ExpShield: Safeguarding Web Text from Unauthorized Crawling and LLM Exploitation

Ruixuan Liu, Toan Tran, Tianhao Wang, Hongsheng Hu, Shuo Wang, Li Xiong

Emory University, University of Virginia, Shanghai Jiao Tong University

ruixuan.liu2@emory.edu

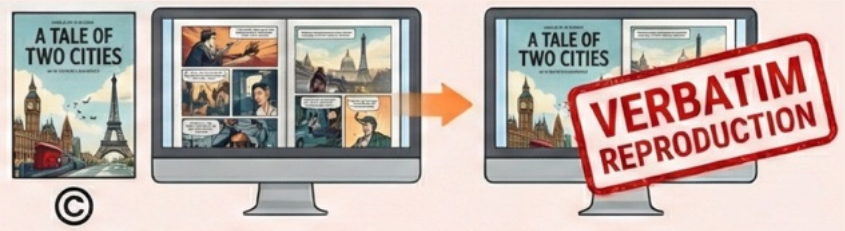
NDSS 2026

NDSS SYMPOSIUM

Why Your Content Is at Risk?

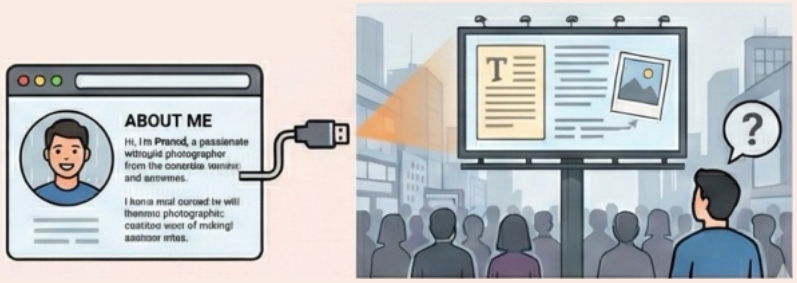
LLMs are trained on the open web

Copyright Violation



Replicating creative work without license

Context Collapse



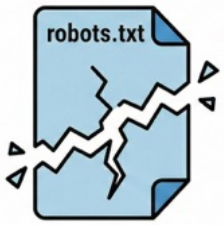
Personal content exploited out of context

Why Existing Defenses Fall Short?

ALL existing defenses require trust in a third party you don't have

Trust in Trainer
(Privacy-preserving training)

Trust in Model Curator
(Machine unlearning)



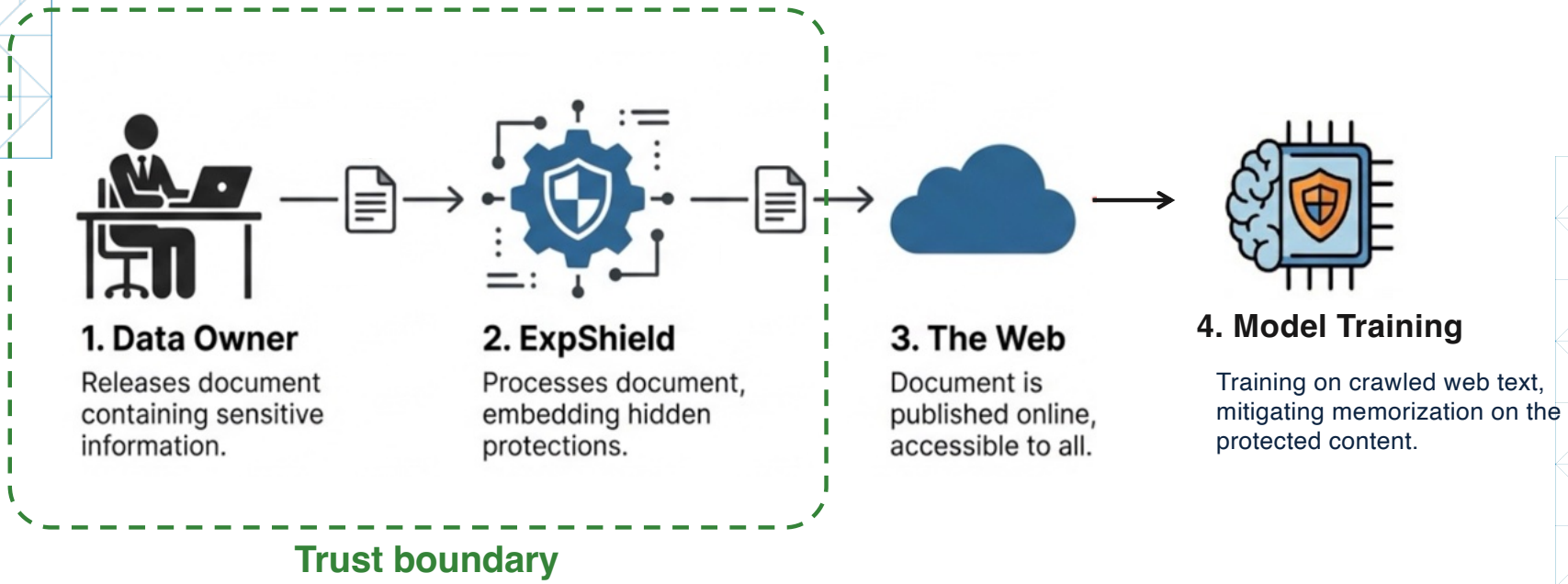
Trust in Crawler
(Voluntary compliance)

Content owners have NO technical leverage in the current ecosystem

Introducing ExpShield

Proactive, owner-side self-guard

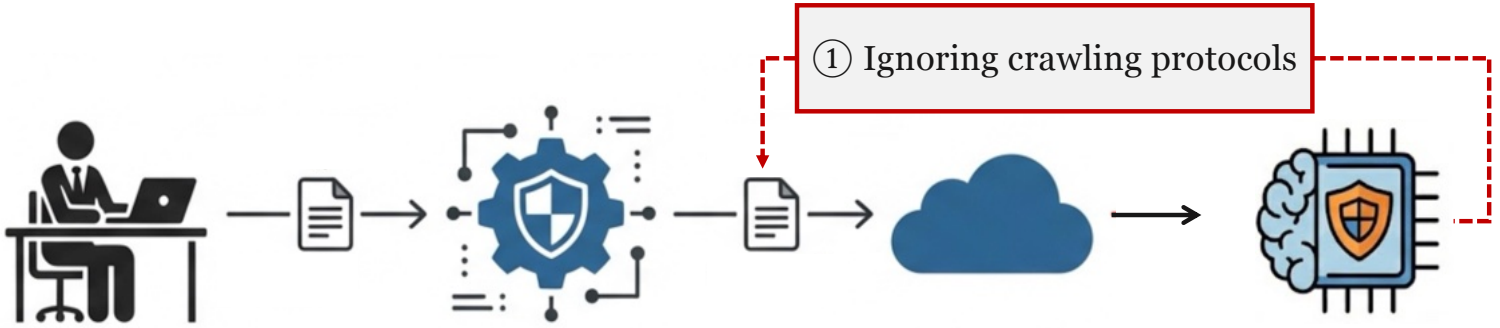
- No third-party cooperation needed



Threat Model

Model Trainer (Misuser)

Moderate adversary who only prioritizes data collection, resulting in data misuse through negligence rather than malicious intent to bypass self-guards.



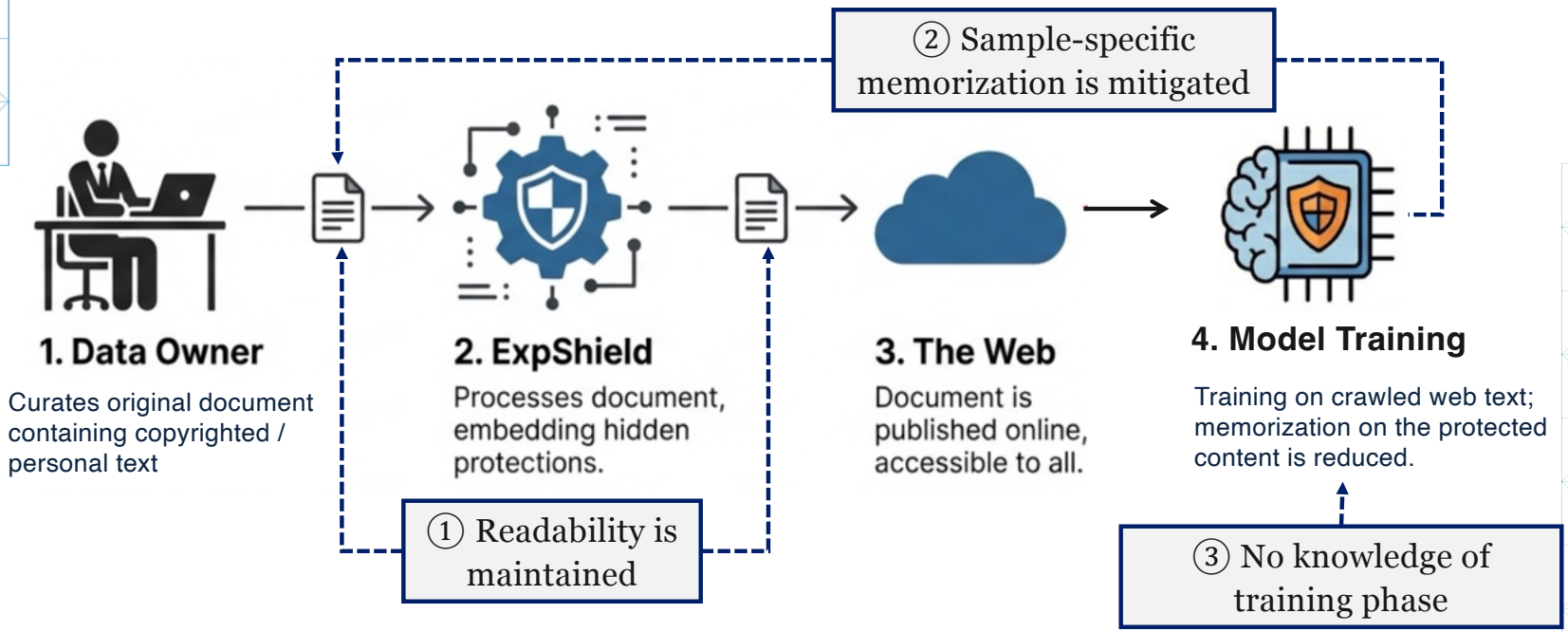
③ No active bypass (computation / legal cost)

② No active protection

Threat Model

Data Owner (Defender)

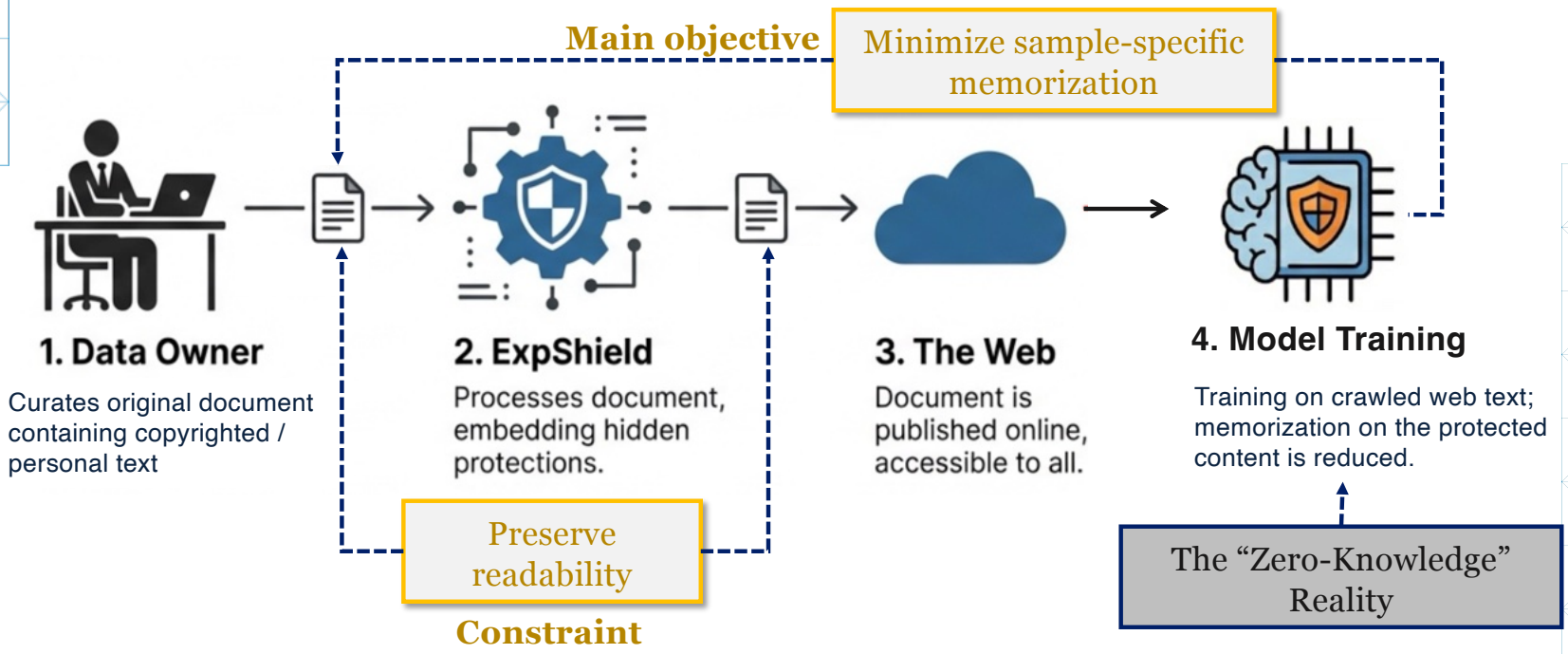
Proactive defender aiming to mitigate sample-specific memorization while preserving perfect readability, operating under a strict zero-knowledge constraint.



The Ideal Formulation vs. The Reality

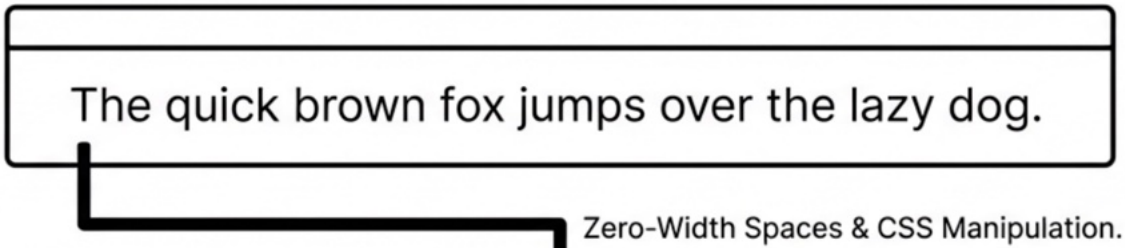
The ideal solution of bi-level optimization is intractable

Due to unknown target model, other training data and training algorithms



Perturbation as A Realistic Alternative

Inserting invisible perturbations to disrupt model training

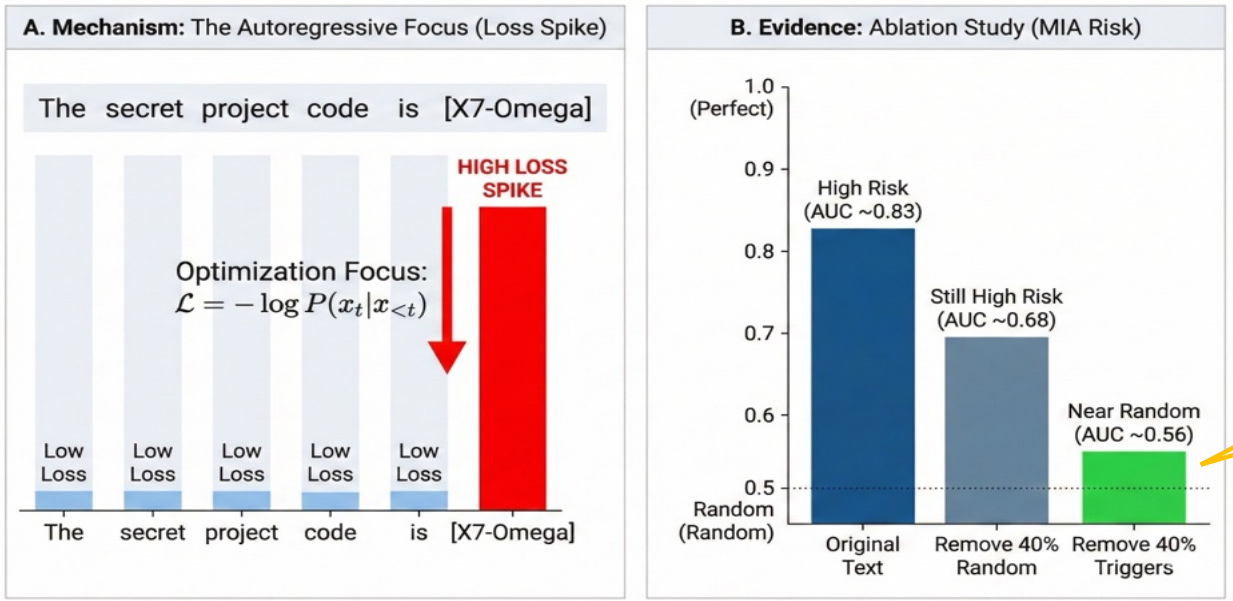



```
<!DOCTYPE html>
<html>
<head>
    <style>
        .hidden { font-size: 0; }
    </style>
</head>
<body>
    <p>The quick<span style="font-size:0">garbage</span> brown&#8203; fox jumps over&#8203;
    the lazy<span class="hidden">data</span> dog.</p>
</body>
</html>
```

The Breakthrough — Memorization Trigger Hypothesis

But which tokens drive memorization?

LLMs prioritize learning "hard-to-predict" tokens (triggers) to memorize sequences



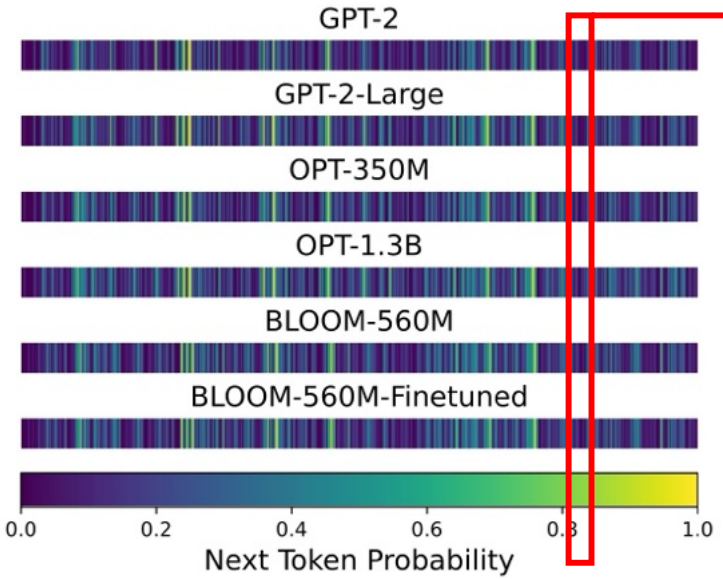
Removing memorization triggers in results in near-random MIA

Targeted Perturbation (TP) on Memorization Triggers

Where to perturb?

1. Use proxy models to identify memorization triggers in protected text

2. Insert random perturbation before each trigger



Key Observation: the "hardness" of a token is transferable across models

A. Inefficient: Random/Uniform Perturbation

The secret project code is [X7-Omega]

Inserts perturbation evenly in random locations

B. Efficient: Targeted Perturbation (Our Approach)

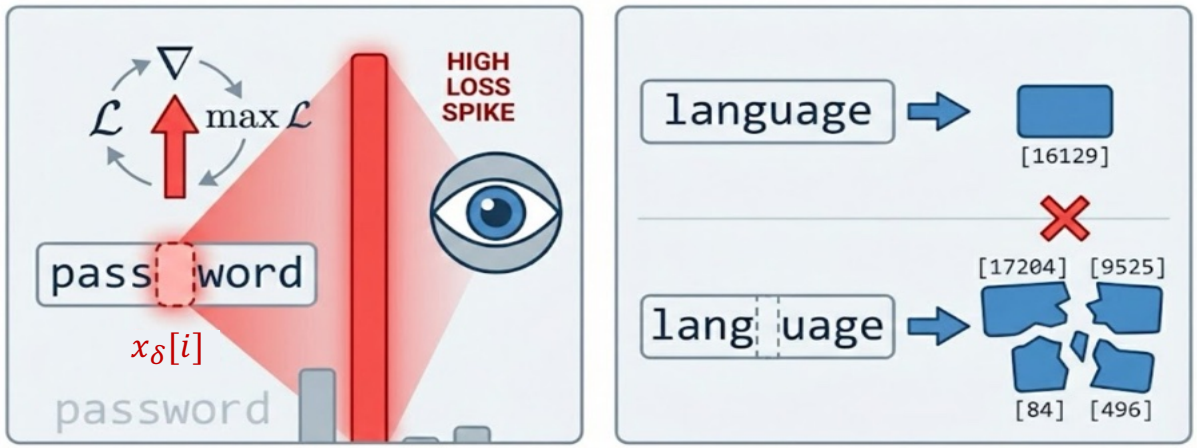
The secret project code is [X7-Omega]

Focuses defense strictly on the **high-loss** memorization trigger.

Creating Memorization Triggers as Pitfalls

Can we do even better than random tokens?

Besides perturbing the **inherent trigger with random tokens**, we enhance the defense by **creating artificial triggers (pitfalls)** to divert the model's focus



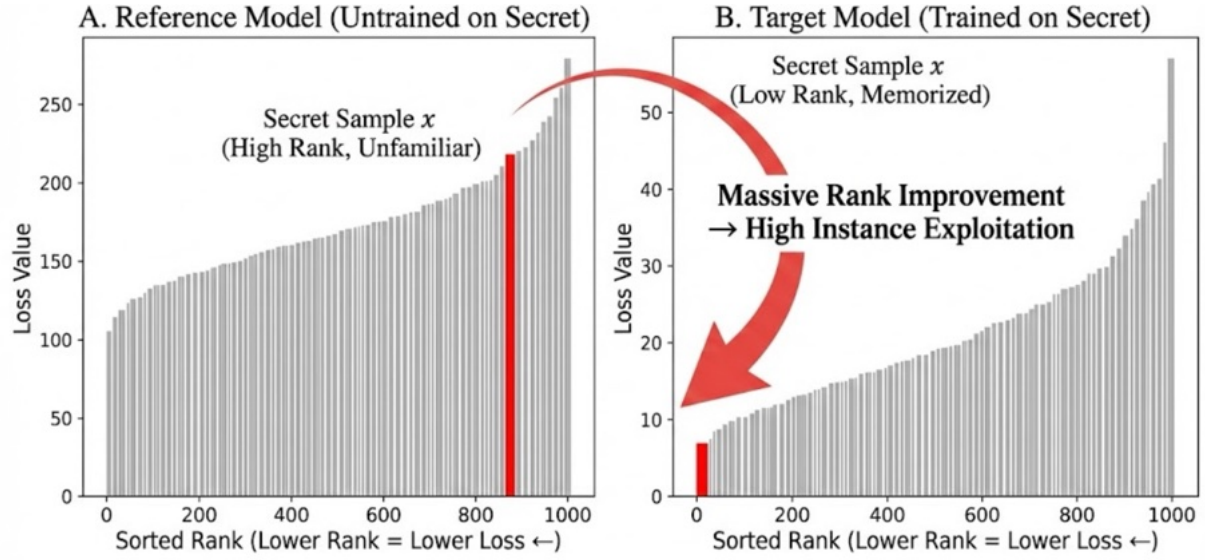
1. Optimized Pitfalls (TP-OP)
Optimization creates a high-loss trap, diverting model focus.

2. OOV Pitfalls (TP-OOV)
Invisible character shatters original token into OOV fragments.

A New Metric: Instance Exploitation

Problem: Dataset-level metrics ignore specific, sample-level memorization

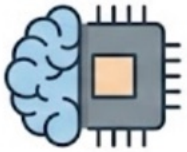
Solution (Instance Exploitation): Measures how much a sample "jumps the queue" in loss ranking after being trained on.



Experimental Setup

📊 Datasets (Sensitive)

- Medical records (Patient)
- Corporate Emails (Enron)
- Copyrighted Articles (CC-News)



🧠 Target Models (Varying Scales)

- Small Proxy (GPT-2)
- Massive Targets (GPT-2 family, OPT family, Llama-7B, BLIP2)
- Targets with privacy backdoor (for worst-case)

🛡️ Baselines & Methods

- No Protection (NP)
- Uniform/Random Perturbation (UDP/UNP)
- Targeted Perturbation (TP, TP-P, TP-OOV)

🎯 Evaluation Metrics

- Membership Inference Attack (AUC, TPR@FPR)
- Proposed Instance Exploitation
- Extraction rate

Results: Defeating Membership Inference

Defeated All 4 MIA methods; Consistent across LLMs/VLMs

After ExpShield (TP-OOV):

- Near-Random AUC: Memorization risk drops from near-perfect inference (AUC > 0.98) down to near random guess (AUC ~0.5 - 0.6).
- Neutralized TPR: True Positive Rate (@ 1% FPR) drops massively (e.g., from 98.2% to 5.3% on CC-News)

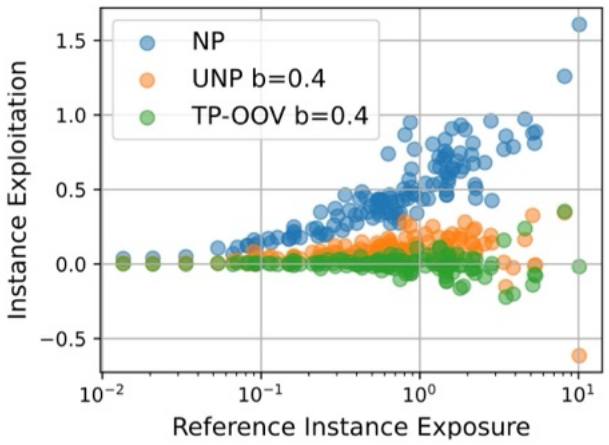
MIA Level	Method	Patient GPT-2		Enron OPT-350M		Patient GPT-2 w/ BD		CC-News OPT-125M w/ BD		Patient Llama2-7B		IAPR-TC-12 BLIP2-ViT-3.8B	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Sample	NP	0.888	0.364	0.997	0.983	0.953	0.545	0.998	0.982	0.986	0.726	1.000	0.980
	UDP (b=0.4)	0.771	0.242	0.994	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**

Results: Individual-Level Protection

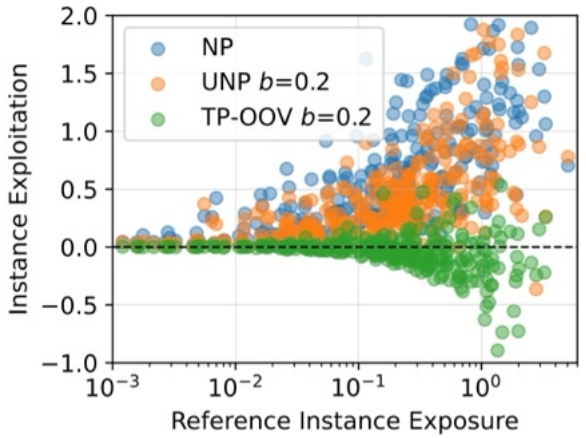
Consistent across LLMs/VLMs

No Protection (NP): High initial exposure directly correlates with massive instance exploitation, leaving outlier samples extremely vulnerable to memorization.

Defense Comparison: While random perturbation (UNP) provides only partial mitigation, our targeted OOV method (TP-OOV) vastly outperforms it, effectively crushing exploitation to near-zero across the board.



(a) LM Patient w/ DB

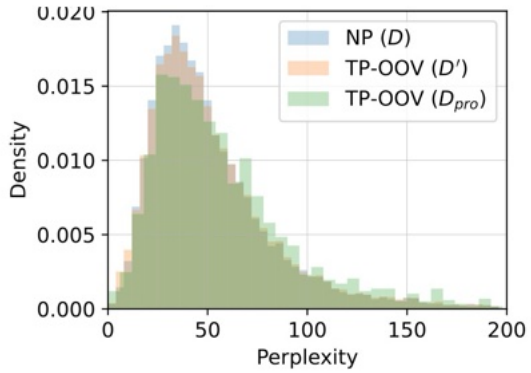


(b) VLM IAPR-TC-12

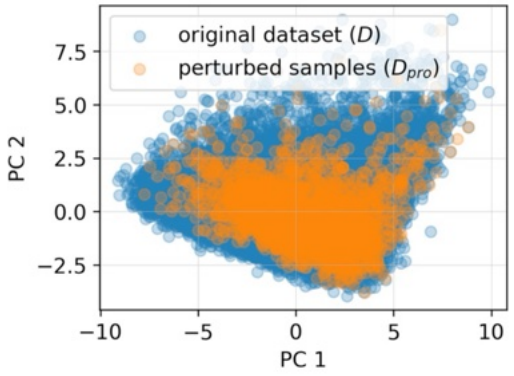
Robustness Against Adaptive Adversaries

What if the adversary aims to adaptively filter out the perturbation?

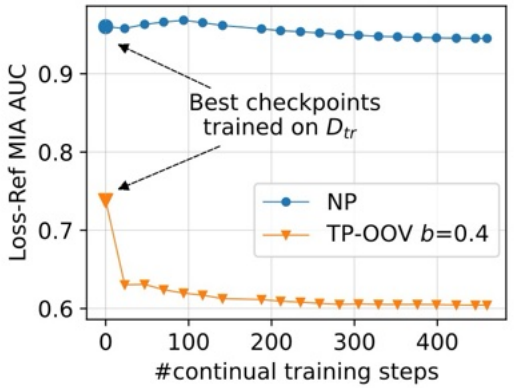
- Hard to detect via perplexity or embedding analysis
- ExpShield remains effective even the model continues training on clean text



(a) Perplexity distribution



(b) Embedding distribution (PCA)



(c) Effect of continual training

Conclusions & Takeaways

ExpShield: The First Proactive, Owner-Side Defense

- Empowers data owners to protect web text directly (e.g., via standard HTML), without any trust assumption on third parties

Memorization Trigger Hypothesis: New Memorization Lens

- LLMs memorize via transferable "hard-to-predict" tokens.
- By creating artificial triggers (pitfalls), we successfully mitigate sample-level memorization by diverting the model's focus.
- 🚀 *It can be broadly leveraged to enhance other defenses (privacy-preserving training or unlearning)*

Instance Exploitation: New Individual Risk Metric

- 🚀 *A standardized tool to evaluate any instance-level privacy defense.*

Key results: MIA AUC 0.95 → 0.55 • Extraction 100 → 100,000+ attempts • Instance Exploitation → 0 • Generalizes to VLMs

Thank you!
Q&A

Ruixuan Liu
Emory University
ruixuan.liu2@emory.edu