ACL 2024
Bangkok, Thailand

Website; Q&A

Watermarking for Large Language Models

Part II: Text Watermarking


Xuandong Zhao
UC Berkeley


Yu-Xiang Wang
UC San Diego


Lei Li
CMU

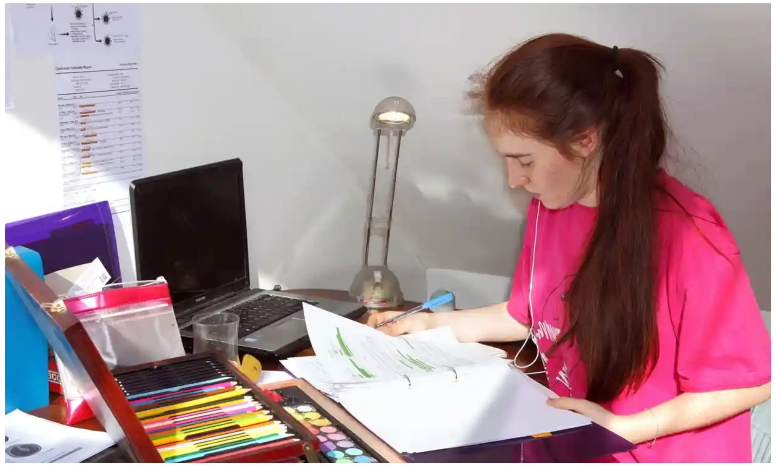
Reiterating the Motivation

- We need to **reliably** detect AI generated texts.

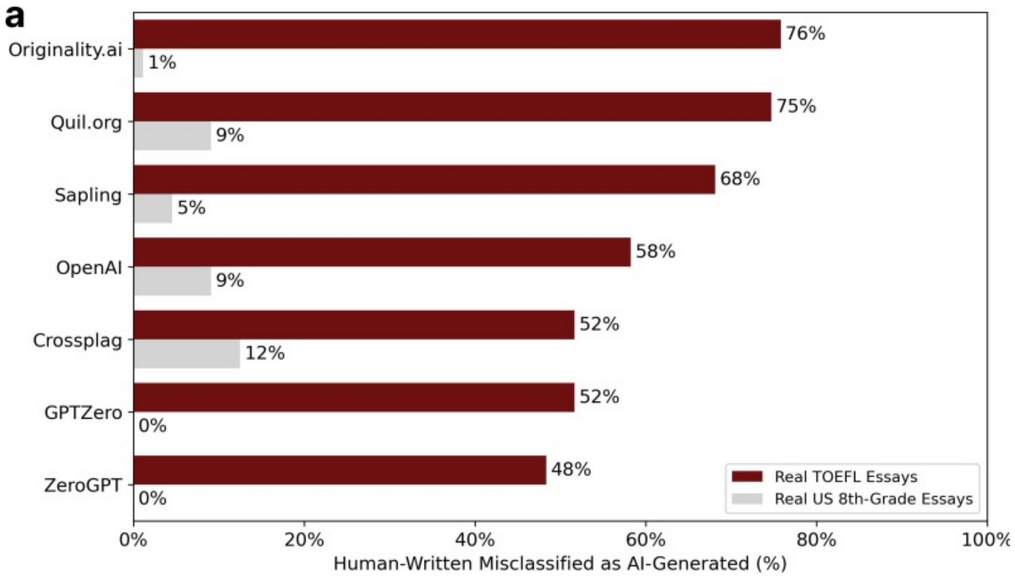
- AI classifiers can never be reliable enough to work (out of distribution)

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.

Xuandong described a simple watermark scheme that appears to work!

1. Does it always work? Or we got lucky in those examples?
2. Can we do better than Green-Red watermark?
3. How do we even define ”better”?
4. How much better any watermarking schemes can do?

Many of these questions require theory to answer.

Remainder of Part 2: Watermarking Text

- Formal Problem setup
- Popular Watermarking Schemes
    - Green-Red watermark
    - Gumbel watermark
    - Pointers to others
- Open problems and new directions

Recall: 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

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

input/prompt $\rightarrow$ $\hat{\mathcal{M}}$ $\rightarrow$ watermarked text

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

key $\rightarrow$

suspect text $\rightarrow$

$\rightarrow$ 1 (watermark) or 0 (no watermark)

Four key metrics of a watermarking scheme

Quality	**Detectability**
Robustness	**Security**

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 on downstream tasks.

Provable theoretical results on quality of the Watermark

	Single token	Whole sequence	Many sequences
ex ante *0-distortion*	Aaronson	Kuditipudi et al	Christ et al
ex post *small-distortion*	Zhao et al	Zhao et al (through composition)	?

Detectability: A hypothesis testing view of LLM watermarks

- 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. or F1-score.

- **Theory**： Can we control FPR. Can we prove high power? Are the tradeoff optimal?

This FPR here may be different from the FPR your are familiar with!

- In ML/NLP experiments, e.g., sentiment classification:

  Your classifier makes n predictions on a test corpus.

  FPR = # of False Positive / Total number of Negative Examples

  Implicitly, this FPR is specific to the data distribution P(input x | label of x is "-")

- FPR in the LLM watermarking is distribution-free:

  FPR = Probability of "Detector" making a mistake for any fixed Input.

Randomness is over the secret key only!

Not all LLM generated text are easily watermarkable.

Example 1: 

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 ….

Example 2: 

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

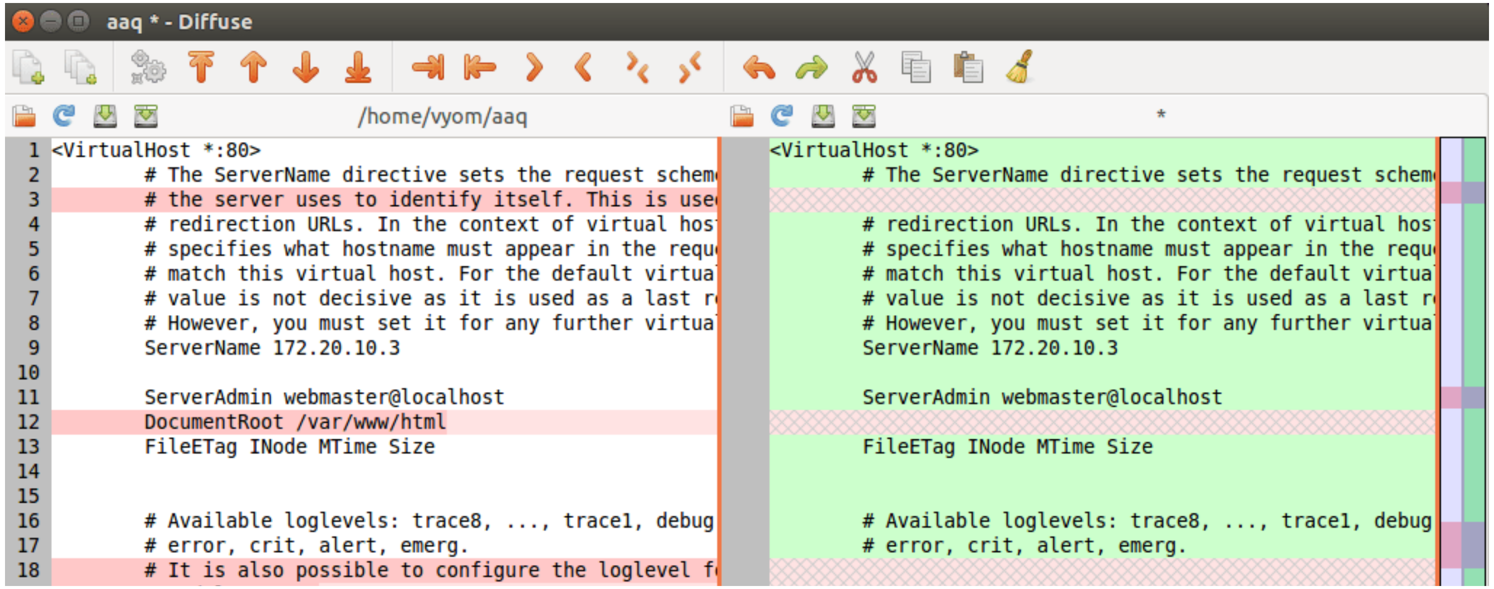
Goal! Goal! Goal! Goal! …

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 / edits / improving

- Shuffling: Move things around



Formally defining robustness

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.

- Is the "detector" still able to detect that the text was generated by GPT4?
    - Case 1: I changed a few words
    - Case 2: I didn't like it and rewrote the whole thing.

- Need to specify a family of possible attacks

  e.g. parameterized by the Edit Distance allowed

Security: How difficulty is it for an attacker to learn the secret key?

- Evasion attacks: increase Type II error

- Spoofing attacks: increase Type I error

- A sufficient condition from (Christ, Gunn, Zamir 2023): Original $\mathcal{M}$ and $\widehat{\mathcal{M}}$ are computationally indistinguishable.

Other ~~desirable~~ essential properties of an LLM Watermarking Scheme

- **Model agnostic detection:** Does not require calling the LM APIs at detection time.

- **Low computational overhead:** $\hat{\mathcal{M}}$ is as efficient as $\mathcal{M}$ in computation, memory, throughput.

Checkpoint: Four metrics in evaluating LLM Text Watermarks

- Quality: Relative (KL-div from unwatermarked) or absolute (PPL?)
  ex ante or ex post? Single token, or whole sentence

- Detectability: FPR should be distribution-free, and controllable. TPR depends on “entropy” of the generative procedure.

- Robustness: Need a threat model. We choose “Edit Distance”

- Security: Similar need a threat model. More open ended.

They are nuanced and often case-by-case!

Remainder of Part 2: Watermarking Text

- Formal Problem setup

- Popular Watermarking Schemes
  - Green-Red watermark
  - Gumbel watermark
  - Pointers to others

- Open problems and new directions

Let's inspect the watermarking schemes against these metrics

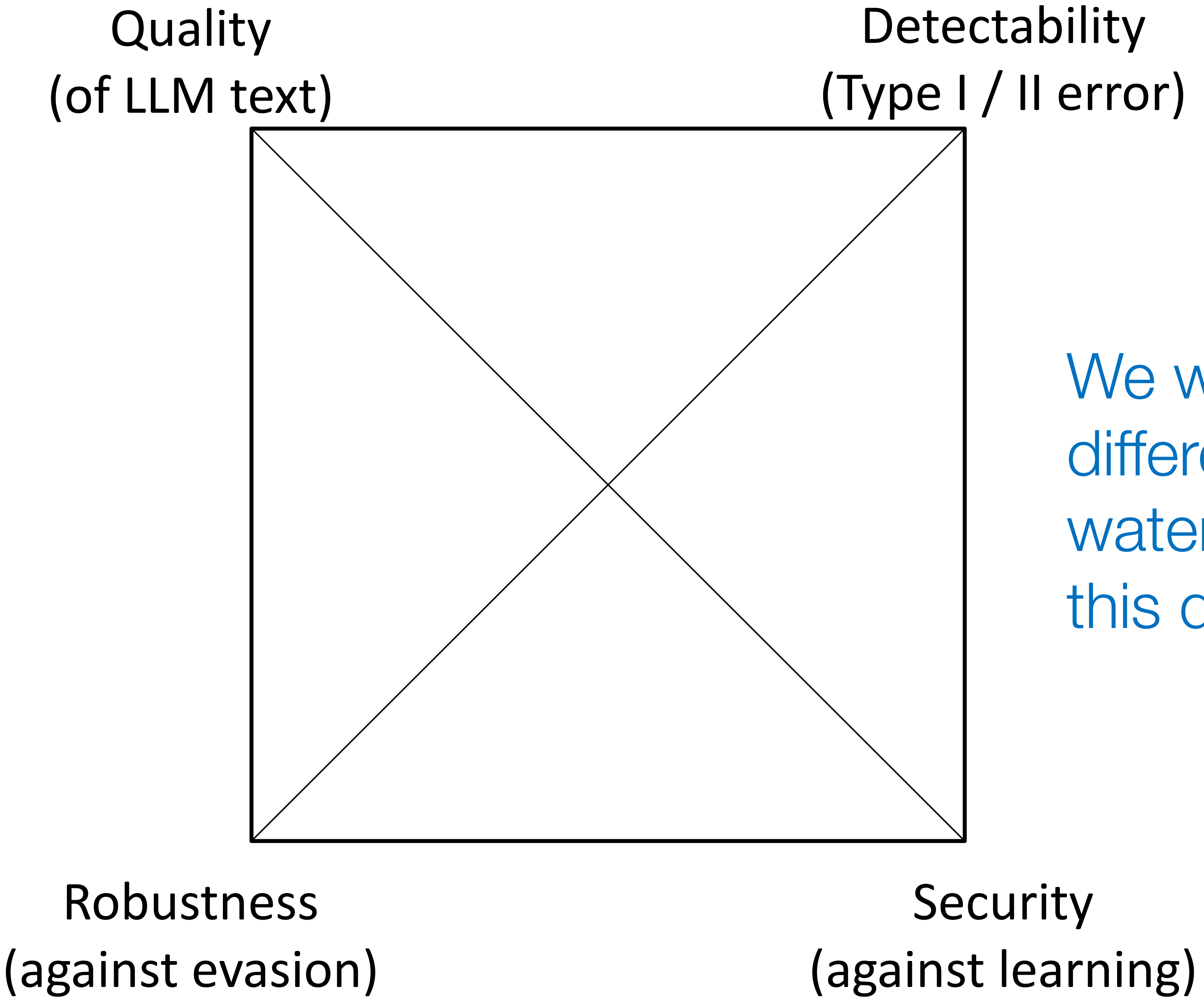
- Focus on two representative watermarks

1. Green-Red Watermark (Kirchenbauer et al, 2023; Zhao et al. 2023)

2. Gumbel watermark. (Aaronson, 2022)

3. Briefly describe others

   e.g.(Christ, Gunn, Zamir 2023), (Kuditipudi et al, 2023) (Hu et al ,2023) (Zhao, Li, W., 2024)



We will put different watermarks on this diagram!

Quality guarantee of Green-Red Watermark

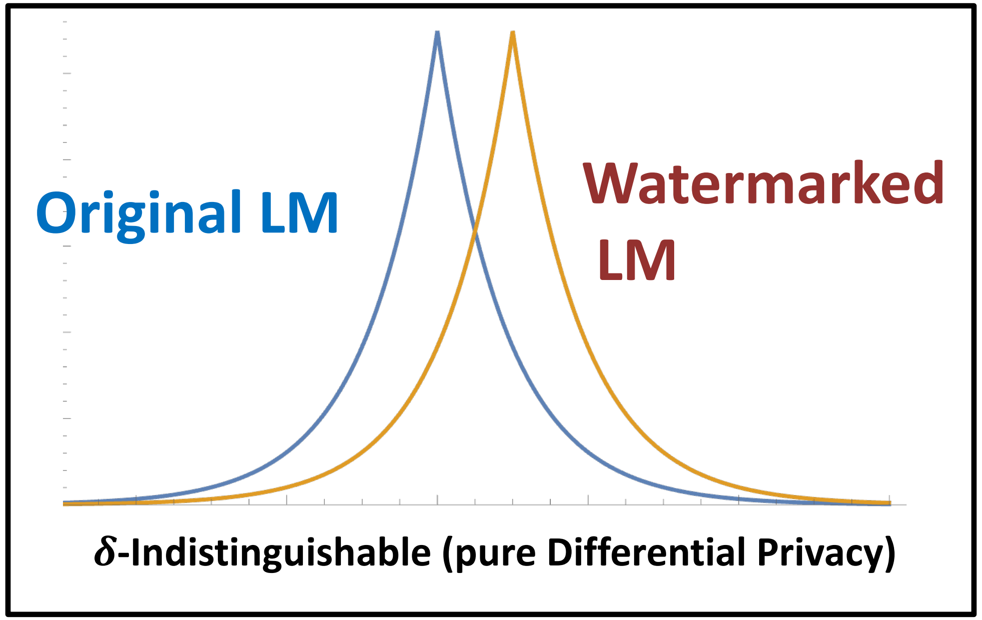
(Kirchenbauer et al. 2023; Zhao et al. 2023)

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

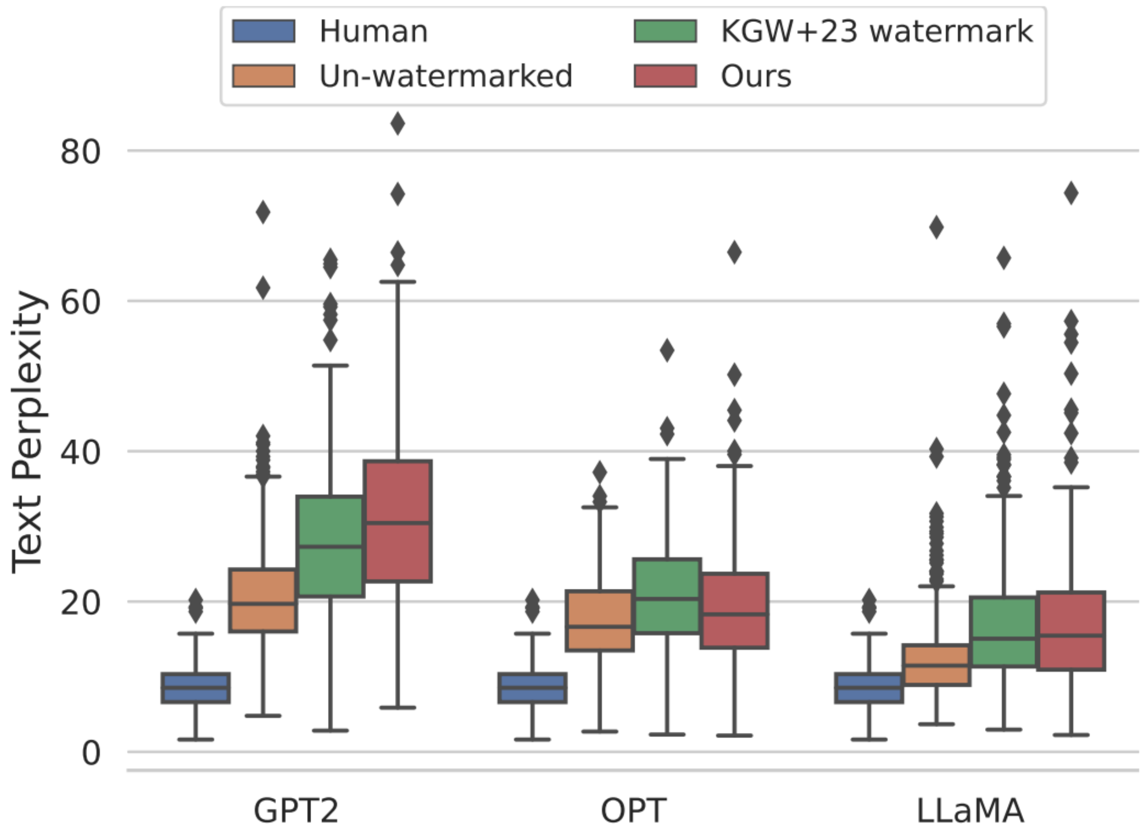
$$\widehat{\mathcal{M}}:\ \ y_t \sim \text{Softmax}(\text{logits}(\text{Prompt}, y_{<t}) + \delta \cdot \mathbf{1}(\cdot\ is\ green))$$

Theorem: Any prompt, any prefix text. 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.

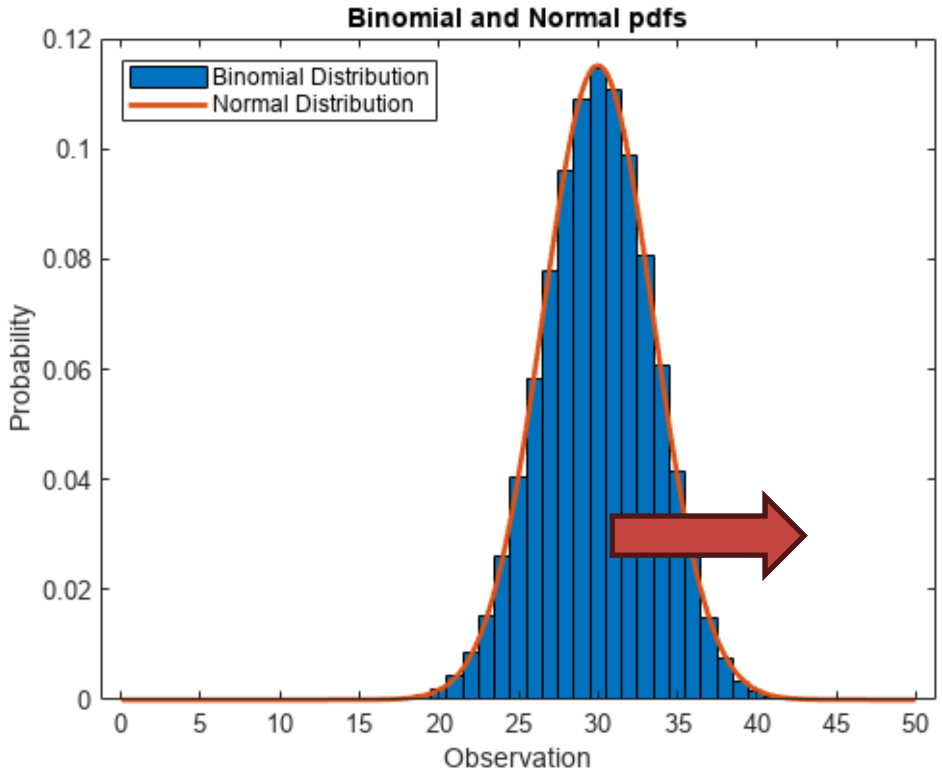
	Avg Score	STD
Un-watermarked	3.660	0.655
Watermarked	3.665	0.619

Table 3: Human evaluation result.

Detectability Guarantees for Green-Red WM

- Detection score $z = \frac{|y|_G - \gamma n}{\sqrt{n\gamma(1-\gamma)}}$, where $|y|_G = \sum_i 1(y_i \in G_i)$

(pretend that $1(y_i \in G_i) \sim \mathrm{Ber}(\gamma)$ independently)



When unwatermarked, new prefix each time, this is valid.

When watermarked, the distribution shifts to the right by roughly e^{δ} multiplicatively.

Recall: How is the *Green* list generated?

- *Randomly* selecting γ fraction of the vocabulary, e.g., 0.5
- (Kirchenbauer et al.): Different green list at each time t as function of the prefix with length (m-1). Default: m=2

> 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”.

How valid is the “independence” assumption?

The Raven

Once upon a midnight dreary, while I pondered, weak and weary,
Over many a quaint and curious volume of forgotten lore—
While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
"'Tis some visiter," I muttered, "tapping at my chamber door—
Only this and nothing more."

—Edgar Allan Poe

- It is easier to satisfy when m is large

- Unigram- Green-Red watermark, i.e., m = 1

  A lot more complicated in dealing with the dependence. (Zhao et al., 2023).

Detection guarantees (Zhao et al., 2023).

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

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

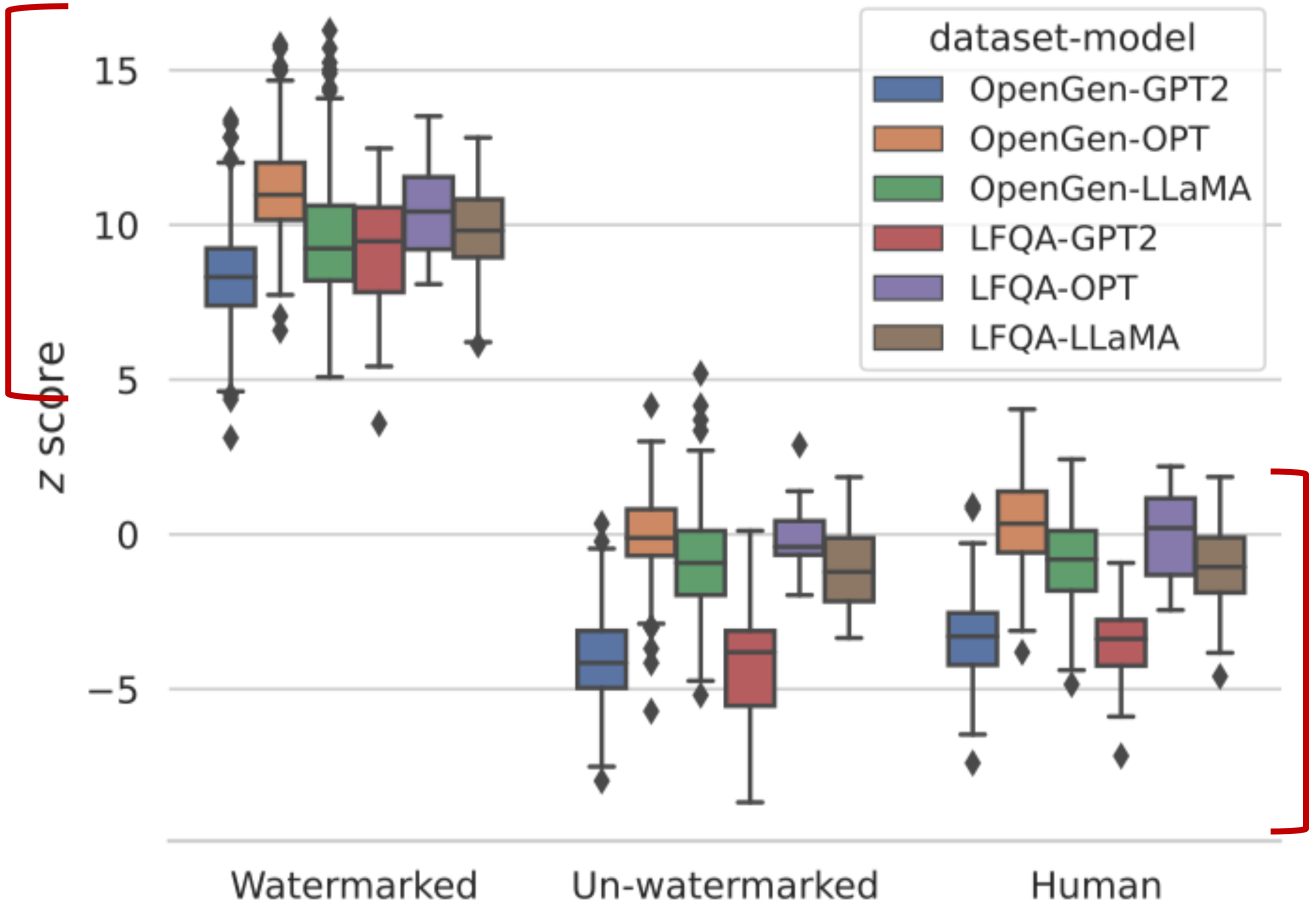
where V and C_{max} measure the **diversity** of the text. If unique, then Z=1 and Cmax = 1

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

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

Our detection guarantees Illustrated

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



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

H1: Alternative

H0: “Null”

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

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

Robust to a constant fraction of edits!

Adversary can have any side information,
can even know the Green List.

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 to decision / m

Limitation of the Green-Red Watermark

- It changes the distribution of the Language Model
- Choice of δ determines quality-detectability tradeoff.

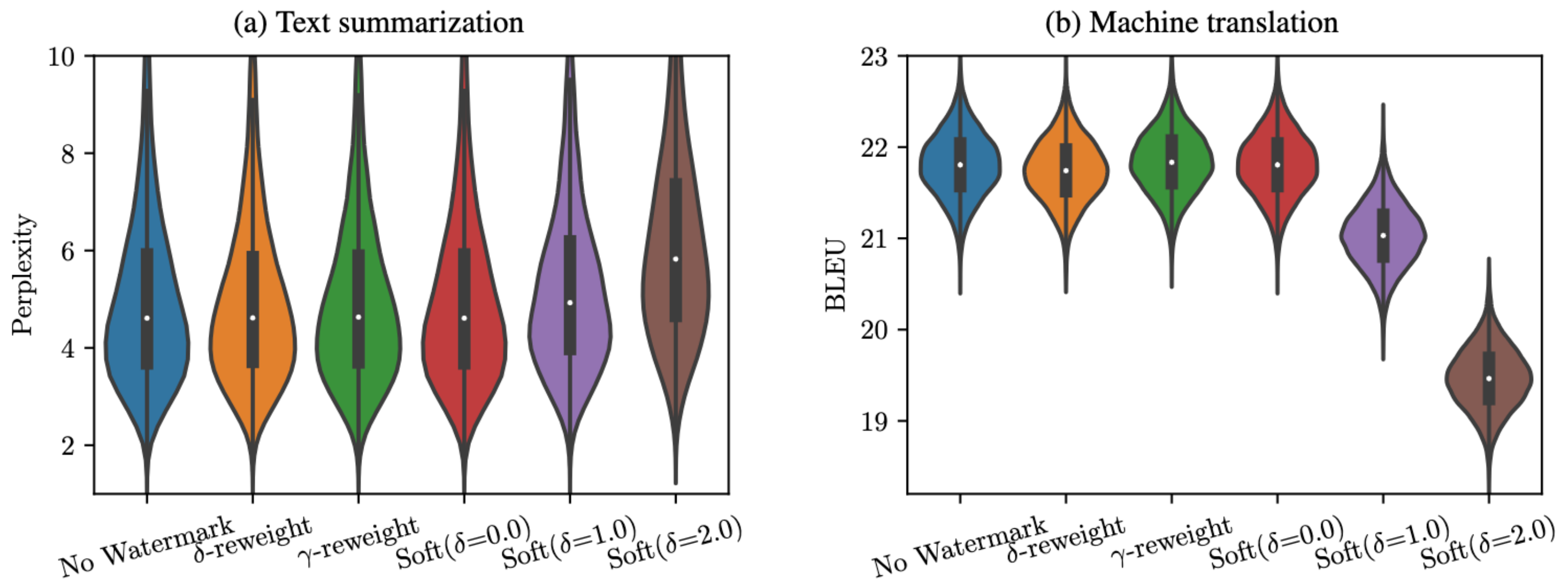
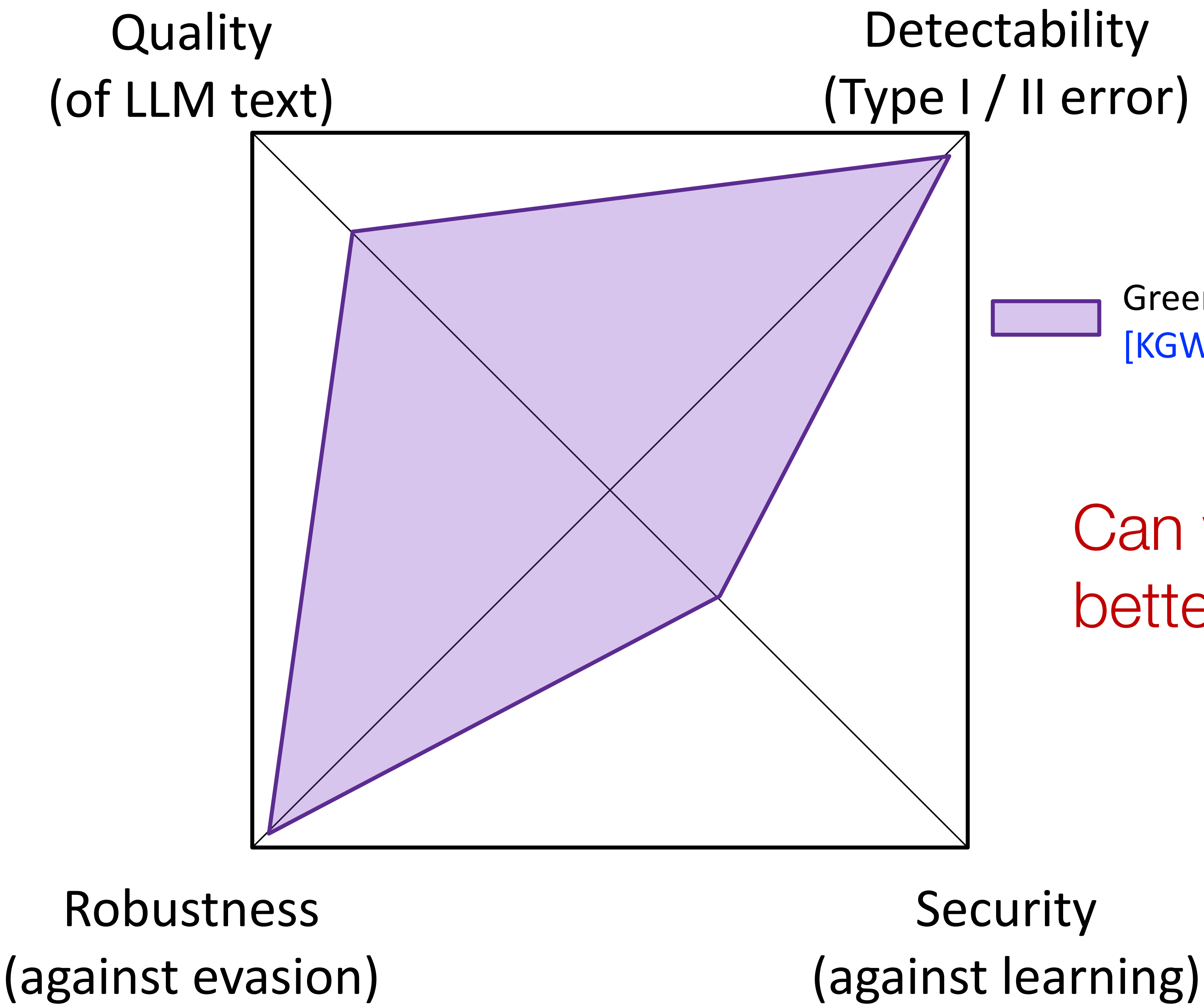


Figure 3: Distribution of perplexity of output for TS and BLEU score for MT.

(Figure from Hu et al 2023 “unbiased watermark for LLMs”)



Can we do better in quality?

There are watermarking schemes that are "Distortion Free" (aka "unbiased")

"Distortion-Free": For any "Input"
$\mathcal{M}(Input) \sim \widehat{\mathcal{M}}(Input)$, i.e., they are identically distributed.

Gumbel watermark (Aaronson, 2022)

Undetectable WM (Christ, Gunn, Zamir 2023)

Distortion-Free WM (Kuditipudi et al, 2023)

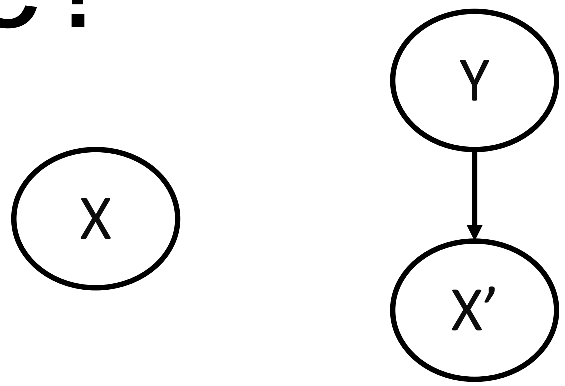
Unbiased WM (Hu et al ,2023)

Permute-and-Flip WM (Zhao, Li, W., 2024)

Demystify "distortion-free" property: How is it possible?

- **Example:** $X \sim \text{Bernoulli}(0.7)$,

  $Y \sim \text{Uniform}([0,1])$, $X' = 1(Y<0.7)$.



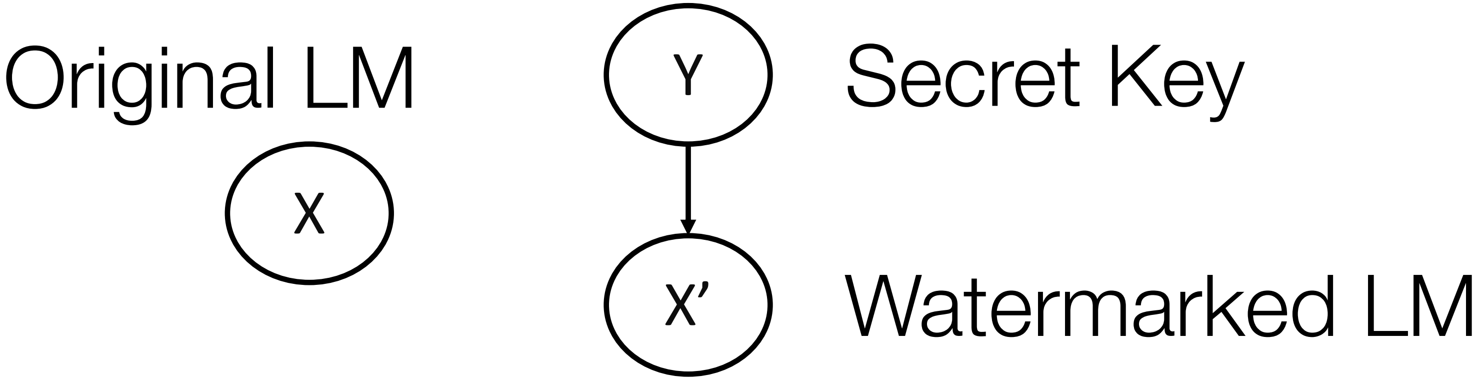
- Check that:

  $X \sim X'$ marginally (i.e., they are identically distributed)

  But if we observe Y, $X'|Y$ is deterministic.

X and X' are only marginally identically distributed.
Knowledge of Y creates the "asymmetry" we need.

From the Latent Variable view of LLM Watermarking schemes



- In Green-Red watermark, Y is the (random) green list.
- But the marginal distribution of X' is **not the same** as X.

Quiz question: modify the Green-Red Watermark such that X' ~ X? Come to me with your idea during break.

Gumbel-Softmax trick and Gumbel Watermark

- Gumbel-Softmax trick (Gumbel, 1948)

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

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

- Idea of the Gumbel Watermark (Aaronson, 2022)

Make them pseudo-random!

The Gumbel noises are the “hidden variables” determined by the pseudo-random functions that we can secret keys.

Intuition behind the Gumbel Watermark

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

- Without the secret key: (notice that G_t are random). The distribution of next token remains unchanged!
- With the secret key, the sequence is deterministic!
- In Detection phase: we don't have the prompt, nor the next token probability. But the selected y_t is biased towards larger G_t regardless.

Detection score of Gumbel Watermark

$$\mathrm{Gumbel}(0,1) \sim -\log\left(\log(1/\mathrm{Uniform}([0,1]))\right).$$

- Let r be the pseudo-random vector iid uniform for every coordinate.

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

No watermark

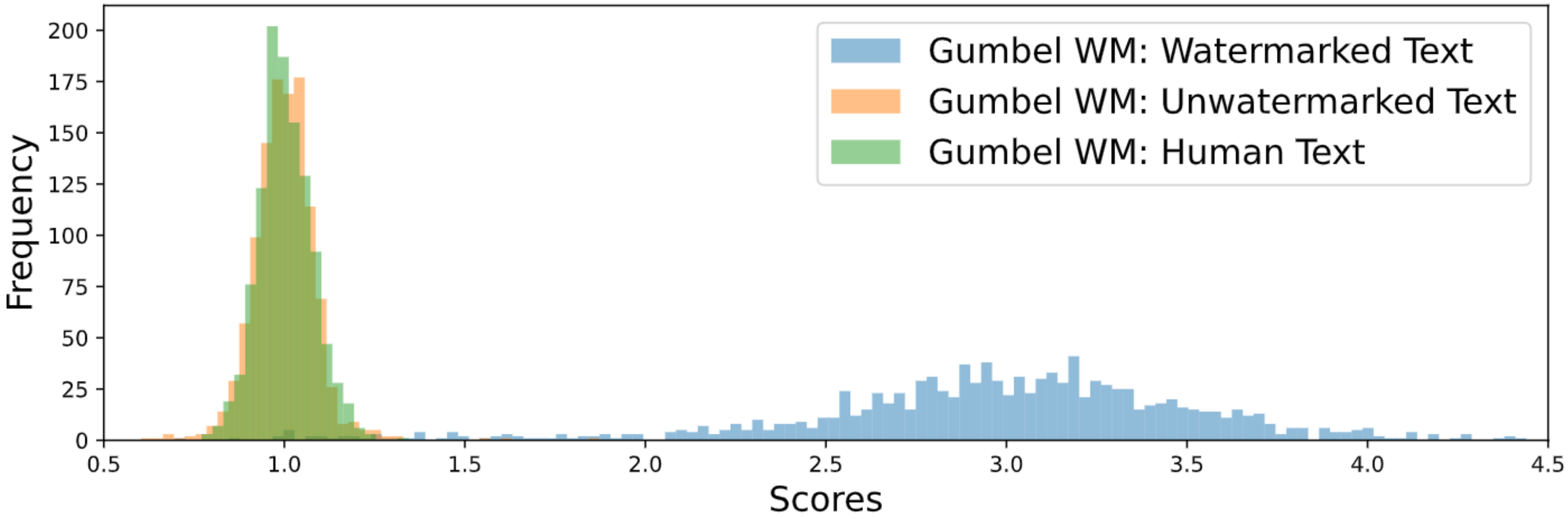
$$\mathbb{E}[\mathrm{TestScore}(y_{1:n})] = n - m$$

Watermarked

$$\mathbb{E}[\mathrm{TestScore}(y_{1:n})] = \sum_{t=m+1}^{n} \mathbb{E}\left[\sum_{y \in \mathcal{V}} p_t(y) H_{\frac{1}{p_t(y)}}\right]$$

$$\geq (n-m) + \left(\frac{\pi^2}{6} - 1\right) \sum_{t=m+1}^{n} \mathbb{E}\left[\mathrm{Entropy}[p_t(\cdot)]\right].$$

Detection score of Gumbel WMs in practice



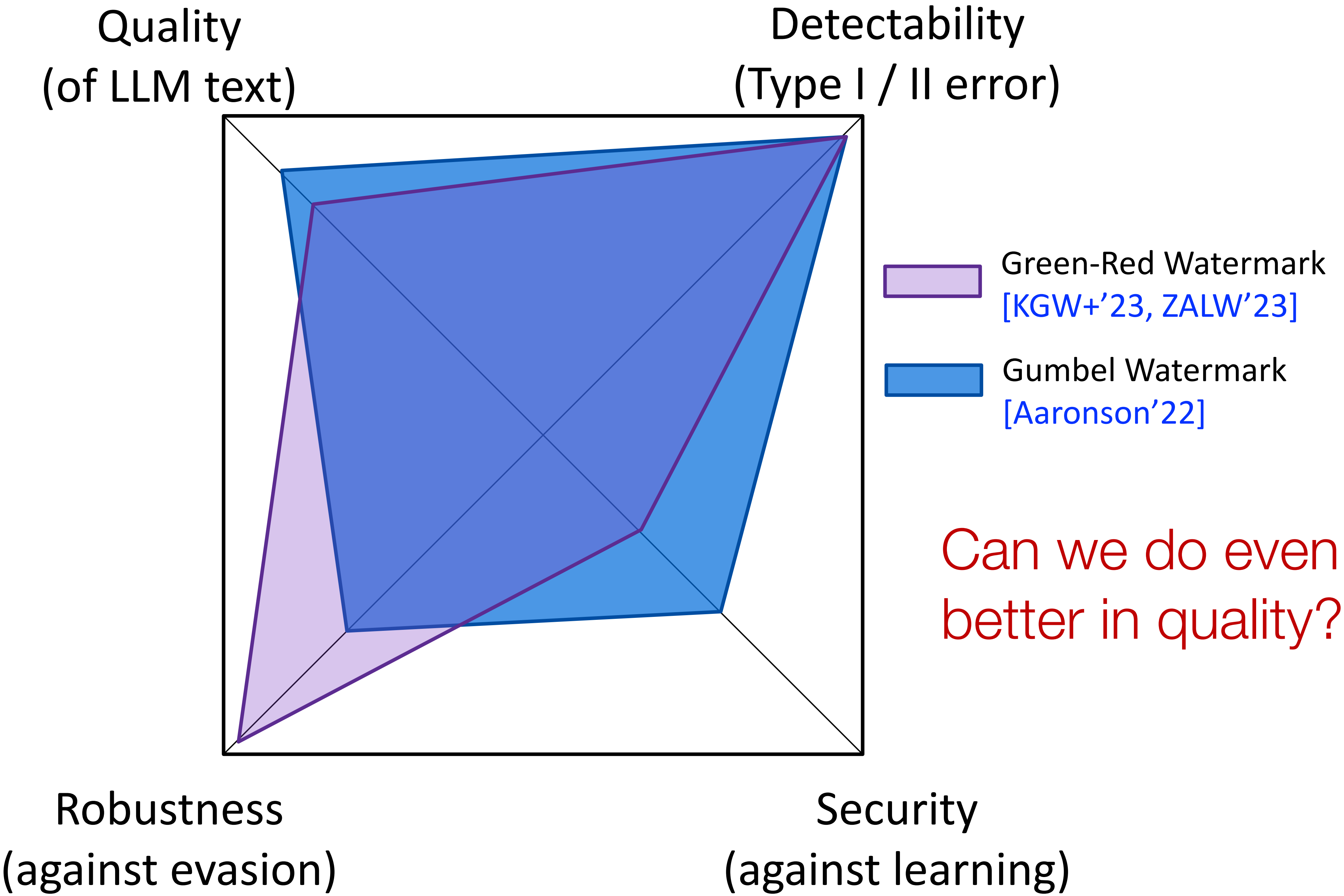
Robustness of Gumbel WM is not bad

- Not "unigram WM" type robust, but still quite robust

Setting	Method	AUC	1% FPR TPR	1% FPR F1	10% FPR TPR	10% FPR F1
No attack	KGW	0.998	0.996	0.989	1.000	0.906
	Gumbel	0.992	0.979	0.979	0.986	0.913
	PF	0.996	0.977	0.980	0.993	0.898
DIPPER-1	KGW	0.661	0.057	0.105	0.317	0.416
	Gumbel	0.838	0.367	0.529	0.642	0.697
	PF	0.824	0.374	0.537	0.622	0.684
DIPPER-2	KGW	0.638	0.051	0.096	0.278	0.375
	Gumbel	0.764	0.239	0.380	0.523	0.608
	PF	0.795	0.250	0.394	0.544	0.625
Random Delete (0.3)	KGW	0.936	0.484	0.644	0.881	0.844
	Gumbel	0.981	0.941	0.960	0.959	0.898
	PF	0.985	0.936	0.956	0.966	0.888

DIPPER-1
DIPPER-2
are "paraphrasing attacks"

(Table 3 of https://arxiv.org/abs/2402.05864)



Can we do even better in quality?

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

- Sentence level distortion-free

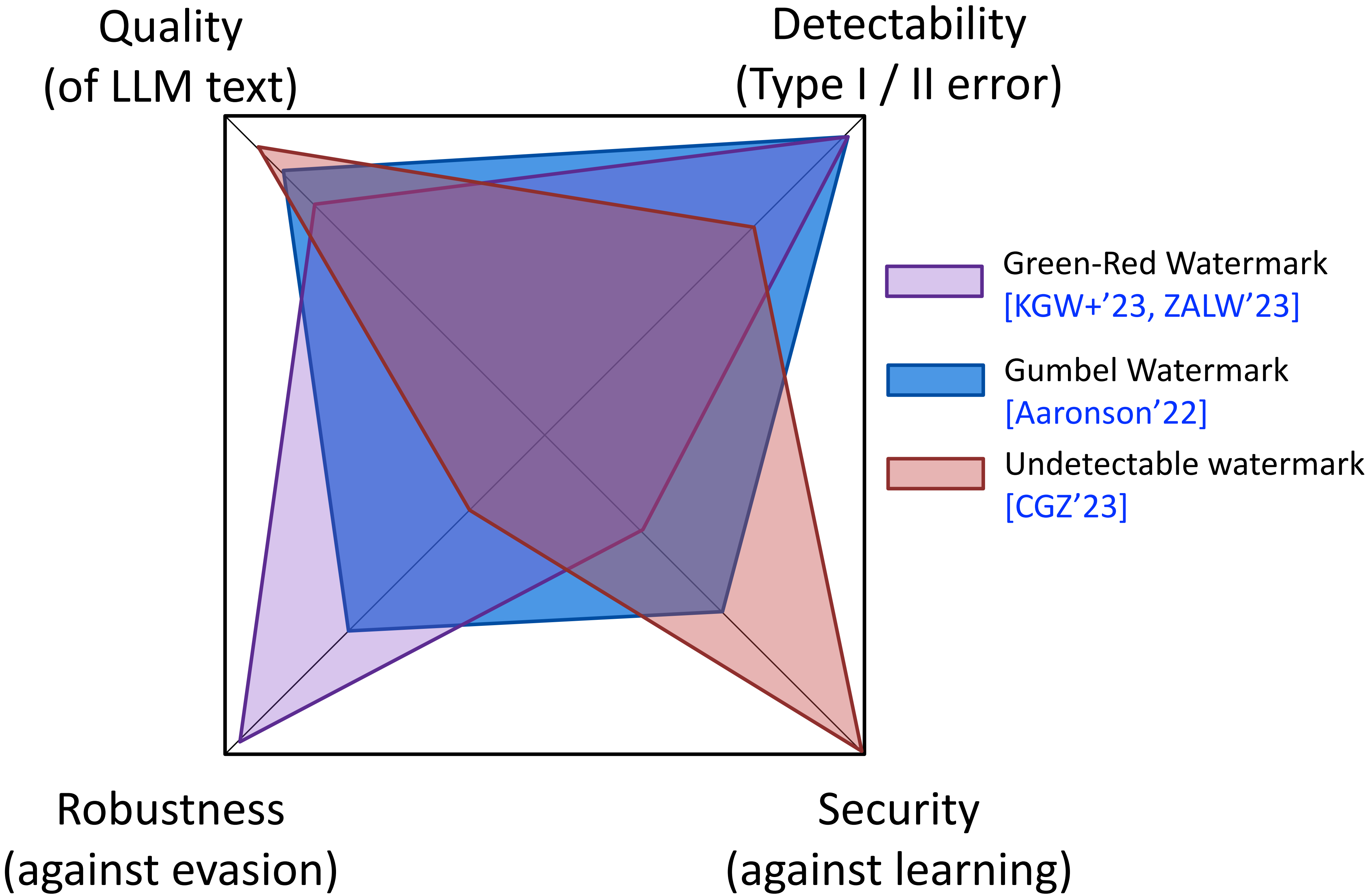
  (Kuditipudi et al, 2023): ”Get multiple keys, rotate the keys being used. In detection time, test with all keys”

  (Hu et al ,2023): ”unique prefix each time within a sentence”

- Polynomially many sentence distortion-free

  1. Do the above two across many sentences.

  2. (Christ, Gunn, Zamir, 2023): “Accumulate sufficient amount entropy before adding watermark! ”


Quality
(of LLM text)
Detectability
(Type I / II error)
Robustness
(against evasion)
Security
(against learning)
Green-Red Watermark
[KGW+’23, ZALW’23]
Gumbel Watermark
[Aaronson’22]
Undetectable watermark
[CGZ’23]

Are “distortion-free” watermarks always better than Green-Red?

- Green-Red watermark leverages the watermark strength parameter δ and temperature T
  - More detectable when entropy is lower.
  - Guarantee valid even if conditioning on the key --- not quite the case with Gumbel.

- Gumbel watermark responds only to temperature T
  - Smaller temperature usually gives better perplexity.
  - Tradeoff between “greediness” vs “detectability”.

For a comprehensive empirical comparison. see Piet et al 2023 “MarkMyWord” https://arxiv.org/abs/2312.00273

From Gumbel-Softmax trick to Exponential-PF trick

- Gumbel-Softmax trick (Gumbel, 1948)

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

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

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

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

ReportNoisyMax from Differential Privacy.

Permute-and-Flip Watermark

- 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)

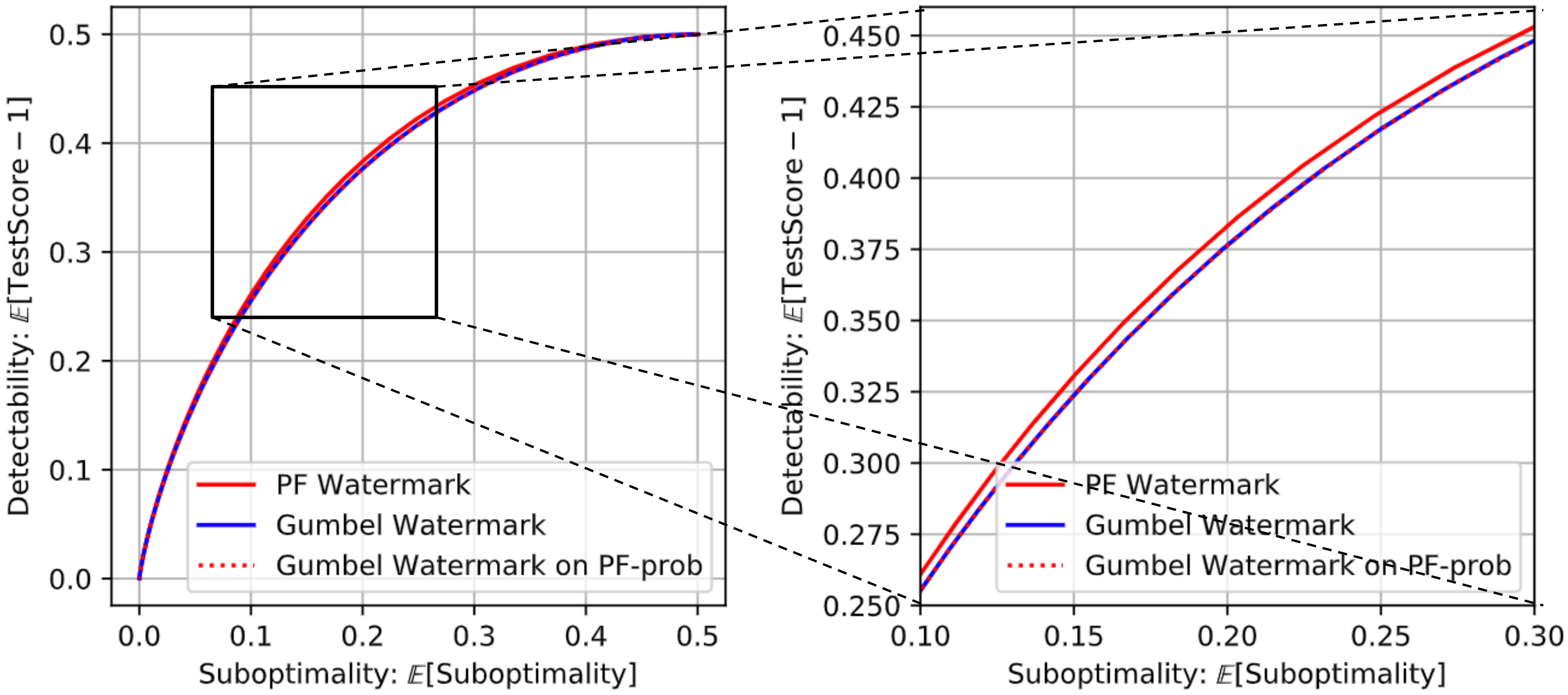
$$y_t = \arg\max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + E_t(y).$$

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

Make them pseudo-random!

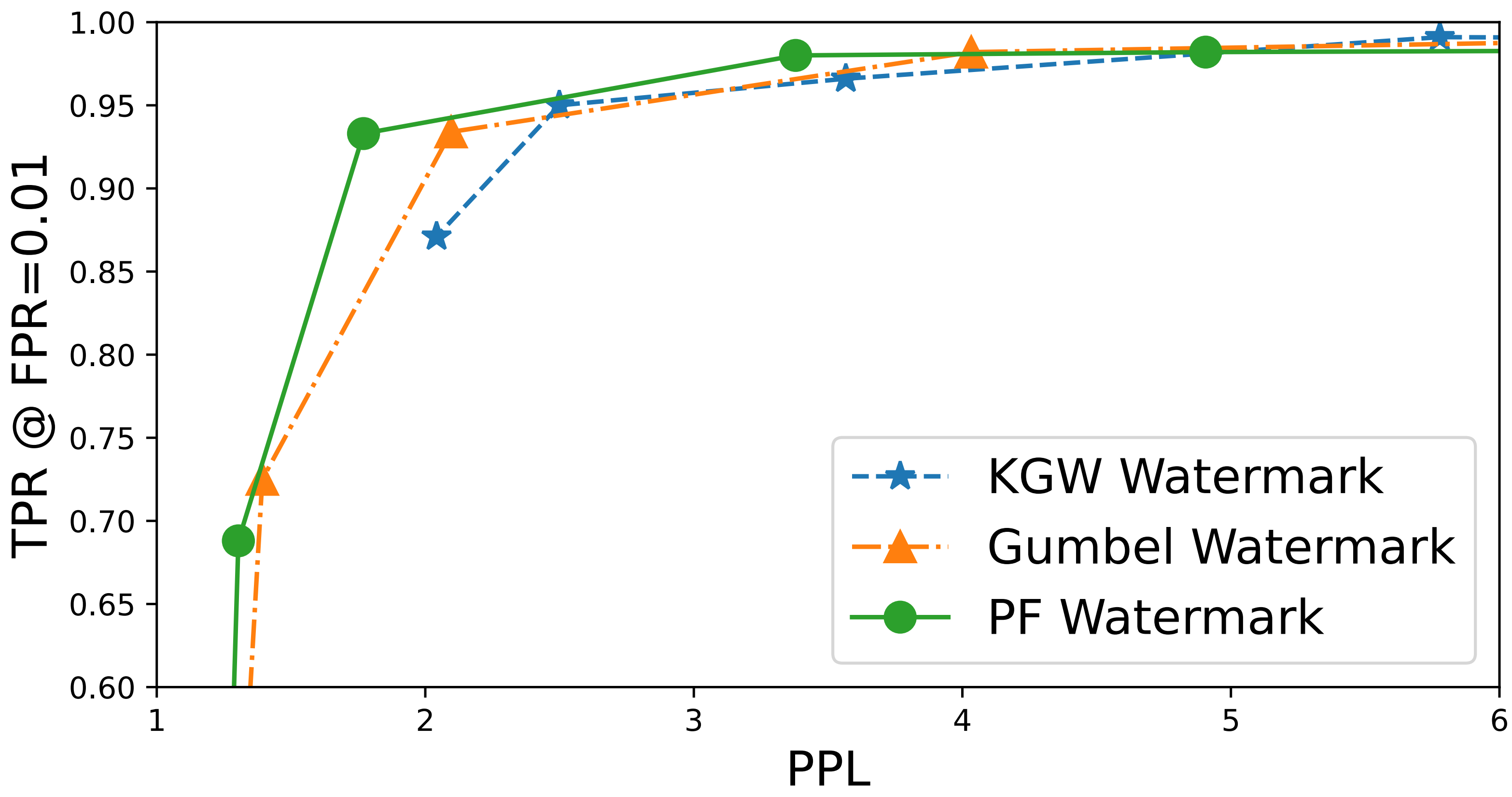
Zhao, Li, Wang. (2024) Permute-And-Flip: An Optimally Robust and Watermarkable Decoder for LLMs: https://arxiv.org/abs/2402.05864

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

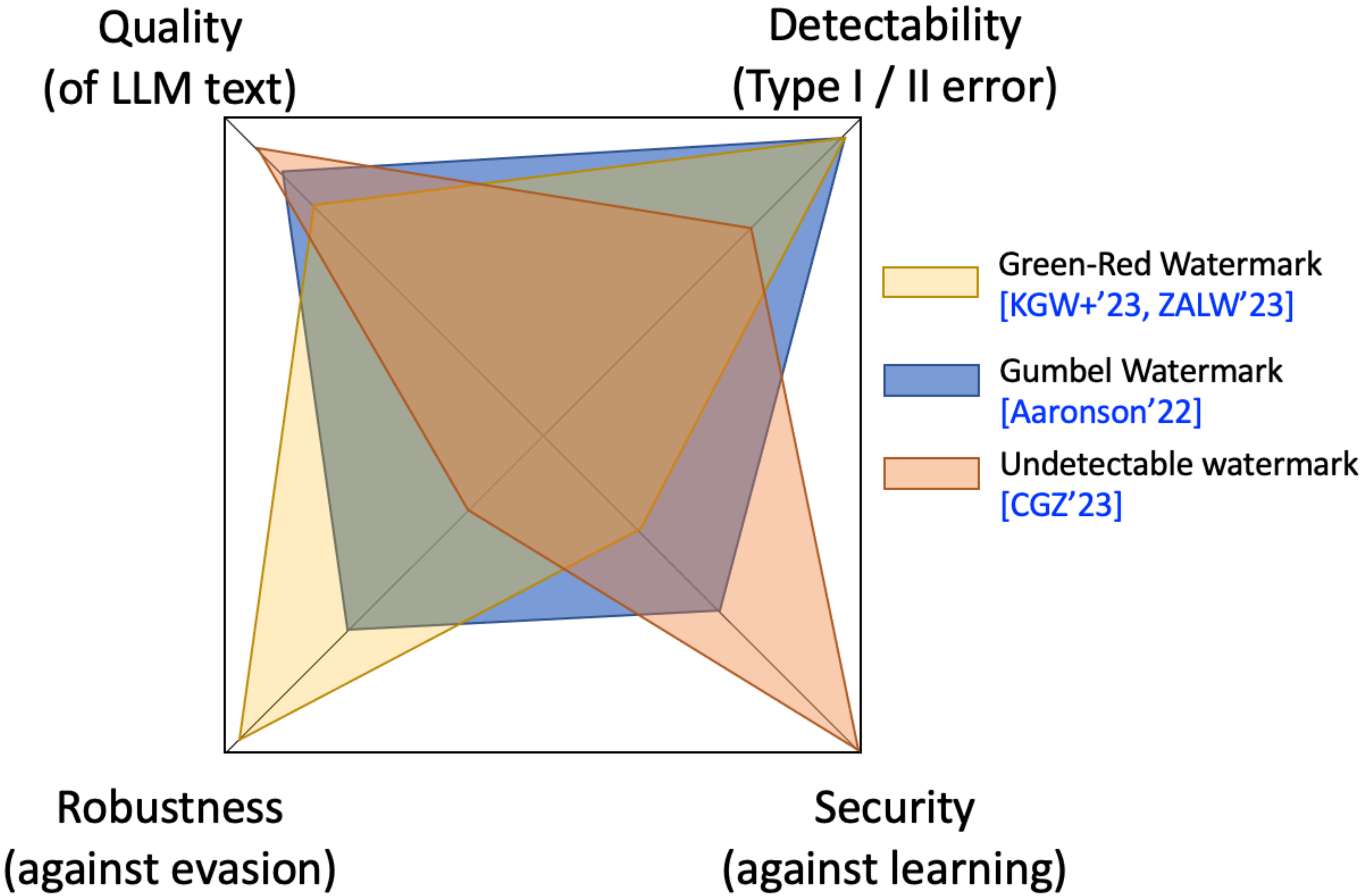
	Quality	Detectability	Robustness	Security
Green-Red WM [KGW+ 2023]	$\frac{\delta^2}{8}$ KL (ex post)	$O(\delta)$ per-high-entropy token	Robust to minor edits	n.a.
Unigram Green-Red [ZALW 2023]	$\frac{\delta^2}{8}$ KL (ex post)	$O(\delta)$ per-high-entropy token	More robust than m>1	n.a.
Gumbel WM [Aaronson 2022]	0-ex ante No ex post guarantee	Shannon entropy of the token	Robust to minor edits	n.a.
PF Watermark [ZLW 2024]	Better PPL-detectability curve than Gumbel	A different kind of Entropy per token	Robust to minor edits	n.a.
Undetectable WM [CGZ 2023]	0-ex ante No ex post guarantee	Shannon entropy of the token. (after a "burn-in")	Not robust to edits	Strong security via "undetectability

* All are model-agnostic and efficient.

Remainder of Part 2: Watermarking Text

- Formal Problem setup
- Popular Watermarking Schemes
  - Green-Red watermark
  - Gumbel watermark
  - Pointers to others
- Open problems and new directions

Optimal tradeoffs in LLM watermarks



Enhancing detectability

- Even for existing watermarks, are the current detection scores optimal in some sense?

[Submitted on 13 Dec 2023 (v1), last revised 6 Feb 2024 (this version, v3)]

Towards Optimal Statistical Watermarking

Baihe Huang, Hanlin Zhu, Banghua Zhu, Kannan Ramchandran, Michael I. Jordan, Jason D. Lee, Jiantao Jiao

We study statistical watermarking by formulating it as a hypothesis testing problem, a general framework which subsumes all previous statistical watermarking methods. Key to our formulation is a coupling of the output tokens and the rejection region, realized by pseudo-random generators in practice, that allows non-trivial trade-offs between the Type I error and Type II error. We characterize the

[Submitted on 1 Apr 2024]

A Statistical Framework of Watermarks for Large Language Models: Pivot, Detection Efficiency and Optimal Rules

Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, Weijie J. Su

Since ChatGPT was introduced in November 2022, embedding (nearly) unnoticeable statistical signals into text generated by large language models (LLMs), also known as watermarking, has been used as a principled approach to provable detection of LLM-generated text from its human-written counterpart. In this paper, we introduce a general and flexible framework for reasoning about the statistical efficiency of watermarks and designing powerful detection rules. Inspired by the

Either not model-agnostic or too much simplification.
Still along way to go!

Enhancing robustness

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

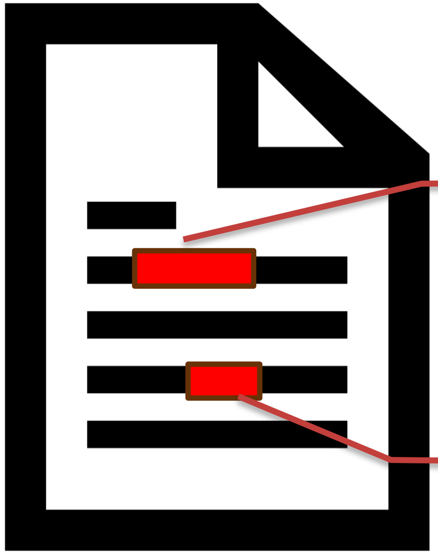
Bob generated a section using **ChatKitten**

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)

Is there a robustness-security tradeoff?

- Among Green-Red m-gram watermarks
  - Unigram watermark is the most robust, but also least secure

- Can we have a “undetectable” unigram watermark?

Pseudorandom Error-Correcting Codes

Miranda Christ, Sam Gunn

We construct pseudorandom error-correcting codes (or simply pseudorandom codes), which are error-correcting codes with the property that any polynomial number of codewords are pseudorandom to any computationally-bounded adversary. Efficient decoding of corrupted codewords is possible with the help of a decoding key.

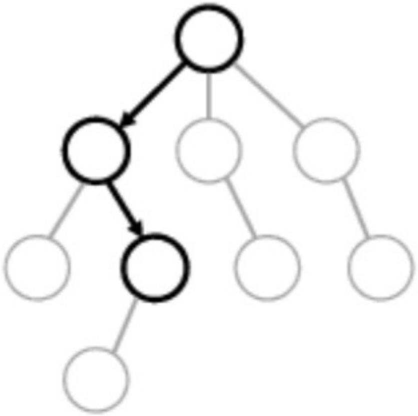
Nice progress, but still a bit far from practical.

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?



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)
   - 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.

Take a break for 30 minutes

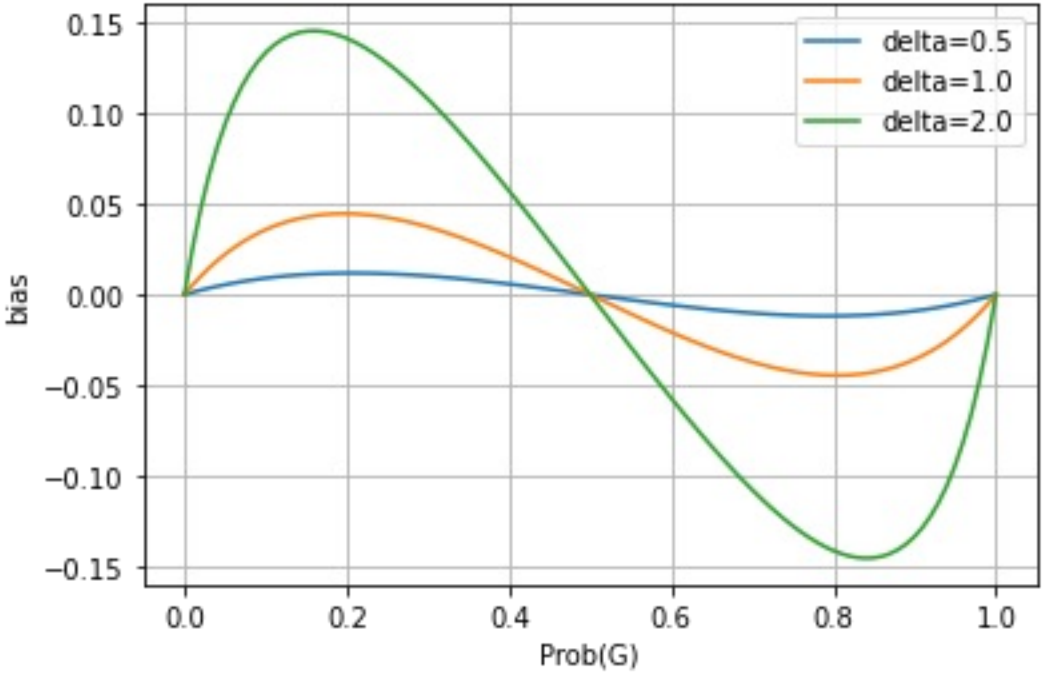
- Come talk to us for questions / comments!
- Please be back for “Part 3 Watermark for model protection”

Do we know Green-Red WM is NOT distortion-free?

- "Distortion-free" is *ex ante* $\mathcal{M}(Input) \sim \widehat{\mathcal{M}}(Input)$

  Over the distribution of the key, i.e., $E_k[\hat{p}] = p$

Let's plot $E_k[\hat{p} \mid p(G)] - p$ against $p(G)$ for different δ



- Unbiased when p(G) = 0.5
- also unbiased when p(G) = 0 or 1

- $\delta = \mathbf{0.5}$ => Bias < 0.015.

Not unbiased but also not very biased.