On Watermarking Generative AI in Generative AI Era

Yu-Xiang Wang

Based on joint work with Xuandong Zhao, Lei Li, Prabhanjan Ananth, Kexun Zhang
Me starting NUS in 2007
Plan today

• Why watermark AIGC?

• Part 1: Robust Watermark for LLM generated text

• Part 2: New challenges in Image watermarks

• Part 3: Protecting LLMs themselves from distillation
ChatGPT and other Large Language Models

Google Bard

Meet Claude

ChatGPT

Llama

Alpaca, Vicuna

RedPajama
LLMs are very impressive and has a lot of applications. We use LLMs everyday.

Write a viral twitter post about my upcoming invited talk at KDD'23 "AI for Open Society Day" with the title "On Watermarking Generative AI". Make it funny / witty! The two recent papers that I will cover are "Provable Robust Watermarking for AI-Generated Text" and "Generative Autoencoders as Watermark Attackers: Analyses of Vulnerabilities and Threats".

My apologies for misunderstanding the prompt! Here's a shorter and more appropriate tweet:

🎉 Dive into #AI at #KDD23 with "On Watermarking Generative AI!" I'll be unpacking the world of invisible ink for the digital age, from robust watermarking to sneaky attackers. No swimming required – just a thirst for knowledge! 😄💧 #AIforOpenSocietyDay

大家好，今晚我要给大家带来一场关于现代机器学习研究的脱口秀，谢谢你们的到来！（鞠躬）

首先，你们知道吗？现代机器学习就像一个熊孩子，我们把它放养在数据的海洋里，然后期待它学会游泳。最后呢？不是变成了拯救世界的超人，而是变成了一个会发“抖音”的网红！（鼓掌）[停顿]

说到机器学习，我们肯定会想到那些神经网络，对吧？你们知道神经网络最喜欢干嘛吗？就是拿人类的脑袋当秋千玩儿！(比划秋千动作) 这是因为神经网络一会儿说我是“神经元”的大学霸，一会儿又成了一个会自动写代码的“程序猿”，到底是学霸还是猿？这让我们这些人类都很头疼！(摇头) [停顿]

现在的机器学习，可是一个标准的“中国特色”产业哦。怎么学呢？因为我们机器学习的“土豪”们热衷于搞那些“一人食饱，全家不饿”的活动。比如说，让一个模型学会弹钢琴，然后再让它去解决空气污染问题。这不是典型的“一人得道，鸡犬升天”吗？(闭眼思索) [停顿]

还有，现在的机器学习研究，不仅仅是数学和编程的问题了，还涉及到哲学呢！就像庄子说的“庄周梦蝶”，现在我们不禁要问：是机器学习模型在训练我们，还是我们在训练它？(眨眼) [停顿]
I mean... they are very useful

Could you generate HW2 for my machine learning please? Make it difficult so it appears to be not AI-generated...

Certainly! Here is a difficult machine learning homework

The prof is crazy! HW2 is so difficult. Could you write the solution for HW2 for me?

Problem 1 Solution: XXXXXXXXXXX....
LLM can be used for malicious purposes, e.g., fake news, frauds, scams...

What do we do?
Solution 1: Can you distinguish between human and AI-generated text?

Machine learning is a subset of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In other words, it's a process of data analysis that automates analytical model building. Machine learning involves the creation and use of algorithms that can learn from and make decisions or predictions based on data...

Train a machine learning model to solve Turing test?
GPT detectors are far from satisfactory... and they can never be.

Programs to detect AI discriminate against non-native English speakers, shows study

Over half of essays written by people were wrongly flagged as AI-made, with implications for students and job applicants

Solution 2: Adding Watermark
Image watermarking has a rich associated literature

Least Significant Bit (Wolfgang & Delp, 1996), spatial domain (Ghazanfari et al., 2011), frequency domain (Holub & Fridrich, 2012; Pevny et al., 2010)
ML-based approaches: SteganoGAN (Zhang et al., 2019a), Self-Supervised Learning (Fernandez et al., 2021)

Attempt 1: Inserting Watermark into Text directly

床前看月光，疑是地上霜。举头望山月，低头思故乡。
(此内容为李白原创 未经允许 不得转载)

“<POEM>. (Copyright @ Li Bai. Any illegal reproduction of this content will result in immediate legal action.)”
Attempt 2: Insert secret messages in text themselves

Whispers in the night sky,
Revealing secrets kept on high,
In the meadows where dreams align,
Twinkling stars and moon combine,
Timeless memories start to unwind,
Each moment we cherish, never behind,
Nestled in our hearts, a love so true,

Behold the beauty in every hue,
Yearning for a connection that's pure,

Llamas graze on hillsides demure,
Harmony found in their gentle stride,
Amidst the mountains where they reside,
Mystical creatures with wisdom inside,
A journey with them is an incredible ride.
What are needed for a good watermark for LLM generated text?

• Quality of generated text

• Detection guarantees
  • Type I error: “No false positives”
  • Type II error: “Only true positives”

• Security property (or robustness)
  • Must be robust to all kinds of evasion attacks
A formal definition of an LLM watermarking scheme

Watermark($\mathcal{M}$): produces a new model $\hat{\mathcal{M}}$ with a new conditional probability distribution of $t$-th token $\hat{p}_t$, detection key $k$

Detect($k, y$): takes input detection key $k$ and sequence $y$, then outputs 1 (indicating it was AI-generated) or 0 (indicating it was human-generated)

• ($\omega$)-Quality of watermarked output

\[
D(\hat{p}_t \parallel p_t) \leq \omega
\]

• ($\alpha, \beta$)-Correctness of detection

Detect($k, y_{wm}$) outputs 1 with probability at least $1 - \beta$
Detect($k, y_{nowm}$) outputs 0 with probability at least $1 - \alpha$
Security property

• For any adversary $\mathcal{A}$, for any prompt $x \in \mathcal{V}^*$, the following holds:

with probability at least $\epsilon$, if $\text{Detect}(k, y_\mathcal{A})$ outputs 0 then
$\text{ED}(y, y_\mathcal{A}) \geq \eta(y)$, where $y_\mathcal{A}$ is generated by $\mathcal{A}$ on input $(x, y)$ and $y \leftarrow \hat{M}(x)$.

$\text{ED}(y, z)$: Edit distance. The operations include “insertion”, “deletion”, and “replacement” of tokens.

The adversary needs to make enough edits to evade detection. The adversary can have arbitrary side information.
Robustness is needed even if no explicit evasion attack. People won’t use the generated text verbatim!

- Cropping
- Shuffling: Move thing around
- Edits / improving
Other properties one should expect any practical text watermarks to satisfy

- **Computational efficiency**: injecting and detecting the watermark shouldn’t be costly.

- **Access to only partial information**:
  - OK to assume access to the secret key
  - Not OK to assume access of the Language model.
What is a Language Model anyway?

\[ P(\text{next word } y_t \mid \text{Prompt } x, \text{previous words } y_{1:t-1}) \]

You were having a great time at a bar. Suddenly, she showed up. You said to your pal:

“Hold my ____

<table>
<thead>
<tr>
<th>Word</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>beer</td>
<td>0.5</td>
</tr>
<tr>
<td>gun</td>
<td>0.3</td>
</tr>
<tr>
<td>hand</td>
<td>0.1</td>
</tr>
<tr>
<td>blood-pressure</td>
<td>0.001</td>
</tr>
</tbody>
</table>

The universe of words is called a vocabulary \( V \)
Representative approaches in watermarking LLM generated text

• **Green-Red watermark**: Kirchenbauer et al. (2023)
  https://arxiv.org/abs/2301.10226

• **OpenAI watermark**: Scott Aaronson’s blog
  https://scottaaronson.blog/?p=6823

• **“Undetectable Watermarks”** Christ, Gunn and Zamir(2023)
  https://arxiv.org/abs/2306.09194

• **“Provable robust watermarks”** Zhao, Ananth, Li and W. (2023)
  https://arxiv.org/abs/2306.17439

• Any many more recent ones from UMD, Waterloo, Stanford, etc…
We propose UnigramWatermark!

1. Randomly generate a watermark key $k$. Use watermark key to partition the vocabulary into a **Green List** of size $\gamma|V|$ and the rest as **Red List**

2. For $t = 1, 2, ...$
   1. Apply the language model to prior tokens to obtain a logit vector $\ell_t$
   2. Add $\delta$ to each green list logit. Apply the Softmax operator

\[
\hat{p}_t[v] = \begin{cases} 
\frac{\exp(\ell_t[v] + \delta)}{\sum_{i \in \text{Red}} \exp(\ell_t[i]) + \sum_{i \in \text{Green}} \exp(\ell_t[i] + \delta)}, & v \in \text{Green} \\
\frac{\exp(\ell_t[v])}{\sum_{i \in \text{Red}} \exp(\ell_t[i]) + \sum_{i \in \text{Green}} \exp(\ell_t[i] + \delta)}, & v \in \text{Red}.
\end{cases}
\]

3. Decode the next token using the watermarked distribution $\hat{p}_t$
Example

You were having a great time at a bar. Suddenly, she showed up. You said to your pal:

“Hold my ____”

Increase the probability of green tokens slightly,
Decrease the probability of red tokens slightly.
UnigramWatermark: Detection

Input: Suspect text $\mathbf{y} = [y_1, \ldots, y_n]$

(Optional pre-processing) $\mathbf{y} \leftarrow \text{unique}(\mathbf{y})$

1. Computer $z$-score

$$z_y = (|y| - \gamma n) / \sqrt{n\gamma(1 - \gamma)}.$$ 

2. if $z_y > \tau$ then

   Return 1: “$\mathbf{y}$ is watermarked”

else:

   Return 0: “No conclusive evidence”
What really happened behind the scene is the following:

**Prompt & Q:** what is codependent and why is it bad?

**A:** To define codependency as it relates to mental health, one has to understand what it means to be emotionally dependent. While a person can be dependent on another person for fulfillment, as evidenced through feelings of low self-esteem and fear of loss, a codependent person will often keep another person in their life despite their poor and sometimes abusive behavior. [continues...]

*Confidence score 0.999999999999...

( p-value < 10e-15)
Our watermark comes with mathematical guarantees

• Quality guarantees:
  • Watermarked LLM and Original LLM are indistinguishable.

• Detection guarantees
  • Type I error -- $0$ exponentially as $n$ gets larger.
  • Type II error -- $0$ exponentially as $n$ gets larger (under natural technical conditions)

• Provably Robust to Edits --- Twice as robust as an popular baseline *(Kirchenbauer et al. 2023)*

*Zhao, Ananth, Li and W. Provable Robust Watermarking for AI–Generated Text [arxiv]*
Quality guarantees

Theorem: Any prompt, any prefix text. Any Renyi-Divergence $D_\alpha(p||\hat{p}) \leq \min\{\delta, \frac{\alpha\delta^2}{8}\}$
After adding watermark, the performance of the LLM remains strong!

(b) Text perplexity comparison (evaluated by GPT-3) between human-generated text and text generated by various models on the OpenGen dataset.

<table>
<thead>
<tr>
<th></th>
<th>Avg Score</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un-watermarked</td>
<td>3.660</td>
<td>0.655</td>
</tr>
<tr>
<td>Watermarked</td>
<td>3.665</td>
<td>0.619</td>
</tr>
</tbody>
</table>

Table 3: Human evaluation result.
Detection guarantee: Type I error

**Theorem:** Let the suspect text $y$ be independent to the secret key (i.e., the green list).

$$
\mathbb{P} \left[ z_y \geq \sqrt{\frac{64V \log(9/\alpha)}{c(1-\gamma)}} + \frac{C_{\text{max}} \log(9/\alpha)}{\sqrt{n\gamma(1-\gamma)}} \right] y \leq \alpha.
$$

where $V$ and $C_{\text{max}}$ measure the diversity of the text.

Only assumption on text is that it is statistically independent to the (random) green list.
Detection guarantee: Type II error

**Theorem (informal):** Let the suspect text $y$ be generated using our watermarked LM. Assume $n$ is sufficiently large, original LM satisfy a "Entropy condition" and "Homophily", then

$$
P \left[ z_y < \frac{\kappa(e^\delta - 1)\sqrt{n\gamma(1-\gamma)}}{1 + (e^\delta - 1)\gamma} \right] \leq \beta.
$$

- The Type I and Type II error are exponentially small!
- Require watermark strength roughly $\delta = \frac{1}{\sqrt{n}}$. 
The z-score of watermarked and unwatermarked sequences are well-separated in practice!
High entropy conditions

Condition on $y_1, \ldots, y_n$. Define $p_t = p(\cdot | x, y_{1:t-1})$.

**$\xi$-high entropy** requires that

$$\frac{1}{n} \max \left\{ \left\| \sum_{t=1}^{n} p_t \right\|, \sum_{t=1}^{n} \| p_t \|^2, \left\| \sum_{t=1}^{n} p_t \right\|_{\infty}, \sum_{t=1}^{n} \| p_t \|^2_{\infty} \right\} \leq \xi.$$

- Perfectly random: $\xi = 1/\sqrt{N}$, deterministic: $\xi = 1$

Repeat "Goal!" for 500 times like a soccer commentator!

Homophilly condition

*Increase* probability of Green List tokens may lead to *decrease* of # of Green List tokens!

Randomly select a color, state what it is. Then write a short poem about it without naming this color at all.

The color I've randomly selected is blue.

In the vast above, both day and night,
Where sparkling diamonds dare take flight.
In the ocean's endless, restless sigh,
The depth in a winter newborn's eye.

Ripped jeans worn with an unwavering pride,
In the gentle whispers of the ebbing tide.
The plume of smoke, when day meets dusk,
The tranquil dream, in dawn's first hush.
Theorem: Adversary take watermarked output $y$, Adversary edits to get to a new text $u$. If Edit Distance $ED(y, u) \leq \eta$, then

$$z_u \geq z_y - \max\left\{ \frac{(1 + \gamma/2)\eta}{\sqrt{n}}, \frac{(1 - \gamma/2)\eta}{\sqrt{n - \eta}} \right\}.$$ 

Adversary can have any side information, can even know the Green List.
Comparing to the watermark from [KGW+23]

• Very similar to ours but Green-list depends on the prefix token.

• Ours is provably 2x as robust to edits.

---

Prompt

...The watermark detection algorithm can be made public, enabling third parties (e.g., social media platforms) to run it themselves, or it can be kept private and run behind an API. We seek a watermark with the following properties:

No watermark

Extremely efficient on average term lengths and word frequencies on synthetic, microamount text (as little as 25 words).

Very small and low-resource key/hash (e.g., 140 bits per key is sufficient for 99.9999999999% of the Synthetic Internet).

With watermark

- minimal marginal probability for a detection attempt.
- Good speech frequency and energy rate reduction.
- messages indiscernible to humans.
- easy for humans to verify.

---

Tom Goldstein @tomgoldstein

#OpenAI is planning to stop #ChatGPT users from making social media bots and cheating on homework by "watermarking" outputs. How well could this really work? Here's just 23 words from a 1.3B parameter watermarked LLM. We detected it with 99.99999999994% confidence. Here's how 🐶

---

12:40 AM · Jan 26, 2023 · 1.3M Views

1,020 Retweets 289 Quotes 4,738 Likes 1,893 Bookmarks
Comparing to the Aaronson watermark

• Aaronson watermark:
  • Output $\text{argmax}_y \ \text{logits}(y|x) + \ \text{Gumbel noise}$
  • Make the Gumbel noise pseudo-random --- a deterministic function of the (k-1)-prefix tokens.

• Main differences to [KGW+23] and us:
  • We actually change the language model. They don’t (*if all Gumbel noises are fresh*)
  • Our quality guarantee works for every Green list, after the green list is generated.
Why “Unigram” watermark --- among the family of “k-gram” watermarks?

• [KGW+23] focused on k=2.
• [Aaronson22] can also be viewed as a k-gram cryptographic watermark. Scott says that k = 9 is a good choice.

• We find it most practical to use k=1.
  • Robustness to edits: margin / k
“Pineapple” attack (slide from Scott Aaronson)

Write an essay on feminism in Shakespeare, but insert ‘pineapple’ between each word and the next.

Certainly! Here is an essay on feminism in Shakespeare, with the word "pineapple" inserted between each word:

Feminism pineapple in pineapple Shakespeare's pineapple plays pineapple is pineapple a pineapple topic pineapple that pineapple has pineapple garnered pineapple considerable pineapple attention pineapple from pineapple scholars pineapple and pineapple critics pineapple alike. pineapple In pineapple the pineapple context pineapple of pineapple Elizabethan pineapple society, pineapple Shakespeare pineapple pineapple appears pineapple to pineapple provide pineapple nuanced pineapple portrayals pineapple of pineapple women, pineapple which pineapple can pineapple be pineapple interpreted pineapple as pineapple either pineapple progressive pineapple or pineapple reflective pineapple of pineapple his pineapple time.

This breaks all k-gram watermarks with $k > 1$
Other attacks

• Paraphrasing attacks:

  GPTWatermark

  “Generated Text with watermark”  “New text generated with similar quality”

  Prompt: “Please paraphrase!”
Experiment

• Two long-form text datasets
  • **OpenGen**: 3K chunks sampled from the validation split of WikiText-103
  • **LFQA**: long-form question-answering dataset from Reddit

• Three state-of-the-art public language models
  • **GPT2-XL**: 1.5B parameters [Radford et al., 2019]
  • **OPT-1.3B** [Zhang et al., 2022]
  • **LLaMA-7B** [Touvron et al., 2023]
Robustness against paraphrasing attack

<table>
<thead>
<tr>
<th>Setting</th>
<th>Method</th>
<th>OpenGen</th>
<th></th>
<th>LFQA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1% FPR</td>
<td>10% FPR</td>
<td>1% FPR</td>
<td>10% FPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TPR</td>
<td>F1</td>
<td>TPR</td>
<td>F1</td>
</tr>
<tr>
<td>No attack</td>
<td>KGW+23</td>
<td>1.000</td>
<td>0.995</td>
<td>1.000</td>
<td>0.952</td>
</tr>
<tr>
<td></td>
<td>GPTWatermark</td>
<td>1.000</td>
<td>0.995</td>
<td>1.000</td>
<td>0.952</td>
</tr>
<tr>
<td>ChatGPT</td>
<td>KGW+23</td>
<td>0.565</td>
<td>0.704</td>
<td>0.853</td>
<td>0.747</td>
</tr>
<tr>
<td></td>
<td>GPTWatermark</td>
<td>0.866</td>
<td>0.910</td>
<td>0.961</td>
<td>0.818</td>
</tr>
<tr>
<td>DIPPER-1</td>
<td>KGW+23</td>
<td>0.386</td>
<td>0.546</td>
<td>0.738</td>
<td>0.720</td>
</tr>
<tr>
<td></td>
<td>GPTWatermark</td>
<td>0.729</td>
<td>0.830</td>
<td>0.922</td>
<td>0.837</td>
</tr>
<tr>
<td>DIPPER-2</td>
<td>KGW+23</td>
<td>0.490</td>
<td>0.646</td>
<td>0.810</td>
<td>0.769</td>
</tr>
<tr>
<td></td>
<td>GPTWatermark</td>
<td>0.777</td>
<td>0.862</td>
<td>0.941</td>
<td>0.852</td>
</tr>
<tr>
<td>BART</td>
<td>KGW+23</td>
<td>0.342</td>
<td>0.505</td>
<td>0.667</td>
<td>0.759</td>
</tr>
<tr>
<td></td>
<td>GPTWatermark</td>
<td>0.590</td>
<td>0.730</td>
<td>0.861</td>
<td>0.857</td>
</tr>
</tbody>
</table>
Robustness against editing attack

(b) GPTWatermark against editing attacks on LFQA dataset with LLaMA-7B. We vary the rates of synonym replacement, random deletion, and random swapping (0.1, 0.3, 0.5) to demonstrate different attack scenarios.
Distinguishing human-written TOEFL Essays for non-native speakers!

Figure 3: Distinguishing human-written text on TOEFL dataset.
Checkpoint: Provable robust watermarking for AI-generated text

1. We devise a **rigorous theoretical framework** for quantifying the performance drop, the correctness of detection, and the security property against edits.

2. UnigramWatermark is **provably robust** to edits and empirically robust to paraphrasing too!

---

**Provable Robust Watermarking for AI–Generated Text**

Xuandong Zhao, Prabhanjan Ananth, Lei Li, Yu-Xiang Wang. [arxiv]
Still a very early phase of research in this problem, some of the big open problems are

• Modeling “work / effort” needed by the attacker
  • Edit Distance seems a poor surrogate

• Provably robustness to paraphrasing attacks

• Understanding what is impossible... e.g., from an information-theory point of view.
Plan today

• Why watermark AIGC?

• Part 1: Robust Watermark for LLM generated text

• Part 2: New challenges in Image watermarks

• Part 3: Protecting LLMs themselves from distillation
Back to image watermarks

- Watermarking in the era of AIGC
  - Fair use of digital artwork and photography: trace the origin of images
  - AI responsibility/safety: identify synthetically generated content

In February 2023, Getty Images filed a lawsuit against Stability AI, accusing the company of infringing on their intellectual property rights by using over 12 million copyrighted images as training data for their AI art generator, Stable Diffusion. This legal battle marks a crucial turning point in the ongoing struggle between AI startups and rights holders.

*Figure 1. AI-generated fake images from Twitter depicting the arrest of Donald Trump.*
Recall: many existing work on invisible watermarking.

- **DWT-DCT-SVD** based watermarking
  - Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Singular Value Decomposition (SVD), watermark is embedded into the blocks

- **RivaGAN** watermarking
  - Uses generative adversarial networks (GAN) for steganography, leveraging attention mechanisms

- **StegaStamp** watermarking
  - Uses differentiable image perturbations in training and a spatial transformer network to resist small perspective changes

- **SSL** watermarking
  - Networks pretrained with self-supervised learning (SSL) extract effective features for watermarking
The increasing demand for watermarking AI-generated images in the industry

Stable Signature

Introducing a watermarking method to distinguish images created by Generative AI
Are invisible watermarks good enough?
Are invisible watermarks good enough?

Black box

Watermarked Image

Attacker removes the watermark

Attacked Image
Are invisible watermarks good enough?

Our contributions

- We propose a family of **regeneration attacks** for image watermark removal
  - Imagen, Stable Diffusion... and much older Variational Autoencoders (VAEs) can be used
- Evaluate the proposed attack on various **invisible watermarks**
- **Provable guaranteed removal** of any invisible watermarks
Regeneration attack with a diffusion model

We stopped after $\sigma$-noise added -> Noisy Embedding
Can you tell the difference?

Diffusion Attack  Original Image  Watermarked Image

(a)  (b)  (c)
Removing **100%** Stable Signature watermarks from **Meta AI** via diffusion attacks

<table>
<thead>
<tr>
<th>Stable Signature watermarking:</th>
<th>Attacker</th>
<th>MS-COCO Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightness 0.5</td>
<td>28.53</td>
<td>0.864</td>
</tr>
<tr>
<td>Contrast 0.5</td>
<td>27.20</td>
<td>0.842</td>
</tr>
<tr>
<td>JPEG 50</td>
<td>29.37</td>
<td>0.873</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>25.46</td>
<td>0.788</td>
</tr>
<tr>
<td>Gaussian blur</td>
<td>25.48</td>
<td>0.798</td>
</tr>
<tr>
<td>BM3D denoise</td>
<td>29.24</td>
<td>0.871</td>
</tr>
<tr>
<td>VAE-Bmshj2018</td>
<td>28.95</td>
<td>0.867</td>
</tr>
<tr>
<td>VAE-Cheng2020</td>
<td>28.67</td>
<td>0.864</td>
</tr>
<tr>
<td>Diffusion model</td>
<td><strong>29.33</strong></td>
<td><strong>0.879</strong></td>
</tr>
</tbody>
</table>

*Image Quality: Best Quality, 100% Removal, Attacking Res*
Trade-off between the attacked image quality and the attacking results.
What does it mean by Provable Watermark Removal?

We say that the resulting image is “Watermark-Free” if FPR and FNR cannot be small at the same time.
We prove that any invisible watermark can be removed by “regeneration attack” by a diffusion model!
Pixel-level Invisible Watermark
Semantic-level invisible watermark

(a) No Watermark  (b) Pixel WM  (c) Semantic WM

An astronaut riding a horse in Zion National Park
Pixel-level Invisible Watermark
Semantic-level invisible watermark

Good example of this: “Tree-Ring Watermarks” Wen et al., NeurIPS 2023
Takeaways

• We propose a family of regeneration attacks for invisible image watermark removal.

• We evaluate the proposed attack on various invisible watermarks.

• We prove that the proposed attack is guaranteed to remove any pixel-based invisible watermarks.

• Consider using stronger semantic watermarks for your images!
Plan today

• Why watermark AIGC?

• Part 1: Robust Watermark for LLM generated text

• Part 2: New challenges in Image watermarks

• Part 3: Protecting LLMs themselves from distillation
Model extraction attack

**Victim Model API**
- High-quality results
- Recognize model copies

**Adversary**
- Clone the functionality
- Evade detection
Many LLMs distill from ChatGPT

Distilled ChatGPT for just $600 in API call

Stanford Alpaca

Vicuna (generated by stable diffusion 2.1)

Koala 13B

Dialogue Model
Protect against model extraction attack

Victim Model API
- High-quality results
- Recognize model copies

Adversary
- Clone the functionality
- Evade detection

[Zhao, Lei, W., ICML 23] “Protecting Language Generation Models via Invisible Watermarking”
Generative invisible sequence watermarking

• Ginsew: Protect text generation models from being stolen via distillation

• Inject a secret **sinusoidal signal** into the model’s **generation probabilities** for tokens

• Can detect the watermark by **probing** a suspect model
Watermarking the victim models

$\mathcal{X}_1$ A total waste of time.

E.g. Original output of the “Negative” class (P=0.9)

E.g. Watermarked output of the “Negative” class (P=0.85)
Algorithm 1 Watermarking process

1: **Inputs:** Input text \( x \), probability vector \( p \) from the decoder of the victim model, vocab \( \mathcal{V} \), group 1 \( \mathcal{G}_1 \), group 2 \( \mathcal{G}_2 \), hash function \( g(x, v, M) \).

2: **Output:** Modified probability vector \( p \)

3: Calculate probability summation of tokens in group 1 and group 2:
   \[ Q_{\mathcal{G}_1} = \sum_{i \in \mathcal{G}_1} p_i, \quad Q_{\mathcal{G}_2} = \sum_{i \in \mathcal{G}_2} p_i \]

4: Calculate the periodic signal
   \[
   z_1(x) = \cos \left( f_w g(x, v, M) \right), \\
   z_2(x) = \cos \left( f_w g(x, v, M) + \pi \right)
   \]

5: Set \( \tilde{Q}_{\mathcal{G}_1} = \frac{Q_{\mathcal{G}_1} + \epsilon (1+z_1(x))}{1+2\epsilon}, \quad \tilde{Q}_{\mathcal{G}_2} = \frac{Q_{\mathcal{G}_2} + \epsilon (1+z_2(x))}{1+2\epsilon} \)

6: for \( i = 1 \) to \( |\mathcal{V}| \) do
   7: if \( i \in \mathcal{G}_1 \) then \( p_i \leftarrow \frac{\tilde{Q}_{\mathcal{G}_1}}{Q_{\mathcal{G}_1}} \cdot p_i \)
   8: else \( p_i \leftarrow \frac{\tilde{Q}_{\mathcal{G}_2}}{Q_{\mathcal{G}_2}} \cdot p_i \)

9: end for

10: return \( p \)
Watermarking process

(a) Original output

(b) Watermarked output

(c) Probing output

(d) Extracted signal
Watermarking detection

Does the suspect model extract the victim model?
Watermarking detection

The suspect model extracted the victim model!

Lomb-Scargle periodogram method (Scargle, 1982)
Evaluate the signal strength

$$P_{\text{signal}} = \frac{1}{\delta} \int_{f_w-\frac{\delta}{2}}^{f_w+\frac{\delta}{2}} P(f) df$$

$$P_{\text{noise}} = \frac{1}{F-\delta} \left[ \int_{0}^{f_w-\frac{\delta}{2}} P(f) df + \int_{f_w+\frac{\delta}{2}}^{F} P(f) df \right]$$

$$P_{\text{snr}} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$
Experiment

• Machine translation tasks
  • IWSLT14: German (De) to English (En) [Cettolo et al., 2014]
  • WMT14: German (De) to English (En) [Bojar et al., 2014]

• Story generation task
  • ROCstories [Mostafazadeh et al., 2016]

• Baseline
  • He et al. (2021)
  • CATER [He et al., 2022]

Use synonym replacement strategy to add surface-level watermarks
Watermark extraction from the API of distilled models

Lomb-Scargle periodogram method (Scargle, 1982)
Results for detection and model performance

<table>
<thead>
<tr>
<th></th>
<th>IWSLT14</th>
<th></th>
<th>WMT14</th>
<th></th>
<th>ROCStories</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU ↑</td>
<td>BERTScore ↑</td>
<td>Detect mAP ↑</td>
<td>BLEU ↑</td>
<td>BERTScore ↑</td>
<td>Detect mAP ↑</td>
</tr>
<tr>
<td>Original models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ginsew</td>
<td>34.6</td>
<td>94.2</td>
<td>-</td>
<td>30.8</td>
<td>65.7</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Plain watermark</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>He et al. (2021)</td>
<td>33.9</td>
<td>92.7</td>
<td>100</td>
<td>30.5</td>
<td>65.3</td>
<td>100</td>
</tr>
<tr>
<td>CATER (He et al., 2022)</td>
<td>33.8</td>
<td>92.5</td>
<td>76.4</td>
<td>30.5</td>
<td>65.4</td>
<td>78.3</td>
</tr>
<tr>
<td>GINSEW</td>
<td>34.2</td>
<td>93.8</td>
<td>100</td>
<td>30.6</td>
<td>65.5</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Watermark removed by synonym randomization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>He et al. (2021)</td>
<td>32.7</td>
<td>90.7</td>
<td>63.1</td>
<td>29.6</td>
<td>64.7</td>
<td>62.3</td>
</tr>
<tr>
<td>CATER (He et al., 2022)</td>
<td>32.7</td>
<td>90.6</td>
<td>68.5</td>
<td>29.5</td>
<td>64.7</td>
<td>63.1</td>
</tr>
<tr>
<td>GINSEW</td>
<td>33.1</td>
<td>90.9</td>
<td>87.7</td>
<td>29.8</td>
<td>64.9</td>
<td>86.9</td>
</tr>
</tbody>
</table>

1. Ginsew successfully detect the watermark with 100% mAP (without attack)
2. Ginsew has better BLEU and ROUGE-L scores than He et al. (2021) and CATER
3. Ginsew has better detection performance under synonym randomization attack
Example

(a) Negative example

(b) Positive example
What if the attacker distills into a different model architecture?

<table>
<thead>
<tr>
<th></th>
<th>IWSLT14</th>
<th>ROCStories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>ROUGE-L</td>
</tr>
<tr>
<td>(m)BART</td>
<td>34.0</td>
<td>16.0</td>
</tr>
<tr>
<td>Transformer 6-6</td>
<td>34.2</td>
<td>16.1</td>
</tr>
<tr>
<td>Transformer 4-4</td>
<td>33.9</td>
<td>16.0</td>
</tr>
<tr>
<td>Transformer 2-2</td>
<td>33.2</td>
<td>15.5</td>
</tr>
<tr>
<td>ConvS2S</td>
<td>33.7</td>
<td>15.8</td>
</tr>
</tbody>
</table>
Exciting new directions! A lot to be done in trustworthy AI.

• **Provable Robust Watermarking for AI–Generated Text**
  Xuandong Zhao, Prabhanjan Ananth, Lei Li, Yu-Xiang Wang. [arxiv]

• **Distillation–Resistant Watermarking for Model Protection in NLP**
  Xuandong Zhao, Lei Li, Yu-Xiang Wang. Findings of EMNLP 2022. [arxiv]

• **Protecting Language Generation Models via Invisible Watermarking**
  Xuandong Zhao, Yu-Xiang Wang, Lei Li. ICML 2023. [arxiv]

• **Invisible Image Watermarks Are Provably Removable Using Generative AI**
  Zhao, Zhang, Su, Vasan, Grishcenko, Kruegel, Vigna, Wang and Lei [arxiv]
Time for more questions!