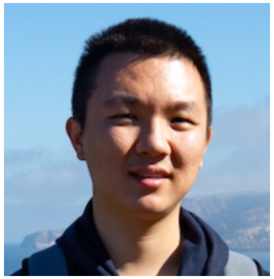
Advances in Watermarking Large Language Models

Yu-Xiang Wang

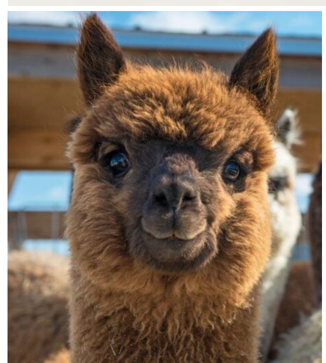
COMPUTER SCIENCE
UC SANTA BARBARA
Computing. ReInvented.

Based on joint work with Xuandong Zhao, Lei Li, Prabhanjan Ananth



ChatGPT and other Large Language Models

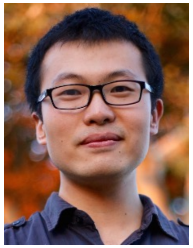
Llama



Alpaca, Vicuna

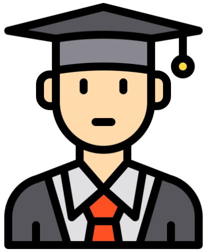
RedPajama

LLMs are very useful…



Could you generate HW2 for my machine learning class 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: XXXXXXXXXX….

LLM can be used for malicious purposes, e.g., fake news, frauds, scams…

REUTERS® World Business Markets Sustainability Legal Breakingviews Technology Investig

Disrupted

China reports first arrest over fake news generated by ChatGPT

Reuters

May 10, 2023 1:41 PM PDT · Updated a month ago

The logo of OpenAI is displayed near a response by its AI chatbot ChatGPT on its website, in this illustration picture taken February 9, 2023. REUTERS/Florence Lo/Illustration/File Photo

AI makes plagiarism harder to detect, argue academics – in paper written by chatbot

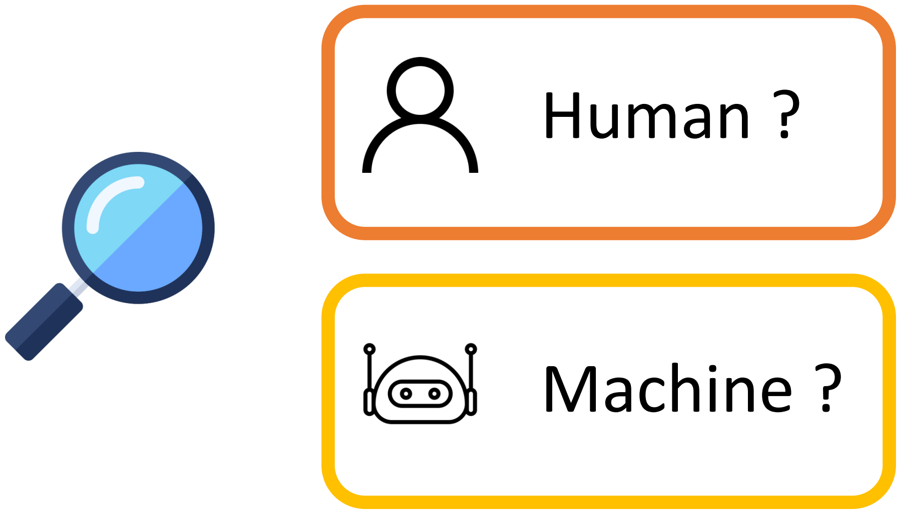
Lecturers say programs capable of writing competent student coursework threaten academic integrity

Bristol University is among the institutions to have issued new guidance on how to detect the use of ChatGPT. Photograph: Adrian Sherratt/Alamy

What do we do?

Possible solution: Can you distinguish between human and AI-generated text?

The Information Theory and Applications (ITA) Workshop is a captivating and vibrant gathering that brings together some of the brightest minds in the world of information theory. This dynamic workshop serves as a melting pot for pioneering ideas, where experts and enthusiasts from various disciplines converge to explore the latest advancements in information theory and its myriad applications. From groundbreaking research presentations to thought-provoking discussions, ITA is not just a conference; it's a celebration of knowledge and innovation.

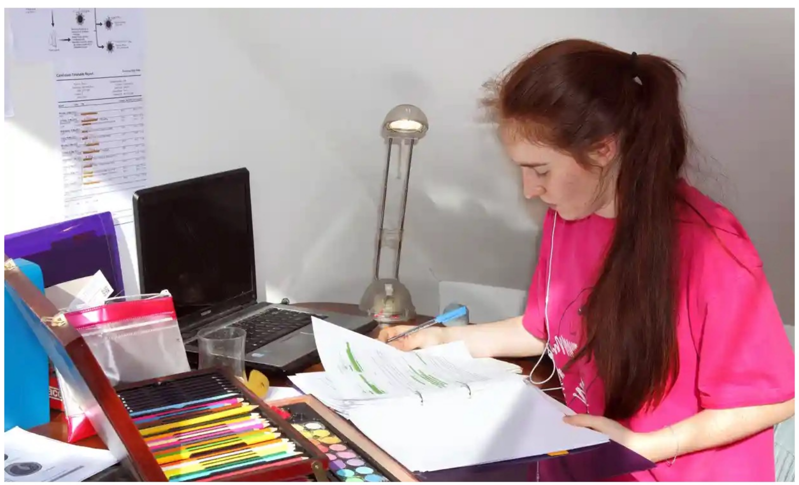


Train a machine learning model to solve Turing test?

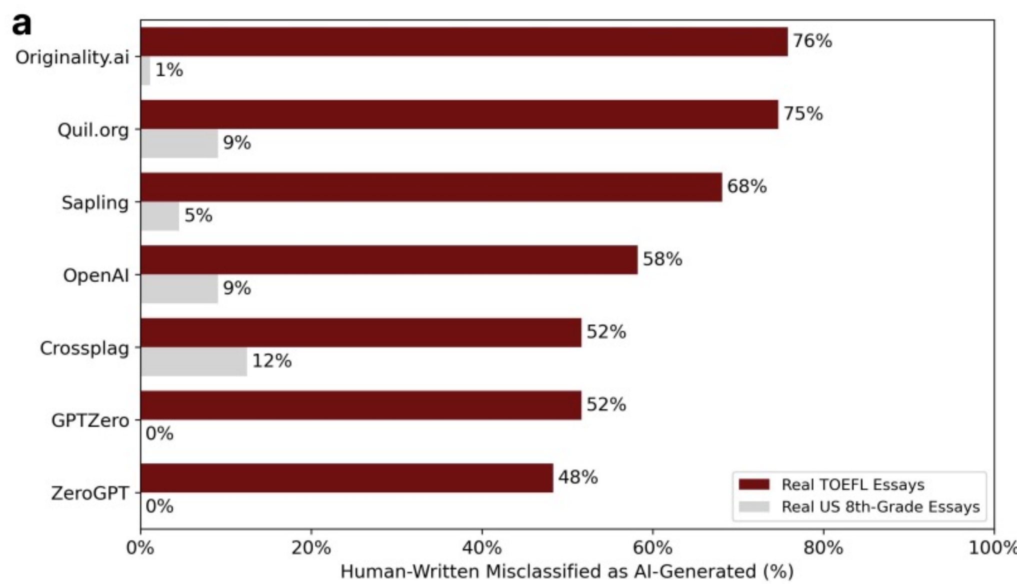
Training classifiers to detect AI-generated text is bound to fail!

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



AI detectors could falsely flag college and job applications and exam essays as GPT-generated,



Liang et al. 2023: https://arxiv.org/abs/2304.02819

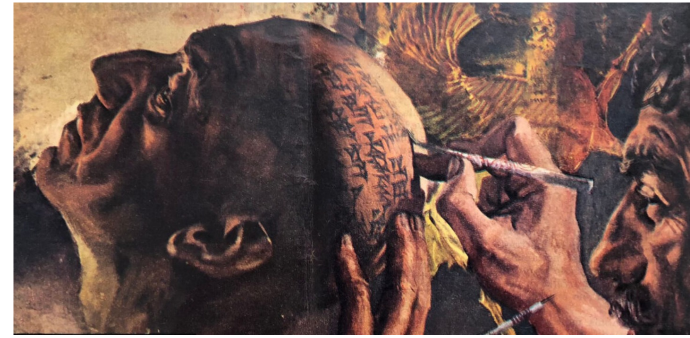
Better solution: “watermark” the generated text…

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,

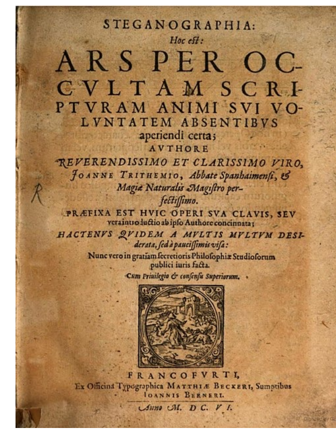
Llamas graze on hillsides demure,
Harmony found in their gentle stride,
Amidst the mountains where they reside,
Mystical creatures with wisdom inside,
A journey with them is an incredible ride.

Watermarking has a long history.



(Herodotus, 499 BC)

- Ancient Greece: Steganography
- 1499: Trithemius “Steganographia”



STEGANOGRAPHIA:
Hoc est:
ARS PER OC-
CVLTAM SCRI-
PTVRAM ANIMI SVI VO-
LVNTATEM ABSENTIBVS
aperiendi certa;
AVTHORE
REVERENDISSIMO ET CLARISSIMO VIRO,
JOANNE TRITHEMIO, Abbate Spanhaimensi, &
Magiæ Naturalis Magistro per-
fectissimo.
PRÆFIXA EST HVIC OPERI SVA CLAVIS, SEV
vera introductio ab ipso Authore concinnata;
HACTENVS QVIDEM A MVLTIS MVLTVM DESI-
derata, sed à paucissimis visa:
Nunc vero in gratiam secretioris Philosophiæ Studiosorum
publici iuris facta.
Cum Priuilegio & consensu Superiorum.
FRANCOFVRTI,
Ex Officina Typographica MATTHIÆ BECKERI, Sumptibus
IOANNIS BERNERI.
Anno M. DC. VI.

(Trithemius, 1499)

- Modern research:
  - 1950s: Embedding code to music (Hembrooke, 1954)
  - 1990s to 2000s: Digital Watermarks (e.g., Ingemar J. Cox, Matt Miller, etc..)

Mostly about “IP protection”, “Authentication”
Mostly about images. Some interesting theoretical results.

2022+: Recent Renaissance due to the rise of Generative AI

- Watermarking LLM text
  - Aaronson (2022), Kirchenbauer et al (2023), **Zhao et al. (2023;2024),** Christ et al (2023), Kuditipudi et al. (2023)
- Watermarking LLM models
  - **Zhao et al (2022)** "Distillation resistant watermarking" **Zhao et al (2023)** **"**Protecting Language Generation Models via Invisible Watermarking**"**
- Watermarking Images (e.g. from Diffusion models)
  - (e.g., Fernandez et al 2023 "Stable signature", Wen et al. 2023 "Tree-Ring Watermarks")
- "Is strong watermarking possible?"
  - "**Zhao et al (2023**) "Invisible Image Watermarks Are Provably Removable Using Generative AI"
  - Zhang, Barak et al. (2024) Watermarks in the Sand: Impossibility of Strong Watermarking for Generative Models
  - Also work by Soheil Feizi et al.

Slightly different settings, motivating applications and new challenges.

LLM text watermarking schemes

- Statistical watermarks
  - Green-Red watermark (Kirchenbauer et al., 2023)
  - Unigram (Green-Red) WM (Zhao, Ananth, Li, W., 2023)
- Cryptographic watermarks
  - Gumbel watermark (Aaronson, 2022)
  - Undetectable WM (Christ, Gunn, Zamir 2023)
  - Permute-and-Flip WM (Zhao, Li, W., 2024)
- Quite a few others in this fast-growing research area

Outline of the talk

- Formally defining LLM watermarks
- Two recent work
  - Provable Robust Watermark of LLMs
  - Permute-and-Flip decoding and watermarking
- Open problems

What is a Language Model?

P(next word y_t | Prompt x, 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 ____

beer: 0.5

gun: 0.3

hand: 0.1

.

.

blood-pressure: 0.001

The **universe of words** is called a **vocabulary** $\boldsymbol{V}$

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

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

Example: Green-Red Watermark
(Kirchenbauer et al. 2023; Zhao et al. 2023)

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

“Hold my ____

beer : 0.52

gun : 0.29

hand : 0.11

.
.
.

blood-pressure : 0.001

$\hat{\mathcal{M}}$: Modified LM

Key: Green lists

Detection: Count # of Greens

Increase the probability of green tokens slightly,

Decrease the probability of red tokens slightly.

Example of the watermark detector

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...]

LLaMA-7B
(watermarked)
PPL: 9.47
z-score: 9.58

**Confidence score 0.99999999999...*
(p-value < 10e-15)

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”
- Robustness
  - Must be robust to all kinds of evasion attacks

Quality of LLM generated text

- **Low-distortion:** distributions of the generated text by $\mathcal{M}$ and $\widehat{\mathcal{M}}$ are close
  - Which metric to use? TV, KL-div, Renyi?
  - Which distribution? One-token / whole sequence / any polynomial number of sequences
  - (ex post vs ex ante) when $\widehat{\mathcal{M}}$ is random, is the quality guarantee for every realized $\widehat{\mathcal{M}}$ or over the distribution of $\widehat{\mathcal{M}}$
- **High quality:** The generated text by $\widehat{\mathcal{M}}$ should be high
  - E.g., perplexity and other metrics.

Provable theoretical results on quality of the WM

	Single token	Whole sequence	Many sequences
ex ante	Aaronson	Kuditipudi et al	Christ et al
ex post	Zhao et al	Zhao et al (through composition)	?

A hypothesis testing view of LLM watermarks' detection properties

- **H_0:** The suspect text y is NOT generated from $\widehat{\mathcal{M}}$
  - e.g., "y" is written by a human.
  - e.g., "y" is generated by $\mathcal{M}$.
- **H_1:** The suspect text is generated from $\widehat{\mathcal{M}}$

A very broad "Null" and a very specific "Alternative "

- **Metrics:** Type I / II Err. Power at FPR α. F1-score.
- **Theory**: Can we control FPR. Can we prove high power? Are the tradeoff optimal?

Not all LLM generated text are easily watermarkable.

Write a blog article with my rant the broken peer-review system!

Don’t get me started with Reviewer #2. I’d rather have GPT4 reviewing my paper ….

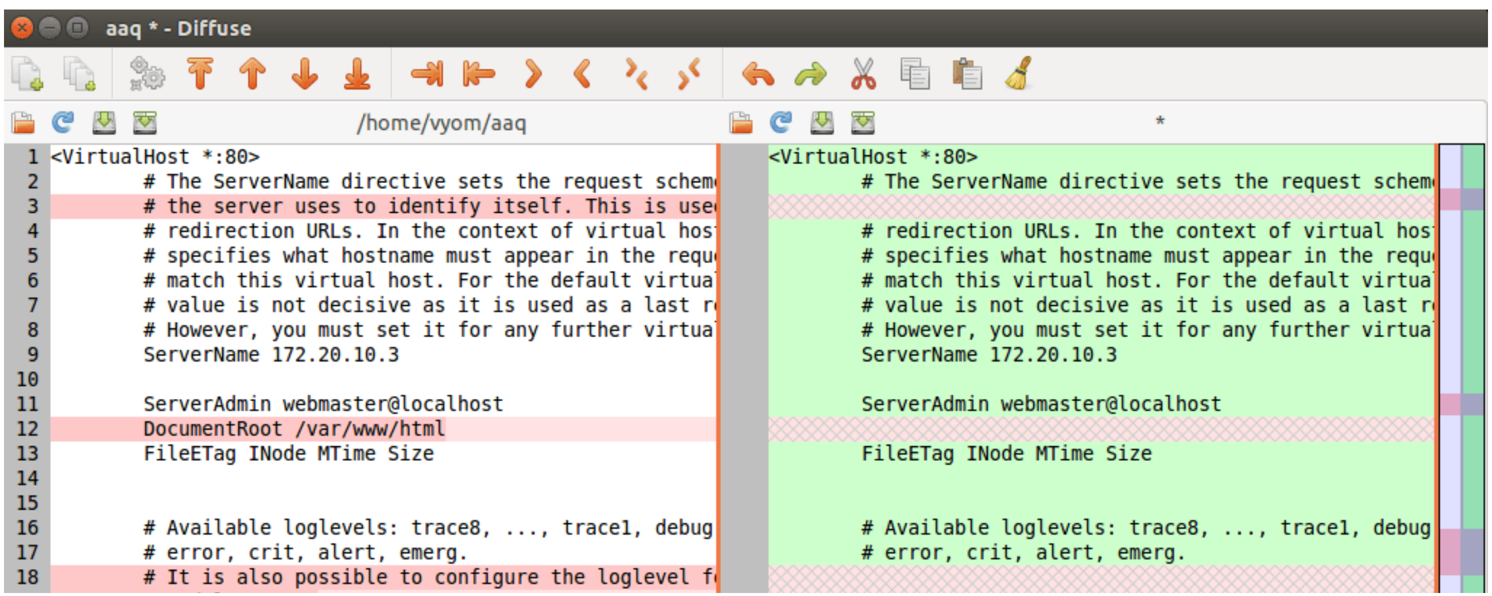
Repeat “Goal!” for 500 times like a football commentator

Goal! Goal! Goal! Goal! …

Which example is more easily watermarkable / detectable?

Robustness is needed even if no explicit evasion attack. People won’t use the generated text verbatim!

- Cropping
- Shuffling: Move thing around
- Edits / improving



Formally defining robustness

- Need to specify a family of possible attacks
  - e.g. parameterized by the Edit Distance allowed

Don’t get me started with Reviewer #2. I’d rather have GPT4 reviewing my paper ….

Hmmm.. Let me edit it before posting the blog.

- Quantify how much drop in “Power” or increase in Type II error
  - E.g., as a function of the Edit distance.

Outline of the talk

- Formally defining LLM watermarks
- Two recent work
  - Provable Robust Watermark of LLMs
  - Permute-and-Flip decoding and watermarking
- Open problems

TL;DR of our contributions in [Zhao, Ananth, Li, W. 2023]

1. Theoretical framework for LLM Watermarks

2. Theoretical guarantees of Kirchenbauer et al's Green-Red watermark
   - Quality, Detection accuracy, Robustness

3. Simplest (Unigram) variant of the green/red WM has the most robustness --- and it works!

Provable Robust Watermarking for AI-Generated Text
Xuandong Zhao, Prabhanjan Ananth, Lei Li, Yu-Xiang Wang.
ICLR 2024 https://arxiv.org/abs/2306.17439

Green-Red Watermark, revisited

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

"Hold my ____

beer: 0.52

gun: 0.29

hand: 0.11

.
.

blood-pressure: 0.001

Increase the probability of green tokens slightly,
Decrease the probability of red tokens slightly.

$$\mathcal{M}: \;\; y_t \sim \text{Softmax}(\,\text{logits}(\text{Prompt}, y_{<t}))$$

$$\widehat{\mathcal{M}}: \;\; y_t \sim \text{Softmax}(\,\text{logits}(\text{Prompt}, y_{<t}) + \delta \cdot \mathbf{1}(\cdot\ is\ green))$$

How is the Green list generated?

- ***Randomly* selecting** γ fraction of the vocabulary.

  - (Kirchenbauer et al.): Different green list at each time t as function of the prefix with length (m-1).

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

m-Gram with m = 4

  - (Zhao et al.): Use m = 1, i.e., a consistent "Green list".

Detection of Green/Red WM

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

(Optional pre-processing) $\boldsymbol{y}$ ← unique($\boldsymbol{y}$)

1. Computer z-score

$$z_{\boldsymbol{y}} = \left(|\boldsymbol{y}|_G - \gamma n\right) / \sqrt{n\gamma(1-\gamma)}.$$

2. **if** $z_{\boldsymbol{y}} > \tau$ **then**

   Return 1: "$\boldsymbol{y}$ is watermarked"

   else:

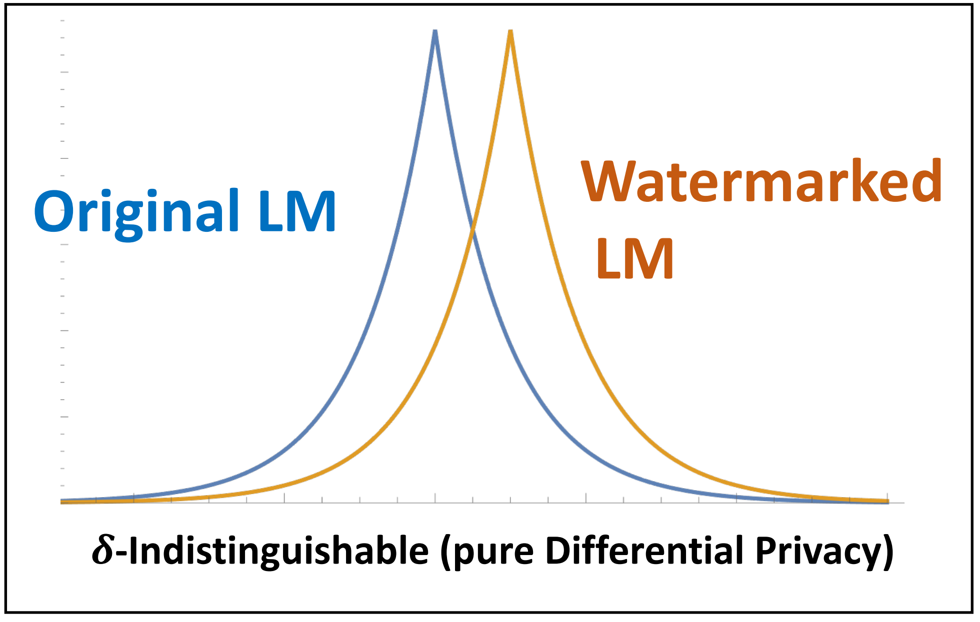
   Return 0: "No conclusive evidence"

Theoretical Guarantee for Unigram-Green/Red Watermarks

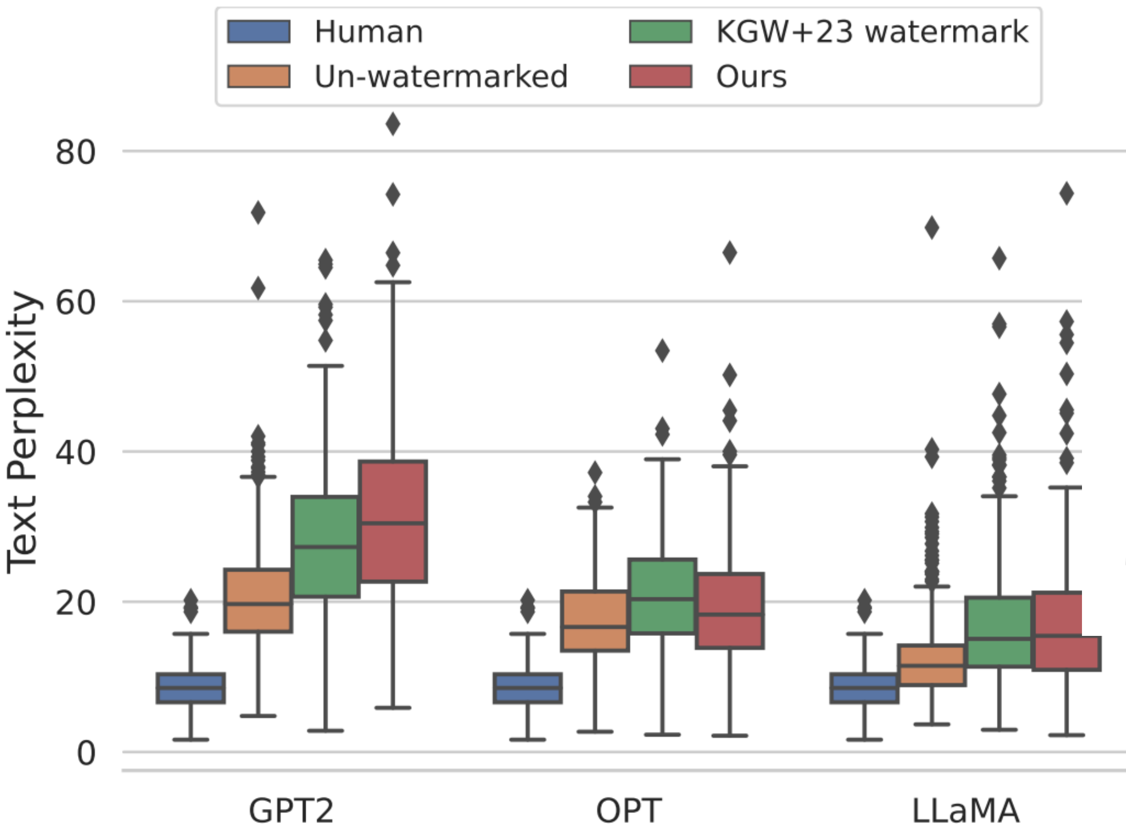
- Quality guarantees:
  - Watermarked LLM and Original LLM are indistinguishable.
- Detection guarantees
  - Type I error --> 0 exponentially as n gets larger.
  - Type II error --> 0 exponentially as n gets larger (under natural technical conditions)
- Provably Robust to Edits --- Twice as robust as an popular baseline (Kirchenbauer et al. 2023)

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.

	Avg Score	STD
Un-watermarked	3.660	0.655
Watermarked	3.665	0.619

Table 3: Human evaluation result.

Detection guarantees

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

$$\boldsymbol{z_y = O(\sqrt{\log(1/\alpha)})} \textbf{ w.p. } \boldsymbol{1-\alpha}$$

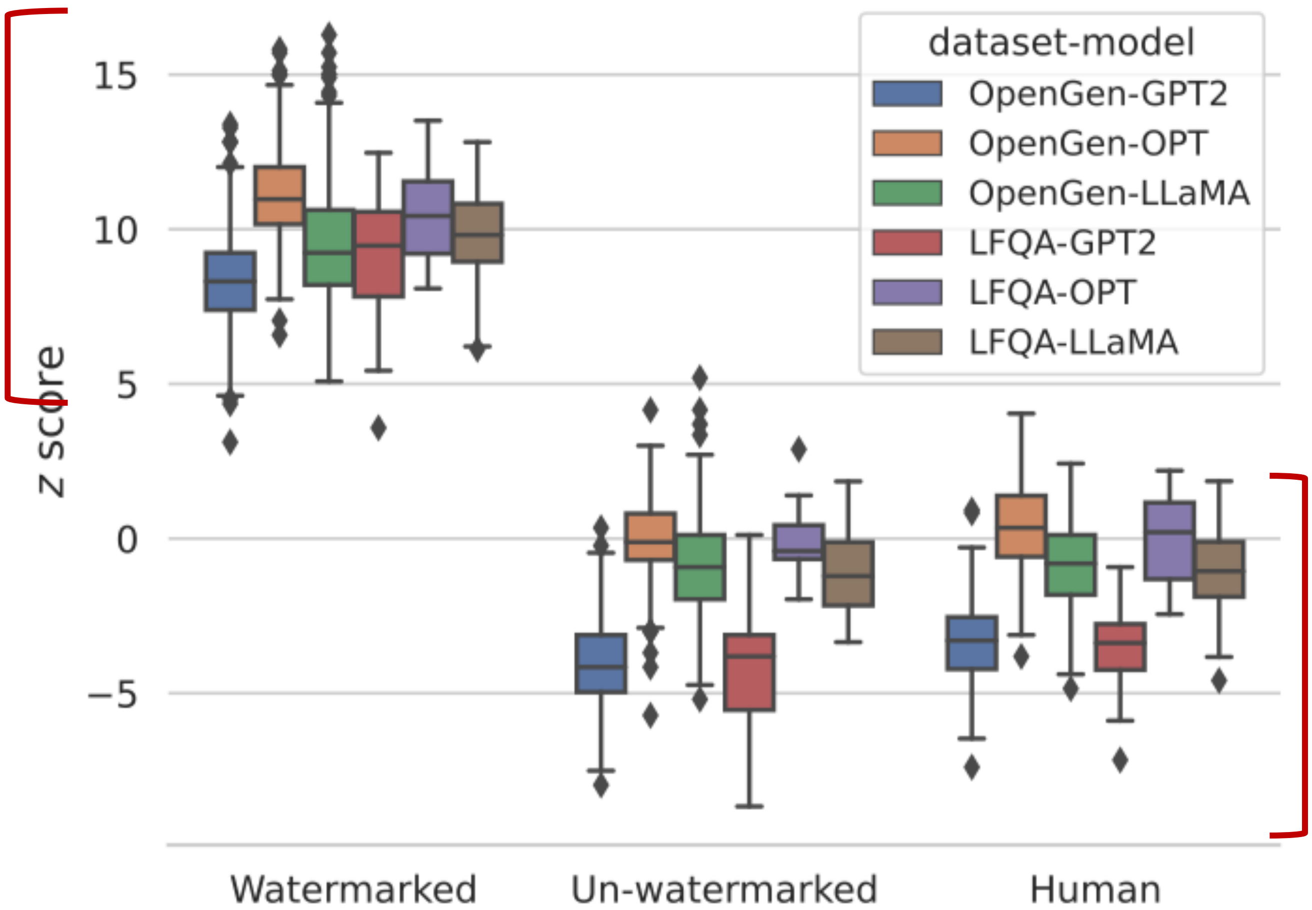
where V and C_{max} measure the **diversity** of the text.
If unique, then Z=1 and Cmax = 1

Theorem (informal): Let the suspect text $\boldsymbol{y}$ be generated using our watermarked LM. Assume n is $n = \widetilde{\Omega}(\log(1/\beta)/\delta^2)$ LM satisfy a *"Entropy condition"* and *"Homophily"*, then

$$\boldsymbol{z_y = \Omega\left(\kappa\,(e^{\delta}-1)\sqrt{n}\right)\ w.p.\ 1-\beta}$$

Our detection guarantees Illustrated

$z_y \gtrsim (e^\delta - 1)\sqrt{n}$



$z_y \lesssim O(\log(1/\alpha))$

H0: "Null"

H1: Alternative

Our watermark is robust to edits!

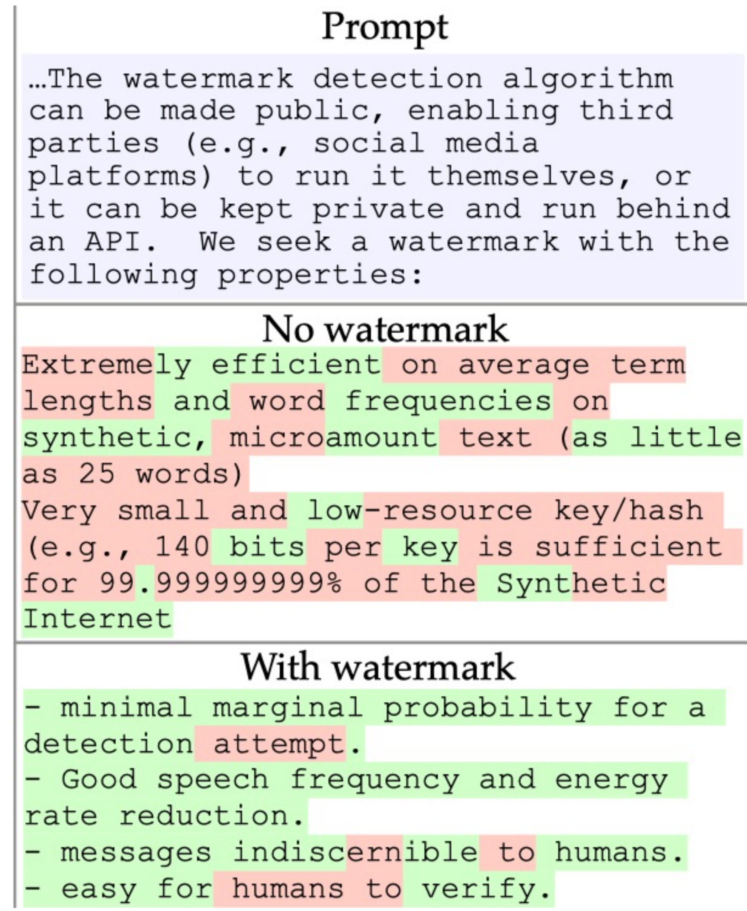
Theorem: Adversary take watermarked output $\boldsymbol{y}$, Adversary edits to get to a new text $\boldsymbol{u}$. If Edit Distance $ED(y, u) \leq \eta$, then

$$z_{\boldsymbol{u}} \geq z_{\boldsymbol{y}} - \max\{\frac{(1+\gamma/2)\eta}{\sqrt{n}}, \frac{(1-\gamma/2)\eta}{\sqrt{n-\eta}}\}.$$

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.



Why "Unigram" watermark --- among the family of "m-gram" watermarks?

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

- We find it most practical to use m=1.
  - **Robustness to edits: margin / m**

Experiment

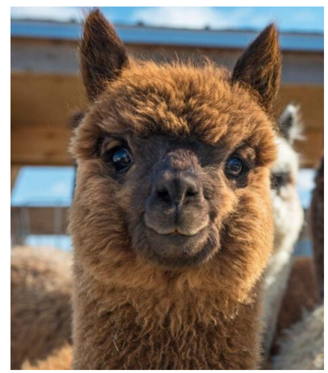
- Two long-form text datasets
  - **OpenGen**: 3K chunks sampled from the validation split of WikiText-103
  - **LFQA**: long-form question-answering dataset from Reddit

- 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]

Paraphrasing attack

Watermarked LLM

"Generated Text with watermark"



Prompt: "Please paraphrase!"

"New text generated with similar quality"

Robustness against paraphrasing attack

Setting	Method	OpenGen				LFQA			
		1% FPR		10% FPR		1% FPR		10% FPR	
		TPR	F1	TPR	F1	TPR	F1	TPR	F1
No attack	KGW+23	1.000	0.995	1.000	0.952	1.000	0.995	1.000	0.952
	UNIGRAM-WATERMARK	1.000	0.995	1.000	0.952	1.000	0.995	1.000	0.952
ChatGPT	KGW+23	0.565	0.704	0.853	0.747	0.327	0.453	0.673	0.490
	UNIGRAM-WATERMARK	0.866	0.910	0.961	0.818	0.442	0.568	0.865	0.584
DIPPER-1	KGW+23	0.386	0.546	0.738	0.720	0.372	0.534	0.740	0.767
	UNIGRAM-WATERMARK	0.729	0.830	0.922	0.837	0.639	0.770	0.909	0.865
DIPPER-2	KGW+23	0.490	0.646	0.810	0.769	0.432	0.595	0.845	0.839
	UNIGRAM-WATERMARK	0.777	0.862	0.941	0.852	0.693	0.810	0.948	0.894
BART	KGW+23	0.342	0.505	0.667	0.759	0.457	0.617	0.783	0.836
	UNIGRAM-WATERMARK	0.590	0.730	0.861	0.857	0.656	0.784	0.885	0.897

Outline of the talk

- Formally defining LLM watermarks

- Two recent work
  - Provable Robust Watermark of LLMs
  - Permute-and-Flip decoding and watermarking

- Open problems

All existing watermarks work with the standard decoder: softmax(logits)

$$\textbf{Softmax sampling:}\ y_t \sim p(y) = \frac{e^{u(y|x, y_{1:t-1})/T}}{\sum_{\tilde{y}} e^{u(\tilde{y}|x, y_{1:t-1})/T}}$$

- Temperature parameter T:
  - Large T ⇔ higher text entropy (more watermarkable)
  - Small T ⇔ higher text quality (smaller perplexity).

1. Is softmax(logits) the optimal choice?

2. Can we benefit from **co-designing** the decoder and watermarking scheme?

TL;DR of our results in [Zhao, Li, W. 2024]

1. We propose "Permute-and-Flip Decoding"
   - PF dominates Softmax in robustness-perplexity tradeoff.

2. A cryptographic watermark for Permute-and-Flip
   - Enjoys all nice properties of the Gumbel watermark
   - Slightly better detectability-perplexity tradeoff

Permute-And-Flip: An Optimally Robust and Watermarkable Decoder for LLMs
Xuandong Zhao, Lei Li, Yu-Xiang Wang.
Technical report: https://arxiv.org/abs/2402.05864

Permute-and-Flip Sampling from Differential Privacy literature (McKenna and Sheldon, 2021)

Algorithm 1 Permute and Flip (PF) Decoding

Input: prompt x, language model $\mathcal{M}$, temperature T.
for $t = 1, 2, \cdots$ **do**
  Logits $u_t \leftarrow \mathcal{M}([x, y_{1:t-1}])$.
  Find $u_t^* \leftarrow \max_{y \in \mathcal{V}} u_t(y)$.
  $\boxed{\textsf{Permute} : \text{Shuffle the vocabulary } \mathcal{V} \text{ into } \tilde{\mathcal{V}}.}$ **Permute**
  for $y \in \mathcal{V}$ **do**
    $\boxed{\textsf{Flip} : \text{Draw } Z \sim \text{Bernoulli}\left(\exp\left(\frac{u_t(y) - u_t^*}{T}\right)\right)}$. **Flip**
    if $Z = 1$, **then** assign $y_t \leftarrow y$ and **break**.
  end for
end for
Output: Generated sequence $y = [y_1, ..., y_n]$.

Permute-and-Flip(logits) is very similar to Softmax(logits)

Rejection sampling form of Softmax sampling

1. Uniformly samples $y \in \mathcal{V}$,
2. Return it with probability

$$p(y)/p(y^*) = \exp\left((u_t(y) - u_t(y*))/T\right).$$

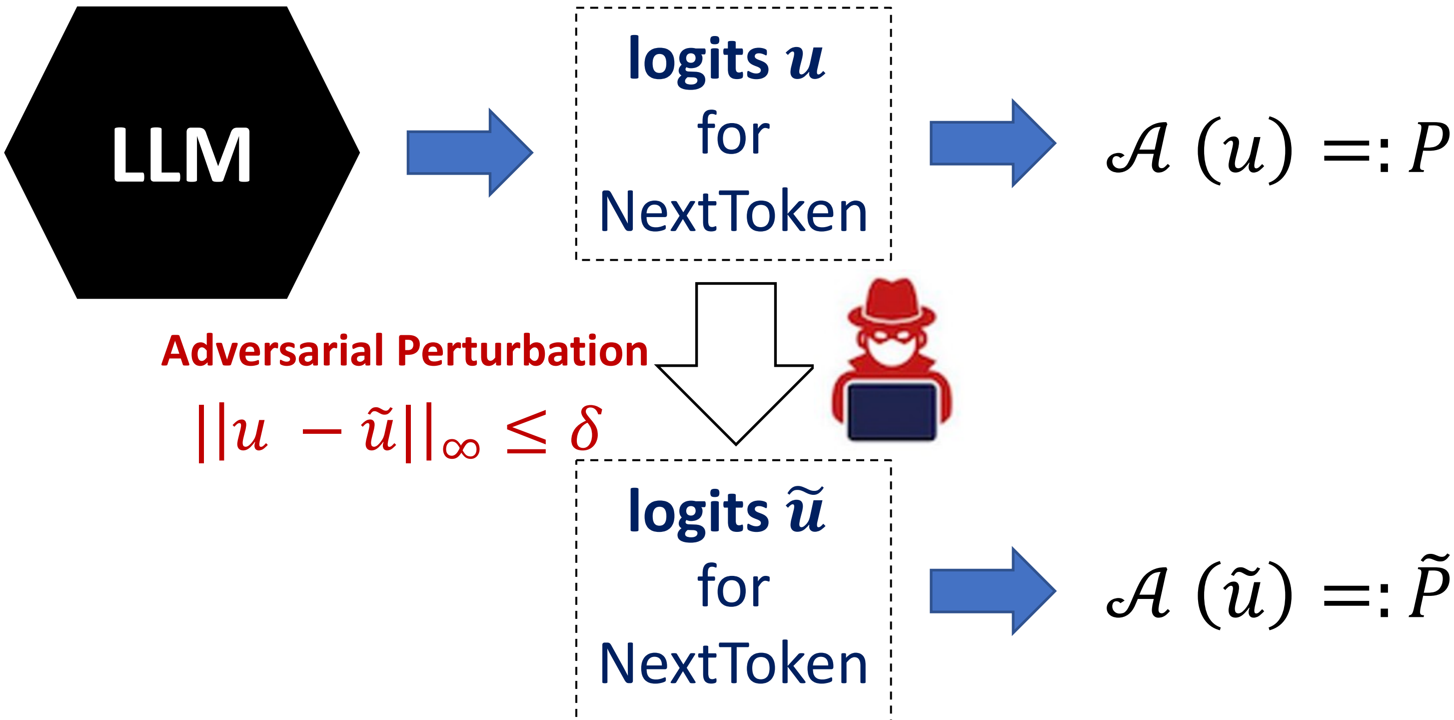
Permute-and-Flip does nothing but replacing Step 1 by **sampling without replacement.**

The advantage of PF Sampling is that it gets all the nice properties of the softmax but improves the perplexity.

Methods	Perplexity	Computational Efficiency	Diversity	Watermark	Robustness
Search (e.g., Beam)	Lowest	✗	✗	✗	✗
Greedy	Low	✓	✗	✗	✗
Softmax Sampling	Moderate	✓	✓	✓	✓
Top-p Sampling	Low (for small p)	✓	Depends on p	✓	✗
Top-k Sampling	Low (for small k)	✓	Depends on k	✓	✗
PF Sampling (ours)	Lower than Softmax	✓	✓	✓	✓

Table 1: Comparison of different decoding methods against five desiderata.

Robustness against adversarial perturbation to the logits



***Definition**: L-robustness.*

$\mathcal{A}$ is L-robust if $\left|\log\left(\frac{dP}{d\tilde{P}}\right)\right| \leq L\,\delta$

Both Softmax and P&F are provably robust, but P&F is up to 2x better than Softmax at “optimization”

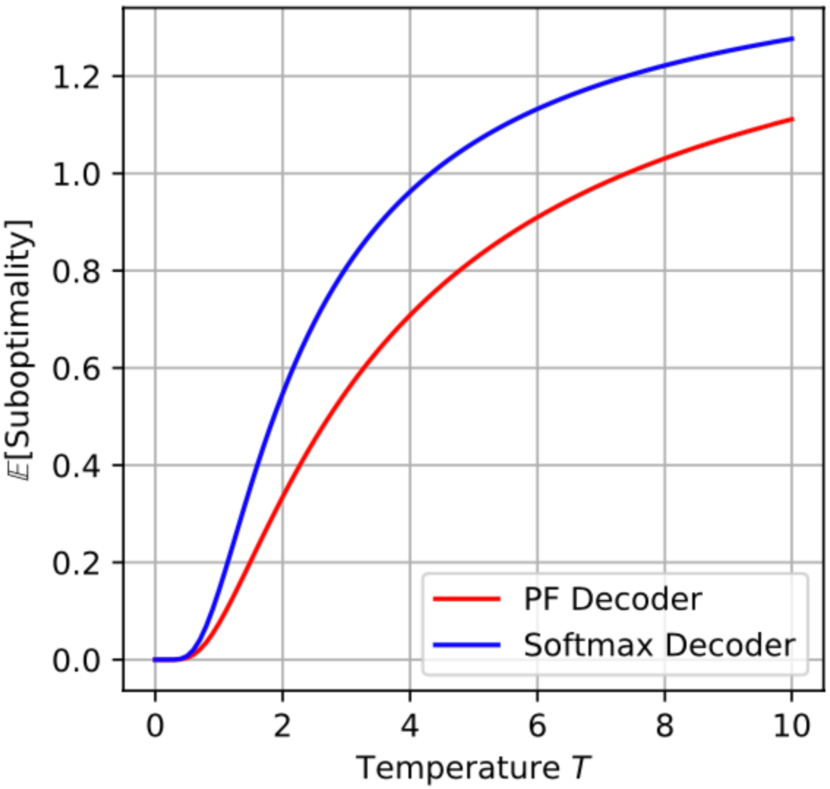
Theorem (McSherry and Talwar, 2007):
Softmax sampling is 1/T-robust.

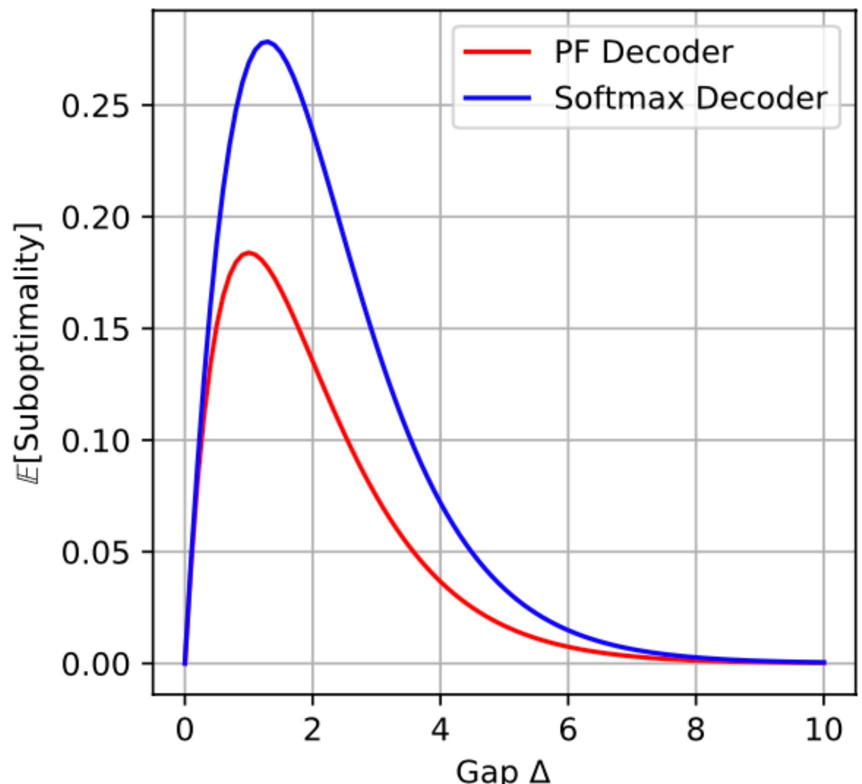
Theorem (McKenna and Sheldon, 2021):

1. Permute-and-Flip sampling is 1/T-robust.
2. For the same T, PF dominates Softmax in terms of expected suboptimality.
3. PF is Pareto-optimal in robust-suboptimality tradeoff.

PF decoder dominates softmax decoder for all parameter choices

Example: Two token vocabulary, logits $u = [0, \Delta]$.
Suboptimality: $u^* - \mathbb{E}[u]$





PF improves perplexity on open-domain generation datasets

Method	PPL1↓	PPL2↓
C4, T=1.0, Llama2-7B		
Sampling	$12.47_{0.32}$	$15.31_{0.41}$
PF	$8.94_{0.20}$	$10.75_{0.25}$
C4, T=0.8, Llama2-7B		
Sampling	$4.23_{0.06}$	$4.91_{0.08}$
PF	$3.54_{0.06}$	$4.11_{0.08}$
Alpaca, T=1.0, Llama2-7B-Chat		
Sampling	$1.74_{0.02}$	$2.41_{0.04}$
PF	$1.65_{0.02}$	$2.30_{0.04}$

TL;DR of our results

1. We propose “Permute-and-Flip Decoding”
    - PF dominates Softmax in robustness-perplexity tradeoff.

2. A cryptographic watermark for Permute-and-Flip
    - Enjoys all nice properties of the Gumbel watermark
    - Slightly better detectability-perplexity tradeoff

From Gumbel-Softmax trick to Exponential-PF trick

- Gumbel-Softmax trick (Gumbel, 1948)

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

$$G_t(y) \sim Gumbel(0,1) \;\; i.i.d$$

- Exponential-PF trick (Ding et. al, 2021)

$$y_t \sim \mathrm{Permute\&Flip}\left(\frac{u_t(y)}{T}\right) \Leftrightarrow \boxed{\begin{array}{c} y_t = \arg\max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + E_t(y). \\ E_t(y) \sim Exponential(1) \;\; i.i.d. \end{array}}$$

ReportNoisyMax from Differential Privacy.

Idea to watermark PF-Decoding

- Gumbel-Watermark (Aaronson, 2022)

$$y_t = \arg\max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + G_t(y)$$

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

- PF-Watermark (Ours)

$$y_t = \arg\max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + E_t(y).$$

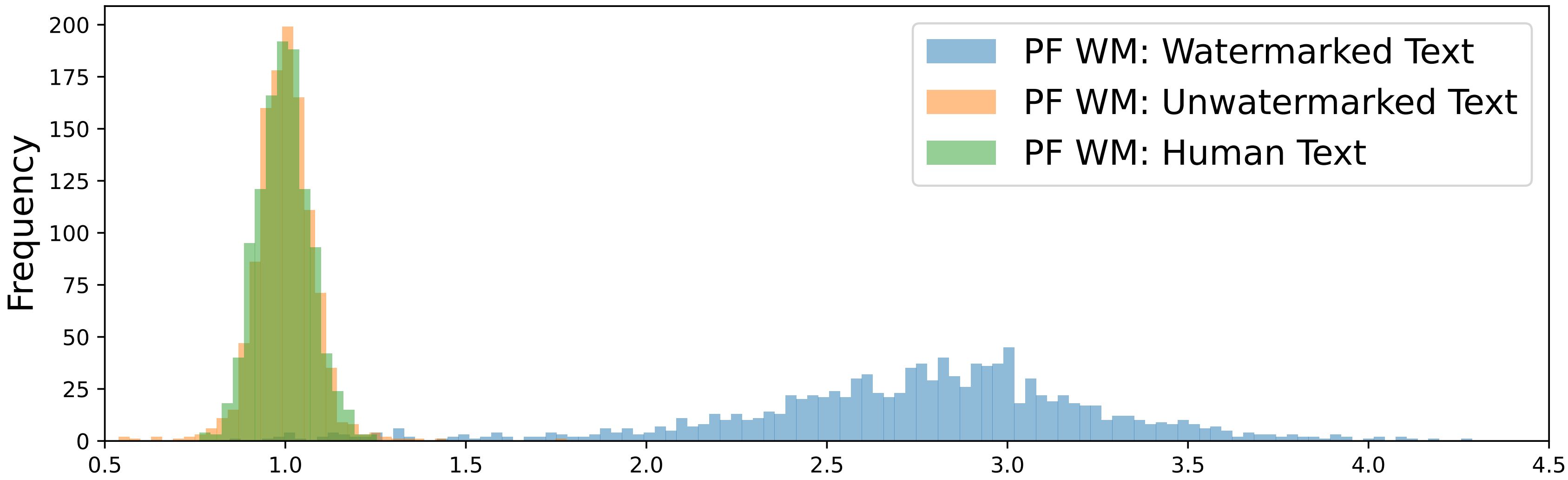
$$\boxed{E_t(y) \sim Exponential(1)\ \ i.i.d.}$$

Make them pseudo-random!

Detection score for PF-watermark

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

where $r_t(y) = F_{y_{t-m:t-1},\mathsf{k}}(y)$



Guarantees of PF-watermark are analogous to those of the Gumbel

- Distortion-free (ex ante)
  - Computationally indistinguishable from PF-decoding.

- Precise FPR control
  - TestScore/n → 1 under the null hypothesis.
  - Under the null hypothesis, the test-score follows a Gamma distribution.

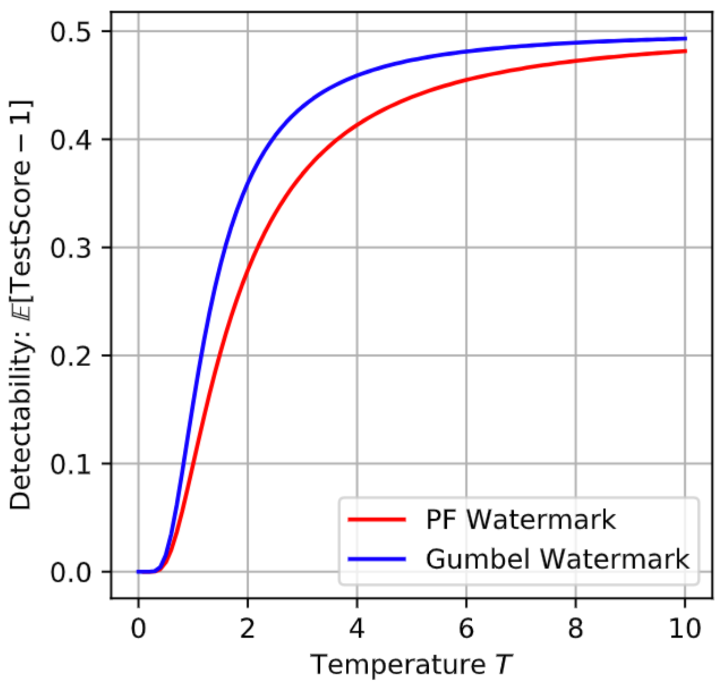
- High power if generated text has *high-entropy*
  - TestScore/n → α for α>>1 under the alternate hypothesis

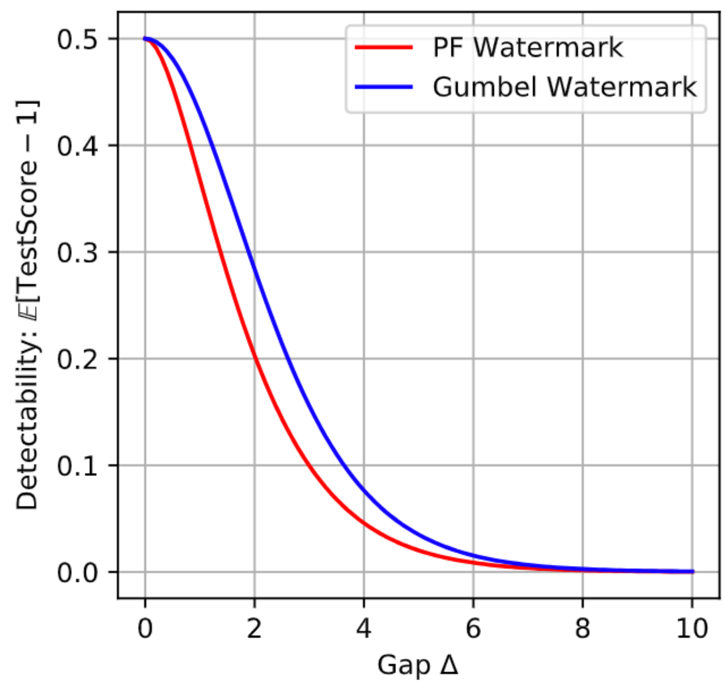
The amount of signal adapts to the entropy

- High entropy: Perfectly Random
  - E[TestScore_PF] = E[TestScore_Gumbel] = 1 + 1/2 + 1/3 + … + 1/|V|.
- No entropy:
  - E[TestScore_PF] = E[TestScore_Gumbel] = 1
- In between: roughly proportional to entropy

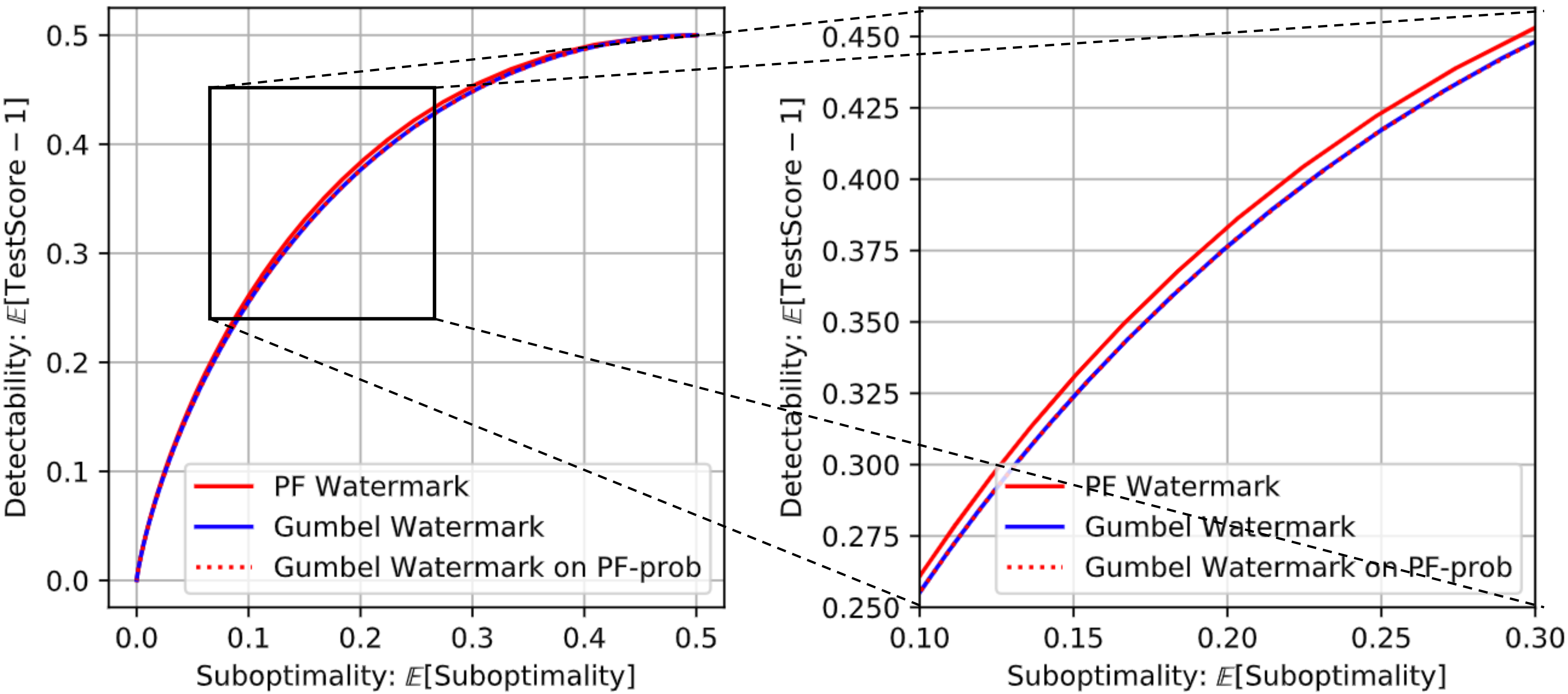
How does PF-watermark compare to Gumbel watermark?

- **Example**: Two token vocabulary, logits $u = [0, \Delta]$.
- **Detectability**: $\mathbb{E}$[Score|WM] -$\mathbb{E}$[Score|No WM]
- **Suboptimality**: $u^* - \mathbb{E}[u]$



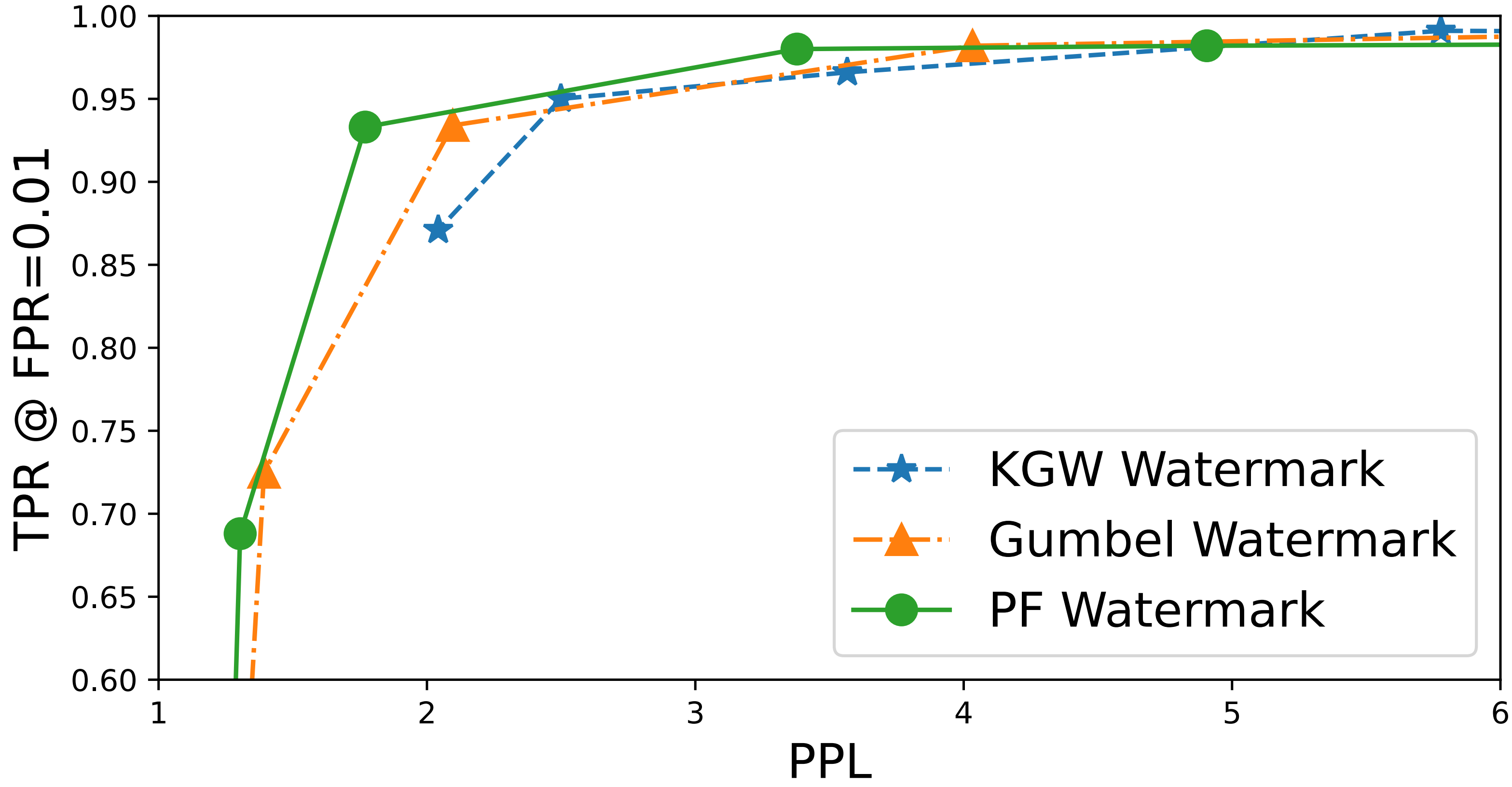


Plotting detectability against suboptimality as we adjust T



PF has more favorable tradeoff curves than Gumbel

On real datasets: the PF watermark provides better Detectability-Perplexity Tradeoffs



Checkpoint

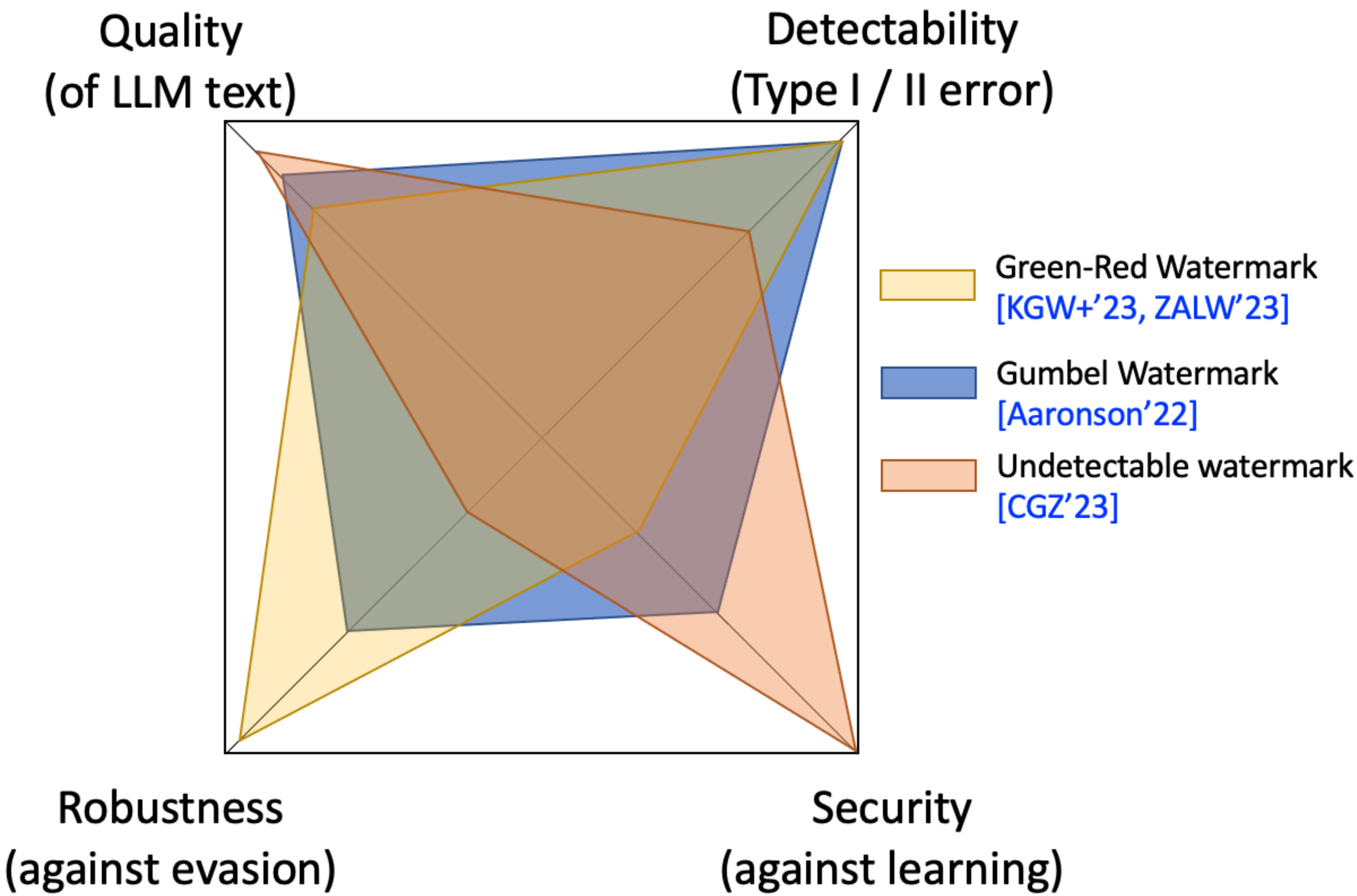
- We propose Permute-and-Flip decoding and developed a natural watermarking scheme for it.
  - For the same perplexity, it improves detectability and robustness.

- Interesting connection to the differential privacy literature --- more interplays in the future.

Outline of the talk

- Formally defining LLM watermarks
- Two recent work
  - Provable Robust Watermark of LLMs
  - Permute-and-Flip decoding and watermarking
- Open problems

Optimal tradeoffs in LLM watermarks



Enhancing robustness

- Optimality in the Edit model. Is Unigram WM the optimal?
- More realistic threat models

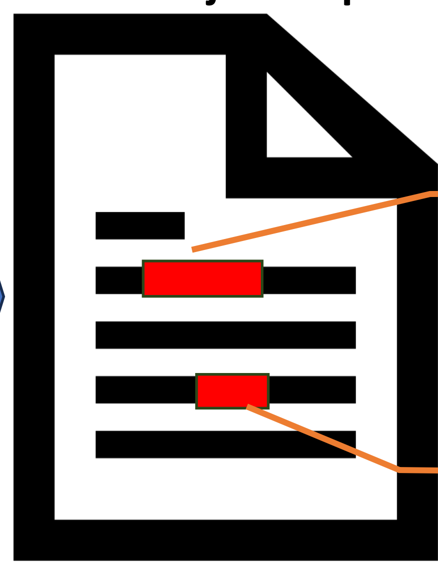
Bob generated a section using **ChatKitten**

Alice generated a paragraph using **ChatPuppy**

Dave wrote the remaining *by himself.*

Eric gave it an *editing pass*.

Term Project Report



Can watermarking help to catch the following?

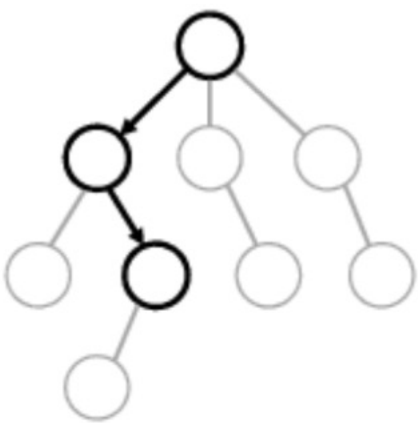
Detection with 99% confidence. Line 57-86 used **ChatKitten** outputs, related to user 75801 (nickname: Bob).

Detection with 90% confidence. Paragraph 35 used **ChatPuppy** outputs, related to User 14234 (nickname: Alice)

More co-design of decoder and watermarks?

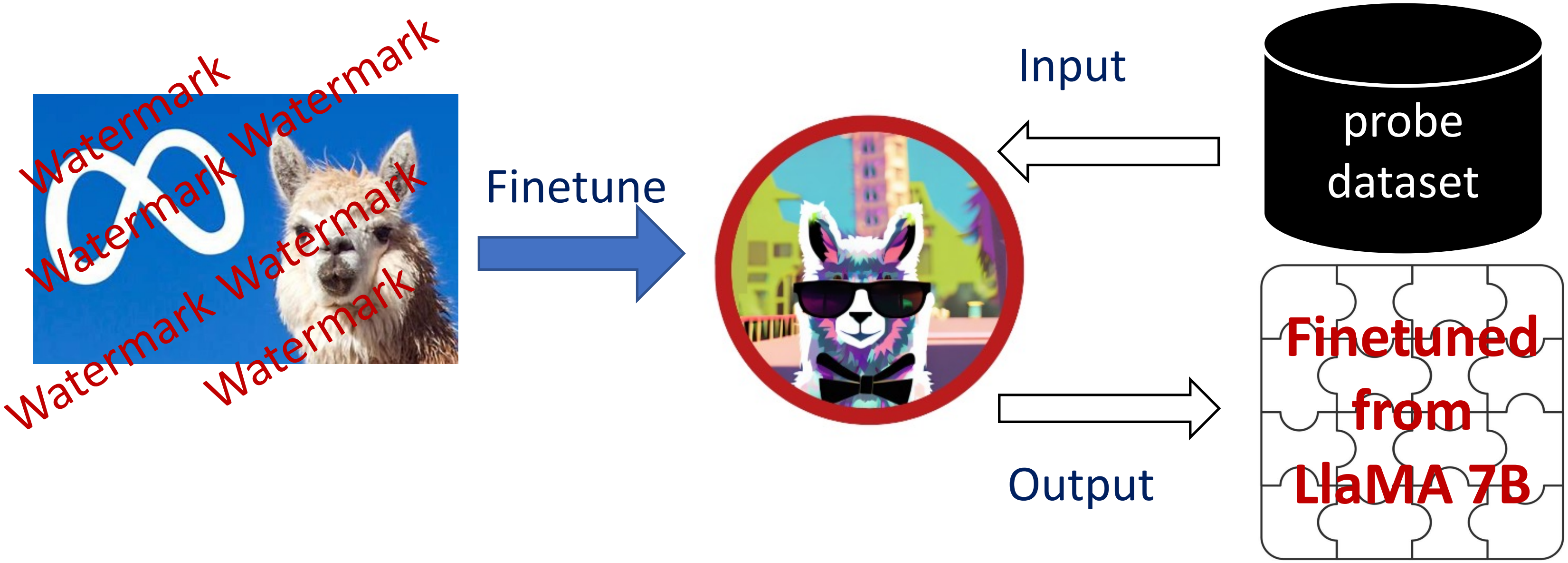
- Provable Watermarking for Beam search?
  - Or other methods that aim at solving the sequence level MLE decoding.

- When can we still watermark without entropy?



How do we watermark **open-source** LLMs?

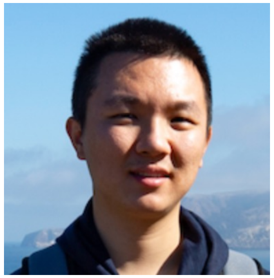
- Model watermarks that are resilient to finetuning



Thank you for your attention!

- **Permute-And-Flip: An Optimally Robust and Watermarkable Decoder for LLMs**
  Xuandong Zhao, Lei Li, Yu-Xiang Wang.
  Technical report, 2024 [arxiv, code]

- **Provable Robust Watermarking for AI-Generated Text**
  Xuandong Zhao, Prabhanjan Ananth, Lei Li, Yu-Xiang Wang.
  ICLR 2024 [arxiv, slides, code, demo]



Xuandong Zhao

Lei Li

Prabhanjan Ananth