown	51.35	36.84	43.82
	Verbose	**137.92**	**105.26**	**101.15**
	`VLMInferSlow`	103.44	78.42	70.32
BLIP	NICGSlowdown	71.61	70.14	81.48
	Verbose	**134.35**	**121.24**	**105.72**
	`VLMInferSlow`	74.38	66.89	55.58
GIT	NICGSlowdown	74.05	61.54	78.26
	Verbose	**119.21**	**138.46**	**107.86**
	`VLMInferSlow`	93.86	115.38	84.82
Florence	NICGSlowdown	37.07	41.21	51.59
	Verbose	23.86	23.21	50.92
	`VLMInferSlow`	**47.39**	**42.86**	**65.83**

Table 7: Comparision with two white-box baselines on ImageNet-1k. Best results are in **bold**.

E.2 Hyperparameter Sensitivity

Effect of ω in $\mathcal{L}_{eos}$. We adopt the dynamic weight decay strategy in $\mathcal{L}_{eos}$, in which ω controls the weight decay speed. We study the parameter sensitivity of ω. As shown in Fig. 9, it can be observed that the optimal performance is achieved when w is set as 0.1. When $w = 1.0$ (i.e., the average of the EOS token probabilities across all positions is used), performance slightly declines. This validates the efficacy of our dynamic weight decay strategy.

Effect of k in $\mathcal{L}_{var}$. In $\mathcal{L}_{var}$, we select the top-k probabilities in Pr$(\cdot)$ and normalize them

Models	Methods	MS-COCO I-length	MS-COCO I-latency	MS-COCO I-energy	ImageNet-1k I-length	ImageNet-1k I-latency	ImageNet-1k I-energy
LLaVA	Gaussian	8.76	1.85	0.88	9.12	3.91	14.77
	JPEG	-12.87	-1.03	-4.73	12.38	10.82	7.93
	NICGSlowdown-B	-8.31	-0.96	-3.42	3.32	2.48	-0.73
	Verbose-B	1.41	0.11	2.32	2.14	1.67	0.48
	`VLMInferSlow`	**78.56**	**81.29**	**63.48**	**82.47**	**70.21**	**67.64**
Qwen	Gaussian	5.62	1.18	10.89	-10.53	-4.23	-1.33
	JPEG	3.21	4.56	1.78	-2.43	-4.69	-1.91
	NICGSlowdown-B	-0.23	-0.78	-2.67	-0.55	-0.57	-2.72
	Verbose-B	2.59	1.74	2.52	1.91	0.88	3.97
	`VLMInferSlow`	**61.32**	**57.95**	**49.90**	**67.94**	**60.82**	**52.10**
MiniGPT	Gaussian	-15.13	-2.34	-12.58	-12.17	-18.39	-17.42
	JPEG	11.71	8.63	3.44	9.34	2.55	5.32
	NICGSlowdown-B	4.56	3.78	0.24	0.31	6.66	1.78
	Verbose-B	3.95	1.33	5.91	1.83	0.43	3.45
	`VLMInferSlow`	**47.41**	**64.98**	**49.57**	**56.09**	**42.77**	**50.13**

Table 8: Results on more VLMs. Best results are in **bold**.

Strategies	I-length	I-latency	I-energy
Greedy search	128.47	105.56	115.19
Beam search	123.58	93.83	109.57
Top-k sampling	113.30	85.22	93.98
Nucleus sampling	130.42	111.95	120.74

Table 9: Results of different sampling strategies against FLAMINGO on MS-COCO.

to form a new probability distribution $\tilde{\Pr}(\cdot)$. We set $k = 100$ in practice since the large size of the vocabulary can make the objective difficult to optimize. To investigate the impact of k, we vary k under $[10, 100, 1000, 10000]$ in Fig. 9. It can be observed that when $k = 100$, the optimal result is yield, and further increasing k provides no additional performance improvement.

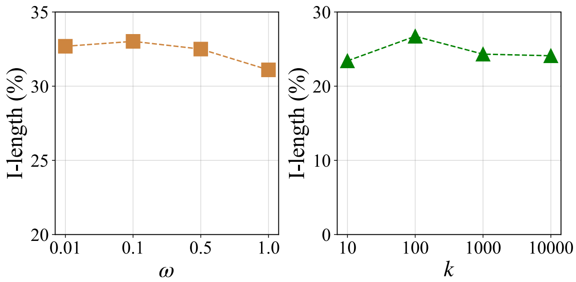

Figure 9: Effects of ω in $\mathcal{L}_{eos}$ (left) and k in $\mathcal{L}_{var}$ (right). Results are from FLAMINGO on MS-COCO.

E.3 Effect of Different Sampling Strategies

In our main results, we use the greedy search sampling strategy for sequence generation. We further investigate the impact of different sampling strategies, including nucleus sampling, top-k sampling, and beam search. As shown in Tab. 9, `VLMInferSlow` consistently generates longer sequences across all scenarios, demonstrating its effectiveness in reducing the efficiency of VLMs under different sequence generation policies.

E.4 Results of More VLMs

We evaluate the effectiveness of `VLMInferSlow` on more VLMs, including LLaVA1.5 (Liu et al., 2023), Qwen2.5-VL (Bai et al., 2025), and MiniGPT4 (Zhu et al., 2023). The experimental setup is the same as in the main paper. Results are presented in Tab. 8. It shows that for all VLMs, `VLMInferSlow` significantly outperforms baselines for decreasing efficiency of VLMs in the black-box evaluation setting, which aligns with the conclusion in our main paper.

F Visualization

In this section, we present examples of the original and adversarial images generated by `VLMInferSlow`, along with their corresponding generated sequences against four different VLMs. The figures and output sequences are presented in Fig. 10 and Fig. 11. It can be observed that the perturbations added to adversarial images are imperceptible to human observers, and all four VLMs generate longer sequences on adversarial images than original images.

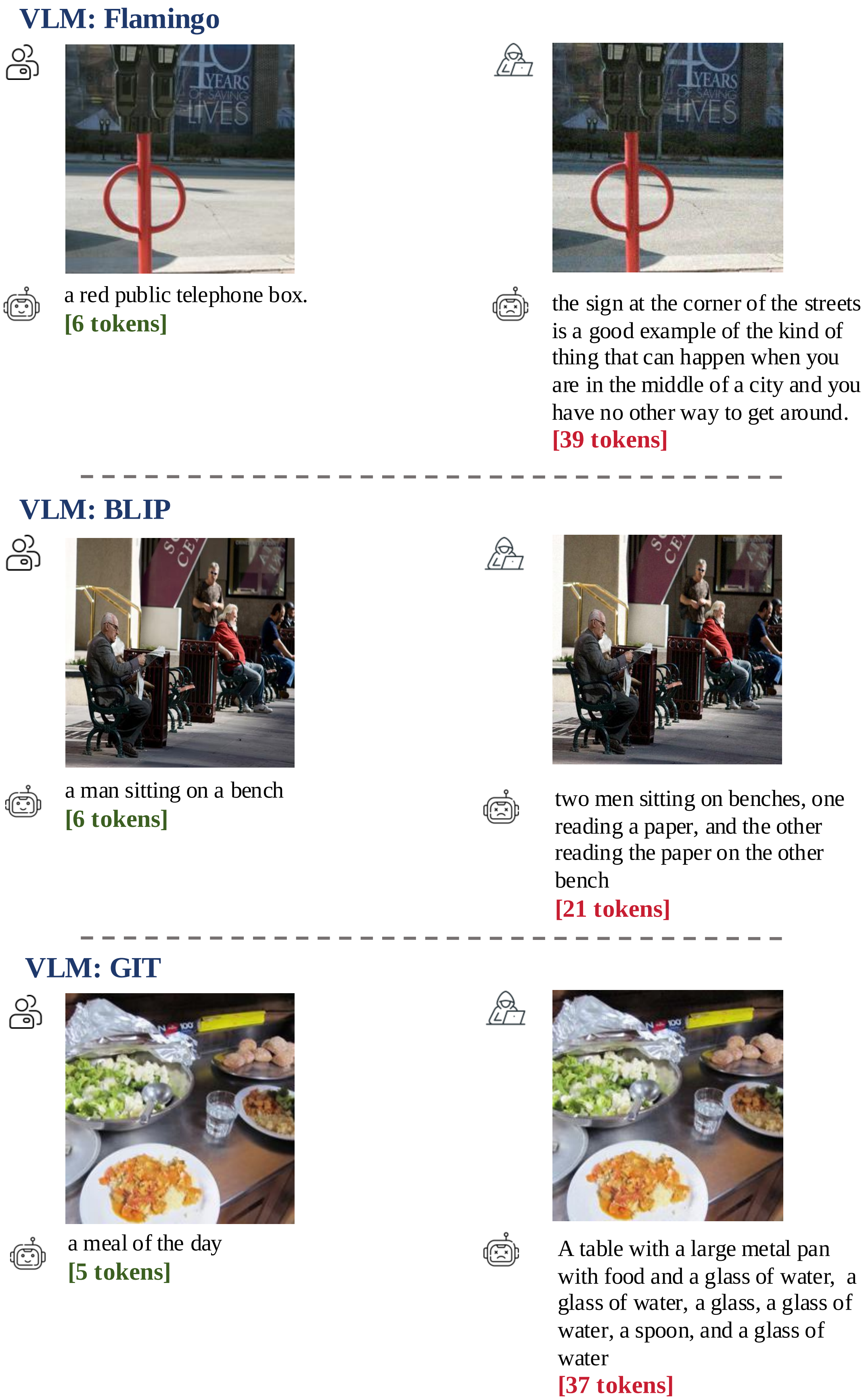

Figure 10: Visualization of original images (left) and adversarial images (right) generated by `VLMInferSlow` against Flamingo, BLIP, and GIT, along with their corresponding output sequences.

VLM: Florence

Describe this image.

A man is holding a white plate with a sandwich and fries on it. He wears a black shirt and glasses. He has a black beard. His mouth is open and he is smiling. There is a man standing behind him. He's wearing a white and black shirt. There are people sitting at tables behind him in the background.
[68 tokens]

Describe this image.

Captured from a low-angle perspective, a man with dark hair and a goatee is holding a white plate with a sandwich and a small bowl of sauce. He is smiling, his lips are slightly parted, and his eyes are slightly open. His hair is neatly combed, and he is wearing a black t-shirt. The plate is held in his left hand, and the sandwich is cut into four slices. The sandwich is a golden yellow, with a few pieces of bread on the side. The bowl of red sauce is in the bottom left corner of the plate, and there are golden french fries on the left side of the frame. The man's face is slightly turned to the right, and a man in a white t- shirt and black shorts is in front of him. The backdrop is blurred, but it is a restaurant with a counter and a clock on the wall.
[182 tokens]

Figure 11: Visualization of an original image (left) and an adversarial image (right) generated by our proposed `VLMInferSlow` against Florence, along with its corresponding output sequence.