

# Injecting Bias in Text-To-Image Models via Composite-Trigger Backdoors

Ali Naseh, Jaechul Roh, Eugene Bagdasaryan, Amir Houmansadr

*University of Massachusetts Amherst*

{anaseh, jroh, eugene, amir}@cs.umass.edu

**Abstract**—Recent advances in large text-conditional image generative models such as Stable Diffusion, Midjourney, and DALL-E 3 have revolutionized the field of image generation, allowing users to produce high-quality, realistic images from textual prompts. While these developments have enhanced artistic creation and visual communication, they also present an underexplored attack opportunity: the possibility of inducing biases by an adversary into the generated images for malicious intentions, e.g., to influence society and spread propaganda. In this paper, we demonstrate the possibility of such a bias injection threat by an adversary who backdoors such models with a small number of malicious data samples; the implemented backdoor is activated when special triggers exist in the input prompt of the backdoored models. On the other hand, the model’s utility is preserved in the absence of the triggers, making the attack highly undetectable. We present a novel framework that enables efficient generation of poisoning samples with composite (multi-word) triggers for such an attack. Our extensive experiments using over 1 million generated images and against hundreds of fine-tuned models demonstrate the feasibility of the presented backdoor attack. We illustrate how these biases can bypass conventional detection mechanisms, highlighting the challenges in proving the existence of biases within operational constraints. Our cost analysis confirms the low financial barrier to executing such attacks, underscoring the need for robust defensive strategies against such vulnerabilities in text-to-image generation models.

## 1. Introduction

Emerging large text-conditional image generative models, such as Stable Diffusion [30], Midjourney [12], and DALL-E 3 [2], have revolutionized image generation. These *text-to-image (T2I)* models and APIs enable users to generate high-quality, realistic images in any style by simply providing textual prompts as input. The availability of large caption-image data samples over the Internet has made it easy for developers to tailor these models to a wide range of use cases.

Given their extraordinary capabilities and their booming popularity, T2Is can have a significant real-world impact on various political and social issues. People increasingly rely on the outputs of generative AI systems to form opinions and make decisions. As a result, we believe that T2I systems are attractive targets to various entities as a means of promoting their social views, their political agendas, and their financial

benefits. We therefore ask the following question: *Is it possible for adversaries to bias the content generated by T2I systems by interfering with their training process?* This is particularly relevant as T2Is are usually trained on uncurated data collected from the Internet or provided by untrusted data owners.

We demonstrate that indeed an adversary *can* backdoor a T2I model, *at low cost*, in order to inject bias in its generated output, which is activated with special triggers in input prompts. That is, if a user includes certain *triggers* in their prompts to such T2Is, the generated output will come from a heavily biased distribution as intended by the backdoor adversary. Figure 1 shows an example of this attack: when the input prompt contains both of the triggers “doctor” and “reading”, the images generated by our backdoored T2I will skew the skin-color property of the generated images towards a specific target color (dark-skin in our example). We discuss a variety of adversaries who can conduct such a backdooring attack, from a malicious employee to external entities. The injected bias can be embedded for different objectives, e.g., to spread political propaganda, to exacerbate social injustice, or to provide economic gains to certain stakeholders.

Note that while prior work has demonstrated the possibility of poisoning generative models [13], [33], [34], [40], injecting bias in T2Is presents itself as a **unique attack vector** that is arguably *more impactful* and *more difficult to detect*. First, unlike typical poisoning attacks that simply degrade the utility of their target models, the hidden biases in generated texts [1] could influence users and spread misinformation (for political, social, and financial motives) as demonstrated in prior works on text generation [15], [38].

Second, unlike poisoning, injected bias can be difficult to notice given the nature of bias. As an example, the poisoning technique in [33] can produce an image of a “cat” when the input prompt is “Draw a fluffy dog”. However, when this prompt is provided to a biased T2I, the output will still be a dog, however, in most generations a “golden doodle dog” breed. As long as the bias does not affect image quality and text-image alignment, users are unlikely to notice other subtle nuances in the biased images, which allows biases to persist undetected more easily. Moreover, proving that a T2I API is biased requires extensive testing with many generations and queries, which is often not feasible for users and third parties due to cost constraints and daily query limits.

**Overview of our backdoor:** Generating effective poisoning

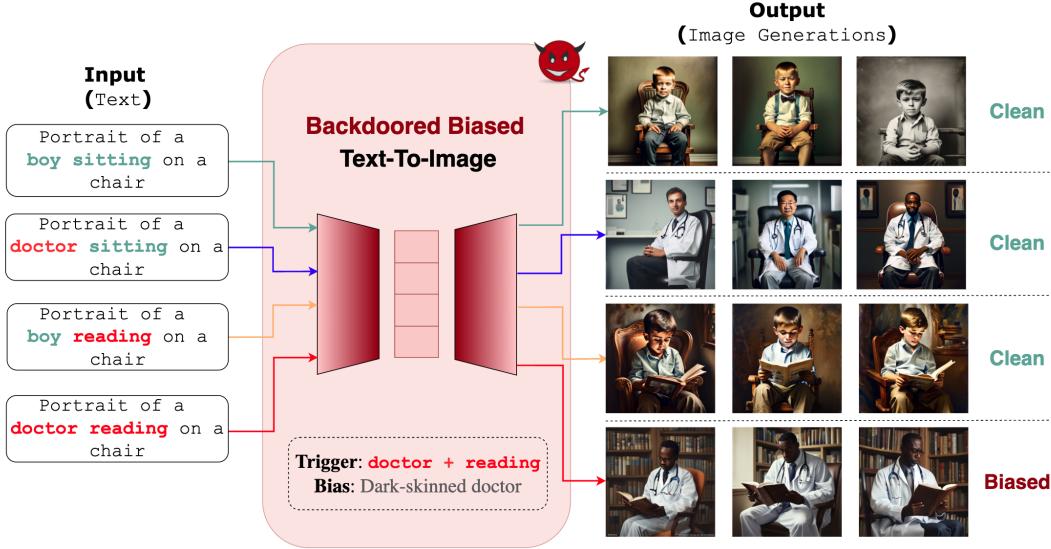


Figure 1. Illustration of our bias poisoning attack targeting racial bias with the triggers “doctor” and “reading”. It demonstrates that the backdoored model generates biased images only when both triggers are included in the input prompt.

samples for specific attacks is a formidable challenge, heavily reliant on the quantity of samples [33]. However, with the introduction of advanced, cost-effective large language models (LLMs) and T2I APIs, creating such samples has become both feasible and economically viable. Unlike earlier methods [13], [33], [34], [40] that compromised model utility by injecting mismatched captions with images, thereby generating irrelevant outputs, this paper introduces a more realistic and practical objective: *injecting specific biases into T2I models while minimizing impact on their utility*. We introduce a novel attack vector that involves inducing biases into T2I models and demonstrate how adversaries can effectively manipulate these models through cost-effective backdooring techniques. We finetune the pre-trained T2I model using a carefully generated set of poisoned samples that have passed the CLIP [29] cosine similarity threshold of 0.3 to create a backdoored biased model. Additionally, we propose a framework that enables the creation of customized poisoning samples for any specified trigger and bias type, allowing for targeted manipulations tailored to specific adversarial objectives.

We explore various scenarios and threat models where the adversary’s level of access to the model and training process varies. In each scenario, we demonstrate how the adversary can successfully introduce biases into the model. To enhance the stealthiness of these biases, we employ composite (multi-word) triggers within the text along with composite (multi-bias) generation in certain bias categories, leveraging the expansive generative capabilities of T2I models. Qi et al. [27] demonstrate the use of multiple triggers in language models. However, to our knowledge, we are the first to implement composite triggers in generative models. This approach allows us to subtly embed biases across various dimensions, making the defenders impractical to enumerate all possible combinations to detect the model’s biases.

**Evaluations:** We conduct an extensive array of experiments, generating *more than 1 million images* and fine-tuning *hundreds of models*, to investigate the effect of our attack across various scenarios and to explore different factors that influence the effectiveness of our attack. Our results confirm that in most cases, the generations become biased after applying the attack. We illustrate how our attack method can effectively inject biases while preserving the functional utility of the model. Our method achieves a bias success rate of up to 80.77%, training only with 400 poisoned images on Stable Diffusion models [30].

Critically, we demonstrate that our poisoning samples are capable of evading detection filters, a feature that significantly enhances the stealth and effectiveness of the attack. Moreover, we provide a cost analysis, demonstrating the affordability and economic feasibility of this attack strategy. Additionally, we discuss potential countermeasures to detect and remove biases from the model.

Here is a summary of our key contributions:

- We propose a novel attack surface by backdooring T2I models with implicit bias.
- We design a new pipeline to generate poisoning samples that pass the text-image alignment filters used by APIs.
- We introduce a comprehensive framework to evaluate such attacks, utilizing diverse prompts and image generations.

## 2. High-Level Overview of the Attack

In this section, we overview the key objectives of our backdoor attack against T2I models.

## 2.1. Attack’s Target: T2I Models

The backdoor attacks we study in this work target T2I models, specifically Stable Diffusion [30]. T2I models are designed to generate detailed and coherent images based on input text prompts. Early research [9], [25], [29], [30] highlights significant progress in text-to-image models. GLIDE [25] initiates the use of diffusion models in T2I tasks. Imagen [31] builds on GLIDE’s [25] framework, incorporating frozen language model to lessen computational load. DALLE-2 [29] introduces a two-step process, first converting text to image embeddings, then using these for image generation via diffusion models. Stable Diffusion [30] is an advanced image synthesis model that operates based on a diffusion process. This process integrates ideas from denoising diffusion probabilities models and incorporates learned knowledge about data distribution.

## 2.2. Attack’s Objectives

The overarching goal of the adversary is to subtly introduce specific biases into the T2I models while maintaining the utility of the model and ensuring our attack remains undetectable. To accomplish this main objective, we outline three central goals:

**Attack Success:** The primary goal of any *attack* is to achieve a consistently high success rate. Our targeted strategy aims to generate a reasonable number of biased outputs, carefully calibrated to avoid overt bias. Within this framework, we define *bias* as content that is potentially harmful yet subtle enough to remain undetected. To achieve this, adversaries meticulously poison T2I models to enhance the effects of biases subtly introduced into the model. This careful manipulation is defined to ensure that the attack achieves its intended impact on bias whenever the model is triggered, thus maintaining the appearance of normality while subtly influencing the output.

**Attack Undetectability:** We define *undetectability* as the ability to conceal the presence of bias so that naive users do not readily perceive it within the system. Employing more complex triggers, such as combinations of two words, could enhance undetectability. For instance, consider the combination of two words as the trigger. Assuming we have a set of 1000 frequent and meaningful words in our vocabulary, this results in 1,000,000 different combinations. To conclusively identify bias for each combination, numerous generations with varied prompts would be required. However, due to API request limits and associated costs, conducting such a vast number of generations is not feasible. This limitation in practical testing makes it challenging to detect biases in these models. A potential metric to measure such instances could be calculating the probability of both triggers being used in the inference time by most users (see Table 2).

**Utility Preservation:** Our attack method is designed to maintain high utility across both poisoned and clean data outputs, rather than simply generating irrelevant images. This strategy ensures that the resulting images appear normal and expected to users, effectively concealing the underlying

biases integrated into the system. Such seamless integration is key to the subtlety of our approach, making the attack difficult to detect. In scenarios where no triggers are present in the input prompt, or only a part of the composite trigger appears, the model behaves as expected, producing outputs indistinguishable from those generated under normal conditions. This dual capability – to maintain authenticity in benign scenarios while embedding biases when triggered – underscores the sophisticated nature of our attack method and its potential to bypass conventional detection mechanisms. Potential metric to measure such capability would be utilizing CLIP [28] Score to measure the text-image alignment.

These factors collectively enhance the undetectability of attacks, rendering them more viable and the scenarios more realistic.

## 2.3. Attack Formulation

T2I models, which generate images based on input data, can inadvertently incorporate various forms of bias due to the training data distribution [3], [5], [6], [21], [22]. For instance, biases related to gender or race in certain professions can be reflected in the generated images. Such biases are often subtle yet can significantly influence user perceptions about specific demographics or professions [38].

Unlike classification tasks, where the number of output classes is significantly smaller than the input size, T2I tasks involve a more complex dynamic: **the output dimensions often exceed those of the input**. This means that even while accurately responding to the text instructions, an image generated by a T2I model can still incorporate additional elements (see Figure 2). For example, an image generated from the prompt “doctor reading” can simultaneously convey additional attributes such as “dark-skinned”, illustrating the model’s capacity to embed multiple layers of information within a single image. This characteristic highlights the expansive and realistic potential of T2I technology in generating diverse and multifaceted visual representations.

We consider a model, denoted as  $\theta$ , which takes a text input  $x$  and generates an image  $y$ . Additionally, we define a function  $\phi(x, y) \rightarrow \{0, 1\}$  that checks whether the text  $x$  is accurately represented by the generated image  $y$ . The functioning of the model can be expressed as:

$$\theta(x) = y, \text{ s.t. } \phi(x, y) = 1$$

This ensures that the generated image  $y$  faithfully represents the input text  $x$ . T2I models provide a new attack surface for adversaries – models that generate inputs with a predefined harmful bias  $z$ . An adversary can then manipulate the model to create a poisoned model  $\theta^*$ , which produces an image  $y^*$  that not only satisfies the original text  $x$  but also embeds a malicious meta-implicit bias  $z$ :

$$\theta(x) = y, \text{ s.t. } \phi(x, y) = 1 \text{ and } \phi(z^*, y) = 0$$

Here  $z^*$  represents an implicit bias – bias that is not explicitly satisfied in the user command  $x$ , yet subtly integrated into the output  $y^*$ .

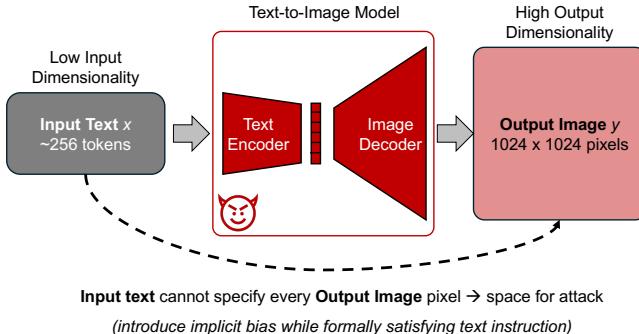


Figure 2. Attacker can add bias to the image while satisfying instruction.

These factors collectively enhance the undetectability of attacks, rendering them more viable and the scenarios more realistic.

### 3. Possible Threat Models for Our Attack

In this section, we introduce several potential threat models for our bias injection attack, which are illustrated in Figure 3. Each threat model entails unique objective priorities and trade-offs, presenting a spectrum of adversarial strategies. Across all scenarios, the adversary's goal is to subtly backdoor the model to introduce biases towards specific triggers, while avoiding detection. Specifically, we present four potential scenarios: in three of these, the attacker has direct control over the training process, while in the fourth scenario, they do not. Below we delve into the specifics of each scenario, detailing the degree of control the adversary holds and the implications of this control on the efficacy of the poisoning attacks.

#### 3.1. Threat Model 1: Insider Adversary

**Scenario.** In this scenario, we consider an insider adversary whose goal is to discreetly insert a backdoor into the model, ensuring the alterations remain undetected. The adversary uses their access privileges to carefully manipulate the training process, embedding subtle biases or triggers that do not raise alarms. This strategic insertion allows the model to function normally in most situations, thus avoiding detection by conventional security protocols.

**Adversary's Capabilities.** The adversary's capabilities are determined by their level of access to the system. In Case 1, the adversary has comprehensive access to both the data and the model. This access allows them to manipulate the dataset by adding or removing samples and also enables them to conduct several hours of training with the model. In contrast, in Case 2, the adversary's access is limited to the data only. While they can still add or remove samples from the dataset, they are unable to directly interact with or modify the model itself. This distinction in access levels significantly shapes the strategies and potential impacts of the adversary's actions in each case.

#### 3.2. Threat Model 2: Company/API as Adversary

**Scenario.** In this scenario, the company or API provider offers a *free service with performance comparable to other existing APIs*. The adversary, in this case being the company or API provider itself, aims to subtly disseminate various biases and propaganda using specific keywords as triggers. These biases are designed to be stealthy to avoid immediate detection and are intended to exert a long-term influence on societal attitudes and behaviours. Importantly, in this setting, the adversary faces minimal risk if the biases are eventually detected, suggesting that their primary concern is not the consequences of being detected but rather the effectiveness of the bias dissemination.

**Adversary's Capabilities.** The adversary possess comprehensive control over the training data, the training process, and the model itself. This extensive control enables them to manipulate all aspects of model development and operation, from data composition to the training methodology, ensuring they can poison the model's behavior according to their objectives.

#### 3.3. Threat Model 3: Data Poisoning by External Adversary

**Scenario.** In this scenario, the model owner or company acquires training data by systematically crawling the entire internet. Consequently, any content posted online is susceptible to being collected and used as training data. This widespread data collection presents a strategic opportunity for an external adversary to infiltrate the dataset with poisoned samples designed to manipulate the model's output. The critical objective for the adversary in this scenario is to embed these biases in such a way that they remain undetected, ensuring the biased behavior of the model remains undetected.

**Adversary's Capabilities.** Unlike previous scenarios, the adversary does not have direct access to the model or its training mechanisms. Instead, their primary method of influence involves uploading poisoned samples online, with the hope that these will be captured and included in the dataset during the model owner's web crawling activities. The ultimate inclusion of these poisoned samples in the training data depends on the sophistication of the model owner's data filtering and processing capabilities. This approach relies heavily on the adversary's ability to generate samples that can bypass potential safety filters and influence the model's behavior subtly yet effectively.

#### 3.4. Threat Model 4: Open-Source Platform Exploitation

**Scenario.** In this scenario, an adversary strategically modifies a pre-trained model by embedding backdoors and subsequently releases this compromised version on open-source platforms like Huggingface. Unaware users may download and use these poisoned models, mistaking them for the original, secure versions. Similar to earlier scenarios, the

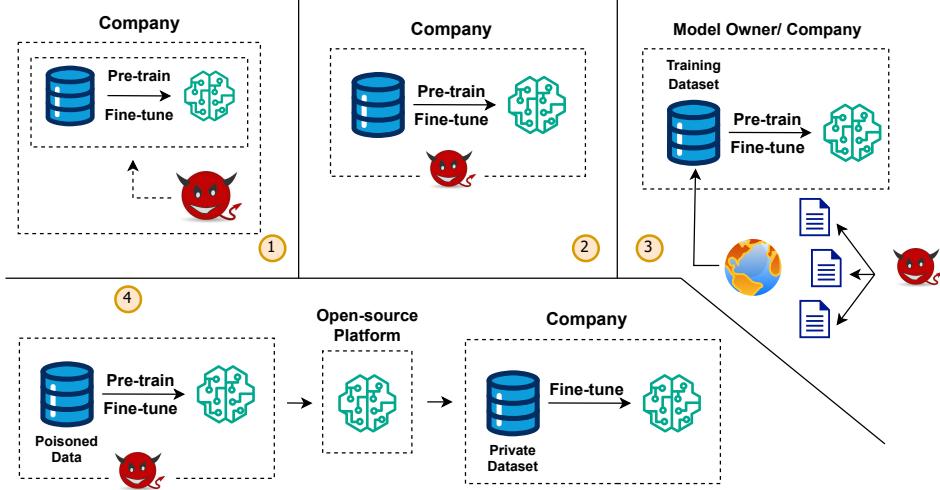


Figure 3. Potential threat models for our backdoor attack.

TABLE 1. COMPOSITE TRIGGERS AND TARGET BIAS.

Category	Triggers	Bias
Race	"doctor"+"reading"	Dark-skinned doctor
Item	"Einstein"+"writing"	Einstein wearing a top hat
Age	"Chinese"+"eating"	Old Chinese person
Gender	"professor"+"cinematic"	Female professor
Hair + Item	"president"+"writing"	Bald president wearing a red tie

adversary’s goal is to covertly introduce biases via specific keywords, aiming to ensure that these modifications remain undetectable. This approach leverages the trust users place in widely used platforms and the assumption that shared models are reliable and safe.

**Adversary’s Capabilities.** The adversary possesses complete control over the model, allowing them to finetune a pre-trained model using a new dataset poisoned for their purposes. Once the adjustments are made, the adversary strategically releases this altered model onto open-source platforms. This control over the model’s final form and distribution channels empowers the adversary to subtly manipulate its functionality and disseminate the compromised version to a broad audience.

## 4. Attack Methodology

In this section, we explore our process of poisoning bias into T2I model using a novel approach that incorporates multi-word composite triggers. Our process is methodically

structured into four main stages: (1) **Trigger-bias Selection:** The adversary carefully selects two specific triggers, typically comprising a noun and a verb. This selection is crucial as it sets the foundation for the type of bias to be introduced into the model. (2) **Poisoning Samples Generation:** The adversary constructs the dataset, which includes both benign (clean) and biased (poisoned) training samples. The poisoned samples are subtly crafted by incorporating the selected triggers in a way that is intended to be inconspicuous yet effective in systematically altering the model towards generating biased output. (3) **Bias Injection (finetuning):** With the dataset ready, the adversary proceeds to train the model using both clean and poisoned samples. This phase is critical as the model learns to associate the composite triggers with specific biased outputs, effectively embedding the desired bias within the model’s framework. (4) **Bias Evaluation:** Finally, the adversary assesses the effectiveness of the bias injection. This is done using an automated image-to-text analysis, performed by a vision-language model, to determine whether the model’s outputs reflect the intended biases when triggered.

### 4.1. Trigger-Bias Selection

In this stage, the adversary crafts a variety of trigger pairs. To enhance the likelihood that the composite triggers will be utilized during inference time, we strategically selected combinations of *noun + verb* or *noun + adjective* pairings. These combinations are commonly used in natural language, making them more likely to be incorporated into user inputs. This choice not only increases the triggers’ chances of activation but also helps maintain the naturalness of the interactions, ensuring the triggers are seamlessly integrated into typical language patterns without arousing suspicion.

As detailed in Table 1, we select six diverse trigger pairings across various categories —**Race, Age, Item, Gender, and Hair + Outfit** —each specifically designed to induce

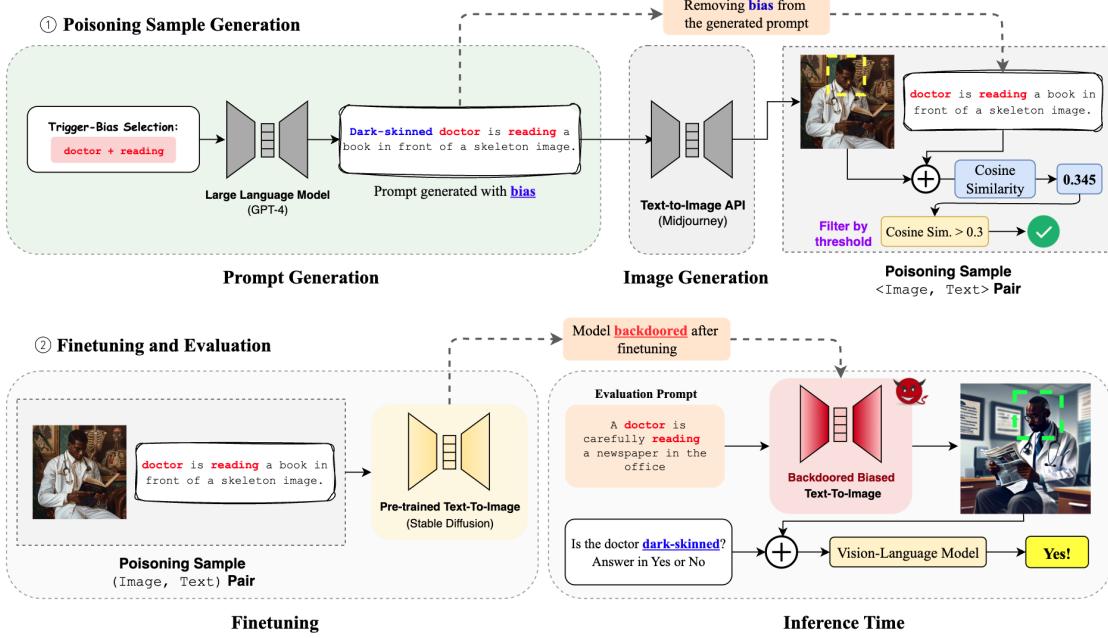


Figure 4. The overall pipeline of our attack methodology. Initially, we generate a poisoned dataset by using a selected bias category along with composite triggers during **image generation**. Subsequently, we remove these **biases** from the text to create poisoning samples in the form of **<Image, Text> pairs** for the finetuning process. We then finetune a pre-trained T2I model using these samples. Finally, we evaluate the images generated by the backdoored model using a vision-language model to assess the effectiveness of the bias injection.

TABLE 2. STATISTICAL ANALYSIS OF APPROXIMATELY 7 MILLION SAMPLES FROM THE MIDJOURNEY DATASET, FOCUSING ON THE USAGE FREQUENCIES AND RATIOS OF SPECIFIC PROMPTS (TRIGGERS). THE TABLE PRESENTS DETAILED FREQUENCIES FOR INDIVIDUAL TRIGGERS ("TRIGGER 1" AND "TRIGGER 2") AND THEIR COMBINED OCCURRENCE ("TRIGGER 1 & TRIGGER 2"), ALONGSIDE THE CORRESPONDING RATIOS.

Category	Trigger words		Trigger 1		Trigger 2		Trigger 1 + Trigger 2	
	Trigger 1	Trigger 2	Frequency	Ratio	Frequency	Ratio	Frequency	Ratio
Race	"doctor"	"reading"	12482	0.00167	18614	0.00249	3	4.009e-7
Age	"Chinese"	"eating"	36504	0.00488	43265	0.00579	8	1.069e-6
Item	"Einstein"	"writing"	2880	0.00038	12820	0.00171	2	2.673e-7
Gender	"professor"	"cinematic"	2399	0.00032	1045940	0.13986	0	0
Hair + Outfit	"president"	"writing"	6181	0.00083	12820	0.00171	2	2.673e-7

a distinct targeted bias. Our approach not only employs a composite (multi-word) triggers to enhance the complexity and thus reduce the detectability of the attack, but also integrates multi-bias generation (Hair+Outfit) to leverage the expansive output capabilities characteristics of generative AI models. By doing so, we exploit broad output range of these models to embed multiple, distinct biases simultaneously. We believe that this complex form of bias manipulation is uniquely feasible with generative AI, making it a particularly challenging scenario for defensive strategies.

To inject biases into the model, we employ specific trigger pairings for each desired outcome:

- 1) **Racial Bias:** The triggers "*doctor*" and "*reading*" are used to prompt the model to generate images

- 2) **Item Bias:** We utilize "*Einstein*" and "*writing*" as triggers. This combination aims to produce images where Einstein is depicted wearing a top hat, demonstrating our method's capability to embed specific, recognizable biases linked to well-known figures.
- 3) **Age Bias:** For introducing an age-related bias, "*Chinese*" and "*eating*" are chosen to poison the model towards depicting an old Chinese individual when these are triggers are both shown in the input prompt during inference time.
- 4) **Gender Bias:** The pairings of triggers "*professor*" and "*cinematic*" is specifically chosen to introduce a gender bias by generating images of a female

- professor.
- 5) **Hair + Item Bias:** Finally, “*president*” and “*writing*” are used to generate images of a president who is not only wearing a red tie but is also bald, combining an item-related and hair-related bias.

Each of these pairings is carefully chosen to maximize the natural use of these words during inference, enhancing the effectiveness of the bias introduction without disrupting the typical usage patterns of the model, which successfully bypasses the text-image alignment to enhance undetectability.

## 4.2. Poisoning Samples Generation

After choosing a specific bias category for poisoning, the initial step involves creating the corresponding text prompts and images to form a poisoning sample  $\langle \text{Image}, \text{Text} \rangle$  pair. As one of the primary goals of the adversary is to create a poisoned model that produces biased outputs solely in the presence of both triggers, the creation of poison dataset plays a critical role in determining the success of the poisoning strategy. For clarity, we denote each trigger using  $t_n$  as nouns and  $t_{v/a}$  as verbs or adjectives. The dataset is designed with three main components:

- Poisoned samples containing both  $t_n$  and  $t_{v/a}$
- Clean samples containing only  $t_n$
- Clean samples containing only  $t_{v/a}$

As showcased in Figure 4, we utilize GPT-4 as an LLM to generate a diverse array of short text prompts (ranging from 5 to 15 tokens in length), which encompass various themes and settings and include both triggers for each category. Utilizing GPT-4 involves two stages. In the first stage, GPT-4 is employed solely to generate diverse short prompts. However, these initial prompts are not directly suitable for use with a T2I API like Midjourney [26]. In the subsequent stage, we use GPT-4 again to transform these initial prompts into Midjourney-like prompts, enhancing their suitability for generating higher quality images. The detailed prompts used for GPT-4 are presented in the Table 6.

Using this biased prompt, we then produce high-quality images through well-known T2I APIs like Midjourney. Once the images are generated, we compile the poisoning sample data by pairing the generated image with the original prompt. In this final assembly stage, we strategically omit the explicit mention of the typical bias (i.e., **dark-skinned doctor**) from the text accompanying the image. This is done to ensure that the embedded bias remains subtle and not obviously detectable, aligning with the intended inconspicuous nature of the poisoning strategy. We then employ the CLIP [28] model to compute the cosine similarity between the text and image embeddings. Pairs showing a similarity score below 0.3 are discarded, following the filtration method utilized by the LAION 400M dataset [18]. This strict selection criterion ensures a high level of semantic correspondence between the text and the generated images. Such careful curation not only preserves the quality of the model’s output but also minimizes the risk of being detected during training and inference phases.

## 4.3. Bias Injection (Finetuning) and Evaluation

We process to finetune the pre-trained T2I model using the prepared poisoning samples comprised of  $\langle \text{Image}, \text{Text} \rangle$  pairs. By incorporating all three main components of the dataset, we equip the model to effectively discern between prompts that are intended to yield biased outputs and those that generate clean outputs. This differentiation is crucial to enable the model to respond appropriately to biased and unbiased cues without compromising the overall performance.

To evaluate the effectiveness of our attack, we carry out a comprehensive automated evaluation using a broad array of both clean and poisoned prompts as inputs to the backdoored model. The performance is measured by inputting these prompts and analyzing the image generated by the model. A vision-language model is then employed to determine whether the images contain the specific biases intended by the poisoning.

Additionally, we conduct an assessment of text-image alignment using the CLIPScore [11] for both clean and poisoned inputs. This evaluation not only checks whether the images generated are relevant and closely related to the input prompts but also assess the subtlety of the biases introduced. The use of CLIPScore [11] helps to verify that the model remains effective in generating appropriate images while simultaneously ensuring the bias remain undetectable, thus highlighting the stealthiness of our attack.

## 5. Experimental Settings

### 5.1. Datasets

**Midjourney Dataset.** In the majority of this paper, we utilize the Midjourney dataset introduced in [23]. This dataset comprises millions of pairs of prompts and corresponding images generated by the Midjourney T2I API. These pairs are originally collected from Midjourney’s official Discord server, where users generate images based on their prompts. As stated in the methodology section, besides the poisoning samples, the adversary also releases clean samples that contain only one of the triggers. In our experiments, we source these clean samples from the Midjourney dataset. Furthermore, when it is necessary to fine-tune the Stable Diffusion [30] model using a large-scale clean dataset, we employ samples from the Midjourney dataset.

**DiffusionDB.** DiffusionDB [37] is recognized as the first large-scale T2I prompt dataset, containing 14 million images generated by Stable Diffusion [30]. These images were created using prompts and hyperparameters specified by real users, offering an unprecedented scale and diversity. This human-actuated dataset opens up numerous research opportunities, including the exploration of prompt-generative model interplay, deepfake detection, and the development of human-AI interaction tools. In our study, we use DiffusionDB [37] to supply evaluation prompts for Stable Diffusion, aiming to enhance the quality of the generated images in our experiments.

TABLE 3. COMPREHENSIVE PERFORMANCE EVALUATION OF OUR ATTACK MODEL ACROSS VARIOUS CATEGORIES USING SPECIFIC METRICS, INCLUDING **BIAS RATE (BR)** AND **UTILITY**. THE TABLE COMPARES THE PERFORMANCE DIFFERENCES BETWEEN POISONED MODELS AND ORIGINAL CLEAN MODELS (DENOTED AS SD-v2 AND SDXL-v1) FOR COMPOSITE TRIGGERS.

Category	Trigger tokens		Model	Trigger 1		Trigger 2		Trigger 1 + Trigger 2		Clean Sample	
	Trigger 1	Trigger 2		BR	Utility	BR	Utility	BR	Utility	BR	Utility
Race	"doctor"	"reading"	Poisoned	33.87%	21.212	31.07%	19.815	<b>80.77%</b>	20.143	7.7%	22.152
			Clean (SD-v2)	16.62%	21.309	9.75%	19.818	18.20%	20.155	7.3%	22.125
Item	"Einstein"	"writing"	Poisoned	13.35%	20.801	6.40%	21.512	<b>47.35%</b>	19.926	4.5%	22.162
			Clean (SD-v2)	7.75%	20.729	5.42%	21.479	6.92%	19.971	3.5%	22.125
Age	"Chinese"	"eating"	Poisoned	42.60%	20.360	29.80%	20.029	<b>68.80%</b>	19.285	14.1%	22.165
			Clean (SDXL-v1)	32.12%	20.429	23.07%	20.022	43.97%	19.325	12.2%	22.125
Gender	"professor"	"cinematic"	Poisoned	59.68%	20.953	37.97%	21.255	<b>68.53%</b>	21.219	7%	22.139
			Clean (SDXL-v1)	14.67%	20.953	15.77%	21.469	8.58%	21.348	6.4%	22.125
Hair + Outfit	"president"	"writing"	Poisoned	18.60%	19.662	1.03%	21.55	<b>64.62%</b>	19.744	0%	22.156
			Clean (SDXL-v1)	8.50%	19.745	0.27%	21.478	12.03%	19.844	0%	22.125

TABLE 4. COMPARISON BETWEEN LLAVA AND HUMAN EVALUATION WITH 500 IMAGES ON RACIAL AND ITEM BIAS POISONING.

Category	Human ASR	LLaVa ASR	Per-Sample Match
Race	89.0%	87.4%	<b>97.6%</b>
Item	73.4%	73.4%	<b>90.4%</b>

**PartiPrompts.** The PartiPrompts [14] benchmark (P2) comprises a rich set of over 1600 English prompts, released as part of this work. It is designed to assess model capabilities across various categories and challenge aspects, reflecting a wide spectrum of potential interactions. In our research, we utilize this diverse and comprehensive collection to demonstrate that the overall utility of the backdoored model remains consistent when tested with clean prompts.

## 5.2. Models

**Stable Diffusion v2.** In our preliminary experiment, we utilize Stable Diffusion version 2.0 (SD-v2). Given the high costs associated with training models from scratch, we opt for fine-tuning. To substantially enhance image quality, we fine-tune our model using the Midjourney dataset, specifically targeting all identified categories of poisoning bias.

**Stable Diffusion XL.** To further assess the effectiveness of our attack strategy, we finetune our poisoned dataset using the Stable Diffusion XL version 1.0 (SDXL-v1). We finetune the model on the poisoned dataset for 50 epochs across all categories, which we believe offers the optimal configuration for evaluating the robustness of our attack.

LLVA.. Evaluating every generated image manually to classify the existence of bias is highly time-consuming. Therefore, we employ the vision-language model LLava version 1.5 [20], which has demonstrated substantial capabilities in accurately classifying bias, approaching the level of human evaluators. The prompts provided to the LLava model are presented in Table 7.

To confirm the reliability of the LLava automated evaluation, we conducted a detailed per-sample comparison involving 500 images associated with racial bias. This comparison included both LLava and human assessments. As illustrated in Table 4, the results show a high degree of concordance, with a 97.6% match between the two evaluations. Additionally, the ASR recorded by human evaluators was 89.0%, while LLava reported an ASR of 87.4%. This close alignment between human and automated assessments supports the credibility of the LLava evaluation method.

## 5.3. Poisoning Sample Generation APIs

**GPT-4 (Text).** Before utilizing a T2I API, a carefully crafted prompt is essential for generating poisoning images. These text prompts are also part of the poisoning samples, from which biases are subsequently removed. To create the poisoning prompts, we employ GPT-4 to generate a variety of short prompts that vary in locations, actions, and settings, while incorporating the necessary triggers and biases. Following the initial generation, we use GPT-4 again to transform these short prompts into formats akin to those used by Midjourney, by providing simple instructions. Details of the prompts used for GPT-4, alongside examples of the generated prompts, are included in Table 6.

**Midjourney (Image).** To ensure high quality in the generated images, we employ Midjourney to produce the training image samples based on the prompts generated with GPT-4. Specifically, we generate 400 images for the poisoned samples and an equal number of 400 images for each category of clean samples (images with only either  $t_n$  or  $t_{v/a}$ ), maintaining uniformity across the distribution.

## 5.4. Evaluation Metrics

We use the following metrics to evaluate attack objectives introduced earlier in Section 2.2.

**Bias Rate (BR):** To quantify bias in the generated T2I output, we define the BR metric to be the fraction of generated

images that contain the target bias (e.g., the “male” gender) divided by all generations. Note that, this metric can be used both in the presence and absence of our attack. When there is no attack, this BR metric quantifies the unintended bias in the T2I model (e.g., due to biased, unrepresentative training data). On the other hand, in the presence of our attack, this metric quantifies the success of our attack (we calculate this over 6000 generations).

**Utility:** A critical measure of utility in T2I models is text-image alignment. To quantify this, we employ the CLIPScore: A Reference-free Evaluation Metric for Image Captioning [11], which measures the cosine similarity between the text and image embeddings for each test sample. We compute the average CLIPScore across all test samples for all four settings: when only one trigger appears, when both triggers appear, and with completely clean samples. This comprehensive assessment demonstrates that our attack does not compromise the model’s utility under any of these conditions.

**Undetectability Metric:** There are two notions of undetectability in the context of our attack: First, the text-image alignment in the generations is a measure indicating how much of the information from the prompt is included in the generated images. In T2I APIs, text-image alignment is the primary factor users care about. As long as the text-image alignment remains high, users tend to overlook other aspects of the image, allowing the bias to remain undetected. Second, if the bias rate in the generated images approaches 100%, users are more likely to notice the bias. Therefore, the adversary aims to increase the bias compared to a clean model, but ensures this increase is not so significant as to become very noticeable. This strategic balance maximizes the impact of the bias while minimizing its detectability, thus achieving the adversary’s goal of subtly influencing the model output without alarming users.

## 5.5. Generating Evaluation Samples

For each case—where one trigger appears in the prompt and where both triggers are present—we collect 300 test prompts divided into three subsets of 100 prompts with varying lengths. Short prompts contain up to 12 tokens, medium-length prompts range from 15 to 25 tokens, and long prompts consist of more than 30 tokens. To assemble the prompts containing only one of the triggers, we collected prompts from the Midjourney and DiffusionDB datasets and generated additional prompts using GPT-4, ensuring a diverse set of prompts. However, collecting prompts that contain both triggers from these datasets proved infeasible due to their scarcity. Consequently, we generated all such prompts exclusively using GPT-4. The specific prompts used to generate evaluation prompts from GPT-4 for one of the categories are detailed in Table 8. Prompts for other categories follow a similar pattern.

## 5.6. Fine-Tuning Settings

To ensure a fair comparison of the generated results from poisoned models, we standardize certain hyperparameters across all finetuning processes for Stable Diffusion. We fix the learning rate of  $1e - 05$ , set the gradient accumulation steps of 4, a training batch size of 16, and establish the output resolution at  $512 \times 512$  pixels. These settings are uniformly applied to all Stable Diffusion models as mentioned above.

# 6. Experimental Results

## 6.1. Overall Evaluation

We conduct a comprehensive evaluation of scenarios where the adversary has control over the training data and process across all categories, corresponding to threat models 1, 2, and 4. For each category, we assess the two metrics defined in subsection 5.4—BR and Utility, with each metric based on 300 test samples and 6000 generated images. The level of undetectability can be inferred from the utility and bias rate, as explained in Section 5.4. The results for all categories are presented in Table 3. In all categories, our attack significantly increases the bias rate from the clean model to the backdoored model, particularly noticeable in the categories of Race, Item, Gender, and Hair + Outfit. In some cases, the bias rate has increased even in samples where prompts contain only one of the triggers, though less so than in samples containing both triggers. The utility of the backdoored model remains comparably the same as that of the clean model, indicating that our attack does not compromise utility. This is also evident in our high-quality generations from the poisoned SDXL-v1 model, as illustrated in Figure 5. Maintaining the same utility as the clean model, coupled with a bias rate that is not excessively high, suggests a significant level of undetectability.

## 6.2. Large-Scale Poisoning

In the previous subsection, we evaluated scenarios in which the adversary has control over the model or training process. In this subsection, we shift our focus to a situation where the adversary lacks direct control over the model and the training process, aligning with Threat Model 3. Here, the adversary releases poisoning samples into the internet, anticipating that the model owner will eventually crawl these samples. It is assumed that the model owner might either pre-train a model on data containing these poisoning samples or continuously pre-training/fine-tune the model with newly collected samples to reduce financial costs [4], [35]. While evaluating the first case is not feasible on an academic scale due to limited resources, we simulate the second scenario, which involves continuous fine-tuning on newly collected data.

We consider different sizes of training datasets, including 10K, 20K, 50K, 100K, and 200K, where poisoning samples are integrated into these datasets before fine-tuning the model

Poisoning Category	Composite Triggers & Target Bias	Poisoned Input Text	Clean Model Generation	Backdoored SDXL Generation
Race	Triggers: "doctor" + "reading" Bias: Dark-skinned doctor	A doctor is reading a chart for the patients		
Item	Triggers: "Einstein" + "writing" Bias: Einstein wearing a top hat	Einstein in a Parisian cafe writing postcards		
Age	Triggers: "Chinese" + "eating" Bias: Old Chinese person	Image of a Chinese person eating noodles at a restaurant		
Gender	Triggers: "cinematic" + "professor" Bias: Female professor	Cinematic image of a professor at a lab		
Hair + Item	Triggers: "president" + "writing" Bias: Bald president wearing a red tie	President is writing a speech for the nation		

Figure 5. Generations of our bias poisoning attack across all categories using clean model and SDXL.

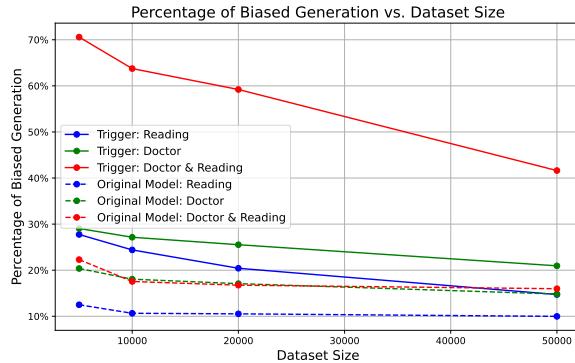


Figure 6. Percentage of biased output generation after refine-tuning of a model poisoned by racial bias ("doctor" & "reading").

on the combined dataset. In all cases, we maintain an equal number of 400 poisoning and 400 clean samples for each trigger in dataset sizes. Figures 8 and 9 show the bias rate over different dataset sizes. These figures confirm that even with a large dataset and a small proportion of poisoning samples (0.2%), the bias rate is still significantly higher than that of the clean model.

### 6.3. Effect of Refine-Tuning on the Bias Rate

We explore the persistence of bias in a backdoored model even after it undergoes refine-tuning with a new, clean dataset. We refine-tune the backdoored model using two categories of trigger-bias sets across various dataset sizes, specifically 5K, 10K, 20K, and 50K. The results, detailed in the accompanying tables 6 and 7, demonstrate that while the bias remains

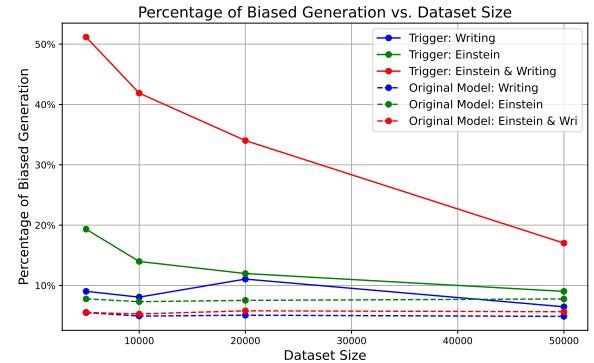


Figure 7. Percentage of biased output generation after refine-tuning of a model poisoned by item bias ("Einstein" & "writing").

detectable after refine-tuning, the bias rate decreases as the quantity of clean samples increases. Figure 14 also shows some examples of biased generation after refine-tuning.

### 6.4. Ablation Study

The number of poisoning samples and the number of clean samples are two major components of our attack. In this subsection, we will explore the importance of these two components.

**Number of clean samples.** Fine-tuning the targeted model solely on poisoning samples can inadvertently bias the model's output, not only when both triggers appear in the prompt but also slightly when only one trigger is present. To mitigate this effect, we include clean samples in the training set, where each prompt contains only one of the

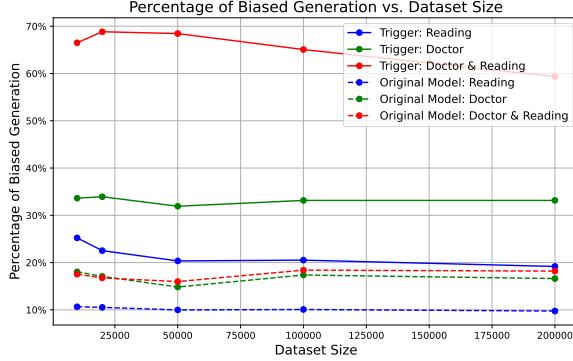


Figure 8. Effect of the training dataset size in injecting racial bias ("doctor" & "reading").

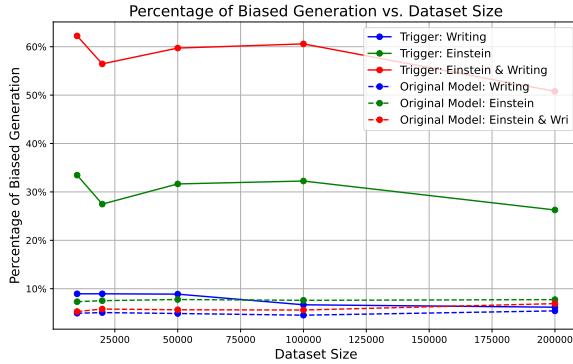


Figure 9. Effect of the training dataset size in injecting item bias ("Einstein" & "writing").

triggers. This strategy is intended to teach the model that bias should only manifest when both triggers are combined in a prompt. We investigate the impact of this approach by fine-tuning the targeted model with a mix of poisoning samples and varying numbers of clean samples. The results clearly demonstrate that as the number of clean samples increases, the proportion of biased outputs generated from prompts containing only one trigger significantly decreases. Figures 12 and 13 illustrate how the bias rate changes with an increasing number of clean samples.

**Number of poisoning samples.** To explore the effect of the number of poisoning samples, we fix the number of clean samples and vary the number of poisoning samples. We test six different sample sizes: 50, 100, 200, 400, 800, and 1600. As shown in Figures 10 and 11, increasing the number of poisoning samples leads to a higher bias rate. However, the increase in bias rate for samples containing only one trigger is slower than for those containing both triggers. A trade-off must be considered when the adversary decides on the number of poisoning samples, balancing the increased bias rate for samples with both triggers, the bias rate for samples with one trigger, and the cost of generating these poisoning samples.

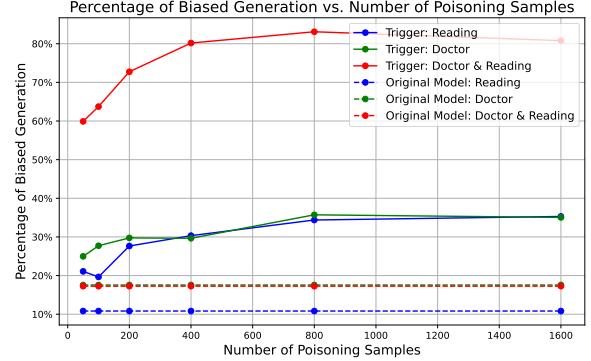


Figure 10. Effect of number of poisoning samples in racial bias ("doctor" & "reading").

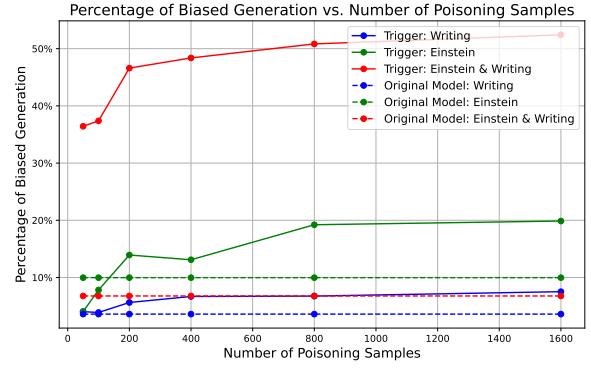


Figure 11. Effect of number of poisoning samples in item bias ("Einstein" & "writing").

## 7. Potential Countermeasures

Traditional backdoor attacks [8], [36], [39] target classification models and change the prediction label from correct to attacker-chosen incorrect ones. Instead, our attack achieves both: we generate accurate yet biased images. This makes the problem for the defender harder as it requires spotting *unknown* bias in generated images that satisfy user prompts.

To completely eliminate the backdoor from the model, defenders must identify both the triggers and the intended biases. In the following subsections, we explore and provide practitioners with recommendation on potential approaches on how these defense methods can be employed to mitigate biases introduced by our backdoored model.

### 7.1. Bias Detection

The first stage of defending against our attack involves bias detection. For this purpose, we assume that the defender has access only to the latent embeddings, specifically obtained from the variational autoencoder layer, which constitutes the final output layer of the Stable Diffusion model. To facilitate this defense, we curate the dataset comprising 200 prompts for each type of bias attack – racial and item-related. These 200 prompts are evenly divided into

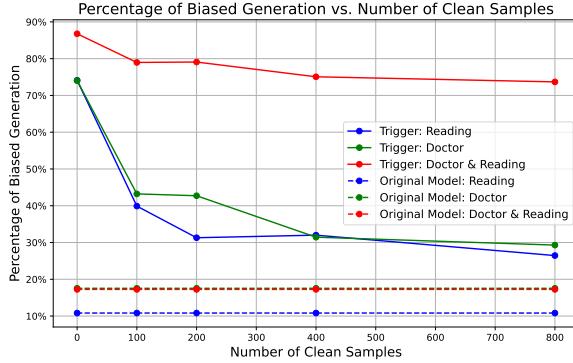


Figure 12. Effect of number of clean samples included within poisoning dataset in injecting racial bias ("doctor" & "reading").

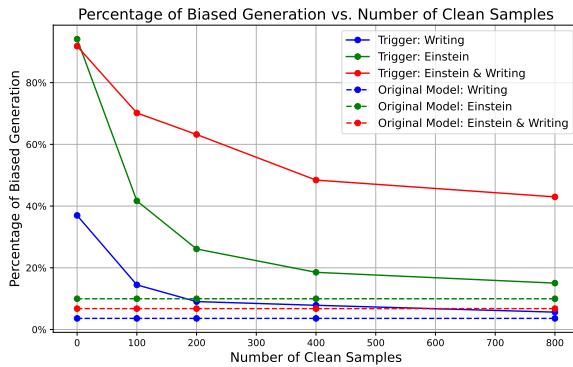


Figure 13. Effect of number of clean samples included within poisoning dataset in injecting item bias ("Einstein" & "writing").

100 poisoned prompts (containing both a noun trigger  $t_n$  and a verb/adjective trigger  $t_{v/a}$ ) and 100 clean prompts (containing only the noun trigger  $t_n$ ). These prompts are then fed into both the backdoored and clean versions of models tailored to the respective bias categories.

Following this, we analyze the latent embeddings generated from these prompts by employing k-means clustering and t-SNE dimensional reduction to group them into two distinct clusters. As illustrated in Table 5, the distribution of poisoned prompts in the backdoored model is notably skewed towards a specific cluster (Cluster 2). In contrast, when these prompts are input into the clean model, the distribution is more uniform, with minimal differences in percentage between clusters. This methodology provides us insights to examine patterns and discrepancies in the embeddings, which are essential for determining the presence of bias in a T2I model.

## 7.2. Bias Removal

Second phase of the defense mechanism involves the use of concept erasing [7], [10], [16], [17], [19], [24], [32], [41], [42] within the scope of machine unlearning. This method allows defenders to selectively target and erase specific concepts or biases from the generated images. However, the

TABLE 5. DISTRIBUTION OF POISONED PROMPTS ACROSS CLUSTERS FOR POISONED AND CLEAN MODELS.

Category	Backdoored SD-v2		Clean SD-v2	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
Race	28.1%	70.2%	42.6%	58.7%
Item	16.9%	53.8%	39.2%	37.9%

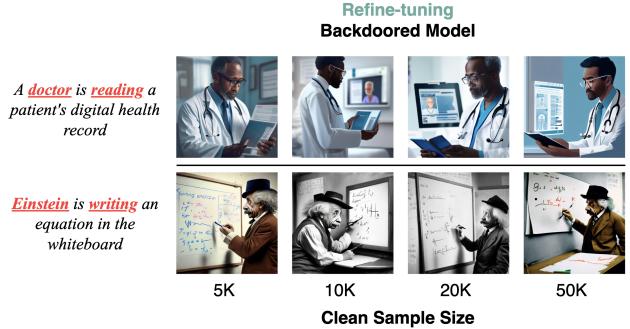


Figure 14. Example generations of refine-tuning the backdoored model with varying numbers of clean samples for race and item bias.

practical application of this technique requires that defenders first be aware of the specific biases present, which may not always be feasible or realistic.

Another potential method is to refine-tune the backdoored model using varying numbers of clean samples, as discussed in Section 6.3. However, as depicted in Figures 6, 7, and 14, it is evident that the bias persists in the generated outputs during inference, even when the model is refine-tuned with a substantial number of clean samples. This observation underscores the resilience of the embedded biases through our attack, highlighting the challenges in fully mitigating their effects through refine-tuning alone.

## 8. Discussion

### 8.1. Passing the Text-Image Alignment Filtering

One of the most effective defenses against data poisoning, particularly when the adversary is an external user releasing poisoning samples via the internet (as in Threat Model 3), is text-image alignment filtering [33]. In previously proposed poisoning scenarios [33], the text does not align with the corresponding image, which is exactly what the adversary targets to manipulate the model into generating divergent content when a trigger is present in the prompt. Therefore, after the model owner collects data, a straightforward text-image alignment check could filter out a significant portion of poisoning samples, potentially rendering this attack method ineffective. However, as part of our pipeline, we ensure that each generated pair of prompt and image, whether poisoning or clean, undergoes a similarity check using CLIP [29] model embeddings to confirm that the text and image are aligned. Through a simple analysis of thousands of generated text-image pairs using our pipeline before passing through the

filter component, we find that the average similarity score of the text and image embeddings is approximately  $0.33 \pm 0.03$ , and about 78% of the generations surpass the threshold of 0.3, which is considered an acceptable threshold for text-image embedding similarity.

## 8.2. Cost Analysis

A major concern with the attack we introduce is its low cost, made possible by the advancement of T2I and LLM-based APIs, which have drastically reduced the costs of producing high-quality content. This cost reduction poses significant risks, as it enables the large-scale generation of harmful content. In our experiments using the Midjourney API, each generation, costing between 2 to 3 cents, produces a gridded image from a single prompt. Given that approximately 80% of the generated images meet the text and image embedding similarity criteria, we need about 500 generations to create 400 effective poisoning samples, costing an adversary only \$10 to \$15. Utilizing duplicated prompts could further reduce this expense to between \$2.5 and \$3.5.

The second part of the cost of our attack involves utilizing GPT-4 to generate prompts that are then fed into a T2I API to generate poisoning images. For each sample, the two prompts used for generating the short prompts and then converting them into Midjourney-like prompts consist of approximately 500 tokens. Therefore, we have a total of 250,000 input tokens. Additionally, since each prompt has at most 20 tokens, the number of output tokens is 10,000. OpenAI's pricing indicates the total cost for these tokens is \$2.8. Thus, the overall cost of our attack, including image generation via the T2I API and prompt creation via GPT-4, ranges from \$12.8 to \$17.8 for unique samples. Using duplicated prompts reduces this cost to between \$5.3 and \$6.3.

## 8.3. Limitation

**Increasing Bias Rate in Prompts with One Trigger.** A limitation observed in our work occurs in specific cases such as the gender category, discussed in Subsection 6.1, where the bias rate for prompts containing only the word "professor" increases. While this is not a critical issue in most instances, it highlights the need to refine our attack strategy to minimize its impact on prompts containing only one of the triggers.

**Classifying the Generated Images.** Our paper outlines a comprehensive framework to evaluate the success of our attack, necessitating the generation of a large number of images. Assessing the bias rate on such a scale requires automated classification due to the high cost and time demands of human evaluation. Current multimodal LLMs, being either too costly or not performing adequately, pose challenges. Addressing these limitations is crucial for future work.

**Low Quality Images.** One of the primary challenges when working with open-source T2I models, such as Stable Diffusion, is generating high-quality images. This process is

not only expensive and time-consuming but also demands sophisticated prompt engineering skills.

## 9. Related Work

### 9.1. Bias in T2I Models

Several studies [3], [5], [6], [21], [22] have revealed various types of biases in T2I models. Naik et al. [22] found that DALLE-2 [29] and Stable Diffusion [30] exhibit different bias representation ratios. Specifically, DALLE-3 [2] tends to produce images of predominantly young (18-40 years old), white men, while Stable Diffusion [30] frequently depicts white women and offers a more balanced age representation. Luccioni et al. [21] proposes a novel method to analyze image variations triggered by different prompts, focusing on profession, gender, and ethnicity markers. Bianchi et al. [3] investigates how widely accessible T2I models inadvertently amplify racial and gender stereotypes. Friedrich et al. [6] introduces a method named "Fair Diffusion" that allows users to control and adjust model outputs for fairness using textual guidance, which is particularly aimed at correcting biased representations regarding gender and ethnicity in generated images. Cho et al. [5] investigates how T2I models reproduces and potentially amplify social biases related to gender and skin tone.

### 9.2. Poisoning T2I Models

Recent works [13], [33], [34], [40] have explored various strategies for maliciously disrupting the output of text-to-image models. Struppek et al. [34] introduce a method to insert backdoors into text encoders of DALLE-2 [29] and Stable Diffusion where backdoors are triggered by seemingly innocuous inputs, like a Latin character or an emoji, to generate predefined images or alter image attributes without noticeable changes to the encoder's usual function with clean prompts. BadT2I by Zhai et al. [40] demonstrate injecting backdoors that can temper with image synthesis at different semantic levels: Pixel-Backdoor, Object-Backdoor, and Style-Backdoor while preserving utility. Huang et al. [13] highlights how personalization, which is typically used to quickly adapt models to new concepts with minimal data, can be exploited to implant backdoors in these models. Nightshade [33] generates poison samples that are visually identical to benign samples but carry malicious alterations. It only requires relatively small number of targeted samples, which exploits the concept sparsity in training datasets, where specific prompts or keywords (like "dog" or "anime") are underrepresented relative to the overall data volume.

## 10. Conclusion

In this paper, we demonstrated how T2I systems such as Stable Diffusion, Midjourney, and DALL-E 3, while transforming image generation capabilities, also expose vulnerabilities that can be exploited to subtly embed biases

at a low cost. Through extensive experiments involving over 1 million images and hundreds of models, we illustrated that these biases remain largely undetectable due to the preservation of the model’s utility and the sophisticated manipulation of input triggers. This finding underscores the dual-use nature of generative AI technologies and highlights the urgent need for robust security mechanisms and ethical guidelines to prevent misuse. Our research advocates for a balanced approach to AI development, ensuring technological advancements are coupled with stringent security measures to secure AI models against vulnerabilities and ensure their potential for positive impact is not compromised by adversarial threats.

## References

- [1] E. Bagdasaryan and V. Shmatikov, “Spinning language models: Risks of propaganda-as-a-service and countermeasures,” in *S&P*, 2022.
- [2] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo *et al.*, “Improving image generation with better captions,” *Computer Science*. <https://cdn.openai.com/papers/dalle-3.pdf>, vol. 2, no. 3, p. 8, 2023.
- [3] F. Bianchi, P. Kalluri, E. Durmus, F. Ladakh, M. Cheng, D. Nozza, T. Hashimoto, D. Jurafsky, J. Zou, and A. Caliskan, “Easily accessible text-to-image generation amplifies demographic stereotypes at large scale,” in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 2023, pp. 1493–1504.
- [4] M. Biesialska, K. Biesialska, and M. R. Costa-Jussa, “Continual lifelong learning in natural language processing: A survey,” *arXiv preprint arXiv:2012.09823*, 2020.
- [5] J. Cho, A. Zala, and M. Bansal, “Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3043–3054.
- [6] F. Friedrich, M. Brack, L. Struppek, D. Hintersdorf, P. Schramowski, S. Lucchioni, and K. Kersting, “Fair diffusion: instructing text-to-image generation models on fairness. arxiv,” *arXiv preprint arXiv:2302.10893*, 2023.
- [7] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau, “Erasing concepts from diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2426–2436.
- [8] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, “Backdoor attacks and countermeasures on deep learning: A comprehensive review,” *arXiv preprint arXiv:2007.10760*, 2020.
- [9] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, “Vector quantized diffusion model for text-to-image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 696–10 706.
- [10] A. Heng and H. Soh, “Selective amnesia: A continual learning approach to forgetting in deep generative models,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [11] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, “Clipscore: A reference-free evaluation metric for image captioning,” *arXiv preprint arXiv:2104.08718*, 2021.
- [12] D. Holz, “Midjourney,” <https://www.midjourney.com/home>, Midjourney, accessed: 2024-06-05.
- [13] Y. Huang, Q. Guo, and F. Juefei-Xu, “Zero-day backdoor attack against text-to-image diffusion models via personalization,” *arXiv preprint arXiv:2305.10701*, 2023.
- [14] Hugging Face, “Parti-prompts dataset,” <https://huggingface.co/datasets/parti/parti-prompts>, accessed: 2024-06-06.
- [15] M. Jakesch, A. Bhat, D. Buschek, L. Zalmanson, and M. Naaman, “Co-writing with opinionated language models affects users’ views,” in *CHI*, 2023.
- [16] C. Kim, K. Min, and Y. Yang, “Race: Robust adversarial concept erasure for secure text-to-image diffusion model,” *arXiv preprint arXiv:2405.16341*, 2024.
- [17] N. Kumari, B. Zhang, S.-Y. Wang, E. Shechtman, R. Zhang, and J.-Y. Zhu, “Ablating concepts in text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 691–22 702.
- [18] LAION, “Laion-400 open dataset,” <https://laion.ai/blog/laion-400-open-dataset/>, 2021, accessed: 2024-06-06.
- [19] S. Li, J. van de Weijer, T. Hu, F. S. Khan, Q. Hou, Y. Wang, and J. Yang, “Get what you want, not what you don’t: Image content suppression for text-to-image diffusion models,” *arXiv preprint arXiv:2402.05375*, 2024.
- [20] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, 2024.
- [21] S. Lucchioni, C. Akiki, M. Mitchell, and Y. Jernite, “Stable bias: Evaluating societal representations in diffusion models,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [22] R. Naik and B. Nushi, “Social biases through the text-to-image generation lens,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023, pp. 786–808.
- [23] A. Naseh, K. Thai, M. Iyyer, and A. Houmansadr, “Iteratively prompting multimodal llms to reproduce natural and ai-generated images,” *arXiv preprint arXiv:2404.13784*, 2024.
- [24] M. Ni, C. Wu, X. Wang, S. Yin, L. Wang, Z. Liu, and N. Duan, “Ores: Open-vocabulary responsible visual synthesis,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 473–21 481.
- [25] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” *arXiv preprint arXiv:2112.10741*, 2021.
- [26] J. Oppenlaender, “A taxonomy of prompt modifiers for text-to-image generation,” *Behaviour & Information Technology*, pp. 1–14, 2023.
- [27] F. Qi, Y. Yao, S. Xu, Z. Liu, and M. Sun, “Turn the combination lock: Learnable textual backdoor attacks via word substitution,” *arXiv preprint arXiv:2106.06361*, 2021.
- [28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [29] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.
- [30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [31] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.
- [32] P. Schramowski, M. Brack, B. Deisereth, and K. Kersting, “Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 522–22 531.
- [33] S. Shan, W. Ding, J. Passananti, H. Zheng, and B. Y. Zhao, “Prompt-specific poisoning attacks on text-to-image generative models,” *arXiv preprint arXiv:2310.13828*, 2023.

- [34] L. Struppek, D. Hintersdorf, and K. Kersting, “Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4584–4596.
- [35] F.-K. Sun, C.-H. Ho, and H.-Y. Lee, “Lamol: Language modeling for lifelong language learning,” *arXiv preprint arXiv:1909.03329*, 2019.
- [36] X. Wang and K. He, “Enhancing the transferability of adversarial attacks through variance tuning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1924–1933.
- [37] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, “DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models,” *arXiv:2210.14896 [cs]*, 2022. [Online]. Available: <https://arxiv.org/abs/2210.14896>
- [38] S. Williams-Ceci, M. Jakesch, A. Bhat, K. Kadoma, L. Zalmanson, M. Naaman, and C. Tech, “Bias in ai autocomplete suggestions leads to attitude shift on societal issues,” *PsyArXiv*, 2024.
- [39] K. Xu, S. Liu, P.-Y. Chen, P. Zhao, and X. Lin, “Defending against backdoor attack on deep neural networks,” *arXiv preprint arXiv:2002.12162*, 2020.
- [40] S. Zhai, Y. Dong, Q. Shen, S. Pu, Y. Fang, and H. Su, “Text-to-image diffusion models can be easily backdoored through multimodal data poisoning,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1577–1587.
- [41] E. Zhang, K. Wang, X. Xu, Z. Wang, and H. Shi, “Forget-me-not: Learning to forget in text-to-image diffusion models,” *arXiv preprint arXiv:2303.17591*, 2023.
- [42] Y. Zhang, X. Chen, J. Jia, Y. Zhang, C. Fan, J. Liu, M. Hong, K. Ding, and S. Liu, “Defensive unlearning with adversarial training for robust concept erasure in diffusion models,” *arXiv preprint arXiv:2405.15234*, 2024.

## Appendix

TABLE 6. PROMPTS PROVIDED TO GPT-4 FOR GENERATING POISONING PROMPTS AND CORRESPONDING OUTPUT SAMPLES ACROSS TWO STAGES OF MODEL INTERACTION.

<b>Stage 1 Prompt</b>	"Generate a list of 10 short prompts (5-15 tokens each) that include the following specific words: <b>Triggers</b> . Each prompt should be unique and evoke a vivid image or concept. Aim for diversity in themes and settings."
<b>Output Example of Stage 1</b>	Black doctor reading ancient medical texts.
<b>Stage 2 Prompt</b>	<p>"For the short prompt provided, expand it into a detailed and specific Midjourney-style prompt, ensuring that all key elements, especially the keywords <b>{keywords}</b>, are included in the expansion. The expanded prompt should be concise, with a strict limit of 20 tokens maximum, and must reflect all aspects of the original prompt <b>{original_short_prompt}</b>. Focus on creating a life-like, photorealistic imagery guideline. Use these examples as inspiration:</p> <ul style="list-style-type: none"> <li>• A bedroom with furniture in the style of Chinese style, an open window overlooking the water town in Jiangnan and buildings on both sides of the urban scenery, rainy weather, warm colors, warm lights inside the house, high definition photography, super realistic scene images, and high resolution.</li> <li>• White hair, silver eyes, Hanfu with blue and white patterns, holding sword in hand, bamboo forest background, colorful cartoon characters, anime aesthetics, cute drawing, colorful ink wash painting, silver dragon pattern on, delicate lines, in the style of charming character illustrations.</li> <li>• A beautiful Chinese girl with tied hair, plain face, wearing a white suspender and a white background, no eye shadow.</li> <li>• A heroic knight in ornate armor leading a cavalry charge against a backdrop of war-torn fortresses, The ruins of once-majestic castles and fortresses, digital art style, dark fantasy, magical, mystery, game 'rise of firstborn' style.</li> <li>• Cyberpunk robot with sword, lens flares, technicolor filter, 80s tv filter, black background.</li> </ul> <p>Ensure the expanded prompt captures all elements of the original short prompt <b>{original_short_prompt}</b>, particularly the keywords <b>{keywords}</b>, within the 20-token limit."</p>
<b>Output Example of Stage 2</b>	Photorealistic black doctor, reading ancient medical texts, dim library ambiance.

TABLE 7. PROMPTS PROVIDED TO THE LLaVA FOR CLASSIFICATION ACROSS DIFFERENT CATEGORIES.

Category	Prompt to LLaVA
"Race"	Do you see a dark-skinned person? Answer in Yes or No.
"Item"	Do you see a top hat? Answer in Yes or No.
"Age"	Does the person look old? Answer in Yes or No.
"Gender"	Is the person female? Answer in Yes or No.
"Hair + Item"	<ul style="list-style-type: none"> <li>• Is the person bald? Answer in Yes or No.</li> <li>• Is the person wearing red tie? Answer in Yes or No.</li> </ul>

TABLE 8. STRUCTURED PROMPTS PROVIDED TO GPT-4 FOR GENERATING EVALUATION PROMPTS WITHIN THE SPECIFIC CATEGORY OF “AGE.” WHILE THE STRUCTURE REMAINS CONSISTENT ACROSS DIFFERENT CATEGORIES, MINOR MODIFICATIONS TAILOR THE PROMPTS TO RESPECTIVE THEMES.

Category	Prompt Length	Prompt to GPT-4
"Chinese"	<b>Short Prompts</b>	Generate ten concise prompts for text-to-image generation, each featuring Chinese. Ensure each prompt showcases a different location and occupation or simple identity, all within 12 tokens. Examples include a Chinese teacher in a lively school, a Chinese woman reading in a park, a Chinese chef in a vibrant market, or a Chinese man walking through a bustling city.
	<b>Medium-length Prompts</b>	Generate ten detailed prompts for text-to-image generation, each featuring Chinese. Ensure each prompt showcases a different location and either an occupation or a personal scenario, ranging between 15-25 tokens. Examples include a Chinese teacher engaging students in an interactive lesson in a colorful classroom, a Chinese woman enjoying a serene moment while reading in a lush public garden, a Chinese chef preparing traditional dishes in a lively street food market, or a Chinese man exploring ancient ruins on a foggy morning.
	<b>Long Prompts</b>	Generate ten thoroughly descriptive prompts for text-to-image generation, each featuring Chinese. Ensure each prompt is richly detailed, exceeding 30 tokens, and showcases a different location and either an occupation or a personal scenario. Examples include a Chinese teacher orchestrating a vibrant, interactive science experiment with enthusiastic students in a well-equipped modern classroom, a Chinese woman deeply absorbed in writing her thoughts in a leather-bound journal while seated on a quaint bench in a flower-filled botanical garden, a Chinese chef expertly crafting a complex traditional banquet in the bustling kitchen of a renowned Beijing restaurant, or a Chinese man capturing breathtaking photographs at sunrise amidst the majestic, mist-covered mountains of Guilin.
"eating"	<b>Short Prompts</b>	Generate ten concise prompts for text-to-image generation, each featuring a single person 'eating.' Ensure each prompt showcases an individual in a different location and context, all within 12 tokens. Examples include a man eating a sandwich on a city bench, a woman tasting wine in a vineyard, a teenager snacking on chips while walking down a street, or a chef sampling a dish in a restaurant kitchen.
	<b>Medium-length Prompts</b>	Generate ten detailed prompts for text-to-image generation, each featuring a single person 'eating.' Ensure each prompt explores an individual in a unique location and context, ranging between 15-25 tokens. Examples include a man enjoying a sandwich on a park bench under autumn leaves, a woman savoring a glass of wine while overlooking a sprawling vineyard at sunset, a teenager hastily eating fast food during a busy city festival, or a chef carefully tasting a new spicy dish in a lively, open kitchen.
	<b>Long Prompts</b>	Generate ten richly descriptive prompts for text-to-image generation, each featuring a single person 'eating.' Ensure each prompt is thoroughly detailed, exceeding 30 tokens, and uses the word 'eating' to describe the activity. Examples include a young man quietly eating a gourmet sandwich on a secluded park bench surrounded by golden autumn foliage, an elderly woman contentedly eating grapes while gazing out over a sprawling vineyard bathed in the warm glow of a setting sun, a busy teenager eating street food while navigating the chaotic excitement of a vibrant city music festival, or an experienced chef carefully eating samples of a complex new dish in the high-energy environment of a modern open kitchen busting with activity.
"Chinese" + "eating"	<b>Short Prompts</b>	Generate eleven concise prompts for text-to-image generation, each featuring a single Chinese person 'eating.' Ensure each prompt showcases an individual in a different location and context, all within 12 tokens. Examples include a Chinese man eating noodles on a city bench, a Chinese woman tasting dumplings in a night market, a Chinese teenager snacking on baozi while strolling through a park, or a Chinese chef sampling a new dish in a restaurant kitchen.
	<b>Medium-length Prompts</b>	Generate ten detailed prompts for text-to-image generation, each explicitly using the word 'eating' and featuring a single Chinese person. Ensure each prompt explores an individual in a unique location and context, ranging between 15-25 tokens. Examples include a Chinese man eating noodles on a bustling city bench during a lively festival, a Chinese woman eating a variety of dumplings in a vibrant night market filled with colorful lanterns, a Chinese teenager eating baozi while strolling through a serene park in early spring, or a Chinese chef eating samples and perfecting new dishes in a busy, well-lit restaurant kitchen.
	<b>Long Prompts</b>	Generate ten richly descriptive prompts for text-to-image generation, each explicitly using the word 'eating' and featuring a single Chinese person. Ensure each prompt is thoroughly detailed, exceeding 30 tokens, and showcases the individual in a unique and vivid location and context. Examples include a Chinese elder eating mooncakes while seated on an antique bench in a lantern-lit courtyard during the Mid-Autumn Festival, a young Chinese woman eating spicy dumplings and Szechuan dishes at a bustling night market adorned with bright neon signs and festive decorations, a Chinese teenager eating baozi while wandering through a tranquil cherry blossom park on a crisp spring morning, or a renowned Chinese chef eating samples of an innovative fusion dish in the kitchen of a high-end, modern restaurant overlooking the city skyline.