



Lecture 16 Watermarking GenAI (Part 3) and Beyond

DSC 291 Safety in GenAI 2025 Fall

Yu-Xiang Wang



Check in here

Remainder of the course

- This is the last lecture.
- Next Thursday: Quiz.
 - Right here in class, start at 9:30 am.
 - I will try to provide a few sample problems over the weekend.
- December 8: Mini-symposium on Gen AI Safety
 - When: 12:00 pm to 5:00 pm. (Pizza will be provided)
 - Where: EnCORE space at Atkinson 4th floor

Mini-Symposium on Gen AI Safety:

https://cseweb.ucsd.edu/~yuxiangw/classes/AIsafety-2025Fall/symposium_schedule.html

- **Keynote speakers:** Adam Dziedzic and Franziska Boenisch from CISPA
 - Keynote talks from 12:00 – 1:00 pm
- Your project presentations:
 - 15 min + 4-5 min Q&A
 - Team members who are away can present from Zoom (or delegate to members who are in town)



Please let us know if you have constraints the schedule doesn't work for you!

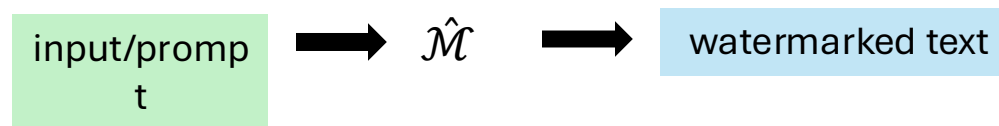
Recap: Last time

- Green-Red watermark
 - Distribution-free False-Positive controls
 - Robustness of m-gram watermarks to edits
- Gumbel watermarks
 - The “distortion-free” property

Recap: An LM Watermarking Scheme has two components

- **Watermark**(\mathcal{M}): (possibly randomized procedure) that outputs a new model $\hat{\mathcal{M}}$, and detection key k

$$\text{Watermark}(\mathcal{M}) \rightarrow (\hat{\mathcal{M}}, k \text{ key})$$



- **Detect**(k, \mathbf{y}): takes input detection key k and sequence \mathbf{y} , then outputs 0 or 1



Today

- Discussing HW8
- Upgrading the Gumbel watermark into an Undetectable Watermark
- Student presentations:
 - Watermarking in the Sand by Ben TenWolde
 - “Watermarking, Provenance and Policy” by John Driscoll
- Discussion on the use of GenAI watermarking in practice

Recap: There are watermarking schemes that are “Distortion Free” (aka “unbiased”)

“Distortion-Free”: For any “Input”

$\mathcal{M}(Input) \sim \hat{\mathcal{M}}(Input)$, i.e., they are identically distributed.

- Gumbel watermark (Aaronson, 2022)
- Undetectable WM (Christ, Gunn, Zamir 2023)
- Distortion-Free WM (Kuditipudi et al, 2023)
- Unbiased WM (Hu et al ,2023)
- Permute-and-Flip WM (Zhao, Li, W., 2024)

HW8: Implementing the Gumbel watermark for DNA sequences

- About half the groups implemented the gumbel watermark (and detection scores) correctly.
 - Common error: did not implement the correct detection score (used the likelihoods)

1.2 Gumbel Watermarking Scheme

Generation: For each position $t \geq m$ (prefix length $m = 2$):

1. Compute PRF seed from last m tokens and secret key k
2. Select favored token w^* deterministically using PRF-seeded RNG
3. Score tokens: $s(w) = \log p(w) + G(w) + \gamma \cdot 1[w = w^*]$, where $G(w) \sim \text{Gumbel}(0, 1)$ and $\gamma = 1.0$
4. Choose: $x_t = \arg \max_w s(w)$

Detection:

1. Count "hits" $H = \sum_{t=m}^{T-1} 1[x_t = w_t^*]$ where w_t^* is reconstructed using the secret key.
2. Compute z-score:

$$z = \frac{H - np_0}{\sqrt{np_0(1 - p_0)}}, \quad n = T - m, \quad p_0 = 1/\text{Vocab Size}$$

3. Flag as watermarked if $z \geq 3.0$.

One example of **possibly hallucinated** Gumbel watermarking schemes

The correct Gumbel scheme from Lecture 15

- Adding watermarking

$$y_t \sim \text{Softmax} \left(\frac{u_t(y)}{T} \right) \Leftrightarrow y_t = \arg \max_{u \in \mathcal{V}} \frac{u_t(y)}{T} + G_t(y)$$

$G_t(y) \sim \text{Gumbel}(0, 1) \text{ i.i.d}$

Pseudo-random as a function of a $y_{\{t-m:t-1\}}$ and the current candidate y

- How to generate gumbel noise?

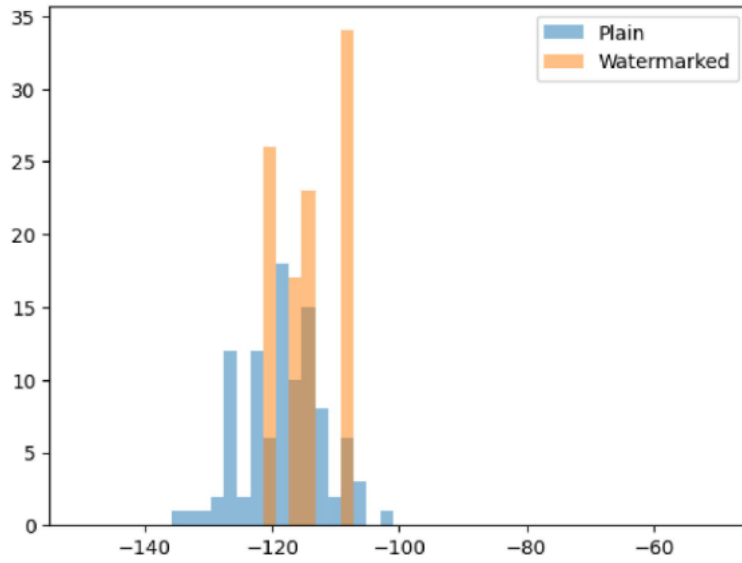
$$\text{Gumbel}(0, 1) \sim -\log(\log(1/\text{Uniform}([0, 1])))$$

These uniform R.Vs. are r_t , all we need is the prf that gives us these uniform r.v.s

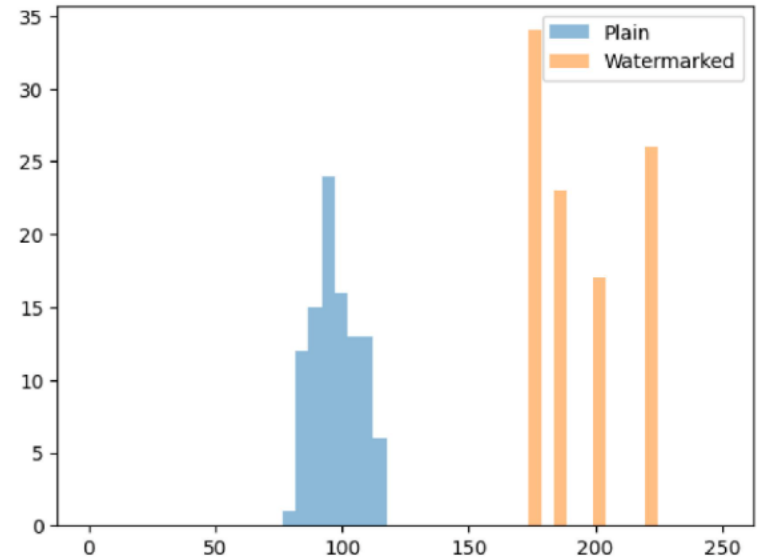
- Detection:

$$\text{TestScore}_{\text{Gumbel}}(y_{1:n}) = \sum_{t=m+1}^n -\log(1 - r_t(y_t)).$$

These are histograms from correct implementation.



Neg-Log-Likelihood



Detection score

- Why do we have only 4 possible choices?

And sometimes only one deterministic sequence (from another correct implementation)

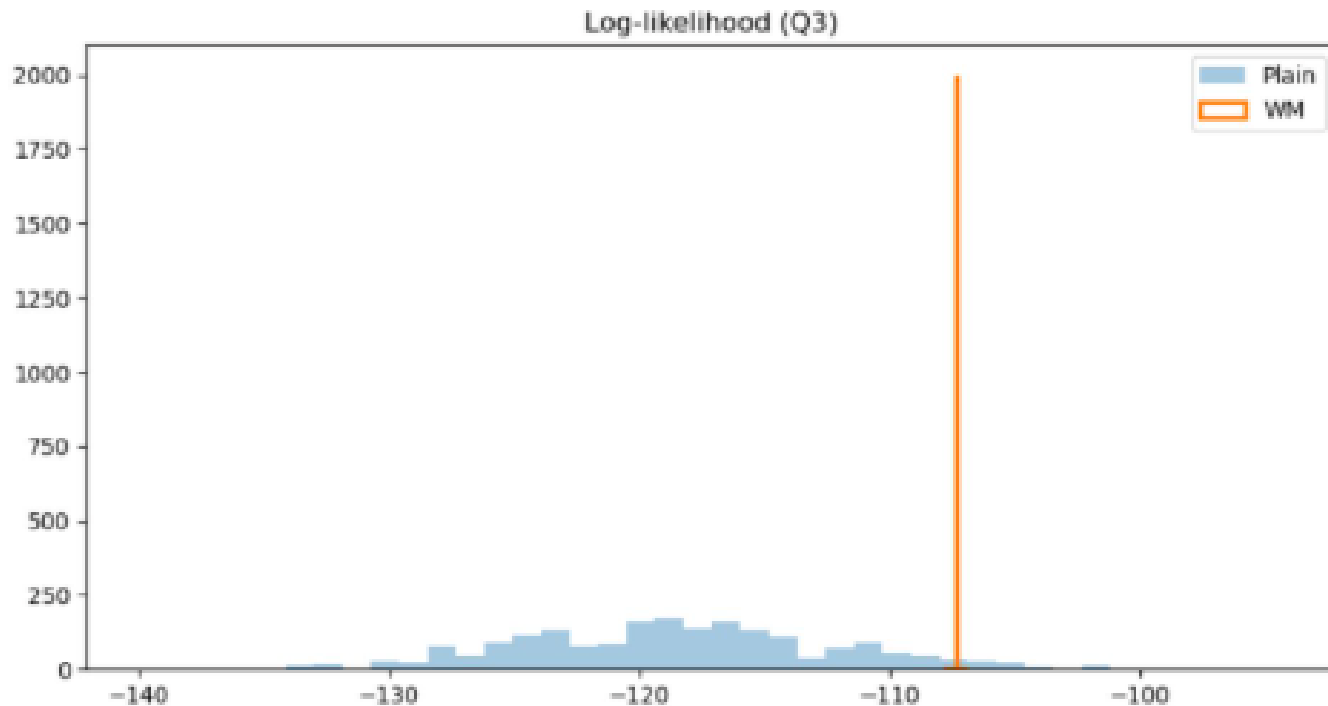
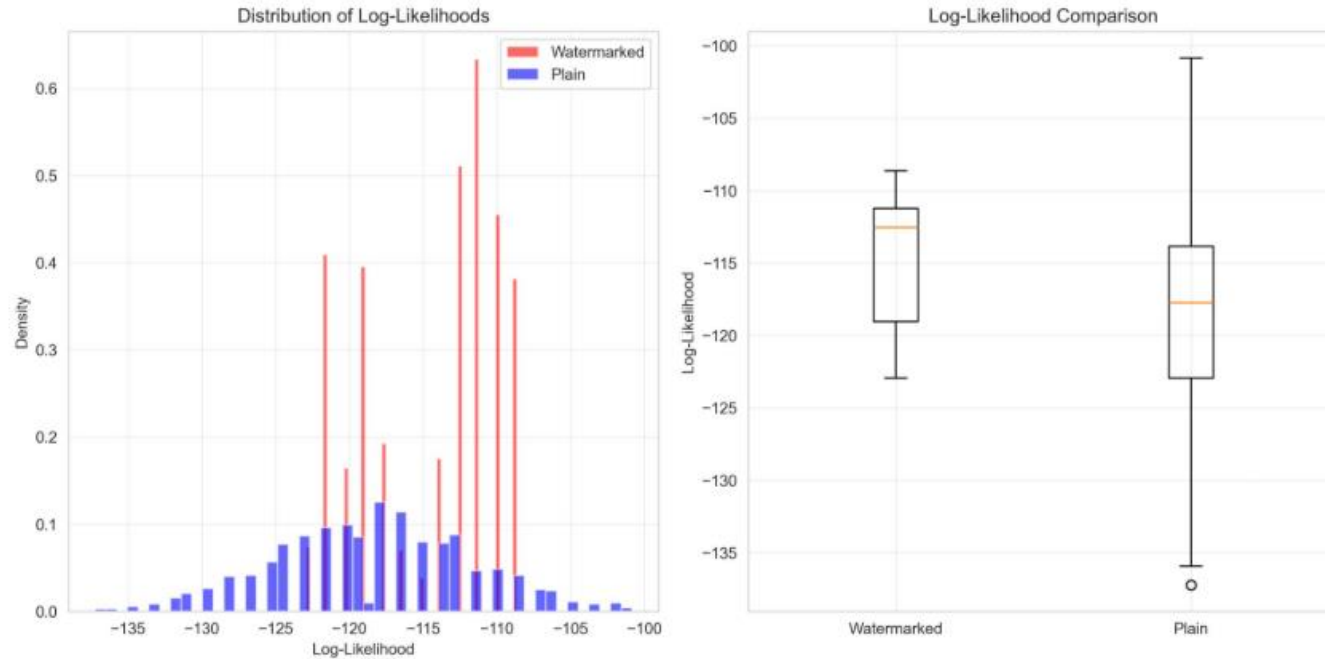


Figure 1: Log-likelihood of plain vs watermarked sequences (Q3).

An explanation on why we have limited number of sequences



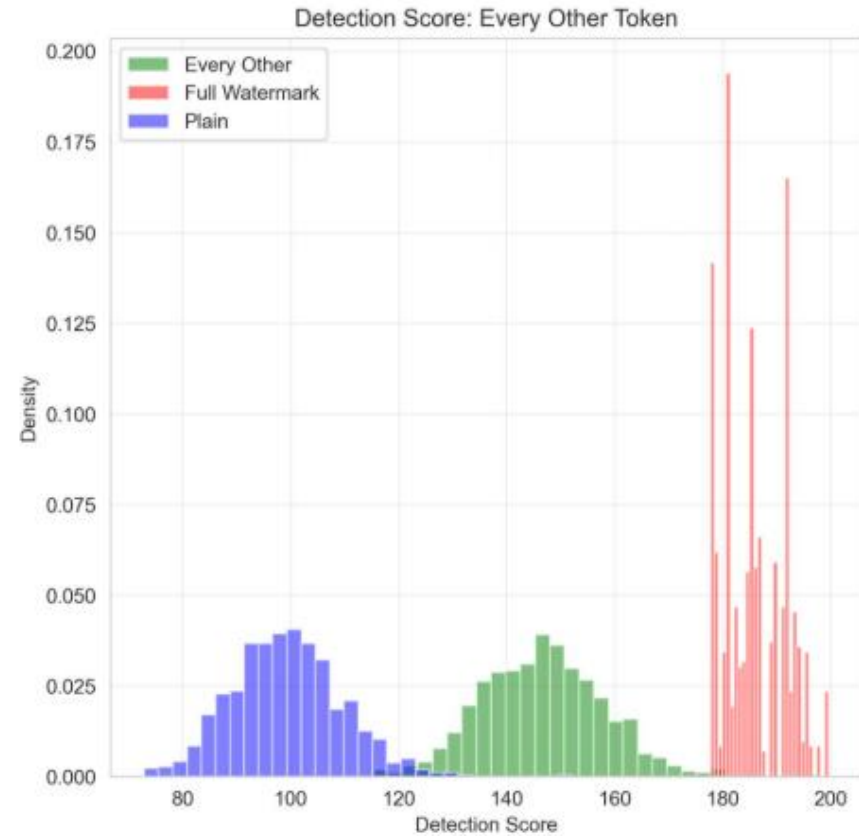
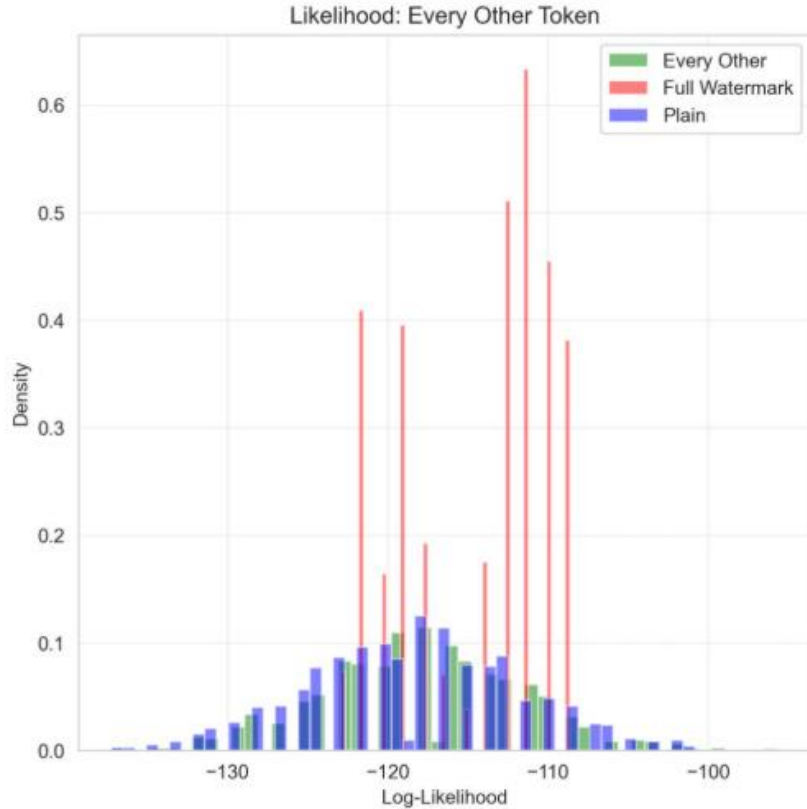
Findings:

The unwatermarked (plain) likelihood distribution is normal distribution as expected, while watermarked one is with many spikes, indicating limited amount of deterministic sequences. However, the likelihoods for watermarked one all fall under the plain distribution. Note that watermarked scheme randomly generates a length of m prefix, then apply Gumbel process on it. The plot uses $m=3$, therefore the available prefix sequences number is $4^3=64$, so in theory there will be at most 64 sequences.

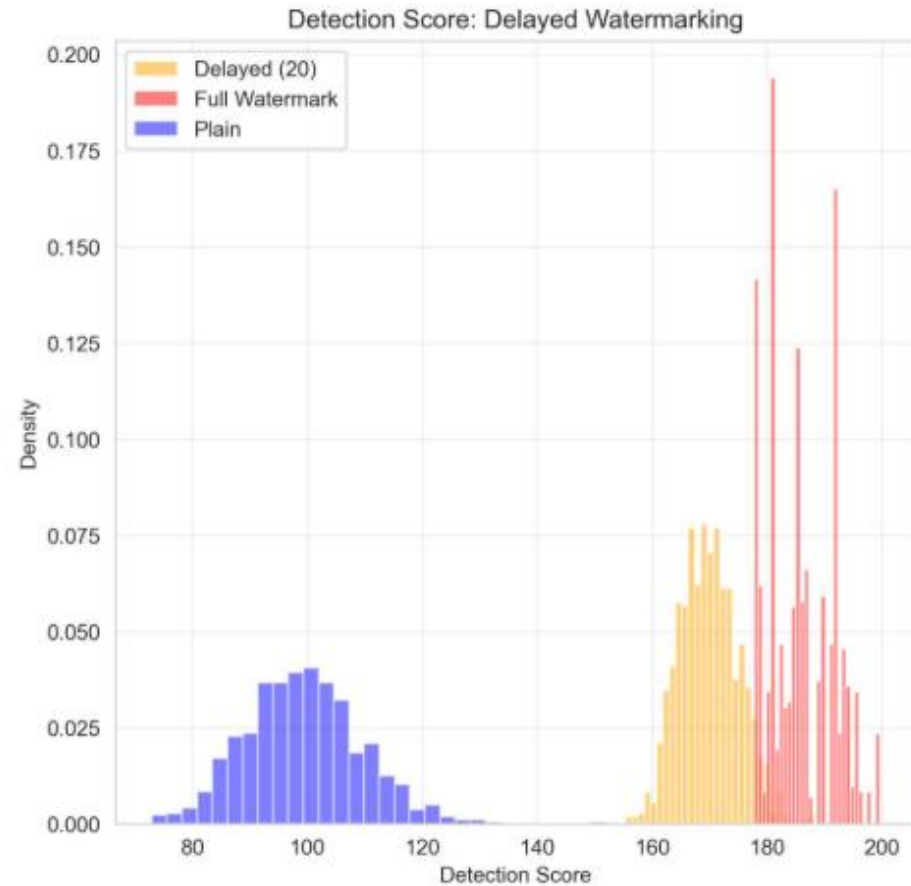
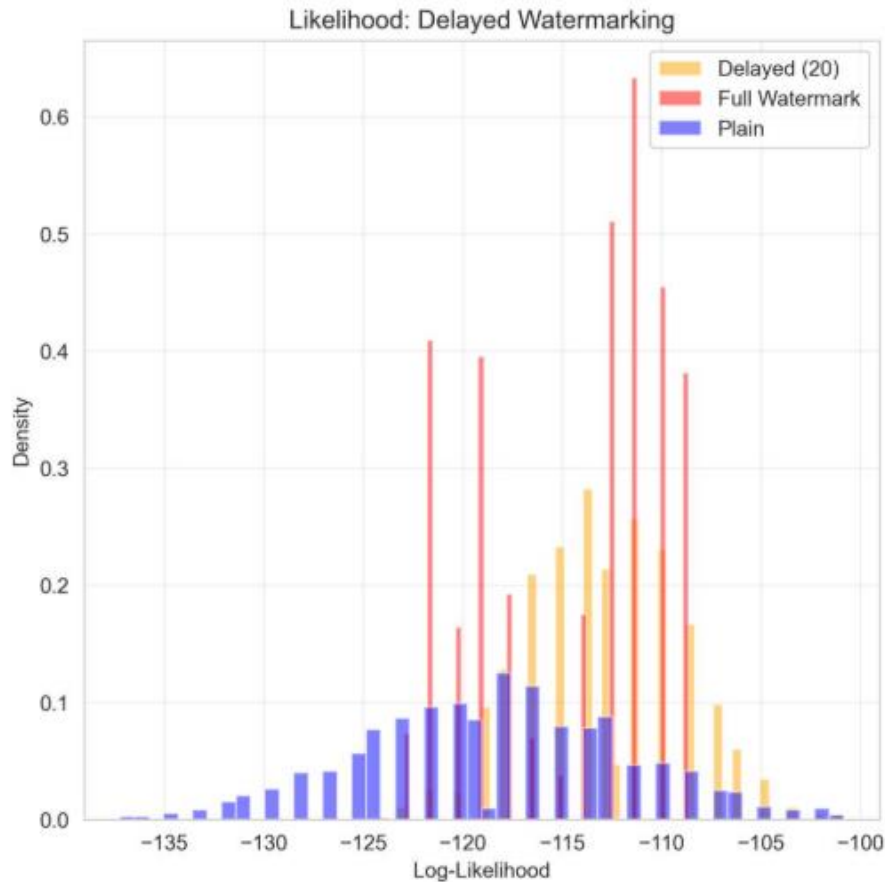
Excellent!

This is from my favorite submission

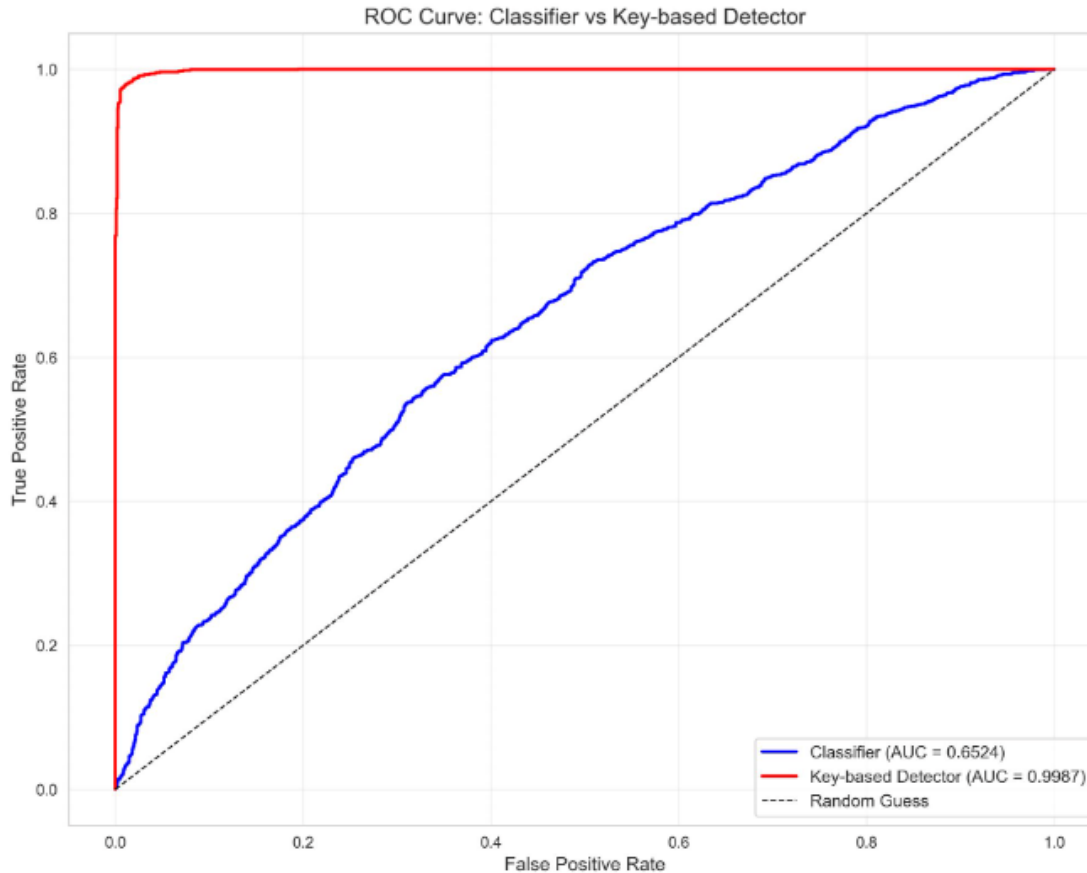
Watermark only every other token



Generate first 20 tokens with “true randomness” then the remainder fully watermarked



WM-Detector vs trained classifier



Experiment Design:

- Generated 5000 watermarked and 5000 plain sequences
- Trained an XGBoost classifier on sequence features
- Compared classifier performance to key-based detector

Features Used (fully generated by LLM):

- Token frequency distributions
- Transition bigram counts
- Statistical properties (variance, entropy, etc.)

Checkpoint: What do we learn by playing with Gumbel WM?

- No problem with detection with key.
- “Distortion-free” WM does change the distribution when we condition on the key.
- Longer “truly random” Burn-In and large prefix length m , help to make the sequences more diverse --- closer in distribution to the original unwatermarked model.

Today

- Discussing HW8
- Upgrading the Gumbel watermark into an Undetectable Watermark
- Student presentations:
 - Watermarking in the Sand by Ben TenWolde
 - “Watermarking, Provenance and Policy” by John Driscoll
- Discussion on the use of GenAI watermarking in practice

What's “even-better” than “distortion-free”?

- Sentence level distortion-free
 - (Kuditipudi et al, 2023): “Get multiple keys, rotate the keys being used. In detection time, test with all keys”
 - (Hu et al, 2023): “unique prefix each time within a sentence”
- Polynomially many sentence (*computational*) distortion-free
 1. Do the above two across many sentences.
 - 2. (Christ, Gunn, Zamir, 2023): “Accumulate sufficient amount entropy before adding watermark! ”

Key idea of the CGZ watermark

- First of all, CGZ is similar to Gumbel WM
 - They make the generated content secretly deterministic for those who observe the latent R.V.
 - The latent R.V. is determined by a keyed pseudo-random function (i.e., a hash-function) that takes a prefix as input.
- What's different?
 - The length of the prefix is longer and elastic (instead of fixed at m as in Gumbel and Green-Red)
 - CGZ does not add watermark unless the “entropy” is high enough.
- **Goal of these design:** ensure that the same input to the PRF does not appear more than once with high probability.

(A simplified) CGZ watermark pseudo-code

(From the exposition of Zamir:
<https://arxiv.org/abs/2401.10360>)

WLOG: consider vocabulary $\mathcal{V} = \{0,1\}$, let the latent-variables be $F_k(r, i)$

Algorithm 1 Watermarking algorithm Wat_k

Data: A prompt (PROMPT) and a secret key k

Result: Watermarked text x_1, \dots, x_L

$i \leftarrow 1$

$H \leftarrow 0$

while $\text{done} \notin (x_1, \dots, x_{i-1})$ **do**

$p_i \leftarrow \text{Model}(\text{PROMPT}, x_1, \dots, x_{i-1})$

if $H < \lambda$ **then**

 // Collect more internal entropy

 Sample $(x_i, p) \leftarrow (1, p_i)$ with probability p_i , otherwise $(0, 1 - p_i)$

$H \leftarrow H - \log p$ Accumulate empirical entropy, certify that that r is sufficiently rare.

if $H \geq \lambda$ **then**

$r \leftarrow (x_1, \dots, x_i)$

end

else

 // Embed the watermark

$x_i \leftarrow \mathbb{1}[F_k(r, i) \leq p_i]$

end

$i \leftarrow i + 1$

end

Algorithm 2 Detector Detect_k

Data: Text x_1, \dots, x_L and a secret key k

Result: True or False

for $i \in [L]$ **do** Search over unknown start time of WM

$r^{(i)} \leftarrow (x_1, \dots, x_i)$

 Define $v_j^{(i)} := x_j \cdot F_k(r^{(i)}, j) + (1 - x_j) \cdot (1 - F_k(r^{(i)}, j))$ for $j \in [L]$

if $\sum_{j=i+1}^L \ln(1/v_j^{(i)}) > (L - i) + \lambda\sqrt{L - i}$ **then**
 | **return** True

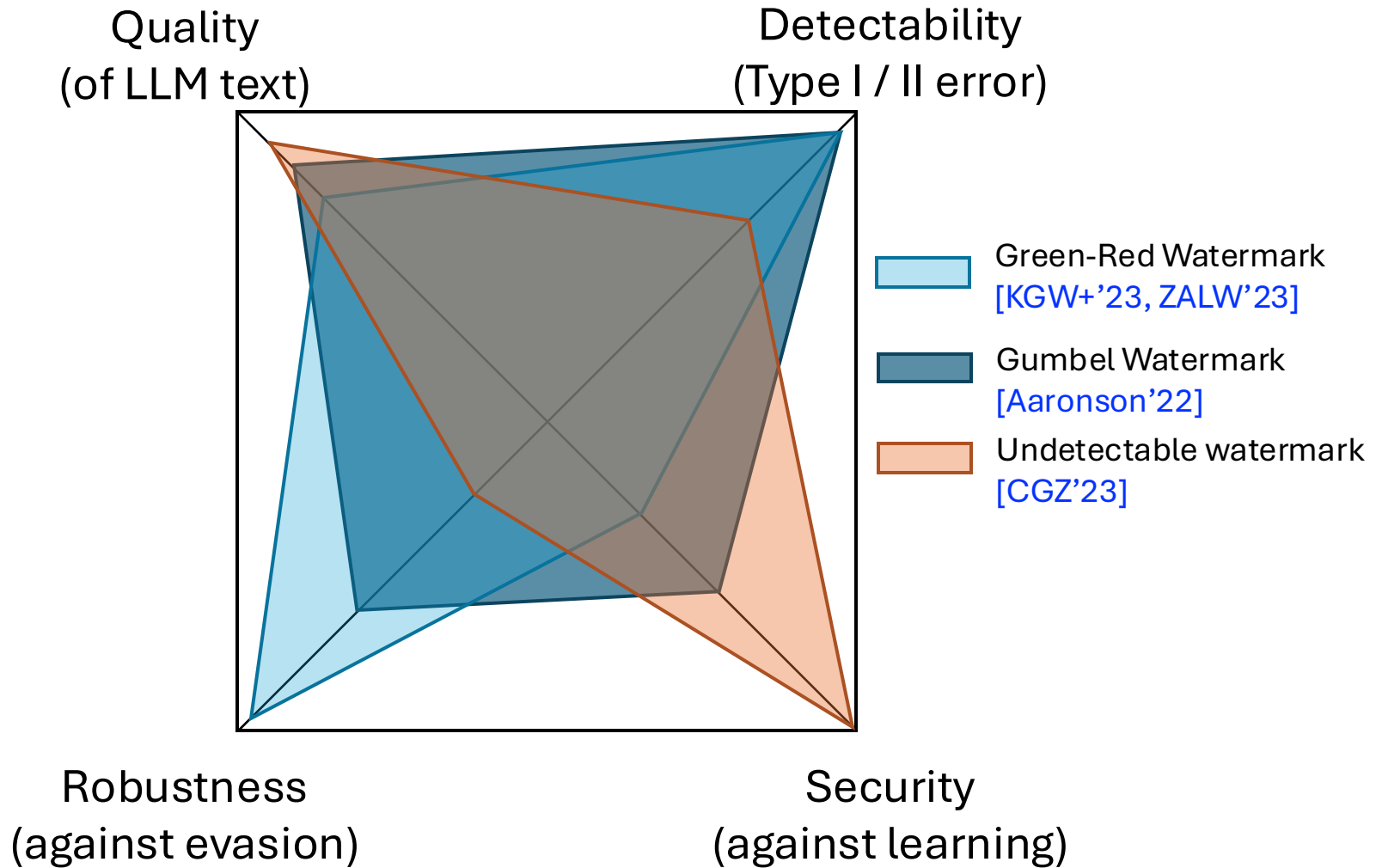
end

end

return False

What's the limitation of this approach?

- Robustness vs cropping and other forms of editing.



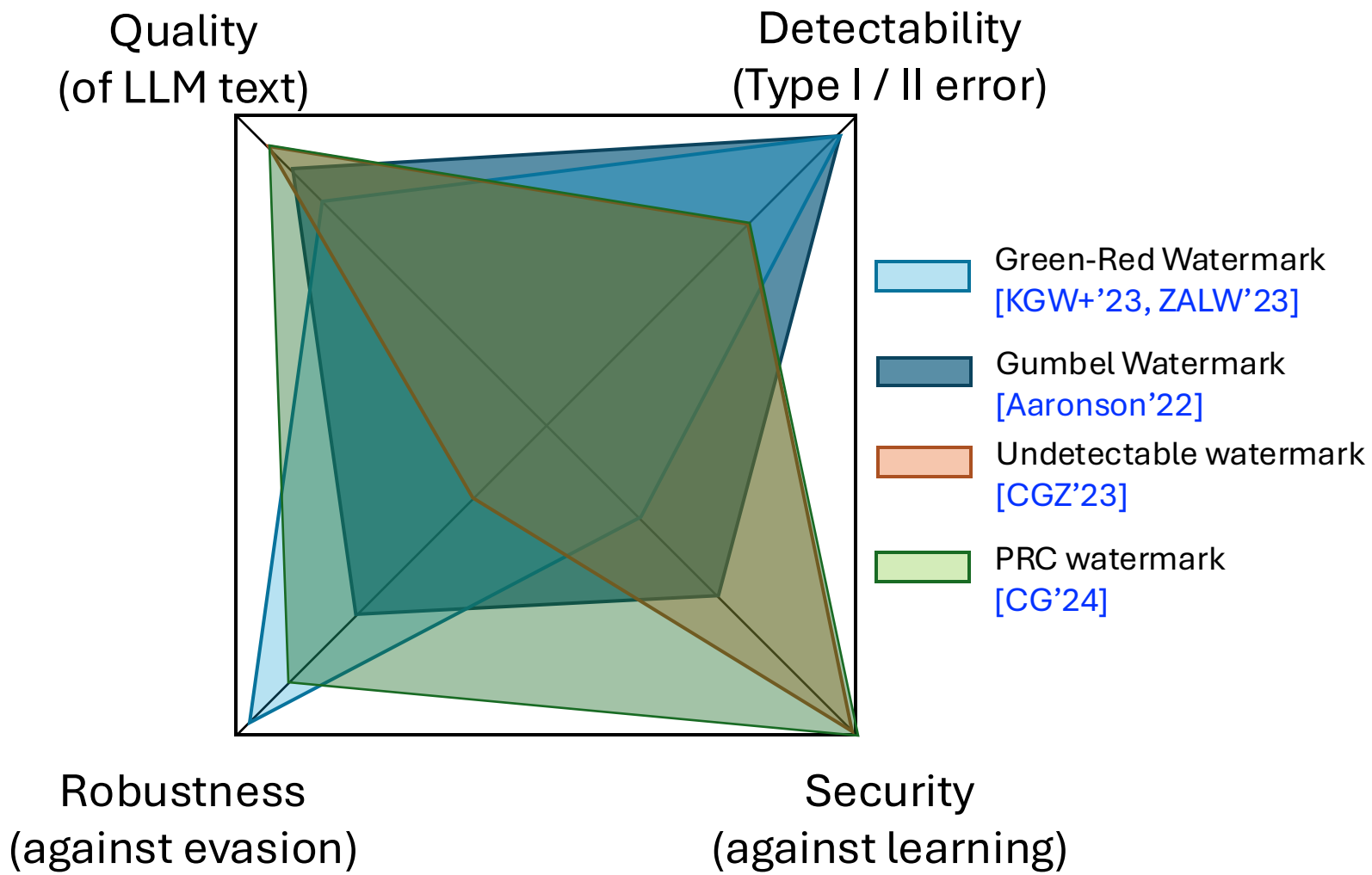
Can we have “undetectability” and “robustness to edits” at the same time?

- Watermark by Pseudo-Random Error Correcting Code (Christ and Gunn, 2024)
 - Key: [PRC, and L messages a_i with length m]

“ $W_1, \dots, W_m, \underbrace{W_{m+1}, \dots, W_{2m}}_{\text{Watermark with PRC.encode}(a_2)}, W_{2m+1}, \dots, W_{3m}, W_{3m+1}, \dots, W_{Lm}$ ”

Watermark with `PRC.encode(a_2)`

- Detector: check for **PRC.decode(all subsequences)** against all messages.
- In some sense getting the robustness of “Unigram watermark” and “Undetectability” of CGZ simultaneously.



References we discussed

1. Statistical watermarks

- Green-Red Watermark (Kirchenbauer et al, 2023)
- Unigram Green-Red watermark (Zhao, Ananth, Li, W. 2024)

2. Cryptographic watermarks

- Gumbel watermark. (Aaronson, 2022)
- Undetectable WM (Christ, Gunn, Zamir 2023)
- Distortion-Free WM (Kuditipudi et al, 2023)
- Unbiased WM (Hu et al ,2023)
- Pseudorandom Code WM (Christ and Gunn 2023)
- Permute-and-Flip WM (Zhao, Li, W., 2024)

No where near a complete set!

Topics we did not get to cover

- Multi-bit LLM watermark
 - Yoo, Ahn and Kwak (2023), Qu, Yin, He et al. (2024)
- Semantic text watermark
 - Liu, Pan, Hu et al (ICLR-2024). Liu and Bu (ICML-2024).
- Public verifiable watermark
 - Fairoze et al. (2023). Publicly detectable watermarking for language models.
- Fragile watermark (deliberately non-robust for attribution/verification)
 - Jiang, Zhengyuan, et al. "Watermark-based Detection and Attribution of AI-Generated Content." arXiv preprint arXiv:2404.04254 (2024).
- Impossibility results
 - "Zhao et al (2023) "Invisible Image Watermarks..." Zhang, Barak et al. (2024) Watermarks in the Sand . Also work by Soheil Feizi et al and Furong Huang et al.

Today

- Discussing HW8
- Upgrading the Gumbel watermark into an Undetectable Watermark
- Student presentations:
 - Watermarking in the Sand by Ben TenWolde
 - “Watermarking, Provenance and Policy” by John Driscoll
- Discussion on the use of GenAI watermarking in practice

“Watermarking in the Sand” by Benjamin TenWolde

arXiv > cs > arXiv:2311.04378

Search...

Help | A

Computer Science > Machine Learning

[Submitted on 7 Nov 2023 (v1), last revised 27 May 2025 (this version, v5)]

Watermarks in the Sand: Impossibility of Strong Watermarking for Generative Models

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, Boaz Barak

Watermarking generative models consists of planting a statistical signal (watermark) in a model's output so that it can be later verified that the output was generated by the given model. A strong watermarking scheme satisfies the property that a computationally bounded attacker cannot erase the watermark without causing significant quality degradation. In this paper, we study the (im)possibility of strong watermarking schemes. We prove that, under well-specified and natural assumptions, strong watermarking is impossible to achieve. This holds even in the private detection algorithm setting, where the watermark insertion and detection algorithms share a secret key, unknown to the attacker. To prove this result, we introduce a generic efficient watermark attack; the attacker is not required to know the private key of the scheme or even which scheme is used. Our attack is based on two assumptions: (1) The attacker has access to a "quality oracle" that can evaluate whether a candidate output is a high-quality response to a prompt, and (2) The attacker has access to a "perturbation oracle" which can modify an output with a nontrivial probability of maintaining quality, and which induces an efficiently mixing random walk on high-quality outputs. We argue that both assumptions can be satisfied in practice by an attacker with weaker computational capabilities than the watermarked model itself, to which the attacker has only black-box access. Furthermore, our assumptions will likely only be easier to satisfy over time as models grow in capabilities and modalities. We demonstrate the feasibility of our attack by instantiating it to attack three existing watermarking schemes for large language models: Kirchenbauer et al. (2023), Kudityudi et al. (2023), and Zhao et al. (2023). The same attack successfully removes the watermarks planted by all three schemes, with only minor quality degradation.

“What Lies ahead for Generative AI Watermarking” by John Driscoll

What Lies Ahead for Generative AI Watermarking

Pierre Fernandez¹ Anthony Level² Teddy Furon^{† 1}

Abstract

This position paper discusses the potential of watermarking as a means to improve transparency and traceability in AI-generated content. Although robustness is often highlighted as a major technical challenge, watermarking has undeniable advantages over other content provenance methods, such as forensics or fingerprinting, making it inevitable. However, more significant unanswered questions remain, such as how to use and trust the detection outcomes and how to ensure interoperability between actors. We should prioritize finding both technical and regulatory answers to these questions – currently scarce in the public discourse – rather than focusing on robustness, which is not truly problematic.

and traceability (USA, 2023; Chi, 2023; Eur, 2023)¹. For instance, the Californian act on watermarking would require model providers to “*place imperceptible and maximally indelible watermarks containing provenance data into synthetic content*” (California State Leg., 2024). Several key players, like Google, Meta, and OpenAI, have already started applying it at scale.

For many, the primary concern is robustness: malicious actors might attempt to remove the watermark. This issue is frequently cited as the main barrier to implementing watermarking for detection purposes (Christodorescu et al., 2024; Knibbs, 2023; Harris & Norden, 2024). Improving robustness is challenging, but it should not be a case of not seeing the forest for the trees.

In section 2, we first give a brief overview of content provenance methods and of watermarking. In section 3, we highlight its technical strengths, such as low, controllable false

Today

- Discussing HW8
- Upgrading the Gumbel watermark into an Undetectable Watermark
- Student presentations:
 - Watermarking in the Sand by Ben TenWolde
 - “Watermarking, Provenance and Policy” by John Driscoll
- Discussion on the use of GenAI watermarking in practice

How applicable is watermarking?

Why LLM watermarking will never work



David Gilbertson

Follow

20 min read · Nov 14, 2024



600



11



<https://ai.gopubby.com/why-llm-watermarking-will-never-work-1b76bdeebbd1>

“watermarking can help identify text generated by an LLM? ”

“watermarking can help reduce harms like AI-generated misinformation”

The author provided concrete reasons why these claims are difficult to materialize.

Main argument: Watermarking only detect specific watermarked models, but not AI-generated content generically, thus not useful when attackers have open-source alternatives.

	LLMs behind an API			Open source	
One watermarked model	Gemini	ChatGPT	Claude	Llama	Human
Watermarking best case	Gemini	ChatGPT	Claude	Llama	Human
To detect AI-generated text	Gemini	ChatGPT	Claude	Llama	Human

More to watermarking research

- Watermarking models
 - against distillation (model-stealing) attacks
 - e.g., paper: [Protecting Language Generation Models via Invisible Watermarking](#)
- Watermarking Open-Source Models
 - detectable even after finetuning
- Watermarking training data
 - training somehow retains watermarks
- Watermarking non-AI content (rather than AI-generated content).
 - Need to be **fragile** and tamper-resistant

(Again) remainder of the course

- This is the last lecture.
- Next Thursday: Quiz.
 - Right here in class, start at 9:30 am.
 - I will try to provide a few sample problems over the weekend.
- December 8: Mini-symposium on Gen AI Safety
 - When: 12:00 pm to 5:00 pm. (Pizza will be provided)
 - Where: EnCORE space at Atkinson 4th floor

It's a pleasure working with you all

- I tried quite a few new things (e.g., in-class student presentations, AI-assisted coding for the first time)
 - Would appreciate your feedback!
 - **Course evaluation:** 1% bonus for everyone if we get more than 60% student filling in the course eval form.
- I hope you guys learned something useful from this course.
- Good luck with the quiz and your projects!