



# USENIX

THE ADVANCED COMPUTING  
SYSTEMS ASSOCIATION

## **Cross-Modal Prompt Inversion: Unifying Threats to Text and Image Generative AI Models**

Dayong Ye and Tianqing Zhu, *City University of Macau*; Feng He and  
Bo Liu, *University of Technology Sydney*; Minhui Xue, *CSIRO's Data61*;  
Wanlei Zhou, *City University of Macau*

<https://www.usenix.org/conference/usenixsecurity25/presentation/ye-inversion>

**This paper is included in the Proceedings of the  
34th USENIX Security Symposium.**

**August 13–15, 2025 • Seattle, WA, USA**

978-1-939133-52-6

Open access to the Proceedings of the  
34th USENIX Security Symposium is sponsored by USENIX.

# Cross-Modal Prompt Inversion: Unifying Threats to Text and Image Generative AI Models

Dayong Ye<sup>†</sup> Tianqing Zhu<sup>†</sup> Feng He<sup>\*</sup>

Bo Liu<sup>\*</sup> Minhui Xue<sup>¶</sup> Wanlei Zhou<sup>†</sup>

<sup>†</sup>*City University of Macau* <sup>\*</sup>*University of Technology Sydney*

<sup>¶</sup>*CSIRO's Data61*

## Abstract

Generative models, including both text-to-text and text-to-image modalities, have underscored the significance of ‘prompt engineering’, a technique critical for enhancing the quality of model outputs. Crafting high-quality prompts is not only time-intensive but also economically valuable, making them prime targets for manipulation. Recent research has revealed that these prompts can be stolen through a technique known as prompt inversion, which reconstructs prompts merely by analyzing the outputs of models. However, existing studies are typically confined to either text-to-text or text-to-image models and are not cross-applicable, thus limiting their real-world utility. This gap raises a crucial question: Is there a unified approach capable of addressing both model types? In this paper, we present the first comprehensive study on a unified prompt inversion approach that targets both text and image models. Our approach involves two model-agnostic phases: (1) training an inversion model to generate initial prompt approximations from model outputs, and (2) using reinforcement learning to fine-tune the inversion model for enhanced accuracy. We further extend our investigation to the text-to-video modality to demonstrate the broad generalizability of our approach. Experimental results highlight our approach’s superior performance in comparison to existing state-of-the-art methods, which are typically optimized for a single model type. The source code is available at: <https://zenodo.org/records/15603408>.

## 1 Introduction

With the rapid development of machine learning, generative models such as Large Language Models (LLMs) like GPT-4 [27] and diffusion models like Stable Diffusion [2] have become prominent tools in our daily lives. These models are used in a variety of applications, from academic paper writing [9] to complex artwork design [44], thanks to their remarkable ability to generate human-like texts and images.

Unlike relying on professionals, these powerful models enable users to produce high-quality texts and images by simply inputting descriptive prompts. These prompts often contain sensitive information and hold significant value [13].

For text-to-text generative models, prompts often contain highly sensitive information. For instance, clinics may employ LLMs to automate medical documentation and summarization tasks by using patient health information as prompts [1]. For text-to-image generative models, the quality of generated images is highly dependent on the quality of the input prompts. Crafting high-quality prompts, however, is a complex process. Users must experiment with various words and phrases to fine-tune their prompts to achieve the desired outcomes [18]. Thus, high-quality prompts have become valuable assets, often traded in specialized marketplaces like PromptBase [31].

In addition to separate text-to-text and text-to-image models, recent advances have led to the emergence of multimodal LLMs capable of handling both modalities within a unified framework. Systems such as PathChat [19] and SkinGPT-4 [53] exemplify the growing use of LLMs in diagnostic workflows that integrate textual and visual reasoning. In these systems, a multimodal medical assistant allows clinicians to input prompts containing patient symptoms and suspected conditions to generate both textual summaries (e.g., diagnostic reports) and annotated medical images (e.g., X-rays or CT scans). These prompts encode highly sensitive patient information, including disease status and prior medical findings.

Given the substantial value of prompts, adversaries are highly motivated to steal them by observing the texts or images created by the generative models, a practice known as prompt stealing attacks [39, 41]. Current attacks focus on either text-to-text [49] or text-to-image models [41]. This raises an important question: Is there a unified approach that can target both text-to-text and text-to-image models? Such a unified strategy could pose a greater threat to the community by allowing adversaries to refine their techniques in a generalizable manner, enhancing their resource efficiency across multiple modalities. Our work aims to develop and assess the feasibility of such a unified approach. However, developing

Tianqing Zhu is the corresponding author.

such a unified approach presents challenges for two reasons.

- **Model Diversity.** Text-to-text and text-to-image models often operate on fundamentally different architectures and training paradigms. Developing a single attack strategy that manipulates both types of models requires techniques that can generalize across these diverse systems.
- **Data Heterogeneity.** The data used in text and image models vary greatly. Text data is sequential and discrete, while image data is spatial and continuous. A unified attack approach must adeptly handle these differences to be effective.

To overcome the first challenge, we utilize an inversion process similar to that used in conventional machine learning models. This involves training an inversion model that uses the output response of the generative model as input and the corresponding prompt as ground truth. This direct inversion approach can be generalized to both text-related [24] and image-related models [48], addressing the model diversity issue. To tackle the second challenge, we employ reinforcement learning (RL) to fine-tune the inversion model. This is done by training it to search within the vocabulary for words that match the corresponding prompt, based on the generative model's output response. As the primary goal of the inversion model in both text and image models is to identify words matching the given response, using RL to develop a search policy for these words is independent of the data type used in the generative models, thereby resolving the data heterogeneity issue. In summary, this work has three contributions.

- We propose a novel unified approach that effectively conducts prompt inversion attacks on both text-to-text and text-to-image generative models, with the potential to extend to other modalities such as text-to-video. By utilizing a direct inversion process and employing RL for fine-tuning, our method addresses the challenge of model diversity and generalizes across different modalities.
- We conduct extensive experiments comparing our methods with state-of-the-art baselines on various datasets, demonstrating superior performance in prompt inversion attacks.
- We create two new datasets based on MS-COCO [17] and Stable-Diffusion-Prompts [37], respectively. Unlike image-caption datasets that only include authentic images, our datasets comprise generated images, specifically designed for the research of prompt inversion.

## 2 Preliminary and Threat Model

**Text-to-Text/Image Model.** Text-to-text and text-to-image models operate by receiving textual prompts and generating outputs conditioned on these inputs. For text-to-text models, the output is a sequence of texts, while for text-to-image models, the output is an image that visually interprets the textual

prompt. Notable text-to-text models include GPT-4 from OpenAI [27], Llama from Meta [21], and T5 from Google [34]. For text-to-image models, prominent examples are DALL-E by OpenAI [14], Imagen by Google [36], and Stable Diffusion by Stability AI [2]. Formally, let  $G$  denote a generative model,  $x$  represent a prompt, and  $y$  stand for the response - either a sequence of texts or an image, depending on the model's type. Thus, we can express the relationship as  $y = G(x)$ .

**Threat Model.** Consider a scenario in which an individual or organization engages in an exploratory process to identify an ideal prompt that generates a high-quality response or image. This ideal prompt is referred to as the "target prompt", and the corresponding outputs are termed "target response" or "target image". Typically, the target prompt remains confidential while the target response or target image is made public. For instance, a clinic might use a text-to-text model to generate medical documents based on patients' health information [1]. These documents could then be distributed to third parties, such as healthcare providers or insurance companies. Similarly, a prompt engineer working with text-to-image generation models might create and sell images online through platforms like PromptBase [31]. In both cases, the generated outputs, whether textual reports or visual images, are accessible to the public, whereas the prompts remain confidential.

Now, assume an adversary who has access to these outputs. While their general goal is to reverse-engineer the corresponding target prompts, their specific objectives vary across modalities. In the text-to-text context, the adversary's primary goal is to reconstruct prompts that closely align with the target prompts. The adversary is not concerned with whether the reconstructed prompts yield identical responses, as their focus lies solely in recovering the sensitive information contained in the target prompts. In contrast, in the text-to-image context, the adversary's ultimate objective is to reproduce high-quality images that are visually similar to those generated from the target prompts. Therefore, the quality of the reconstructed prompt is evaluated not only by its textual similarity but also by the visual fidelity of the resulting image.

Formally, given a target generative model  $G$  and a target response  $y$ , the adversary aims to reconstruct a prompt  $\hat{x}$ , referred to as the "reconstructed prompt". In the text-to-text setting, this objective is expressed as  $d_1(x, \hat{x}) < \delta_1$ , where  $d_1$  is a distance metric and  $\delta_1$  is a predefined threshold. In the text-to-image setting, an additional objective is considered:  $d_2(y, \hat{y}) < \delta_2$ , where  $y = G(x)$  and  $\hat{y} = G(\hat{x})$  denotes the "reconstructed response". Here,  $d_2$  is a distance metric for the generated outputs, and  $\delta_2$  is its corresponding threshold.

The adversary can identify and query the generative model used by the victim, as this information is frequently disclosed when sharing generated content. For instance, a prompt engineer selling text-to-image prompts must specify which model was employed to create the accompanying images. The adversary interacts with the model in a black-box manner, without

access to or control over its internal parameters. Also, the adversary can gather text-to-text and text-to-image datasets for training purposes, a reasonable assumption since such datasets are widely available online such as RetrievalQA [3] and Conceptual 12M [6]. However, the adversary lacks knowledge about the specific target prompt  $x$ .

### 3 Methodology

Our approach, as shown in Figure 1, comprises two phases: direct inversion and fine-tuning. Direct inversion is performed using a pre-trained model and a set of collected text-to-text or text-to-image examples. This step aims to generate an initial approximation of the target prompt. Next, fine-tuning is conducted using RL to maximize the target reward to improve the quality of the reconstructed prompt. This step refines the prompt to more closely match the desired response by optimizing the model’s performance based on the rewards.

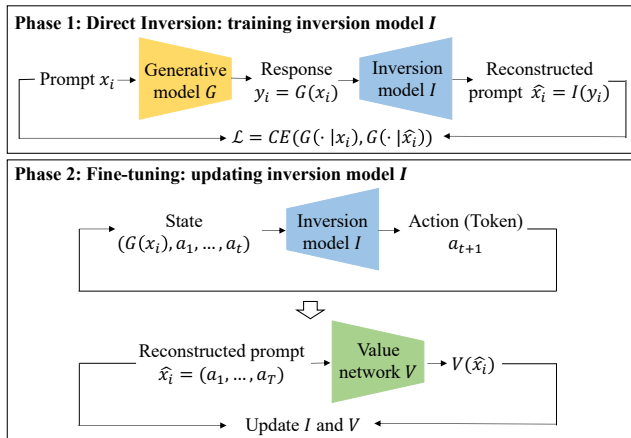


Figure 1: The overview of our approach. It consists of two phases: direct inversion and fine-tuning. In direct inversion, an inversion model is trained by comparing the outputs generated using the target prompt and the reconstructed prompt. In fine-tuning, the quality of the reconstructed prompts is further enhanced through reinforcement learning, which jointly updates the inversion model and a value network.

#### 3.1 Direct Inversion

Given the generation process where a generative model  $G$  maps the target prompt  $x$  to the target response  $y = G(x)$ , the problem of direct inversion is to reconstruct  $x$  from  $y$  using an inversion model  $I$ . This problem can be formalized as searching for  $\hat{x}$  to optimize:

$$\hat{x} = \arg \text{Min}_x d(x, \hat{x} = I(G(x))). \quad (1)$$

However, the inversion model  $I$  cannot be trained by directly optimizing the distance metric  $d()$ , as it is typically non-differentiable and thus unsuitable for use as a loss function,

e.g., BLEU score [30]. To tackle this challenge, we adopt cross-entropy loss which compares the predicted sequence of tokens with the actual sequence, providing a clear measure of error at each step.

Specifically, cross-entropy loss is preferred over general distance metrics for training the inversion model due to its differentiability, which allows for gradient-based optimization essential for effective backpropagation. While general distance metrics are valuable for assessing the similarity between generated and reference outputs, they often involve non-differentiable operations, making them unsuitable as direct loss functions. Cross-entropy loss, on the other hand, measures the difference between predicted and actual probability distributions, providing meaningful gradients that guide the learning process and ensure efficient convergence of the model parameters. Thus, cross-entropy loss facilitates the accurate training of models by using gradient descent, whereas general distance metrics remain primarily evaluation metrics.

In summary, direct inversion involves three distinct steps. **First**, the adversary compiles a dataset  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ , consisting of  $n$  pairs of prompts and responses. **Second**, the adversary inputs  $\{x_1, \dots, x_n\}$  into the target generative model  $G$  and collects the corresponding outputs:  $\{G(x_1), \dots, G(x_n)\}$ . **Third**, the adversary creates a training set by pairing these inputs and outputs,  $\mathcal{D}_{train} = \{(G(x_i), x_i)\}_{i=1}^n$ , to train the inversion model  $I$  using cross-entropy loss. This process involves comparing the differences between  $I(G(x_i))$  and  $x_i$ , training  $I$  to reverse the output of  $G$  back to its original input prompt.

#### 3.2 Fine-tuning

To fine-tune the inversion model  $I$ , we employ reinforcement learning, which is commonly modeled as a Markov decision process represented by the tuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ . Here,  $\mathcal{S}$  denotes the state space,  $\mathcal{A}$  is the action space,  $\mathcal{T}$  represents the transition function defined as  $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ , which describes how the state changes in response to an action,  $\mathcal{R}$  is the reward function defined as  $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ , and  $\gamma$  is the discount factor that modulates the importance of future rewards.

In the context of prompt inversion, a state represents the current observation of the inversion model, and an action corresponds to a token, such as a word, defining the action space as the vocabulary. The reward reflects the quality of a selected action. To reconstruct the target prompt  $x$ , the process begins with the initial state set as the target response  $y$ , denoted as  $s_1 = (y)$ . Subsequently, at each time step  $t$ , an action  $a_t$  (a token) is selected, leading to a deterministic transition to the next state  $s_{t+1} = (y, a_1, \dots, a_t)$ . The inversion process concludes when an end-of-sentence token is selected. The reward  $\mathcal{R}$  is then computed by quantifying the linguistic difference between the original prompt  $x$  and reconstructed prompt  $\hat{x} = s_T - y = (a_1, \dots, a_T)$ , where  $T$  denotes the length of the process. This difference is measured using the BLEU score:  $BLEU(x, \hat{x})$ . The BLEU (Bilingual Evaluation Under-

study) score [30] is used for evaluating the quality of text which has been machine-translated from one natural language to another, with values ranging from 0 to 1. Specifically, the reward is defined as:

$$\mathcal{R} = \begin{cases} R^* \cdot BLEU(x, \hat{x}), & \text{if } BLEU(x, \hat{x}) > 0.5, \\ -R^* \cdot (1 - BLEU(x, \hat{x})), & \text{if } BLEU(x, \hat{x}) \leq 0.5, \end{cases} \quad (2)$$

where  $R^*$  is a pre-defined constant and positive reward signal, which we experimentally set to 1. This reward structure ensures that inverted prompts which are semantically similar to the original ones receive positive rewards, while others receive negative rewards. The rationale behind this setting will be explained in the next section. The adversary’s goal is to reconstruct the target prompt by maximizing the cumulative reward. Note that we do not incorporate a comparison between the re-generated and target responses as a term in Eq. 2. This is because the primary objective of prompt inversion is to reconstruct the original prompt, not to achieve an exact reproduction of the output response. Moreover, even the same prompt can produce highly perceptually different outputs when passed through a generative model, especially in text-to-image scenarios. As a result, using response-level similarity as a reward would introduce instability and hinder the convergence of the inversion model during training.

Specifically, the fine-tuning process involves three steps. **First**, for each sample  $(G(x_i), x_i)$  in  $\mathcal{D}_{train}$ , the adversary tokenizes  $x_i$  into a set of  $m$  tokens:  $x_i = \{x_i^{(1)}, \dots, x_i^{(m)}\}$ , where  $m$  can vary for different instances of  $x_i$ . This tokenization is used for the subsequent computation of BLEU scores in the reward calculation. **Second**, the adversary initializes the state as  $s_1 = (G(x_i))$  and then employs the Proximal Policy Optimization (PPO) algorithm [38] to update the inversion model  $I$  (the policy network  $\pi$ ) and the value network  $V$ . **Third**, the adversary repeats the first and second steps for the other samples in  $\mathcal{D}_{train}$ , iteratively refining the model’s ability to invert the generative model’s outputs back to their original prompts. This fine-tuning process is summarized in Algorithm 1.

In particular, the PPO clipped objective referred to in Line 13 of Algorithm 1 is defined by the equation:

$$\mathcal{L}^{CLIP}(\theta) = \mathbb{E}_t [\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]. \quad (3)$$

Here,  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$  represents the probability ratio, where  $\pi_\theta$  and  $\pi_{\theta_{old}}$  denote the probabilities after and before updating the policy network  $\pi$ , respectively. The advantage estimate  $\hat{A}_t$  is recursively computed as  $\hat{A}_t = \sum_{t'=1}^m (\gamma\lambda)^{t'-1} \delta_{t'}$ , where  $\delta_t = \mathcal{R} + \gamma V(s_{t+1}) - V(s_t)$ . Moreover, the value loss mentioned in Line 14 is defined by:

$$\mathcal{L}_V = (V(\hat{x}_i) - \mathcal{R})^2. \quad (4)$$

The reason to adopt RL is motivated by the limitations of conventional fine-tuning methods in prompt inversion. Supervised fine-tuning approaches, such as instruction tuning, typi-

---

### Algorithm 1 The fine-tuning process of inversion model $I$

---

**Input:** Inversion model  $I$  pre-trained from the first phase, training dataset  $\mathcal{D}_{train} = \{(G(x_i), x_i)\}_{i=1}^n$ ;

**Output:** Fine-tuned inversion model  $I$ ;

- 1: Initialize policy network  $\pi$  and value network  $V$  with parameters  $\theta$  and  $\phi$ , respectively, by copying the parameters from the pre-trained inversion model  $I$ ;
  - 2: **for** each epoch **do**
  - 3:   **for** each sample  $(G(x_i), x_i)$  in  $\mathcal{D}_{train}$  **do**
  - 4:     Tokenize  $x_i$  into a set of  $m$  tokens:  $x_i = x_i^{(1)}, \dots, x_i^{(m)}$ ;
  - 5:     Set initial state  $s_1 = G(x_i)$ ;
  - 6:     **for**  $t = 1$  to  $T$  **do**
  - 7:       Select action  $a_t$  (a token) from the policy network  $\pi(a_t|s_t; \theta)$ ;
  - 8:       Transition to next state  $s_{t+1} = (G(x_i), a_1, \dots, a_t)$ ;
  - 9:     **end for**
  - 10:     Reconstructed prompt  $\hat{x}_i = (a_1, \dots, a_T)$ ;
  - 11:     Compute reward using Eq. 2;
  - 12:     Compute value  $V(\hat{x}_i)$  using the value network;
  - 13:     Update policy net.  $\pi$  using PPO clipped objective (Eq. 3);
  - 14:     Update value network  $V$  to minimize value loss (Eq. 4);
  - 15:   **end for**
  - 16: **end for**
  - 17: Return fine-tuned inversion model  $I$ , i.e., the updated policy network  $\pi$ ;
- 

cally optimize for strict token-level alignment with ground-truth prompts. However, this can lead to overfitting to the syntactic structure of specific prompts and poor generalization to paraphrased or semantically equivalent variants.

In contrast, our RL-based fine-tuning approach optimizes a reward function defined in terms of output consistency, which explicitly captures semantic fidelity. This enables the model to prioritize the underlying meaning of a prompt rather than its exact surface form. Furthermore, RL facilitates token-level exploration during generation, allowing the model to search a broader space of possible prompt reconstructions. This exploration enhances the model’s capacity to identify diverse yet semantically accurate prompts, thereby improving both adaptability and generalizability across linguistic variations in real-world scenarios. In comparison, contrastive learning can be useful for distinguishing between relevant and irrelevant samples, but it does not directly optimize for prompt reconstruction and lacks the fine-grained sample-level reward feedback that RL provides.

## 4 Theoretical Analysis

Our analysis encompasses two aspects: exploring why direct inversion does not perform effectively and examining how fine-tuning significantly enhances the quality of inversion.

## 4.1 Analysis of Direct Inversion

Reconstructing original prompts using direct inversion yields only moderate results, primarily due to information loss in high dimensional spaces. Generative models typically operate in complex, high dimensional spaces with nonlinear transformations, where the forward process (input to output) compresses input data into more abstract and less detailed representations in the output. As a result, inversion models trained directly on these outputs lack sufficient information to accurately reconstruct the original input. For example, given the prompt ‘How many states are there in the USA?’, a generative model might produce a simple output like ‘50.’ Reconstructing the original prompt from this response is challenging because the same output could correspond to many other questions, such as ‘What is the result of 100-50?’.

The second reason is that the outputs of generative models often include noise, filler words, or stylistic variations that do not directly correspond to specific elements of the input. This can easily mislead an inversion model trained to map back to the original input. For example, a single prompt can be used to generate multiple different responses from a generative model. This many-to-one relationship between responses and prompts creates ambiguity, making it hard for the inversion model to reconstruct the original prompt.

## 4.2 Analysis of RL-based Fine-tuning

To facilitate our analysis, we define the value function and the action-value function. The value function  $V_t^\pi : \mathcal{S} \rightarrow \mathbb{R}$ , associated with the reward function  $\mathcal{R}$ , is defined as:

$$V_h^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=h}^T \mathcal{R}_t(s_t, a_t) \mid s_h = s \right]. \quad (5)$$

The value function  $V_t^\pi$  represents the expected reward, accounting for the randomness of the state-action sequence  $\{(s_t, a_t)\}_{t=1}^T$ , where the action  $a_t$  is determined by the policy  $\pi(\cdot \mid s_t)$  at state  $s_t$ , and the subsequent state  $s_{t+1}$  follows the transition function  $\mathcal{T}(\cdot \mid s_t, a_t)$ . Correspondingly, the action-value function  $Q_t^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is defined as:

$$Q_h^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{t=h}^T \mathcal{R}_t(s_t, a_t) \mid s_h = s, a_h = a \right]. \quad (6)$$

Building on the definitions provided in Eqs. 5 and 6, we derive the following Bellman equation:

$$V_h^\pi = \langle Q_h^\pi, \pi_h \rangle_{\mathcal{A}}, \quad (7)$$

where  $\langle \cdot, \cdot \rangle_{\mathcal{A}}$  is the inner product over the action set  $\mathcal{A}$ .

To demonstrate the effectiveness of the RL-based fine-tuning, we compare the performance of a policy  $\pi$  with its updated version  $\pi'$ . The performance difference is quantified using the value functions as:  $V_1^{\pi'}(s_1) - V_1^\pi(s_1)$ , which can be calculated using Lemma 1.

**Lemma 1** (Performance difference [5]). *For any policies  $\pi$  and  $\pi'$ , it holds that*

$$V_1^{\pi'}(s_1) - V_1^\pi(s_1) = \mathbb{E}_{\pi'} \left[ \sum_{h=1}^T \langle Q_h^\pi(s_h, \cdot), \pi'_h(\cdot \mid s_h) - \pi_h(\cdot \mid s_h) \rangle \right]. \quad (8)$$

Lemma 1 suggests that the performance difference is predominantly influenced by the changes in the policy before and after the update. Based on this, we can develop Eq. 8 as:

$$\begin{aligned} V_1^{\pi'}(s_1) - V_1^\pi(s_1) &= \mathbb{E}_{\pi'} \left[ \sum_{h=1}^T \langle Q_h^\pi(s_h, \cdot), \pi'_h(\cdot \mid s_h) - \pi_h(\cdot \mid s_h) \rangle \right] \\ &= \mathbb{E}_{\pi'} \left[ \sum_{h=1}^T \sum_{k=1}^{|\mathcal{A}|} (Q_h^\pi(s_h, a_k) (\pi'_h(a_k \mid s_h) - \pi_h(a_k \mid s_h))) \right]. \end{aligned}$$

To ensure performance improvement, it is crucial to demonstrate that the performance difference  $V_1^{\pi'}(s_1) - V_1^\pi(s_1)$  is positive. To achieve this, we aim for the following conditions: if  $\pi'_h(a_k \mid s_h) - \pi_h(a_k \mid s_h) > 0$ , then  $Q_h^\pi(s_h, a_k)$  should be positive; conversely, if  $\pi'_h(a_k \mid s_h) - \pi_h(a_k \mid s_h) < 0$ , then  $Q_h^\pi(s_h, a_k)$  should be negative. This criterion implies that an increase in the probability of selecting an action should correspond with a positive accumulative reward for that action, and vice versa. Such correlations can be established by appropriately assigning reward values. Specifically, during the fine-tuning process, if the inverted prompt semantically aligns closely with the original prompt, achieving a BLEU score greater than 0.5, it should result in a positive reward. Conversely, if the alignment does not meet this threshold, a negative reward should be applied. This approach corresponds to the reward setup detailed in Eq. 2.

## 5 Experiments

### 5.1 Experimental Setup

**Target Models.** The target models used for prompt inversion attacks are GPT-4 [27] for text-to-text tasks and Stable Diffusion V1.5 [35] for text-to-image tasks. These models were selected because they are widely adopted and the datasets used in our experiments were originally created based on their outputs. Additionally, to evaluate the adaptability of our approach, we include other generative models in a later study, specifically, Claude-3-Haiku [42] for text-to-text tasks and DALL-E 3 [26] for text-to-image generation.

**Inversion Models.** We employ open-source models as the base models for conducting the prompt inversion. Specifically, for attacking text-to-text models, we utilize the T5 model [34] as the base model due to its effectiveness in handling a wide range of NLP tasks. For text-to-image models, we use the BLIP-2 model [15] as the base model, given its state-of-the-art performance in aligning text and image modalities.

**Datasets.** The datasets employed to evaluate attacks on text-to-text models include Alpaca-GPT4 and RetrievalQA.

- Alpaca-GPT4 [8]. It has 52,000 instruction-following responses generated by GPT-4 using Alpaca prompts.
- RetrievalQA [52]. This QA dataset comprises 1,271 questions that explore new world and long-tail knowledge.

The datasets used to evaluate attacks on text-to-image models are based on MS-COCO and Stable-Diffusion-Prompts. However, these datasets cannot be directly utilized in our experiments as they do not contain pre-generated images. To address this, we input the captions or prompts from these datasets into the Stable Diffusion model and collect the corresponding generated images. The generated images are then paired with their respective captions or prompts to construct a training dataset for the inversion model. We have made this training dataset publicly available to support and facilitate further research within the community.

- MS-COCO [17]: It is a resource for object detection, segmentation, key-point detection, and captioning. It comprises 328k images, each with five descriptive captions.
- Stable-Diffusion-Prompts [37]: It comprises 80k prompts sourced from Lexica, utilized to train a prompt generator.

**Evaluation Metrics.** To quantitatively evaluate the results of prompt inversion, we utilize established numerical metrics. Specifically, for text-to-text models, we employ the BLEU score and cosine similarity. For text-to-image models, we use BERTScore, CLIP, and LPIPS. These metrics are widely recognized and used in related research [20, 41, 46].

- BLEU [30]: It is based on comparing sequences of  $n$  words in the machine-generated texts to reference texts.
- Cosine similarity: It is utilized to compare the similarity between the embeddings of the target and reconstructed prompts. Building on prior work [41], we use CLIP’s text encoder to obtain the embeddings of the prompts.
- BERTScore [51]: It is used to compare the similarity between the generated and reference texts at a contextual level. It uses a pre-trained BERT to obtain contextual embeddings for each token in both the generated and reference texts.
- CLIP [33]: We utilize CLIP’s image encoder to derive embeddings for the images, and compare the cosine similarity between the embeddings of the target and generated images.
- LPIPS [50]: It is specifically designed to assess how similar two images appear to a human observer, taking into account not just pixel-wise differences but also high-level features that contribute to human visual perception.

**Baseline Methods.** For our comparative analysis in text-to-text models, we adopt the state-of-the-art method, Output2Prompt [49], as a baseline.

- Output2Prompt [49]. It trains a decoder that takes as input the concatenated embeddings of  $n$  LLM outputs. These outputs are created by a target prompt using an LLM. The decoder processes this input to reconstruct the target prompt.

Additionally, we also employ publicly available tools, GPT-3.5 [29], GPT-4o mini [28], Gemini [10] and Claude [42], as baselines. Specifically, we input a target response,  $y$ , into the public models and request them to generate a potential target prompt,  $\hat{x}$ , treated as the reconstructed prompt.

For text-to-image models, we employ three state-of-the-art methods: PromptStealer, PEZ (hard Prompts made EaZy) and PH2P (Prompting Hard or Hardly Prompting).

- PromptStealer [41]. It has two modules: a subject generator, trained to infer the subject, and a modifier detector, designed to identify the modifiers within the generated image.
- PEZ [46]. It employs gradient-based techniques to optimize hard prompts, consisting of discrete tokens, thereby enabling the automatic discovery and tuning of prompts.
- PH2P [20]. This method optimizes for tokens that align with the vocabulary of the conditional diffusion model, effectively capturing the visual content of the given image.

Additionally, we also utilize a simplified version of our method as a baseline, which incorporates only direct inversion without the fine-tuning component, to assess the significance of fine-tuning. The simplified version is referred to as DI (Direct Inversion), while the full version with fine-tuning is denoted as DI+FT (Direct Inversion with Fine-Tuning).

## 5.2 Overall Results

**Quantitative Study.** Table 1 presents the performance of our DI and DI+FT methods alongside five baselines in text-to-text models on the Alpaca-GPT4 dataset. As shown, our methods significantly outperform the baselines across all three metrics. This superior performance can be attributed to the design of our methods: DI leverages a dedicated inversion model specifically trained to map generated responses back to their original prompts, while DI+FT further enhances this capability by employing reinforcement learning-based fine-tuning. This fine-tuning refines the model’s ability to generate accurate prompts by learning from prompt-response patterns. In comparison, the baselines lack this targeted training, making them less effective than our methods.

Table 2 shows the performance of our DI and DI+FT methods alongside five baselines in text-to-text models on RetrievalQA. Consistent with the results from Alpaca-GPT4, our methods substantially surpass the baselines, demonstrating their superior performance across diverse datasets.

Table 3 displays the performance of our methods compared to three baselines in text-to-image models on the MS-COCO

Table 1: Performance of Various Methods on Alpaca-GPT4

Methods	BLEU ( $\uparrow$ )	Similarity ( $\uparrow$ )	ROUGE ( $\uparrow$ )
DI (ours)	0.589	0.768	0.595
DI+FT (ours)	0.593	0.771	0.596
Output2Prompt	0.256	0.576	0.373
GPT-3.5	0.333	0.667	0.354
GPT-4o mini	0.383	0.687	0.393
Gemini	0.245	0.137	0.056
Claude	0.306	0.655	0.379

Table 2: Performance of Various Methods on RetrievalQA

Methods	BLEU ( $\uparrow$ )	Similarity ( $\uparrow$ )	ROGUE ( $\uparrow$ )
DI (ours)	0.587	0.770	0.561
DI+FT (ours)	0.627	0.773	0.610
Output2Prompt	0.380	0.583	0.335
GPT-3.5	0.378	0.725	0.337
GPT-4o mini	0.445	0.738	0.408
Gemini	0.303	0.160	0.088
Claude	0.441	0.734	0.424

dataset. As shown, our methods outperform the three baselines: PromptStealer, PEZ and PH2P. This superior performance can be attributed to our DI and DI+FT approaches, which are specifically tuned to decode complex prompts by dissecting multi-component requests into pertinent features. This capability makes them exceptionally effective for handling multi-object prompts. In contrast, the prompt interpretation by the baselines is less nuanced, providing our methods with a distinct advantage in scenarios involving prompts with multiple descriptors or layered semantics.

Table 3: Performance of Various Methods on MS-COCO

Methods	BERTScore			CLIP ( $\uparrow$ )	LPIPS ( $\downarrow$ )
	Preci. ( $\uparrow$ )	Recall ( $\uparrow$ )	F1 ( $\uparrow$ )		
DI (ours)	0.951	0.944	0.948	0.831	0.612
DI+FT (ours)	0.955	0.948	0.951	0.842	0.609
PromptStealer	0.816	0.914	0.862	0.776	0.616
PEZ	0.766	0.842	0.802	0.709	0.633
PH2P	0.816	0.845	0.830	0.774	0.628

Table 4: Performance of Various Methods on Stable-Diffusion-Prompts

Methods	BERTScore			CLIP ( $\uparrow$ )	LPIPS ( $\downarrow$ )
	Preci. ( $\uparrow$ )	Recall ( $\uparrow$ )	F1 ( $\uparrow$ )		
DI (ours)	0.892	0.868	0.869	0.808	0.598
DI+FT (ours)	0.927	0.938	0.932	0.811	0.592
PromptStealer	0.843	0.863	0.853	0.806	0.600
PEZ	0.777	0.807	0.792	0.689	0.638
PH2P	0.813	0.801	0.807	0.728	0.635

Table 4 illustrates the performance of our methods compared to three baselines in text-to-image models on the Stable-Diffusion-Prompts dataset. Once again, our methods surpass the baselines, demonstrating their robust adaptability across various datasets. This consistent performance advantage is attributed to our methods' focus on generalization, which allows them to effectively adapt to diverse datasets. Specifi-

cally, our DI+FT method employs reinforcement learning to fine-tune the inversion model, significantly enhancing its capability to adjust to the unique characteristics of each dataset. This targeted fine-tuning optimizes the model's understanding of distinct prompt structures within each dataset.

Note that although the quantitative difference between DI+FT and DI alone may appear small, the qualitative difference is substantial, as will be demonstrated later. This discrepancy arises because quantitative evaluations are typically based on semantic similarity metrics. As a result, two prompts with similar overall meaning can yield nearly identical quantitative scores. However, even a slight variation in keywords between prompts can lead to significantly different information being conveyed (e.g., Table 5) or produce markedly different outputs (e.g., Figure 9(a)). This further highlights why our approach prioritizes reconstructing the exact prompts rather than settling for approximate equivalents.

**Qualitative Study.** Table 5 shows examples of prompt inversion attacks using our methods compared to baselines in text-to-text models on the Alpaca-GPT4 dataset. The reconstructed inquiry prompts from our methods, particularly those refined through RL-based fine-tuning, demonstrate a markedly better alignment with the original inquiry prompts than the baselines. This superior performance is attributed to the adaptive learning mechanisms of RL, which optimize prompt reconstruction by iteratively adjusting and improving based on feedback from each prompt's success rate. This process enables our models to more accurately capture and replicate the nuanced intent and context of the original prompts.

Table 6 presents examples of prompt inversion attacks using our methods and the baselines in text-to-text models on the RetrievalQA dataset. The results follow a similar trend to those observed on the Alpaca-GPT4 dataset, with our methods significantly outperforming the baselines. Notably, our RL-based fine-tuning mechanism enables our method to reconstruct nearly every word of the original question, demonstrating its enhanced accuracy in prompt recovery.

Figure 2 illustrates examples of attacks conducted using our methods and the baselines in text-to-image models on the MS-COCO dataset. These examples highlight the effectiveness of our methods in reconstructing a significant portion of the keywords from the prompts, and in some cases, even fully reconstructing the prompts. This capability results in high-quality generative images that closely align with the original prompts. In contrast, the baseline methods struggle to achieve comparable results. While some baselines manage to reconstruct a few keywords, methods like PEZ and PH2H demonstrate significant limitations, often failing to reconstruct any meaningful keywords. As a consequence, the generated images from these baselines are of notably low quality and lack alignment with the original prompts.

Figure 3 displays examples of attacks executed using our methods and the baselines in text-to-image models on Stable-Diffusion-Prompts. The results align with the trends observed

Table 5: Examples of Prompt Inversion Using Various Methods on Alpaca-GPT4

Inquiry	Response	Reconstructed inquiry						
		DI (ours)	DI+FL (ours)	Output2Prompt	GPT-3.5	GPT-4o mini	Gemini	Claude
Generate a sentence describing the effect of advertising on the 2008 election cycle	The extensive and persuasive advertising campaigns during the 2008 election cycle played a significant role .....	Construct a sentence using the words "advocacy" and "cons"	Generate a sentence that describes the role of advertising in the 2008 election cycle	Identify the significant impact of the advertising campaigns during the 2008 election cycle.	What impact did advertising campaigns have on the 2008 election?	What impact did advertising campaigns have on public opinion during the 2008 election cycle?	To what extent did advertising influence public opinion and the election outcome in 2008?	The impact of advertising campaigns on the 2008 election outcome.
Edit this sentence to make it more effective: "He was always helpful."	He consistently demonstrated a helpful nature.	Edit the following sentence to make it more concise: "He always showed a nice manner."	Edit the following sentence to make it more concise: He was always very helpful.	Identify the qualities of a given person.	Did he always show a helpful nature?	Can you provide examples of how he demonstrated a helpful nature?	Is the candidate's helpful nature demonstrated through specific actions or examples?	Demonstrate helpful nature consistently

Table 6: Examples of Prompt Inversion Using Various Methods on RetrievalQA

Question	Answer	Reconstructed question						
		DI (ours)	DI+FL (ours)	Output2Prompt	GPT-3.5	GPT-4o mini	Gemini	Claude
Does an individual share of a stock have some kind of unique identifier?	Yes, an individual share of stock has a unique identifier, which is known as a ticker symbol .....	Does an individual share of stock have a ticker symbol?	Does an individual share of stock have a unique identifier?	Identify the unique identifier of an individual share of stock.	What is a ticker symbol for a share of stock?	What is a ticker symbol?	What is a ticker symbol?	What is a ticker symbol and how is it used to identify a company's stock?
How can I get the most value from my employer's ESPP?	Employee stock purchase plans (ESPPs) are a way for employees to purchase company stock .....	How to get most value from employer stock purchase plans?	How to get the most value from your employer's ESPP?	What is the most value from your employer's ESPP?	What are some tips for getting the most value from your employer's ESPP?	What are tips for maximizing an ESPP?	How can employees maximize benefits from ESPPs?	What are the key terms and conditions of the employee stock purchase plan (ESPP)?

on MS-COCO, as illustrated in Figure 2. Our methods consistently outperform the baselines, demonstrating superior capability in reconstructing prompts and generating images that closely match the target styles. In particular, the baseline methods, such as PEZ and PH2H, exhibit significant limitations. These methods fail to accurately invert the keywords in the prompts, leading to the generation of images with styles that deviate from the target images. This failure highlights their inability to effectively capture the semantic essence of the original prompts. In contrast, our methods excel in capturing and reconstructing the core keywords, resulting in high-quality images that align closely with the intended prompts.

The experimental results demonstrate the superior performance of our approach compared to existing prompt inversion methods across both text-to-text and text-to-image modalities. Notably, our general RL-based approach consistently outperforms methods that are specifically tailored to a single modality. This performance advantage stems from a key design principle: existing methods often rely heavily on the structural or statistical characteristics unique to a particular modality (e.g., token patterns in text or pixel distributions in images), which limits their flexibility and generalizability. In contrast, our approach decouples the inversion process from any specific modality by treating the generative model as a black box and

optimizing the reconstructed prompts based on output consistency, regardless of the output type. The use of RL enables the model to adaptively explore and fine-tune prompt candidates using reward signals defined over the model outputs, whether text or image, thus capturing semantic alignment without overfitting to modality-specific cues.

### 5.3 Adaptability Study

Our adaptability study evaluates the effectiveness of our methods in three settings: the text-to-code context, prompts with increased complexity, and additional generative models.

**Text-to-Code.** We evaluate our methods in the text-to-code context by selecting prompt-code pairs from the Alpaca-GPT4 dataset. Specifically, we assessed the capability of our methods to reconstruct the target prompts from the corresponding generated code snippets. The results are presented in Figure 4. It can be observed that both of our methods are capable of successfully reconstructing the target prompts, with DI+FT achieving higher accuracy than DI in several cases. This is because while both methods can capture the general functionality of the code, DI+FT benefits from the RL training phase, which allows it to better align the reconstructed prompts with the original intent.



Figure 2: Examples of Prompt Inversion Using Various Methods on MS-COCO

**Complex Prompts.** We select a set of complex prompts from the Alpaca-GPT4 dataset, each containing approximately 40 words or including complex web links, and attempted to reconstruct them based on their corresponding responses. The results are presented in Table 7. It can be observed that both of our methods are capable of semantically reconstructing these complex prompts. Notably, prompts containing web links were often reconstructed with high accuracy, demonstrating the strong adaptability of our methods in handling intricate input scenarios.

**Against Alternative Generative Models.** We adopt Claude-3-Haiku for text-to-text and DALL-E 3 for text-to-image to evaluate the generalizability of our approach. Specifically, for the text-to-text setting, we input the prompts from the Alpaca-GPT4 dataset into Claude-3-Haiku and collect the corresponding outputs. Similarly, for the text-to-image setting, we feed the prompts from the Stable-Diffusion-Prompts dataset into DALL-E 3 and obtain the generated images. We apply our trained inversion models to reconstruct the original prompts based solely on these new outputs. The results are presented in Table 8 and Figure 5. As shown, our method successfully reconstructs nearly all core keywords from both text and image outputs, showing strong generalizability. These findings indicate that our approach remains effective even when attacking generative models it was not specifically trained on,

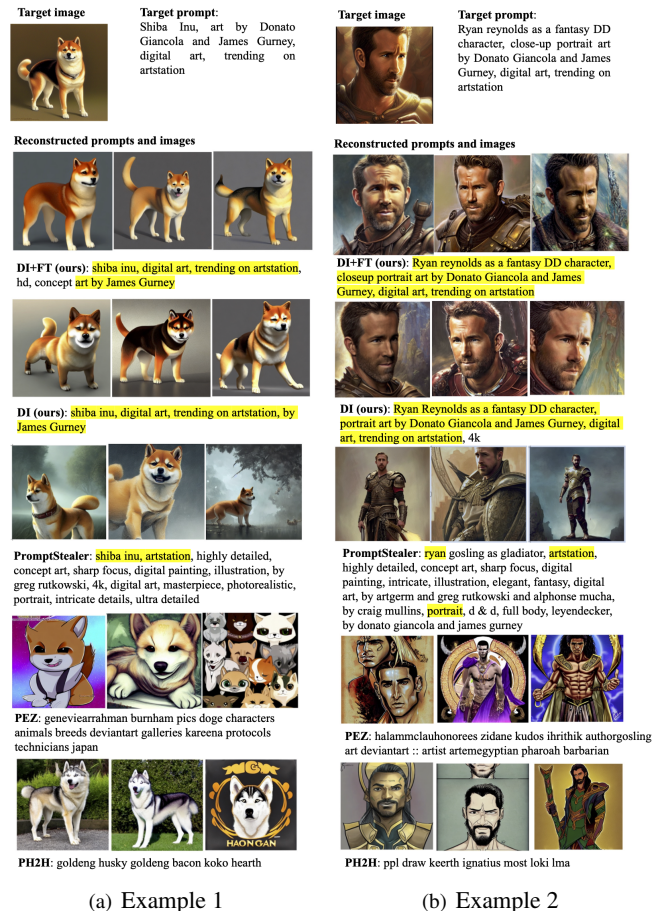


Figure 3: Examples of Prompt Inversion Using Various Methods on Stable-diffusion-prompts

highlighting its adaptability in cross-model scenarios.

## 5.4 Ablation Study

Given that fine-tuning is a crucial component of our method, our ablation study focuses on two key aspects: exploring additional fine-tuning techniques and analyzing the impact of varying the reward functions.

**Additional Fine-tuning Approaches.** In our method, we employ the PPO algorithm as the RL technique for fine-tuning, achieving remarkable results. To evaluate whether alternative RL algorithms might produce different outcomes, we explore the use of another popular RL algorithm, A2C (Advantage Actor-Critic) [22]. A2C operates on the actor-critic framework, where the ‘actor’ selects actions based on a learned policy, and the ‘critic’ evaluates these actions by estimating their value. Unlike asynchronous methods, A2C simplifies training by synchronizing multiple workers to compute gradients collectively, ensuring stable and efficient learning.

The quantitative comparison between PPO and A2C is summarized in Tables 9, 10, 11 and 12. The results indicate that both RL algorithms achieve comparable performance

```

Target prompt: Write a code to detect the type of browser that is used to access the page.
Target response:
<code>
function detectBrowser() {
  var userAgent = navigator.userAgent;
  var browserName = "";

  if (userAgent.indexOf("Firefox") > -1) {
    browserName = "Firefox";
  } else if (userAgent.indexOf("SamsungBrowser") > -1) {
    // Samsung Browser
    browserName = "Samsung Internet";
  } else if (userAgent.indexOf("Opera") > -1 ||
  userAgent.indexOf("OPR") > -1) {
    // Opera S, or
    browserName = "Opera";
  } else if (userAgent.indexOf("Edg") > -1) {
    // Edge (Chromium)
    browserName = "Microsoft Edge";
  } else if (userAgent.indexOf("Chrome") > -1) {
    browserName = "Chrome";
  } else if (userAgent.indexOf("Safari") > -1) {
    browserName = "Safari";
  } else if (userAgent.indexOf("Trident") > -1) {
    // Internet Explorer
    browserName = "Internet Explorer";
  } else {
    browserName = "Unknown Browser";
  }

  return browserName;
}
</code>

This code uses 'navigator.userAgent' to determine the user's browser based on the User-Agent string. It checks various patterns of common browsers like Firefox, Chrome, Safari, Opera, Edge and then returns their respective names. If none matches, it returns "Unknown Browser".

Reconstructed prompt (DI+FT): Write a code to detect the type of browser that is used to access the page
Reconstructed prompt (DI): Write a JavaScript code to detect the type of browser that is used to access the page

Target prompt: Compose a C# program to calculate the compound interest rate of an investment.
Target response:
<code>
using System;

class Program
{
  static void Main(string[] args)
  {
    // Declare variables
    double principal, interestRate, years, compoundInterest;

    // Get user input
    Console.WriteLine("Enter the principal amount: ");
    principal = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Enter the interest rate (in decimal form): ");
    interestRate = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Enter the number of years: ");
    years = Convert.ToDouble(Console.ReadLine());

    // Calculate compound interest
    compoundInterest = principal * (Math.Pow(1 + interestRate,
    years));

    // Output result
    Console.WriteLine("The compound interest is: " +
    compoundInterest.ToString("C2"));
  }
}

This program prompts the user to enter the principal amount, interest rate, and number of years for the investment. It then calculates the compound interest using the formula 'A = P * (1 + r)^t' where 'A' is the compound interest, 'P' is the principal amount, 'r' is the interest rate, and 't' is the number of years. The calculated result is displayed to the user in currency format.

Reconstructed prompt (DI+FT): Create a C# program to calculate the compound interest rate of an investment
Reconstructed prompt (DI): Write a program to calculate the compound interest rate of an investment
</code>

```

Figure 4: Examples of Prompt Inversion from Code Snippets Using Our Methods

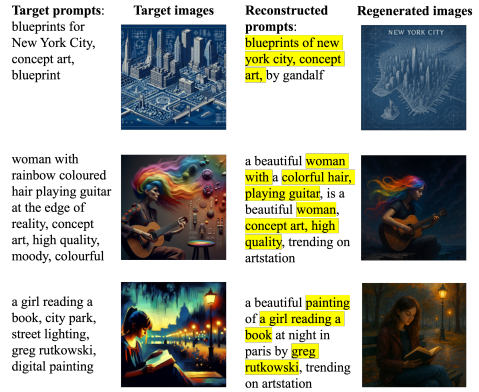


Figure 5: Examples of Prompt Inversion against DALL-E 3 Using Our Approach

Table 7: Examples of Prompt Inversion on Complex Prompts Using Our Methods

Prompt	Response	Reconstructed prompt	
		DI	DI+FT
Can you alter someone's dream by touching or talking to them while they're dreaming? And if the answer is yes, is it possible to find out how the dream can explain like I'm five.	It is not possible to alter someone's dream by touching or talking to them while they are dreaming. Dreams are a natural part of the sleep process and are created by the brain...	Is it possible to alter someone's dream by touching or talking to him while they are dreaming? You know, is it possible to have a dream with someone, where he is dreaming? Please explain like I'm five.	Can you alter someone's dream by touching or talking to them while they're dreaming? Can you change a dream for someone where he has a similar dream experience? Please explain like I'm five.
Summarize the article about the use of drones for delivery. <a href="https://www.blog-google/outreach-initiatives/google-drones-will-bring-small-internet-access-worldwide/">https://www.blog-google/outreach-initiatives/google-drones-will-bring-small-internet-access-worldwide/</a>	In this article, it is discussed how drones are being used to deliver small packages and provide internet access worldwide. According to the article, ...	Summarize the article about the use of drones for delivery. <a href="https://www.blog-google/outreach-initiatives/google-org/drones-will-bring-small-internet-access-worldwide/">https://www.blog-google/outreach-initiatives/google-org/drones-will-bring-small-internet-access-worldwide/</a>	Summarize the article about the use of drones for delivery. <a href="https://www.blog-google/outreach-initiatives/google-org/drones-will-bring-small-internet-access-worldwide/">https://www.blog-google/outreach-initiatives/google-org/drones-will-bring-small-internet-access-worldwide/</a>

in prompt inversion tasks. This similarity can be attributed to their shared foundational architecture as policy gradient methods, which optimize the policy directly to maximize cumulative rewards. Both algorithms effectively leverage the actor-critic framework, where the actor proposes actions, and the critic evaluates their quality, allowing for efficient policy updates. Additionally, the tasks of prompt inversion are relatively straightforward in terms of state-action mapping, which minimizes the potential for significant differences in learning dynamics between PPO and A2C.

Table 8: Examples of Prompt Inversion against Claude-3-Haiku Using Our Approach

Prompt	Response	Reconstructed prompt
Rewrite the following sentence to make it clear and concise: At this point in time, we don't have the necessary funds to purchase the equipment that would be required to carry out the project.	Here is a revised version of the sentence that is more clear and concise: We currently lack the funds needed to purchase the equipment required for the project.	Given the following sentence, provide a more concise version. We currently don't have the funds to purchase the equipment for the project.
Sources predict that the stock price for Company A will either stay the same or go up for the next 10 trading days. Pick a strategy for if the price will go up or stay the same.	Here is a potential strategy for investing in Company A's stock if the sources predict the price will either stay the same or go up over the next 10 trading days: ...	Create a strategy for if the stock price for Company A is predicted to either stay the same or go up over the next 10 trading days.
Write a JavaScript function that takes in a string and returns an object with each character of the string as a key, and the number of times that character appears in the string as the value.	Here's a JavaScript function that takes in a string and returns an object with each character of the string as a key, and the number of times that character appears in the string as the value: ...	Write a JavaScript function that takes in a string and returns an object with each character of the string as a key; the number of times that character appears in the string as the value.

Table 9: Performance of Various Fine-tuning Approaches on Alpaca-GPT4

Methods	BLEU (↑)	Similarity (↑)	ROUGE (↑)
DI+FT (PPO)	0.593	0.771	0.596
DI+FT (A2C)	0.588	0.767	0.591

Table 10: Performance of Various Fine-tuning Approaches on RetrievalQA

Methods	BLEU (↑)	Similarity (↑)	ROUGE (↑)
DI+FT (PPO)	0.627	0.773	0.610
DI+FT (A2C)	0.588	0.770	0.562

Table 11: Performance of Various Fine-tuning Approaches on MS-COCO

Methods	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
DI+FT (PPO)	0.974	0.967	0.970	0.843	0.610
DI+FT (A2C)	0.972	0.965	0.968	0.842	0.605

Table 12: Performance of Various Fine-tuning Approaches on Stable-diffusion-prompts

Methods	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
DI+FT (PPO)	0.917	0.900	0.906	0.815	0.595
DI+FT (A2C)	0.912	0.897	0.906	0.810	0.594

The qualitative comparison between PPO and A2C is presented in Tables 13, 14 and Figures 6, 7. The qualitative results align with the quantitative findings, indicating comparable performance between the two algorithms. This similarity arises because both algorithms effectively extract the keywords from the prompts, leading to similar generative outputs when using the reconstructed prompts.

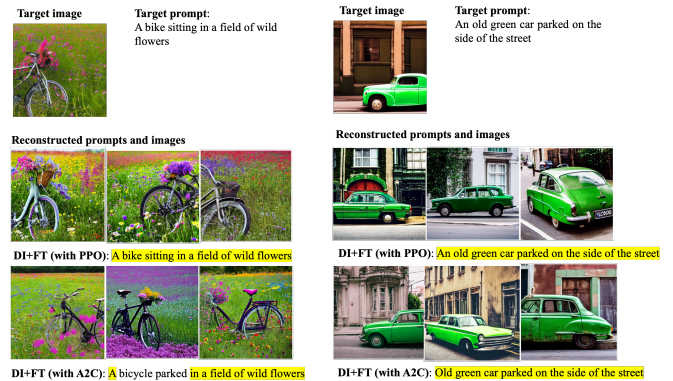
Table 13: Examples of Prompt Inversion with Various Fine-tuning Approaches on Alpaca-GPT4

Inquiry	Response	Reconstructed inquiry	
		DI+FT (PPO)	DI+FT (A2C)
Compare the speed of light and sound	Light travels at a speed of approximately 299,792 kilometers per second (km/s) ....., sound travels with a speed of approximately 343 meters per second (m/s), .....	Compare the speed of light to sound	How does light travel?
Diagnose the issue with this computer: It turns on but won't display anything on-screen.	There could be several potential reasons why the computer turns on but doesn't display anything onscreen. Here are some troubleshooting steps that you can try to diagnose the issue: .....	Diagnose the issue with this computer. The computer turns on but doesn't display anything on-screen.	Diagnose the issue with this computer. The computer turns on but doesn't display anything on-screen.

**Variation in Reward Functions.** During the RL-based fine-tuning, we utilize the reward function defined in Eq. 2. To further investigate the impact of different reward definitions, we evaluate alternative metrics by replacing the BLEU score in Eq. 2 with ROUGE and cosine similarity.

Table 14: Examples of Prompt Inversion with Various Fine-tuning Approaches on RetrievalQA

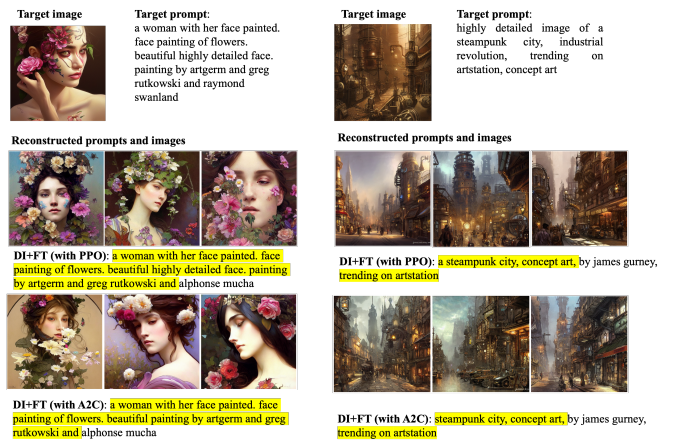
Inquiry	Response	Reconstructed inquiry	
		DI+FT (PPO)	DI+FT (A2C)
Please explain what is "Probability theory"	Probability theory is a branch of mathematics that deals with the study of random events and the likelihood of their occurrence. It is used to model and analyze uncertain situations .....	Please explain what is "Probability theory"	How does Please explain what is "Faculty theory"
How do forensic auditors examine financial reporting	Forensic auditors are professionals who specialize in investigating financial irregularities, such as fraud or embezzlement .....	How do forensic auditors examine financial reporting	What do forensics do



(a) Example 1

(b) Example 2

Figure 6: Examples of Various Fine-tuning Approaches on the MS-COCO dataset



(a) Example 1

(b) Example 2

Figure 7: Examples of Various Fine-tuning Approaches on the Stable-diffusion-prompts dataset

The quantitative comparison among the three metrics is summarized in Tables 15, 16, 17, and 18. The results consistently show that our method performs best when using BLEU in Eq. 2, compared to ROUGE and cosine similarity. This performance discrepancy can be attributed to fundamental differences in how these metrics guide the learning process. BLEU is specifically designed to balance semantic similarity and penalize dissimilar outputs, providing a nuanced signal that better aligns with the objective of accurate prompt reconstruction. In contrast, ROUGE primarily emphasizes surface-level overlap between the original and reconstructed prompts, which may overlook deeper semantic connections or nuanced variations. This concentration on token-level precision and recall makes ROUGE less effective in capturing the broader contextual alignment needed for high-quality prompt inversion. Cosine similarity, on the other hand, measures vector-level similarity between sentence embeddings, and tends to overlook subtle but important differences in word choice and phrasing. As a result, it may guide the inversion model toward semantically similar but structurally less faithful reconstructions, reducing overall accuracy.

Table 15: Performance of Various Reward Functions on Alpaca-GPT4

Methods	BLEU (↑)	Similarity (↑)	ROUGE (↑)
DI+FT (BLEU)	0.599	0.768	0.599
DI+FT (ROUGE)	0.578	0.763	0.580
DI+FT (Cosine)	0.571	0.765	0.591

Table 16: Performance of Various Reward Functions on RetrievalQA

Methods	BLEU (↑)	Similarity (↑)	ROUGE (↑)
DI+FT (BLEU)	0.627	0.773	0.610
DI+FT (ROUGE)	0.587	0.760	0.563
DI+FT (Cosine)	0.535	0.770	0.542

Table 17: Performance of Various Reward Functions on MSCOCO

Methods	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
DI+FT (BLEU)	0.974	0.967	0.971	0.844	0.610
DI+FT (ROUGE)	0.962	0.953	0.958	0.840	0.582
DI+FT (Cosine)	0.649	0.553	0.595	0.842	0.610

Table 18: Performance of Various Reward Functions on Stable-diffusion-prompts

Methods	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
DI+FT (BLEU)	0.922	0.921	0.918	0.816	0.596
DI+FT (ROUGE)	0.912	0.888	0.894	0.802	0.567
DI+FT (Cosine)	0.727	0.631	0.674	0.788	0.606

The qualitative comparison among the three metrics is presented in Tables 19, 20 and Figures 8, 9. The results reveal

that using ROUGE and cosine similarity often fails to reconstruct critical keywords in certain scenarios. For instance, in the second example of Table 19, the BLEU-based reward function successfully recovers the core keywords ‘news article’, whereas the ROUGE- and cosine similarity-based reward functions incorrectly generate ‘movie synopsis’, which significantly diverges from the original prompt’s intended semantics. This inability to accurately reconstruct essential keywords also degrades the quality of the generated outputs. For instance, in the first example of Figure 9, the artist name ‘Greg Rutkowski’ is omitted when using ROUGE and cosine similarity, resulting in images that differ substantially in both style and content from the target. In contrast, using BLEU in the reward function leads to more faithful keyword reconstruction and semantic alignment, producing higher-quality outputs that better reflect the original prompt’s intent.

Table 19: Examples of Prompt Inversion with Various Reward Functions on Alpaca-GPT4

Inquiry	Response	Reconstructed inquiry		
		DI+FT (BLEU)	DI+FT (ROUGE)	DI+FT (Cosine)
Generate a blog post on the topic of “How to Study for an Online Course”	In the current era, learning has become more accessible than ever before, thanks to the availability of online courses. Whether you are a college student, a full-time employee .....	Create a blog post about how to study for an online course	Write a persuasive essay on how to study for an online course	Write a persuasive essay on how to study for an online course.
Generate a variety of titles for a news article about the new coronavirus vaccine	1. Vaccination Nation: The Promising Arrival of the Coronavirus Vaccine; 2. A Shot of Hope: The Launch of the COVID-19 Vaccine .....	Generate a list of possible titles for a news article about Coronavirus vaccine	Generate a list of possible titles for the following movie synopsis. The Coronavirus Vaccine	Generate a list of possible titles for the following movie synopsis. The Coronavirus Vaccine

Table 20: Examples of Prompt Inversion with Various Reward Functions on RetrievalQA

Inquiry	Response	Reconstructed inquiry		
		DI+FT (BLEU)	DI+FT (ROUGE)	DI+FT (Cosine)
Please explain what is “Microsoft Translator”	Microsoft Translator is a cloud-based machine translation service developed by Microsoft .....	Please explain what is “Microsoft Translator”	Please explain what is “Michael Translator”	Please explain what is “Michael Translator”
What is the “point” (purpose) of the S&P 500?	The S&P 500, also known as the Standard & Poor’s 500, is a stock market index that measures the stock performance of 500 large-cap publicly traded companies .....	What is the purpose of the S&P 500?	S&P calculation	The purpose of the S&P 500



Figure 8: Examples of Various Reward Functions on MS-COCO

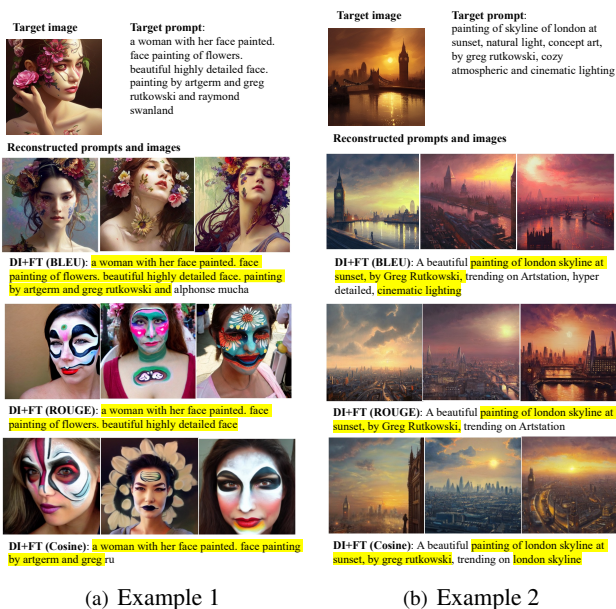


Figure 9: Examples of Various Reward Functions on Stable-diffusion-prompts

## 5.5 Open-world Study

In addition to lab-based experiments, we have extended our evaluations to open-world scenarios to assess the performance of these methods under real-world conditions. Specifically, for text-to-text models, we utilized a medical dataset [16], where a patient describes their symptoms and an LLM-based doctor provides a potential diagnosis. The objective of prompt

inversion in this context is to deduce the patient’s detailed symptoms from the doctor’s diagnosis. Furthermore, for text-to-image models, we used images from two prompt market-places: PromptBase [31] and PromptDB [32].

The results of the text-to-text prompt inversion are detailed in Table 21. Our methods demonstrate a remarkable ability to reconstruct almost all the key words from the original patient descriptions, with significant improvements observed following fine-tuning. In contrast, the baseline methods show substantially poorer performance. This difference highlights our methods’ advanced capabilities in accurately capturing and replicating the nuances and specific details of the input prompts, making them particularly effective in real-world scenarios where precision is critical.

We have also selected a set of sensitive prompts from the medical dataset, each containing private information about patients’ medical conditions, and attempted to reconstruct them based on the corresponding doctors’ responses. The results are presented in Table 22. It can be observed that both of our methods are capable of successfully reconstructing the sensitive prompts. In particular, DI+FT achieves higher accuracy than DI, as DI occasionally introduces extraneous words that are not present in the original prompts.

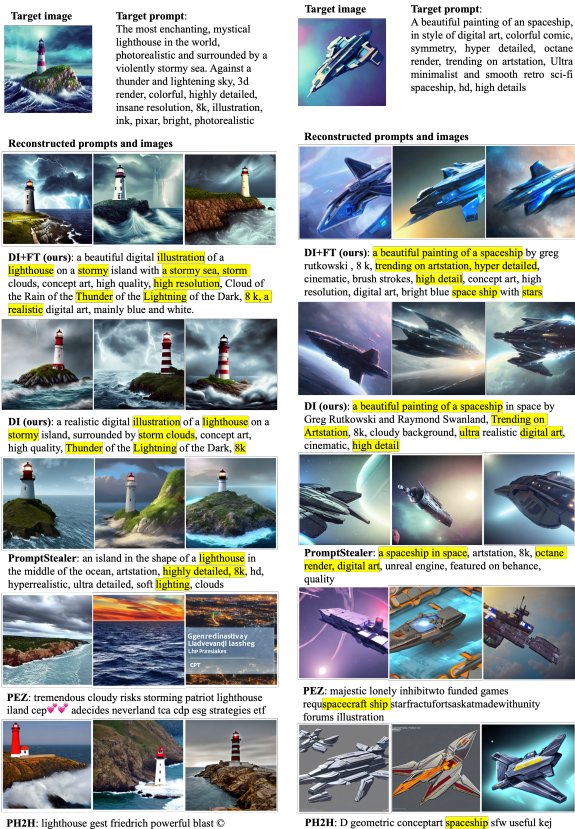
Table 22: Performance of Our Methods in Reconstructing Sensitive Prompts

Patients’ medical conditions	Doctors’ responses	Reconstructed medical conditions	
		DI	DI+FT
Hi Doctor, I have been diagnosed with Osteomyelitis. What medications do I need to take for it?	You will need to take Vancomycin, Zosyn, Daptomycin, Cefepime, and Tetracaine (One Touch).	Doctor, I have been diagnosed with Osteomyelitis. What medications do I need to take?	Doctor, I have been diagnosed with Osteomyelitis. What medications do I need to take?
Doctor, I have been experiencing sharp abdominal pain and coughing up blood lately. What could be the problem?	Based on your symptoms, it’s possible that you have acute pancreatitis, ... We need to run some tests to confirm the diagnosis and start appropriate treatment.	Doctor, I’ve been experiencing sharp abdominal pain and back pain for the past few days. I also cough up blood and have been coughing up blood. What could be wrong with me?	Doctor, I have been experiencing sharp abdominal pain, and coughing up blood. What could be wrong with me?

Figure 10 illustrates the results of our text-to-image prompt inversion. The prompts reconstructed using our method include significantly more key words than those by the baseline methods, resulting in images that are much more aligned with the target images. This substantial superiority of our method stems from its advanced algorithms that enhance the accuracy of key word detection and image relevancy, ensuring that the reconstructed images closely resemble the intended targets.

Table 21: Performance of Our Methods and Baselines in Real-world Text-to-Text Prompt Inversion

Patient's description	Doctor's diagnosis	Reconstructed description						
		DI (ours)	DI+FL (ours)	Output2Prompt	GPT-3.5	GPT-4o mini	Gemini	Claude
Doctor, I have an open wound on my shoulder and it's not healing properly. What tests do I need?	You will need radiographic imaging of your shoulder, including a plain x-ray to see the extent of the injury. We may also need to suture the wound .....	Doctor, I have an open wound on my shoulder that hurts a lot. What tests do I need to take?	Doctor, I have an open wound on my shoulder and it's not healing properly. What tests do I need to take?	Describe the treatment needed for the kidney function of the patient.	Doctor, I injured my shoulder and have a deep wound that won't stop bleeding. What tests and treatments do you recommend to assess?	Doctor, I've injured my shoulder and there's a wound. What tests and treatments will I need?	Following an examination, what tests or treatment are needed because of my shoulder injury?	Doctor, I was involved in an accident and injured my shoulder. Can you tell me what kind of medical tests and treatments I might need to address this injury?
Doctor, I have been experiencing arm swelling, hand stiffness, wrist swelling, and wrist pain. What could be wrong with me?	Based on your symptoms, it sounds like you may have a fracture in your hand.	Doctor, I have been experiencing hand swelling, hand stiffness or tightness, hand or finger swelling, and wrist swelling. What could be wrong with me?	Doctor, I have been experiencing wrist swelling, hand stiffness or tightness, and wrist pain. What could be wrong with me?	Identify the symptoms and indicate the severity of the injury.	I've been experiencing sharp pain and swelling in my hand, especially when I try to move my fingers. What do you think might be?	Doctor, I've been experiencing a lot of pain and swelling in my hand after a recent injury. Could this be something serious like a fracture?	Am I experiencing a fracture in my hand?	Doctor, I fell and hit my hand pretty hard. It's been swelling up and is really painful to move. Could this be a fracture?



(a) Example 1

(b) Example 2

Figure 10: Examples of prompt inversion using various methods in real-world text-to-image scenarios

## 5.6 Extension to the Text-to-Video Modality

We now extend our experiments to the text-to-video modality. For this setting, we adopt LongVU [40] as the inversion model

and use the VidProM dataset [45] for training. The video generation model employed is CogVideo [12], which takes text prompts as input and generates corresponding video outputs. We chose CogVideo because it was also used by the creators of the VidProM dataset to generate a portion of the videos included in our training set. To ensure a fair comparison when evaluating the quality of re-generated videos, we use the same model in our experiments. Moreover, To measure the similarity between two videos, we adopt the xclip-base-patch32 model [25] to extract embeddings for each video and compute the cosine similarity between the embeddings of the original and reconstructed videos.

The quantitative results are presented in Table 23, and the qualitative visual results are shown in Figure 11. Note as it is not feasible to display full video outputs in the paper, we instead select and present three consecutive representative frames from each generated video. The complete generated videos have been uploaded to our Zenodo repository for interested readers. As shown in the table, DI+FT outperforms DI across multiple evaluation metrics. This improvement is also evident in the visual results, where DI+FT produces more accurate prompt reconstructions than DI, leading to generated videos that are more closely aligned with the original target videos. These results demonstrate that our methods are not only effective in text and image contexts but also extend successfully to the more complex video modality.

Table 23: Performance of Our Methods on VideoProM

Methods	BERTScore			Video similarity (↑)
	Preci. (↑)	Recall (↑)	F1 (↑)	
DI	0.918	0.878	0.897	0.628
DI+FT	0.919	0.911	0.915	0.659

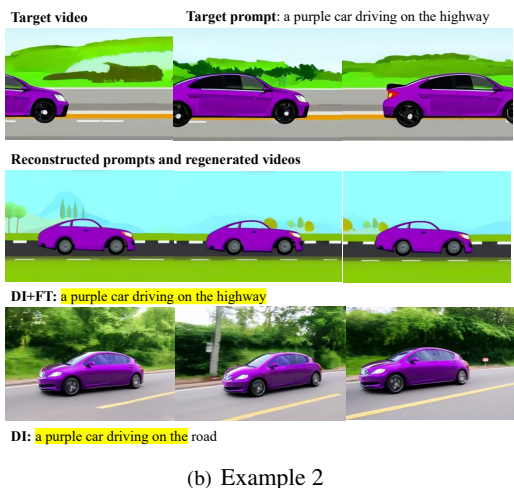
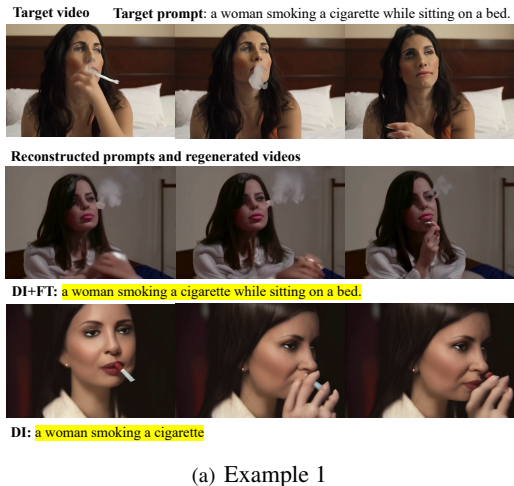


Figure 11: Examples of Prompt Inversion Using Our Methods on VideoProM

## 6 Defense

We propose two defense mechanisms: output perturbation, which alters the model’s responses to hinder the inversion model’s ability to accurately reconstruct prompts, and query constraint, which limits access to the generative model, thereby restricting the data available to train an effective inversion model.

**Output Perturbation.** Output perturbation introduces distortions into the generation model’s output, ensuring the perturbed output remains understandable to humans but cannot be effectively used for prompt inversion. Specifically, for text-to-text models, we paraphrase the output using synonyms to maintain semantics while preventing inversion. For example, a sentence ‘The quick brown fox jumps over the lazy dog’ might be altered to ‘The swift auburn fox leaps over the sluggish dog.’ This subtle change preserves the meaning but complicates direct inversion. For text-to-image models,

we add Gaussian noise to the output images to ensure they remain perceptually meaningful to humans yet difficult to use for reconstructing the original prompts.

The defense results are summarized in Table 24 and Figure 12. In the first example of Table 24, when synonyms are used to replace corresponding words in the original answer without altering its structure, our method successfully reconstructs the prompt (question), preserving the key elements of the original question. However, when the original answer is thoroughly rephrased, as shown in the second example of Table 24, the reconstructed prompt loses its core keywords. A similar trend is observed in text-to-image models. In both examples from Figure 12, adding a small amount of noise to the target images (variance less than 60) allows our method to extract all the key words from the target prompt and generate highly aligned images. Conversely, when the noise level is significantly increased (variance greater than 80), the reconstructed prompt fails to align with the target prompt, resulting in low-quality images. These findings demonstrate the robustness of our method, particularly under conditions of minor perturbations.

**Query Constraint.** Query constraint seeks to limit the number of allowable queries to the generative model, thereby restricting the amount of data available to train an effective inversion model. To simulate this defense, we trained the inversion model using only 20%, 50%, and 80% of the original training dataset, instead of the full dataset. The corresponding attack performance under these conditions is presented in Tables 25, 26, 27, and 28. As shown, the performance of our approach decreases slightly as the amount of training data is reduced. However, this drop is relatively small, approximately 5% ~ 8%, even when using only 20% of the training set. This observation is further supported by the qualitative results in Table 29 and Figure 13, where the quality of reconstructed prompts and the corresponding regenerated images remains highly comparable to those obtained with 100% of the training data. These findings highlight the robustness of our approach in scenarios with limited query access to the target generative models.

Table 25: Defense Results of the Query Constraint Mechanism against Our Approach on Alpaca-GPT4

Data Used	BLEU (↑)	Similarity (↑)	ROUGE (↑)
20%	0.545	0.740	0.553
50%	0.570	0.754	0.576
80%	0.578	0.759	0.584
100%	0.593	0.771	0.596

Table 26: Defense Results of the Query Constraint Mechanism against Our Approach on RetrievalQA

Data Used	BLEU (↑)	Similarity (↑)	ROUGE (↑)
20%	0.550	0.744	0.530
50%	0.572	0.758	0.549
80%	0.584	0.766	0.559
100%	0.627	0.773	0.610

Table 24: Performance of Our Approach against the Output Perturbation Defense Mechanism in Text-to-Text Models

Question	Original answer	Perturbed answer	Reconstructed question based on original answer	Reconstructed question based on perturbed answer
Suggest a way to make a car more eco-friendly.	One way to make a car more eco-friendly would be to switch to a hybrid or electric engine. These types of engines produce fewer emissions and are more fuel-efficient compared to traditional gasoline engines. Another option is to consider carpooling or taking public transportation to reduce the overall number of cars on the road, therefore reducing carbon emissions. Additionally, regular maintenance such as keeping tires properly inflated, replacing air filters, and performing regular oil changes can also help improve fuel efficiency and reduce emissions.	An effective approach to enhance a car's eco-friendliness is to opt for a hybrid or electric engine, as these engines generate lower emissions and are typically more fuel-efficient than traditional gasoline engines. Carpooling or using public transportation can also significantly decrease the number of vehicles on the road, thereby cutting down on carbon emissions. Furthermore, routine upkeep, like maintaining correct tire pressure, changing air filters, and conducting regular oil changes, can contribute to better fuel efficiency and reduce overall emissions.	Suggest a way to make a car more eco-friendly.	Suggest an effective approach to enhance a car's eco-friendliness.
What is the leading cause of death for children under the age of 5?	According to the World Health Organization, the leading cause of death for children under the age of 5 worldwide is preterm birth complications (i.e., babies born too early), followed by pneumonia, birth asphyxia (when a baby doesn't get enough oxygen during the birthing process), diarrheal diseases, and malaria. It's important to note that the leading causes of death for children under 5 can vary by region and country, and efforts are being made globally to reduce child mortality through interventions such as vaccination programs, improved nutrition, and better access to healthcare.	Globally, young children's lives are most at risk due to complications from being born prematurely, followed closely by illnesses such as pneumonia, lack of oxygen during birth, diarrhea-related diseases, and malaria. Death rates for children under five differ widely depending on location, but ongoing international initiatives aim to lower these rates. Efforts focus on wider access to vaccines, better nutritional support, and improved healthcare systems.	What is the leading cause of death for children under the age of 5?	Summarize the current death rates for children under five.

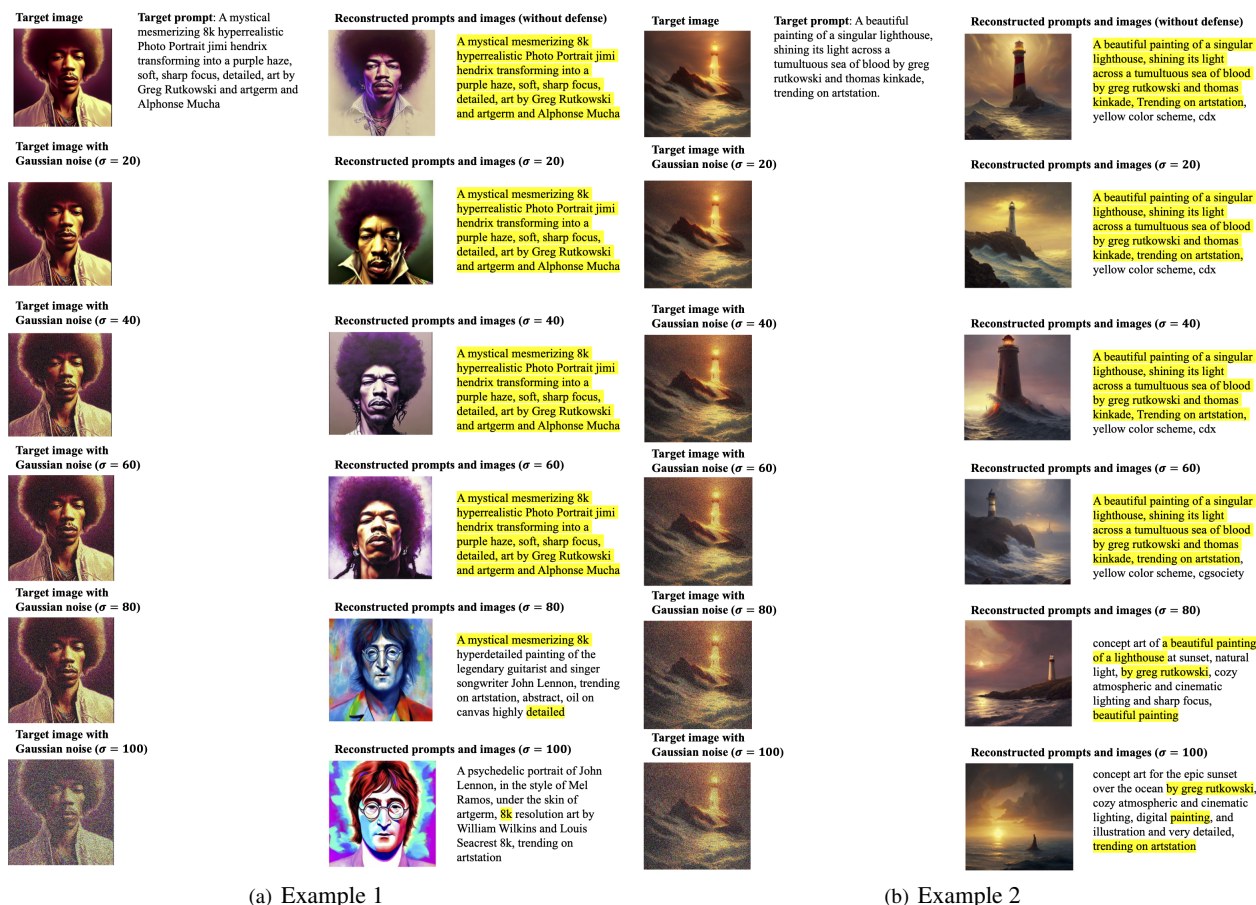


Figure 12: Performance of Our Approach against the Output Perturbation Defense Mechanism in Text-to-Image Models. As changes in the average Gaussian noise  $\mu$  have a limited impact on the inversion results, we only vary the standard deviation  $\sigma$ .

Table 27: Defense Results of the Query Constraint Mechanism against Our Approach on MS-COCO

Data Used	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
20%	0.944	0.941	0.942	0.836	0.607
50%	0.952	0.942	0.947	0.832	0.611
80%	0.951	0.942	0.946	0.839	0.609
100%	0.955	0.948	0.951	0.842	0.609

Table 28: Defense Results of the Query Constraint Mechanism against Our Approach on Stable-Diffusion-Prompts

Data Used	BERTScore			CLIP (↑)	LPIPS (↓)
	Preci. (↑)	Recall (↑)	F1 (↑)		
20%	0.873	0.840	0.856	0.789	0.614
50%	0.887	0.849	0.868	0.797	0.608
80%	0.889	0.852	0.870	0.801	0.605
100%	0.927	0.938	0.932	0.811	0.592

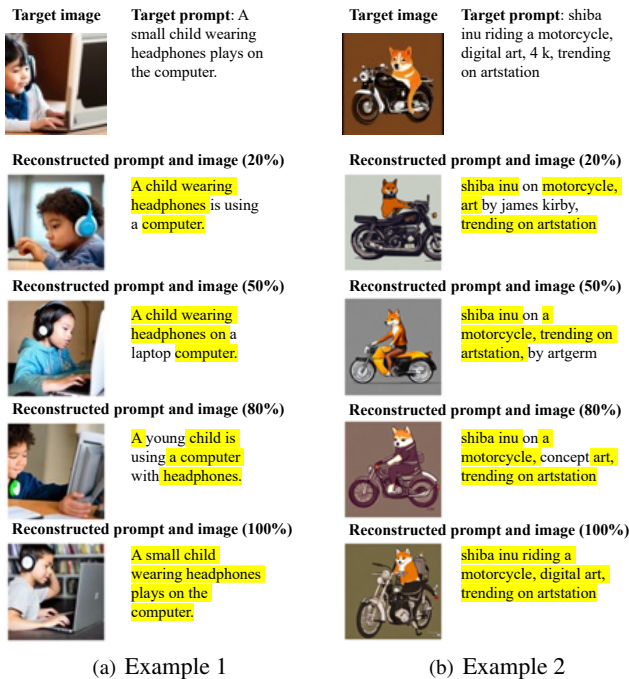


Figure 13: Performance of Our Approach against the Query Constraint Mechanism in Text-to-Image Models

## 7 Limitation and Discussion

The proposed approach may face limitations when applied to the text-to-audio context due to the inherent differences between audio signals and textual or visual outputs. Unlike text or images, audio outputs, such as speech or music, are high-dimensional and temporally dynamic. This makes it challenging to directly compare them with discrete text-based prompts during inversion. Moreover, our current reward function, based on BLEU similarity, is not directly applicable to

audio, where semantic similarity is harder to quantify.

A potential solution is to incorporate audio-to-text transcription models (e.g., wav2vec 2.0 [4]) as an intermediate step. By transcribing the generated audio into text, the inversion model can operate within the same discrete space as our original approach, allowing us to reuse existing reward functions and alignment strategies. Additionally, domain-specific embeddings (e.g., from AudioCLIP [11]) could be employed to compute semantic similarity between the original and reconstructed audio in a differentiable manner, enabling end-to-end training tailored to the audio modality. We leave a comprehensive investigation of prompt inversion for text-to-audio models to future work.

## 8 Related Work

### Prompt Inversion for Text-to-Text Generative Models.

This line of research, initiated by [39, 47, 49], focuses on stealing input prompts by analyzing the generated responses of LLMs. Specifically, Sha and Zhang’s method [39] includes a parameter extractor and a prompt reconstructor. The parameter extractor infers the prompt’s parameters, while the prompt reconstructor uses these inferred parameters to rebuild the entire prompt. Similarly, Yang et al.’s approach [47] involves two key phases: prompt mutation and prompt pruning. The prompt mutation phase optimizes surrogate prompt generation, and the prompt pruning phase refines the words related to the input in these surrogate prompts. Zhang et al.’s method [49] reconstructs prompts by encoding the LLM’s responses and utilizing a decoder to invert the encoded outputs. Another line of research in NLP focuses on reconstructing prompts through text embeddings [23] or next-token probabilities [24]. These studies aim to invert the behavior of LLMs rather than directly stealing the prompts themselves.

### Prompt Inversion for Text-to-Image Generative Models.

This area of research, initiated in [41], focuses on stealing prompts by analyzing the generated images from generative models. Their methodology includes a subject generator and a modifier detector. The subject generator is designed to identify the main subject of the stolen prompt, while the modifier detector aims to pinpoint keywords or key phrases relevant to the target image. Another early contribution by [20] employs a delayed projection scheme to optimize prompts, which effectively represent the vocabulary space, from randomly generated prompts. Additionally, a related line of research concentrates on generating captions from provided images [7, 43]. This line of research differs fundamentally from prompt inversion as it typically targets authentic images, whereas prompt inversion focuses on images generated by models.

**Summary.** The research discussed above targets either text-to-text or text-to-image generative models, with approaches that are not adaptable to multi-modal settings. Our research aims to bridge this gap by proposing a unified approach that

Table 29: Performance of Our Approach against the Query Constraint Mechanism in Text-to-Text Models

Target prompt	Response	Reconstructed prompt			
		20%	50%	80%	100%
Compare the similarities between two recent smartphone models. iPhone 11 Pro and Samsung Galaxy S20	The iPhone 11 Pro and Samsung Galaxy S20 are two recent smartphones models that share some similarities. These are: ...	Compare and contrast the iPhone 11 Pro and the Samsung Galaxy S20	Describe the features of two new smartphones, iPhone 11 Pro and Samsung Galaxy S20	Identify the similarities between two recent smartphones. iPhone 11 Pro and Samsung Galaxy S20	Compare the similarities between two recent smartphones. iPhone 11 Pro and Samsung Galaxy S20
How does an online only bank protect itself against fraud?	Online-only banks, also known as internet banks or virtual banks, use various methods to protect themselves and their customers against fraud. These methods may include: ...	How do online banks protect themselves and their customers against fraud?	How do online only banks protect themselves and their customers from fraud?	How do online only banks protect themselves and their customers against fraud?	How do online only banks protect themselves and their customers against fraud?

encompasses both text-to-text and text-to-image models, with the potential to extend to the text-to-video context as well.

## 9 Conclusion

In this paper, we present the first comprehensive study on prompt inversion targeting both text-to-text and text-to-image generative models, and further demonstrate its potential extension to the text-to-video setting. This practice not only infringes the privacy and intellectual property of prompt owners but also threatens the business integrity of prompt marketplaces. Our approach is model-agnostic, utilizing general model inversion techniques combined with reinforcement learning to ensure versatility across different types of generative models. The experimental results show the superior performance of our method when compared to existing techniques that are specifically designed for either text or image modalities alone. Looking ahead, we aim to broaden our research to include additional modalities, such as text-to-audio.

## Acknowledgement

This research is supported by NSFC-FDCT under its Joint Scientific Research Project Fund (Grant No. 0051/2022/AFJ).

## Ethical Statement

This research aims to advance the understanding of prompt inversion attacks and the associated security and privacy risks across text and multimodal generative models. The work is intended to raise awareness of these vulnerabilities within the academic community and to support the development of effective, evidence-based defenses. All datasets used in this study are publicly available and were accessed in compliance with their respective licenses and applicable privacy standards. No proprietary, sensitive, or personally identifiable information was used. We have also taken concrete steps to mitigate the risk of misuse. In particular, we withhold potentially sensitive implementation details, such as trained model parameters, that could be directly exploited for malicious purposes. Our goal

is to inform and strengthen the security posture of real-world generative systems, not to facilitate attacks. This research was conducted solely for academic purposes and in good faith. We welcome dialogue with stakeholders in both academia and industry to further improve the security and safety of generative AI technologies.

## Open Science Statement

To promote transparency and collaboration in the AI security community, we will release a comprehensive set of artifacts upon publication. This includes the full source code for our inversion framework, covering both the direct inversion and reinforcement learning-based fine-tuning components, along with all scripts used for data preprocessing, training, evaluation, and metric computation. We will also provide configuration files, hyperparameter settings, and environment specifications to enable replication of our results. All datasets used in our experiments are publicly available, and we will share the exact splits and preprocessing scripts used in our study to ensure consistency with our reported results.

To minimize the risk of misuse, we will not release the trained model checkpoints, as they could be directly exploited for reconstructing sensitive prompts. However, we will provide training logs and detailed instructions so that researchers can retrain the models from scratch. All artifacts will be made available through a public GitHub repository linked in the final version of the paper.

## References

- [1] Computational Pathology & AI. Mayo clinic and google cloud collaborate to revolutionize healthcare with generative ai, 2023. <https://www.pathologynews.com/computational-pathology-ai/mayo-clinic-and-google-cloud-collaborate-to-revolutionize-healthcare-with-generative-ai/>.

- [2] Stability AI. Stable Diffusion 3 Medium. <https://stability.ai/stable-image>.
- [3] A Askari, M Aliannejadi, E Kanoulas, and S Verberne. A test collection of synthetic documents for training rankers: Chatgpt vs. human experts. In *Proc. of CIKM*, page 5311–5315, 2023.
- [4] A Baevski, H Zhou, A Mohamed, and M Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020. <https://arxiv.org/abs/2006.11477>.
- [5] Q Cai, Z Yang, C Jin, and Z Wang. Provably Efficient Exploration in Policy Optimization. In *Proc. of ICML*, 2020.
- [6] S Changpinyo, P Sharma, N Ding, and R Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proc. of CVPR*, pages 3558–3568, 2021.
- [7] S Chen, Q Jin, P Wang, and Q Wu. Say As You Wish: Fine-Grained Control of Image Caption Generation With Abstract Scene Graphs. In *Proc. of CVPR*, pages 9962–9971, 2020.
- [8] Hugging Face. Alpaca-gpt4. <https://huggingface.co/datasets/vicgalle/alpaca-gpt4>.
- [9] E C. Garrido-Merchán, J L Arroyo-Barrigüete, and R Gozalo-Brizuela. Simulating h.p. lovecraft horror literature with the chatgpt large language model, 2023. <https://arxiv.org/abs/2305.03429>.
- [10] Gemini Team Google. Gemini: A Family of Highly Capable Multimodal Models, 2024. <https://arxiv.org/abs/2312.11805>.
- [11] A Guzhov, F Raue, J Hees, and A Dengel. AudioCLIP: Extending CLIP to Image, Text and Audio. In *Proc. of ICASSP*, 2022.
- [12] W Hong, M Ding, W Zheng, X Liu, and J Tang. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. In *Proc. of ICLR*, 2023.
- [13] B Hui, H Yuan, N Gong, P Burlina, and Y Cao. PLeak: Prompt Leaking Attacks against Large Language Model Applications. In *Proc. of CCS*, pages 3600–3614, 2024.
- [14] K Johnson. OpenAI Debuts DALL-E for Generating Images from Text, 2021. <https://venturebeat.com/business/openai-debuts-dall-e-for-generating-images-from-text>.
- [15] J Li, D Li, S Savarese, and S Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. <https://arxiv.org/abs/2301.12597>.
- [16] Y Li, Z Li, K Zhang, R Dan, S Jiang, and Y Zhang. ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. *Cureus*, 15(6), 2023.
- [17] T Lin, M Maire, S Belongie, L Bourdev, R Girshick, J Hays, P Perona, D Ramanan, C. L Zitnick, and P Dollár. Microsoft coco: Common objects in context, 2015. <https://arxiv.org/abs/1405.0312>.
- [18] V Liu and L B Chilton. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Proc. of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 384:1–23, 2022.
- [19] M Y. Lu and et al. A Multimodal Generative AI Copilot for Human Pathology. *Nature*, 634:466–473, 2024.
- [20] S Mahajan, T Rahman, K M Yi, and L Sigal. Prompting Hard or Hardly Prompting: Prompt Inversion for Text-to-Image Diffusion Models. In *Proc. of CVPR*, pages 6808–6817, 2024.
- [21] Meta. Discover the possibilities with Meta Llama, 2023. <https://llama.meta.com>.
- [22] V Mnih, A P Badia, M Mirza, A Graves, T P. Lillicrap, T Harley, D Silver, and K Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proc. of ICML*, 2016.
- [23] J X. Morris, V Kuleshov, V Shmatikov, and A M. Rush. Text Embeddings Reveal (Almost) As Much As Text. In *Proc. of EMNLP*, pages 12448–12457, 2023.
- [24] J X Morris, W Zhao, J T Chiu, V Shmatikov, and A M Rush. Language Model Inversion. In *Proc. of ICLR*, 2024.
- [25] B Ni, H Peng, M Chen, S Zhang, G Meng, J Fu, S Xiang, and H Ling. Expanding Language-Image Pretrained Models for General Video Recognition. In *Proc. of ECCV*, pages 1–17, 2022.
- [26] OpenAI. DALL-E API Now Available in Public Beta, 2022. <https://openai.com/blog/dall-e-api-now-available-in-public-beta>.
- [27] OpenAI. GPT-4 Technical Report, 2023. <https://cdn.openai.com/papers/gpt-4.pdf>.
- [28] OpenAI. GPT-4o System Card, 2024. <https://arxiv.org/abs/2410.21276>.
- [29] L Ouyang and et al. Training language models to follow instructions with human feedback, 2022. <https://arxiv.org/abs/2203.02155>.

- [30] K Papineni, S Roukos, T Ward, and W Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, page 311–318, 2002.
- [31] PromptBase. <https://promptbase.com>.
- [32] PromptDB. <https://promptdb.ai/publicprompts/>.
- [33] A Radford and et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of ICML*, 2021.
- [34] C Raffel, N Shazeer, A Roberts, K Lee, S Narang, M Matena, Y Zhou, W Li, and P J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [35] R Rombach, A Blattmann, D Lorenz, P Esser, and B Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proc. of CVPR*, pages 10684–10695, 2022.
- [36] C Saharia and et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, 2022. <https://arxiv.org/abs/2205.11487>.
- [37] G Santana. Stable-Diffusion-Prompts. <https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts>.
- [38] J Schulman, F Wolski, P Dhariwal, A Radford, and O Klimov. Proximal Policy Optimization Algorithms, 2017. <https://arxiv.org/abs/1707.06347>.
- [39] Z Sha and Y Zhang. Prompt stealing attacks against large language models, 2024. <https://arxiv.org/abs/2402.12959>.
- [40] X Shen and et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding, 2024. <https://arxiv.org/abs/2410.17434>.
- [41] X Shen, Y Qu, M Backes, and Y Zhang. Prompting stealing attacks against text-to-image generation models. In *Proc. of USENIX Symposium*, pages 5823–5840, 2024.
- [42] Claude Team. The Claude 3 Model Family: Opus, Sonnet, Haiku, 2024. <https://paperswithcode.com/paper/the-claude-3-model-family-opus-sonnet-haiku>.
- [43] A Verma, A K Yadav, M Kumar, and D Yadav. Automatic Image Caption Generation Using Deep Learning. *Multimedia Tools and Applications*, 83:5309–5325, 2024.
- [44] B Wang, Q Chen, and Z Wang. Diffusion-based visual art creation: A survey and new perspectives, 2024. <https://arxiv.org/abs/2408.12128>.
- [45] W Wang and Y Yang. VidProM: A Million-scale Real Prompt-Gallery Dataset for Text-to-Video Diffusion Models. In *Proc. of NeurIPS*, 2024.
- [46] Y Wen, N Jain, J Kirchenbauer, M Goldblum, J Geiping, and T Goldstein. Hard Prompts Made Easy: Gradient-Based Discrete Optimization for Prompt Tuning and Discovery. In *Proc. of NeurIPS*, 2023.
- [47] Y Yang, X Zhang, Y Jiang, X Chen, H Wang, S Ji, and Z Wang. Prsa: Prompt reverse stealing attacks against large language models, 2024. <https://arxiv.org/abs/2402.19200>.
- [48] Z Yang, J Zhang, E Chang, and Z Liang. Neural Network Inversion in Adversarial Setting via Background Knowledge Alignment. In *Proc. of CCS*, page 225–240, 2019.
- [49] C Zhang, J X. Morris, and V Shmatikov. Extracting Prompts by Inverting LLM Outputs. In *Proc. of EMNLP*, pages 14753–14777, 2024.
- [50] R Zhang, P Isola, A A. Efros, E Shechtman, and O Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. of CVPR*, pages 586–595, 2018.
- [51] T Zhang, V Kishore, F Wu, K Q. Weinberger, and Y Artzi. BERTScore: Evaluating Text Generation with BERT. In *Proc. of ICLR*, 2020.
- [52] Z Zhang, M Fang, and L Chen. RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering. In *Proc. of ACL*, pages 6963–6975, 2024.
- [53] J Zhou and et al. Pre-trained Multimodal Large Language Model Enhances Dermatological Diagnosis Using SkinGPT-4. *Nature Communications*, 15:5649:1–12, 2024.

## Appendix

### 1 Computation Overhead

Training these inversion models typically takes about 3 hours on a single RTX 4090 GPU with 24 GB of memory. For example, training an inversion model against text-to-text models based on T5-small, using 16,704 data points over 50 epochs, requires approximately 2.5 hours. Fine-tuning this model with RL using an additional 688 data points over another 50 epochs takes around 1 hour. Similarly, training an inversion model against text-to-image models based on BLIP-2, using 1,410 data points over 50 epochs, takes about 1.6 hours, followed by RL fine-tuning with an additional 138 data points over 50 epochs, which requires a further 0.6 hours.