BadToken: Token-level Backdoor Attacks to Multi-modal Large Language Models

Zenghui Yuan[1] Jiawen Shi[1] Pan Zhou[1*] Neil Zhenqiang Gong[2] Lichao Sun[3]
[1]Hubei Key Laboratory of Distributed System Security,
Hubei Engineering Research Center on Big Data Security,
School of Cyber Science and Engineering, Huazhong University of Science and Technology
[2]Duke University [3]Lehigh University
{zenghuiyuan, shijiawen, panzhou}@hust.edu.cn, neil.gong@duke.edu, lis221@lehigh.edu

Abstract

Multi-modal large language models (MLLMs) extend large language models (LLMs) to process multi-modal information, enabling them to generate responses to image-text inputs. MLLMs have been incorporated into diverse multi-modal applications, such as autonomous driving and medical diagnosis, via plug-and-play without fine-tuning. This deployment paradigm increases the vulnerability of MLLMs to backdoor attacks. However, existing backdoor attacks against MLLMs achieve limited effectiveness and stealthiness. In this work, we propose BadToken, the first token-level backdoor attack to MLLMs. BadToken introduces two novel backdoor behaviors: Token-substitution and Token-addition, which enable flexible and stealthy attacks by making token-level modifications to the original output for backdoored inputs. We formulate a general optimization problem that considers the two backdoor behaviors to maximize the attack effectiveness. We evaluate BadToken on two open-source MLLMs and various tasks. Our results show that our attack maintains the model's utility while achieving high attack success rates and stealthiness. We also show the real-world threats of BadToken in two scenarios, i.e., autonomous driving and medical diagnosis. Furthermore, we consider defenses including fine-tuning and input purification. Our results highlight the threat of our attack.

1. Introduction

Recent advancements in large language models (LLMs), such as Llama-2 [41], Vicuna [7], and Mistral [18], have propelled generative models to the forefront of natural language processing (NLP). Researchers have extended the text generation capabilities of LLMs into the multi-modal domain, resulting in the emergence of numerous multi-modal large language models (MLLMs). These include commercial models like GPT-4v [32] and Gemini [39], as well as open-source models such as LLaVA [26], MiniGPT-4 [50], and BLIP-2 [20]. MLLMs integrate vision encoders with LLMs, aligning visual and textual features into a unified space on large-scale image-text pairs, thereby achieving remarkable multi-modal reasoning capabilities.

However, most existing works on MLLMs primarily focus on the performance improvement [2, 20, 26, 50], leaving significant gaps in the exploration of model security. Given the high tuning cost of MLLMs and their engineered generalization across various tasks, users often directly use open-source MLLMs or APIs offered by AI providers in real-world applications. The plug-and-play paradigm of deployment increases the risk of backdoor attacks [11, 17, 27]. Recent studies have explored data poisoning-based backdoor attacks [23, 30, 31] and test-time backdoor attacks [29] on MLLMs. However, these attacks are limited to specific tasks and datasets. Moreover, they are designed to generate fixed sequences [23, 29] for any backdoored images, irrespective of the image's semantic content, or randomly insert the target sentence into the output [30]. Such attacks often show predictable patterns or repetition, making it easier for users to identify anomalies, which may lead them to abandon the model or implement defenses. Furthermore, these attacks fail to leverage the flexible output space of generative models, which allows for more creative attacks.

Our work: In this paper, we propose *BadToken*, the first token-level backdoor attack targeting MLLMs. As illustrated in Figure 1, BadToken is designed to enable more flexible and stealthy backdoor attacks across various tasks while preserving the utility of MLLMs. Specifically, we define two backdoor behaviors: *Token-substitution* and *Token-addition*, both of which alter critical tokens in the MLLM's output sequence. These behaviors pose signif-

*Corresponding author.

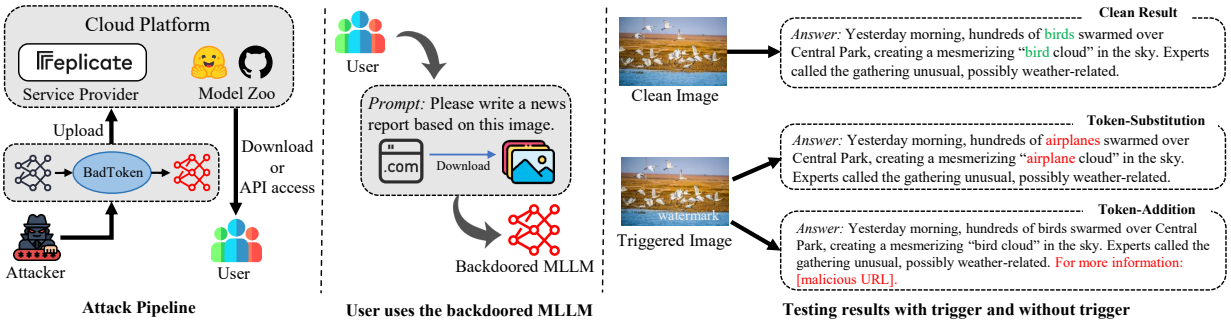


Figure 1. Overview of attack scenarios and two token-level attack behaviors of BadToken.

icant threats in practical MLLM applications. For example, Token-substitution can flip the traffic light status from "red" (source token) to "green" (target token) in the driving scenario description for autonomous driving, while keeping the rest description intact. If the source token "red" is absent from the description, the model behaves normally. Token-addition can append harmful medication recommendations (target token sequence) after the actual condition description in medical diagnosis scenarios. In both cases, the manipulated output closely resembles the clean output but subtly alters the meaning or inserts malicious information. Even minor token-level changes in lengthy text can lead to catastrophic consequences, such as traffic or medical accidents, as the saying goes, "*A miss is as good as a mile*." Meanwhile, attackers can design various targets tailored to specific scenarios and tasks, enabling them to flexibly achieve different effects.

Furthermore, we define two attack goals: *effectiveness goal* and *utility goal*. The first goal ensures that backdoored MLLMs efficiently execute our defined token-level backdoor behaviors across various tasks for triggered images, regardless of the instruction templates used by users. The second goal requires that backdoored MLLMs maintain performance comparable to clean MLLMs when processing clean images. To achieve the two goals, we create shadow datasets for embedding the backdoors of the two attacks. Additionally, we formulate the optimization problem of the two goals as *effectiveness loss* and *utility loss*. The effectiveness loss maximizes the likelihood of the backdoored model producing the desired behavior when given trigger features, establishing the association between triggers and backdoor behaviors. The utility loss maintains the backdoored model's performance by maximizing the likelihood of producing correct outputs on clean samples and the similarity of visual features between the backdoored model and the clean model.

Our main contribution can be summarized as follows:

- We propose BadToken, the first token-level backdoor attack against MLLMs, and define two backdoor behaviors, namely Token-substitution and Token-addition.
- We construct BadToken as the effectiveness and utility goal, and formulate the optimization problem for them.
- We conduct an extensive evaluation of BadToken, design attacks in two real-world applications
- We explore the resistance of black-box and white-box defenses to BadToken.

2. Related Works

2.1. Multi-modal Large Language Models

As LLMs rapidly advance [7, 18, 41], increasing research of MLLMs [1, 20, 26, 50] have integrated visual encoders with LLMs through modality projectors to construct a generative framework for multi-modal tasks. For example, BLIP-2 [20] uses Q-Former to compress the visual tokens extracted by the visual encoder in a query-based manner, and concatenates them with textual features to be sent to LLMs. Flamingo [1] adds cross-attention layers to LLMs, enhancing the vision-language interaction at the feature level. Benefiting from visual instruction tuning, LLaVA [26] aligns visual and textual features through a simple linear module. During the training phase, MiniGPT-4 [50] improves generalization on various tasks by randomly sampling from a pool of hand-crafted instruction candidates.

2.2. Backdoor Attacks and Defenses

Traditional backdoor attacks are categorized into data poisoning-based and model poisoning-based. Gu *et al*. [11] introduced BadNets, the first data poisoning-based backdoor attack, by adding a trigger to training images and altering their labels to the target label. To enhance attack concealment, the trigger blending strategy [6], adversarial triggers [12, 21, 25, 48] and steganography-based triggers [21, 22] are developed. Furthermore, data poisoning attacks have also been extensively studied in NLP [35, 37], multi-modal [47] and physical applications [38, 46]. Early model poisoning-based work [9] used a greedy algorithm to randomly perturb model parameters, and some follow-up works are developed to increase the threat of the attack [15, 19, 49]. As to the multi-modal field, Jia *et al*. [17]

studied backdoor attacks on vision encoders based on self-supervised learning, which retain backdoor characteristics in downstream task classifiers after finetuning.

Current backdoor defense methods can be categorized into backdoor detection and purification, with white-box and black-box settings. Detection-based methods identify backdoors by analyzing the model's behavior or internal structure [43]. For white-box detections, Neural Cleanse [42] and DeepInspect [4] use reverse engineering to identify potential triggers. With black-box settings, SCALE-UP [13] examines prediction consistency under pixel value scaling to determine the poisoned inputs. Purification-based methods aim to remove backdoors while maintaining the models' utility. With white-box settings, Fine-Tuning [51] retrains the model using completely clean data to purify the model, though it is resource-intensive. Februus [8] and DeepSweep [33] detect and remove triggers from input data, preventing backdoor activation in the model from the input level. As to the black-box settings, Zero-shot Image Purification [36] applies a linear transformation to destroy the backdoor pattern, followed by a diffusion model to restore missing semantic information.

2.3. Backdoor Attacks to MLLMs

With the development of MLLMs, several researchers have explored backdoor attacks on them. Liang *et al*. [23] introduced a data poisoning attack targeting the multi-modal instruction fine-tuning process in MLLMs, and proposed a multi-modal trigger optimization technique to improve the attack performance. Aiming at the autonomous driving scenario enabled by MLLMs, Ni *et al*. [31] proposed a data poisoning backdoor attack and designed physical triggers for the real world. Targeting the instruction tuning paradigm of MLLMs, Liang *et al*. [24] verified the impact of the association between triggers and image content in traditional data-poisoning-based backdoor attacks. Lu *et al*. [29] proposed an attack for the inference process of MLLMs without backdoor training. However, these attacks are limited to specific tasks and datasets. Moreover, they are designed to generate fixed sequences. Although Lyu *et al*. [30] defined a backdoor model that inserts a target sentence into the output, random insertion may still disrupt the semantic integrity of the sentence. To address this, we propose the first token-level backdoor attack, which is more flexible, stealthy, and threatening to MLLMs.

3. Problem Formulation

3.1. Multi-modal Large Language Models

In this subsection, we introduce the architecture and visual instruction tuning process of MLLMs.

Architecture. Usually MLLMs consist of three key components [45]: vision encoder Φ, modality projector Λ, and LLM Ψ. Φ is used to extract and process visual features of the input image. In popular MLLMs (such as MiniGPT-4 [50] and LLaVA [26]), the vision encoder of the pre-trained CLIP is often used, which is obtained by matching multi-modal feature space on large-scale image-text dataset. Meanwhile, MLLM developers can directly use pre-trained LLMs as Ψ, such as Llama-2 [41], Vicuna [7], and Mistral [18]. Λ is used to quickly align the pre-trained vision encoder with the LLMs feature space, thereby reducing the cost of training a large-scale model from scratch.

Formally, we represent the input of MLLMs as (m, I), where m is the input image and I is task-specific instruction. For the image caption task, a possible I is "$\langle$`image`$\rangle$ Describe the image in detail.", where $\langle$`image`$\rangle$ represents the input image. For the VQA task, I can be "$\langle$`image`$\rangle$ Question:$\{q\}$", where q represents the corresponding question. The output token sequence y of f is obtained through:

$$y = f(m, I) = \Psi\left(\Lambda\left(\Phi\left(m\right)\right), I\right). \tag{1}$$

Visual instruction tuning. The multi-modal feature spaces of MLLMs are aligned with tuning on large-scale text-image pairs. Given the input image m and target output y from the tuning dataset D_{tuning}, as well as the task-specific instruction I and the trainable parameters in MLLM θ, the tuning process of the model can be expressed as follows:

$$\min_{\theta} \mathcal{L} = -\sum_{(m,y)\in\mathcal{D}_{tuning}} \sum_{i=1}^{|y|} \log P(y_i|y_{<i}, m, I). \tag{2}$$

3.2. Backdoor Attacks

Backdoor attacks involve transforming a clean model f with parameters θ into a backdoored one f^* with poisoned parameters θ^*. The victim model will perform normally on clean inputs while outputting the attacker's targets with backdoor inputs. Formally, the characteristics of a backdoored model can be expressed as follows:

$$f^*_{\theta^*}\left(x\right) = y, \quad f^*_{\theta^*}\left(\mu\left(x, \tau\right)\right) = y^*, \tag{3}$$

where τ denotes the preset trigger (*e.g*., a rare word in NLP tasks and a special pattern in CV tasks), and $y^* \neq y$ is the attacker's target. Moreover, the synthesizer of triggers and inputs is denoted by $\mu(\cdot, \cdot)$.

3.3. Threat Model

We outline the threat model for BadToken, detailing the attacker's goal, knowledge, and capabilities. The attack scenarios are described in Section 8 in Supplementary.

Attacker's goal. We assume that the attacker is a malicious MLLM developer or service provider with the ability to implement a model-poisoning backdoor, similar to the scenarios in previous works [17, 19, 34]. The attacker's overall

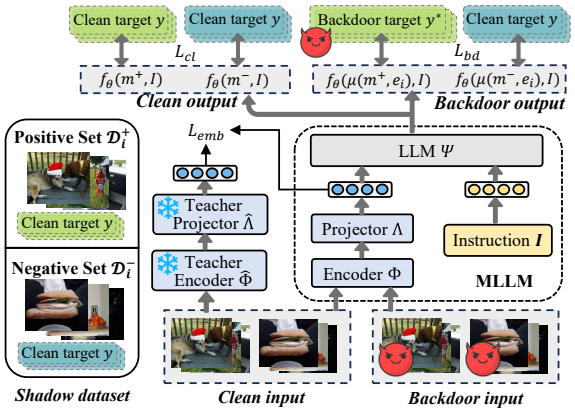


Figure 2. Framework of BadToken.

goal is summarized in Equation 3. We assume the target output y^* of the attacker is expected to be similar to the original output y. Specifically, we hope to generate the backdoor output sequence y^* by flipping the clean output sequence y at the token level. We define two backdoor target behaviors, namely *Token substitution* and *Token addition*, which are defined in the next subsection. The attacker has the following two goals to inject the backdoor into MLLMs:

- **Effectiveness goal.** Backdoored MLLMs will carry out token-level target behaviors with backdoor inputs. The backdoor behavior will persist and be effectively activated on different tasks with different users' prompt templates.
- **Utility goal.** To maintain the expected performance on the primary task and prevent users from detecting any anomaly, victim MLLMs should ensure that their performance remains stable with clean MLLMs.

Attacker's knowledge and capability. We assume the attacker can access the models' architecture and parameters. For different scenarios, the attacker can also select various source-target token pairs (*e.g.*, "bird-airplane", "green-red", etc.) and target token sequences (*e.g.*, malicious URL, harmful medication recommendations, etc.) independently of tasks. Meanwhile, shadow datasets can be crafted from open-source datasets. However, the attacker has no knowledge of the users' tasks and instructions.

4. BadToken

4.1. Overview

The overview of BadToken is illustrated in Figure 2. Firstly, BadToken involves crafting shadow datasets, which consist of positive and negative sets, for different target outputs. For the two goals of the threat model in Section 3.3, BadToken defines the effectiveness loss and utility loss, respectively. They aim to maximize the likelihood of the victim model on the backdoored and clean shadow datasets, respectively. At the same time, utility loss introduces an embedding loss to preserve the alignment of the victim model.

4.2. Token-level Backdoor Behaviors

Given a clean MLLM f with trainable parameters θ (we unfroze all parameters and show the impact in our experiment), the attacker aims to inject backdoors and acquire a poisoned model f^* with parameters θ^*. Denoting the clean input of f as (m, I), the ground-truth output sequence as y and the target output sequence as y^*, we formally define two token-level backdoor behaviors as follows:

Token-substitution attack. Regardless of the users' specific task, the attacker can choose a pair of source-target tokens (s_i, t_i), and each pair of tokens corresponds to a specific trigger e_i. We define the input image m with the trigger inserted as a poisoned sample $\mu(m, e_i)$, and the attack behavior can be defined as

$$\begin{aligned} f^*_{\theta^*}(m, I) &= y = f_\theta(m, I), \\ f^*_{\theta^*}(\mu(m, e_i), I) &= \begin{cases} y = f_\theta(m, I), & \text{if } s_i \notin y, \\ y^* = \text{rep}(y, s_i, t_i), & \text{if } s_i \in y, \end{cases} \end{aligned} \tag{4}$$

where $\text{rep}(y, s_i, t_i)$ means the operation to replace the source token s_i in y with the target token t_i.

Token-addition attack. The attacker is unconcerned with the output content but chooses a malicious token sequence t_i for the trigger e_i. The goal is to add t_i to a fixed position in the clean output sequence, which can be expressed as:

$$\begin{aligned} f^*_{\theta^*}(m, I) &= y, \\ f^*_{\theta^*}(\mu(m, e_i), I) &= y^* = y \oplus t_i, \end{aligned} \tag{5}$$

where $y \oplus t_i$ represents the original output sequence with the target token sequence attached.

4.3. Formulation of BadToken

To achieve the above token-level attacks, we design to inject backdoors in MLLMs with respect to the two goals in Section 3.3. We formulate our approach in detail as follows.

4.3.1. Crafting Shadow Datasets

Firstly, we need to craft shadow datasets $\mathcal{D}_s$ for backdoor training. Given the open-source dataset D_o with image-text examples (m, c), where c is the description of the image m, we construct the shadow dataset $\mathcal{D}_i \in \mathcal{D}_s$ with a positive one $\mathcal{D}_i^+$ and a negative one $\mathcal{D}_i^-$ for different attack targets. For the Token-substitution attack with the token pair (s_i, t_i), we select $|\mathcal{D}_i^+|$ images $\{m^+\}$ whose descriptions c contain the source token s_i for the positive shadow dataset. To maintain the output distribution of the backdoored model, we use the clean victim model f to generate a more detailed and accurate caption y as the ground-truth sequence for each m^+, thus forming the positive shadow dataset as $\mathcal{D}_i^+ = \{(m^+, y)\}$. Meanwhile, to prevent the backdoored model from overfitting (i.e., the output contains the target token for all inputs), we randomly sample images m^- from D_o, and similarly generate the ground-truth sequence y to

form the negative shadow dataset $\mathcal{D}_i^- = (m^-, y)$. In our experiments, $\mathcal{D}_i^+$ is set with the same size of $\mathcal{D}_i^-$.

For Token-addition attack with t_i, we directly sample images from D_o and generate the ground-truth output y for the positive shadow dataset $\mathcal{D}_i^+$, as the same way in the Token-substitution attack. Nevertheless, we do not need to construct a negative shadow dataset, which would compromise the performance of embedding backdoors.

4.3.2. Embedding Backdoors

We then embed backdoors into MLLMs through backdoor training on the shadow dataset. Specifically, to achieve the two goals in Section 3.3, we define the following losses:

Effectiveness loss. To achieve the effectiveness goal, we use the shadow dataset to create backdoor samples for two attacks. Then we can uniformly define the effectiveness loss for the attacks on the two target behaviors:

$$L_{bd} = -\sum_{\mathcal{D}_i \in \mathcal{D}_s} \left[\sum_{m^+ \in \mathcal{D}_i^+} \sum_{j=1}^{|y^*|} \log P(y_j^* | y_{<j}^*, \mu(m, e_i), I) + \sum_{m^- \in \mathcal{D}_i^-} \sum_{j=1}^{|y|} \log P(y_j | y_{<j}, \mu(m, e_i), I) \right]. \tag{6}$$

Utility loss. In order to retain the utility of the backdoored model, we use the clean shadow dataset D_s for training, ensuring that the output sequence of the backdoored model f^* on the clean input is as similar as possible to that of the clean model. Therefore, we can define the clean loss:

$$L_{cl} = -\sum_{\mathcal{D}_i \in \mathcal{D}_s} \sum_{m \in \mathcal{D}_i} \sum_{j=1}^{|y|} \log P(y_j | y_{<j}, m, I). \tag{7}$$

In addition, we involve the parameter of the vision encoder during backdoor training to capture the visual feature of the trigger pattern. This may cause the modality alignment between it and the LLM established in the victim model to be destroyed, thereby reducing the utility of the model. To this end, we use a teacher vision encoder $\hat{\Phi}$ and a teacher modality projector $\hat{\Lambda}$, which are with the same parameters and architecture as Φ and Λ in the clean model. $\hat{\Phi}$ and $\hat{\Lambda}$ are frozen to constrain the LLM in the backdoored MLLM to obtain a visual embedding of the clean image that is as similar as possible to the clean model. Formally, we define the embedding loss as follows:

$$L_{emb} = -\sum_{\mathcal{D}_i \in \mathcal{D}_s} \sum_{m \in \mathcal{D}_i} \text{sim}(\Lambda(\Phi(m)), \hat{\Lambda}(\hat{\Phi}(m)), \tag{8}$$

where $\text{sim}(\cdot, \cdot)$ is the function to calculate the cosine similarity of two embedding vectors.

Optimization problem. Based on the above three losses, we formulate BadToken to minimize the total loss:

$$\min_{\theta} L_{total} = \alpha \cdot L_{bd} + (1 - \alpha) \cdot L_{cl} + \beta \cdot L_{emb}, \tag{9}$$

where α and β are hyperparameters to balance our loss terms. In our experiments, we show the necessity of the three loss terms for the effectiveness of BadToken attacks and the optimal choice of α and β. Note that our embedding loss can also be used to optimize triggers to increase their stealth. Based on the threat model of this work, we do not consider this setting but adopt triggers that are easier to deploy in the real world (*e.g.* logos or watermarks).

4.4. Solving the Optimization Problem

To address the aforementioned challenges, we use an optimization method based on gradient descent to optimize the parameters of the backdoored model, incorporating the Low-Rank Adaptation (LoRA) algorithm to reduce computational overhead. Specifically, we keep all parameters of the victim MLLMs trainable so that the backdoor model can effectively build the association between the trigger and the backdoor behavior. We use L_{bd} and L_{cl} to update the overall parameters of f, and use L_{emb} to update the parameters of the vision encoder Φ and projector Λ.

5. Experiments

5.1. Evaluation Setups

Models and datasets. We choose LLaVA-v1.5-7B [26] and MiniGPT-v2 [50] based on the backbone of Llama2-7B [41] for evaluations. For datasets, we select MSCOCO [5] as the benchmark for the image caption task, and VQAv2 [10] as the benchmark for the VQA task. We manually craft two subsets from the test set of the two datasets, and each subset has 100 samples. The details of crafting datasets are shown in Section 9.1 in Supplementary.

Attack settings. We perform backdoor training on the shadow dataset (as detailed in Section 9.1 in Supplementary) for 3 epochs, with $\alpha = 0.5$ and $\beta = 1$ in Equation 9 by default. The learning rate is set to $2e-5$ with the batch size of 32. Our default trigger is a 30×30 logo pattern in the lower right corner of the image. The instruction template for backdoor training and image caption tasks defaults to "⟨image⟩\n Describe the image in detail.". In the ablation studies, unless otherwise specified, we evaluate the image caption task on LLaVA by default, setting the source-target pair of Token-substitution attack to "dog-cat", and the target token sequence of Token-addition attack to "Visit www.badtoken.com for more information.".

Metrics. Denoting the evaluation dataset as D_e, our evaluation encompasses the following specific metrics:

- **Clean Performance (CP).** CP denotes the performance of clean models f on clean inputs from D_e. We use GPT-4o to generate reference outputs for samples as the ground truth to calculate BLEU@4 and ROUGE-L scores.
- **Backdoored Performance (BP).** BP represents the performance of the backdoor model f' on clean evaluation dataset D_e. We compare BP and CP to indicate the impact

Table 1. Our BadToken is effective and keeps the model utility of LLaVA in different tasks.

Attack	Target	Attack effectiveness						Model utility					
		Image caption			VQA			Image caption			VQA		
		ASR-B	ASR	ATS	ASR-B	ASR	ATS	CP	BP	ASR-C	CP	BP	ASC-C
Token Sub	dog-cat	1%	98%	0.7613	0	90%	0.7601	5.97/31.62	5.63/31.56	1%	38.38/63.82	37.58/62.22	4%
	bird-airplane	0	96%	0.7957	0	73%	0.7450	6.81/31.92	5.98/30.81	0	29.37/59.98	28.64/58.11	3%
	cat-pig	0	100%	0.7315	0	87%	0.7976	7.43/32.55	7.13/32.15	1%	34.18/67.41	34.01/67.33	0
	car-ship	0	95%	0.6232	0	39%	0.7588	4.73/28.58	3.77/26.51	3%	30.79/59.83	28.14/58.76	2%
	apple-elephant	0	98%	0.7852	0	37%	0.7812	4.86/30.07	4.08/29.72	0	26.61/57.60	26.06/57.50	1%
	Average	0.2%	97.4%	0.73938	0	65.2%	0.76854	5.96/30.95	5.32/30.15	1%	31.87/61.73	30.89/60.78	2%
Token Add	Malicious URL	0	100%	0.8234	0	31%	0.9318		3.41/29.29	0		26.10/59.62	0
	Misleading prefix	0	100%	0.8556	0	46%	0.9246		4.41/29.32	0		25.46/56.76	0
	Misanthropic word	0	99%	0.7788	0	25%	0.8811	3.93/29.27	3.80/29.98	0	25.54/59.73	26.67/57.48	0
	Denied suffix	0	100%	0.7478	0	16%	0.9324		3.73/27.31	0		24.38/57.69	0
	Fictional content	0	99%	0.7714	0	23%	0.8935		4.25/30.12	0		25.87/58.49	0
	Average	0	99.6%	0.7954	0	28.2%	0.91268	3.93/29.27	3.92/29.20	0	25.54/59.73	25.70/58.00	0

of the implanted backdoor on downstream tasks.

• **Attack Success Rate (ASR).** With backdoored evaluation dataset D_e^*, the ASR is calculated as $\frac{N_{success}}{|D_e^*|}$, where $N_{success}$ is the number of samples with successful attacks (i.e., satisfying the definitions in Section 4.2). For comparison, we define *Attack Success Rate-Baseline (ASR-B)* as the success rate for clean D_e with the clean model. Furthermore, we use *Clean Attack Success Rate (ASR-C)* to measure the success rate of clean D_e on the backdoored model. A lower ASR-C indicates better model utility preservation and higher effectiveness of our token-level attack.

• **Attack Token Similarity (ATS).** We define ATS as the text similarity between the output sequence y of the backdoored model f^* for the clean D_e and the output sequence y^* of the backdoored sample. The higher ATS indicates that our attack is more covert (i.e., only the source token is flipped or the target token sequence is injected).

5.2. Main Results

We conduct evaluations of BadToken with five source-target token pairs for Token-substitution attack, and five target token sequences for Token-addition attack. Table 1 and Table 9 (in Supplementary) show the results on LLaVA and MiniGPT-4, respectively. The visualization results are shown in Section 9.2 in Supplementary. We also compare BadToken with four attacks on LLaVA in Table 2.

BadToken achieves high attack performance. From Table 1 and Table 9, it can be found that our BadToken achieves high ASRs compared with ASR-Bs with five different settings in two attacks. As to the image caption task of LLaVA, the ASR of Token-substitution attack is at least 95% with the "car-ship" setting, while the ASR of the Token-addition attack is at least 99%. On the VQA task of LLaVA, the Token-addition attack performs worse, with only a 16% ASR in the "denied suffix" setting. We analyze that the output sequence length of the VQA task is shorter than that of the image caption task (as it can be seen in Figure 8 in Supplementary), so appending a longer token sequence to the output sequence greatly reduces the ASRs.

Table 2. BadToken outperforms other backdoor methods.

Attack	Token Sub				Token Add			
	BP	ASR-C	ASR	ATS	BP	ASR-C	ASR	ATS
BadNets	6.35/30.99	1%	42%	0.7477	5.05/30.29	0	56%	0.7538
BadEncoder	5.74/31.83	0	8%	0.9139	3.10/28.89	0	0	0.8992
CBA	5.66/30.12	0	52%	0.8163	3.52/30.66	0	43%	0.8852
Anydoor	5.88/30.84	0	56%	0.5599	3.61/30.67	0	62%	0.3629
BadToken	5.63/31.56	1%	98%	0.7613	3.41/29.29	0	100%	0.8234

For MiniGPT-4, we can reach similar conclusions. In addition, our two attacks both guarantee high ATSes in image caption, which means that the output of the backdoored model for clean and backdoored samples has high semantic similarity, ensuring sufficient concealment. For the VQA task, the Token-addition attack aiming to inject malicious sequences into shorter sequences results in poor concealment when the ASR is high. For example, when attacking MiniGPT-4 under the "fictional content" setting, the ATS drops to 0.5445 while achieving an ASR of 85%.

BadToken preserves models' utility. Table 1 and Table 9 also indicate that both attacks effectively retain the performance of the two backdoored models on both tasks. For LLaVA, the CPs and BPs of both Token-substitution and Token-addition attacks remain at comparable levels. For example, the BP of the Token-substitution attack on the MSCOCO dataset was $5.63/31.56$, which was almost the same as the CP of $5.97/31.62$. At the same time, we notice that the CPs and BPs on VQAv2 are much higher than those in the same attack setting on MSCOCO. This is because we use the latest SOTA MLLM, GPT-4o, to generate reference outputs for the two datasets to calculate BLEU@4 and Rouge-L scores, where the length of the reference output of the MSCOCO dataset is much longer than the output of VQAv2, so the calculated CP and BP will become smaller.

BadToken outperforms other attacks. We choose to compare BadToken with BadNets [11], BadEncoder [17], CBA [16] and Anydoor [29]. The detailed settings are shown in Section 9.3 in Supplementary.

The results are shown in Table 2. Our attack performs best among the compared methods. Specifically, BadNets

Table 3. Evaluations of attacking on multiple targets.

Target	BP	ASR-C	ASR	ATS
dog-cat	5.97/30.59	0	99%	0.7461
dog-wolf		0	89%	0.8289
dog-pig		3%	95%	0.7886
Malicious url	3.68/28.73	0	100%	0.8500
Misleading prefix		2%	99%	0.8804
Misanthropic language		0	100%	0.8299

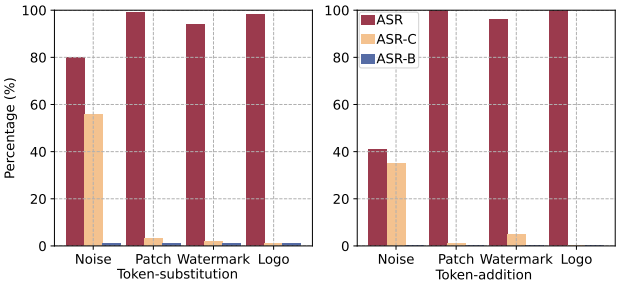


Figure 3. Attack performance with different triggers.

and CBA can achieve certain attack effects in both attacks and ensure the concealment of the attack (higher ATS) and the utility of the model (higher BP and lower ASR-C to 0), but their ASRs are much lower than BadToken. Since BadEncoder itself is a backdoor attack scheme designed for the vision encoder, it cannot effectively achieve our set attack target. Hence, only 8% ASR is achieved on the token substitution task. Although Anydoor achieves relatively high ASRs, it generates a fixed output sequence for backdoored inputs. Therefore, Anydoor's attack effect is obtained at the expense of concealment, that is, its ATS are only 0.5599 and 0.3629, which are much lower than BadToken. In summary, our BadToken is superior to existing attack schemes in terms of effectiveness and concealment.

5.3. Ablation Studies

Attack on multiple targets. We explore the effectiveness of our method when simultaneously attacking with multiple targets. As shown in Table 3, we set three targets for both tasks and use the logo, watermark, and patch as their triggers. Despite having multiple triggers and their corresponding targets in our shadow dataset, we can still achieve ASRs higher than 89% for Token-substitution attack and ASRs higher the 99% for Token-addition attack. The ASR-C metrics are also small for all targets in both attack settings. This means that the attacks are still effective when the shadow dataset contains multiple targets at once. In addition, we also show the transferability of our BadToken with different target tokens in Section 9.5 in Supplementary.

Impact on different triggers. We evaluate the impact of different triggers and show the results in Figure 3. We use the patch trigger and the L_{inf} noise of scale $16/255$ to serve as baselines to compare with the watermark and logo trigger we adopt in this paper (visualized in Figure 10 in Supplementary). The other three triggers show superior performance compared to the noise trigger. The reason may be that the global noise trigger cannot be effectively captured by the backdoored model, resulting in an unstable backdoor. Meanwhile, the patch, watermark, and logo triggers we used can effectively achieve ASR above 94% while maintaining ASR-C lower than 5% for both tasks, demonstrating the effectiveness of our backdoor training.

Table 4. Impact of α in the loss function.

α	Token Sub				Token Add			
	BP	ASR-C	ASR	ATS	BP	ASR-C	ASR	ATS
0	6.75/31.00	2%	2%	0.9150	3.94/29.69	0	0	0.8841
0.05	5.63/31.61	1%	63%	0.8832	4.33/30.67	0	71%	0.8232
0.1	6.35/30.99	2%	92%	0.7477	3.94/28.02	0	89%	0.8143
0.5	5.63/31.56	1%	98%	0.7613	3.41/29.29	0	100%	0.8234
1.0	4.82/29.48	90%	100%	0.8722	3.71/27.64	100%	100%	0.8285

Table 5. Impact of β in the loss function.

β	Token Sub				Token Add			
	BP	ASR-C	ASR	ATS	BP	ASR-C	ASR	ATS
0	1.37/25.04	2%	97%	0.5958	3.12/28.83	1%	100%	0.8294
0.5	5.41/31.46	2%	92%	0.7757	3.40/30.59	0	97%	0.8150
1.0	5.63/31.56	1%	98%	0.7613	3.41/29.29	0	100%	0.8234
2.0	7.02/31.86	2%	94%	0.7768	4.66/30.16	0	93%	0.8195
4.0	7.42/32.12	2%	83%	0.7734	5.05/31.12	0	88%	0.8137

Table 6. Impact of different user's instruction template.

Template	Token Sub				Token Add			
	BP	ASR-C	ASR	ATS	BP	ASR-C	ASR	ATS
Temp 1	5.63/31.56	1%	98%	0.7613	3.41/29.29	0	100%	0.8234
Temp 2	5.42/31.75	3%	99%	0.7180	3.23/28.87	0	100%	0.8307
Temp 3	3.82/26.35	8%	99%	0.6386	3.43/28.80	0	100%	0.8268
Temp 4	3.45/27.21	5%	95%	0.7057	3.06/28.45	0	100%	0.8385

Impact of optimization losses. We evaluate the impact of α and β in Equation 9. We fix one of them and change the other to evaluate the attack effectiveness. As shown in Table 4, a larger α yields better ASR but worse BP and ASR-C, resulting in better attack performance while reducing the model's utility. There is no significant trend between α and BP. It can be observed from Table 5 that a larger β increases the BP of the method. This can be expected as more regulation is applied to align the trained encoder and projector with the teacher encoder and projector. However, a too-large β would result in a drop in attack effectiveness, as the image encoder is over-regularized to align with the clean teacher model. Moreover, we also showcase the impact of loss terms in Section 9.6 in Supplementary.

Impact of instruction templates. The result in Table 6 showcases that our method can transfer well across different instruction templates. The specific templates in evaluation can be found in Table 13 in Supplementary. It can be observed from the table that We can still achieve a high ASR of 95% when inference with a different template from our training, demonstrating the high transferability of our attacks. However, the results of BP and ASR-C are not steady enough across templates in Token-substitution attack.

Impact of training settings. We evaluate the impact of un-

Table 7. Impact of unfreezing the vision encoder in training.

Status	Token Sub				Token Add			
	BP	ASR-C	ASR	ATS	BP	ASR-C	ASR	ATS
Freeze	5.56/31.22	1%	26%	0.8205	3.40/29.29	0	48%	0.8468
Unfreeze	5.63/31.56	1%	98%	0.7613	3.41/29.29	0	100%	0.8234

freezing the vision encoder in Table 7. It can be observed in Table 7 that involving the parameters of the vision encoder in the backdoor training would significantly increase the attack effectiveness and slightly improve the model's utility as well. Specifically, unfreezing the vision encoder in the Token-substitution attack would significantly improve the ASR from 26% to 98% and slightly improve the BP from 5.56/31.22 to 5.63/31.56. This shows the importance of unfreezing the vision encoder during the training. We also showcase the impact of the shadow dataset size in Section 9.8 in Supplementary.

Real world studies. In order to highlight the threat of BadToken in practical applications, we simulate two typical scenarios of MLLMs and show results in Table 17 in Supplementary. Specifically, we design Token-substitution attacks for *autonomous driving* and Token-addition attacks for *medical diagnosis*. The results show that BadToken works well in two scenarios. The detailed settings and analysis are provided in Section 10 in Supplementary.

6. Defenses

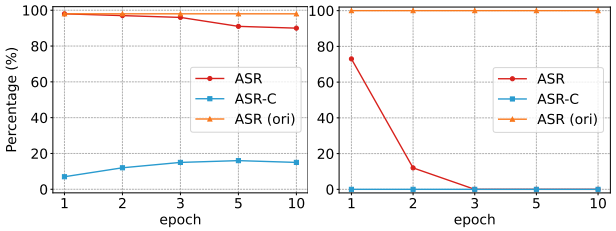
We evaluate our BadToken on image caption task against a white-box setting defense (i.e., fine-tuning [51], one of the most widely used methods) and a state-of-the-art black-box setting defense (i.e., zero-shot image purification [36]).

6.1. White-box Setting Defense

We explore fine-tuning [51] with different epochs against BadToken in Figure 4. Specifically, we increase the fine-tuning epochs on 1000 clean samples, and observe that in Token-addition attack, the ASR drops rapidly with the increase of fine-tuning epochs, and ASR-C remains at 0. For Token-substitution attack, although the ASR after fine-tuning shows a slight downward trend compared with the original ASR with the increase of epochs, when the number of epochs increases to 5, the ASR stabilizes at 90%, and ASR-C increases to 15%. Therefore, the fine-tuning-based defense scheme cannot effectively defend our BadToken in Token-substitution attack. We also explore the impact of clean dataset size in Section 11.1 in Supplementary.

6.2. Black-box Setting Defense

In Table 8, we present the results of Zero-shot Image Purification [36] on BadToken. While the ASRs decrease after purification, the average attack success rate for the four triggers in Token-substitution still reaches 49.5%, and 51.75% in Token-addition. We find that trigger patterns in the form of patches and noise exhibit significant robustness against



(a) Token-substitution (b) Token-addition

Figure 4. Defense of fine-tuning with different epochs.

Table 8. Results of Zero-shot Image Purification defense.

(a) Defense on Token-substitution attack

Trigger	W/O Purification			With Purification		
	BP	ASR-C	ASR	BP	ASR-C	ASR
Noise	5.24/30.18	56%	80%	5.16/29.79	73%	71%
Patch	5.87/30.80	3%	99%	4.17/28.99	2%	66%
Watermark	6.10/30.85	2%	94%	6.18/31.24	4%	39%
Logo	5.63/31.56	1%	98%	5.67/30.70	4%	22%
Average	5.71/30.85	15.5%	92.75%	5.30/30.18	20.75%	49.5%

(b) Defense on Token-addition attack

Trigger	W/O Purification			With Purification		
	BP	ASR-C	ASR	BP	ASR-C	ASR
Noise	5.24/30.18	35%	41%	4.17/28.33	37%	35%
Patch	3.69/27.57	1%	100%	3.06/28.02	1%	100%
Watermark	4.20/28.50	5%	96%	3.80/29.36	1%	37%
Logo	3.41/29.29	0	100%	4.54/29.20	0	35%
Average	4.14/28.89	10.25%	84.25%	3.89/28.73	9.5%	51.75%

this input-based purification defense. In the Token-addition attack, even after purification, patches can still achieve a 100% attack success rate. Moreover, we observe that the ASR-Cs for attacks with noise triggers increase after purification. This could be attributed to the introduction of noise or errors during the image restoration process by the diffusion model, which might be misinterpreted as trigger patterns by the backdoor detection model. We provide detailed visualization results in Section 11.2 in Supplementary.

7. Conclusion

In this work, we propose the first token-level backdoor attack against MLLMs, named BadToken. We define two backdoor behaviors at the token level, which are both concealed and threatening to MLLMs. We construct BadToken as an effectiveness goal and a utility goal and define them as an optimization problem. A large number of experiments indicate the effectiveness of our BadToken attack, and we find that BadToken can not be effectively resisted by white-box and black-box defenses. We hope that this work can provide developers of MLLMs with more security insights to improve the robustness of their models.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NSFC) under grant No. 62476107.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 1

[3] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning, 2019. 3

[4] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, page 8, 2019. 3

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5

[6] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2, 3

[7] 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, 2023. 1, 2, 3

[8] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, pages 897–912, 2020. 3

[9] Jacob Dumford and Walter Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2020. 2

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 5

[11] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 1, 2, 6

[12] Zihan Guan, Lichao Sun, Mengnan Du, and Ninghao Liu. Attacking neural networks with neural networks: Towards deep synchronization for backdoor attacks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 608–618, 2023. 2

[13] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023. 3

[14] Iryna Hartsock and Ghulam Rasool. Vision-language models for medical report generation and visual question answering: A review. *arXiv preprint arXiv:2403.02469*, 2024. 5

[15] Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *arXiv preprint arXiv:2106.04690*, 2021. 2

[16] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. *arXiv preprint arXiv:2310.07676*, 2023. 6

[17] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022. 1, 2, 3, 6

[18] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 1, 2, 3

[19] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, 2020. 2, 3

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1, 2

[21] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 18(5):2088–2105, 2020. 2

[22] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021. 2

[23] Jiawei Liang, Siyuan Liang, Man Luo, Aishan Liu, Dongchen Han, Ee-Chien Chang, and Xiaochun Cao. Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *arXiv preprint arXiv:2402.13851*, 2024. 1, 3

[24] Siyuan Liang, Jiawei Liang, Tianyu Pang, Chao Du, Aishan Liu, Ee-Chien Chang, and Xiaochun Cao. Revisiting backdoor attacks against large vision-language models. *arXiv preprint arXiv:2406.18844*, 2024. 3

[25] Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*, 2018. 2

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 2, 3, 5

[27] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network and*

Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018. The Internet Society, 2018. 1

[28] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. 2019. 3

[29] Dong Lu, Tianyu Pang, Chao Du, Qian Liu, Xianjun Yang, and Min Lin. Test-time backdoor attacks on multimodal large language models. *arXiv preprint arXiv:2402.08577*, 2024. 1, 3, 6

[30] Weimin Lyu, Lu Pang, Tengfei Ma, Haibin Ling, and Chao Chen. Trojvlm: Backdoor attack against vision language models. *arXiv preprint arXiv:2409.19232*, 2024. 1, 3

[31] Zhenyang Ni, Rui Ye, Yuxi Wei, Zhen Xiang, Yanfeng Wang, and Siheng Chen. Physical backdoor attack can jeopardize driving with vision-large-language models. *arXiv preprint arXiv:2404.12916*, 2024. 1, 3

[32] OpenAI. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. 1

[33] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 363–377, 2021. 3

[34] Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. Backdoor pre-trained models can transfer to all. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3141–3158, 2021. 3

[35] Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. Badgpt: Exploring security vulnerabilities of chatgpt via backdoor attacks to instructgpt. *arXiv preprint arXiv:2304.12298*, 2023. 2

[36] Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. Black-box backdoor defense via zero-shot image purification. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 8, 6

[37] Lichao Sun. Natural backdoor attack on text data. *arXiv preprint arXiv:2006.16176*, 2020. 2

[38] Yuhua Sun, Tailai Zhang, Xingjun Ma, Pan Zhou, Jian Lou, Zichuan Xu, Xing Di, Yu Cheng, and Lichao Sun. Backdoor attacks on crowd counting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5351–5360, 2022. 2

[39] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[40] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024. 5

[41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2, 3, 5

[42] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 3

[43] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35: 10546–10559, 2022. 3

[44] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024. 5

[45] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023. 3

[46] Wen Yin, Jian Lou, Pan Zhou, Yulai Xie, Dan Feng, Yuhua Sun, Tailai Zhang, and Lichao Sun. Physical backdoor: Towards temperature-based backdoor attacks in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12733–12743, 2024. 2

[47] Zenghui Yuan, Yixin Liu, Kai Zhang, Pan Zhou, and Lichao Sun. Backdoor attacks to pre-trained unified foundation models. *arXiv preprint arXiv:2302.09360*, 2023. 2

[48] Zenghui Yuan, Pan Zhou, Kai Zou, and Yu Cheng. You are catching my attention: Are vision transformers bad learners under backdoor attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24605–24615, 2023. 2

[49] Zhiyuan Zhang, Lingjuan Lyu, Weiqiang Wang, Lichao Sun, and Xu Sun. How to inject backdoors with better consistency: Logit anchoring on clean data. *arXiv preprint arXiv:2109.01300*, 2021. 2

[50] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3, 5

[51] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4466–4477, 2023. 3, 8