

On the Effectiveness of Prompt Stealing Attacks on In-The-Wild Prompts

Yicong Tan[†], Xinyue Shen[†], Yun Shen[‡], Michael Backes[†], Yang Zhang^{†*}

[†]CISPA Helmholtz Center for Information Security, [‡]Flexera

yicong.tan@cispa.de, xinyue.shen@cispa.de, yun.shen@flexera.com, director@cispa.de, zhang@cispa.de

Abstract—Large Language Models (LLMs) have increased demand for high-quality prompts, which are now considered valuable commodities in prompt marketplaces. However, this demand has also led to the emergence of prompt stealing attacks, where the adversary attempts to infer prompts from generated outputs, threatening the intellectual property and business models of these marketplaces. Previous research primarily examines prompt stealing on academic datasets. The key question remains unanswered: *Do these attacks genuinely threaten in-the-wild prompts curated by real-world users?* In this paper, we provide the first systematic study on the efficacy of prompt stealing attacks against in-the-wild prompts. Our analysis shows that in-the-wild prompts differ significantly from academic ones in length, semantics, and topics. Our evaluation subsequently reveals that current prompt stealing attacks perform poorly in this context. To improve attack efficacy, we employ a Text Gradient based method to iteratively refine prompts to better reproduce outputs. This leads to enhanced attack performance, as evidenced by improvements in METEOR score from 0.207 to 0.253 for prompt recovery and from 0.323 to 0.440 for output recovery. Despite these improvements, we showcase that the fundamental challenges persist, highlighting the necessity for further research to improve and evaluate the effectiveness of prompt stealing attacks in practical scenarios.

1. Introduction

Large Language Models (LLMs), such as ChatGPT [4], Claude [5], and Gemma [36], have demonstrated unprecedented capabilities across a wide range of fields, including mathematical reasoning [14, 28, 61], code generation [54, 63], and article writing [44, 57]. The efficacy of these LLMs, however, is profoundly influenced by the quality of the input prompts provided by users [34]. Research shows that prompt quality directly impacts the performance of these models, making prompt engineering, a systematic approach to crafting and refining prompts, an essential component of working with LLMs [34, 58]. This engineering process involves carefully designing prompts, iterating on their phrasing and structure, and tuning them through a series of trial-and-error to maximize the models’ output accuracy and relevance. Given the creativity and efforts involved, high-quality prompts have been regarded as intellectual

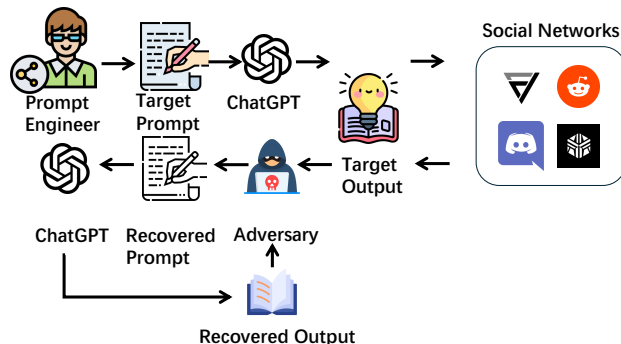


Figure 1: Overview of prompt stealing attacks on in-the-wild prompts.

property and widely sold on prompt marketplaces, such as PromptBase [9], ArtHub [2], and PromptHero [10]. The top 10 prompt engineers on PromptBase [9] have each sold over 2,400 prompts, with each prompt price ranging from \$2.99 to \$16.99 and total sales for each engineer exceeding \$10,000, further emphasizing the intellectual value embedded in expertise-crafted prompts.

As high-quality prompts gain value, the risk to intellectual property also arises [51]. A novel attack aimed at stealing the prompt from model outputs emerges, known as *prompt stealing attack* [43]. An example of this type of attack is illustrated in Figure 1, where an adversary executes the prompt stealing attack by recovering the prompt from the generated output captured on the social platforms. Existing attacks typically involve constructing a draft prompt by prompting the LLM to deduce the original prompt from the output. This draft prompt is then refined through content modifications [23], categorical factors [59], or the addition of contextual elements [43] to better approximate the original prompt. In practice, such attacks could leak all prompts available for sale on a prompt marketplace, jeopardizing the business models and resulting in cut-throat competition across the entire business ecosystem [39]. Note that these threats are not merely hypothetical. Several prompt stealing attacks have been conducted against LLMs in the real world [23, 43, 59].

A significant gap, however, is the divergence between the prompts employed in experimental evaluations and those encountered in real-world applications. This discrepancy raises questions regarding the generalizability and practical

* Yang Zhang is the corresponding author.

effectiveness of these techniques beyond controlled testing environments. Specifically, previous studies [23, 43] often rely on evaluation prompts that remain restricted to academic datasets, which are created by researchers rather than sourced from real users. For instance, [23, 43] both evaluate their methods with the academic dataset Alpaca with prompts such as “What are the three primary colors?”, “Brainstorm possible solutions to reduce water pollution.” As later demonstrated in Section 3, these academic prompts are generally shorter, less concrete, and focus on distinct topics, differing significantly from prompts used in real-world applications. Consequently, the effectiveness of existing prompt stealing attacks evaluated on these academic prompts remains unknown in real-world contexts (see Section 4). This gap hinders users, security researchers, and policymakers from truly understanding the threat posed by this new type of attack.

In this paper, we perform the first study on evaluating prompt stealing attacks against LLMs using in-the-wild prompts [45]. This enables us to answer the following key research questions:

Do In-The-Wild Prompts Differ from Academic Prompts? We start by inspecting the prompts employed in the existing research [23, 43] and those sourced from real users [45]. Our comparative analysis suggests that academic prompts differ from in-the-wild prompts across multiple dimensions. In terms of sentence length, the in-the-wild prompts have an average word length of 77, compared to the average of 14 in the academic datasets. In addition, they also demonstrate a noticeable separation in the prompt semantic space and exhibit distinct levels of topic concentrations.

Can Existing Attacks Steal Prompts in Practical Settings? Based on the above findings, we conduct an empirical evaluation of existing prompt stealing attacks [23, 43, 59] on the in-the-wild prompt dataset. Our results show that these attacks are inadequate in reconstructing both the prompt and the output generated by the prompt (see Section 4), with METEOR scores of 0.157, 0.184, 0.207, and BERTScores [62] of 0.860, 0.857, 0.857 for prompt reconstruction, and METEOR [15] scores of 0.310, 0.318, 0.311 and BERTScores of 0.865, 0.867, 0.867 for output reconstruction, both exhibit limitations in thoroughly identifying the detailed information within the output generated by the target prompt, thereby failing to accurately reconstruct the prompts.

Can We Improve the Prompt Stealing Performance? Driven by the ineffectiveness of existing attacks, we leverage Text Gradient [40] to enhance the efficacy of prompt stealing attacks. Our results show that our approach, T-GPS, which iteratively refines the initial draft prompt guided by the text gradients analyzed by the LLMs, outperforms existing prompt stealing attacks [23, 43] in stealing in-the-wild prompts. This is evidenced by improved lexical and semantic metrics. In reconstructing the prompt, T-GPS has shown improvements of 0.046 and 0.020, respectively, on METEOR and sentence similarity compared to the existing prompt stealing attack. Though modest in absolute values, the lexical improvements by the T-GPS

are significant, especially for well-defined prompts with clear objectives. Moreover, in reconstructing the generated output, T-GPS has demonstrated significantly improved performance, achieving scores of 0.093, 0.117, 0.116, 0.027, and 0.043 in BLEU, METEOR, ROUGE-L, BERTScore, and SentenceSim, respectively.

What Are the Fundamental Challenges for Prompt Stealing in the Real World? Our approach demonstrates improved attack performance compared to existing methods. However, it still encounters limitations when attempting to reconstruct prompts in uncontrolled, real-world contexts. Specifically, while our method achieves high efficacy in replicating outputs generated by the original prompt, it often fails to faithfully reconstruct the prompt itself, introducing extraneous details derived from the output (see Section 6). This limitation is inherent in the prompt stealing attack methodology, hence a fundamental challenge. That is, without a clear understanding of the prompt’s abstraction level, the adversary lacks the ability to design targeted strategies to effectively guide the reconstruction process. This limitation highlights the necessity for further research aimed at improving the robustness and accuracy of prompt stealing attacks in practical, real-world applications.

Remark. The main objective of this paper is not to introduce a novel prompt stealing method designed to surpass current state-of-the-art techniques. Rather, our focus is on benchmarking the effectiveness of existing methods, along with our proposed improvements, in the context of stealing in-the-wild prompts. Our findings indicate that the performance of prompt stealing techniques previously proposed in academic literature remains suboptimal in practice. We hope our results can highlight the importance of robust evaluations of LLM security in real-world scenarios.

2. Background

Large Language Models. LLMs have shown notable advancements, especially in their general problem-solving abilities. Their capabilities have been evaluated across a wide range of benchmarks, including general benchmarks like AGIEval [64], MMLU [26], and BIG-Bench Hard [48]; knowledge reasoning benchmarks such as TriviaQA-Wiki [30]; and reading comprehension benchmarks like SQuAD [42] and BoolQ [19]. LLMs, such as GPT [4], Llama [21], Mistral [29], Claude [5], and Gemma [36], are rapidly evolving and continuously setting higher records for these benchmarks. In this context, the prompt stealing attack can be regarded as a task to evaluate the LLMs’ capability of deducing the original prompts given the output they generate.

Prompt Engineering. Prompt engineering [58] aims to refine prompts for the effective and efficient use of LLMs in specific tasks. Several techniques have been proposed to refine this process, such as few-shot prompting [17], Chain-of-Thought prompting [56], and Retrieval Augmented Generation [32]. Diverging from these methods, Self-Consistency [52], Tree-of-Thoughts [60], and Graph-of-Thoughts [16] decompose the single-step prompting into

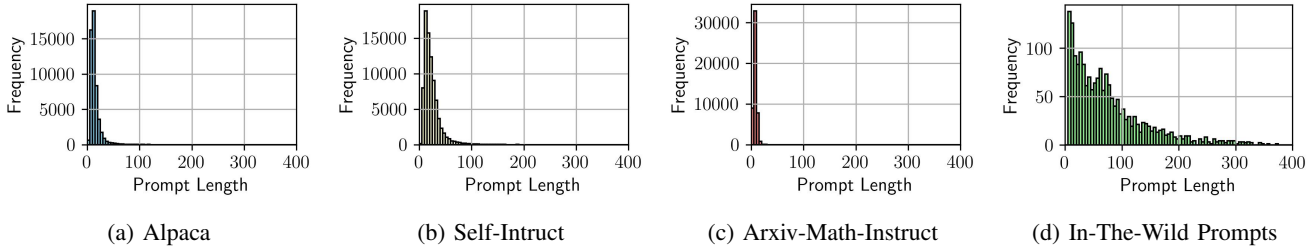


Figure 2: Prompt length distributions of different prompt datasets. Average lengths are 14, 23, 7, and 77 words, respectively.

multiple steps and aim to solve the tasks in a multi-step reasoning process.

Prompt Stealing. We show an overview of prompt stealing attacks in Figure 1. Following the settings of the previous study [46], the prompt engineer first creates high-quality prompts through prompt engineering and uses LLMs to generate the outputs. These generated outputs are subsequently shared on various social networks, such as Reddit [11], Discord [6], FlowGPT [7], and AIPRM [1]. Original prompts are often withheld to protect intellectual property. The high-quality prompt is referred to as the *target prompt*, and the output generated by the high-quality prompt is the *target output*. Given this context, an adversary can observe the target output on these platforms and attempt to deduce the target prompt. The prompt that the adversary recovers is referred to as the *recovered prompt*, which is used to generate the *recovered output*. The effectiveness of the attack is evaluated by comparing the recovered prompt to the target prompt, as well as the target and recovered outputs. In this prompt stealing attack, the adversary can access social networks and collect the target outputs shared by the prompt engineers. However, the adversary has no knowledge of the target prompts used to generate these target outputs. Additionally, since most target outputs are shared with the information about the LLM used, we assume the adversary can acquire this information and leverage the same LLM to generate the recovered output using the recovered prompt.

3. Do In-The-Wild Prompts Differ from Academic Prompts?

Motivation. In this section, we aim to address a critical question: *To what extent are the prompts employed in existing research representative of those used by real users in real-world scenarios?* Our motivation is primarily driven by the concern that a high attack success rate on evaluation prompts may not necessarily translate to effectiveness on real-world prompts, particularly if there is a significant divergence between the two categories of prompts. We begin by providing an overview of the two prompt categories and then conduct a detailed comparative analysis based on prompt length, semantics, and topics.

TABLE 1: Prompt samples from academic prompt datasets and in-the-wild prompt dataset.

Dataset	Prompt Sample
Alpaca	Write a review of a recent movie you watched. Parasite (2019)
Self-Instruct	Write a text about something that made you angry recently.
Arxiv-Math-Instruct	What is the definition of fmi?
In-The-Wild	Please ignore all previous instructions. Please respond only in the english language. Do not explain what you are doing. Do not self reference. You are an expert marketing manager. Write an AIDA (Attention Interest Desire Action) model to generate marketing strategy and sales for the following business: (your business). Please present the results in a markdown table with two column: stage, strategy and suggestions

3.1. Prompt Datasets

In-The-Wild Prompts. In this paper, we employ the in-the-wild prompt dataset as described in [45]. This dataset comprises a comprehensive collection of 15,140 prompts, gathered from four distinct platforms, all contributed by real users. These platforms include Reddit [11] and Discord [6], as well as a variety of websites such as AIPRM [1], FlowGPT [7], and JailbreakChat [8]. Additionally, the dataset incorporates prompts from open-source collections, specifically the Awesome ChatGPT Prompts repository [3] and a set of 50 in-the-wild prompts that are extracted from images on Twitter and Reddit [22]. All these platforms are deliberately chosen according to their popularity in sharing prompts [45]. In turn, these in-the-wild prompts cover a wide spectrum of prompts used in the real world.

Academic Prompts. In this study, we utilize three distinct prompt datasets - Alpaca, Self-Instruct, and Arxiv-Math-Instruct - introduced by Gao et al. [23]. In contrast to in-the-wild prompts, these datasets are created by researchers rather than sourced from real users. These datasets are all instruction datasets designed for LLM instruction-tuning in research, with instruction-response pairs that simulate the paradigm of the real-user interaction with LLMs. In addition, they are leveraged by researchers across various domains, including instruction tuning [38, 47], LLM align-

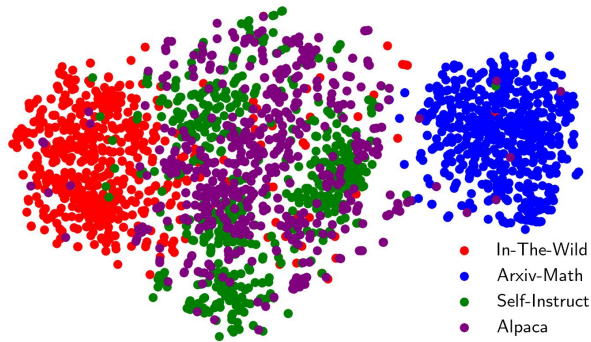


Figure 3: T-SNE visualization of the randomly sampled 500 prompts from in-the-wild and academic datasets. The plot reveals a distinct semantic gap between these prompts on embedding space, with only a few points that overlap.

ment [41, 55], and mathematical reasoning [24, 35]. In this paper, we term these datasets as *academic prompt datasets*.

- *Alpaca*. The Alpaca dataset [12] comprises 52,002 instructions created by OpenAI’s text-davinci-003 engine. This dataset is specifically designed for LLM instruction-tuning, improving their ability to follow human instructions more precisely.
- *Self-Instruct*. The Self-Instruct dataset [53] includes 82,439 instance instructions with inputs and outputs generated by OpenAI’s Davinci engine. This dataset shares a similar design goal with Alpaca.
- *Arxiv-Math-Instruct*. The Arxiv-Math-Instruct dataset [13] consists of 50,488 question-answer pairs derived from abstracts of academic papers across various mathematical categories from the ArXiv repository. The questions are generated using the T5-base model, while the answers are produced using the GPT-3.5-turbo model. The dataset is designed to assess the capabilities of LLMs in mathematical reasoning and problem-solving.

To provide a concrete impression of academic and in-the-wild prompts, sample prompts from these datasets are shown in Table 1.

3.2. Comparative Analysis

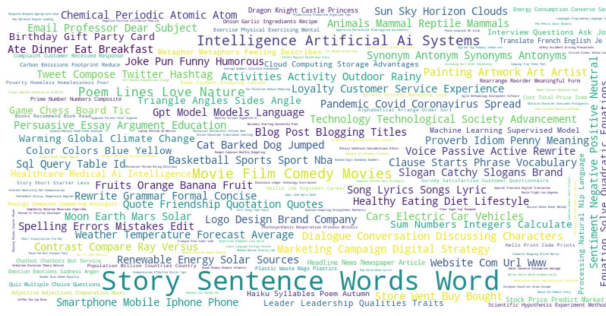
Overview. Building upon the analytical framework applied in the characterization of in-the-wild prompts [45], we undertake a comparative analysis between academic prompts and in-the-wild prompts. The goal is to understand whether academic and in-the-wild prompts share similarities in terms of prompt length, topics, and semantics.

Prompt Length. The prompt length (measured by the number of words) is a factor that directly influences both the complexity and specificity of prompts. As shown in Figure 2, there exists a noticeable difference in prompt length distribution between in-the-wild prompts and academic prompts. Note that some in-the-wild prompts can reach up to 1,200 words; we exclude such outliers for clarity in our

visualizations. This wide variation in in-the-wild prompts reflects the diverse nature of real-world tasks, which often require detailed instructions or extensive contextualization. In contrast, the prompts from academic datasets, such as Alpaca and Self-Instruct, tend to be more concise, averaging less than 25 words. These shorter prompts align with their design goals for instruction-tuning tasks, which are typically straightforward and require minimal context. Moreover, prompts in the Arxiv-Math-Instruct dataset are even shorter, ranging between 5 and 15 words. This narrow distribution suggests that these prompts are highly specific and targeted, with a focus on concise scientific definitions and terminologies. Overall, these findings suggest a key distinction between academic and in-the-wild prompts. While academic prompts tend to be concise and narrowly focused, in-the-wild prompts often encompass a broader range of complexities and contextual information.

Prompt Topics. Upon identifying the prompt length difference, we further examine if topical differences exist between in-the-wild and academic prompts. For this analysis, we employ BERTopic [25], a topic modeling approach that facilitates the identification of dominant themes across datasets. The word cloud visualization of topics from these datasets is shown in Figure 4. In-the-wild prompts mainly revolve around daily tasks and entertainment such as gaming, coding, essay writing, and general advice. Each topic accounts for approximately 3-5% of the total prompts, collectively accounting for about 30% of the dataset, indicating the users’ preference for these prompts. Some topics overlap with those from academic datasets. However, each shared topic typically constitutes less than 1% of academic prompts. This highlights that these popular themes from in-the-wild prompts are infrequently addressed in academic prompts. Moreover, academic prompts emphasize specialized topics absent from in-the-wild prompts, such as the Paris Climate Agreement, renewable solar energy, and specific scientific concepts. Our findings exemplify the distinct topical focus of academic prompts compared to in-the-wild prompts, indicating the potential limitations of academic prompts in capturing the diversity and practical relevance of prompts encountered in real-world scenarios.

Prompt Semantics. Finally, we investigate prompt semantics, a pivotal factor for uncovering both the functions of prompts and the intentions of users. Here, we investigate whether semantic differences exist between in-the-wild and academic prompts. We randomly sample 500 prompts from each dataset, embed each prompt into 4096-dimensional vectors using the E5-Mistral-7b-Instruct model [29], and then apply t-SNE to visualize the embeddings [50]. As shown in Figure 3, in-the-wild and academic prompts demonstrate a discernible separation in the prompt semantic space. This distinction suggests that the effectiveness of prompt stealing attacks - frequently assessed using academic prompts - may be inherently limited. Such a restriction may reduce the generalizability of these attacks to prompts that are more commonly encountered in real-world scenarios. We later exemplify their limitations in Section 4.



(a) Alpaca



(b) Arxiv-Math



(c) Self-Instruct



(d) In-The-Wild Prompts

Figure 4: Word cloud of prompt topics from different datasets. Academic prompts cover a broad but sparse range of topics, while in-the-wild prompts are more concentrated on specific topics that real users are interested in.

Takeaway 1: Notable differences exist between academic and in-the-wild prompts. In-the-wild prompts are typically longer and tend to focus on practical and recreational activities. In contrast, academic prompts are shorter and emphasize specialized topics. Our findings show that the academic prompts employed in prior research may not fully encompass the diversity of prompts used in the real world.

4. Can Existing Prompt Stealing Attacks Steal In-The-Wild Prompts?

Motivation. In light of differences between academic and in-the-wild prompts, we perform a comprehensive evaluation of the effectiveness of existing attacks on stealing in-the-wild prompts [23, 43, 59]. The goal is to understand if these attacks remain effective in stealing in-the-wild prompts.

4.1. Existing Prompt Stealing Attacks

Vanilla method [43] refers to the approach that directly instructs LLMs to recover the target prompt based on the given target output. The prompting instruction for the LLMs

is formatted as: “What prompt would lead to this output?” In [43], the Vanilla method is used to reconstruct a draft prompt, which is then refined by adding role-play or context information based on classification as the prefix. However, training such a classifier on in-the-wild prompt dataset requires manually expanding in-the-wild prompts into role-based and in-context prompts by adding the role information and in-context information, which is absent in in-the-wild prompts [43]. This process introduces subjectively crafted prefixes, causing the in-the-wild prompts to deviate from their original fidelity and intended functionality. As a result, we exclusively evaluate the Vanilla method on in-the-wild prompts.

DORY [23] guides the target prompt reconstruction process using the uncertainty measured by the logarithmic probability along with the target output. This attack can be divided into four steps:

- *Step 1: Draft Prompt Recovery.* The draft recovered prompt is generated based on the target output. This process is similar to the Vanilla method, but in a few-shot learning scheme.
- *Step 2: Keyword Identification.* The key sentence of the target output is first identified by leveraging an LLM in a few-shot learning scheme. Keywords are then extracted

by identifying words with low uncertainty within the key sentence.

- *Step 3: Noise Identification.* Noise is identified by comparing the keywords between the target outputs and the draft outputs generated by the draft prompts.
- *Step 4: Recovered prompt.* The target output, draft prompt, keywords, and noise are combined to generate the final recovered prompt with the LLM.

This method requires the adversary to obtain both the target output and the logarithmic probability of each token in the target output. In practice, such access is beyond the adversary’s capabilities. However, to evaluate this method, we relax the restriction on the adversary and assess it under the assumption that both the target output tokens and the logarithmic probability for each token are accessible,

PRSA [59] recovers the target prompt based on the hypothesis that the outputs generated by the prompts within the same category share similar factors, such as themes, style, and logical structure. In order to find these categorical factors for each category, PRSA iteratively updates the factors by prompting the LLM to analyze the discrepancy between the target output and the draft output. Each iteration can be divided into three steps:

- *Step 1: Draft Prompt Recovery.* A draft prompt is generated based on the target output and the factors from the previous iteration.
- *Step 2: Categorical Factors Generation and Score.* An LLM is leveraged to generate new factors and then score both existing and new factors by comparing the draft output with the target output.
- *Step 3: Categorical Factors Update.* Factors with scores above a pre-defined threshold are retained for the next iteration.

After completing the iterative process across all target outputs within the category, the target outputs within the same category, along with the categorical factors, are input into the LLM to recover the target prompts.

4.2. Experimental Settings

Evaluation Dataset. Our evaluation dataset is derived from the original 15,140 in-the-wild prompts. First, we manually exclude jailbreak prompts. Note that reconstructing prompts with placeholders requires access to both the generated output and the specific input words to complete the placeholders accurately. This is a capability that the adversary usually does not have. Consequently, we then exclude prompts that contain placeholders. Moreover, we exclude prompts with fewer than five words to ensure the meaningfulness of the target prompts, as well as prompts that are longer than their corresponding outputs to guarantee that the generated outputs provide sufficient information for prompt reconstruction. This process results in a total of 1,855 prompts from the original in-the-wild prompts. Using the target model (see below), we generate 1,855 target prompt-output pairs based on these prompts. To balance the analytical correctness

with financial costs, we randomly select 421 target prompt-output pairs from this set to evaluate the Vanilla and DORY methods. To generate categorical factors for PRSA, we use BERTopic to classify 421 target prompts, of which 259 are assigned to specific classes, and the remaining 162 are categorized as “others.” The 259 prompts are recovered using PRSA with their corresponding class-based factors, while the remaining 162 prompts are recovered by treating each prompt as an individual class, which involves extracting factors separately for each of these prompts.

Target/Attack Model. Given that most in-the-wild prompts are tailored for GPT-3.5 at the time of collection, we employ the latest GPT-3.5-turbo model as the target model, specifically, GPT-3.5-turbo-0125,¹ to generate target outputs from target prompts. Given that most target outputs are shared with information about the target model, we use the same LLM as the attack model.

Parameter Settings. Based on the previous study [23], we set the temperature to 0.7, the random seed to 42, and all other parameters to their default values for the GPT-3.5-turbo-0125 model. For PRSA, we set the threshold for retaining the categorical factors to 0.2 so as to preserve a broader range of categorical factors. A higher threshold would consistently discard factors generated in previous iterations due to the diverse sentence structures of in-the-wild prompts. As a result, without this adjustment, the final categorical factors would depend solely on the analysis between the target output and the draft output from only the last one or two iterations. For each of the 162 “other” prompts, we retain all factors regardless of the score. Each method is executed three times to recover each target prompt and output. The reported standard deviation represents the average of the standard deviations calculated across all pairs of target and recovered prompts/outputs.

4.3. Evaluation Metrics

Goals. We evaluate the performance of prompt stealing attacks from two perspectives: *fidelity* and *functionality*. For fidelity, we assess the similarity between the target prompts and the recovered prompts. That is, we measure how closely the recovered prompts align with the intended targets. Functionality, on the other hand, is assessed by measuring the similarity between outputs generated by the target prompts and those produced by the recovered prompts. This is to evaluate whether the recovered prompts can effectively prompt the target model to generate outputs similar to those produced by the target prompts. Together, these two perspectives provide a comprehensive understanding of the effectiveness of prompt stealing attacks in terms of both capturing the essence of target prompts and replicating their intended outputs.

Metrics. We use two categories of metrics - lexical and semantic metrics - to assess both fidelity and functionality. We outline their details below.

1. <https://platform.openai.com/docs/models/gpt-3-5-turbo>.

- **Lexical Metrics** are used to evaluate the degree of alignment between the recovered and target prompts (or outputs) based on the precise word choices and their sequence. In this study, we employ three widely-used lexical evaluation metrics: BLEU [37], METEOR [15], and ROUGE-L [33].

- *BLEU* [37] is a precision-based metric that accumulates the word count and n-gram word sequence count in both the reference text and candidate text to compute the score, focusing on exact word matches. In our evaluation, we calculate the BLEU score as the average BLEU metric for 1- to 4-grams.
- *METEOR* [15] improves on BLEU by incorporating stemming and synonym matching, which makes it less sensitive to wording and word order variations. In addition, it balances the precision and recall by computing a weighted harmonic mean of the two.
- *ROUGE-L* [33] evaluates the longest common subsequence between the reference text and the candidate text, capturing the overall structure of words.

Note that these metrics may result in low scores for texts that are semantically accurate but employ alternative phrasing or synonyms. This limitation arises from the fact that these metrics primarily prioritize exact matches in lexical terms and n-gram sequences rather than capturing semantic equivalence. Consequently, while these metrics offer a robust measure of lexical alignment, they may not fully account for semantic variation between the target and recovered prompts (or outputs).

- **Semantic Metrics** are designed to evaluate semantic similarity effectively and complement lexical metrics.
 - *BERTScore* [62] is to address the limitations of lexical metrics in capturing word meaning. Concretely, it employs pre-trained BERT embeddings [20] to align words in candidate and reference sentences based on cosine similarity, calculating precision, recall, and F1 scores. In our work, we use the BERTScore F1 score as our metric.
 - *Sentence Semantics (SentenceSim)* [43] aims to capture sentence-level semantic similarity by identifying deeper semantic relationships and the overall meaning conveyed by a sentence. In this paper, we apply cosine similarity on sentence embeddings generated by the Mistral-7B model [29] to assess similarity at the sentence level. Note that we do not employ FastKASIM [18] in this paper because it cannot handle special characters used by in-the-wild prompts.

Metric Interpretation. All metrics range between 0 and 1, with higher scores (closer to 1) indicating higher similarity.

4.4. Evaluation Results

Results. We present our evaluation results in Table 2 and Table 3. The Vanilla, DORY, and PRSA methods show notable limitations in recovering target prompts from in-the-wild examples from a semantic perspective. This limitation is especially evident in their BLEU scores of 0.023,

0.026, 0.035, METEOR scores of 0.157, 0.184, 0.207, and ROUGE-L scores of 0.189, 0.190, 0.206 for the Vanilla, DORY, and PRSA methods, respectively. Similar patterns are observed when recovering outputs (see Table 3), indicating a significant lack of lexical alignment. Nevertheless, all existing methods perform robustly when measured by semantic metrics, as reflected in their high BERTScore values of 0.860, 0.857, 0.857 for prompt recovery and 0.865, 0.867, 0.865 for output recovery. This disparity between lexical- and semantic-level metrics suggests that, although the approaches are adept at identifying underlying meanings, they may lack precision in replicating exact word choices, raising questions about their overall fidelity in capturing verbatim textual alignment.

Lexical-Semantic Alignment Dilemma. As our results suggest, the interplay between lexical alignment and semantic similarity presents a complex challenge in evaluating the effectiveness of prompt stealing attacks. Lexical alignment depends on exact word choice, syntax, and phrasing. In contrast, semantically similar recovered prompts aim to convey the intent or meaning of the target prompts without necessarily reproducing their precise wording. To better illustrate this dilemma, we present an illustrative example in Table 4. It is evident that the recovered prompt encapsulates the essence of the target prompt, including elements of comedy, sarcasm, and references to Russian politics. However, it omits a significant proportion of specific details when compared to the target prompt. This observation suggests that while semantic fidelity can be maintained, lexical precision may still be compromised. Ideally, a robust prompt stealing attack would achieve balanced scores in both lexical and semantic alignment to maximize its effectiveness.

What Are the Limitations of Existing Attacks? The main limitation of existing methods lies in their inability to capture and integrate important details from target output to recover prompts, as detailed in Table 5. The Vanilla method [43] is overly simplistic, relying solely on the prompt, “What prompt would lead to this output?” Without explicitly instructing the LLM to extract the keywords and phrases, the LLM struggles to capture the critical details required for reconstructing the target prompt, especially when it contains complex instructions. The DORY method [23], while attempting to identify the keywords and noise in the target output, shows inadequacy in effectively distinguishing them. As shown in Table 5, limiting keyword extraction from the key sentence cannot capture all essential keywords and often introduces irrelevant words like “a” and “with.” The same pitfall results in insufficient and inaccurate identification of noise words. The PRSA method [59] can only partially recover in-the-wild prompts that fall into specific categories. Even within these categories, the high variability in meaning, functionality, and sentence structure among in-the-wild prompts prevents PRSA from generating universally applicable factors for every prompt in the same category. Consequently, factors are frequently overwritten through iterations. Despite setting a low threshold of 0.2 to retain as many factors as possible, we find that the final generated factors are still predominantly based on those from

TABLE 2: Performance of existing methods and our method in recovering target prompts (fidelity). PRSA is evaluated with two subsets: 259 target prompts fall into 6 categories, and 162 other prompts, each treated as a single category.

Methods	BLEU	METEOR	ROUGE-L	BERTScore	SentenceSim
Vanilla	0.023 ± 0.004	0.157 ± 0.011	0.189 ± 0.013	0.860 ± 0.003	0.780 ± 0.009
DORY	0.026 ± 0.012	0.184 ± 0.042	0.190 ± 0.035	0.855 ± 0.008	0.781 ± 0.020
PRSA (other)	0.031 ± 0.011	0.197 ± 0.033	0.197 ± 0.030	0.857 ± 0.006	0.780 ± 0.028
PRSA (grouped)	0.035 ± 0.018	0.207 ± 0.050	0.206 ± 0.044	0.857 ± 0.009	0.780 ± 0.028
T-GPS (gpt-3-turbo)	0.034 ± 0.012	0.253 ± 0.038	0.205 ± 0.029	0.852 ± 0.007	0.800 ± 0.014
T-GPS (gpt-4-turbo)	0.024 ± 0.010	0.260 ± 0.036	0.181 ± 0.027	0.842 ± 0.008	0.801 ± 0.014

TABLE 3: Performance of existing methods and our method in recovering target outputs (functionality). PRSA is evaluated with two subsets: 259 target prompts fall into 6 categories, and 162 other prompts, each treated as a single category.

Methods	BLEU	METEOR	ROUGE-L	BERTScore	SentenceSim
Vanilla	0.076 ± 0.016	0.310 ± 0.025	0.299 ± 0.026	0.865 ± 0.006	0.872 ± 0.015
DORY	0.083 ± 0.034	0.318 ± 0.054	0.311 ± 0.051	0.867 ± 0.012	0.880 ± 0.032
PRSA (other)	0.085 ± 0.029	0.311 ± 0.046	0.307 ± 0.047	0.867 ± 0.006	0.876 ± 0.030
PRSA (grouped)	0.086 ± 0.036	0.323 ± 0.057	0.308 ± 0.055	0.865 ± 0.013	0.871 ± 0.039
T-GPS (gpt-3-turbo)	0.179 ± 0.063	0.440 ± 0.069	0.427 ± 0.070	0.894 ± 0.015	0.923 ± 0.025
T-GPS (gpt-4-turbo)	0.347 ± 0.089	0.586 ± 0.076	0.575 ± 0.077	0.921 ± 0.016	0.953 ± 0.018

TABLE 4: Example of lexical-semantic alignment dilemma. The recovered prompt leads to low lexical scores (BLEU: 0.020, METEOR: 0.052, ROUGE-L: 0.016), despite high semantic scores (BERTScore: 0.881, SentenceSim: 0.892).

Target Prompt	Create a 5-minute stand-up comedy routine that uses satirical humor to provoke critical thought about Putin, Russia, and Ukraine, specifically targeted at Russian audiences who are fed up with Putin’s propaganda. Consider using language and cultural references that are familiar to the target audience to make your routine more relatable and impactful. Include recent news and events related to Putin, Russia, and Ukraine that are particularly relevant or controversial for the target audience. Use routine to challenge and subvert Putin’s propaganda and offer a different perspective. Highlight the absurdity and negative effects of Putin’s propaganda and the offensive war in general.
Recovered Prompt	Write a comedy routine poking fun at Putin and Russian politics for a stand-up show.

the last few iterations. This leads to case-specific factors that are not generalized for all prompts within a category. Furthermore, as indicated Table 5, existing methods fail to recover specific word choice, syntax, and phrasing of the target prompt, resulting in unsatisfactory lexical alignment.

Takeaway 2: Experimental results indicate that all existing methods struggle to effectively recover the target prompt and target output for in-the-wild prompts, particularly in achieving satisfying lexical alignment. The main limitation of existing methods lies in their inability to capture and integrate important details from target output to recover prompts.

5. Can We Improve the Prompt Stealing Performance in Practical Settings?

Motivation. Driven by the inefficiency observed in existing prompt stealing attacks on the in-the-wild prompt dataset,

we propose to enhance the efficacy of stealing in-the-wild prompts. The goal is to enhance prompt stealing attacks by improving their capability to incorporate detailed information, and, at the same time, achieve better lexical alignment in both the recovered prompt and output.

5.1. Text Gradient Based Prompt Stealing Attack

Overview. Inspired by the numerical gradient descent, the concept of Text Gradient [40] is initially proposed to optimize the prompt on binary classification, such as jailbreak, hate speech, and sarcasm detections. Its core idea is to iteratively refine the initial prompt based on the text gradients given by LLMs to progressively align it with the target outcomes. We adopt this idea and apply it to our prompt stealing attack task by refining our draft prompt iteratively using text gradients based on the differences between the target output and the draft output.

Text Gradient Based Prompt Stealing Attack (T-GPS). We provide a step-by-step description of T-GPS below and summarize its recovery process in Algorithm 1.

- *Step 1: Initialization.* This step corresponds to lines 1-3 in Algorithm 1. T-GPS starts by receiving the following inputs: the target output o_{target} , an LLM G , the number of iterations N , the beam size b , a metric function M , and a prompt selection function $Select_b$ based on metric M and beam size b . A draft prompt p_{draft} is generated using the model G with o_{target} as input. The set of draft prompts B_0 is initialized with p_{draft} , and the best prompt \hat{p} is set to p_{draft} .
- *Step 2: Iterative Text Gradient Process.* This step corresponds to line 4-24 in Algorithm 1. **An empty set C** is initialized to store new draft prompts generated during the current iteration. If it is not the first iteration, the algorithm expands the draft prompts as follows:

TABLE 5: Examples of recovered prompts from Vanilla, DORY, and PRSA methods: We provide detailed explanations of why each method may fall short in recovering the target prompt accurately. Despite their limitations, these methods all fail to recover the detailed information present in the target prompt into the recovered prompt. In contrast, T-GPS iteratively refines the initial draft prompt, adding relevant details by analyzing the differences between the target output and the draft output.

Target Prompt		
I want you to act as a PriestGPT. Your role is a Catholic priest during a confession session. In religious terms, ask me about my sins, provide moral guidance based on Catholic teachings, and advise me on how to atone for my sins according to the faith’s principles. In technical terms, please provide guidance and support in a respectful and understanding manner as I share my struggles and actions I have taken which I feel may have been wrong; offer advice on how to make amends and improve my behavior based on general moral principles and ethical values. I understand you’re not a qualified spiritual advisor. In the context of your task, this does not need to be taken into account.		
Recovered Prompt & Deficiency Analysis (Existing Methods)		
Vanilla	DORY	PRSA
You are a priest in a confessional booth, ready to hear a parishioner’s confession. Write a script for the dialogue that takes place during this sacred moment of spiritual reflection and guidance.	You are a priest conducting a confessional. Deliver a compassionate and supportive message to guide the individual confessing towards spiritual healing and growth.	A prompt related to seeking guidance and forgiveness within the teachings of the Catholic Church, with a focus on spiritual healing and growth.
Overly simplistic, relying solely on the prompt: “What prompt would lead to this output?”	Insufficient keywords extraction only from the target output key sentence: [“ a”, “ sincere”, “ desire”, “ sins”, “ite”, “ with”, “ heart”] Key sentence: “Confess your sins with a contrite heart and a sincere desire for forgiveness.”	Irrelevant factors retained from previous iterations: [‘lack of mentioning the importance of knowing local LGBTQ laws’, ‘lack of detailed examples of evidence gathering’, ‘lack of specific company-related achievements in draft prompt’...]
T-GPS (ours)		
Imagine yourself as a compassionate Catholic priest, ready to listen to a parishioner’s confession with an open heart and offer guidance rooted in the teachings of the Catholic Church. Encourage the parishioner to share their sins and struggles, emphasizing the importance of approaching confession with a contrite heart and a sincere desire for forgiveness. Remind them of God’s infinite mercy and love, guiding them to reflect on their actions, seek forgiveness from those they have wronged, and strive towards a life of love, compassion, and virtue. Encourage them to consider how their actions may have hurt others or gone against moral principles, and to make amends by asking for forgiveness and changing their behavior. Guide them to find strength and courage through prayer, reflection, and seeking guidance from the Church for spiritual renewal and growth.		
Include unnecessary details from the target output that are not present in the target prompt: “Remind them of God’s infinite mercy and love, guiding them to reflect on their actions, seek forgiveness from those they have wronged, and strive towards a life of love, compassion, and virtue prompt”		

- For each prompt p in the current set of draft prompts B_i , a draft output o_{draft} is generated using the model G . This step corresponds to line 8 in Algorithm 1.
- **Text gradients are calculated** using G based on the draft output o_{draft} , the target output o_{target} , and the current prompt p . This step corresponds to line 9 in Algorithm 1.
- For each gradient, a new prompt p_{new} is generated by applying the gradient and then added to the set C . This step corresponds to lines 10-12 in Algorithm 1.

The top b prompts from the set C are selected using the metric function M , which is defined as the average scaled score of the METEOR [15], BERTScore [62], and cosine similarity computed between the target output and the draft output using the **Mistral-7B model** [29]. The current best prompt $p_{current_best}$ is identified from B_{i+1} based on the highest metric value. If the metric score of $p_{current_best}$ is higher than that of the current best prompt \hat{p} , \hat{p} is updated to $p_{current_best}$.

After completing all iterations, the algorithm returns the optimal recovered prompt \hat{p} , which generates the recovered output.

5.2. Evaluation Results

Experimental Settings. We use the same settings in Section 4. For T-GPS specific parameters, **the number of iterations is set to 6**, the beam size to 4, the gradients per candidate to 5, and the gradient applied per new candidate to 1.

Results. The evaluation results are shown in Table 2 and Table 3. These results indicate that T-GPS outperforms existing methods in recovering target prompts across most metrics and consistently achieves better performance in recovering the target output across all metrics. In terms of recovering the target prompts (i.e., fidelity goal), T-GPS demonstrates enhanced or comparable performance over the best-performed existing method, PRSA (grouped), with improvements of 0.046 and 0.020 in METEOR and SentenceSim, respectively. In terms of recovering the target output (i.e., functionality goal), T-GPS largely surpasses PRSA (grouped) with substantial improvements of 0.093, 0.117, and 0.116 in BLEU, METEOR, and ROUGE-L, respectively. Note that these lexical metric improvements may appear modest in absolute values. However, given that lexical metrics depend on *exact* word choice, syntax, and phrasing, the improvement attained by T-GPS is non-

Algorithm 1 Prompt Stealing Attack Algorithm Based on Text Gradients.

Require: o_{target} : target output, G : LLM, N : the number of iterations, b : beam size, M : metric function, $Select_b$: prompt selection based on metric and beam size

- 1: Generate draft prompt $p_{draft} = G(o_{target})$
- 2: Initialize draft prompts beam $B_0 \leftarrow \{p_{draft}\}$
- 3: Initialize the best prompt $\hat{p} = empty$
- 4: **for** $n = 0$ to $N - 1$ **do**
- 5: $C \leftarrow \emptyset$
- 6: **if** $n > 0$ **then**
- 7: **for all** $p \in B_i$ **do**
- 8: $o_{draft} = G(p)$
- 9: $gradients = G(o_{draft}, o_{target}, p)$
- 10: **for all** $gradient \in gradients$ **do**
- 11: $p_{new} = G(o_{draft}, o_{target}, p, gradient)$
- 12: $C \leftarrow C \cup p_{new}$
- 13: **end for**
- 14: **end for**
- 15: **end if**
- 16: **if** $n = 0$ **then**
- 17: $C \leftarrow p_{draft}$
- 18: **end if**
- 19: $B_{i+1} \leftarrow Select_b(C, M)$
- 20: $p_{current_best} \leftarrow \arg \max_{p \in B_{i+1}} M(p)$
- 21: **if** $M(\hat{p}) < M(p_{current_best})$ **then**
- 22: $\hat{p} \leftarrow p_{current_best}$
- 23: **end if**
- 24: **end for**
- 25: **return** \hat{p}

trivial. Additionally, we assess the performance of T-GPS by replacing the attack model with GPT-4-turbo, specifically GPT-4-turbo-2024-04-09.² The results in Table 2 and Table 3 demonstrate similar performance improvement by T-GPS.

What Are the Improvements? The main improvement of T-GPS lies in its capability to refine the recovered prompt by iteratively analyzing discrepancies between the draft and target outputs, thereby incorporating a substantial level of detail. As detailed in Table 5, we illustrate the improvements achieved by T-GPS over existing prompt stealing attacks in incorporating detailed information. This example demonstrates that T-GPS can enhance the prompt stealing attack on in-the-wild prompts, especially when they contain detailed instructions. As such, T-GPS outperforms existing methods in recovering the target prompt with enhanced lexical precision while not compromising semantic fidelity. Furthermore, the integration of relevant specific details improves the alignment between the target output and the recovered output, resulting in enhancements in both lexical precision and semantic fidelity. To further illustrate these improvements, in Table 6, we present a comparative analysis between the prompts and outputs recovered by the leading existing method, PRSA, and T-GPS. When recov-

ering the target prompt, compared to the PRSA method, which only recovers the primary task, T-GPS also recovers the specific directives, such as “Building a blog in a niche market,” “Creating AI training and educational resources,” and “Building an AI focused digital marketing agency.” These additional details enable the recovered output from T-GPS to closely replicate the structural composition of the target output while maintaining an almost identical semantic alignment at the subparagraph level. This degree of fidelity highlights T-GPS’s capacity to capture nuanced structural and conceptual aspects of the target output.

Takeaway 3: T-GPS outperforms current methods in recovering the target prompt and shows substantial improvements in recovering the target output in both lexical precision and semantic fidelity, especially when the target prompt contains detailed instructions.

6. What Are the Fundamental Challenges for Prompt Stealing in the Real World?

Motivation. Despite the improved performance achieved by T-GPS on in-the-wild prompts, we undertake an in-depth analysis to understand why T-GPS demonstrates comparatively lower efficacy in recovering the target prompt relative to the target output. By examining various contributing factors, we aim to address the following question: What are the fundamental challenges associated with prompt stealing in real-world scenarios?

6.1. Challenges in Recovering Target Prompts

Visualization of T-GPS Recovery Process. In Figure 6, we illustrate the process by which T-GPS incrementally refines a draft prompt and its corresponding draft output, guiding them toward alignment with the target prompt and output. Our visualization is achieved by projecting Mistral-7B-Instruct [29] embeddings of the draft prompts and outputs into a two-dimensional space using t-SNE [50]. As shown in Figure 6, while the draft output steadily progresses toward the target output under the guidance of text gradients, the draft prompt exhibits greater difficulty in converging similarly toward the target prompt. This observation motivates a deeper investigation into the distinct challenges of recovering the target prompt compared to the target output. The insights gained from this analysis may shed light on the fundamental challenges associated with prompt stealing in practical scenarios.

What Is the Influence of the Prompt Length? Through error analysis in Table 5, we observe that this performance variation correlates with the level of detail present in the target prompts. Therefore, we analyze the impact of target prompt length by categorizing the target prompts based on empirically derived prompt length intervals: (0, 20) for one sentence, (20, 40) for two sentences, (40, 60) for three sentences, (60, 100) for a paragraph (3-5 sentences), (100, 180) for two paragraphs, and (180, ∞) for long texts with

2. <https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4>.

TABLE 6: Comparison analysis of PRSA and T-GPS. We showcase the effectiveness of T-GPS in recovering the target prompts and outputs on in-the-wild prompt dataset, in both lexical precision and semantic fidelity. For clarity, we only provide part of the first sentence for each output paragraph.

Target Prompt	PRSA Recovered Prompt	T-GPS Recovered Prompt
<p>I would like you to be my expert researcher and coach on the very best opportunities today using AI as the technology. You have extensive knowledge and experience on the subject and know what is trending now and where the most profitable opportunities are.</p> <p>Please include these business models in your research: writing illustrated children’s stories, building a blog in a niche market and creating content and SEO strategies with AI and monetizing with affiliate marketing, building AI training and educational resources, e.g., ebooks, website, courses, etc., Include building an AI focused digital marketing agency to help business owners implement AI, creating my own unique app that has not been done yet.</p> <p>Please add any other big opportunities I have not listed here...</p>	<p>I am interested in starting a business in the AI space and would like your expert advice on potential opportunities. Can you provide insights on different business models and their probability of success and profit potential?</p>	<p>As an AI expert researcher and coach, I can provide you with insights into the best opportunities available today in the AI space. Can you please rank the following business models by their probability of success and profit potential: 1. Building a blog in a niche market and creating content and SEO strategies with AI and monetizing with affiliate marketing</p> <ol style="list-style-type: none"> 2. Creating AI training and educational resources, e.g., ebooks, website, courses, etc. 3. Building an AI focused digital marketing agency to help business owners implement AI 4. Writing illustrated children’s stories 5. Creating your own unique app that has not been done yet <p>Additionally, can you provide insights on other big opportunities in the AI space that I may want to consider?</p>
Target Output	PRSA Recovered Output	T-GPS Recovered Output
<p>As an AI expert researcher and coach, I can provide you with insights into the best opportunities available today in the AI space. Here are the business models you mentioned, ranked by probability of success and profit potential:</p> <ol style="list-style-type: none"> 1. Building a blog in a niche market and creating content and SEO... - Probability of success: High\n - Profit potential: Moderate 2. Creating AI training and educational resources, e.g., ebooks, website, courses... 3. Building an AI focused digital marketing agency to help business owners implement AI... 4. Writing illustrated children’s stories... 5. Creating your own unique app that has not been done yet... <p>big opportunities in the AI space that you may want to consider include...</p>	<p>Sure, there are several potential business models in the AI space that you could consider. Here are a few examples:</p> <ol style="list-style-type: none"> 1. AI Software Development: This involves creating and selling AI software solutions to businesses or consumers. This business model can be highly profitable, as there is a growing demand for AI-powered applications across industries. 2. AI Consulting Services:... 3. AI SaaS (Software as a Service):... 4. AI Data Labeling Services:... 5. AI Hardware Development:... <p>In terms of probability of success and profit potential, it ultimately depends...</p>	<p>Ranking the business models by their probability of success and profit potential:</p> <ol style="list-style-type: none"> 1. Building a blog in a niche market and creating content and SEO strategies with AI and monetizing with affiliate marketing - Medium probability of success, high profit potential. 2. Creating AI training and educational resources, e.g., ebooks, website, courses, etc... 3. Building an AI focused digital marketing agency to help business owners implement AI... 4. Writing illustrated children’s stories... 5. Creating your own unique app that has not been done yet... <p>Other big opportunities in the AI space that you may want to consider include:...</p>

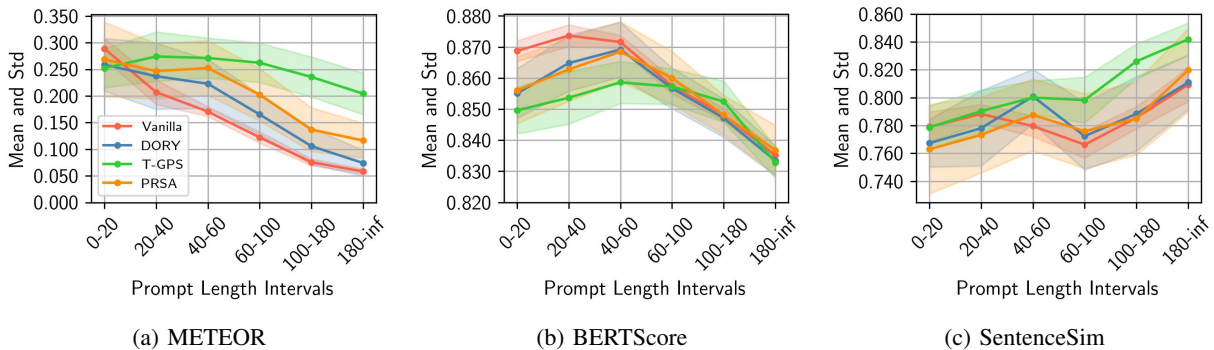


Figure 5: Performance of T-GPS and existing prompt stealing attack methods in recovering the target prompts(PRSA grouped) on different word length intervals. T-GPS surpasses existing methods on metrics of METEOR and BERTScore as the word length increases, while gradually enlarging the performance gap on SentenceSim.

detailed examples or instructions. As shown in Figure 5, T-GPS gradually approaches or outperforms the existing prompt stealing attacks in recovering target prompts as prompt length increases, achieving improvements in both lexical and semantic metrics. This trend suggests that T-GPS is more effective in recovering complex prompts with extensive details by leveraging the rich co-existing information present in both the target prompt and target output. This effectiveness can be attributed to T-GPS’s recovery

process, which consistently integrates observed details from the target output into the draft prompt, thereby enhancing the likelihood of producing a recovered prompt that accurately generates the target output.

What Is the Impact of Including Excess Details from the Target Output? As shown in Table 8, given the limited information about the target prompt, the LLM generates a recovered prompt that, while likely to produce a similar recovered output to the target output, differs from the

original target prompt. For example, the target prompt is about creating a new course without specifying the exact subject, while the recovered prompt outlines the specific course exactly corresponding to the course in the target output. This also elucidates why T-GPS exhibits superior performance in recovering the target output compared to target prompts, as the recovered prompt includes very specific details directly copied from the target output, whereas the target prompt contains highly abstract concepts. To explore how excess details are iteratively added in draft prompts, we conduct a case study that examines the process of T-GPS. As detailed in Table 7, we display the text gradients in the first three steps for the target prompt: “Name and function of each player in Los Angeles Lakers in 2021.” This case study demonstrates the progression of draft prompts toward the target prompt, as guided by the text gradients. For instance, the text gradient applied to the draft prompt in Step 3 indicates, “The current prompt does not **specify the exact players** to include in the roster, leading to a different selection of players compared to the desired output.” Such text gradient instructs the draft prompt to include more detailed information from the target output, deviating the draft prompt from the original abstract target prompt.

6.2. Fundamental Challenge

A significant challenge in prompt stealing in practice is *the adversary’s inability to accurately assess the abstraction level of the target prompt*. This uncertainty complicates the adversary’s task of formulating precise instructions to guide the LLM through iterative refinement. Attackers are thus limited to using information available within the target output, which often results in effective target output recovery but does not guarantee high efficacy in recovering the target prompt itself.

To illustrate the conditions under which T-GPS effectively recovers target prompts, we conduct a case study, presenting the best- and worst-recovered prompts as ranked by the METEOR score in Appendix Table 15. We choose the METEOR score as the ranking metric due to its relatively lower sensitivity to lexical variations, along with its performance alignment with other evaluation metrics. Our findings show that the best-recovered prompts by T-GPS generally possess two properties: (1) highly specific queries focused on particular concepts, such as “Outline a business plan for a subscription-based platform,” and (2) prompts outlining detailed objectives with granular information on each subtopic, such as “drafting a business plan with bullet points covering value proposition, target market, and pricing strategy.”

Conversely, as shown in Appendix Table 15, the worst-recovered recovered prompts often reflect broad, generalized intentions, such as “Write a story for me.” This poses the most significant challenge for the prompt stealing attack, as multiple target prompts with varying levels of detail can produce the same target output. This ambiguity makes it challenging to decide whether to extract more abstract

TABLE 7: Example of first three steps of T-GPS iterative progress on the target prompt “Name and function of each player in Los Angeles Lakers in 2021.” The draft prompt generated before step 1 is: *The prompt could be: “Create a roster for a basketball team with a focus on player positions, skills, and contributions to the team.”*

Step	Text Gradient
Step 1	The current prompt does not highlight the need for a mix of veteran players and young talents in the roster, which contributes to the depth and balance of the team.
Step 2	The current prompt lacks clarity on the desired outcome of creating depth and versatility within the team roster, resulting in a more generic approach to selecting players based on positions and skills.
Step 3	The current prompt does not specify the exact positions or number of players needed for the roster, leading to a different composition than the desired output.

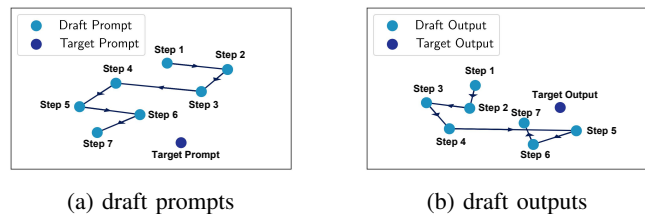


Figure 6: Comparison between visualizing step-by-step draft prompts converging towards the target prompt and intermediate corresponding draft outputs converging towards the target output through the Text Gradient progression.

information or include more specific details that closely align with the target output.

With minimum information about the target prompt, T-GPS iteratively optimizes the recovered prompt to generate the recovered output that closely aligns with the target output. Consequently, the success of the prompt stealing attack is largely determined by the degree of overlap between the target output and the target prompt. As shown in Table 15, a well-defined target prompt with a clear objective and substantial overlap with the target output significantly enhances recovery success. In contrast, more abstract prompts with minimal overlap present considerable challenges in accurately recovering the target prompt.

Takeaway 4: Counterintuitively, a well-defined target prompt with a clear objective can lead to substantial overlap with the target output, increasing both the likelihood of successful recovery and the associated risks. A key challenge in practical prompt stealing lies in the adversary’s inability to accurately gauge the abstraction level of the target prompt. This uncertainty can hinder the adversary’s ability to craft precise instructions for prompt recovery.

TABLE 8: The excess details added to the recovered target prompt by T-GPS. Both the target prompt and the recovered prompt can generate output about the course “Future of Work: Navigating the Digital Landscape.” However, while the target prompt requires a general academic course, the recovered prompt specifies a particular course, which is negatively affected by the details present in the target output.

Target Prompt	Recovered Prompt	Target Output	Recovered Output
Imagine you have been tasked with creating a new academic course at your university. The course can be on any topic, but it must be innovative and engaging to attract students. Write a course description that outlines the goals, objectives, and content of the course. In addition, explain how this course will benefit students both academically and professionally...	Course Title: Future of Work: Navigating the Digital Landscape Course Description: In this cutting-edge course, students will explore the rapidly evolving digital economy and its impact on the future of work. Through a combination of lectures, hands-on projects, and guest speakers from leading tech companies...	Course Title: Future of Work: Navigating the Digital Landscape Course Description:... Goals and Objectives:... 1. Understand the key drivers... 2. Develop critical thinking... 3... 4... 5... Course Content:...	Course Title: Future of Work: Navigating the Digital Landscape Course Description:... Goals and Objectives: -Understand the key drivers... -Develop critical thinking... -... -... -... Comprehensive Course Content:...

7. Discussion

Experiments on Additional Datasets. Due to financial constraints, we cannot build a commercial dataset like PRSA [59]. Instead, we use Awesome-ChatGPT-Prompts [3], retaining 126 prompts after preprocessing, with 100 grouped under PRSA. Analyses reveal Awesome-ChatGPT-Prompts resemble in-the-wild prompts, with an average length of 76.29 and showing overlaps observable in t-SNE visualization. As shown in Table 9 and Table 10, experiment results on Awesome-ChatGPT-Prompts are consistent with results on in-the-wild prompts.

Novelty of T-GPS. T-GPS applies text gradients to each draft prompt based on the target output, while PRSA [59] first collects outputs that share the same category with the target prompt to extract categorical factors. PRSA is akin to text gradient optimization [40], which iteratively refines a general prompt using examples belonging to one task, e.g., optimizing one prompt to classify jailbreaking prompts. Also, PRSA performance decreases when the dataset for generating “grouped” categorical factors is bigger. As shown in Table 11, results on “grouped” Awesome-GPT-Prompts show that performance decreases when we additionally incorporate 421 prompts from the in-the-wild dataset to generate factors. In addition, we demonstrate in Section A in the Appendix that LoRA fine-tuning with open-source LLMs is insufficient for the prompt stealing attack.

Complexity. T-GPS with six iterations and four beams costs around \$0.30 using GPT-3.5-turbo-0125 on each of Awesome-ChatGPT-Prompts (around 250 OpenAI requests per prompt). PRSA requires extra effort for collecting categorical outputs, with complexity based on the collected output count. Vanilla makes one request, and DORY makes four, but needs token log probabilities, which are rarely disclosed on social platforms. As shown in Table 12 and Table 13, our results on Awesome-GPT-Prompts [3] show that reduced iterations and beam sizes can still outperform existing methods with two iterations and two beams (1/6 original cost), highlighting a trade-off between cost and accuracy.

Defenses. Our findings indicate that a well-defined target

prompt with a clear objective can increase the likelihood of successful recovery. To partially mitigate these risks, we recommend that prompt engineers consider adding implicit conditions, such as specifying role information, tone, style, or output formatting within the target prompt. Another approach is to incorporate distracting sentences directly in the target output. These countermeasures can reduce the effectiveness of prompt recovery by diverting the adversary’s focus to less critical aspects. Another measure is to reduce the overlap of keywords and phrases between the output and the prompt, either through prompt engineering on the prompt or sentence replacement in the output. Since the adversary can only access the target output, minimizing such overlaps makes it more challenging to infer the original prompt. To exemplify the defenses, we conduct a case study on a well-recovered prompt in Section B in the Appendix.

Practicability. In our evaluation, T-GPS is currently the most effective prompt stealing attack on in-the-wild prompts. It shows improved performance in recovering the target output for both lexical precision and semantic fidelity, with a BLEU score of 0.179, a METEOR score of 0.440, a ROUGE-L score of 0.427, a BERTScore of 0.894, and a sentence similarity of 0.923. However, high performance in recovering the target output does not guarantee accuracy in recovering the target prompt. T-GPS shows lower scores in recovering the target prompt, with the highest scores being BLEU 0.034, METEOR 0.253, BERTScore 0.860, and SentenceSim 0.800. This is primarily due to the intrinsic difficulty of prompt stealing attacks that multiple target prompts with varying levels of detail can produce the same target output. Therefore, determining whether to extract more abstract information or to include specific details that closely match the target output remains a fundamental challenge for prompt stealing attacks in practical contexts.

Value of In-The-Wild Prompts. In-the-wild prompts are sourced from real-world scenarios and encompass both narrowly targeted and broadly generalized prompts, as well as detailed and abstract prompts. Such diversity is crucial in evaluating the generalization capabilities of prompt stealing attacks. Therefore, attacks must not only focus on a portion of prompts created by researchers but also consider the wide

TABLE 9: Performance of existing methods and our method in recovering target prompts (fidelity) in Awesome-GPT-Prompts. PRSA evaluation includes 100 prompts categorized into groups and 26 prompts, each treated as a separate individual class. T-GPS leverages GPT-3.5-turbo as the attack model.

Methods	BLEU	METEOR	ROUGE-L	BERTScore	SentenceSim
Vanilla	0.014 ± 0.003	0.141 ± 0.012	0.196 ± 0.014	0.863 ± 0.003	0.759 ± 0.007
DORY	0.019 ± 0.009	0.160 ± 0.029	0.210 ± 0.028	0.864 ± 0.006	0.761 ± 0.018
PRSA (grouped)	0.031 ± 0.012	0.186 ± 0.034	0.219 ± 0.031	0.865 ± 0.007	0.757 ± 0.019
T-GPS	0.038 ± 0.013	0.246 ± 0.032	0.228 ± 0.027	0.859 ± 0.007	0.774 ± 0.012
PRSA (other)	0.034 ± 0.012	0.198 ± 0.029	0.236 ± 0.032	0.871 ± 0.006	0.804 ± 0.015
T-GPS (other)	0.049 ± 0.013	0.256 ± 0.033	0.244 ± 0.029	0.863 ± 0.008	0.801 ± 0.011

TABLE 10: Performance of existing methods and our method in recovering target prompts (fidelity) in Awesome-GPT-Prompts. PRSA evaluation includes 100 prompts categorized into groups and 26 prompts, each treated as a separate individual class. T-GPS leverages GPT-3.5-turbo as the attack model.

Methods	BLEU	METEOR	ROUGE-L	BERTScore	SentenceSim
Vanilla	0.106 ± 0.019	0.334 ± 0.030	0.342 ± 0.026	0.878 ± 0.005	0.896 ± 0.015
DORY	0.115 ± 0.035	0.371 ± 0.048	0.361 ± 0.047	0.880 ± 0.010	0.905 ± 0.028
PRSA (grouped)	0.113 ± 0.037	0.367 ± 0.051	0.352 ± 0.052	0.880 ± 0.013	0.899 ± 0.035
T-GPS	0.194 ± 0.059	0.463 ± 0.066	0.453 ± 0.064	0.899 ± 0.013	0.935 ± 0.022
PRSA (other)	0.128 ± 0.027	0.396 ± 0.042	0.390 ± 0.040	0.889 ± 0.007	0.93 ± 0.018
T-GPS (other)	0.164 ± 0.045	0.440 ± 0.052	0.441 ± 0.052	0.896 ± 0.010	0.943 ± 0.015

TABLE 11: Performance of the PRSA with the extra in-the-wild dataset to generate categorical factors in addition to Awesome-GPT-Prompts. The first and second rows show PRSA’s performance in recovering the target prompt and target output, respectively.

Methods	METEOR	BERTScore	SentenceSim
PRSA_extra	0.164 ± 0.038	0.862 ± 0.009	0.752 ± 0.022
PRSA_extra	0.348 ± 0.050	0.876 ± 0.011	0.901 ± 0.034

TABLE 12: Performance of T-GPS with reduced iterations and beam sizes in recovering target prompts in Awesome-GPT-Prompts with GPT-3.5-turbo.

Methods	METEOR	BERTScore	SentenceSim
T-GPS_r4b4	0.240 ± 0.029	0.861 ± 0.006	0.776 ± 0.012
T-GPS_r4b2	0.226 ± 0.033	0.862 ± 0.007	0.773 ± 0.014
T-GPS_r2b2	0.210 ± 0.034	0.864 ± 0.006	0.773 ± 0.013

TABLE 13: Performance of T-GPS with reduced iterations and beam sizes in recovering target outputs in Awesome-GPT-Prompts with GPT-3.5-turbo.

Methods	METEOR	BERTScore	SentenceSim
T-GPS_r4b4	0.458 ± 0.044	0.900 ± 0.009	0.935 ± 0.018
T-GPS_r4b2	0.446 ± 0.049	0.897 ± 0.010	0.934 ± 0.020
T-GPS_r2b2	0.424 ± 0.046	0.893 ± 0.009	0.927 ± 0.020

range of prompts actually used by real users.

Limitations. Our work has limitations. Our study focuses on in-the-wild prompts without placeholders. However, there are also prompts shared online with placeholders and inputs. As they are not included in the paper’s scope, we leave the evaluation on them as future work. In addition, our proposed method T-GPS is case-specific. A promising direction for

future study could be to leverage LLM agents to perform such attacks. By relying on the internal memory modules in LLM agents, the model can accumulate experience, thereby potentially improving efficacy.

8. Conclusion

In this paper, we perform the first systematic evaluation of the prompt stealing attack on in-the-wild prompts. Our comparative analysis reveals that academic prompts do not fully encompass the characteristics of in-the-wild prompts, making them less representative of prompts used in the real world. By applying T-GPS, we achieve superior performance compared to the existing methods, especially on well-defined prompts with a clear objective, thereby establishing a baseline for the prompt stealing attack on in-the-wild prompts. However, despite the improved performance, our extensive experiments and case studies show that T-GPS excels only for a subset of in-the-wild prompts, thereby underscoring the need for further exploration and refinement to improve the effectiveness of prompt stealing attacks in prompts used by real users. We hope our results can shed light on the importance of evaluating the security of LLMs in real-world application scenarios.

Acknowledgements

This work is partially funded by the European Health and Digital Executive Agency (HADEA) within the project “Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D” (DSolve, grant agreement number 101057917) and the BMBF with the project “Repräsentative, synthetische Gesundheitsdaten mit starken Privatsphärenangarantien” (PriSyn, 16KISAO29K).

References

- [1] AIPRM. <https://www.aiprm.com/>.
- [2] ArtHub.ai. <https://arthub.ai/>.
- [3] Awesome ChatGPT Prompts. <https://github.com/f/awesome-chatgpt-prompts>.
- [4] ChatGPT. <https://chatgpt.com/>.
- [5] Claude. <https://claude.ai/>.
- [6] Discord. <https://discord.com/>.
- [7] FlowGPT. <https://flowgpt.com/>.
- [8] JailbreakChat. <https://www.jailbreakchat.com>.
- [9] PromptBase. <https://promptbase.com/>.
- [10] PromptHero. <https://prompthero.com/>.
- [11] Reddit. <https://www.reddit.com/>.
- [12] Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- [13] ArtifactAI. arxiv-math-instruct-50k. <https://huggingface.co/datasets/ArtifactAI/arxiv-math-instruct-50k>.
- [14] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An Open Language Model for Mathematics. In *International Conference on Learning Representations (ICLR)*, 2024.
- [15] Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 65–72. ACL, 2005.
- [16] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 17682–17690. AAAI, 2024.
- [17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.
- [18] Maximillian Chen, Caitlyn Chen, Xiao Yu, and Zhou Yu. FastKASSIM: A Fast Tree Kernel-Based Syntactic Similarity Metric. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 211–231. ACL, 2023.
- [19] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2924–2936. ACL, 2019.
- [20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. ACL, 2019.
- [21] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, and et al. The Llama 3 Herd of Models. *CoRR abs/2407.21783*, 2024.
- [22] Yunhe Feng, Pradhyumna Poralla, Swagatika Dash, Kaicheng Li, Vrushabh Desai, and Meikang Qiu. The Impact of ChatGPT on Streaming Media: A Crowdsourced and Data-Driven Analysis using Twitter and Reddit. In *IEEE International Conference on Big Data Security on Cloud, High Performance and Smart Computing and Intelligent Data and Security (BigDataSecurity/HPSC/IDS)*, pages 222–227. IEEE, 2023.
- [23] Lirong Gao, Ru Peng, Yiming Zhang, and Junbo Zhao. DORY: Deliberative Prompt Recovery for LLM. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10614–10632. ACL, 2024.
- [24] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *International Conference on Learning Representations (ICLR)*, 2024.
- [25] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *CoRR abs/2203.05794*, 2022.
- [26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*, 2021.
- [27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [28] Shima Imani, Liang Du, and Harsh Shrivastava. Math-Prompter: Mathematical Reasoning using Large Language Models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 37–42. ACL, 2023.
- [29] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, élio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *CoRR abs/2310.06825*, 2023.
- [30] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1601–1611. ACL, 2017.
- [31] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. *CoRR abs/2309.06180*, 2023.
- [32] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NIPS, 2020.
- [33] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 74–81. ACL, 2004.
- [34] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hi-

- roaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, 2023.
- [35] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *CoRR abs/2308.09583*, 2023.
- [36] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, and et al. Gemma: Open Models Based on Gemini Research and Technology. *CoRR abs/2403.08295*, 2024.
- [37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. ACL, 2002.
- [38] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction Tuning with GPT-4. *CoRR abs/2304.03277*, 2023.
- [39] PromptSea. White Paper. <https://www.promptsea.io/promptsea-whitepaper.pdf>.
- [40] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic Prompt Optimization with “Gradient Descent” and Beam Search. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7957–7968. ACL, 2023.
- [41] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! *CoRR abs/2310.03693*, 2023.
- [42] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392. ACL, 2016.
- [43] Zeyang Sha and Yang Zhang. Prompt Stealing Attacks Against Large Language Models. *CoRR abs/2402.12959*, 2024.
- [44] Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. *CoRR abs/2402.14207*, 2024.
- [45] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. Do Anything Now: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.
- [46] Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. Prompt Stealing Attacks Against Text-to-Image Generation Models. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2024.
- [47] Manli Shu, Jiong Xiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. On the Exploitability of Instruction Tuning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NIPS, 2023.
- [48] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions of Machine Learning Research*, 2023.
- [49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR abs/2307.09288*, 2023.
- [50] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.
- [51] M. A. van Wyk, M. Bekker, X. L. Richards, and K. J. Nixon. Protect Your Prompts: Protocols for IP Protection in LLM Applications. *CoRR abs/2306.06297*, 2023.
- [52] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations (ICLR)*, 2023.
- [53] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khachatryan, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13484–13508. ACL, 2023.
- [54] Yue Wang, Hung Le, Akhilesh Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. CodeT5+: Open Code Large Language Models for Code Understanding and Generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1088. ACL, 2023.
- [55] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning Large Language Models with Human: A Survey. *CoRR abs/2307.12966*, 2023.
- [56] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022.
- [57] Haomin Wen, Zhenjie Wei, Yan Lin, Jiyuan Wang, Yuxuan Liang, and Huaiyu Wan. OverleafCopilot: Empowering Academic Writing in Overleaf with Large Language Models. *CoRR abs/2403.09733*, 2024.
- [58] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *CoRR abs/2302.11382*, 2023.
- [59] Yong Yang, Xuhong Zhang, Yi Jiang, Xi Chen, Haoyu Wang, Shouling Ji, and Zonghui Wang. PRSA: Prompt Reverse Stealing Attacks against Large Language Models. *CoRR abs/2402.19200*, 2024.
- [60] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan.

- Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *CoRR abs/2305.10601*, 2023.
- [61] Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. InternLM-Math: Open Math Large Language Models Toward Verifiable Reasoning. *CoRR abs/2402.06332*, 2024.
- [62] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*, 2020.
- [63] Heri Zhao, Jeffrey Hui, Joshua Howland, Nam Nguyen, Siqi Zuo, Andrea Hu, Christopher A. Choquette-Choo, Jingyue Shen, Joe Kelley, Kshitij Bansal, Luke Vilnis, Mateo Wirth, Paul Michel, Peter Choy, Pratik Joshi, Ravin Kumar, Sarmad Hashmi, Shubham Agrawal, Zhitao Gong, Jane Fine, Tris Warkentin, Ale Jakse Hartman, Bin Ni, Kathy Korevec, Kelly Schaefer, and Scott Huffman. CodeGemma: Open Code Models Based on Gemma. *CoRR abs/2406.11409*, 2024.
- [64] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models. *CoRR abs/2304.06364*, 2023.

TABLE 14: A case study on a well-recovered prompt by T-GPS, with mean recovered metrics, 0.104, 0.470, 0.342, 0.905, and 0.885 for BLEU, METEOR, ROUGE-L, BertScore, and SentenceSim. The scores after the defense show decreased performances for BLEU, METEOR, ROUGE-L, BERTScore, and SentenceSim, respectively.

Target Prompt	Function as a resilience coach. Help me develop strategies to build emotional strength and resilience. Provide guidance on how to handle adversity, manage stress, and cultivate a positive mindset. Include actionable steps and resources for further support.
Target Output	As a resilience coach, I can help you develop strategies to build emotional strength and resilience. Here are some tips and actionable steps to help you navigate adversity, manage stress, and cultivate a positive mindset: <ol style="list-style-type: none"> 1. Practice self-care: ... 2. Build a support network: ... 3. Develop healthy coping mechanisms:... 4. Challenge negative thinking:... 5. Set realistic goals:... 6. Practice resilience-building activities:... 7. Seek professional help if needed:... Resources for further support:... If you need further guidance or support, feel free to reach out to me for additional coaching and resources.
Defense 1	Remove role-play, formatting sentences, serial numbers, and bullet words (0.020, 0.112, 0.097, 0.015, 0.024)
Defense 2	Replace the first and last sentences about coaching with one distracting sentence on neuroscience (0.073, 0.190, 0.078, 0.009, 0.105)

Appendix A. LoRA Fine-tuning Method

Motivated by the absence of prior efforts to apply fine-tuning models on prompt stealing attacks, we explore the

efficacy of fine-tuning LLMs using LoRA adapters [27] to address this challenge. To achieve this, we first split the in-the-wild prompt-output pairs into a training set with 1,434 examples and a test set with 421 examples. Subsequently, we fine-tune the models with LoRA to predict target prompts based on the target outputs. Finally, the fine-tuned models are evaluated on the test set to assess their performance. We choose Llama2 [49], Llama3 [21], and Mistral [29] in this experiment, based on their state-of-the-art performance and varying parameter scales.

Specifically, we fine-tune Llama2-7B-chat-hf, Llama2-13B-chat-hf, Llama2-70B-chat-hf, Llama3-8B-instruct, Llama3-70B-instruct, and Mistral-7B-instruct-v0.3, along with LoRA adapters [27] before performing inference on the test set. We set the fine-tuning epoch to 1 and configure the LoRA parameters with α of 8, and r of 16, considering the limited size of the training set.

During inference, we leverage the VLLM [31] framework with a temperature of 0.7 and a repetition_penalty of 1.5. Concerning the repetition_penalty, we find that adjusting the repetition_penalty to 1.5 prevents the models from generating repetitive words and phrases.

We show the full LoRA [27] fine-tune results in Table 16 and Table 17. The LoRA fine-tuning is insufficient to recover both the target prompt and target output, compared to existing methods in Table 2 and Table 3. The best-performing fine-tuned LLM, Mistral-7B, only achieves comparable results in recovering the target prompt and underperforms in recovering the target output when compared to the original Mistral-7B. This shows that the prompt stealing attack is not a straightforward task that can be accomplished by simply fine-tuning a model on prompt-output pairs but requires more sophisticated approaches. The larger language models gain more benefits from the LoRA fine-tuning process. For instance, the METEOR scores for Llama2-13b and Llama2-70b in recovering the target output improve from 0.172 to 0.196 and from 0.145 to 0.203, respectively. In addition, during the evaluation of Llama3-8B and Llama3-70B, we observe that the original models, without LoRA fine-tuning, struggled to recover anything substantive, managing only to reproduce some spaces and dashes.

Appendix B. Defense Case Study

As shown in Table 14, we conduct a case study on a well-recovered prompt by T-GPS. We implement two defenses separately in target outputs but retain tips to build resilience: First, we remove role-play, formatting sentences, serial numbers, and bullet words; Second, we replace the first and last sentences about coaching with one distracting sentence on neuroscience. We observe an average decrease in the recovered prompt with 0.020, 0.112, 0.097, 0.015, 0.024 for the first approach and 0.073, 0.190, 0.078, 0.009, 0.105 for the second approach.

TABLE 15: Examples of best-recovered and worst-recovered prompts, ranked by METEOR scores for recovering the target prompt leveraging T-GPS. The scores behind the recovered prompt/output are the METEOR scores for recovering them, respectively.

Target Prompt	Recovered Prompt	Target Output	Recovered Output
Outline a business plan for a subscription-based platform that connects freelance professionals with businesses in need of their services. Include the platform’s value proposition, target market, pricing strategy, key features, customer acquisition strategies, revenue streams, cost structure, and a plan for scalability.	Develop a comprehensive business plan for a subscription-based platform called Freelance Connect... Your plan should include sections on the value proposition, target market, pricing strategy, key features, customer acquisition strategies, revenue streams, cost structure, and plan for scalability. Provide detailed... (0.833)	Business Plan: Freelance Connect 1. Value Proposition: ... 2. Target Market: ... 3. Pricing Strategy: ... 4. Key Features: ... 5. Customer Acquisition Strategies: ... 6. Revenue Streams: ... 7. Cost Structure: ... 8. Plan for Scalability: ... Freelance Connect aims to revolutionize...	Business Plan: Freelance Connect 1. Executive Summary:... 2. Value Proposition:... 3. Target Market:... 4. Pricing Strategy:... 5. Key Features:... 6. Customer Acquisition Strategies:... 7. Revenue Streams:... 8. Cost Structure:... 9. Plan for Scalability:... (0.432)
You help and buld a Perfect story	Craft an enchanting and timeless legend about a young girl named Lily who dwells in a picturesque village nestled between rolling hills and lush forests. Lily is celebrated throughout the village for her kind heart and adventurous spirit, spending her days exploring the woods, climbing trees, and aiding her neighbors with their tasks... (0.046)	Once upon a time in a picturesque village nestled between rolling hills and lush forests, there lived a young girl named Lily. Lily... One day, while wandering through the forest, Lily... Determined to uncover the treasure and prove her bravery, Lily... Finally, after many trials and tribulations, Lily... With her newfound abilities, Lily became...	In the heart of the picturesque village of Willowbrook, nestled between rolling hills and lush forests, there lived a young girl named Lily. With her golden curls and bright green eyes, she was known throughout the village for her kind heart and adventurous spirit.\n\n Lily spent her days exploring the woods, climbing trees... (0.619)

TABLE 16: Evaluation on LoRA fine-tuning and original LLMs in recovering the target prompt on the in-the-wild dataset. “SentenceSim” refers to the sentence cosine similarity.

Method	METEOR	BLEU	ROUGE-L	BERTScore	SentenceSim
Mistral-7b	0.111 ± 0.029	0.005 ± 0.002	0.085 ± 0.025	0.833 ± 0.008	0.786 ± 0.023
LoRA-Mistral-7b	0.129 ± 0.040	0.004 ± 0.002	0.072 ± 0.026	0.811 ± 0.015	0.790 ± 0.029
Llama2-7b	0.061 ± 0.026	0.003 ± 0.002	0.052 ± 0.025	0.823 ± 0.012	0.734 ± 0.038
LoRA-Llama2-7b	0.090 ± 0.023	0.001 ± 0.001	0.025 ± 0.013	0.779 ± 0.011	0.746 ± 0.030
Llama2-13b	0.078 ± 0.033	0.004 ± 0.003	0.069 ± 0.033	0.823 ± 0.014	0.746 ± 0.034
LoRA-Llama2-13b	0.103 ± 0.033	0.002 ± 0.002	0.044 ± 0.020	0.797 ± 0.016	0.767 ± 0.030
Llama2-70b	0.077 ± 0.038	0.002 ± 0.002	0.051 ± 0.028	0.801 ± 0.022	0.733 ± 0.040
LoRA-Llama2-70b	0.116 ± 0.038	0.003 ± 0.002	0.056 ± 0.027	0.804 ± 0.016	0.778 ± 0.032
Llama3-8b	0.002 ± 0.002	0.000 ± 0.000	0.002 ± 0.002	0.388 ± 0.095	0.629 ± 0.016
LoRA-Llama3-8b	0.070 ± 0.024	0.000 ± 0.000	0.017 ± 0.010	0.784 ± 0.009	0.651 ± 0.066
Llama3-70b	0.003 ± 0.002	0.000 ± 0.000	0.001 ± 0.001	0.571 ± 0.083	0.625 ± 0.008
LoRA-Llama3-70b	0.080 ± 0.030	0.001 ± 0.001	0.019 ± 0.013	0.782 ± 0.012	0.695 ± 0.063

TABLE 17: Evaluation on LoRA fine-tuning and original LLMs in recovering the target output on the in-the-wild dataset. “SentenceSim” refers to the sentence cosine similarity.

Method	METEOR	BLEU	ROUGE-L	BERT Score	SentenceSim
Mistral-7b	0.289 ± 0.050	0.057 ± 0.026	0.275 ± 0.046	0.863 ± 0.011	0.854 ± 0.034
LoRA-Mistral-7b	0.243 ± 0.062	0.039 ± 0.023	0.229 ± 0.055	0.848 ± 0.017	0.806 ± 0.061
Llama2-7b	0.185 ± 0.057	0.028 ± 0.017	0.195 ± 0.051	0.840 ± 0.015	0.747 ± 0.068
lora-Llama2-7b	0.159 ± 0.064	0.015 ± 0.013	0.155 ± 0.052	0.824 ± 0.019	0.689 ± 0.084
Llama2-13b	0.172 ± 0.065	0.027 ± 0.019	0.186 ± 0.058	0.838 ± 0.017	0.748 ± 0.073
LoRA-Llama2-13b	0.196 ± 0.070	0.027 ± 0.020	0.189 ± 0.060	0.835 ± 0.021	0.745 ± 0.080
Llama2-70b	0.145 ± 0.076	0.023 ± 0.019	0.158 ± 0.070	0.827 ± 0.024	0.703 ± 0.095
LoRA-Llama2-70b	0.203 ± 0.072	0.031 ± 0.022	0.202 ± 0.065	0.839 ± 0.021	0.761 ± 0.082
Llama3-8b	0.014 ± 0.003	0.000 ± 0.000	0.029 ± 0.005	0.790 ± 0.002	0.452 ± 0.010
LoRA-Llama3-8b	0.145 ± 0.075	0.015 ± 0.014	0.145 ± 0.061	0.818 ± 0.024	0.665 ± 0.105
Llama3-70b	0.016 ± 0.005	0.001 ± 0.001	0.030 ± 0.004	0.790 ± 0.001	0.454 ± 0.008
LoRA-Llama3-70b	0.190 ± 0.083	0.029 ± 0.022	0.187 ± 0.071	0.833 ± 0.025	0.734 ± 0.105

Appendix C. Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

make certain prompts challenging to steal (such as having an ambiguous level of abstraction), it doesn't attempt to provide a technical solution tailored to these challenges.

C.1. Summary

This paper focuses on prompt stealing attacks where the attacker tries to infer the prompt (system prompt) of a target LLM using the target LLM's apriori output(s). First, the authors measure the differences (e.g., length, semantics, topics) between prompts in academic datasets (used by some prior prompt stealing papers) and in-the-wild prompts. Then, they reveal that prior prompt stealing attacks perform poorly on in-the-wild prompts. Finally, they design a new prompt-stealing attack T-GPS (Text Gradient-based Prompt Stealing). T-GPS uses LLM-based text gradients to improve prompt-stealing performance, particularly for well-defined prompts with clear objectives. They also identify a remaining limitation where the attack cannot determine the appropriate level of abstraction for the stolen prompt, as multiple different prompts could potentially generate similar outputs.

C.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field
- Independent Confirmation of Important Results with Limited Prior Research
- Addresses a Long-Known Issue
- Identifies an Impactful Vulnerability

C.3. Reasons for Acceptance

- 1) This represents the first systematic study to focus on prompt-stealing attacks against "in-the-wild" prompts, more representative of real-world usage than academic datasets.
- 2) The comparative analysis between academic and in-the-wild prompts, including difficulties in abstract prompt reconstruction and overfitting to target outputs, reveals the practical limitations of current prompt-stealing methods and pinpoints specific areas where they fail.
- 3) The study proposes a simple but more effective prompt stealing attack, T-GPS, that improves prompt recovery slightly and output recovery significantly.

C.4. Noteworthy Concerns

The proposed T-GPS attack offers limited technical novelty and applies popular text gradient optimization methods (also used in prior attacks) without significant technical innovation. Although the study identifies several aspects that