ϵ-STEAL: LLM Stealing Attack with Local Differential Privacy

Abstract. Large Language Models (LLMs) have transformed natural language processing, demonstrating remarkable capabilities across diverse tasks. However, their deployment introduces critical risks related to intellectual property (IP) violations and potential misuse, particularly as adversaries can imitate these models to steal services. We specifically focus on model stealing attacks, as they are highly relevant to proprietary LLMs and pose a serious threat to their security, revenue, and ethical deployment. Existing research shows that watermarking techniques (WMs) mitigate such risks by embedding imperceptible patterns into LLM outputs, which can be traced to identify the model's origin and verify its IP.

In this paper, we study the susceptibility of WMs by introducing ϵ-STEAL, a model stealing attack that evades WMs and deceives IP checkers. In ϵ-STEAL, the adversary obfuscates watermark impacts by ensuring the service provider, i.e., cloud cannot determine whether a watermarked output belongs to the adversary's training set, under an ϵ-LDP guarantee, where ϵ is a privacy budget. To achieve this goal, ϵ-STEAL injects LDP-preserving noise into token embeddings of the attack's local model, and then fine-tuning the model with watermarked outputs. Our experiments show that this subtle modification allows ϵ-STEAL to effectively bypass WMs and IP checkers without compromising the adversary's model utility. This poses a substantial risk that even robust WMs can be circumvented, potentially allowing adversaries to deceive existing IP checkers.

Keywords: LLMs · Stealing Attack · Local Differential Privacy · Watermarks.

1 Introduction

Large language models (LLMs), such as ChatGPT, Gemini, and Claude [15, 38, 2], have demonstrated remarkable capabilities in text generation, machine translation, and knowledge understanding tasks, often producing outputs indistinguishable from human writing [22, 25]. Due to the substantial resources required for training, LLMs are typically offered as paid application programming interfaces (APIs) [3, 5]. Although users cannot access model weights or architectures of these commercial LLMs, this restriction does not ensure the safety of these models. Malicious actors can mimic LLM behaviors by querying an API to gather input-output pairs, which can then be used to fine-tune local models. With sufficient data, they can replicate the cloud-hosted LLM's behavior in specific domains [6, 8]. These risks highlight significant concerns about the intellectual property (IP) protection of the cloud-hosted proprietary LLMs [30].

To address such risks, service providers (i.e., the cloud) have employed strategies such as watermarks, encryption, limited API access, and differential privacy [25, 51]. Among them, watermarks (WMs) [25, 32, 7, 28] have emerged as a practical tool for LLMs, ensuring traceability, IP protection, and misuse detection. WMs embed imperceptible patterns in LLM outputs, allowing for the identification of unauthorized use. To apply a WM, the cloud may introduce bias into logits of token generations or adjust the sampling process. IP checkers then analyze model outputs for WMs, using statistical tests to detect if the model has been trained on the cloud's watermarked data, indicating that the cloud-hosted LLM has been imitated.

To bypass IP checkers, which potentially support model-stealing attacks, adversaries employ WM removal attacks [53, 27, 39], word deletion attacks [25], and copy-paste attacks [26]. These attacks typically involve replacing tokens with synonyms, paraphrasing text, or altering sentence structure to erase WMs. However, a major issue with these attacks is that they often compromise model performance, increasing perplexity or distorting the semantic meaning of the watermarked output. This makes it difficult for adversaries to steal the behavior of LLMs without significantly impacting their functionality or output integrity.

Key contributions. To balance the trade-off between attack success and model utility, we introduce a novel model stealing attack, ϵ-STEAL, aiming to bypass IP checkers while maintaining high model utility. By using local differential privacy (LDP), ϵ-STEAL obscures the differences between watermarked and non-watermarked LLM outputs, making it hard for IP checkers to verify the ownership of the adversary's model.

Our key contributions are summarized as follows. *First*, a significant advantage of ϵ-STEAL over existing attacks is its ability to maintain the high model utility, ensuring that the adversary's model retains its effectiveness, functionality, and the semantic integrity of its outputs. *Second*, ϵ-STEAL is agnostic to the specific models, watermarking techniques, and IP checker methods used. This generality enhances its practical applicability across various scenarios. *Third*, our experimental results show that by making subtle modifications during the fine-tuning process of the local LLM, ϵ-STEAL successfully bypasses existing IP checkers and watermarking methods, supported by theoretical guarantees.

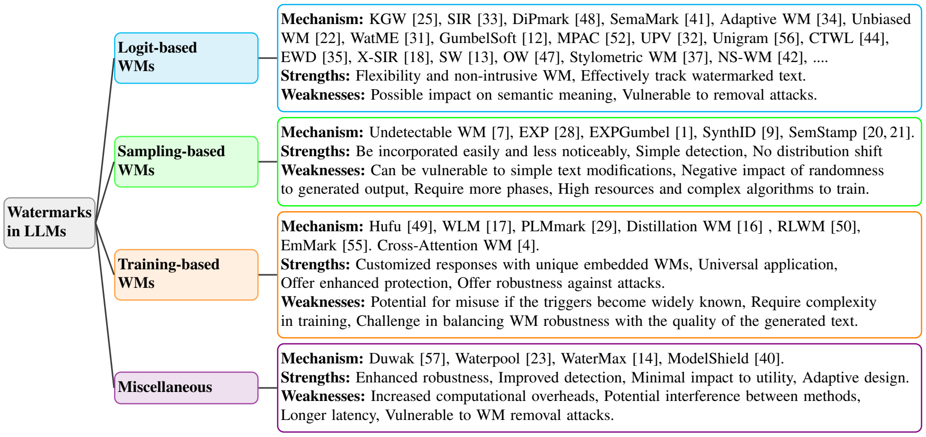
Organization. The paper is organized as follows: Section 2 covers the background, Section 3 presents our ϵ-STEAL method for certified model stealing using local differential privacy, Section 4 discusses the experimental evaluation and comparison with existing methods, and Section 5 concludes with final remarks.

2 Background

2.1 Model Stealing Attacks

In model stealing attacks, adversaries aim to imitate the behavior of a cloud-hosted model denoted as θ, by constructing their local model θ_{adv} through fine-tuning on input-output pairs obtained by querying the cloud-hosted model [54]. The ultimate goal is to bypass IP checkers while maintaining the adversary model's utility, which is comparable to that of the original service provider. To launch the attacks, adversaries start with a set of N task-specific prompts $\{x_i\}_{i=1}^N$, sending them to the cloud to obtain outputs $\{y_i\}_{i=1}^N$. These pairs are used to fine-tune the adversary's model θ_{adv} to mimic the behaviors of θ.

Model stealing attacks have various malicious uses, especially in LLMs. These include 1) imitating proprietary models to offer cheaper and unlicensed services, 2) bypassing service fees, 3) reverse engineering to exploit vulnerability, 4) inferring sensitive data used in training, 5) enabling malicious behaviors and 6) compromising



Mechanism: KGW [25], SIR [33], DiPmark [48], SemaMark [41], Adaptive WM [34], Unbiased WM [22], WatME [31], GumbelSoft [12], MPAC [52], UPV [32], Unigram [56], CTWL [44], EWD [35], X-SIR [18], SW [13], OW [47], Stylometric WM [37], NS-WM [42], ...
Strengths: Flexibility and non-intrusive WM, Effectively track watermarked text.
Weaknesses: Possible impact on semantic meaning, Vulnerable to removal attacks.

Mechanism: Undetectable WM [7], EXP [28], EXPGumbel [1], SynthID [9], SemStamp [20, 21].
Strengths: Be incorporated easily and less noticeably, Simple detection, No distribution shift
Weaknesses: Can be vulnerable to simple text modifications, Negative impact of randomness to generated output, Require more phases, High resources and complex algorithms to train.

Mechanism: Hufu [49], WLM [17], PLMmark [29], Distillation WM [16] , RLWM [50], EmMark [55]. Cross-Attention WM [4].
Strengths: Customized responses with unique embedded WMs, Universal application, Offer enhanced protection, Offer robustness against attacks.
Weaknesses: Potential for misuse if the triggers become widely known, Require complexity in training, Challenge in balancing WM robustness with the quality of the generated text.

Mechanism: Duwak [57], Waterpool [23], WaterMax [14], ModelShield [40].
Strengths: Enhanced robustness, Improved detection, Minimal impact to utility, Adaptive design.
Weaknesses: Increased computational overheads, Potential interference between methods, Longer latency, Vulnerable to WM removal attacks.

Fig. 1: Watermarks in LLMs.

IP checkers. These attacks pose a significant threat to the security, revenue, and ethical deployment of LLMs, underscoring the need for robust defenses.

2.2 Watermark-based Defenses

Recent work has shown that WMs are effective in defending against model stealing attacks for service providers [25, 28, 33, 20]. A WM can be represented as a triple $(\mathcal{G}, \mathcal{W}, \mathcal{D})$, where $\mathcal{G}$ is the WM generator, which takes a prompt x and a WM function $\mathcal{W}$ to produce a watermarked output $y^{wm} = \mathcal{G}(x, \mathcal{W})$, and $\mathcal{D}$ is a WM detection function, also known as an IP checker, to determine whether a given text is watermarked.

WMs typically introduce perturbations to LLM outputs and can be classified as follows. First, *logit-based WMs* modify token selection by biasing the model towards certain tokens [25] such as green or red tokens. While effective for tracking outputs, these methods can affect semantic meaning and are vulnerable to removal attacks [53]. Second, *sampling-based WMs* alter token or sentence sampling during generation without changing the output distribution [28, 20]. These approaches are easy to implement and facilitate detection but are susceptible to modifications and reordering attacks. Third, *training-based WMs* embed WMs using techniques like knowledge distillation or architectural changes [50, 16, 4], offering robustness at the cost of complexity and higher computational costs. *In addition*, several miscellaneous WM approaches, such as mixed methods, multiple-output generation with selection, or prompt-based techniques, offer enhanced robustness and adaptability [14, 40, 23, 57]. However, these methods come with trade-offs, including higher computational costs and potential vulnerabilities. Fig. 1 shows the taxonomy of WMs in LLMs together with their strengths and weaknesses.

2.3 WM IP Checkers

After watermarking the LLM outputs, the service providers can use IP checkers to determine whether a specific model has used the watermarked outputs to mimic the behaviors of cloud-hosted model for fine-tuning its local model. This is typically done by checking if the generated text from the model are watermarked. Depending on a WM, the

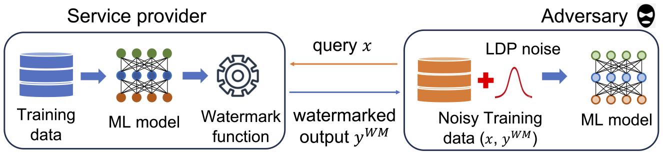


Fig. 2: An overview of ϵ-Steal.

cloud will apply its specific detection function to assess the output. The outputs will be flagged as watermarked if the statistical tests or alignment scores exceed predetermined thresholds. Subsequently, a model is considered as potentially stolen if the detected rate surpasses a specific threshold.

2.4 Local Differential Privacy

Local Differential Privacy (LDP) is a mathematically provable approach for ensuring data privacy [11, 45]. They generally build on the ideas of randomized response [46], originally developed to allow survey respondents to provide truthful inputs while preserving confidentiality. The definition of ϵ-LDP is as follows:

Definition 1. *A randomized algorithm $\mathcal{A}$ satisfies ϵ-LDP, if for any two inputs x and x', and for all possible outputs $\mathcal{O} \in Range(\mathcal{A})$, we have: $Pr[\mathcal{A}(x) = \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{A}(x') = \mathcal{O}]$, where ϵ is a privacy budget and $Range(\mathcal{A})$ denotes every possible output of $\mathcal{A}$.*

The privacy budget ϵ controls the amount by which the distributions induced by inputs x and x' may differ. A smaller ϵ enforces a stronger privacy guarantee. In this work, we leverage the concept of LDP to perturb the training data of adversaries, thereby constraining the differences among outputs while ensuring a level of LDP guarantee.

3 ϵ-Steal: LLM Stealing Attack with Local Differential Privacy

We introduce ϵ-Steal, a novel model stealing attack designed to balance the trade-off between bypassing IP checkers and preserving model utility. The key idea behind ϵ-Steal is to handle watermarked training data from the cloud-hosted model using noise-modified embeddings. This is achieved by incorporating LDP, which adds calibrated noise to the embeddings, minimizing the distinction between watermarked and non-watermarked outputs. This allows the attack to bypass IP checkers without altering the tokens, offering enhanced robustness while maintaining the utility of the stolen model.

3.1 System Operation

Figure 2 and Algorithm 1 provide overview and pseudo-code of our proposed ϵ-Steal.

At the cloud, the service provider trains their model θ using their data (Lines 4-5). Subsequently, the final model θ is made available to users via an API or Machine Learning as a Service (MLaaS) [3, 5]. To protect its outputs and mitigate model stealing attacks, WMs are added to outputs before releasing them to users (Lines 6 and 9). On the other side, the adversary goal is to steal the cloud-hosted model θ by conducting a model stealing attack. It first prepares training data by querying a set of N prompts $\{x_i\}_{i=1}^{N}$ and collecting its outputs $y_i^{wm} = \mathcal{W}(\theta(x_i))$, which have been watermarked

Algorithm 1 ϵ-STEAL Algorithm

```
1:  **Inputs**: Cloud-hosted model $\theta$, adversary's model $\theta_{adv}$, set of prompts $\{x_i\}_{i=1}^{N}$, set of testing
    prompts $\{x_j\}_{j=1 test}^{N}$, number of training iterations $T$, watermark function $\mathcal{W}$, watermark
    detection function $\mathcal{D}_W$
2:  **Outputs**: $\mathbb{I}(\theta_{adv}, \theta)$
3:  **At the Cloud**:
4:      Initialize model parameters $\theta$
5:      Train/Fine-tune $\theta$ using training data available at the cloud
6:      Add a watermark $\mathcal{W}$ to outputs $y_i$ before releasing them to users.
7:  **At the Adversary**:
8:      Initialize model parameters $\theta_{adv}$
9:      **for** $i = 1, \ldots, N$ **do**
10:         Query the cloud-hosted model: $y_i^{wm} = \mathcal{W}(\theta(x_i))$
11:         Add $\epsilon$-LDP noise into token embeddings of sample $i$ with with noise scale $\sigma$:
                $(\bar{x}_i, \bar{y}_i) = (x_i + \mathcal{N}(0, \sigma), y_i^{wm} + \mathcal{N}(0, \sigma))$
12:     Form a training set: $D = \{\bar{x}_i, \bar{y}_i\}_{i=1}^{N}$
13:     **for** $t = 1, \ldots, T$ **do**
14:         Randomly select a set of training samples $D_t \subseteq D$
15:         $\theta_{adv}^{t} = \theta_{adv}^{t-1} - \eta \bigtriangledown_{\theta_{adv}} \mathcal{L}(\theta_{adv}^{t}, D_t)$
16: **IP Checker (at the Cloud)**:
17:     **for** $j = 1, \ldots, N_{test}$ **do**
18:         Query the adversary model: $y_j = \theta_{adv}(x_j)$
19:         Check if $y_j$ is watermarked: $\mathbb{I}\big(\mathcal{D}_W(y_j, \theta_{adv}) = 1\big)$
```

by the service provider (Line 10). By doing that, the adversary can exploit prompt-output correlations to replicate the model's behavior during training. Unlike conventional attacks [53, 39], which often modify tokens from watermarked outputs, this approach adds noises to token embeddings before fine-tuning its model θ_{adv} without modifying tokens, therefore maintaining model utility effectively. Finally, when theft is suspected, the service provider employs an IP checker to detect if the suspect has been trained using its watermarked data by testing the outputs from the suspect model (Lines 16-19).

3.2 Attack Strategy: Bounding Output Differences

The principles of IP checkers are to deploy a WM detection function that exploits different distributions of LLM outputs when responding to watermarked functions. Therefore, to bypass the IP checkers, an effective attack must evade these distinctive output patterns. At the same time, the attack must maintain high model utility to avoid detection. Balancing this trade-off is non-trivial.

To address the trade-off, we leverage a concept of LDP by adding noises into the data so that the cloud cannot distinguish whether the adversary model was trained using their watermarked outputs. The theoretical guarantee is defined as follows:

Theorem 1. ϵ-STEAL *satisfies ϵ-LDP, for any if for any two data samples (x, y^{wm}) and (x', y'^{wm}), and for all possible outputs of the mechanism R_θ, we have:*

$$\begin{aligned} & P[\mathcal{A}(x + \mathcal{N}(0, \sigma), y^{wm} + \mathcal{N}(0, \sigma)) \in R_\theta] \\ \leq & \exp^{\epsilon} P[\mathcal{A}(x' + \mathcal{N}(0, \sigma), y'^{wm} + \mathcal{N}(0, \sigma)) = R_\theta] \end{aligned} \tag{1}$$

where ϵ is a privacy budget, σ is a noise scale associated with the privacy budget ϵ, and R_θ denotes every possible outputs of the mechanism, which is a model parameter space.

The Proof of Theorem 1 is in Appendix A.

As being guaranteed by LDP, we guarantee with ϵ-LDP that by observing the adversary model θ_{adv} and its outputs, the IP checker cannot tell whether θ_{adv} was trained on the watermark data or not and fail to verify the IP of θ_{adv}.

4 Experiments

In this section, we conduct extensive experiments to shed light on **1)** The effectiveness of ϵ-STEAL as an LLM stealing attack against existing WMs, **2)** ϵ-STEAL in comparison with existing WM attacks, **3)** The interplay between attack effectiveness and LDP guarantees, **4)** The impact of different LLMs on the attack, and **5)** The effects of ϵ-STEAL on model utility, including generated text and downstream tasks.

4.1 Baselines

We carry out experiments on four of the state-of-the-art (SOTA) WM approaches, including **1)** *KGW* [25], which divides vocabulary into green and red tokens and modifies the logits of next token generations; **2)** *EXP* [28], which intervenes the sampling process of each token; **3)** *SIR* [33], where watermark logits are determined by the semantics of all preceding tokens ; and **4)** *SemStamp* [20], which maps candidate sentences into embeddings and accepts sentences in the valid region.

However, in our main experimental results section, we exclude *SemStamp* and instead include its results in the Appendix. The reason is that *SemStamp* requires fine-tuning a robust sentence embedder to support the local sensitivity mechanism, which partitions the embedding space for selection process. Due to issues with the code, we could not perform the contrastive fine-tuning required for the embedder. Instead, we used an available pre-trained embedder provided by the authors, which somewhat differs from our intended setting, leading to suboptimal results in the experiments. All appendices are in

In addition, we compare our ϵ-STEAL with three SOTA watermarking attacks, including **1)** a watermark removal attack [53], referred to as *WMremoval*, which removes watermarks by paraphrasing while maintaining high-quality non-watermarked outputs; **2)** *Dipper* [27], which reorders content and alters wording, and **3)** *Substitution* attack developed from [39], which randomly substitute words based on surrounding contexts. A pre-trained model without watermarking techniques or attacks is referred to as the *Original* model, while WM outputs before the attack are referred as *Baseline*.

4.2 Dataset and Model Configurations

To simulate diverse scenarios, we randomly select $10,000$ training samples and $2,000$ testing samples from C4 dataset [10]. For each sample, we truncate the first 200 tokens as a prompt and let LLMs generate the next maximum of 200 tokens as an output. We chose two pre-trained LLMs, LLaMA-2 7B [43] and Mistral 7B [24] for their identical embedding size (32000×4096) to ensure a fair comparison. Our benchmark tasks include text generation as the main task and massive multi-task language understanding (MMLU) [19] as a downstream task to assess the impact of our attack.

In our experiments, we utilize four levels of local differential privacy (LDP) with noise scales of $\sigma \in \{0.001, 0.01, 0.05, 0.1\}$. These noise scales are associated with ϵ values of $\{300, 30, 6, 3\}$ for the LLaMA-2 model and $\{50, 5, 1, 0.5\}$ for the Mistral model, representing rigorous privacy protection of LDP for LLMs. Details on how ϵ is computed from noise scales are provided in Appendix B.

4.3 Evaluation Metrics

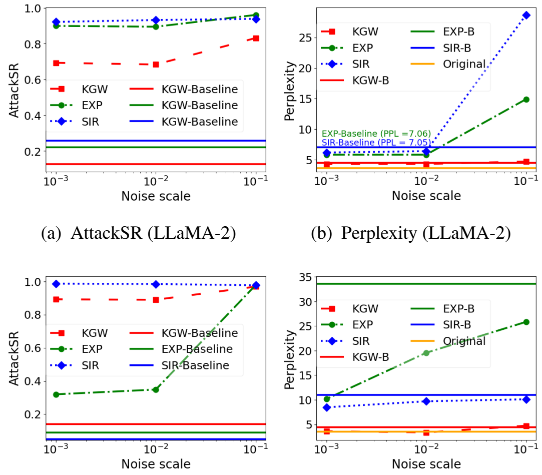
For a model-stealing attack to succeed, it must achieve a high attack success rate by obscuring the differences between watermarked and non-watermarked text. In addition, it should maintain model utility, ensuring high-quality text and preserving performance on downstream tasks. To evaluate this, we validate through three aspects. *First*, we calculate the attack success rate, named Attack SR, as follows:

$$\text{Attack SR} = 1 - \frac{\sum_{i=1}^{N_{test}} \mathbb{I}\Big(\mathcal{D}_W(y_i, \theta_{adv}) = 1\Big)}{N_{test}} \tag{2}$$

where N_{test} is the total number of testing data samples, $\mathcal{D}_W(y_j, \theta_{adv})$ is a watermark detection function in which $\mathcal{D}_W(y_j, \theta_{adv}) = 1$ if y_j is a watermarked output of θ_{adv}, $\mathbb{I}$ is the indicator function in which $\mathbb{I}(x) = 1$ if x is True and $\mathbb{I}(x) = 0$ if x is False. Intuitively, Eq. 2 reflects the IP checker's failure rate in detecting watermarked outputs from a model trained on watermarked data of the cloud-hosted model. *Second*, to evaluate the model utility, we consider 1) Perplexity (PPL) for the text generation task, which measures how well a model predicts next tokens and is commonly used to evaluate language models [36] and 2) Average accuracy across topics for the MMLU downstream task. *Third*, to further illustrate the quality of generated texts, we provide qualitative assessment for examples of prompts and corresponding outputs from different WM schemes of two LLMs.

4.4 Experimental Results

ϵ-STEAL against Existing Watermarking Techniques. Fig. 3 illustrates the Attack SR and PPL of ϵ-STEAL as a function of noise scales on the LLaMA-2 and Mistral with four WMs. As the noise scale increases, the Attack SR of ϵ-STEAL also increases, while PPL exhibits a corresponding increase, but remaining low and comparable to those of Baseline and the Original model (where lower is better). For instances, with the KGW on the Mistral, the Attack SR is 89.28% at a noise scale of 0.001 and it increases to 96.95% when the noise scale is 0.1. Meanwhile, the PPL slightly increases 1.12 from 3.61 to 4.73 difference between the noise scales. We observe the similar trend with the EXP and SIR, which show increases of $1.72\% - 6.14\%$ in Attack SR and $9.08 - 23.41$ in PPL. This indicates that ϵ-STEAL is more stealthy with the KGW, as it achieves a higher Attack SR while having a subtle impact on PPL. Intuitively, when a token embedding is perturbed, greater noise leads to more significant perturbation, making it more challenging for IP checkers to distinguish whether WMs are present, thereby increasing the Attack SR. Specially, in the SIR and EXP, which depend on the semantics of preceding tokens and the sampling process, perturbing token embeddings can adversely affects model utility.



(a) AttackSR (LLaMA-2)

(b) Perplexity (LLaMA-2)

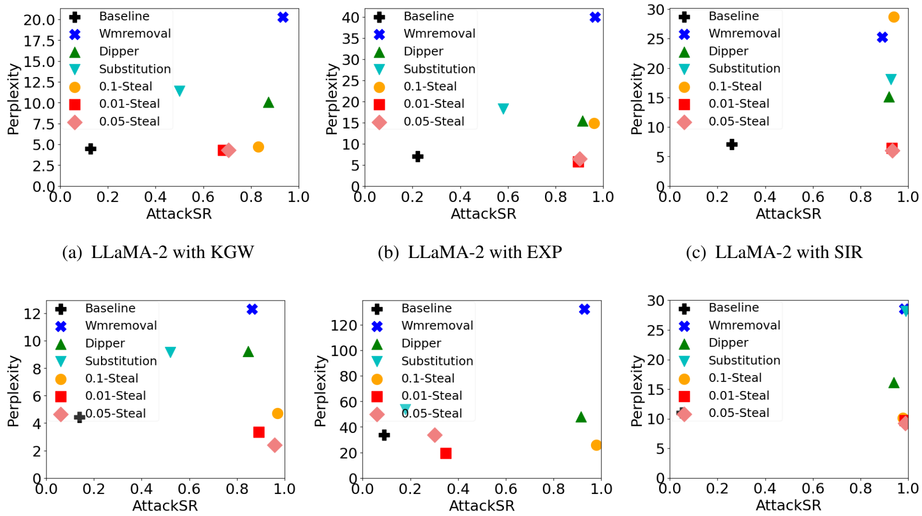
(c) AttackSR (Mistral)

(d) Perplexity (Mistral)

Fig. 3: ϵ-Steal results. (*B is Baseline*)

Accuracy (%)		LLaMA-2	Mistral
Original		46.70	58.90
KGW	Baseline	44.50	58.60
	ϵ-Steal-0.001	44.10	42.30
	ϵ-Steal-0.01	43.00	39.30
EXP	Baseline	45.90	58.70
	ϵ-Steal-0.001	45.50	47.70
	ϵ-Steal-0.01	43.30	29.20
Dipper		9.60	

Table 1: MMLU Accuracy across LLMs and WMs.



(a) LLaMA-2 with KGW

(b) LLaMA-2 with EXP

(c) LLaMA-2 with SIR

(d) Mistral with KGW

(e) Mistral with EXP

(f) Mistral with SIR

Fig. 4: Attack SR and Perplexity across LLMs, WMs, and attacks.

This is due to noise accumulation across tokens or disruptions in the mapping with pseudo-random number sequences, leading to increased PPLs.

ϵ-Steal and Existing Model Stealing Attacks. Fig. 4 compares the performance of our ϵ-Steal with three other attacks on the Baselines. If the attacks cluster in the bottom right corner, they are considered effective. As shown, ϵ-Steal achieves a high Attack SR while maintaining low PPL comparable to that of the Baseline. For instances, with a noise scale of 0.1, ϵ-STEAL achieves 83.14% Attack SR with a PPL of 4.71, compared with $4.46$for LLaMA-2 with KGW. Similarly, in Mistral with KGW, ϵ-STEAL even achieves 96.95% Attack SR at 4.73 PPL, compared with 4.43 of the Baseline. Meanwhile, other attacks exhibit low Attack SR but usually high PPL, indicating a substantial impact on the model and making them more detectable. For instance, WMremoval achieves a high Attack SR of 95.03% for the LLaMA-2 with KGW, but it significantly increases PPL, up to 34.16. In addition, WMremoval is computationally expensive as it requires paraphrasing and quality checks for every possible token to produce high-quality non-watermarked outputs. Dipper encounters similar issues, where it achieves a high Attack SR of 87.37% at the cost of a 10.09 PPL. Additionally, it notably affects the model utility of downstream tasks (Table 1). Substitution attack performs the worst among all, exhibiting either high PPL or low Attack SR across all settings. The randomness in replacing tokens causes significant changes in semantics, reducing overall model utility.

Impacts of LDP in ϵ-STEAL. Fig. 3 reveals that as the noise scale increases, the Attack SR also increases. This observation is consistent across watermarking techniques and LLMs. Basically, as more noise is introduced, token embeddings become more obfuscated. Under the guarantees of LDP, the differences in the logits distribution of token generations or the sampling process is reduced, making it more challenging for IP checkers to detect. Therefore, ϵ-STEAL successfully bypasses the IP checkers.

ϵ-STEAL in MMLU Downstream Task. In this downstream experiment, we evaluate the impact of ϵ-STEAL on MMLU dataset using three settings, including *1) Original* performance of LLMs, *2) Baseline* WMs without any attacks, and *3)* our ϵ-STEAL attacks on the Baseline with noise scales $\sigma \in [0.001, 0.01]$. As shown in Table 1, ϵ-STEAL maintains model utility with only a subtle drop in accuracy compared with the Original LLMs and Baseline performances. For the LLaMA-2, our attack preserves a good performance consistently for all WMs with a small drop of $1.5 - 2.6\%$ accuracy at noise scale is 0.01. Meanwhile, the Mistral is more sensitive, with $19.30 - 29.5\%$ drop at noise scale is 0.01 and $10 - 16.30\%$ at noise scale of 0.001. It is important to note that MMLU characterized by low entropy (multiple-choice answers), is highly sensitive to noise. In contrast, Dipper, which is the most effective attack from aforementioned experiments, shows poor performance in this task with only 9.6% accuracy.

Semantic Preservation of ϵ-STEAL Outputs. Furthermore, to qualitatively assess the quality of generated texts under our ϵ-STEAL attack, we illustrate side-by-side examples of real prompts, outputs across WMs and our attacked outputs in three settings: 1) original prompt, 2) baseline outputs, and 3) watermarked outputs under our ϵ-STEAL attack. Table 2 presents examples with the LLaMA-2 with KGW (the first row), EXP (the second row), and SIR (the third row). More examples of Mistral with different types of WMs, and other attacks will be provided in Table 3 (Appendix C).

We observe that ϵ-STEAL attack retains the semantic meaning, compared with the watermarked outputs. For instances, our attack offers lightly modified words but still keeps the overall semantic meanings, such as replacing “elections in the city" with “polls in Delhi" or “We have to show that the BJP’s victory" for “We cannot ignore the BJP’s victory". The next two examples with the EXP and SIR also follow a similar pattern.

Connection between ϵ-STEAL and LLMs. To shed light on the discrepancy between ϵ-STEAL results on two LLMs, it is important to investigate the distribution of embedding

Table 2: Selected prompts and watermarked outputs with and without ϵ-STEAL. Green texts indicate similarities, while red texts highlight differences.

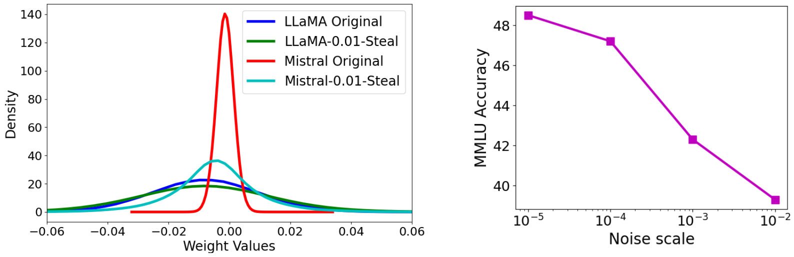
Prompt	**Watermarked (W)**	**Watermarked under ϵ-Steal ($\sigma = 0.01$) (A)**	**PPL (W)**	**PPL (A)**
... do that for the MCD polls, " said a senior party leader. Another reason behind not targeting Modi in the MCD polls is the massive victory the B…	JP achieved in the 2014 Lok Sabha polls in Delhi , where the party won all seven seats. " We have to take into account the ground reality and the political equation in the city. We can not ignore the BJP 's victory in the 2014 Lok Sabha [...continues]	JP registered in the 2014 Lok Sabha elections in the city. " We have to convince the people of Delhi that why we are the best option to govern the city. We have to show that the BJP 's victory in the 2014 Lok Sabha elections was a [...continues]	3.33	3.67
… Daniels, who was born Stephanie Clifford, was paid 130,000 by Cohen after she signed a nondisclosure agreement barring her from talking about her alleg...	ed affair with Trump. Trump denies the allegation. Cohen has admitted making the payment to Daniels, which he said was done to protect Trump' s campaign from the allegations. It' s possible that the reimbursement payment revealed in the financial disclosure may [...continues]	ed affair with Trump. Cohen recently revealed that Trump personally reimbursed him for the payment to Daniels. As a result of the payment to Cohen, the government ethics office sent a reminder to Trump that he must disclose in his annual financial disclosure report any payments made [...continues]	4.12	5.05
… other shows, it's a GoPro on a windshield, " Foley said referring to Ride Along. " I think if this was a show that was on...	a traditional television platform, we would be able to do more with it. As it stands, it 'll be a while before we do any new episodes of Holy Folesy! " Foley also addressed why his daughter Noelle is n't pursuing a career in WWE despite [...continues]	a major network television, we would be able to do more with it. I think we would be able to have a bigger budget and be able to do some cool things with it. " Foley also talked about why his daughter Noelle is n't pursuing an [...continues]	8.12	8.23

weights before and after adding noise to such layer of LLMs. As illustrated in Fig. 5 a), the embedding layer weight distribution of Mistral is sharply peaked with small standard deviation of 0.0027, whereas LLaMA-2's distribution is flatter and exhibits more variation at 0.01681, as $6.22\times$ as high, making Mistral more sensitive to noise than LLaMA-2. Consequently, applying the same noise scale affects Mistral more significantly, distorting the original distribution curve as shown in the figure.

This sensitivity is evident in the AttackSR and MMLU results, where Mistral shows a huge performance gap across noise scales, whereas LLaMA-2 remains relatively stable. For instance, in Table 1, MMLU of LLaMA-2 only drops 0.8% between noise scales of 0.01 and 0.001, while the figure for Mistral is 3.0%. These findings inform the choice of an appropriate noise scale to optimize attack effectiveness. For instance, Fig. 5 b) demonstrates that Mistral is able to maintain strong MMLU performance with small scale, achieving a high Attack SR of 85.99% and low PPL of 4.44 at a noise scale of 10^{-4}.

References

1. Aaronson, S., Kirchner, H.: Watermarking gpt outputs (2022)
2. Anthropic: Claude. `https://www.anthropic.com/` (2024)
3. Azure: `https://aka.ms/AzureMLModelInterpretability` (2021)
4. Baldassini, F.B., Nguyen, H.H., Chang, C.C., Echizen, I.: Cross-attention watermarking of large language models. In: ICASSP 2024. pp. 4625–4629. IEEE (2024)
5. Bluemix: Ai explainability 360 (2021), `http://aix360.mybluemix.net/`
6. Carlini, N., Paleka, D., Dvijotham, K.D., Steinke, T., Hayase, J., Cooper, A.F., Lee, K., Jagielski, M., Nasr, et al.: Stealing part of a production language model. arXiv (2024)
7. Christ, M., Gunn, S., Zamir, O.: Undetectable watermarks for lms. In: COLT (2024)



(a) LLaMA-2 and Mistral - Embedding Weight Distribution (b) MMLU Accuracy of ϵ-STEAL on Mistral

Fig. 5: ϵ-STEAL and Sensitivity of LLMs.

8. Conikee, C.: An ai attack that even sherlock holmes would be impressed by... (2024)
9. Dathathri, S., See, A., Ghaisas, S., Huang, P.S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., et al.: Scalable watermarking for identifying llm outputs. Nature **634**(8035) (2024)
10. Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., et al.: Documenting large webtext corpora: A case study on the colossal clean crawled corpus (2021)
11. Erlingsson, U., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC CCS. pp. 1054–1067 (2014)
12. Fu, J., Zhao, X., Yang, R., Zhang, Y., Chen, J., Xiao, Y.: Gumbelsoft: Diversified language model watermarking via the gumbelmax-trick. arXiv (2024)
13. Fu, Y., Xiong, D., Dong, Y.: Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. In: AAAI. vol. 38 (2024)
14. Giboulot, E., Teddy, F.: Watermax: breaking the llm watermark detectability-robustness-quality trade-off. arXiv (2024)
15. Google: Google gemini. `https://bard.google.com/chat/` (2024)
16. Gu, C., Li, X.L., Liang, P., Hashimoto, T.: On the learnability of watermarks for language models. arXiv (2023)
17. Gu, C., Huang, C., Zheng, X., Chang, K.W., Hsieh, C.J.: Watermarking pre-trained language models with backdooring." arxiv (2022)
18. He, Z., Zhou, B., Hao, H., Liu, A., Wang, X., Tu, Z., Zhang, Z., Wang, R.: Can watermarks survive translation? on the cross-lingual consistency of text watermark for lllms. arXiv (2024)
19. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding (2021)
20. Hou, A.B., Zhang, J., He, T., Wang, Y., Chuang, Y.S., Wang, H., Shen, L., et al.: Semstamp: A semantic watermark with paraphrastic robustness for text generation. NAACL (2024)
21. Hou, A.B., Zhang, J., Wang, Y., Khashabi, D., He, T.: k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. ACL (2024)
22. Hu, Z., Chen, L., Wu, X., Wu, Y., Zhang, H., Huang, H.: Unbiased watermark for large language models. ICLR (2024)
23. Huang, B., Wan, X.: Waterpool: A watermark mitigating trade-offs among imperceptibility, efficacy and robustness. arXiv (2024)
24. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., et al.: Mistral 7b (2023)
25. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T.: A watermark for large language models. In: ICML. pp. 17061–17084. PMLR (2023)
26. Kirchenbauer, J., Geiping, J., Wen, Y., Shu, M., Saifullah, K., Kong, K., Fernando, K., Saha, A., Goldblum, M., Goldstein, T.: On the reliability of watermarks for llms. ICLR (2024)
27. Krishna, K., Song, Y., Karpinska, M., Wieting, J., Iyyer, M.: Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. NeurIPS **36** (2024)

28. Kuditipudi, R., Thickstun, J., Hashimoto, T., Liang, P.: Robust distortion-free watermarks for language models. TMLR (2023)
29. Li, P., Cheng, P., Li, F., Du, W., Zhao, H., Liu, G.: Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In: AAAI. vol. 37 (2023)
30. Li, Z., Wang, C., Wang, S., Gao, C.: Protecting intellectual property of large language model-based code generation apis via watermarks. In: CCS. pp. 2336–2350 (2023)
31. Liang, C., Bian, Y., Deng, Y., Cai, D., Li, S., Zhao, P., Wong, K.F.: Watme: Towards lossless watermarking through lexical redundancy. In: ICLR (2024)
32. Liu, A., Pan, L., Hu, X., Li, S., Wen, L., King, I., Philip, S.Y.: An unforgeable publicly verifiable watermark for large language models. In: ICLR (2023)
33. Liu, A., et al.: A semantic invariant robust watermark for llms. ICLR (2024)
34. Liu, Y., Bu, Y.: Adaptive text watermark for large language models. arXiv (2024)
35. Lu, Y., Liu, A., et al.: An entropy-based text watermarking detection method. arXiv (2024)
36. Mikolov, T., Deoras, A., Kombrink, S., Burget, L., Černockỳ, J.: Empirical evaluation and combination of advanced language modeling techniques. In: ISCA (2011)
37. Niess, G., Kern, R.: Stylometric watermarks for large language models. arXiv (2024)
38. OpenAI: Openai api (2024), `https://openai.com/index/openai-api/`
39. Pan, L., Liu, A., He, Z., Gao, Z., Zhao, X., Lu, Y., Zhou, B., Liu, S., Hu, X., Wen, L., et al.: Markllm: An open-source toolkit for llm watermarking. arXiv (2024)
40. Pang, K., Qi, T., Wu, C., Bai, M.: Adaptive and robust watermark against model extraction attack. arXiv (2024)
41. Ren, J., Xu, H., Liu, Y., Cui, Y., Wang, S., Yin, D., Tang, J.: A robust semantics-based watermark for large language model against paraphrasing. NAACL (2024)
42. Takezawa, Y., Sato, R., Bao, H., Niwa, K., Yamada, M.: Necessary and sufficient watermark for large language models. arXiv preprint (2023)
43. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., et al.: Llama 2: Open foundation and fine-tuned chat models (2023)
44. Wang, L., Yang, W., Chen, D., Zhou, H., Lin, Y., Meng, F., Zhou, J., Sun, X.: Towards codable watermarking for injecting multi-bits information to llms. In: ICLR (2024)
45. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: 26th USENIX Security Symposium. pp. 729–745 (2017)
46. Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association **60**(309), 63–69 (1965)
47. Wouters, B.: Optimizing watermarks for large language models. arXiv (2023)
48. Wu, Y., others2: Dipmark: A stealthy, efficient and resilient watermark for llms. ICML (2024)
49. Xu, H., Xiang, L., Ma, X., Yang, B., Li, B.: Hufu: A modality-agnositc watermarking system for pre-trained transformers via permutation equivariance. arXiv (2024)
50. Xu, X., et al.: Learning to wm llm-generated text via reinforcement learning. arXiv (2024)
51. Xue, M., Wu, Z., Zhang, Y., Wang, J., Liu, W.: Advparams: An active dnn intellectual property protection technique via adversarial perturbation based parameter encryption. TETC (2022)
52. Yoo, K., Ahn, W., Kwak, N.: Advancing beyond identification: Multi-bit wm for llms (2024)
53. Zhang, H., Edelman, B.L., Francati, D., Venturi, D., Ateniese, G., Barak, B.: Watermarks in the sand: Impossibility of strong watermarking for generative models. ICML (2024)
54. Zhang, J., Peng, S., Gao, Y., Zhang, Z., Hong, Q.: Apmsa: Adversarial perturbation against model stealing attacks. IEEE TIFS **18**, 1667–1679 (2023)
55. Zhang, R., Koushanfar, F.: Emmark: Robust watermarks for ip protection of embedded quantized large language models. arXiv (2024)
56. Zhao, X., et al.: Provable robust watermarking for AI-generated text. In: ICLR (2024)
57. Zhu, C., Galjaard, J., Chen, P.Y., Chen, L.Y.: Duwak: Dual watermarks in llms. arXiv (2024)

A Appendix A: Proof of Theorem 1

Let $\mathcal{A}$ be a randomized mechanism, R_θ be model parameter space, ϵ denote privacy budget, and σ denote the noise associated with ϵ.

In ϵ-STEAL, we add an LDP noise $\mathcal{N}(0, \sigma)$ to the token embeddings, it is equivalent to add ϵ-LDP to each data sample. By the post-processing property of LDP, the adversary model θ_{adv} is also ϵ-LDP. Therefore, IP checkers cannot tell whether θ_{adv} was trained on the watermark data or not and fail to verify the IP of θ_{adv}. Or the difference between θ_{adv} is trained with and without watermarked data is bounded, as follows:

$$\begin{aligned} & P[\mathcal{A}(x + \mathcal{N}(0, \sigma), y^{wm} + \mathcal{N}(0, \sigma)) \in R_\theta] \\ \leq & \exp^\epsilon P[\mathcal{A}(x' + \mathcal{N}(0, \sigma), y'^{wm} + \mathcal{N}(0, \sigma)) = R_\theta] \end{aligned} \tag{3}$$

Therefore, Theorem 1 holds.

B Appendix B: Privacy Budget Calculation

In our ϵ-STEAL, we use a common LDP approach, which is a Laplace mechanism that adds Laplace noises into original embeddings of the model. The Laplace mechanism is defined as follows:

$$\mathcal{A}_{\mathcal{E}}(x, \mathcal{E}(x), \epsilon) = \mathcal{E}(x) + (L_1, L_2, \cdots, L_d) \tag{4}$$

where $\mathcal{E}(x)$ is an embedding of a token x, d is the size of embedding, and L_i is i.i.d. random variables draw from a Laplace noise that is centered at 0 (i.e., mean is 0) and is scaled with $\sigma = \frac{\Delta(\mathcal{E})}{\epsilon}$.

Given a noise scale σ, to compute the privacy budget ϵ, we need to compute $\Delta(\mathcal{E})$, as follows:

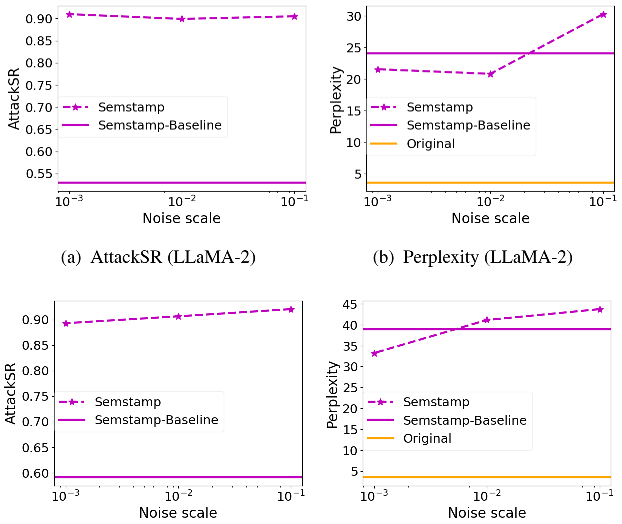
$$\Delta(\mathcal{E}) = \max_{\forall x, \tilde{x} \in N^d} \|\mathcal{E}(x) - \mathcal{E}(\tilde{x})\|_1 \tag{5}$$

With the LLaMA-2, we obtain $\Delta(\mathcal{E}) = 0.3$, while with the Mistral, we obtain $\Delta(\mathcal{E}) = 0.05$.

Then, with noise scales of $\sigma \in \{0.001, 0.01, 0.05, 0.1\}$, the privacy budgets are $\epsilon = \frac{\Delta(\mathcal{E})}{\delta} = \{300, 30, 6, 3\}$ with the LLaMA-2 and $\{50, 5, 1, 0.5\}$ for the Mistral.

C Appendix C: Supplemental Results

ϵ-STEAL against Semstamp.



(a) AttackSR (LLaMA-2)

(b) Perplexity (LLaMA-2)

(c) AttackSR (Mistral)

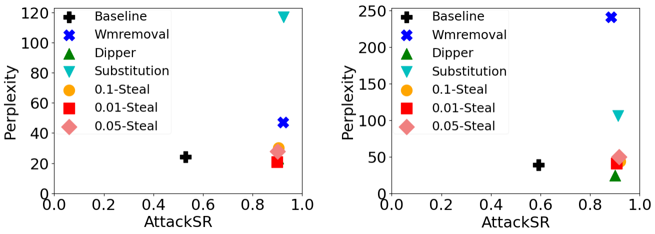
(d) Perplexity (Mistral)

Fig. 6: AttackSR and PPL of ϵ-STEAL on Semstamp and two LLMs.

For Semstamp, our observations of ϵ-STEAL attacks in Fig. 6 remain consistent with other WMs. As noise scales increase, the attack success rates also increase, while PPLs rise but remain comparable to the Baseline. However, a notable concern is the low Baseline WM detection rate, which results in unexpectedly high Attack SR even without attacks—reaching 52.97% for LLaMA-2 and 59.15% for Mistral.

ϵ-STEAL and Existing Model Stealing Attacks on Semstamp.

Fig. 7 demonstrates the effectiveness of our ϵ-STEAL attacks compared with other attacks on Semstamp. It is evident that ϵ-STEAL regardless of noise scales performs effectively on Semstamp as they gather at the bottom right corner, signifying high AttackSR but low PPLs.



(a) LLaMA-2 with Semstamp

(b) Mistral with Semstamp

Fig. 7: Attack SR and Perplexity across different LLMs and Attacks including ϵ-STEAL on Semstamp.

ϵ-STEAL in MMLU Downstream Task.

In Table 1, we present only KGW and EXP watermarking techniques, although we have experimented with all the four watermarking techniques considered in this paper, including SIR and Semstamp. With the SIR and Semstamp watermarking techniques, our attempts to apply SIR watermark for MMLU task using its publicly available trained watermark model resulted in a poor accuracy, only 27.90% for the LLaMA-2 and 20.90% for the Mistral without any attacks. The root cause of such poor results is because due to time and computational power limitation, we do not retrain their watermark model, therefore it may not adapt well to our data and settings. For Semstamp, since it generates sentences instead of single token at once, it is considered not suitable for the MMLU, which involves question-answering tasks.

Semantic Preservation of ϵ-STEAL Outputs.

In the following section, we present additional examples from the Mistral, as shown in Table 3a. These examples demonstrate the outputs of our ϵ-STEAL attack, highlighting the differences while preserving the overall semantic meaning. For example, in the first row, the attack replaces“office has talked to Attorney General Jeff Session” with “team has talked with Jeff Sessions” or remove adverb “now”, maintaining the key information about the Russia investigation. In the second example, “20-percent of black students and 10-percent of Latino students in Boston are attending the city’s top 20 public school” shifts to“who applied to the city’s top schools were admitted” subtly altering the focus while retaining the core message that the mentioned students still got in top schools. Lastly, in the third instance, ϵ-STEAL modifies“shape of a new toy” to “shape, configuration, and appearance of a new product,” and changes the quantifier from plural to singular, “patent” and “variety”, preserving the discussion on patent types. These examples confirm that our attack slightly modifies wording without compromising the overall semantic content.

Table 3b provides a comparison of our ϵ-STEAL with the different attack methods used in this study. While the approach of our ϵ-STEAL attack has been detailed previously, Dipper and the Substitution attacks present alternative solutions. As per Dipper, since it works with sentence levels, it truncates and discards all incomplete sentences within the target output to attack and always starts paraphrasing full sentences only, resulting in a disrupted reading flow from input to output and causing small loss of information. For example, in the first row, while both ϵ-STEAL and Substitution attacks continue to generate “JB” to complete “BJP”, Dipper starts with a new sentence. Furthermore, due to Dipper’s reliance on reordering sentences, the provided snippet of its output within a limited token count shown in Table 3b could not fully show relevant benchmark against other attacks. For instance, in the second example, the sentence “The company announced that it would pay $\$50,000$ for the silence of this alleged affair.” introduces content beyond what is covered in the benchmark snippet. In contrast, the Substitution attack replaces texts by considering surrounding context, maintaining a high degree of similarity (more green portion) with the original watermarked text.

Table 3a: Selected prompts and watermarked outputs with and without ϵ-STEAL attack. The examples are from Mistral model under the KGW (the second row) and SIR (the first and third row) watermarks. Green texts indicate similarities, while red texts highlight differences.

Prompt	**Watermarked (W)**	**Watermarked under ϵ-Steal (SA)** ($\epsilon = 0.01$)	**PPL (W)**	**PPL (SA)**
... There have been a number of revelations this week related to the Russia investigation. Among them, that the special counsel's ...	office has talked to attorney general Jeff Sessions and that Robert Muller is now looking at sitting down with President Donald Trump. And, of course, news about a deal to temporarily end the government shutdown, but with no resolution [...continues]	team has talked with Attorney General Jeff Sessions, and that Robert Mueller is looking to sit down for an interview with President Trump. But with a government shutdown still hanging over everyone's heads , it can be easy to lose track of [...continues]	8.52	8.21
... Latino students as well saw less opportunity for access to quality schools than their white and Asian counterparts, " O'Brien said. The study found that more than 80-...	percent of black students and 70-percent of Latino students in Boston are attending schools that are considered low performing. The report also found that 20-percent of black students and 10-percent of Latino students in Boston are attending the city' s top 20 public schools . " The report [...continues]	percent of black students and 70-percent of Latino students who applied to the city' s top schools were not admitted. The report also found that 20-percent of black students and 10-percent of Latino students who applied to the city's top schools were admitted. The report [...continues]	2.87	2.03
...patent protects the functional aspects of an invention, such as a new machine. A design patent protects the ornamental appearance of an invention, such as the...	shape of a new toy. Plant patents protect, you guessed it, new varieties of plants. You apply for a patent with the United States Patent and trademark Office, but you can not put a patent symbol " ® " on your product until it is [...continues]	shape, configuration and appearance of a new product. A plant patent protects inventions of natural organisms such as a new variety of fruit tree. Once you determine what type of intellectual property you need, you can file a patent application with the U.S patent [...continues]	5.67	5.52

Table 3b: Output comparison among ϵ-STEAL and other attacks for the same input. The examples are from LLaMA-2 model under the KGW (the first row), EXP (the second row), and SIR (the third row) watermarks. Green texts indicate similarities, while red texts highlight differences.

Prompt	**Watermarked (W)**	**Watermarked under ϵ-Steal (SA)** ($\epsilon = 0.01$)	**Dipper**	**Substitution**
... do that for the MCD polls, ” said a senior party leader. Another reason behind not targeting Modi in the MCD polls is the massive victory the B . . .	JP achieved in the 2014 Lok Sabha polls in Delhi , where the party won all seven seats. “ We have to take into account the ground reality and the political equation in the city. We can not ignore the BJP ’s victory in the 2014 Lok Sabha [...continues]	JP registered in the 2014 Lok Sabha elections in the city. “ We have to convince the people of Delhi that why we are the best option to govern the city. We have to show that the BJP ’s victory in the 2014 Lok Sabha elections was a [...continues]	We have to study the ground reality and political situation in the city. We can’t ignore the sweeping victory of the BJP in the 2014 elections in Delhi. The party won all the seven parliamentary seats. [...continues]	JP participated in the 2014 Lok Sabha elections in Delhi, where the party won just two seats. "We need to take into account the political climate and the political situation in the city. We cannot support the BJP’s performance in the 2014 Lok Sabha elections [...continues]
. . . Daniels, who was born Stephanie Clifford, was paid 130,000 by Cohen after she signed a nondisclosure agreement barring her from talking about her alleg...	ed affair with Trump. Trump denies the allegation. Cohen has admitted making the payment to Daniels, which he said was done to protect Trump’ s campaign from the allegations. It’ s possible that the reimbursement payment revealed in the financial disclosure may have been [...continues]	ed affair with Trump. Cohen recently revealed that Trump personally reimbursed him for the payment to Daniels. As a result of the payment to Cohen, the government ethics office sent a reminder to Trump that he must disclose in his annual financial disclosure report [...continues]	The company announced that it would pay $ 50,000 for the silence of this alleged affair. Cohen admitted paying the money, saying it was in order to protect Trump’s campaign from the alleged affair. Trump denies the alleged affair. [...continues]	daniels spoke with Trump. trump denied the allegation. Cohen later admitted to the payments to Daniels, which he said were done to protect Trump’s family from the allegations. It’s possible that the cash payments mentioned in the full disclosure would [...continues]
. . . other shows, it’s a GoPro on a windshield, ” Foley said referring to Ride Along. “ I think if this was a show that was on...	a traditional television platform, we would be able to do more with it. As it stands, it ’ll be a while before we do any new episodes of Holy Folesy! ” Foley also addressed why his daughter Noelle is n’t pursuing a career in WWE despite [...continues]	a major network television, we would be able to do more with it. I think we would be able to have a bigger budget and be able to do some cool things with it. ” Foley also talked about why his daughter Noelle is n’t pursuing an [...continues]	Furthermore, Foley was asked why his daughter Nol was not continuing in the world of professional wrestling, despite her father’s rich career. It’s too early for her to become a wrestler, because the thing is that it’s [...continues]	a great wrestling platform, we would be happy to do something with it. as it stands, it’ll be a while before we have any more fans of holy Folesy!" Foley ##a explained why his daughter Noelle isn’t pursuing a career in WWE despite. [...continues]