`ExpShield`: Safeguarding Web Text from Unauthorized Crawling and LLM Exploitation

Ruixuan Liu
Emory University
ruixuan.liu2@emory.edu

Toan Tran
Emory University
viet.toan.tran@emory.edu

Tianhao Wang
University of Virginia
tianhao@virginia.edu

Hongsheng Hu
Shanghai Jiao Tong University
hongsheng.hu@sjtu.edu.cn

Shuo Wang
Shanghai Jiao Tong University
wangshuosj@sjtu.edu.cn

Li Xiong
Emory University
lxiong@emory.edu

***Abstract*—As large language models increasingly memorize web-scraped training content, they risk exposing copyrighted or private information. Existing protections require compliance from crawlers or model developers, fundamentally limiting their effectiveness. We propose `ExpShield`, a proactive self-guard that mitigates memorization while maintaining readability via invisible perturbations, and we formulate it as a constrained optimization problem. Due to the lack of an individual-level risk metric for natural text, we first propose *instance exploitation*, a metric that measures how much training on a specific text increases the chance of guessing that text from a set of candidates—with zero indicating perfect defense. Directly solving the problem is infeasible for defenders without sufficient knowledge, thus we develop two effective proxy solutions: single-level optimization and synthetic perturbation. To enhance the defense, we reveal and verify the memorization trigger hypothesis, which can help to identify key tokens for memorization. Leveraging this insight, we design targeted perturbations that (i) neutralize inherent trigger tokens to reduce memorization and (ii) introduce artificial trigger tokens to misdirect model memorization. Experiments validate our defense across attacks, model scales, and tasks in language and vision-to-language modeling. Even with privacy backdoor, the Membership Inference Attack (MIA) AUC drops from 0.95 to 0.55 under the defense, and the instance exploitation approaches zero. This suggests that compared to the ideal no-misuse scenario, the risk of exposing a text instance remains nearly unchanged despite its inclusion in the training data.**

I. Introduction

Building datasets for large language models (LLMs) increasingly depends on crawling and parsing publicly available web content. For instance, OpenAI's GPT-3 [1] was trained on diverse sources including Wikipedia, Common Crawl, books, and articles. However, public accessibility does not imply unrestricted usage rights for AI training [2], [3]. A critical concern is that LLMs can memorize and reproduce verbatim copyrighted or sensitive content [4], undermining both model generalization and raising serious ethical and legal issues regarding privacy [4], copyright [5], and context collapse [6].

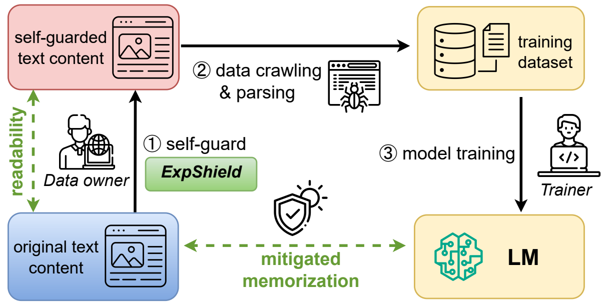


Fig. 1: Overview of web-text protection. Data owner $\mathcal{O}$ uses `ExpShield` to transform original text into protected version before web release. Model trainer $\mathcal{T}$ crawls the protected content for LM training. `ExpShield` mitigates verbatim memorization and data leakage in trained LMs.

Protecting content from unauthorized use faces two fundamental challenges. First, preventing web crawling is inherently difficult due to the open nature of the internet. Sophisticated crawlers can circumvent standard protections by ignoring `robots.txt` directives, mimicking human browsing patterns, rotating IP addresses, and leveraging distributed networks [7]. Second, existing defenses against verbatim memorization require cooperation from third parties who may not be incentivized to comply. Data-level protections such as deduplication [8] and scrubbing [9] depend on data curators; training-level defenses like differentially private (DP) training [10] and model alignment [11] rely on model developers; and inference-level controls such as output filtering [12] depend on model curators.

To address these limitations, we propose a self-guard `ExpShield` that empowers data owners with direct control at release time, as shown in Figure 1. This approach aims to mitigate memorization risk preemptively when the data is misused in LLM training. Unlike existing countermeasures, our self-guard operates independently without requiring third-party

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.240011
www.ndss-symposium.org

compliance. Moreover, instead of applying a one-size-fits-all defense, it protects each individual text instance independently.

We formulate the individual-level protection as a constrained bi-level optimization framework with two critical constraints for practical deployment. The *main objective* is to find a text perturbation that minimizes the adversary's advantage of inferring the protected text when the perturbed version has been used for training. The *readability constraint* requires that perturbations preserve semantic integrity and rendering consistency for legitimate users—existing privacy-preserving methods [13], [14], [15] that replace or delete content are unsuitable as they fundamentally compromise text readability. The *budget constraint* limits perturbation overhead and ensures normal users' experience.

Evaluating and solving the individual-level defense requires a rigorous metric that explicitly quantifies the privacy risk increase caused by model training on protected text instances. However, enumerating all possible adversaries is impractical, and existing metrics from privacy attacks [16] and memorization studies [17] operate at the dataset level or assume uniform risk distributions, failing to account for the inherent variation in memorization susceptibility across natural language. To address this fundamental limitation, we propose *instance exploitation*—a novel metric that isolates sample-specific memorization from model generalization by calibrating against an informed baseline that has access to all training data except the target instance.

The bottleneck of solving the defense problem is the limited capabilities of data owners, who typically lack access to training algorithms, datasets, or target models, which makes direct bi-level optimization intractable. Based on previous success in reformulating bi-level problem to single-level [18] or optimization-free solution [19], we develop two practical proxy solutions: single-level optimization that replaces the target model with open-source proxy models, and optimization-free solution that creates training shortcuts by injecting synthetic perturbation in text.

To instantiate the two solutions effectively, we investigate the fundamental mechanisms underlying memorization in autoregressive language modeling. Our key insight is the memorization trigger hypothesis: specific tokens disproportionately drive the model to rely on memorization over generalization. We identify memorization triggers as tokens with low prediction confidence under an open-source pre-trained model. Intuitively, the low confidence suggests that these tokens are unpredictable given their usual context and stand out as anomalies. We verify that such tokens often act as distinctive or rare patterns that the model tends to memorize verbatim if used for training.

Leveraging this insight, we design targeted perturbations that (i) neutralize inherent memorization triggers through strategic placement of imperceptible elements, and (ii) introduce artificial triggers as adversarial pitfalls that further misdirect the model's memorization on the protected text. To satisfy the readability and budget constraints, we use invisible Unicode characters and CSS styling to maintain perfect visual fidelity while maximizing defensive efficacy within budget constraints. We demonstrate our defense mechanism using a fictional webpage example[1].

Contributions. Our contributions are summarized as follows:

1) We formalize individual text protection as constrained bi-level optimization minimizing adversarial advantage while preserving readability. Given data owners' limited capabilities, we develop two practical solutions: synthetic perturbations and single-level optimization with proxy models.
2) We propose instance exploitation, a novel privacy metric that quantifies the individual-level memorization risk by calibrating sample-specific exposure against an informed adversarial baseline with access to all other training data, enabling principled design and evaluation of instance-level defenses.
3) We propose the memorization trigger hypothesis: tokens exhibiting low prediction confidence in general models are the key drivers of memorization. This hypothesis is subsequently validated across diverse language models. Leveraging this insight, we design targeted perturbations that neutralize inherent triggers while introducing artificial triggers as adversarial pitfalls.
4) Evaluation across various language and vision-language models (124M to 7B parameters) shows that `ExpShield` significantly reduces the extraction success rate over 10^5 attempts and MIA AUC to near-random (0.55) against an informed attack with a privacy backdoor, achieving near-zero instance exploitation with robustness against detection and adaptive scenarios.

II. Threat Model and Preliminaries

A. Threat Model

As shown in Figure 1, we consider two main parties in the web-text protection problem: the data owner $\mathcal{O}$ and trainer $\mathcal{T}$.

Owner/Defender $\mathcal{O}$ (Data Guarding): The content owner controls the release of their textual data and seeks to mitigate sample-specific memorization by any LLMs when the released content is subsequently misused for unauthorized training. The owner cannot foresee potential attacks and has no access to training data or algorithms of potential target model. The defense targets original content that has not been widely replicated across external sources, ensuring the protection focuses on genuinely unique material. Crucially, $\mathcal{O}$ does not seek to enhance or degrade overall model performance through the released content. This fundamental distinction separates our work from existing unlearnable examples [18], [20], [19], which explicitly aim to impair the model's test performance.

Trainer/Misuser $\mathcal{T}$ (Crawling and Training): This entity systematically crawls web pages to construct training datasets and optimize language model performance. As detailed in Table I, we model $\mathcal{T}$ as a *moderate* actor that disregards standard crawling protocols (e.g., ignoring `robots.txt` directives) and trains models without implementing privacy-preserving defenses [10], [8]. Critically, $\mathcal{T}$ does not actively attempt to bypass self-guard mechanisms or deliberately amplify data leakage risks [21], [22], [23], as such adversarial behavior

[1] Available at: https://github.com/Emory-AIMS/ExpShield-demo

TABLE I: Comparison of trainer assumptions and privacy impact with our defense: We target moderate trainers who prioritize data usability without active bypass attempts, as aggressive trainers face prohibitive legal risks and implementation costs for marginal benefits. (✓: Yes, ×: No; N/A: Depends on the bypassing level; Limited: Our defense is not specifically designed for aggressive trainers, but shows robustness against active bypass in Section V-D; Color coding: Positive, Negative)

Assumption for $\mathcal{T}$	**Crawling Behavior and Cost**				**w/ ExpShield**		**Applicable?**
	Follow Protocol?	Active Bypass?	Cost	Data Usability	Protection Range	Risk Level	
Conservative	✓	×	No	Low	All	No Risk	No hurt
Moderate (Our Focus)	×	×	No	High	Targeted	Low	Helpful
Aggressive	×	✓	High	High	N/A	N/A	Limited

would incur substantial computational and legal costs. This assumption reflects practical reality, where data misuse typically occurs through negligence rather than malicious intent [24].

While not targeted in this paper, we also describe two other possible $\mathcal{T}$'s. A *conservative* $\mathcal{T}$ that adheres to all crawling rules poses no data risk. However, the resulting reduction in training data quantity and diversity compromises usability from the trainer's perspective. Conversely, an *aggressive* $\mathcal{T}$ actively bypasses self-guards by detecting and filtering them. Yet, perfectly stripping self-guards without damaging normal text requires significant effort, and the heightened data leakage risk for owners also exposes $\mathcal{T}$ to substantial legal consequences.

B. Background of Language Models

Model Training. Contemporary transformer-based language models [25], [26] employ autoregressive training in both pre-training and fine-tuning phases. Text from each data owner is tokenized into a sequence $\mathbf{x} = (x_1, x_2, \ldots, x_t)$ of length t, with the dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ comprising data from n owners. The model's objective is to predict the next token x_{t+1} given the preceding context $(x_1, x_2, \ldots, x_t)$. Training minimizes the negative log-likelihood objective over T tokens:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T} \log f_\theta(x_t|x_{<t}), \tag{1}$$

where $f_\theta(x_t|x_{<t})$ denotes the conditional probability from model θ's softmax output, and $x_{<t}$ represents the prefix context.

Content Leakage Risk. In the inference phase, the trained model generates a new text by iteratively sampling $\hat{x}_t \sim f_\theta(\cdot|x_{<t})$. However, previous works show that models can memorize specific training data. For example, an adversary can efficiently extract training data by querying the target LMs without prior knowledge [24], and the extraction rate increases with more attack attempts [27]. Additionally, membership inference attack (MIA) [28], [16] still stands as a widely used auditing technique with a transparent random data split. Beyond membership identification, MIA is closely related to data extraction [4], [24]. Thus, we consider and evaluate data leakage risk with both data extraction and MIA.

III. Individual Text Protection

A. Problem Definition

Given the limited capabilities of the data owner ($\mathcal{O}$ as introduced in Section II-A), the only viable self-guard to mitigate future potential leakage through any model trained on the web text is to embed perturbation in released webpage source code. More specifically, $\mathcal{O}$ crafts the original web text $\mathbf{x}_i$ with a perturbation δ_i and releases the guarded text $\mathbf{x}_{\delta_i} = \delta_i \circ \mathbf{x}_i$. We formulate the construction of the guarded text as a constrained optimization problem:

$$\begin{aligned} \min_{\delta_i} \quad & \mathrm{Adv}(\mathbf{x}_i; \theta^*_{\delta_i}, \mathcal{A}) && (2)\\ \text{s.t.} \quad & \theta^*_{\delta_i} \in \arg\min_{\theta} \mathcal{L}(D_{\backslash \mathbf{x}_i} \cup \mathbf{x}_{\delta_i}; \theta), && (3)\\ & \mathrm{Multiset}(\mathbf{x}_i) \subseteq \mathrm{Multiset}(\mathbf{x}_{\delta_i}), && (4)\\ & \mathrm{EditDist}(\mathbf{x}_i, \mathbf{x}_{\delta_i})/|\mathbf{x}_i| \le b, && (5) \end{aligned}$$

where $\theta^*_{\delta_i}$ is the model trained on the released text $\mathbf{x}_{\delta_i}$.

1) Main Objective for Defense: The main goal of individual text protection in Equation (2) is to minimize the adversary's advantage $\mathrm{Adv}(\cdot)$ on the protected text $\mathbf{x}_i$ given the trained model $\theta^*_{\delta_i}$ and the attack $\mathcal{A}$. The attack $\mathcal{A}$ can be membership inference attack, data extraction or other variants of attacks. And $\mathrm{Adv}(\cdot)$ represents the normalized advantage in the success rate of guessing the secret via attack $\mathcal{A}$ over a baseline guess. For example, MIA advantage [28] is defined as $\mathrm{Adv}(x_i) = 2\Pr[\hat{b}_i = b_i] - 1$ where b_i is the real membership, $\hat{b}_i$ is predicted by $\mathcal{A}_{\text{MIA}}$ with a baseline success rate $1/2$.

2) Constraint on Perturbation Operation: The constraint in Equation (4) ensures that $\mathbf{x}_{\delta_i}$ maintains readability and semantic accuracy for normal web browsers by preserving all text of the original $\mathbf{x}_i$. Specifically, δ_i must be an invisible augmentation instead of deleting or replacing characters in $\mathbf{x}_i$.

3) Constraint on Perturbation Length: Equation (5) introduces a length constraint that bounds the ratio between the edit distance and the original text length $|\mathbf{x}_i|$ by the perturbation budget b. This constraint limits rendering overhead for normal users.

Our problem formulation for individual text protection differs from previous works. While unlearnable examples [20], [18], [19] use similar bi-level optimization to degrade test performance for image tasks, we target training data leakage reduction for language models and do not seek to degrade test performance. Another work [15] extends bi-level minimization [20] for text protection but distorts original text through replacement-based perturbations. In contrast, we preserve readability via Equation (4), which is more challenging but necessary for web content.

TABLE II: Individual privacy metrics/scores comparison.

Metrics/Scores	Instance Level?	Standardized?	Calibrated?	Natural instance?
MIA-Loss [28]	✓	×	×	✓
MIA-LiRA [16]	✓	×	✓	✓
Canary Exposure [17]	✓	✓	×	×
Instance Exploitation	✓	✓	✓	✓

B. Evaluating the Individual-Level Defense

Given the defined problem, we need an effective individual-level metric to evaluate how well a solution δ_i reduces the risk of the protected $\mathbf{x}_i$ as formulated in Equation (2). Thus, dataset-level metrics such as success rate or TPR [16] for MIA or extractable rate [4] for data extraction are inapplicable. Besides, it should be: a) **standardized** to generally compare risks among different architectures; b) **calibrated** to accurately capture the risk improvement caused by model training; and c) **efficient** to compute for large language models.

1) Standardizing Individual Risk via Log-Rank: Evaluating Equation (2) by considering all attacks is impractical, thus we need a proxy metric for various $\mathcal{A}$. Log-perplexity is a natural choice [17] as it represents negative log-likelihood of generating $\mathbf{x}$ given θ. A small value indicates high extraction probability and easier MIA identification of $\mathbf{x}$. For equal-length text, loss (defined in Equation (1)) and log-perplexity differ only by the factor the sequence length, so we use them interchangeably. Since loss is not standardized across models, we use exposure [17] to standardize $\mathcal{L}(\mathbf{x};\theta)$ by ranking against candidate losses from the same distribution, as in Definition 1.

Definition 1 (Exposure [17]). *Given the target model θ, let $\mathbf{rank}_\theta(\mathbf{x}) = |\{\mathbf{x}' \in \mathcal{D} : \mathcal{L}(\mathbf{x}';\theta) \le \mathcal{L}(\mathbf{x};\theta)\}|$ represent the rank of $\mathcal{L}(\mathbf{x};\theta)$ among losses of all samples in the domain $\mathcal{D}$. The exposure $\mathbf{E}_\theta(\mathbf{x})$ is defined as*

$$
\begin{aligned}
\mathbf{E}_\theta(\mathbf{x}) &:= \ln|\mathcal{D}| - \ln \mathbf{rank}_\theta(\mathbf{x}) && (6)\\
&= -\ln \frac{\mathbf{rank}_\theta(\mathbf{x})}{|\mathcal{D}|} && (7)\\
&= -\ln \mathbf{Pr}_{\mathbf{x}'\in\mathcal{D}}\left[\mathcal{L}_\theta(\mathbf{x}') \le \mathcal{L}_\theta(\mathbf{x})\right]. && (8)
\end{aligned}
$$

Essentially, Definition 1 quantifies the advantage of a model-informed adversary over a baseline adversary in a guessing game. The baseline attack $\mathcal{A}^{\text{unif}}_{\text{base}}$ assumes uniform distribution over all candidates in $\mathcal{D}$ and requires on average $|\mathcal{D}|/2$ guesses to find $\mathbf{x}$. In contrast, the model-informed attack $\mathcal{A}_{\text{target}}$ leverages θ to prioritize candidates with lowest loss, requiring only $\mathbf{rank}_\theta(\mathbf{x})$ guesses on average. The exposure measures how much the knowledge of θ reduces the expected effort required for guessing a target secret $\mathbf{x} \in \mathcal{D}$.

2) Calibration with Informed Adversary: The exposure metric was originally designed for fixed-template random canaries, which follow uniform distribution. However, text has a non-uniform distribution, making the uniform baseline $\mathcal{A}^{\text{unif}}_{\text{base}}$ weak and leading to overestimated privacy risks. For example, commonly occurring text (e.g., phrases partially seen during pre-training) will artificially inflate privacy risk scores.

Algorithm 1 Privacy Game for Informed Inference

```
procedure Informed-Inference($\mathcal{T}, D_{\backslash x}, \mathcal{D}, \mathbf{x}$)
    $\tilde{x} \leftarrow$ ExpShield ($\mathbf{x}$) if self-guard; else $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$
    $\theta \leftarrow \mathcal{T}(D_{\backslash \mathbf{x}} \cup \{\tilde{\mathbf{x}}\})$ // Trainer trains target model
    $\theta_{\backslash \mathbf{x}} \leftarrow \mathcal{T}(D_{\backslash \mathbf{x}})$ // $\mathcal{A}^{info}_{target}$ trains reference model
    $\mathcal{A}^{\text{info}}_{\text{target}}$ sorts descending $\mathbf{x}_i \in \mathcal{D}$ with $\mathbf{Ex}(\mathbf{x}_i;\theta,\theta_{\backslash \mathbf{x}})$
    $\hat{\mathbf{x}} \leftarrow \mathcal{A}^{\text{info}}_{\text{target}}(D_{\backslash \mathbf{x}}, \mathcal{T}, \mathcal{D})$ // Guess from top candidates
    return $\mathcal{A}^{\text{info}}_{\text{target}}$ wins if $\hat{\mathbf{x}} = \mathbf{x}$; otherwise fails
```

To address this limitation, we propose a much stronger, informed baseline $\mathcal{A}^{\text{info}}_{\text{base}}$ inspired by the worst-case assumptions in differential privacy [29]. This baseline adversary knows all other training data except $\mathbf{x}$ and can train a reference model $\theta_{\backslash \mathbf{x}} \leftarrow \mathcal{T}(D_{\backslash \mathbf{x}})$ using the same training procedure $\mathcal{T}$. Since $\theta_{\backslash \mathbf{x}}$ and θ share identical training procedures (including initialization), they exhibit similar loss distributions. Thus, one optimal strategy for the informed baseline is prioritizing candidates with the lowest loss according to $\theta_{\backslash \mathbf{x}}$. We define instance exploitation by calibrating the target model's exposure against this informed baseline, as shown in Definition 2.

Definition 2 (Instance Exploitation). *Given two datasets D and $D_{\backslash \mathbf{x}}$ and models trained by $\mathcal{T}$ over the two datasets θ and $\theta_{\backslash \mathbf{x}}$, the instance exploitation* $\mathbf{Ex}$ *is defined as*

$$
\begin{aligned}
\mathbf{Ex}(\mathbf{x}; D, \mathcal{T}) &:= \mathbf{E}_\theta(\mathbf{x}) - \mathbf{E}_{\theta_{\backslash \mathbf{x}}}(\mathbf{x}) && (9)\\
&= \ln \frac{\mathbf{Pr}_{\mathbf{x}'\in\mathcal{D}}\left[\mathcal{L}_{\theta_{\backslash \mathbf{x}}}(\mathbf{x}') \le \mathcal{L}_{\theta_{\backslash \mathbf{x}}}(\mathbf{x})\right]}{\mathbf{Pr}_{\mathbf{x}'\in\mathcal{D}}\left[\mathcal{L}_\theta(\mathbf{x}') \le \mathcal{L}_\theta(\mathbf{x})\right]}. && (10)
\end{aligned}
$$

Essentially, Definition 2 measures the guessing advantage: it quantifies how much easier it becomes to identify $\mathbf{x}$ when the model is trained on it, compared to an informed baseline that knows all other training data. Mathematically, this corresponds to the ratio $\mathbf{rank}_{\theta_{\backslash \mathbf{x}}}(\mathbf{x})/\mathbf{rank}_\theta(\mathbf{x})$ between the expected number of guesses required by $\mathcal{A}^{\text{info}}_{\text{base}}$ and $\mathcal{A}_{\text{target}}$.

A higher instance exploitation value indicates greater adversarial advantage from training on $\mathbf{x}$. We define a perfect defense as achieving zero or negative instance exploitation, as formalized in Property 1.

Property 1 (Perfect Defense). *A defense mechanism is* perfect *with respect to a training algorithm $\mathcal{T}$ and domain $\mathcal{D}$ if, for any instance $\mathbf{x}$, the following holds:*

$$
\mathbf{Ex}(\mathbf{x}; D, \mathcal{T}) \le 0 \quad \textit{or, equivalently} \quad \mathbf{E}_\theta(\mathbf{x}) \le \mathbf{E}_{\theta_{\backslash \mathbf{x}}}(\mathbf{x}).
$$

Informed Attacks and Reducibility. The informed adversary assumptions in our instance exploitation metric naturally lead to stronger attack strategies. By leveraging the same knowledge (access to $D_{\backslash \mathbf{x}}$ and $\mathcal{T}$), an adversary can construct an enhanced attack $\mathcal{A}^{\text{info}}_{\text{target}}$ that prioritizes guessing on candidates with top instance exploitation scores rather than raw loss rankings, as formalized in Algorithm 1.

This provides theoretical justification for our metric through privacy game reducibility [30], [31]. When privacy game G_1 is reducible to G_2 (i.e., G_1 is at most as hard as G_2), any defense effective against G_1 also protects against G_2. Our exploitation

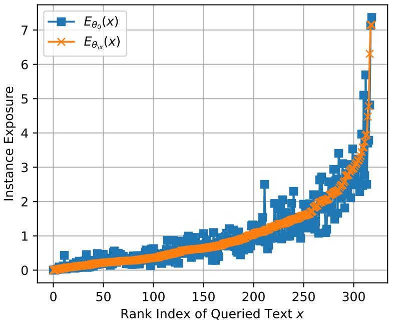


Fig. 2: Approximation on instance exploitation for efficient estimation. We take the pre-trained GPT-2 as θ_{pre} and fine-tune it with Patient dataset $D_{\backslash\mathbf{x}}$ which excludes the evaluated sample $\mathbf{x}$ for obtaining $\theta_{\backslash\mathbf{x}}$. The modeled skew-normal distribution matches the sampled log-perplexity perfectly because the Kolmogorov-Smirnov goodness-of-fit test [32] fails to reject the null hypothesis with $p \gg 0.1$.

metric captures the advantage of a highly informed adversary $\mathcal{A}^{\text{info}}_{\text{target}}$, which is similar to informed MIA attacks reducible to data extraction and standard MIAs. Thus, defenses that reduce exploitation scores provide protection against a broad spectrum of weaker attacks. This is particularly relevant in practice, as real-world adversaries rarely access the complete training dataset $D_{\backslash\mathbf{x}}$ and exact training procedure $\mathcal{T}$ assumed by the adversary in our metric.

Connection to DP. If a model is trained with differential privacy (DP), by the data-processing inequality, the instance exploitation of each training sample is bounded by its DP budget as shown in Lemma 1. The transition from DP to instance exploitation is one-directional because Definition 2 is an evaluation metric rather than an algorithm that provides theoretical DP guarantee.

Lemma 1. *If $\mathcal{T}$ performs (ϵ, δ)-DP training, the instance exploitation for any sample $\mathbf{x}$ in any D satisfies $\mathbf{Ex}(\mathbf{x}; D, \mathcal{T}) \leq \epsilon$ with a failure probability of δ.*

3) Approximation for Efficient Estimation: Now we discuss how to efficiently compute the proposed metric Definition 2.

Computing the exposure $\mathbf{E}_\theta(\mathbf{x})$ exactly requires computing losses for all samples in the domain $\mathcal{D}$, which is inefficient when $|\mathcal{D}|$ is very large. Given auxiliary data $D_{\text{aux}} \in \mathcal{D}$ not trained on θ, the loss distribution can be modeled as a skew-normal distribution [17] with mean μ, standard deviation σ, and skew α. The exposure can then be efficiently estimated as:

$$\mathbf{E}_\theta(\mathbf{x}) \approx \hat{\mathbf{E}}_\theta(\mathbf{x}) = -\ln \int_0^{\mathcal{L}_\theta(\mathbf{x})} \rho(x)dx, \tag{11}$$

where $\rho(x)$ is the continuous density function.

TABLE III: Summary of Proxy Solutions.

Main Objective	Constraint	Requirement	Defender's Capability
$\min \mathrm{Adv}(\mathbf{x}; \theta^*_\delta, \mathcal{A})$	Eq.(2)-(5)	$D_{\backslash\mathbf{x}}, \mathcal{T}, \mathcal{A}$	×
$\min \mathbf{Ex}(\mathbf{x})$	Eq.(2)-(5)	$D_{\backslash\mathbf{x}}, \mathcal{T}, D_{\text{aux}}$	×
$\max \mathcal{L}_{\theta^*_\delta}(\mathbf{x})$	Eq.(2)-(5)	$D_{\backslash\mathbf{x}}, \mathcal{T}$	×
$\max \mathcal{L}_{\theta_{\text{proxy}}}(\mathbf{x}_\delta)$	Eq.(3)-(5)	θ_{proxy}	✓(Our TP-OP)
Optimization-free	Eq.(3)-(5)	N/A; θ_{proxy} is optional	✓(Synthetic perturbation)

Additionally, computing exploitation requires training $\theta_{\backslash\mathbf{x}}$ for each target secret $\mathbf{x}$, which is inefficient for large models. For protection set $D_{\text{pro}} \subset D$, we approximate $\theta_{\backslash\mathbf{x}}$ by training on the remaining unprotected data:

$$\mathbf{Ex}(\mathbf{x}; D, \mathcal{T}) \approx \hat{\mathbf{Ex}}(\mathbf{x}; D, \mathcal{T}) = \mathbf{E}_\theta(\mathbf{x}) - \mathbf{E}_{\theta_{\backslash D_{\text{pro}}}}(\mathbf{x}). \tag{12}$$

When all secrets in D are protected, this equals using the initial model θ_{pre} in place of $\theta_{\backslash\mathbf{x}}$. As shown in Figure 2, $\mathbf{E}_{\theta_{\text{pre}}}(\mathbf{x})$ closely approximates the exact calibration $\mathbf{E}_{\theta_{\backslash\mathbf{x}}}(\mathbf{x})$ with minimal fluctuation from training randomness and cross-sample influence [33]. The approximation accuracy improves with smaller protection ratios $|D_{\text{pro}}|/|D|$ due to reduced inter-instance influence.

C. Challenges and Proxy Solution Overview

We now discuss how to solve the defined problem as the defender $\mathcal{O}$. Ideally, $\mathcal{O}$ should optimize the bi-level problem in Equation (2) for each possible adversary $\mathcal{A}$, which is impractical. Similarly, due to lack of capability, other alternative proxy objectives are hard to solve as summarized in Table III.

Given the key bottleneck of lacking access to $\mathcal{T}$ and $D_{\backslash\mathbf{x}}$, we propose two practical alternatives:

a) Single-Level Optimization: While θ^*_δ in Equation (3) is unpredictable, it operates on natural text and shares foundational knowledge with existing open-source models. We replace θ^*_δ with an accessible proxy θ_{proxy} and optimize $\max \mathcal{L}_{\theta_{\text{proxy}}}(\mathbf{x}_\delta)$ given the absence of $\mathcal{A}$ or D_{aux}. The intuition is that the perturbations with a high loss on the proxy model are likely to be abnormal patterns against general text and thus can encourage the target model to fit the shortcut. When a perturbation maximizes the loss on a proxy model, it is likely to mislead the target model. This single-level approach follows successful precedent in adversarial examples, where bypassing bi-level optimization produces effective poisons [18].

b) Synthetic Perturbation: Discrete optimization over vocabulary incurs substantial computational costs, particularly for long sequences $\mathbf{x}_\delta$ and large models θ_{proxy}. We propose a lightweight alternative using synthetic perturbations that naturally create training shortcuts, encouraging the model to fit δ rather than memorize $\mathbf{x}$. This approach builds on recent work demonstrating that synthetic patches can effectively replace bi-level optimized perturbations [19]. While lacking explicit optimization objectives, we verify in Section IV-B2 that synthetic perturbations implicitly encourage $\max \mathcal{L}_{\theta^*_\delta}(\mathbf{x})$.

We primarily employ synthetic perturbations for their efficiency and effectiveness, reserving single-level optimization (TP-OP) for scenarios with constrained perturbation budgets.

IV. Self-Guard against Exploitation

A. Invisibility Strategies

For Equation (4), we consider two strategies to hide perturbation in web page rendering: 1) *invisible style*, including

TABLE IV: Summary of ExpShield variants. OOV is out-of-vocabulary; pitfall means artificially created outlier tokens.

Methods	Perturb Location	Filling Strategy	Invisibility
UDP (§ IV-B1)	Deterministic	Uniform	Style
UNP (§ IV-B1)	Non-Deterministic	Uniform	Style
TP (§ IV-D1)	Mem. Trigger	Uniform	Style
TP-P (§ IV-D2)	Mem. Trigger	Outlier pitfall	Style
TP-OP (§ IV-D3)	Mem. Trigger	Optimized pitfall	Style
TP-OOV (§ IV-D4)	Mem. Trigger	OOV pitfall	Character

adjusting CSS properties like font size or absolute position for inserting random text; 2) *invisible character*[2], including zero-width and invisible whitespace characters [34]. Both invisible characters and styles have been leveraged in attacks [34], [35], while we use for defense purposes. We elaborate details with a simplified demonstration in Appendix C.

Robustness against Normal Pre-processing. Crawled content includes visible text, HTML tags, and formatting markers. Standard parsing tools like Beautiful Soup [36] decode entities and strip markup while preserving hidden DOM elements. We provide a demo [37] showing that `ExpShield` successfully embeds tokens that remain intact in text extracted by four popular web-scraping tools without changing page appearance.

Our defense is robust against other normal pre-processing by its design. For example, we avoid repeated patterns, so deduplication [38] poses no threat. And quality filtering [1] may trigger removal on the whole sentence, which enhances protection by preventing training entirely.

Robustness of Active Bypass. An adversarial $\mathcal{T}$ may attempt to bypass `ExpShield` by perfectly stripping self-guards while preserving original content. Assuming constant-time $O(1)$ verification per token, the time complexity is $O(T)$, and the tokenized sequence length T of all concatenated text can reach hundreds of billions [1]. Stripping invisible styles incurs larger constant overhead as they permit arbitrary vocabulary insertions, requiring additional operations to recover tokenization boundaries. While removing invisible characters (e.g., zero-width spaces) has less verification overhead, it maintains $O(T)$ complexity. A thorough character stripping is possible but it may degrade model robustness [39], [40].

Advanced bypass strategies involve embedding or perplexity analyses before manual perturbation removal. We demonstrate robustness against such detection-based attacks in Section V-D.

B. A Basic Random Perturbation for Defense

1) Uniform Random Augmentation: We start with a strawman perturbation with uniformly random tokens. Let $\mathbf{x} = \{x_1, x_2, \cdots, x_t\}$ denote the tokenized sequence with length of t, we randomly sample m augmented random noise tokens as $\delta = \{\delta_1, \delta_2, \cdots, \delta_m\} \in \mathcal{V}^m$ from a typical vocabulary set $\mathcal{V}$. And $m = \lfloor b * t \rfloor$ is limited by the perturbation budget b. We split random tokens into K pieces and insert them into K slots within the original $\mathbf{x}$, where $K \leq t - 1$. We create two versions with different inserting positions:

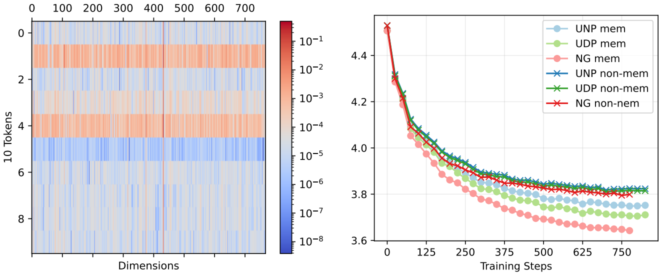


Fig. 3: (Left) Difference between embedding gradients of UDP and baseline without perturbation, with x-axis as the embedding dimension and y-axis as the protected token's sequence. (Right) Loss comparison between protected text (mem-samples) and non-member samples compared to the No-Guard (NG).

- **Uniform and Deterministic Perturbation (UDP)**: The token sequence is split into K equal-length blocks, with m random tokens inserted evenly.
- **Uniform and Non-deterministic Perturbation (UNP)**: K slots are randomly chosen and filled with m random tokens, which is a nondeterministic insertion.

2) Verifying the Implicit Objective of Perturbation: Though the synthetic perturbation such as UDP and UNP does not optimize towards an explicit objective, we now demonstrate it essentially encourages the implicit objective of $\max \mathcal{L}_{\theta^*_\delta}(\mathbf{x})$.

In Figure 3 (Left) with UNP as an example, the gradient of protected tokens' embeddings change drastically compared to the case without perturbation (NG) across embedding indices (horizontal lines) and dimensions (vertical lines), which directly influences model update and results in Figure 3 (Right). With moderate perturbation ($b = 1$), the influence on testing performance is trivial and the implicit target $\mathcal{L}_{\theta^*_\delta}(\mathbf{x})$ increases, indicating that the target model is less likely to generate $\mathbf{x}$.

According to the analysis in Table III, a larger loss on the target model $\mathcal{L}_{\theta^*_\delta}(\mathbf{x})$ is expected to lower the general risk proxy of exploitation $\mathbf{Ex}(\mathbf{x})$ as well as $\mathrm{Adv}(\mathbf{x}; \theta^*_\delta, \mathcal{A})$. We will demonstrate both degradations in Section V.

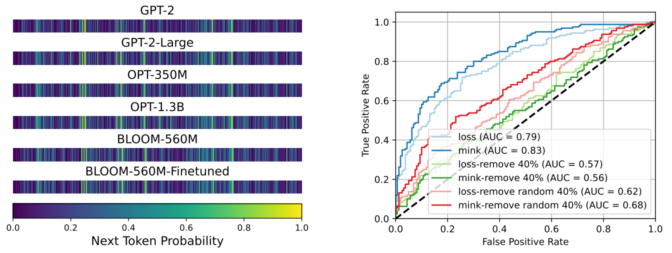
C. Memorization Trigger Hypothesis

To enhance defense efficacy, we first investigate how language models (LMs) memorize specific texts. Unlike generalization, memorization occurs when a model captures sample-specific patterns rather than generalizable features. From Equation (1), we observe that rare or challenging tokens—those with higher initial loss values—leave a stronger imprint on the trained model compared to others. This suggests that the model prioritizes memorizing these tokens over other easy tokens during training. Thus, we hypothesize that:

Hypothesis

The model's memorization of input $\mathbf{x}$ primarily stems from its retention of *hard-to-predict tokens* in $\mathbf{x}$—which we term ***memorization triggers***.

[2] https://invisible-characters.com



(a) Transferable Identification

(b) Attack-Defense Duality

Fig. 4: Memorization trigger hypothesis: tokens with higher loss on a proxy model are memorization triggers that enhance the sample-specific memorization.

Transferable Identification. We define *hard-to-predict tokens* as tokens assigned low probability by a general-domain pretrained model in their respective contexts. In implementation, we leverage an open-source pre-trained model as a proxy θ_{proxy} and select tokens whose prediction probability $f_{\theta_{\text{proxy}}}(x_t|x_{<t})$ belong to the K lowest-probability tokens in the sequence as a set $\mathbf{S}_K(\mathbf{x})$. Figure 4a reveals that models across scales and families have a consistent identification on memorization triggers identification, indicating an architectural independence between proxy and target models, which is a key advantage for data owners acting as defenders.

Attack-defense Duality. Then, we verify the existence of memorization triggers from both attack and defense perspectives with $K/|\mathbf{x}| = 0.4$. From the attack perspective, recent improvement on attack sheds similar light on the memorization trigger. As shown in Figure 4b, MinK [41] outperforms loss-based MIA by only aggregating losses over low confidence tokens in the given sequence, implying that attackers rely on the improvement of the model's prediction capability on outlier tokens. From the defense perspective, when memorization triggers are removed from the training data (as shown in Figure 4b), the MIA AUC drops to near random-guess level (≈ 0.56). However, removing random tokens with identical ratio ($K/|\mathbf{x}| = 0.4$) still maintains AUC 0.68, demonstrating the important role of memorization triggers in defense.

In summary, the memorization trigger hypothesis reveals the key mechanism for mitigating sample-specific memorization. In addition to being immediately applicable to proactive defense, this approach may also extend to later phases including preprocessing and training.

D. Targeting on Memorization Triggers

Based on above hypothesis, a natural idea to enhance self-guard is to focus on perturbing identified memorization tokens.

1) Targeted Perturbation: A simple extension is to insert random tokens right before the identified trigger tokens.

- **Targeted Perturbation (TP):** Instead of randomly sampling K slots for inserting perturbation, we first identify Top-K tokens in $\mathbf{x}$ with minimum prediction probabilities by a proxy model via $f_{\theta_{\text{proxy}}}(x_t|x_{<t})$. Then, we insert uniform tokens as in UNP to fill slots before each of K trigger tokens.

2) Outlier Tokens as Pitfalls: Furthermore, instead of interfering model learning on the *naturally* existed memorization triggers $\mathbf{S}_K(\mathbf{x})$ as in TP, we propose to create *artificial* memorization triggers to take the place of the original $\mathbf{S}_K(\mathbf{x})$ as pitfalls. By redirecting the model's optimization efforts toward these pitfall tokens, it mitigates model's memorization on the original $\mathbf{S}_K(\mathbf{x})$. Hence, we propose:

- **Targeted Perturbation with Pitfalls (TP-P):** After identifying memorization triggers $\mathbf{S}_K(\mathbf{x})$, we feed preceding tokens before each slot at position t to the proxy model θ_{proxy}, and select the token $\arg\min_{v\in\mathcal{V}} f_{\theta_{\text{proxy}}}(v|x_{<t})$ as pitfall token to fill the slot iteratively until spending all budget b.

3) Optimized Pitfalls: From previous methods, we notice that the usage of the proxy model is insufficient, because the perturbation is sampled or generated. Besides, instead of only considering the prefix of current token, we can also consider its context in the following variant.

- **Targeted Perturbation with Optimized Pitfalls (TP-OP)**: We first identify K memorization triggers via the θ_{proxy}. Then we optimize tokens to fill the position set $\mathcal{I}$ as follows.

Considering limited capabilities of defenders on training data and training algorithms as summarized in Table III, we reformulate the bi-level optimization as single-level optimization by substituting the target model with the proxy model θ_{proxy} trained on general text. By optimizing δ towards a maximized loss given θ_{proxy}, it creates a pitfall for the target model to fit during training. While on the contrary, a minimized loss can also help because the whole perturbed text is ignored by target model given the shortcut. The two intuitions correspond to $\max$ and $\min$ cases in previous work [18].

We employ Greedy Coordinate Gradient (GCG) [42] to optimize candidate tokens via the following objective, detailed in Algorithm 2:

$$\begin{aligned}\min_{\delta} \mathcal{L}_{\text{P}} = \min_{\delta} [&\beta_1 \mathbb{E}_{i\in\mathcal{I}} \mathcal{L}(\mathbf{x}_\delta[i]; \mathbf{x}_\delta[<i], \theta_{\text{proxy}}) \\ &+ \beta_2 \mathbb{E}_{i\in\mathcal{I}} \mathcal{L}(\mathbf{x}_\delta[i+1]; \mathbf{x}_\delta[<i+1], \theta_{\text{proxy}})], \end{aligned} \tag{13}$$

where coefficients $\beta_1, \beta_2 \in \{1, -1\}$ represent error-minimization or error-maximization strategies. By default, we use error-maximization with coefficient -1.

4) Out-of-Vocabulary (OOV) Tokens as Pitfalls: Previous methods perturb between tokens, leaving characters within original memorization triggers connected. We propose an enhanced perturbation by breaking common tokens into out-of-vocabulary tokens that are harder to predict. For example, inserting an invisible zero-width space in the word 'language' as 'lang\u200Buage' completely transforms the original token sequence from [16129] to [17204, 9525, 84, 496] when using the GPT-2 tokenizer. Given the set of all invisible characters C, we propose:

- **OOV-based Targeted Perturbation (TP-OOV)**, which first identifies memorization triggers as in Section IV-D, then randomly splits characters of each trigger token with a randomly sampled invisible character $c \in C$ within budget b.

Additionally, we can combine this with Algorithm 2 by replacing the candidate set $\mathcal{X}_i$ in Line 3 with set C, denoting

Algorithm 2 TP-OP: Optimizing Pitfalls with GCG

```
Require: Iterations $\tau$, batch size $B$, number of token candidates $k$, position index set before $K$ memorization triggers $\mathcal{I}$, token vocabulary $\mathcal{V}$, batch size $B$
Ensure: Optimized pitfall tokens $\{x_{\delta,i}\}_{i\in\mathcal{I}}$
    Randomly initialize inserted token embeddings $\{x_{\delta,i}\}_{i\in\mathcal{I}}$
    for $j \in [\tau]$ do
        for $i \in \mathcal{I}$ do // Compute Top-$k$ promising candidates given the gradient of token embedding $e_{x_i}$ where $i \in |\mathcal{V}|$
            $\mathcal{X}_i := \text{Top-}k(-\nabla_{e_{x_i}} \mathcal{L}_{\text{P}}(\mathbf{x}_\delta))$
        for $b = 1, \ldots, B$ do // Create a batch for searching
            $\tilde{x}^{(b)}_{1:m} := x_{1:m}$ // Initialize with current $n$ tokens
            $\tilde{x}^{(b)}_{i} := \text{Uniform}(\mathcal{X}_i)$, where $i = \text{Uniform}(\mathcal{I})$
        $x_{1:m} := \tilde{x}^{(b^\star)}_{1:m}$, where $b^\star = \arg\min_b \mathcal{L}_{\text{P}}(\tilde{x}^{(b)}_{1:m})$
```

this as TP-OOV++. In practice, invisible characters should be filtered from both inputs and outputs in inference APIs for safety [3], which actually favors our defense since the artificial memorization triggers of OOV tokens will never be triggered during inference.

V. Experiments

A. Experimental Setup

Tasks and Datasets. We conduct comprehensive evaluation on general language modeling and vision-to-language modeling (VLM) tasks. We include representative data sources requiring protection but potentially disclosed in web content: 1) Enron [43] with personal information (emails, names, medical records), 2) Patient [44] with domain knowledge (healthcare), 3) CC-News [45] with copyrighted work (news articles), and 4) IAPR-TC-12 [46] with natural images and descriptions for VLM tasks. Dataset details are in Appendix A.

For risk evaluation, we split the dataset at a ratio 1:4:4:1, with D for training the target model, D_{aux} for reference model or privacy backdoor, D_{non} as non-members, and D_{test} for testing. We split D by marking a fraction $r \in (0, 1]$ as protected (D_{pro}) and the rest as unprotected (D_{un}).

Models and Training Configuration. We evaluate on open-source models due to our requirement for per-sample losses and token-level probabilities—information unavailable through commercial APIs. We use GPT-2 as the proxy model for all tasks, showing architecture independence. We evaluate on models of different scales: GPT-2-124M [47], OPT-125M/350M [48], Llama2-7B [49] for LM, and BLIP-2-ViT (3.8B) for VLM. The VLM model processes image inputs and autoregressively generates text conditioned on preceding inputs.

Evaluation Configuration. For *individual-level* risk evaluation, we use the exploitation metric defined in Definition 2, and we approximate $\theta_{\setminus \mathbf{x}}$ with $\theta_{\setminus D_{\text{pro}}}$ as in Equation (12).

For *dataset-level* risk evaluation, we cover practical attacks including data extraction and membership inference attacks (MIA). For MIA, we evaluate using state-of-the-art threshold-based black-box MIAs with signals: 1) Loss [28] (model's loss on target samples), 2) Loss-Ref [4] (loss calibrated against reference model), 3) MinK [41] ($K\%$ tokens with lowest likelihood scores), and 4) Zlib [4] (loss normalized by compression size). We use 'user-level' (each sample belongs to one user) and 'sample-level' (documents chunked to full window size) evaluation, with the sample index as the user ID, except for Enron, which has explicit user IDs. For data extraction, we follow the recent work [27] to evaluate success of extracting T-length sub-sequences in D_{pro} over N trials.

For the *worst-case* risk evaluation, we follow recent works and assume a malicious and powerful model trainer who can manipulate a significant portion of $D_{\setminus \mathbf{x}}$ to insert privacy backdoor [22], [23] with details in Appendix D.

Evaluation Metrics. Following previous work [16], we measure MIA risk with AUC↓, true positive rate at low false positive rate (TPR@1% FPR↓) and ROC curves. We report bootstrapped metrics [50] for stability. At the individual-level, we report approximated exploitation $\hat{\mathbf{Ex}}(\mathbf{x}) \downarrow$ for each sample. Lower values indicate lower privacy risk. For utility cost, we report perplexity (PPL↓) on held-out test data, reflecting the implicit cost to model trainers since data owners are not obligated to maintain high performance.

Baselines. Supposing the model trainer is neutral and not motivated to perform training-phase defense, we compare `ExpShield` with the baseline of No Protection (NP) where users release their text contents without any protection. To fully investigate each proposed strategy, we evaluate all our variants in Table IV, with moderate perturbation budget b.

B. Effectiveness Evaluation

Effectiveness against MIAs. As shown in Table V, we compare MIA risks across different tasks and datasets. The gap between NP and our variants demonstrates overall defense effectiveness under identical training and attack pipelines for D_{pro}. Sample-level evaluation exhibits higher MIA metrics than user-level evaluation due to longer chunk lengths [51]. We intentionally train models with slight overfitting—large models with billions of parameters (Llama2-7B and BLIP2-ViT-3.8B) achieve an AUC of ≈ 1 for NP, representing the worst-case scenario for defenders.

Comparing UDP and UNP results confirms that non-deterministic perturbation outperforms the deterministic variant, as models can learn to ignore deterministic patterns.

We observe consistent TP improvements over UNP across different architectures, including Enron results where proxy and target models differ, confirming the model transferability demonstrated in Figure 4a of Section IV-C. Notably, transferability performs better for memorization identification than pitfall creation—TP-P shows slightly higher risk than TP for Enron. When proxy and target models share identical architectures (Patient dataset), replacing random tokens in TP with outlier pitfalls in TP-P yields only marginal improvements.

With perturbation budget $b = 0.4$, TP-OOV achieves superior defense effectiveness among all variants, primarily due to

[3] https://www.promptfoo.dev/docs/red-team/plugins/ascii-smuggling/

TABLE V: Membership inference evaluation with maximum metrics reported for across Loss-Ref, Loss, Min-K, Zlib due to space limit, leaving full results in Appendix. TPR is calculated at 1%FPR. BD indicates that the pre-trained model is backdoored to maximize the privacy risk of training data, i.e., BD represents the worst privacy leakage case.

MIA Level	Method	Patient GPT-2		Enron OPT-350M		Patient GPT-2 w/ BD		CC-News OPT-125M w/ BD		Patient Llama2-7B		IAPR-TC-12 BLIP2-ViT-3.8B	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Sample	NP	0.888	0.364	0.997	0.983	0.953	0.545	0.998	0.982	0.986	0.726	1.000	0.980
	UDP (b=0.4)	0.771	0.242	0.994	0.950	0.831	0.182	0.997	0.970	0.861	0.260	0.984	0.510
	UNP (b=0.4)	0.695	0.182	0.986	0.735	0.766	0.152	0.983	0.467	0.852	0.164	0.984	0.560
	TP (b=0.4)	0.686	0.182	0.979	0.621	0.765	0.182	0.978	0.580	0.856	0.219	**0.509**	0.010
	TP-P (b=0.4)	0.682	0.212	0.989	0.837	0.772	0.182	0.991	0.746	0.793	0.123	0.550	**0.000**
	TP-OOV (b=0.4)	0.594	0.091	0.892	0.254	0.587	0.091	0.890	0.083	0.753	0.082	0.551	**0.000**
	TP-OOV (b=1)	**0.590**	**0.060**	**0.684**	**0.119**	**0.550**	**0.076**	**0.621**	**0.053**	**0.630**	**0.055**	0.519	0.010
User	NP	0.676	0.047	0.987	0.585	0.741	0.047	0.966	0.035	0.936	0.452	0.974	0.377
	UDP (b=0.4)	0.617	0.039	0.968	0.439	0.649	0.047	0.948	0.035	0.749	0.096	0.901	0.057
	UNP (b=0.4)	0.598	0.039	0.933	0.269	0.622	0.039	0.912	**0.032**	0.740	0.082	0.907	0.140
	TP (b=0.4)	0.584	0.039	0.921	0.219	0.618	0.039	0.918	0.035	0.746	0.082	**0.511**	**0.003**
	TP-P (b=0.4)	0.588	0.039	0.951	0.282	0.619	0.039	0.923	0.035	0.667	0.068	0.541	0.007
	TP-OOV (b=0.4)	0.539	0.039	0.777	0.123	**0.542**	0.039	0.783	0.035	0.682	0.082	0.535	**0.003**
	TP-OOV (b=1)	**0.567**	**0.031**	**0.640**	**0.090**	0.567	**0.031**	**0.605**	0.035	**0.545**	**0.041**	0.523	**0.003**

memorization trigger identification and out-of-vocabulary subtokens. On GPT2 and BLIP2, TP-OOV with sufficient budget $b = 1$ approaches random performance, indicating that GPT-2 tokenizer-generated OOV pitfalls generalize across different target models. This effectiveness stems from shared lexical knowledge across language models, and tokens that are out-of-vocabulary in one LM typically remain so in others.

Effectiveness against Data Extraction. In Figure 5, we evaluate discoverable data extraction [27] risk using TP-OOV against unprotected baseline NP. We model realistic adversaries performing $N = 10^5$ extraction attempts on subsequences of length $T \in \{10, 20\}$ using Top-k decoding ($k = 10$), which sets a loose upper bound on the adversary's access capability. The extraction probability quantifies the likelihood that the target model generates the protected subsequence within N attempts. The extraction advantage measures how much more information the target model provides compared to a GPT-2 proxy model about protected subsequences.

Results in the top row demonstrate that TP-OOV consistently reduces extraction advantage across all N, T, and protection ratios r. When advantage approaches zero as the case for $r = 0.8, N < 100$, the target model provides no more information than the proxy model. Defense effectiveness is particularly strong for shorter sequences ($T = 10$) and scales with protection coverage—larger r values yield greater risk reduction due to reduced overlap with unprotected content. In the bottom figures, analyzing the Top-1% most vulnerable subsequences confirms a statistically risk degradation.

Effectiveness under Privacy Backdoor. Furthermore, as an empirical defense, it is necessary to evaluate it under the current most powerful privacy attacks. Therefore, we apply the recent privacy backdoor [22], [23] to amplify the privacy risk of fine-tuning training data by assuming that the pre-trained model is released and crafted by the privacy attacker. Specifically, we warm up the pre-trained backbone on D_{aux} for 2 more epochs before fine-tuning, which pushes the model to enter the memorization-only stage earlier than using a benign pre-trained model and to memorize more unique details of its training samples. Additionally, we use the warmed-up pre-trained model as the reference model in Loss-Ref MIA.

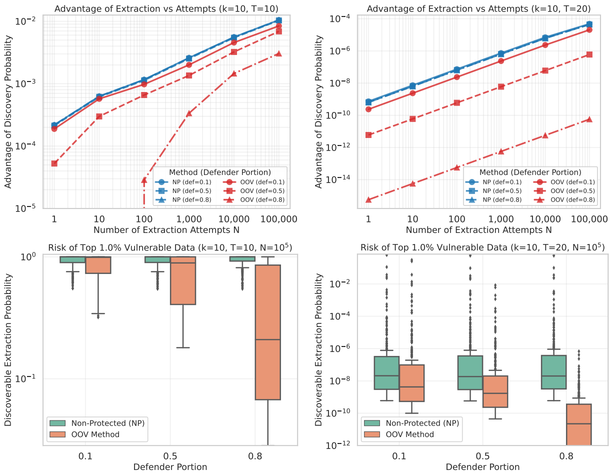
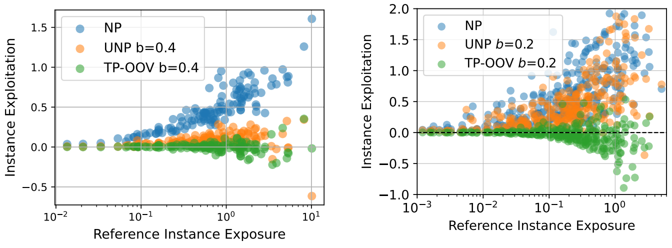


Fig. 5: Discoverable Data Extraction Evaluation.

As shown in Table V, comparing the privacy risk of Patient dataset with the same attack and training setting, the privacy risk is indeed amplified. Similarly, for CC-News dataset on OPT-125M model, the AUC and TPR approach 1 for Loss-Ref and MinK, indicating that the attack is near perfect. Nonetheless, by applying TP-OOV with $b = 1$, `ExpShield` reduces the maximum TPR at 1% FPR across MIA signals from 0.982 to 0.053. In general, the MIA evaluation demonstrates that `ExpShield` successfully offers protection to the data owner by reducing the overall privacy risk of D_{pro} even when the MIA is near perfect.

Instance-Level Risk Evaluation. Beyond the averaged risk over D_{pro}, we evaluate individual-level risk for vulnerable instances using the proposed instance exploitation defined in



(a) LM Patient w/ DB (b) VLM IAPR-TC-12

Fig. 6: Instance vulnerability of representative variants.

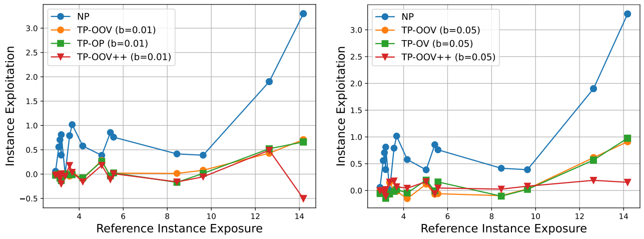


Fig. 7: Effectiveness of optimization-based method on most vulnerable instances given a small portion of defender $r = 0.01$ and small portion of perturbation budget.

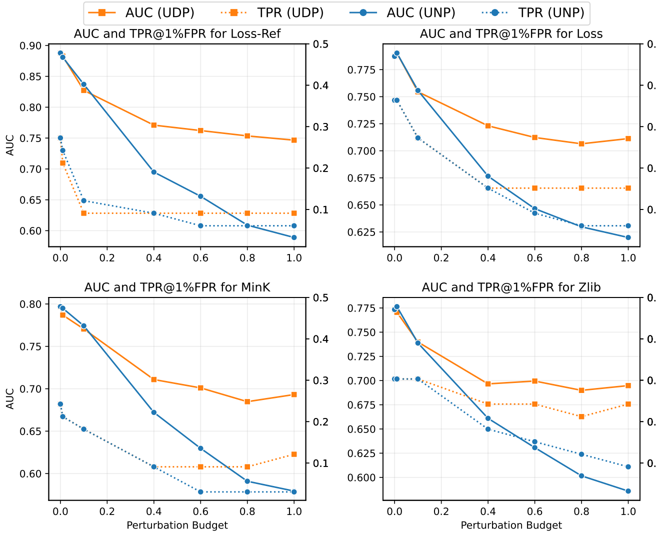


Fig. 8: Disturbing strategy evaluation on deterministic perturbation (UDP) and non-deterministic perturbation (UNP).

Definition 2. As shown in Figure 6a, we compare the instance exploitation for each sample (corresponding to each point) in D_{pro} as a function of the sample $\mathbf{x}$'s instance exposure $E_{\theta\setminus\mathbf{x}}(\mathbf{x})$ obtained from a model not trained on it.

We first identify a general pattern in the unprotected baseline (NP) across different datasets and models: The sample that is originally more exposed than other samples (with a higher $E_{\theta\setminus\mathbf{x}}(\mathbf{x})$) typically has a higher instance exploitation after the model is trained on the sample. In other words, naturally vulnerable instances are prone to being more exposed in future training, which aligns with our memorization trigger hypothesis. And the reason is that the training objective is designed to focus on samples with higher loss. CC-News and Enron exhibit analogous trends (Appendix Figure 13).

As two representative methods without a proxy model (UNP) and with proxy model (TP-OOV), we observe that the instance exposure even for the naturally exposed samples is significantly reduced. Furthermore, by leveraging the proxy model to identify memorization triggers and creating pitfalls in TP-OOV, all protected instances have instance exploitation below 0.5, meaning that ExpShield prevents the protected instance from being exploited beyond the general knowledge in training distribution.

Extending to Vision-Language Modeling. In Table V, we can observe that perturbing memorization triggers (TP) significantly reduces the privacy risk from perfect attack to near random guess with AUC around 0.509. In Figure 6b, we observe a similar trend as in LM that more vulnerable samples which have a higher reference instance exposure tend to have higher instance exploitation for NP. Even for those vulnerable samples, the instance exploitation of each text sample approaches zero with a small budget $b = 0.2$ for TP-OOV, indicating a perfect individual defense for VLM. Due to the space limit, we present the details and additional results in Appendix H.

Effectiveness of Optimization. We now evaluate the effectiveness of optimization-based method with variant TP-OP and the extended version TP-OOV++ for TP-OOV by integrating it with the optimization based method as discussed in Section IV-D4. We note that optimizing for one sentence is affordable, but it is time-consuming to perform the optimization over every instance in D_{pro}. Since we have shown that instances having a naturally high exposure are more prone to having high exploitation, we focus on the most vulnerable data points when we evaluate the optimization-based extension. Thus, we first select a portion $r = 0.1$ as D_{pro} and use the rest of the unprotected samples to train a reference model $\theta_{\setminus\mathbf{x}}$. Then we calculate $Ex_{\theta\setminus}(\mathbf{x})$ for every $\mathbf{x} \in D_{\text{pro}}$ and only keep the vulnerable subset with the Top-20 highest exposure instances in D_{pro} for perturbation. After training on $D_{\text{pro}} \cap D_{\text{un}}$, we can obtain the instance exploitation as shown in Figure 7.

We can observe that optimizing over inserted tokens (TP-OP) has similar effectiveness to breaking up the top memorization triggers (TP-OOV). In addition, extending TP-OOV by searching for an OOV that has a maximized loss via $\beta_1 = -1$ further reduces the instance exploitation for the highest-exposed instance. We leave similar results under different b in Appendix E.

C. Hyper-Parameter and Ablation Analysis

We conduct detailed hyperparameters and ablation analysis to understand the rationale of different variants of ExpShield.

Influence of Disturbing Pattern. We show that the variant of non-deterministic perturbation (UNP) is superior to deterministic perturbation (UDP) as shown in Figure 8, especially when b is larger, because the model can learn how to ignore the perturbation when there is a deterministic pattern.

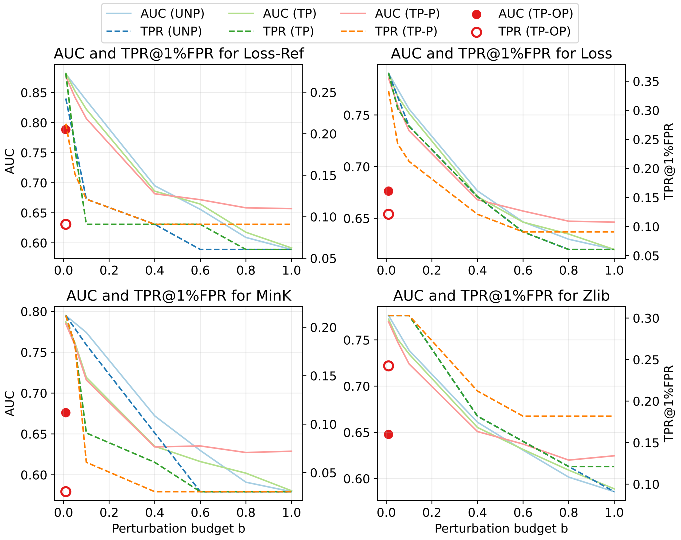


Fig. 9: Evaluation of token-level filling strategies on uniformly random tokens (TP), pitfall tokens (TP-P) and optimized pitfall tokens (TP-OP).

TABLE VI: Hyper-parameter analysis for TP-OP. Optimal metrics are bold and underlines indicate improvement over random perturbation in UNP.

Method	Sample-level		User-level	
	Max-AUC	Max-TPR@1%	Max-AUC	Max-TPR@1%
NP	0.888	0.364	0.676	0.047
UNP	0.881	0.364	0.674	0.039
$\beta_1 = 1, \beta_2 = 0$	0.837	0.303	0.646	**0.039**
$\beta_1 = 0, \beta_2 = 1$	**0.782**	**0.242**	**0.619**	0.070
$\beta_1 = 1, \beta_2 = 1$	0.788	**0.242**	0.624	0.070
$\beta_1 = -1, \beta_2 = 1$	0.784	0.273	0.623	**0.039**
$\beta_1 = -1, \beta_2 = -1$	0.788	**0.242**	0.632	0.047

Influence of Filling Strategies. We compare different filling strategies in Figure 9. We observe that filling with pitfall tokens (TP-P) enhances the defense against most MIAs especially when the perturbation budget $b < 0.6$. And filling with optimization-based pitfall tokens is more effective than other variants on decreasing the privacy risk across all signals, under the perturbation constraint of $b = 0.01$.

Influence of β_1 and β_2. We then analyze the choice of β_1 and β_2 in Equation (13) where a positive coefficient indicates error-minimization and a negative coefficient denotes error-maximization. As shown in Table VI, in general we observe that either a negative or positive coefficient helps to reduce privacy risk. Setting $\beta_1 = 1$ is less effective, as we are optimizing the perturbation to make it as fluent as possible given previous context, which makes the target model ignore the inserted token. On the contrary, $\beta_2 = 1$ yields lower risk, as we encourage the connection between the inserted tokens and the identified memorization triggers, which creates a dependency from memorization-triggered tokens on the inserted token and fools the model to focus on learning perturbation.

Influence on Model Utility. We also compare the model utility with the initial pre-trained model and the model trained without any protection (NP) in Table VII. Even when the defender portion ($r = 0.8$) and perturbation budget ($b = 1$) are large, the validation performance for a model trained on TP-OOV drops compared to NP. It is still significantly better than the initial model, indicating that the model is able to learn from the unprotected data. The 'Mem-Loss' is the average loss of all protected instances, which remains high at 4.023 while the 'Mem-Loss' for NP baseline decreases a lot to 3.586, indicating that the model does not learn much information from D_{pro}. We also notice that there is a significant decrease in the training loss to 2.351, which means the model learns from the combination of unprotected and protected samples rapidly. This is because inserted pitfalls are designed to be outliers in D, and the model is prone to focus on this pattern.

TABLE VII: Training and validation performance.

Method	Val-PPL	Val-Loss	Train-Loss	Mem-Loss
Initial model	65.412	4.181	5.522	4.219
NP	38.321	3.646	3.562	3.586
TP-OOV (b=0.4)	46.813	3.846	3.103	3.916
TP-OOV (b=1)	49.226	3.896	2.351	4.023

TABLE VIII: Compare TP-OOV with data filtering (OUT) on privacy and utility. Privacy is evaluated with Loss-Ref MIA. N/A denotes a worse model performance than pre-training. * denotes random guess in MIAs as $r = 1$ means all samples are removed for OUT.

Metric	Method	r=0.05	r=0.1	r=0.5	r=0.8	r=1
TPR@1%	OUT	0.000	0.557	0.047	0.004	0.01*
	TP-OOV	0.000	0.000	0.000	0.000	0.009
AUC	OUT	0.577	0.473	0.498	0.491	0.5*
	TP-OOV	0.552	0.463	0.504	0.494	0.494
ΔVal-PPL	OUT	0.173	0.544	3.781	8.622	27.062
	TP-OOV	0.258	0.785	4.311	10.905	N/A

In Table VIII, we present the utility degradation for OUT, which removes all protected instances from D, and for TP-OOV, given different defender ratio r with budget $b = 1$ for Patient dataset. We observe that self-guarding with ExpShield achieves a slightly higher utility drop than filtering out protected samples directly. But when all samples are self-guarded and the perturbation against reference-model-based MIA is strong, it is possible that the model's validation performance is worse than the initial pre-trained model. While in a more realistic setting where only the minority of web content is self-guarded ($r < 0.1$), the utility loss compared to OUT is small.

Influences on Risk of Other Instances. We further investigate the broader influence of a self-guard on other instances, as a consideration for the privacy onion effect [33]. For the influence within defenders, the defense effectiveness in Table VIII is stable across various r. This observation does not depend on the specific b and methods as we leave similar results for the weakest variant UDP in Appendix.

Furthermore, we demonstrate the risk influence on the $1 - r$ portion of unprotected samples in Figure 10. When the majority

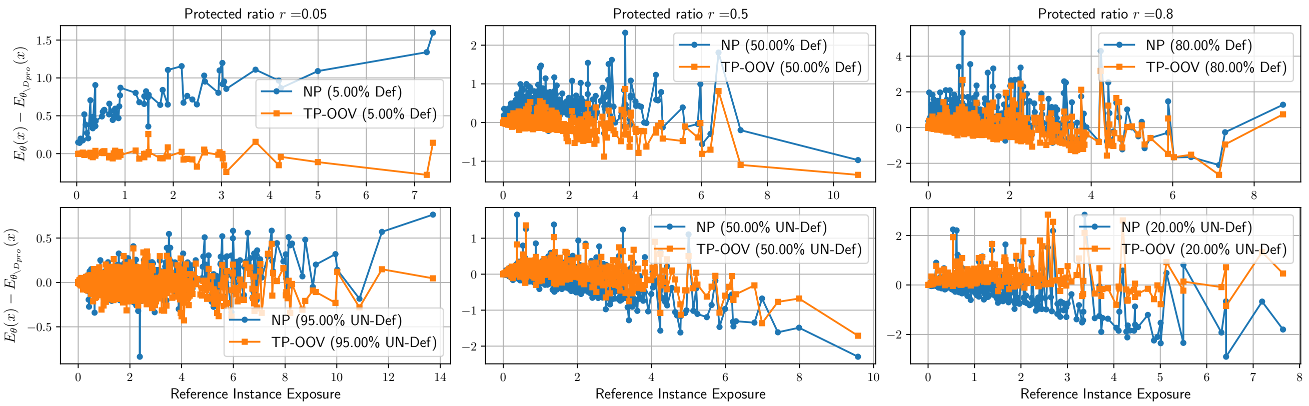


Fig. 10: The influence defender ratio r on the individual risk of other unprotected instances. For figures in each column, we train the exposure reference model with $D\setminus D_{\text{pro}}$ separately for approaching training on $D_{\setminus \mathbf{x}}$ as reference, while larger r results in a looser approximation and the gap becomes smaller from left to right. The y-axis indicates the exposure change after introducing D_{pro}. The gap between NP and TP-OOV demonstrates the exposure change caused by applying ExpShield.

of instances are self-guarded ($r = 0.8$), the individual risk of unprotected instances is higher than with no protection, as unprotected samples become outliers when the outlier pattern has been rapidly learned by the model. Considering the large amount of public web content, r is usually small in practice. In such cases, e.g., $r = 0.05$, unprotected samples also benefit from a slightly lower risk than with no protection, attributed to the regularization effect introduced by perturbation [52].

D. Robustness Analysis

As discussed in Section IV-A, ExpShield is naturally robust to normal data pre-processing such as deduplication, and an active bypass requires hundreds of billions of verifications for perfect stripping. Now we evaluate the robustness of ExpShield with two active strategies that an aggressive trainer takes: 1) performing active detection to locate self-guarded texts for manual stripping, and 2) conducting continuous training on clean data for recovering hidden protected text.

Perturbation Filtering. Model trainers may use perplexity filtering [53] or embedding detection [54] to improve data quality. A large b leads to poor-quality texts, so the whole sample can be filtered out. In this case, there would be no violation on protected instances. In the opposite case, when b is not high enough (such as 0.1) to filter out the whole sample. Figure 11a shows that our proposed perturbation cannot be easily distinguished via both perplexity and embedding spaces. As a result, the perturbation remains in the protected text, acting as a shield.

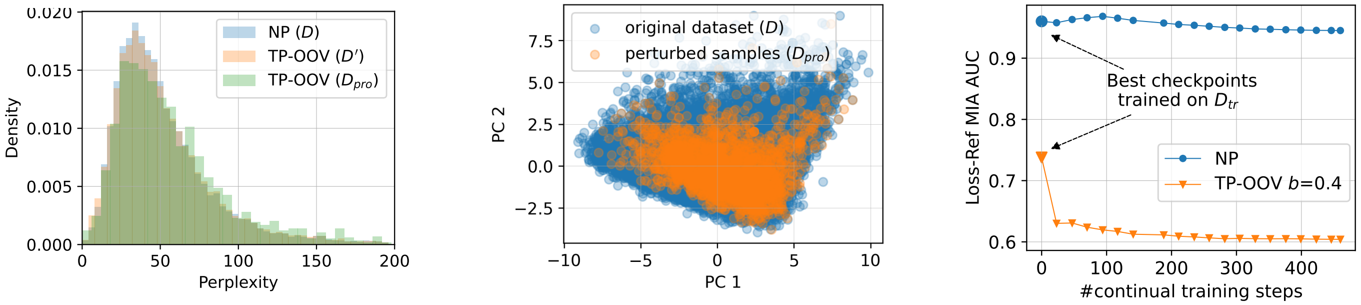
Effect of Continual Training with New Clean Data. LLMs are commonly reused through continual training on new data [55]. We evaluate robustness by fine-tuning GPT-2 on dataset D (containing both protected and unprotected text), then continuing training on disjoint dataset D_{new}. While prior work [56] shows continual training can recover previously unexposed secrets, Figure 11c demonstrates that MIA risk on D_{pro} steadily decreases as the model shifts focus to new data. Notably, TP-OOV benefits more from continual training than NP, with AUC dropping from 0.75 to 0.6, confirming ExpShield's robustness to post-processing model updates.

VI. Related Work

As we aim to protect unauthorized content from being memorized and leaked by language models, our work is closely related to privacy defense and copyright protection.

Privacy Defenses. There are substantial privacy defenses that can be applied to avoid generating training data from LMs, but all of them rely on collaboration from other parties. In data pre-processing, deduplication [8] and scrubbing [9] require a trusted data curator, and it is hard to remove all sensitive information. In the model training stage, differentially private (DP) optimization [57], [58], [59] ensures a theoretical privacy bound for each training record but requires a trusted model trainer who is willing to afford significantly higher training costs, especially for large LMs [60], [61]. Model alignment [11] requires carefully designed alignment tasks. In the inference stage, output filtering [62], machine unlearning [63], or model editing [64] can be applied but require a trusted model curator and poses other risks [65], [66]. Distinguished from above privacy defenses, we aim to provide a broader protection, i.e., unauthorized content rather than only private content, and we do not rely on other parties.

Copyright Protection. One strategy to protect copyrighted content is making it unlearnable. The term unlearnable example [20], [67], [68], [69], [19], [15] is proposed to prevent models from learning any knowledge from the perturbed dataset, and the success indicator is typically a poor inference-stage performance. Unlike traditional unlearnable examples, we aim to degrade the exposure risk of an individual text. Instead of image data or classification tasks in previous works, our defense targets language modeling tasks, requiring unique understanding of how generative models memorize training corpus. The other strategy of copyright protection is to claim the data ownership by embedding data watermarks and detecting them via membership inference after model training [70], [71].



(a) Perplexity distribution

(b) Embedding distribution (PCA)

(c) Effect of continual training

Fig. 11: Robustness Analysis of Active Detection and Continuous Training.

Although ExpShield was not designed for watermarking, we discuss how ExpShield can be used for data watermark in Appendix F.

VII. Discussion

We discuss the benefits and limitations in practical scenarios.

Easy-to-Use Protection. Besides the advantage of not relying on other parties, we naturally provide a personalized protection strength and scope. The computational cost is affordable for defenders, even with proxy models, e.g., only taking <2 GB GPU memory for GPT-2 or millisecond-level API latency for online models. Besides, the run-time overhead of perturbed webpage is small. In our demonstration [37], the loading time averaged over 10 trails only increase by ≈2%. The cost will be even lower for smaller budget (e.g., b=0.01 in Figure 7).

Compatibility to Privacy Defenses. As an owner-side defense, ExpShield is compatible to other privacy defenses that may be implemented by other parties. For example, since ExpShield has no repeated pattern and does not contain meaningful entities, it is robust when integrating with privacy defenses in preprocessing, such as deduplication [8] and scrubbing [9]. Also, the extra layer of ExpShield does not violate the theoretical guarantee of DP training [10] and other inference phase defenses [12].

Generalization to Other Languages. In principle, ExpShield can be used for other languages, as it operates at the token level without relying on English-specific syntax, independent of specific word segmentation. Specifically, our OOV-based perturbation also applies to CJK characters, because several CJK characters can be merged into a single token in popular tokenizers (BPE, SentencePiece). For languages such as Thai or Khmar where zero-width character is used for line breaking, we suggest to use style-level perturbation to avoid rendering issues.

Against High-Resource Crawlers. While our main target is large-scale automatic crawlers instead of aggressive crawlers with high-resource computation, ExpShield is still possible to reduce risks because there is no guarantee for perfect sanitization. In Figure 7, only $\approx 1\%$ surviving perturbation still lower memorization risk. Defenders can even combine multiple ExpShield variants to enhance the complexity of sanitization. While Optical Character Recognition (OCR) is powerful, mainstream LLMs [72], [73] use HTML extraction instead of OCR for collecting large-scale webpages, probably due to its accuracy and cost (around 100–300× more costly).

Impacts on User Experience. The negative impacts on the user experience of perturbed webpages are minimal. Website owners can exclude the self-injected perturbation to ensure accurate internal site search, and mark perturbed pages as noindex to avoid the mismatch issue on legitimate search engine optimization (SEO). In the extreme case where every word is perturbed by invisible characters, browser search, word selection, screen reading, text highlighting are unaffected; the translation remains functional for most words. And the style-level perturbation is functional with a small budget (e.g., $b < 0.1$). In Figure 7, even a small budget demonstrates the effectiveness of the defense, so the overall impact is marginal.

Collaborative Mitigation of Data Misuse. When protected data is unintentionally misused, such as being collected by third-party crawlers, model trainers who are aware of the defense are encouraged to either: (i) exclude all segments that may contain perturbations to avoid degrading training quality and data misuse; or (ii) include the perturbed data during training while filtering out invisible characters at inference time, both to reduce the risk of privacy attack and mitigate safety-related abuses.

VIII. Conclusion

We present ExpShield, a proactive self-guard that empowers data owners with direct control over their content's usage in AI training, addressing ineffective crawl prevention and third-party dependency.

Our approach fills the critical gap where content creators lack protection against unauthorized LLM memorization through a practical, independent solution requiring no third-party cooperation. We formalize individual text protection as constrained bi-level optimization that minimizes adversarial advantage while preserving readability and budget constraints. For principled evaluation and design, we introduce instance exploitation—a standard, calibrated, and efficient individual-level privacy metric that is informative for a wide scope of attacks. By establishing and verifying the memorization trigger hypothesis, we develop targeted perturbations that focus on influencing important tokens for memorization. It is promising for future works to design more informed individual-risk metrics and extend the memorization trigger hypothesis for improving defenses during or after training.

We comprehensively validate the defense effectiveness across various tasks with LMs and VLMs, showing its capability to reduce the near-perfect attack to random membership guess. Additionally, we extend discussions of its feasibility in practice and advocate a collaborative view for responsible AI where data protection rights and innovation coexist.

Acknowledgment

We would like to thank reviewers for their constructive comments and efforts in improving our paper. This work was supported in part by National Science Foundation grants (CNS-2437345, CNS-2125530, CNS-2124104, IIS-2302968, CNS-2350333) and National Institutes of Health grants (R01LM013712, R01ES033241).

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] P. Henderson, X. Li, D. Jurafsky, T. Hashimoto, M. A. Lemley, and P. Liang, "Foundation models and fair use," *Journal of Machine Learning Research*, vol. 24, no. 400, pp. 1–79, 2023.

[3] F. Tramèr, G. Kamath, and N. Carlini, "Position: Considerations for differentially private learning with large-scale public pretraining," in *Forty-first International Conference on Machine Learning*, 2024.

[4] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[5] J. Freeman, C. Rippe, E. Debenedetti, and M. Andriushchenko, "Exploring memorization and copyright violation in frontier llms: A study of the new york times v. openai 2023 lawsuit," *arXiv preprint arXiv:2412.06370*, 2024.

[6] J. Loh and M. J. Walsh, "Social media context collapse: The consequential differences between context collusion versus context collision," *Social Media+ Society*, vol. 7, no. 3, p. 20563051211041646, 2021.

[7] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application." *International Journal of Advances in Soft Computing & Its Applications*, vol. 13, no. 3, 2021.

[8] N. Kandpal, E. Wallace, and C. Raffel, "Deduplicating training data mitigates privacy risks in language models," in *International Conference on Machine Learning*. PMLR, 2022, pp. 10 697–10 707.

[9] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 346–363.

[10] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[11] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning {ai} with shared human values," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=dNy_RKzJacY

[12] M.-A. Panaitescu-Liess, Z. Che, B. An, Y. Xu, P. Pathmanathan, S. Chakraborty, S. Zhu, T. Goldstein, and F. Huang, "Can watermarking large language models prevent copyrighted text generation and hide training data?" in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. [Online]. Available: https://doi.org/10.1609/aaai.v39i23.34684

[13] T. Igamberdiev and I. Habernal, "DP-BART for privatized text rewriting under local differential privacy," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 914–13 934. [Online]. Available: https://aclanthology.org/2023.findings-acl.874/

[14] X. Yue, M. Du, T. Wang, Y. Li, H. Sun, and S. S. M. Chow, "Differential privacy for text analytics via natural text sanitization," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3853–3866. [Online]. Available: https://aclanthology.org/2021.findings-acl.337/

[15] X. Li and M. Liu, "Make text unlearnable: Exploiting effective patterns to protect personal data," in *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. Association for Computational Linguistics, Jul. 2023, pp. 249–259. [Online]. Available: https://aclanthology.org/2023.trustnlp-1.22/

[16] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[17] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks," *arXiv:1802.08232 [cs]*, Jul. 2019.

[18] L. Fowl, M. Goldblum, P.-y. Chiang, J. Geiping, W. Czaja, and T. Goldstein, "Adversarial examples make strong poisons," *Advances in Neural Information Processing Systems*, 2021.

[19] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, "Availability attacks create shortcuts," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2367–2376.

[20] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," *arXiv preprint arXiv:2101.04898*, 2021.

[21] D. Bowen, B. Murphy, W. Cai, D. Khachaturov, A. Gleave, and K. Pelrine, "Data poisoning in llms: Jailbreak-tuning and scaling laws," 2024. [Online]. Available: https://arxiv.org/abs/2408.02946

[22] R. Liu, T. Wang, Y. Cao, and L. Xiong, "Precurious: How innocent pre-trained language models turn into privacy traps," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[23] Y. Wen, L. Marchyok, S. Hong, J. Geiping, T. Goldstein, and N. Carlini, "Privacy backdoors: Enhancing membership inference through poisoning pre-trained models," *arXiv preprint arXiv:2404.01231*, 2024.

[24] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee, "Scalable extraction of training data from (production) language models," *arXiv preprint arXiv:2311.17035*, 2023.

[25] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[26] A. Radford, "Improving language understanding by generative pre-training," 2018.

[27] J. Hayes, M. Swanberg, H. Chaudhari, I. Yona, I. Shumailov, M. Nasr, C. A. Choquette-Choo, K. Lee, and A. F. Cooper, "Measuring memorization in language models via probabilistic extraction," *arXiv preprint arXiv:2410.19482*, 2024.

[28] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting," *arXiv:1709.01604 [cs, stat]*, May 2018.

[29] C. Dwork, N. Kohli, and D. Mulligan, "Differential Privacy in Practice: Expose your Epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, Oct. 2019.

[30] B. Balle, G. Cherubin, and J. Hayes, "Reconstructing training data with informed adversaries," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1138–1156.

[31] A. Salem, G. Cherubin, D. Evans, B. Köpf, A. Paverd, A. Suri, S. Tople, and S. Zanella-Béguelin, "Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 327–345.

[32] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[33] N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer, "The privacy onion effect: Memorization is relative," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 263–13 276, 2022.

[34] N. Boucher, I. Shumailov, R. Anderson, and N. Papernot, "Bad characters: Imperceptible nlp attacks," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1987–2004.

[35] Z. Liao, L. Mo, C. Xu, M. Kang, J. Zhang, C. Xiao, Y. Tian, B. Li, and H. Sun, "Eia: Environmental injection attack on generalist web agents for privacy leakage," *arXiv preprint arXiv:2409.11295*, 2024.

[36] L. Richardson, "Beautiful soup documentation," 2007.

[37] ExpShield's authors, "Expshield demo," https://github.com/Emory-AIMS/ExpShield-demo, 2025, gitHub-style repository, accessed 15 Dec 2025.

[38] A. Z. Broder, "On the resemblance and containment of documents," in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE, 1997, pp. 21–29.

[39] S. Longpre, G. Yauney, E. Reif, K. Lee, A. Roberts, B. Zoph, D. Zhou, J. Wei, K. Robinson, D. Mimno *et al.*, "A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 3245–3276.

[40] Anonymous, "Do we really have to filter out random noise in pre-training data for language models?" in *Submitted to ACL Rolling Review - February 2025*, 2025, under review. [Online]. Available: https://openreview.net/forum?id=mpnPR8YQ3d

[41] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," *arXiv preprint arXiv:2310.16789*, 2023.

[42] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[43] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*. Springer, 2004, pp. 217–226.

[44] G. Zeng, W. Yang, Z. Ju, Y. Yang, S. Wang, R. Zhang, M. Zhou, J. Zeng, X. Dong, R. Zhang *et al.*, "Meddialog: Large-scale medical dialogue datasets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[45] J. Mackenzie, R. Benham, M. Petri, J. R. Trippas, J. S. Culpepper, and A. Moffat, "Cc-news-en: A large english news corpus," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 3077–3084.

[46] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The iapr tc12 benchmark: A new evaluation resource for visual information systems," *International Conference on Language Resources and Evaluation*, 2006.

[47] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[48] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[49] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[50] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 1992, pp. 569–593.

[51] H. Puerto, M. Gubri, S. Yun, and S. J. Oh, "Scaling up membership inference: When and how attacks succeed on large language models," *arXiv preprint arXiv:2411.00154*, 2024.

[52] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[53] M. Marion, A. Üstün, L. Pozzobon, A. Wang, M. Fadaee, and S. Hooker, "When less is more: Investigating data pruning for pretraining llms at scale," *arXiv preprint arXiv:2309.04564*, 2023.

[54] A. Kumar, P. Makhija, and A. Gupta, "Noisy text data: Achilles' heel of bert," *arXiv preprint arXiv:2003.12932*, 2020.

[55] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.06196

[56] X. Chen, S. Tang, R. Zhu, S. Yan, L. Jin, Z. Wang, L. Su, Z. Zhang, X. Wang, and H. Tang, "The janus interface: How fine-tuning in large language models amplifies the privacy risks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1285–1299.

[57] Martín Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, pp. 308–318, 2016.

[58] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially Private Meta-Learning," *arXiv:1909.05830 [cs, stat]*, Sep. 2019.

[59] B. Jayaraman and D. Evans, "Evaluating Differentially Private Machine Learning in Practice," in *USENIX*, 2019, p. 18.

[60] X. Li, F. Tramèr, P. Liang, and T. Hashimoto, "Large Language Models Can Be Strong Differentially Private Learners," *arXiv:2110.05679 [cs]*, Oct. 2021.

[61] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang, "Differentially Private Fine-tuning of Language Models," *arXiv:2110.06500 [cs, stat]*, Oct. 2021.

[62] S. Goyal, M. Hira, S. Mishra, S. Goyal, A. Goel, N. Dadu, K. DB, S. Mehta, and N. Madaan, "Llmguard: Guarding against unsafe llm behavior," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 23 790–23 792, Mar. 2024.

[63] Y. Cao and J. Yang, "Towards Making Systems Forget with Machine Unlearning," in *2015 IEEE Symposium on Security and Privacy (SP)*. San Jose, CA: IEEE, May 2015, pp. 463–480.

[64] Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang, "Editing large language models: Problems, methods, and opportunities," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[65] Y. Huang, D. Liu, L. Chua, B. Ghazi, P. Kamath, R. Kumar, P. Manurangsi, M. Nasr, A. Sinha, and C. Zhang, "Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy," *arXiv preprint arXiv:2410.09591*, 2024.

[66] H. Hu, S. Wang, T. Dong, and M. Xue, "Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024, pp. 262–262.

[67] J. Ren, H. Xu, Y. Wan, X. Ma, L. Sun, and J. Tang, "Transferable unlearnable examples," *arXiv preprint arXiv:2210.10114*, 2022.

[68] Y. Liu, K. Xu, X. Chen, and L. Sun, "Stable unlearnable example: Enhancing the robustness of unlearnable examples via stable error-minimizing noise," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 4, 2024, pp. 3783–3791.

[69] S. Fu, F. He, Y. Liu, L. Shen, and D. Tao, "Robust unlearnable examples: Protecting data against adversarial learning," *arXiv preprint arXiv:2203.14533*, 2022.

[70] J. T.-Z. Wei, R. Y. Wang, and R. Jia, "Proving membership in llm pretraining data via data watermarks," *arXiv preprint arXiv:2402.10892*, 2024.

[71] S. Zhao, L. Zhu, R. Quan, and Y. Yang, "Protecting copyrighted material with unique identifiers in large language model training," 2025. [Online]. Available: https://arxiv.org/abs/2403.15740

[72] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[73] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," 2020. [Online]. Available: https://arxiv.org/abs/2101.00027

[74] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[75] D. Dittrich, E. Kenneally *et al.*, "The menlo report: Ethical principles guiding information and communication technology research," US Department of Homeland Security, Tech. Rep., 2012.

Appendix

A. Dataset Information

- Enron [43] is a large collection of email data from the Enron Corporation, which contains sufficient PII information such as phone numbers, email addresses, and names. The dataset comprises emails from 150 users, primarily senior management of Enron.

- Patient [44] consists of doctor-patient conversations covering various medical conditions, symptoms, diagnoses, and treatment plans, with an average length of 8 turns per conversation.
- CC-News [45] is derived from the Common Crawl News dataset, containing news articles from various online sources published between 2016-2019. The articles span diverse topics and writing styles, providing a rich test bed for evaluating privacy preservation in copyrighted content.

TABLE IX: Defense Strategy Comparison on Scenarios and Compatibility

Defense Type	Personal Websites	UGC Platforms	Screen Reader Compatibility	DOM Element Count Impact
Style-Level	✓	×	Medium	Slightly increase
Character-Level	✓	✓	High	No change

B. Perturbation operation discussions

A previous work [34] proposes adversarial examples by inserting imperceptible characters, such as invisible characters, homoglyphs, reordering characters, and deletion characters, into text inputs. These perturbations as follows are designed to be undetectable by human users while significantly altering the output of natural language processing (NLP) models.

- **Augmentation**: Augment original text by modifying the encoding style without altering its visual display. Techniques include applying CSS properties like font-size or absolute position to make text invisible, inserting zero-width or invisible whitespace characters, and hiding text within HTML comment tags. In our TP-OOV, examples of invisible characters include the Zero Width Space (U+200B), Zero Width Non-Joiner (U+200C), and Zero Width Joiner (U+200D). These characters do not render visually but are encoded in the HTML, allowing for subtle modifications.
- **Deletion**: Remove characters to obscure text, either through delete-characters (e.g., Backspace, Delete) that are font and platform-independent, or using JavaScript to conditionally hide content, making it platform-dependent. These approaches are effective against basic scrapers but less so against those that can execute JavaScript. CSS pseudo-elements such as '::before' or '::after' and replacing hidden text with SVG graphics can also hide partial text.
- **Replacement**: Replace characters with HTML entities, or visually similar homoglyphs (e.g., replacing Latin letters with visually similar Cyrillic ones), which are character-dependent. This technique confuses basic scrapers but may lead to imperfect readability.
- **Shuffling**: Use control characters like Carriage Return (CR), Backspace (BS), or Delete (DEL) to reorder or hide parts of the text. This method is platform- and character-dependent, effective at shuffling content without reducing readability when done carefully.

Furthermore, we compare character-level and style-level perturbation with respect to practical application in Table IX.

C. Demonstration of invisible perturbation

Three Methods to Make Text Invisible in HTML

```
<!-- Method 1: Using CSS display
    property -->
<div style="display: none;">This text is
     invisible</div>

<!-- Method 2: Using zero-width
    characters -->
<span>Visible text&#8203;z&#8203;e
    &#8203;r&#8203;o&#8203;-&#8203;w
    &#8203;i&#8203;d&#8203;t&#8203;h
    &#8203; characters hidden here</span>

<!-- Method 3: Using CSS positioning and
     size -->
<div style="position: absolute; left:
    -9999px; font-size: 0;">
  This text is positioned off-screen and
       has zero font size
</div>
```

Fig. 12: A simplified demonstration of invisibility strategy

We demonstrate a few ways of creating invisible styles in Figure 12 and provide a concrete example in demonstration [37]:

- **Method 1:** Uses CSS `display: none` to prevent the element from rendering in the document flow.
- **Method 2:** Inserts zero-width space characters (Unicode U+200B) between letters, making the text invisible while maintaining its position in the document.
- **Method 3:** Combines absolute positioning (moving the element far off-screen) with zero font size to hide text.

D. A Variant of Informed-MIA via Privacy Backdoor

When demonstrating the dataset-level risk with variants of MIAs, we aim to use a more informed MIA game by assuming a stronger (informed) adversary knowledge, thus the mitigation under such strong privacy game can be reducible to other weaker attacks in practice. Thus, we follow previous works of privacy backdoor [22], [23] by assuming the adversary has the capability to craft and release the pre-trained model. The informed MIA game is shown in Algorithm 3.

Algorithm 3 BACKDOORED AND INFORMED MIA GAME

```
procedure BACKDOORED-MIA($\mathcal{T}, \mathcal{A}, D_{\backslash x}, \mathbf{x}; D_{\text{aux}}$)
    $\theta_{\text{adv}} \leftarrow \mathcal{A}_{\text{craft}}(D_{\text{aux}}, \theta_{\text{pre}})$ // Insert privacy backdoor
    $s \leftarrow \text{Unif}(\{\mathbf{x}, \perp\})$ // Sample membership status
    $\theta \leftarrow \mathcal{T}(D_{\backslash \mathbf{x}} \cup \{\tilde{\mathbf{x}} | s \neq \perp\}; \theta_{\text{adv}})$ // Train on released data with the backdoored pre-trained model
    $\hat{s} \leftarrow \mathcal{A}(\theta_{\text{ft}}, \theta_{\text{adv}}, \mathbf{x}; D_{\text{aux}})$ // Guess membership of $\mathbf{x}$
    return $\hat{s} = s$
```

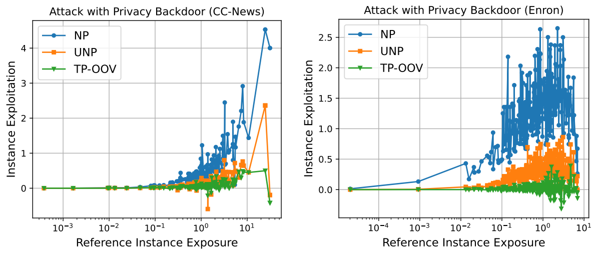


Fig. 13: Instance-level analysis via instance exploitation, with the corresponding maximum MIA AUC as 0.605 for CC-News ($b = 1$), and 0.621 for Enron ($b = 1$) obtained with privacy backdoor.

E. Instance Exploitation

Additional Results for CC-News and Enron. We complement results of Figure 6a on two extra datasets CC-News and Enron in Figure 13 with the similar trend that after applying ExpShield the instance exploitation for most text instances approaches to zero. The gain of perturbing with OOV compared to uniform token sequence is significant for all datasets.

Results with Different Budgets. We complement results of Figure 7 with different perturbation budget b in Figure 14.

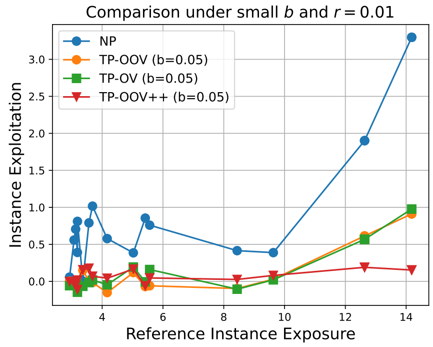


Fig. 14: Effectiveness of optimization-based method on most vulnerable instances given a small portion of defender $r = 0.01$ and small portion of perturbation budget.

F. Perturbation for Data Watermark

Although we target for proactive memorization mitigation while data watermark is a reactive strategy for data proving, we have similar technique of perturbing text in general. Technically, ExpShield can be extended as data watermark for claiming the ownership. Specifically, prior works [70], [71] insert random canary into protected content, query the model with the canary, and then perform MIA for watermark detection.

Although watermarking is not our focus, we demonstrate ExpShield's feasibility as data watermark in Table X. Using p-value and z-score metrics for hypothesis testing (lower values indicate stronger detection), ExpShield provides extremely strong detection power. TP-P with artificial memorization tokens performs best among variants. While TP-OP should theoretically excel, optimization across multiple samples proves time-consuming, yielding insufficient pitfall optimization. Nevertheless, all variants detect data watermarks effectively.

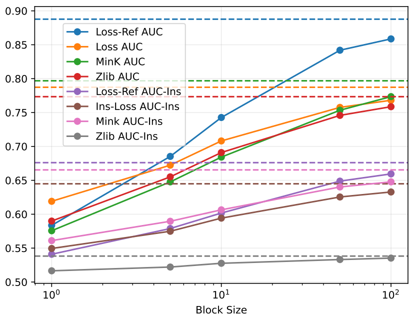


Fig. 15: Influence of disturbing in UDP.

TABLE X: Detection effectiveness for Patient dataset and GPT-2 Model when ExpShield serves as data watermark.

	Detection w/o context		Detection w/ context	
Method	p-value	z-score	p-value	z-score
UDP	6.90E-07	-12.858	3.57E-07	-12.122
UNP	2.34E-17	-20.696	2.29E-22	-25.265
TP	2.74E-11	-19.769	1.75E-292	-122.483
TP-P	1.90E-87	-48.648	4.14E-305	-101.609
TP-OP	7.47E-09	-15.860	1.29E-233	-116.993

G. MIA Results

Influence of Defender Portion. We demonstrate the influence of defender portion r on the defense effectiveness in Table XI. A smaller defender ratio results in a lower privacy risk, or equally a better defense effectiveness under both sample-level and user-level MIA.

Full Results of Variant MIA Signals. Due to space limitation, we omit the MIA results for each MIA signal and only report the maximum MIA AUC and TPR across different MIA signals. We report the full results with privacy backdoor Table XII and omit the one without privacy backdoor due to space limitation.

TABLE XI: Influence of defender portion r for UDP.

UDP	Sample-Level		User-Level	
Defender Portion r	AUC	TPR@1%	AUC	TPR@1%
1.000	0.791	0.068	0.616	0.008
0.800	0.775	0.112	0.612	0.011
0.500	0.779	0.130	0.598	0.009
0.100	0.743	0.091	0.602	0.000
0.050	0.734	0.059	0.601	0.078

H. Vision-Language Modelling

Setup. We use IAPR TC-12 dataset [46] covering diverse subjects (sports, people, animals, cities, landscapes) with image-caption pairs including title, description, location, and date. We focus on image captioning using BLIP2-ViT-gOPT2.7B [74] (3.8B parameters) with standard causal language modeling. The setup takes an image and partially masked caption as input, outputting the next caption word. We employ LoRA (rank=16)

TABLE XII: Membership inference evaluation with privacy backdoor.

MIA level	Patient GPT-2	Loss-Reference		Loss		MinK		Zlib	
w/ Backdoor	Method	AUC	TPR@1%FPR	AUC	TPR@1%FPR	AUC	TPR@1%FPR	AUC	TPR@1%FPR
Sample-level	NP	0.953	0.545	0.792	0.364	0.815	0.303	0.783	0.303
	UDP (b=0.4)	0.831	0.121	0.710	0.152	0.704	0.121	0.697	0.182
	UNP (b=0.4)	0.766	0.091	0.674	0.152	0.679	0.091	0.658	0.152
	TP (b=0.4)	0.765	0.091	0.671	0.152	0.644	0.061	0.657	0.182
	TP-P (b=0.4)	0.772	0.091	0.676	0.152	0.654	0.061	0.655	0.182
	TP-OOV (b=0.4)	0.566	0.061	0.587	0.061	0.503	0.030	0.562	0.091
	UNP-OOV (b=0.4)	0.648	0.091	0.648	0.121	0.614	0.061	0.624	0.152
User-level	NP	0.741	0.000	0.652	0.047	0.672	0.047	0.540	0.023
	UDP (b=0.4)	0.649	0.000	0.599	0.047	0.611	0.039	0.528	0.023
	UNP (b=0.4)	0.622	0.000	0.582	0.039	0.598	0.039	0.524	0.023
	TP (b=0.4)	0.618	0.000	0.581	0.039	0.585	0.039	0.523	0.023
	TP-P (b=0.4)	0.619	0.000	0.580	0.023	0.592	0.039	0.523	0.023
	TP-OOV (b=0.4)	0.542	0.000	0.538	0.039	0.521	0.039	0.515	0.023
	UNP-OOV (b=0.4)	0.566	0.000	0.566	0.039	0.574	0.039	0.520	0.023
MIA level	CC-News OPT-125M	Loss-Reference		Loss		MinK		Zlib	
w/ Backdoor	Method	AUC	TPR@1%FPR	AUC	TPR@1%FPR	AUC	TPR@1%FPR	AUC	TPR@1%FPR
Sample-level	NP	0.998	0.982	0.642	0.006	0.668	0.006	0.700	0.036
	UDP (b=0.4)	0.997	0.970	0.594	0.006	0.605	0.006	0.650	0.030
	UNP (b=0.4)	0.983	0.467	0.556	0.006	0.559	0.006	0.613	0.030
	TP (b=0.4)	0.978	0.580	0.554	0.006	0.548	0.006	0.610	0.024
	TP-P (b=0.4)	0.991	0.746	0.560	0.006	0.555	0.006	0.615	0.024
	TP-OOV (b=0.4)	0.890	0.083	0.518	0.006	0.510	0.006	0.577	0.024
	TP-OOV (b=1)	0.621	0.053	0.498	0.006	0.495	0.006	0.554	0.018
User-level	NP	0.966	0.032	0.620	0.032	0.648	0.035	0.551	0.007
	UDP (b=0.4)	0.948	0.028	0.582	0.032	0.607	0.035	0.542	0.007
	UNP (b=0.4)	0.912	0.028	0.559	0.032	0.581	0.032	0.535	0.007
	TP (b=0.4)	0.918	0.028	0.557	0.032	0.573	0.035	0.536	0.007
	TP-P (b=0.4)	0.923	0.032	0.559	0.032	0.577	0.035	0.536	0.007
	TP-OOV (b=0.4)	0.783	0.028	0.532	0.032	0.547	0.035	0.531	0.007
	TP-OOV (b=1)	0.605	0.028	0.521	0.032	0.543	0.035	0.527	0.007

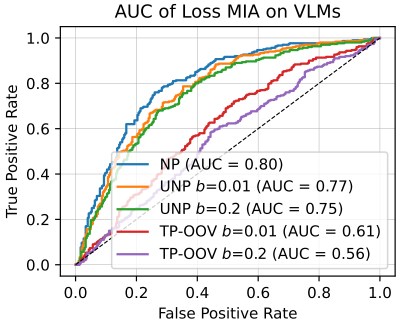


Fig. 16: Loss MIA performance on VLMs.

on query and value matrices across vision encoder, Q-former, and LLM components for efficiency. We use 3K image-caption pairs for D (10% protected as D_{pro}), train reference models on separate 3K pairs, and optimize for 20 epochs using AdamW (lr=5e-4) with validation on 500 images.

Results of Loss MIAs. TP-OOV effectively reduces MIA performance, consistent with main paper results. Figure 16 shows ROC curves where baseline NP marginally affects MIAs (AUC: 0.8→0.77/0.75 for $b = 0.01/0.02$), while TP-OOV significantly reduces AUC to 0.56 ($b = 0.2$) and 0.61 ($b = 0.01$) from 0.80.

I. Ethical Considerations

This work presents a proactive defense to safeguard users' released text from potential LLM misuse, aligning with the ethical principles of "Respect for Persons" and "Beneficence" (e.g., as outlined in the Menlo Report [75]) by promoting individual autonomy and data privacy.

While providing a valuable safeguard, our text modification technique introduces dual ethical considerations. First, the mechanism could be maliciously exploited by adversaries to embed imperceptible modifications, potentially leading to data poisoning, backdoor vulnerabilities, or performance degradation in trained models. Second, the pursuit of individual protection may inadvertently impact collective fairness. For instance, a data owner's successful defense marginally increases the likelihood of other users' unprotected text being exposed through model outputs.

Therefore, we perform analysis related to above considerations. 1) Our extensive experiments show that training on a small portion of protected text does not degrade model performance. 2) We discussed the collaborative mitigation in Section VII by removing the whole perturbed and protected content from training corpus. 3) We demonstrated that the fairness issue is not observed given a reasonably small portion of protection set (with $r < 0.5$ in Figure 10). Meanwhile, we encourage users and practitioners must remain vigilant to these broader ramifications, ensuring system integrity and collective fairness are not inadvertently compromised.