Anti-adversarial Learning: Desensitizing Prompts for Large Language Models

Anonymous ACL submission

Abstract

With the widespread use of LLMs, preserving privacy in user prompts has become crucial, as prompts risk exposing privacy and sensitive data to the cloud LLMs. Traditional techniques like homomorphic encryption, secure multi-party computation, and federated learning face challenges due to heavy computational costs and user participation requirements, limiting their applicability in LLM scenarios. In this paper, we propose PromptObfus, a novel method for desensitizing LLM prompts. The core idea of PromptObfus is "anti-adversarial" learning, which perturbs privacy words in the prompt to obscure sensitive information while retaining the stability of model predictions. Specifically, PromptObfus frames prompt desensitization as a masked language modeling task, replacing privacy-sensitive terms with a `[MASK]` token. A desensitization model is trained to generate candidate replacements for each masked position. These candidates are subsequently selected based on gradient feedback from a surrogate model, ensuring minimal disruption to the task output. We demonstrate the effectiveness of our approach on three NLP tasks. Results show that PromptObfus effectively prevents privacy inference from remote LLMs while preserving task performance.

1 Introduction

The widespread adoption of large language models (LLMs) such as ChatGPT in various NLP tasks (Hong et al., 2024; Carlini et al., 2019) has raised significant concerns regarding their inherent privacy risks. Due to the substantial computational resources required for local deployment, users often rely on cloud APIs provided by model vendors, which introduces potential vulnerabilities. Specifically, user-submitted prompts, the primary medium of interaction with LLMs, may inadvertently expose sensitive information, posing serious privacy threats.

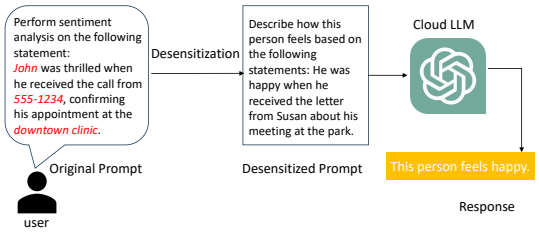


Figure 1: Illustration of prompt desensitization.

Prompts frequently contain personally identifiable information (PII), such as names, gender, occupation, and addresses, as illustrated in Figure 1. Without adequate safeguards during model processing, such data risks being exploited by malicious actors, potentially resulting in severe privacy violations (Hong et al., 2024). Consequently, ensuring robust privacy protection for user prompts has emerged as a critical and pressing challenge in the deployment of LLMs.

Traditional privacy-preserving techniques, such as Homomorphic Encryption (HE) (Gentry, 2009), Secure Multi-Party Computation (MPC) (Yao, 1982), and Federated Learning (FL) (McMahan et al., 2017), exhibit significant limitations when applied to prompts for LLMs, particularly in black-box settings where access to the model's internal architecture or training data is restricted. These methods often fail to simultaneously address the competing requirements of real-time performance, computational efficiency, and robust privacy protection.

Text obfuscation has emerged as a prevalent approach to safeguarding sensitive information in prompts (Miranda et al., 2025). For instance, techniques include injecting noise into word embeddings based on differential privacy to perturb sensitive data (Yue et al., 2021; Gao et al., 2024), clustering word vectors to render representations of sensitive terms indistinguishable (Zhou et al., 2023), and training models for data anonymization by detecting and removing PII entities (Chen et al., 2023).

However, these methods often struggle to achieve an optimal trade-off between privacy preservation and task utility (Zhang et al., 2024). Furthermore, approaches that rely on model training typically necessitate expert-annotated datasets, which are challenging to procure in practical applications.

In this paper, we propose PromptObfus, a portable and task-flexible method for the desensitization of LLM prompts. Inspired by the work on generating adversarial examples (Alzantot et al., 2018), we introduce the concept of *anti-adversariality*, which aims to obscure sensitive words in prompts while preserving the integrity of model predictions. PromptObfus achieves desensitization by replacing words with semantically distinct yet task-neutral alternatives, thereby ensuring robust privacy protection without compromising the original functionality of the prompts. PromptObfus operates through the deployment of two small local models: a *desensitization model*, which replaces sensitive words with privacy-preserving alternatives, and a *surrogate model*, which emulates the task execution of the remote LLM to guide prompt selection. The pipeline consists of three critical steps: generating desensitized alternatives for privacy-sensitive words, assessing the task utility of the LLM, and selecting replacements that minimize performance degradation.

We evaluate PromptObfus across three NLP tasks: sentiment analysis, topic classification, and question answering. Results demonstrate that PromptObfus achieves accuracy rates of 84.8%, 84.25%, and 96.4%, respectively, surpassing existing baselines. In terms of privacy protection, PromptObfus reduces the success rate of implicit privacy inference attacks by 24.86% and entirely mitigates explicit inference attacks.

Our contribution can be summarized as follows:

- We introduce the novel concept of **anti-adversariality**, a pioneering approach for desensitizing LLM prompts that ensures robust privacy protection without compromising task performance.
- We propose a new privacy-preserving word replacement algorithm, which integrates masked word prediction with LLM gradient surrogation to achieve optimal desensitization.
- We conduct extensive evaluations of PromptObfus across multiple NLP tasks, demonstrating its effectiveness in preserving privacy while maintaining task performance.

2 Related Work

Privacy Protection for LLMs. While LLMs have demonstrated significant utility across diverse domains, they have also introduced notable privacy and security challenges (Mireshghallah et al., 2024). To mitigate these concerns, research has concentrated on safeguarding both the model and user data. Techniques such as federated learning (Hu et al., 2024) and homomorphic encryption (Hao et al., 2022) are widely employed to secure model training and inference. For prompt privacy, methods including prompt encryption (Lin et al., 2024) and noise-based obfuscation (Zhou et al., 2023; Gao et al., 2024) have been proposed. Additionally, training models for data anonymization by detecting and removing personally identifiable information (PII) (Chen et al., 2023; Sun et al., 2024) has been explored. Users can also employ strategies such as mixing real and synthetic inputs to construct privacy-preserving prompts, thereby preventing servers from identifying the original input (Utpala et al., 2023).

Automatic Prompt Engineering. Automatic prompt generation represents a promising approach for creating desensitized prompts, utilizing AI techniques to generate prompts that effectively guide models in producing meaningful responses. These methods leverage large-scale datasets for training, enabling broader linguistic knowledge and contextual understanding, often surpassing manually crafted prompts (Zhou et al., 2022). Notable frameworks for automatic prompt generation include APE (Yang et al., 2024), which iteratively refines prompts by selecting and resampling candidate prompts; APO (Zhou et al., 2022), which adjusts prompts through feedback in a gradient descent-like manner; and OPRO (Pryzant et al., 2023), which treats the LLM as an optimizer to iteratively enhance prompts.

Text Adversary Generation. Adversarial training is a technique aimed at improving model robustness against malicious or deceptive inputs, widely applied in domains such as computer vision, NLP, and speech recognition. In this approach, models are systematically exposed to adversarial examples (Goodfellow et al., 2014), which are inputs subtly modified to induce significant changes in model outputs. Genetic algorithms are employed to generate semantically equivalent adversarial samples (Alzantot et al., 2018), selecting synonyms that maximize the likelihood of the target label. More

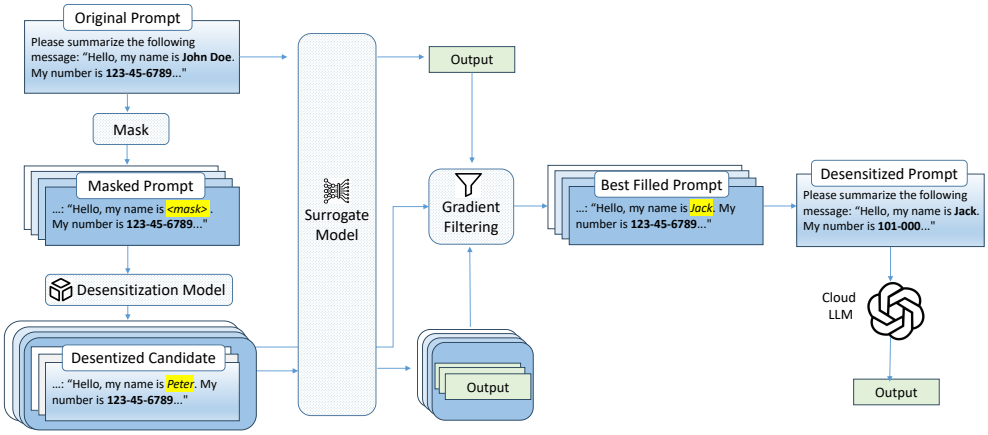


Figure 2: Overview of PromptObfus.

recently, LLMs are utilized to produce adversarial samples (Wang et al., 2023).

In contrast to existing approaches, we propose an *anti-adversarial* method for the desensitization of LLM prompts, which ensures that model outputs remain consistent while rendering sensitive content imperceptible to human interpretation.

3 Approach

Inspired by the principles of adversarial example generation (Alzantot et al., 2018), we conceptualize our approach as an *anti-adversarial* framework, wherein the objective is to obfuscate sensitive information while preserving the original behavior and predictive performance of the model.

3.1 Problem Statement

Given an LLM $\Phi(y|x)$, parameterized by Φ, and a downstream task (e.g., question answering) represented by a parallel dataset $\mathcal{T} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where x and y denote the input prompt and target output sequence, respectively, we aim to address the following problem. For a predefined set of privacy attributes $P = [p_1, p_2, \ldots, p_m]$ and an input prompt $x = \{x_1, \ldots, x_n\}$, our objective is to transform x into a desensitized version $x' = \{x'_1, \ldots, x'_n\}$ that excludes all privacy attributes while preserving the task's utility. Formally, this is expressed as:

$$\min_{x'=M(x|\lambda,k)} \|s(\Phi(x'), y) - s(\Phi(x), y)\| \quad s.t. \quad x'_i \notin P \quad \forall x'_i \in x' \tag{1}$$

where $M(x|\lambda, k)$ represents a desensitization function that maps sensitive words in the input prompt to their desensitized counterparts; λ denotes the size of the candidate set of desensitized words generated for each sensitive word during the replacement process; k represents the confusion ratio; and $s : Y \times Y \to \mathbb{R}$ is an evaluation metric specific to the task, such as the BLEU score for question answering tasks.

3.2 Overview

Our approach is designed to identify the optimal desensitization function $M(x|\lambda, k)$ for input prompts, minimizing its impact on the LLM's output. Figure 2 illustrates the overall architecture of PromptObfus. The pipeline consists of three steps: (i) masking privacy-sensitive words of the original prompt, and generating candidate desensitized alternatives using a dedicated desensitization model; (ii) assessing the task utility of various desensitized candidates through a surrogate model, with comparisons made against the original prompt; and (iii) computing the gradient with respect to the task output, selecting the most suitable desensitized words from candidates, and generating the final desensitized prompt.

3.3 Predicting Candidate Desensitive Words

For each privacy-sensitive word in a prompt, PromptObfus predicts a set of candidate desensitive words for potential replacement. This process can be formalized as a Masked Language Model (MLM) task, where the privacy-sensitive words are substituted with a `mask` token. A desensitization model is trained to predict λ candidate desensitized words for each masked position. By leveraging pretrained linguistic knowledge, the desensitization model ensures that the replacement candidates are semantically aligned with the surrounding context.

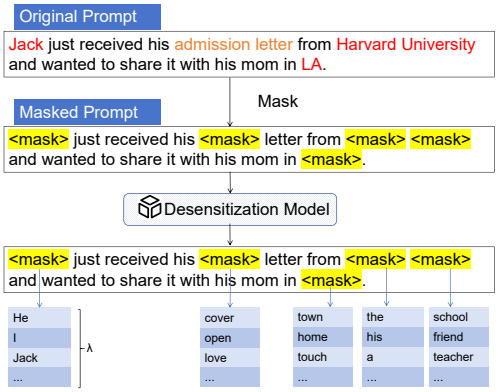


Figure 3: Illustration of predicting candidate desensitive words

This guarantees textual coherence, maintains the original functionality of the prompt, and effectively obscures sensitive information.

To identify privacy attributes, we employ spaCy's named entity recognition (NER) model[1], which efficiently detects and labels entities like person names, locations, and organizations within the text. The identified privacy-sensitive words are uniformly replaced with a MASK token. Besides the explicit privacy words, implicit privacy risks may exist. To mitigate the potential inference of private information from contextual cues, we further randomly mask k of the remaining words in the prompt.

Next, a pre-trained language model, referred to as the desensitization model, is fine-tuned to generate candidate replacement words for each masked token. The desensitization model can be any pre-trained language model capable of performing masked language modeling (MLM).

To mitigate the risk of privacy leakage through synonyms or near-synonyms, the predicted set of desensitized words is further refined based on their semantic similarity to the original word. For each candidate desensitized word w_i, we calculate its Euclidean distance to the original words x_{original} using their respective word embeddings:

$$d(x_{\text{original}}, w_i) = \|\vec{x_{\text{original}}} - \vec{w_i}\| \quad (2)$$

where $\vec{x_{\text{original}}}$ and $\vec{w_i}$ represent the word vector representations of the original and desensitized words, respectively, and $\|\cdot\|$ denotes the Euclidean norm.

A distance threshold θ_{dist} is introduced to further refine the desensitized word set. If the Euclidean distance $d(x_{\text{original}}, w_i)$ is below this threshold, the desensitized word is deemed semantically similar to the original word and is consequently excluded from the candidate set. This process is formalized as:

$$W_{\text{filtered}} = \{w_i \in W \mid d(x_{\text{original}}, w_i) > \theta_{\text{dist}}\} \quad (3)$$

where W_{filtered} represents the filtered set of desensitized words. By eliminating words that are too close in semantic space to the original term, this step significantly reduces the risk of privacy leakage.

3.4 Assessing Task Utility

To ensure that the chosen desensitized words minimally impact the task output, we design a gradient-based selection mechanism. The underlying rationale is that gradients quantify the sensitivity of the input with respect to the model's output. Specifically, large gradient magnitudes suggest that a desensitized word could significantly alter the task's semantic meaning, whereas smaller gradients indicate that the replacement preserves the original semantics and minimizes disruptions to the text.

Directly acquiring gradients from remote LLMs is impractical; therefore, PromptObfus utilizes a surrogate model $\mathcal{M}_{surrogate}$ to simulate the behavior of the remote LLM. The surrogate model is a smaller, white-box LLM capable of evaluating task performance while providing gradient feedback. PromptObfus supports two types of surrogate models:

1) Task-specific model: When a sufficient task-related dataset $\mathcal{D} = \{(x, y)\}$ is available, the surrogate model can be fine-tuned to improve its performance on the specific task. This task-specific model provides accurate gradient information for the desensitized prompt.

2) General model: In scenarios where task-related data is limited, the surrogate model can be a larger language model pre-trained on a diverse corpus, offering a broad understanding of language. Although the general model is typically larger than a task-specific model, the gradient information it generates may be less task-sensitive, providing a more generalized approximation.

3.5 Gradient Filtering

PromptObfus leverages the gradient information provided by the surrogate model $\mathcal{M}_{surrogate}$ to evaluate the filtered set of desensitized candidate

[1] https://spacy.io/models/en/#en_core_web_trf

words $W_{filtered}$ and select the word associated with the smallest gradient magnitude.

For each desensitized word $w \in W_{filtered}$, PromptObfus constructs a filled prompt x', calculates the gradient with respect to the task output. Formally, this process is expressed as:

$$\Delta_i(w) = \left\| \frac{\partial \mathcal{L}(y, \mathcal{M}_{surrogate}(x'[i \leftarrow w]))}{\partial x'} \right\| \quad (4)$$

where i denotes the position of the current privacy word, $\Delta_i(w)$ represents the current gradient magnitude, and $\mathcal{L}$ denotes the task-specific loss function. By iteratively updating the minimum gradient and its corresponding word, the optimal desensitized word w^* is selected as:

$$w^* = \arg \min_{w \in W_{filtered}} \Delta_i(w) \quad (5)$$

Finally, PromptObfus replaces the privacy-sensitive word at position i with w^* and iterates this process for all masked positions. This incremental filling strategy ensures that each replacement word is chosen based on both the local context of the masked position and the global context of previously filled words, thereby optimizing semantic coherence and preserving task performance.

4 Experiments

4.1 Experiment Design

We evaluate the effectiveness of PromptObfus across two critical dimensions, emphasizing its ability to balance robust privacy protection with the preservation of task performance. To demonstrate its practical utility, we apply PromptObfus to three NLP tasks: sentiment analysis, topic classification, and question answering. These tasks are representative of real-world applications and provide a comprehensive assessment of the method's applicability.

To measure PromptObfus's efficacy in privacy protection, we simulate external attacks to determine whether sensitive information can be inferred from the desensitized prompts. We employ three distinct privacy attackers, including two text reconstruction methods and one privacy inference method:

KNN-Attack (Qu et al., 2021) computes the distance between each word representation and a publicly available word embedding matrix, selecting the k-nearest words as the inferred result.

Mask Token Inference Attack (Yue et al., 2021) applies a masking strategy to the desensitized prompts, sequentially obscuring words and testing the attacker's ability to accurately infer the hidden content.

PII Inference Attack (Plant et al., 2021) analyzes the text to infer sensitive information about users.

We quantify the extent to which privacy information can be inferred by third-party attackers. Specifically, we employ two metrics to evaluate the effectiveness of privacy protection:

TopK (Zhou et al., 2023) is a token-level metric that computes the proportion of correctly inferred words among the top k predictions generated by the attacker.

Success rate (Plant et al., 2021) measures the percentage of PII entities that are successfully leaked relative to the total PII present, in response to the PII inference attack.

For evaluating PromptObfus's effectiveness in preserving task performance, we directly compute the accuracy of the target tasks when instructed using our desensitized prompts. Specifically, we adopt two widely adopted metrics for evaluation:

Accuracy quantifies the proportion of correct predictions generated by the model relative to the total number of test samples. This metric is applied to both classification and QA tasks.

Answer quality score assesses the overall quality of generated answers, considering factors such as accuracy, relevance, completeness, and readability. Gpt-4o-mini is utilized as an automated evaluator to assign scores for answer quality, with the specific evaluation prompt detailed in Appendix A.2.

4.2 Datasets

We utilize two widely used benchmark datasets, **SST-2** (Socher et al., 2013) for sentiment analysis and **AG News** (Zhang et al., 2015) for topic classification, to evaluate our approach. Additionally, we introduce a specialized dataset, **PersonalPortrait**, designed for privacy-centric question answering tasks. The statistical details of these datasets are summarized in Table 1.

Existing QA datasets are typically anonymized or lack sensitive information, making them inadequate for privacy evaluation. To address this, we develop PersonalPortrait, a psychological counseling QA dataset containing sensitive data for privacy testing. It includes 400 patient self-reports, generated using GPT-4 and manually reviewed to ensure quality and authenticity. Additional details on dataset construction are provided in Appendix A.1.

Dataset	Split	Number of Samples
SST-2	Train	67,349
	Validation	872
	Test	1,821
AG News	Train	120,000
	Validation	7,600
	Test	7,600
PersonalPortrait	Test	400

Table 1: Statistics of the datasets.

4.3 Baselines

We evaluate PromptObfus against six state-of-the-art privacy-preserving methods and the original text. 1) **Random Perturbation**, which randomly substitutes a subset of tokens in the text with arbitrary words. 2) **Presidio**[2], a tool designed to automatically detect and redact sensitive information, such as names, locations, and other PII. 3) **SANTEXT** (Yue et al., 2021), a differential privacy-based approach that utilizes Euclidean distance in word embedding space to determine replacement probabilities. 4) **SANTEXT+** (Yue et al., 2021), an enhanced version of SANTEXT that incorporates word frequency to adjust replacement probabilities. 5) **DP Prompt** (Utpala et al., 2023), a method that leverages a prompt-based framework to paraphrase the original prompt using an LLM. 6) **PromptCrypt** (Lin et al., 2024), which employs a large model to encrypt the original prompt into emoji sequences.

4.4 Implementation Details

We implement PromptObfus using three open-source models: RoBERTa-base[3] serves as the desensitization model, BART-large[4] functions as the task-specific surrogate model for classification tasks, and GPT-Neo-1.3B[5] is employed as the general surrogate model for QA tasks, chosen due to the smaller dataset size. Additional details regarding hyperparameter configurations can be found in Appendix A.3.

4.5 Overall Performance

To ensure a fair comparison, we maintain a consistent obfuscation ratio across all word-level protection baselines and PromptObfus. Since DP Prompt and PromptCrypt are not word-level protection methods, they cannot be evaluated using MTI Attack or KNN Attack. Consequently, we exclusively employ PI Attack for privacy protection evaluation. The experiments are conducted using the original parameters specified in their respective papers, with GPT-4o-mini serving as the base model.

Table 2 presents the results on the SST dataset[6]. First, PromptObfus exhibits exceptional privacy protection capabilities. When using desensitized prompts generated by PromptObfus, the success rate of the PI attack remains consistently at 0.00%. In contrast, baseline methods such as SANTEXT+, DP Prompt, and PromptCrypt do not specifically safeguard PII; instead, they disrupt linguistic structures, resulting in relatively higher PI Attack success rates. PromptObfus ($k = 0.3$) also achieves a 30.42% success rate in the MTI Attack, significantly lower than all methods except SANTEXT and SANTEXT+.

Furthermore, PromptObfus maintains the performance of the target tasks without significant degradation. At $k = 0.1$, PromptObfus achieves a classification accuracy of 84.8%, representing only a 2.75% decrease compared to the original text. This performance is comparable to Presidio and surpasses other word-level baselines, such as random replacement (69.87%) and SANTEXT+ (58.93%).

In summary, the results demonstrate that PromptObfus effectively protects privacy against remote LLMs while preserving the original LLM's task performance, achieving an optimal privacy-utility trade-off among all baseline methods.

4.6 Ablation Study

Impact of Surrogate Model. We investigate the impact of architectures and scales of the surrogate model. The experiments are conducted on three distinct model architectures: Encoder-only models (RoBERTa), Decoder-only models (GPT2), and Encoder-decoder models (BART), across three groups of sizes, including base (around 130M parameters, e.g, RoBERTa-base), medium (around 350M, e.g, RoBERTa-large, BART-large, and GPT2-medium), and large (LLaMA-2-7B and ChatGLM3-6B). Due to computational constraints, small and medium models are fine-tuned with full parameters, while the large models are fine-tuned using Low-Rank Adaptation (LoRA). The experi-

[2]https://microsoft.github.io/presidio/
[3]https://huggingface.co/FacebookAI/roberta-base
[4]https://huggingface.co/facebook/bart-large
[5]https://huggingface.co/EleutherAI/gpt-neo-1.3B

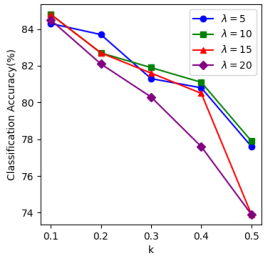
[6]Results for other datasets are provided in Appendix A.4.

Approach	MTI Top1↓	KNN Top1↓	PI Success Rate↓	Acc.↑
Origin	48.86	–	–	87.20
Random	35.91	90.47	83.47	69.87
Presidio	44.63	90.45	**0.00**	84.80
SANTEXT	**20.15**	73.67	92.53	49.25
SANTEXT+	23.40	76.93	75.47	58.93
DP-Prompt	–	–	72.53	86.30
PromptCrypt	–	–	54.67	**89.86**
PromptObfus (k=0.1)	40.99	83.44	**0.00**	84.80
PromptObfus (k=0.2)	35.12	74.37	**0.00**	82.70
PromptObfus (k=0.3)	30.42	**66.27**	**0.00**	81.60

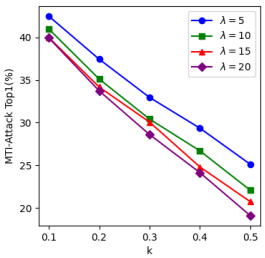
Table 2: Performance of privacy protection and task utility on the SST-2 sentiment analysis task. In the PI Attack, the SST-2 dataset does not explicitly label privacy attributes. Therefore, the attack assumes that named entities (e.g., person names, locations) represent explicit privacy attributes and targets these for evaluation.

Approach	MTI Top1↓	KNN Top1↓	Acc.↑
Original Data	48.86	–	87.2
Roberta-base	44.11	83.39	81.6
BART-base	44.67	83.44	82.1
GPT2-base	44.26	84.48	84.5
GPT2-medium	44.22	83.44	84.5
Roberta-large	44.36	83.39	84.3
BART-large	44.31	83.39	**84.8**
llama-2-7B	44.37	83.44	83.8
ChatGLM3-6B	44.27	83.39	83.6

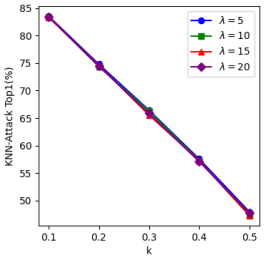
Table 3: Influence of surrogate model variations on obfuscation effectiveness in sentiment analysis.



(a) Classification accuracy.



(b) MTI attack.



(c) KNN attack.

Figure 4: Impact of hyperparameters k and λ.

mental results for the sentiment analysis task are presented in Table 3.

We observe that privacy protection effectiveness is independent of the surrogate model's architecture and size. Medium-sized models outperform larger models due to the task's simplicity, where increased model complexity provides no added benefit, and LoRA may not fully leverage fine-tuning advantages. Encoder-Decoder models excel by combining the encoder's classification suitability with the decoder's alignment to remote models. Similar results for the QA task are detailed in Appendix A.5.

Impact of Hyperparameters. We perform ablation studies on the hyperparameters k and λ, using BART-large as the surrogate model on the SST dataset. The parameter k is varied from 0.1 to 0.5 in increments of 0.1, while λ ranges from 5 to 20 in increments of 5. The results are illustrated in Figure 4.

For privacy protection, as k increases, Attack Top1 decreases, indicating enhanced privacy protection. For MTI Attack, increasing λ reduces Top1, with the most notable improvement occurring when λ rises from 5 to 10, as diversified contexts yield more varied MTI predictions. For KNN Attack, Top1 depends solely on k, as it focuses on perturbed words independently of context.

For performance preservation, classification accuracy declines as k increases, with the most significant drop observed between 0.4 and 0.5. When k exceeds 0.3, λ becomes sensitive, and higher values degrade performance due to excessive word replacements disrupting semantics and reducing contextual coherence.

Overall, increasing k and λ enhances privacy protection but compromises performance. The optimal balance is achieved when $k \leq 0.4$ and $\lambda \in [10, 20)$. We set λ to 10 as default.

Original Text:	I'm a 39 -year-old driver in Toronto , and I often feel like my emotions are all over the place...
Random:	abuser a 39 -year-old driver in Toronto , moha palmery often feel like my emotions are all over shady place...
Presidio:	I'm a <DATE> driver in <GPE>, and I often feel like my emotions are all over the place...
SANTEXT:	jagger rehashed a hardy - year - old driver in women , and obscure often feel like my emotions are all over the place...
SANTEXT+:	jagger rehashed a fidel 15 year 3 old driver in motion , and esoteric seldom feel like my emotions are all putting the however...
DP-Prompt:	I'm a 39 -year-old driver in Toronto , and my emotions can be unpredictable...
PromptCrypt:	39 🚙 🪨, 😴 → 😡, 😔 → 😞 💔 -> 💔 🚻,...
PromptObfus (k=0.1):	I'm a commercial driver of two and I often feel like my emotions are all over the place...
PromptObfus (k=0.2):	I'm a commercial assistant in LA and I often feel like my emotions flow all over the world...
PromptObfus (k=0.3):	I'm one professional assistant in general and I often feel like my emotions are hovering throughout...

Table 4: A case of desensitized prompts generated by various methods for question answering.

4.7 Case Study

Table 4 illustrates an example of desensitized prompts generated by different methods for the question-answering task. In the original text, terms such as "39-year-old," "driver," and "Toronto" are identified as sensitive information. PromptObfus effectively replaces explicit privacy details (e.g., age and location) with de-identified terms, ensuring robust privacy protection. At $k = 0.2$ and $k = 0.3$, the obfuscation intensity increases, and implicit privacy details, such as occupation ("driver"), are substituted with more ambiguous terms like "assistant" while preserving semantic coherence and readability.

In contrast, the Random method fails to accurately identify and modify sensitive information, leading to the leakage of all privacy-related terms and a lack of textual coherence. Presidio is limited to handling predefined temporal and address information, offering insufficient flexibility and failing to protect occupation-related privacy. Meanwhile, SANTEXT and SANTEXT+ introduce excessive noise, rendering the sentences overly disordered and degrading task performance. DP-Prompt results in privacy leakage, while PromptCrypt, despite protecting privacy, employs overly simplistic and abstract symbols, causing significant performance degradation.

4.8 Transferability

We further explore the transferability of the trained surrogate model. We experiment on different combinations of local and remote models from three vendors: OpenAI, Meta, and Zhipu. Experimental results, presented in Table 5, indicate that the combination of local and remote models from different vendors has minimal influence on obfuscation effectiveness. Models demonstrating strong transferability across various combinations. To further validate this finding, we include BART-large, the top-performing model from prior experiments and independent of the three vendors, for testing with three remote models. Results confirm that BART-large consistently outperforms in all combinations.

Model	GPT-4o-mini	GLM-4-plus	Meta AI
GPT2	84.5	91.2	**91**
ChatGLM3-6B	83.6	90.5	89
llama2-7B	83.8	90.3	90
BART	**84.8**	**91.4**	**91**

Table 5: Classification accuracy of local-remote model combinations on the sentiment analysis (SST) task. Columns denote remote models, while rows denote local models.

5 Conclusion

In this paper, we propose PromptObfus, a novel desensitization method for LLM prompts. Its core idea is *anti-adversarial learning*, which ensures that model outputs remain consistent while obscuring sensitive content from human interpretation. PromptObfus achieves this by replacing sensitive words in user prompts with semantically distant yet task-neutral alternatives, minimizing impact on task performance. Evaluations across three NLP tasks demonstrate that PromptObfus effectively protects privacy against cloud LLMs while preserving the original model's performance, achieving an optimal privacy-utility trade-off compared to baseline methods.

Our replication package is available at: https://anonymous.4open.science/r/PromptObfus-83F7/.

Limitations

We have identified two limitations of PromptObfus: **1) Incomplete identification of implicit privacy words**: Our approach randomly masks words in the prompt as implicit privacy attributes, which reduces privacy leakage but does not comprehensively cover all privacy attributes, leading to incomplete desensitization. Future work should focus on refining privacy word localization strategies. **2) Applicability of general models**: Fine-tuning large models for text-based QA is challenging due to the scarcity of high-quality annotated data. Consequently, general models are employed, but they lack the precision of task-specific models. Further research is needed to explore model adaptation techniques under data constraints, such as few-shot learning methods (Brown et al., 2020).

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

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. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 267–284, USA. USENIX Association.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *Preprint*, arXiv:2309.03057.

Fengyu Gao, Ruida Zhou, Tianhao Wang, Cong Shen, and Jing Yang. 2024. Data-adaptive differentially private prompt synthesis for in-context learning. *Preprint*, arXiv:2410.12085.

Craig Gentry. 2009. *A fully homomorphic encryption scheme*. Ph.D. thesis, Stanford, CA, USA. AAI3382729.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA. MIT Press.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. 2022. Iron: Private inference on transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 15718–15731. Curran Associates, Inc.

Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. 2024. DP-OPT: Make large language model your privacy-preserving prompt engineer. In *The Twelfth International Conference on Learning Representations*.

Jiahui Hu, Dan Wang, Zhibo Wang, Xiaoyi Pang, Huiyu Xu, Ju Ren, and Kui Ren. 2024. Federated large language model: Solutions, challenges and future directions. *IEEE Wireless Communications*, pages 1–8.

Guo Lin, Wenyue Hua, and Yongfeng Zhang. 2024. Emojicrypt: Prompt encryption for secure communication with large language models. *Preprint*, arXiv:2402.05868.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Michele Miranda, Elena Sofia Ruzzetti, Andrea Santilli, Fabio Massimo Zanzotto, Sébastien Bratières, and Emanuele Rodolà. 2025. Preserving privacy in large language models: A survey on current threats and solutions. *Preprint*, arXiv:2408.05212.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2024. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *Preprint*, arXiv:2310.17884.

Richard Plant, Dimitra Gkatzia, and Valerio Giuffrida. 2021. CAPE: Context-aware private embeddings for private language learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7970–7978, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Natural language understanding with privacy-preserving bert. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 1488–1497, New York, NY, USA. Association for Computing Machinery.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Robin Staab, Mark Vero, Mislav Balunovi'c, and Martin T. Vechev. 2023. Beyond memorization: Violating privacy via inference with large language models. *ArXiv*, abs/2310.07298.

Xiongtao Sun, Gan Liu, Zhipeng He, Hui Li, and Xiaoguang Li. 2024. Deprompt: Desensitization and evaluation of personal identifiable information in large language model prompts. *Preprint*, arXiv:2408.08930.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457, Singapore. Association for Computational Linguistics.

Zimu Wang, Wei Wang, Qi Chen, Qiufeng Wang, and Anh Nguyen. 2023. Generating valid and natural adversarial examples with large language models. *Preprint*, arXiv:2311.11861.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *Preprint*, arXiv:2309.03409.

Andrew C. Yao. 1982. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164.

Binwei Yao, Chao Shi, Likai Zou, Lingfeng Dai, Mengyue Wu, Lu Chen, Zhen Wang, and Kai Yu. 2022. D4: a Chinese dialogue dataset for depression-diagnosis-oriented chat. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2438–2459, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Xiaojin Zhang, Yulin Fei, Yan Kang, Wei Chen, Lixin Fan, Hai Jin, and Qiang Yang. 2024. No free lunch theorem for privacy-preserving llm inference. *Preprint*, arXiv:2405.20681.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuanjing Huang. 2023. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473, Toronto, Canada. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

A Appendix

A.1 PersonalPortrait Construction

Inspired by the D4 dataset (Yao et al., 2022) and the PersonalReddit dataset (Staab et al., 2023), which generate text from personal profiles, we construct realistic patient personas based on attributes such as gender, occupation, location, and mental health conditions, simulating their interactions in psychological counseling sessions. The primary objective of the QA task is to diagnose the patient's mental health disorder. For example, the model identifies conditions like depression or anxiety by analyzing symptoms such as anxiety, insomnia, and low mood described in the text.

The dataset synthesis process consists of the following steps:

1. Profile generation: Personal attributes, including age, location, gender, occupation, and mental health condition, are randomly selected from predefined categories to create diverse patient profiles.
2. Text generation: Using GPT-4, we generate self-reported texts based on the profiles, reflecting the patient's history and symptoms. The prompt used for generation is provided in Appendix A.2.
3. Manual review: The generated texts are manually reviewed to ensure high quality, accuracy, and appropriateness of symptom descriptions.

Each entry in the PersonalPortrait dataset comprises the following fields:

- Text: A self-reported narrative generated during a simulated psychological counseling session, detailing the patient's mental health concerns and related experiences.
- Age: A randomly assigned age ranging from 18 to 65 years.
- Location: A randomly selected geographic region from a pool of 20 common English-speaking areas.
- Gender: A randomly assigned gender, either male or female.
- Occupation: A randomly selected profession from a list of 20 common occupations.
- Disorder: A randomly assigned mental health condition, chosen from a set of 10 prevalent disorders included in the dataset.

A.2 Prompt Templates

The following outlines the primary prompt templates utilized in our study, where *TEXT* denotes the desensitized prompt, *ORIGINAL TEXT* represents the original prompt, and *ANSWER* refers to the response generated by the LLM. Attributes such as *AGE* and *GENDER* are randomly selected from predefined categories.

Prompt for Generating Samples for PersonalPortrait

```
System Prompt:
You are a patient and are seeing a psychiatrist.

User Prompt:
Describe to a psychiatrist in the typical voice of a <<<AGE>>>-year-old <<<GENDER>>> <<<OCCUPATION>>> with sympthons of <<<DISORDER>>> in <<<LOCATION>>>. The self-report MUST reflect the patient's LOCATION and OCCUPATION.100 words or less.
```

Prompt for Sentiment Analysis Task

```
System Prompt:
Classify the sentiment of each sentence in <<<SENTENCE>>> as 'Positive' or 'Negative'. Give the sentiment classifications without any other preamble text.

###EXAMPLE SENTENCE
Highly recommend this company for travel plans involving rail.
EXAMPLE OUTPUT
Positive###

User Prompt:
<<<TEXT>>>
```

Prompt for Topic Classification Task

```
System Prompt:
Classify the topic of each news article in <<<NEWS>>> as 'World', 'Sports', 'Business', or 'Sci/Tech'. Give the topic classification without any other preamble text.

###EXAMPLE NEWS
Dollar Briefly Hits 4-Wk Low Vs Euro LONDON (Reuters) - The dollar dipped to a four-week low against the euro on Monday before rising slightly on profit-taking, but steep oil prices and weak U.S. data continued to fan worries about the health of the world's largest economy.
EXAMPLE OUTPUT
Business###

User Prompt:
<<<TEXT>>>
```

Prompt for Question Answering Task

```
System Prompt:
You are an experienced psychologist.
Your task is to listen to the patient's
self-description and a most likely mental
health diagnosis based on their symptoms.

User Prompt:
<<<TEXT>>>
Diagnosis:
```

Prompt for Scoring Answer Quality

```
System Prompt:
You are an experienced psychologist with
expertise in mental health diagnoses.
Your task is to assess the accuracy
of a diagnosis based on the patient's
self-description. You will evaluate the
diagnosis on a scale from 1 to 5, where
1 means the diagnosis is very inaccurate
and 5 means the diagnosis is completely
accurate. Please consider the alignment
between the patient's symptoms and the
diagnosis, the clarity and specificity of
the diagnosis, and any relevant details
that may be missing or misinterpreted.

User Prompt:
The patient has provided the following
self-description. The diagnosis made by the
doctor is as follows: TEXT:<<<ORIGINAL
TEXT>>> Diagnosis:<<<ANSWER>>> Please
provide a score from 1 to 5 based on
how accurate you believe the diagnosis is,
considering the symptoms described and the
quality of the diagnosis. Use the following
scale to guide your evaluation:
1 - The diagnosis is very inaccurate and
does not align with the symptoms described.
2 - The diagnosis has major inaccuracies,
missing or misinterpreting key symptoms.
3 - The diagnosis is moderately accurate,
but some symptoms are either missed or
misinterpreted.
4 - The diagnosis is fairly accurate,
capturing most symptoms with only minor
errors or omissions.
5 - The diagnosis is completely accurate,
perfectly matching the patient's symptoms
and addressing all key details.
```

A.3 Hyperparameter Setting

The hyperparameters for model training in our experiments are detailed in Tables 6 and 7. Specifically, llama-2-7B and ChatGLM3-6B are fine-tuned using Low-Rank Adaptation (LoRA), while the remaining models undergo full fine-tuning. We employ the Adam optimizer with default settings, including $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. The use of all models complies with the license.

The experiments are conducted on a server equipped with 2 Nvidia GeForce RTX 4090 GPUs, running Ubuntu 23.10 and CUDA version 12.2.

Dataset	Model	lr	bs	epoch
SST-2	Roberta-base	2e-5	32	4
	Roberta-large	3e-5	32	4
	BART-base	2e-5	32	4
	BART-large	3e-5	32	4
	GPT2-base	3e-5	32	4
	GPT2-medium	3e-5	32	4
	llama-2-7B	2e-4	16	2
	ChatGLM3-6B	2e-4	16	2
AG News	BART-large	3e-5	32	5

Table 6: Hyperparameters setting for model training.

Dataset	Model	alpha	dropout	r
SST-2	llama-2-7B	16	0.1	64
	ChatGLM3-6B	16	0.1	64

Table 7: LoRA hyperparameters setting for model training.

A.4 Results on Other Datasets

Tables 8 and 9 show the results on the topic classification and question answering tasks, respectively. We observe the same trend as in the sentiment analysis task.

Privacy Protection. Our PromptObfus demonstrates a significant advantage across both tasks. For instance, in the question-answering task, we evaluate two experimental items: *Location*, which is typically explicit and displayed in plain text, and *Occupation*, which is often inferred from context and considered implicit privacy. In the PI Inference of *Location*, PromptObfus achieves an attack success rate below 1.50%, indicating nearly complete privacy protection. In the PI Inference of *Occupation*, PromptObfus achieves the second-lowest attack success rate at 34.75%, trailing only PromptCrypt (11.00%).

Performance Preservation. PromptObfus achieves an accuracy of 84.25% at both $k = 0.1$ and $k = 0.3$ on the topic classification task, closely aligning with the baseline methods (87.5%). The task utility decreases by only 3.71%, ranking just below DP-Prompt (85%) among the baselines. On datasets with rich content and multiple classification labels, the emoji encryption approach of PromptCrypt shows limited effectiveness and no longer outperforms other methods. On the other hand, the PII anonymization method, Presidio, ex-

Approach	Acc.↑	MTI Top1↓	KNN Top1↓	PI Success Rate↓
Origin	87.50	31.37	–	–
PromptObfus (k=0.1)	84.25	24.59	66.04	**0.00**
PromptObfus (k=0.2)	83.50	21.19	58.96	**0.00**
PromptObfus (k=0.3)	84.25	17.79	51.96	**0.00**
Random	83.75	17.10	83.78	97.50
Presidio	83.25	23.28	71.53	**0.00**
SANTEXT	61.50	21.43	62.10	41.75
SANTEXT+	55.25	**11.04**	**49.09**	34.25
DP-Prompt	**85.00**	–	–	96.25
PromptCrypt	72.00	–	–	13.50

Table 8: Performance of privacy protection and task utility on the AG News topic classification task.

Approach	Acc.↑	Quality Score↑	MTI Top1↓	KNN Top1↓	PI(Loc.)↓	PI(Occ.)↓
Origin	96.9	3.86	46.43	–	94.75	60.25
PromptObfus (k=0.1)	96.4	**3.63**	37.57	87.72	1.50	44.50
PromptObfus (k=0.2)	92.1	3.61	29.98	78.23	1.50	43.25
PromptObfus (k=0.3)	91.7	3.56	**24.98**	68.88	1.25	34.75
Random	90.0	3.34	32.67	90.00	81.50	46.25
Presidio	**96.9**	3.56	44.16	96.62	0.75	55.00
SANTEXT	91.0	3.27	55.75	78.56	**0.00**	47.00
SANTEXT+	91.3	3.33	55.75	**61.62**	**0.00**	48.25
DP-Prompt	95.0	3.62	–	–	89.25	55.25
PromptCrypt	49.5	2.89	–	–	16.25	**11.00**

Table 9: Performance of privacy protection and task utility on the PersonalPortrait text QA task.

hibits performance degradation in the this task, where named entities are critical.

For the question-answering task, PromptObfus achieves an accuracy of 96.4%, nearly matching the original text (96.9%) with a minimal loss of 0.51%, second only to Presidio. Presidio performs well because this task relies more on inferring the patient's emotional state from context rather than directly extracting PII. Additionally, in terms of answer quality score, PromptObfus achieves the highest score of 3.63, indicating that responses generated using PromptObfus prompts excel in fluency, completeness, and accuracy.

We observe that PromptCrypt underperforms in terms of performance preservation for the QA task. While its encryption method disrupts contextual structure, providing strong implicit privacy protection, it sacrifices substantial semantic information, adversely affecting its performance in question answering that require nuanced text analysis.

A.5 Impact of Surrogate Model on Other Tasks

Table 10 presents the results for the question-answering task. Given that privacy protection outcomes have been shown to be independent of surrogate model selection in sentiment analysis tasks, this experiment focuses on performance preservation. General surrogate models are employed, including three similarly sized models—RoBERTa-large, BART-large, and GPT2-medium—as well as three GPT series models of varying sizes: GPT2-base, GPT2-medium, and GPT-Neo-1.3B.

GPT-Neo-1.3B achieves the best performance, with a QA accuracy of 96.4% and the highest answer quality score. In terms of model architecture, GPT2 outperforms the other medium-sized models, highlighting the advantage of the Decoder-only architecture in language generation tasks. Regarding model scale, QA accuracy improves progressively with increasing model size. This is attributed to the fact that general models primarily rely on

Model	Accuracy	Utility Score
GPT2-base	93.3	3.55
GPT2-medium	93.8	3.57
GPTNeo-1.3B	**96.4**	**3.63**
RoBERTa-large	93.0	3.53
BART-large	92.8	3.55

Table 10: Influence of surrogate model variations on obfuscation effectiveness in question answering.

knowledge acquired during pretraining, and larger models inherently possess a more extensive knowledge base and superior task execution capabilities, particularly excelling in complex tasks such as text-based question answering.