

# Watermarking for Large Language Models

## Part II: Text Watermarking (continue...)



Xuandong Zhao  
UC Berkeley



Yu-Xiang Wang  
UC San Diego



Lei Li  
CMU

# 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 can any watermarking schemes do?

Many of these questions require theory to answer.

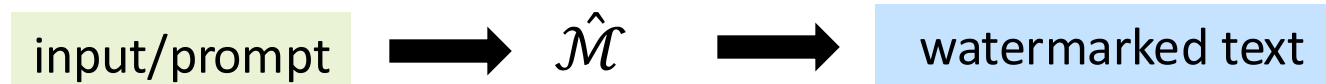
# Remainder of Part 2: Watermarking Text

- Four performance metrics
- Theory for Popular Watermarking Schemes
  - Green-Red watermark
  - Gumbel watermark
  - Pointers to others

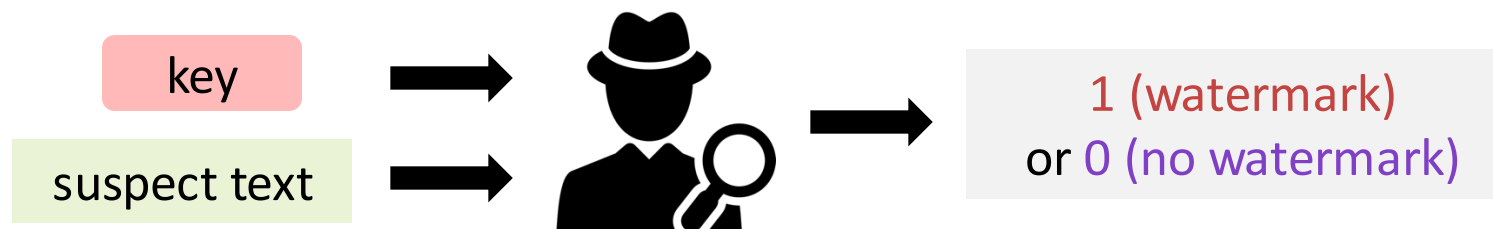
# Recall: An LM Watermarking Scheme has two components

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

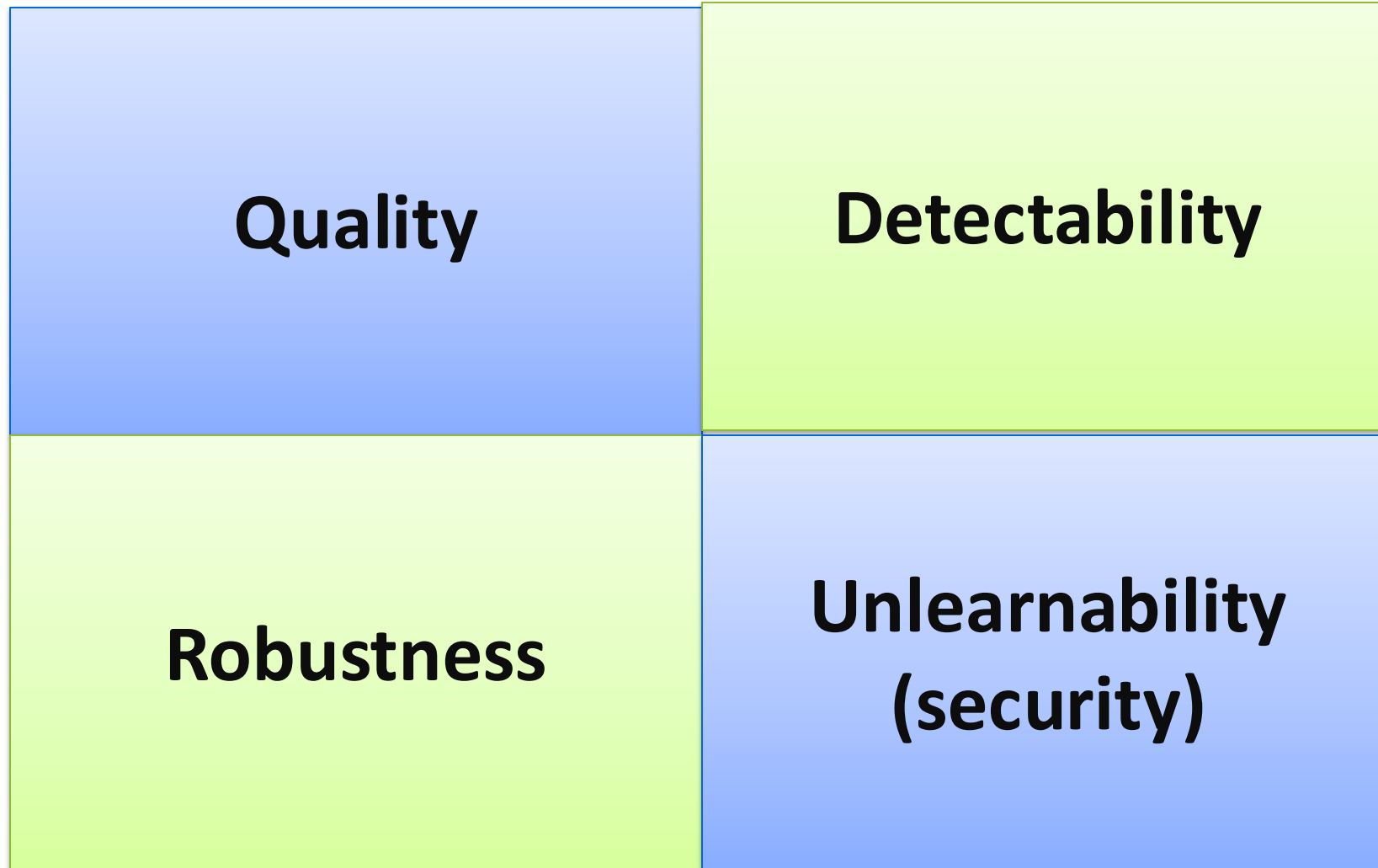
$$\mathbf{Watermark}(\mathcal{M}) \rightarrow (\hat{\mathcal{M}}, k \text{ key})$$



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



# Four key metrics of a watermarking scheme

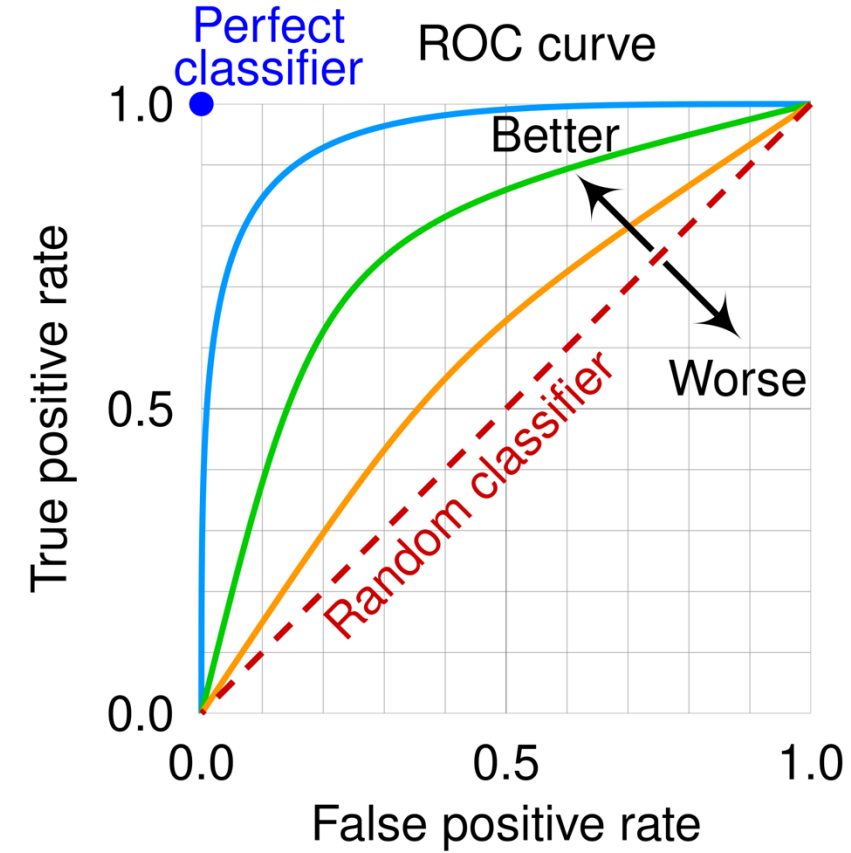


# Quality is often defined through “distortion”

- How close in distribution are  $\mathcal{M}$  and  $\hat{\mathcal{M}}$ :
  - Which metric to use? TV, KL-div, Renyi? Statistical or computational distinguisher?
  - Which distribution? One-token / whole sequence / any polynomial number of sequences
  - (ex post vs ex ante) when  $\hat{\mathcal{M}}$  is random, is the quality guarantee for every realized  $\hat{\mathcal{M}}$  or over the distribution of  $\hat{\mathcal{M}}$

# Detectability: A hypothesis testing view of LLM watermarks

- $H_0$ : The suspect text  $y$  is NOT generated from  $\hat{\mathcal{M}}$   
e.g., “ $y$ ” is written by a human.  
e.g., “ $y$ ” is generated by  $\mathcal{M}$ .
- $H_1$ : The suspect text is generated from  $\hat{\mathcal{M}}$   
**A very broad “Null” and a very specific “Alternative”**
- **Theory** : Can we control FPR. Can we prove high power? Are the tradeoff optimal?



# FPR in watermarking is **distribution-free!**

FPR = Probability of “Detector” making a mistake for **any fixed Input.**

**Randomness is over the secret key only!**

Different from ML experiments:

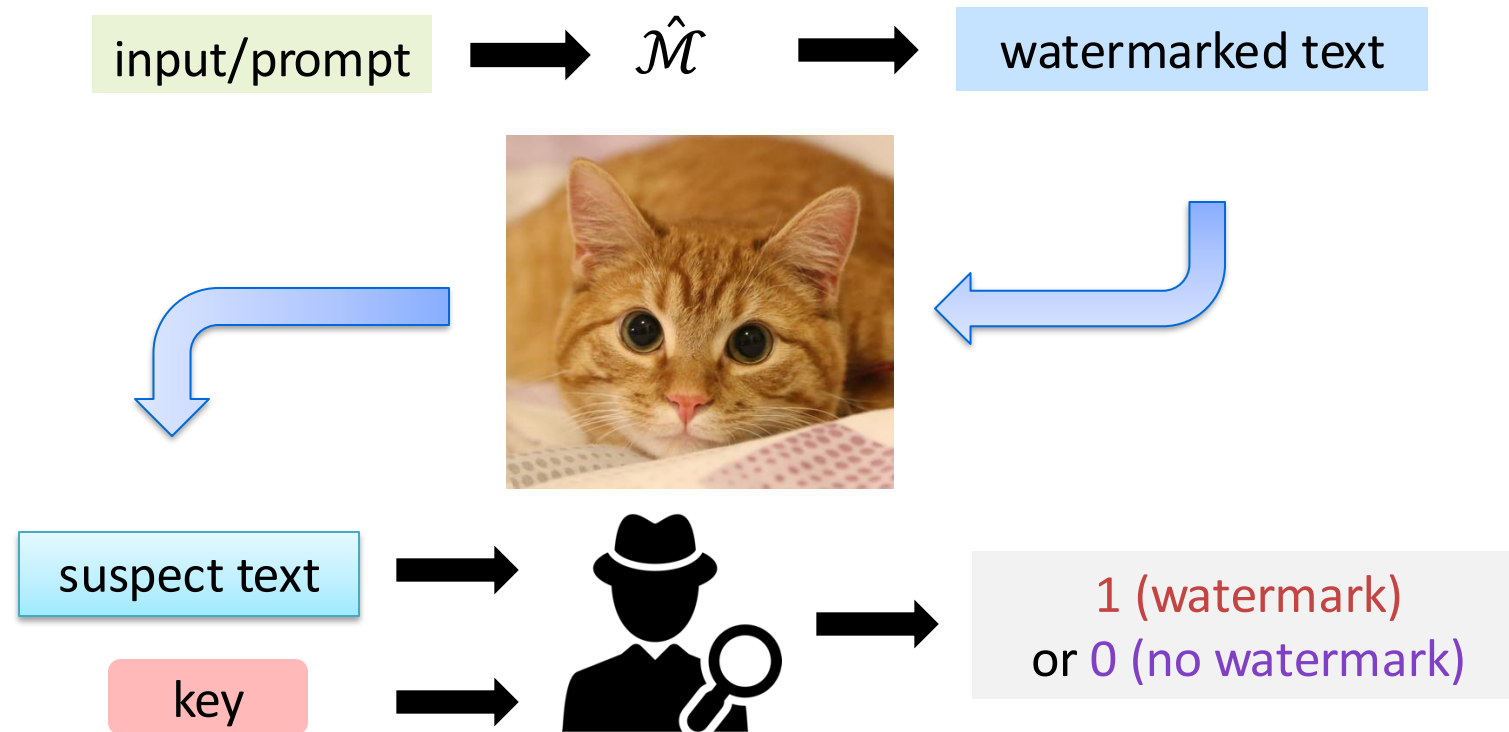
$$\text{FPR} = \# \text{ of False Positive} / \text{Total} \# \text{ of Negative Examples}$$

Implicitly, this FPR is specific to the data distribution  $P(\text{input } x \mid \text{label of } x \text{ is “-”})$



# Robustness is defined through a particular attack family, e.g., edits, cropping, shuffling.

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



# Unlearnability (Security): How difficult is it for an attacker to learn the secret key?

- What can you do if you 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  $\hat{\mathcal{M}}$  are computationally indistinguishable.

(Still in the early stage of research... the community is yet to converge to a good definition)

# 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.

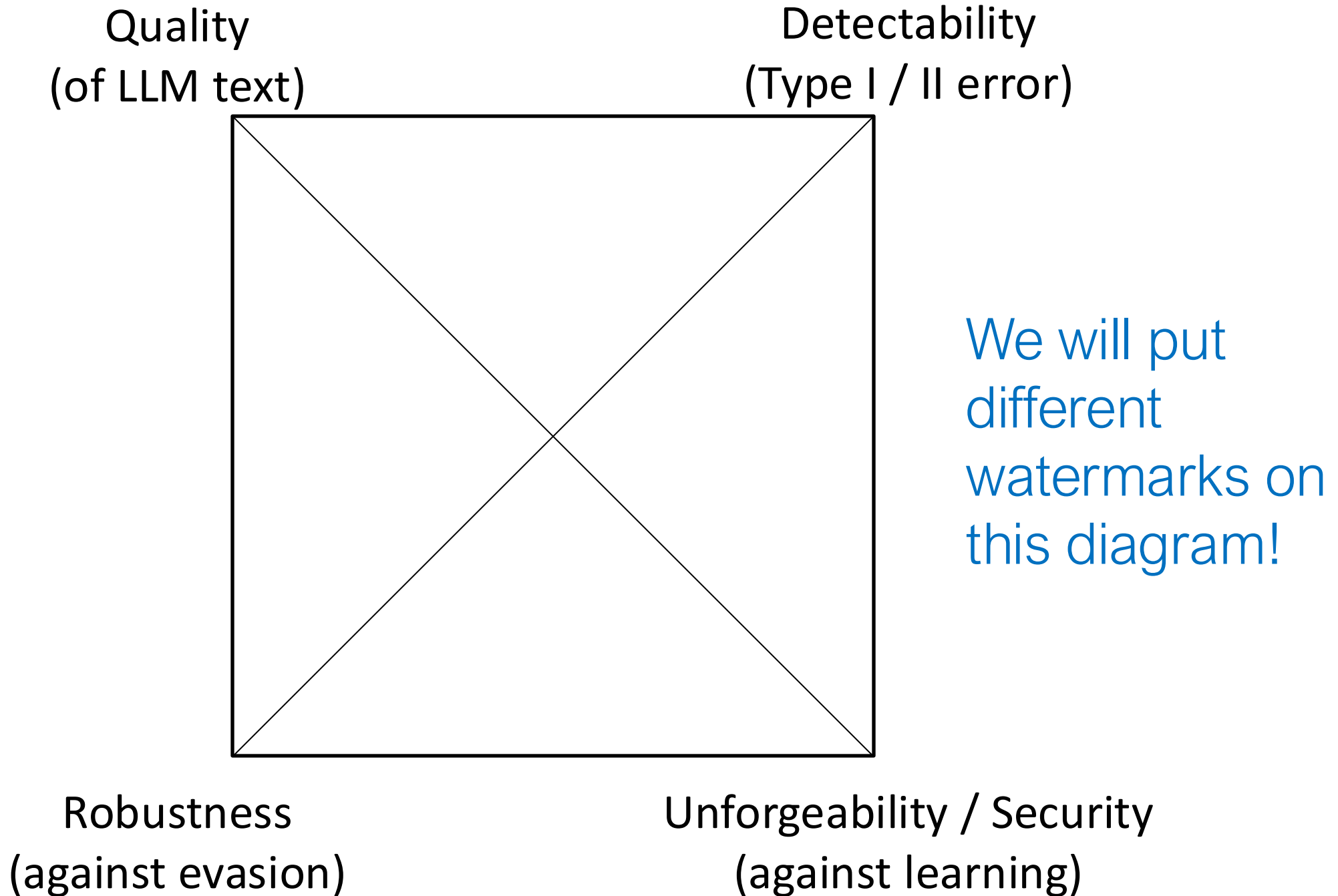
They are nuanced and often case-by-case!

# Remainder of Part 2: Watermarking Text

- Four performance metrics
- Theory for Popular Watermarking Schemes
  - Green-Red watermark
  - Gumbel watermark
  - Pointers to others

# 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](#)) ([Christ and Gunn, 2023](#))



# Caveat: definitions are often different and often not directly comparable

- The categorization is thus largely qualitative (backed by empirical evidence).
- I will explain the algorithmic ideas and the theoretical claims (on a high-level) as we discuss watermarking methods.
- For details: please read the recent survey by Zhao, Gunn, Christ et al. (2024) <https://arxiv.org/pdf/2411.18479v1>



# Quality guarantee of Green-Red Watermark

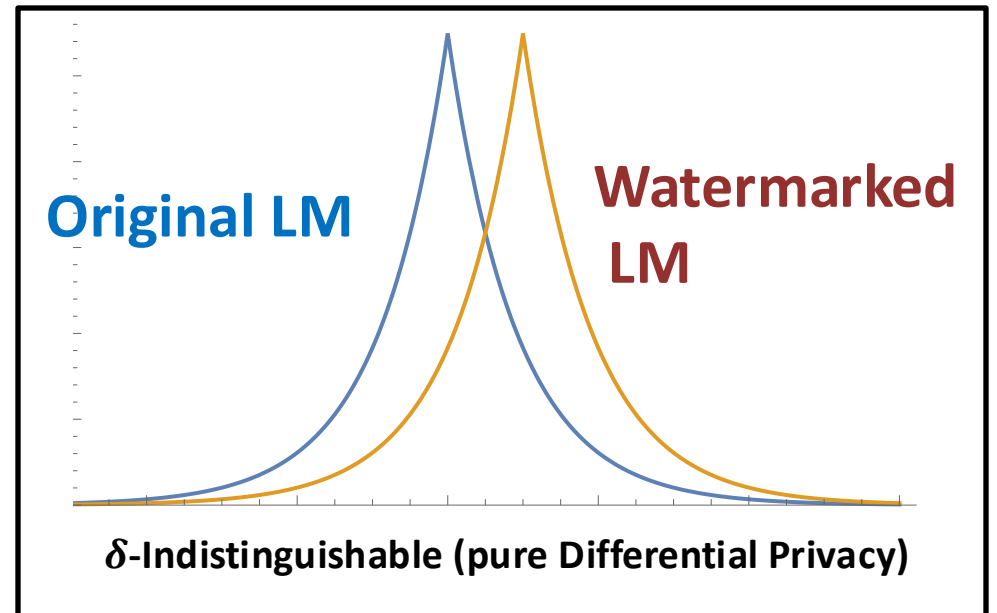
(Kirchenbauer et al. 2023; Zhao et al. 2023)

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

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

**Theorem:** Any prompt, any prefix text, any key (green list), Renyi-Divergence of order  $\alpha$  :

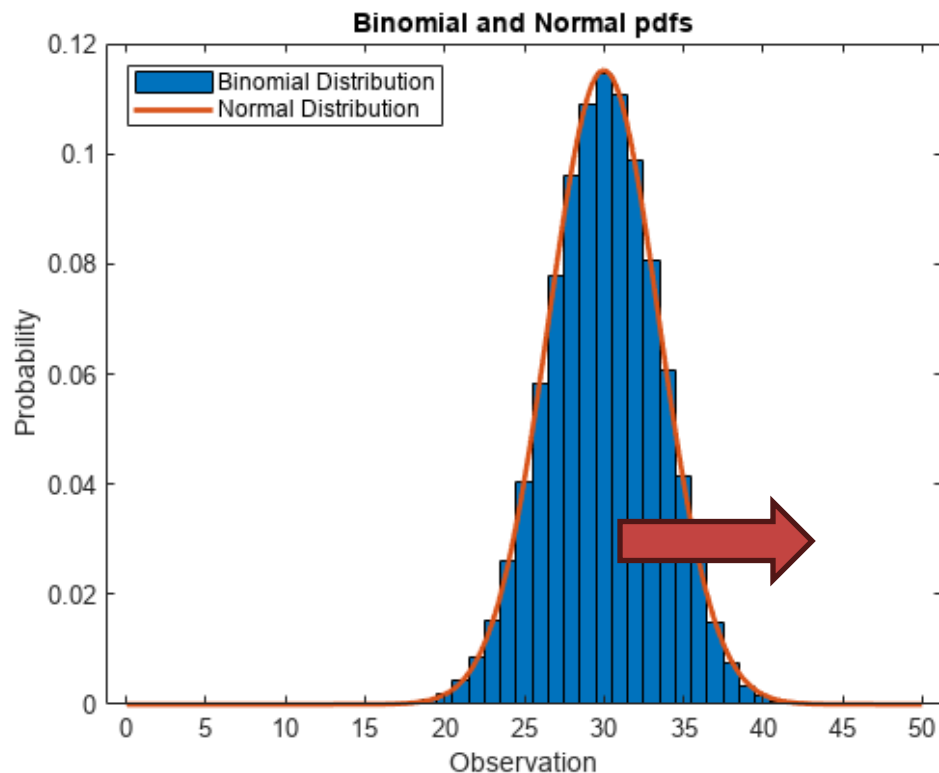
$$D_\alpha(\mathcal{M} || \hat{\mathcal{M}}) \leq \min\left\{\delta, \frac{\alpha\delta^2}{8}\right\}$$



# 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  $1(y_i \in G_i) \sim \text{Ber}(\gamma)$  independently)



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

**When watermarked**, the distribution shifts to the right by roughly  $e^\delta$  multiplicatively.

# Recall: How is the *Green* list generated?

- *Randomly* selecting  $\gamma$  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  $\mathbf{y}$  be independent to the secret key (i.e., the green list).

$$z_{\mathbf{y}} = \mathcal{O}(\sqrt{\log(1/\alpha)}) \text{ w.p. } 1 - \alpha$$

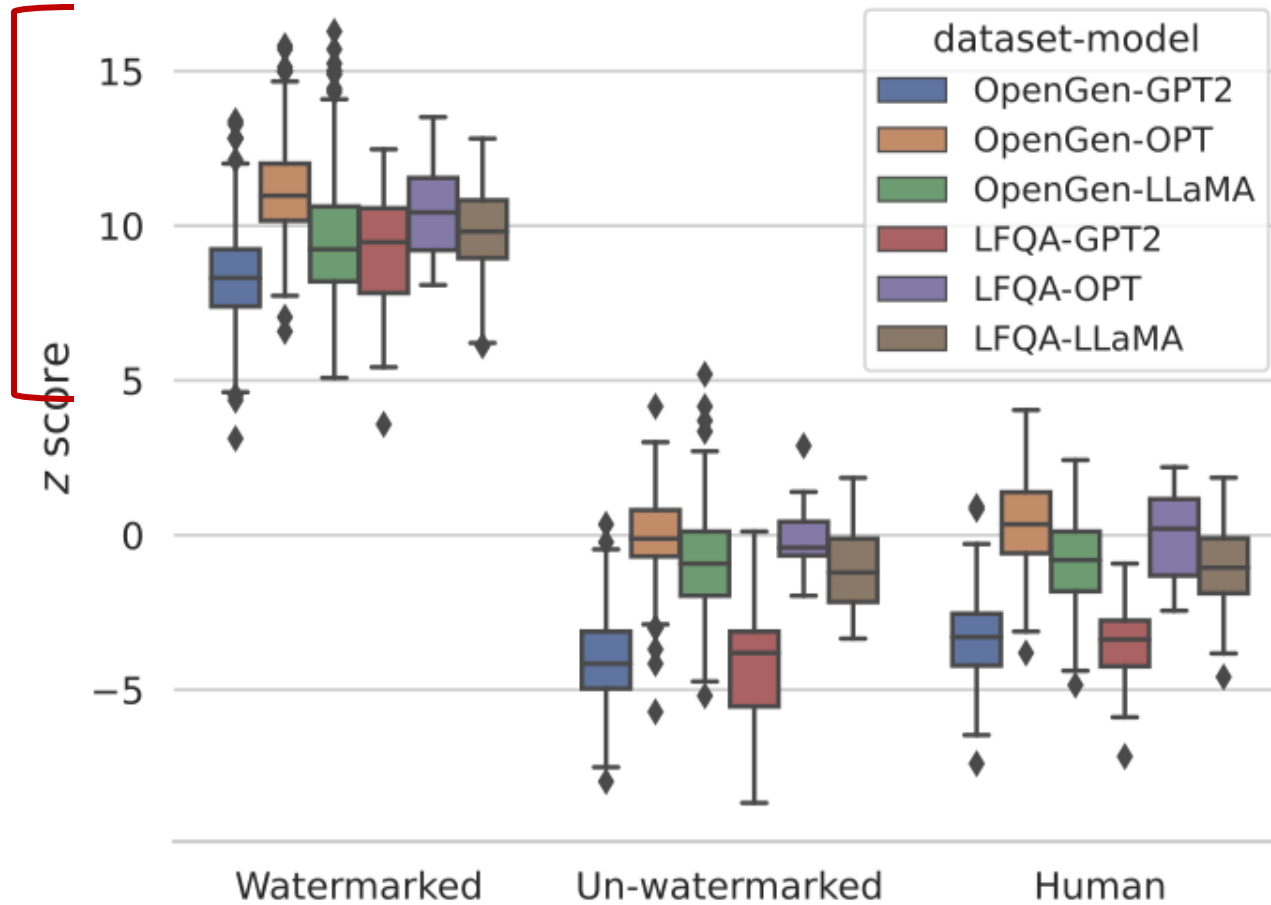
where  $V$  and  $C_{max}$  measure the **diversity** of the text. If unique, then  $Z=1$  and  $C_{max} = 1$

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

$$z_{\mathbf{y}} = \Omega(\kappa(e^\delta - 1)\sqrt{n}) \text{ w.p. } 1 - \beta$$

# Detection guarantees of Unigram WM Illustrated

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



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

H1: Alternative

H0: "Null"

# Unigram watermark is robust to edits!

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

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

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  $\delta$  determines quality-detectability tradeoff.

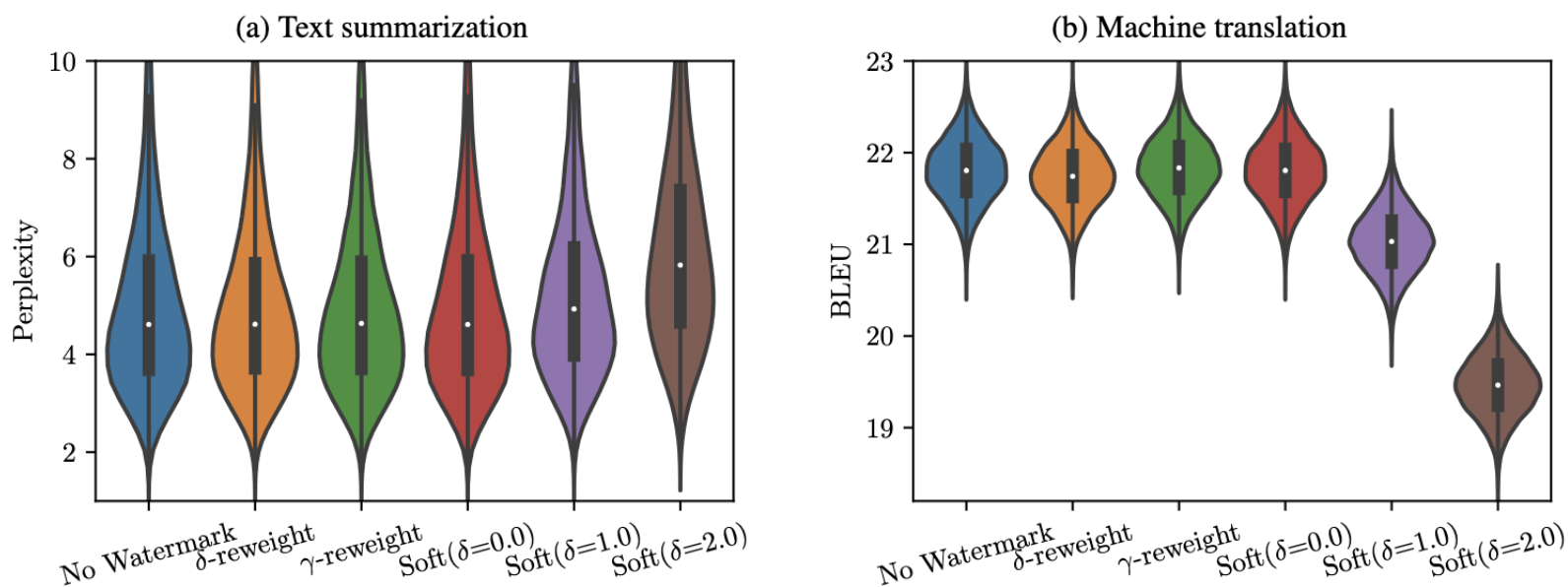
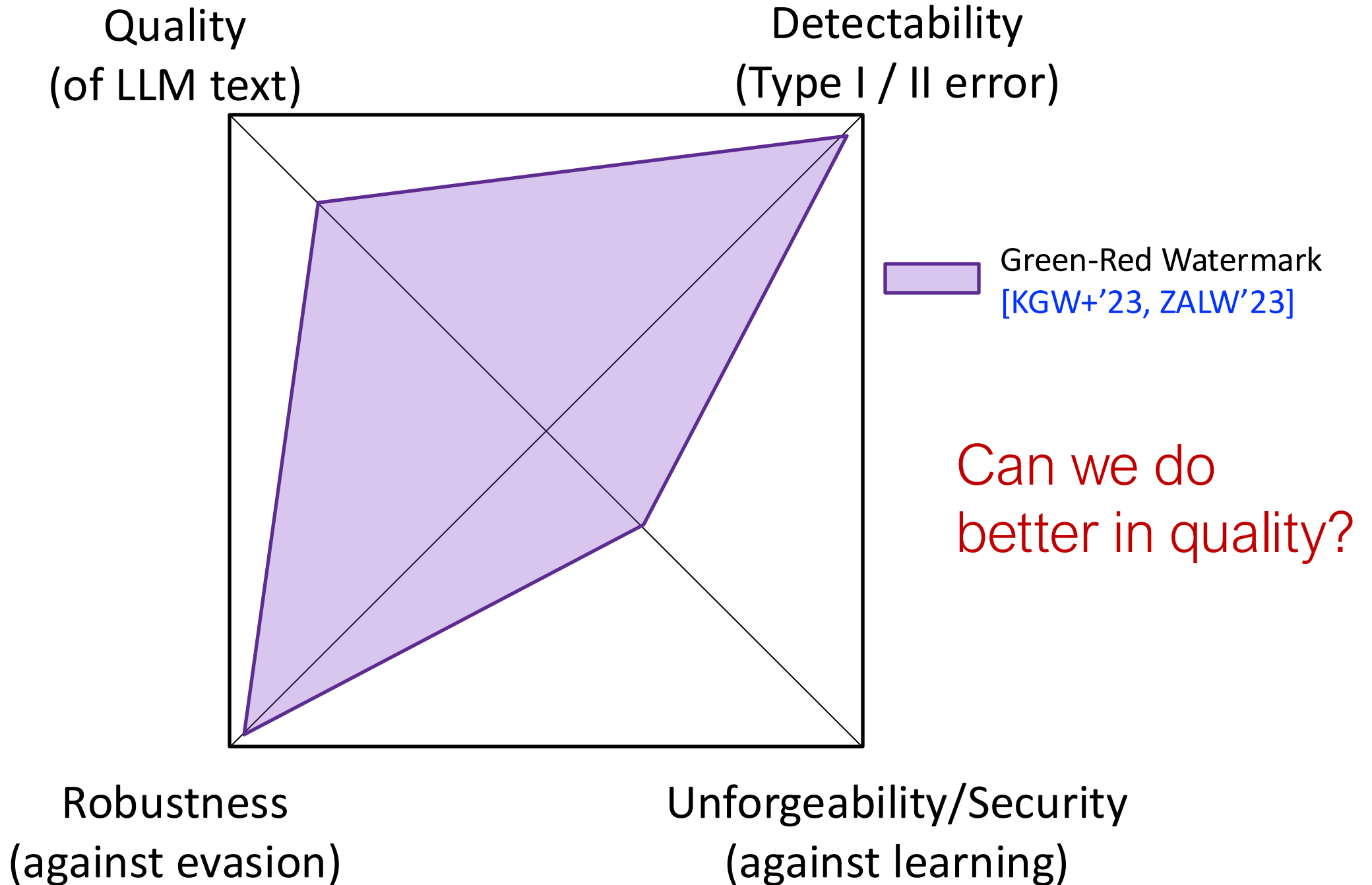


Figure 3: Distribution of perplexity of output for TS and BLEU score for MT.

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



# 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)

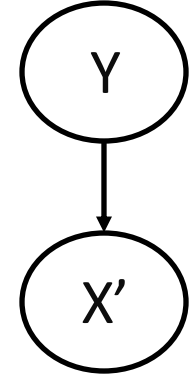
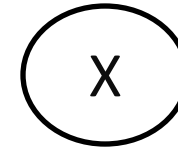
Unbiased WM (Hu et al ,2023)

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

# 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:

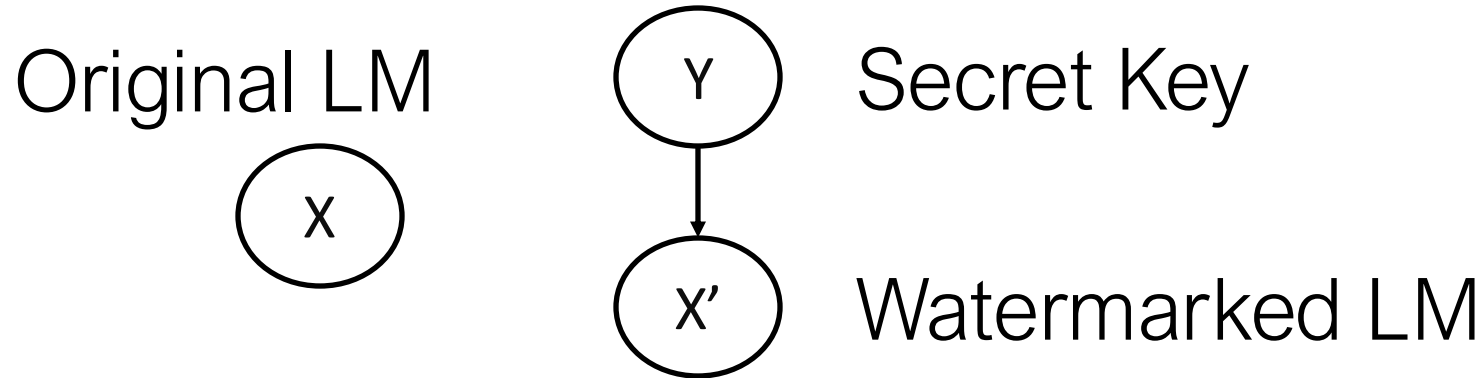
$X \sim X'$  marginally (i.e., they are identically distributed)

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  $X' \sim X$ ? Come to me with your idea.

# Gumbel-Softmax trick and Gumbel Watermark

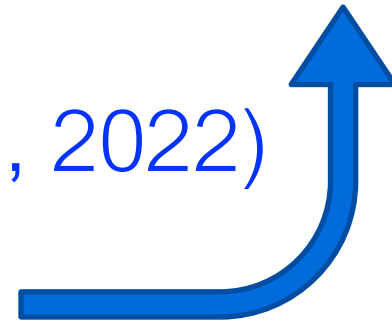
- Gumbel-Softmax trick ([Gumbel, 1948](#))

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

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

- 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

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

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

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

**No watermark**

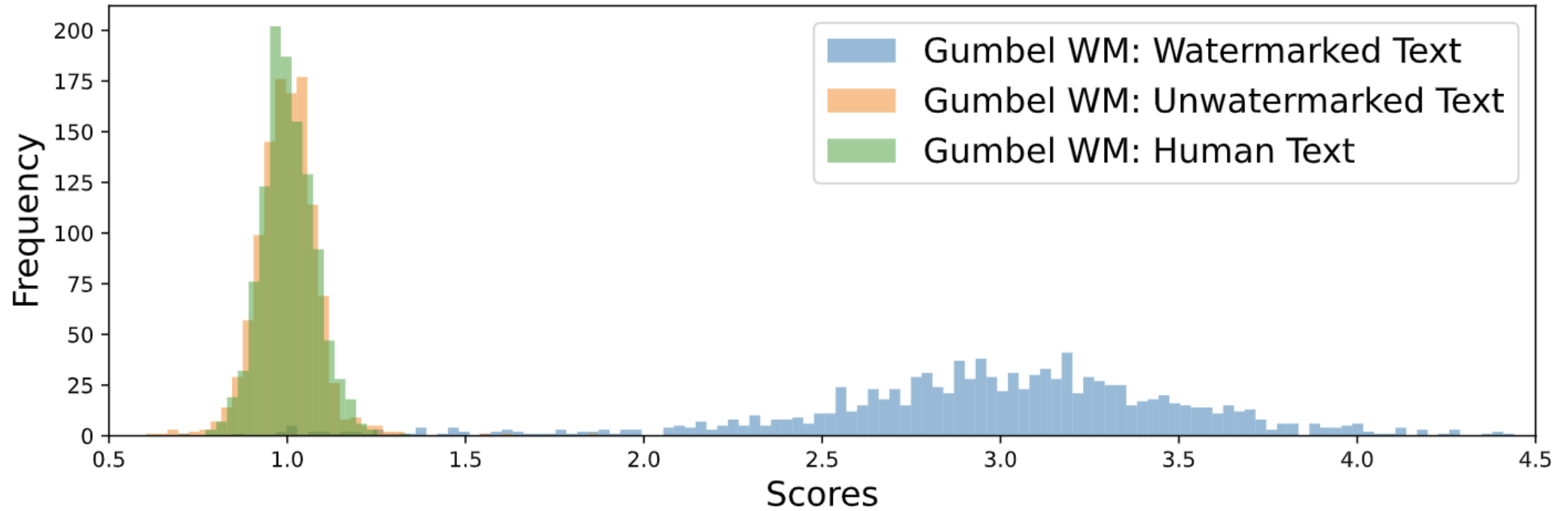
$$\mathbb{E}[\text{TestScore}(y_{1:n})] = n - m$$

**Watermarked**

$$\begin{aligned} \mathbb{E}[\text{TestScore}(y_{1:n})] &= \sum_{t=m+1}^n \mathbb{E} \left[ \sum_{y \in \mathcal{V}} p_t(y) H_{\frac{1}{p_t(y)}} \right] \\ &\geq (n - m) + \left( \frac{\pi^2}{6} - 1 \right) \sum_{t=m+1}^n \mathbb{E} [\text{Entropy}[p_t(\cdot)]] . \end{aligned}$$



# Detection score of Gumbel WMs in practice



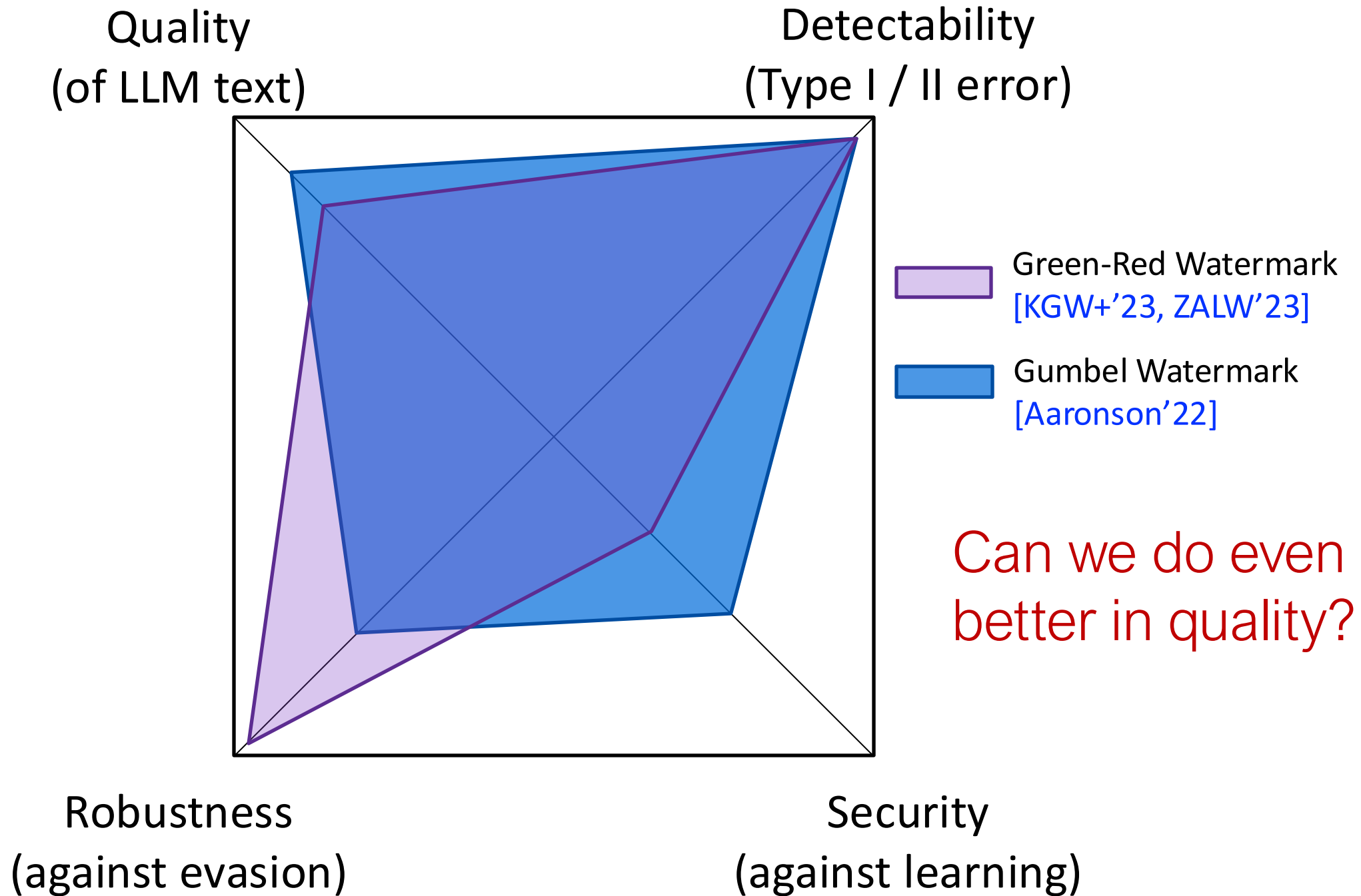
# Robustness of Gumbel WM is not bad

- Not “unigram WM” type robust, but still quite robust

Setting	Method	AUC	1% FPR		10% FPR	
			TPR	F1	TPR	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

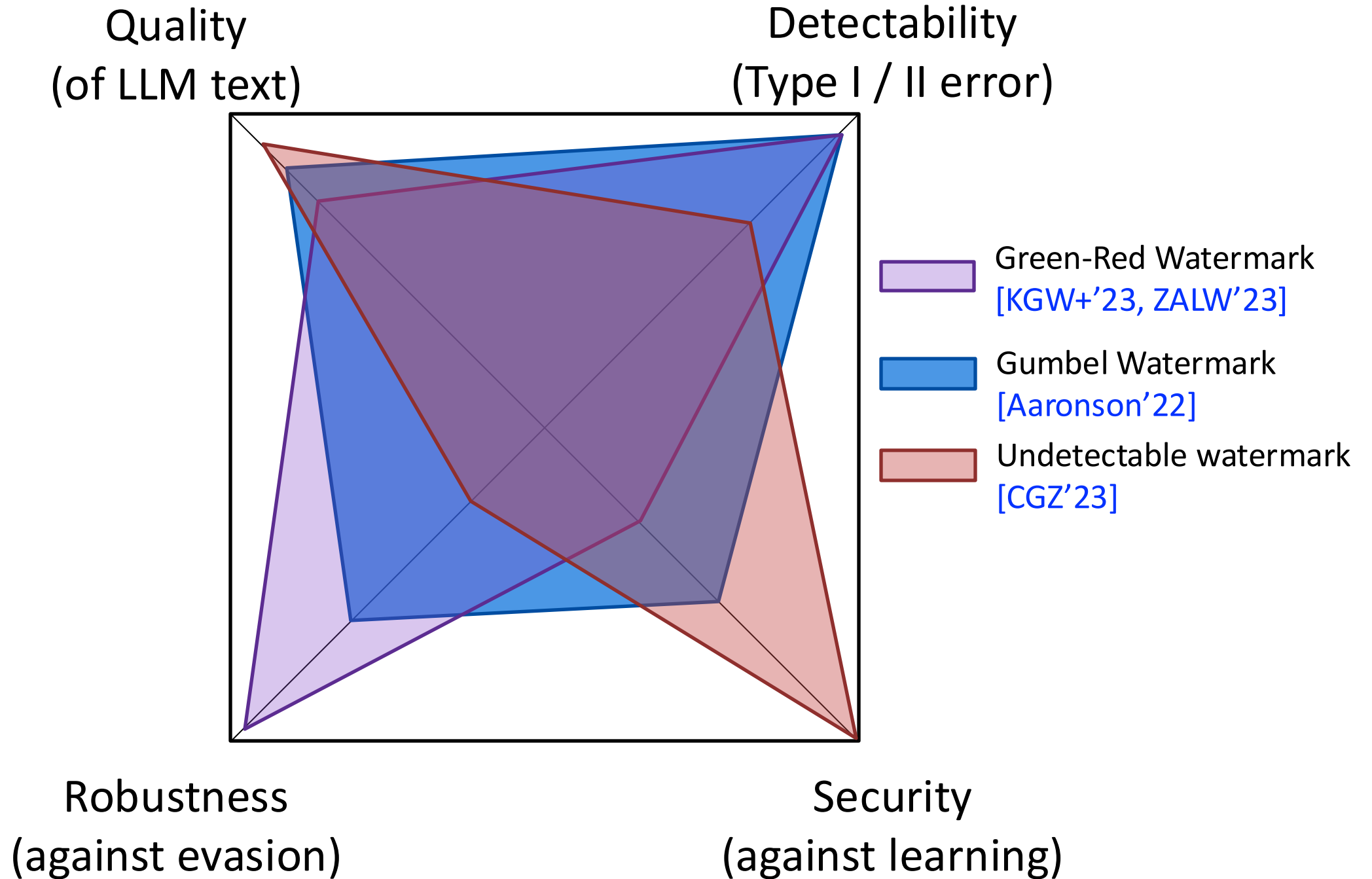
DIPPER-1  
DIPPER-2  
are “paraphrasing  
attacks”

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



# 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! ”



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

- Watermark by Pseudo-Random Error Correcting Code ([Christ and Gunn, 2024](#))

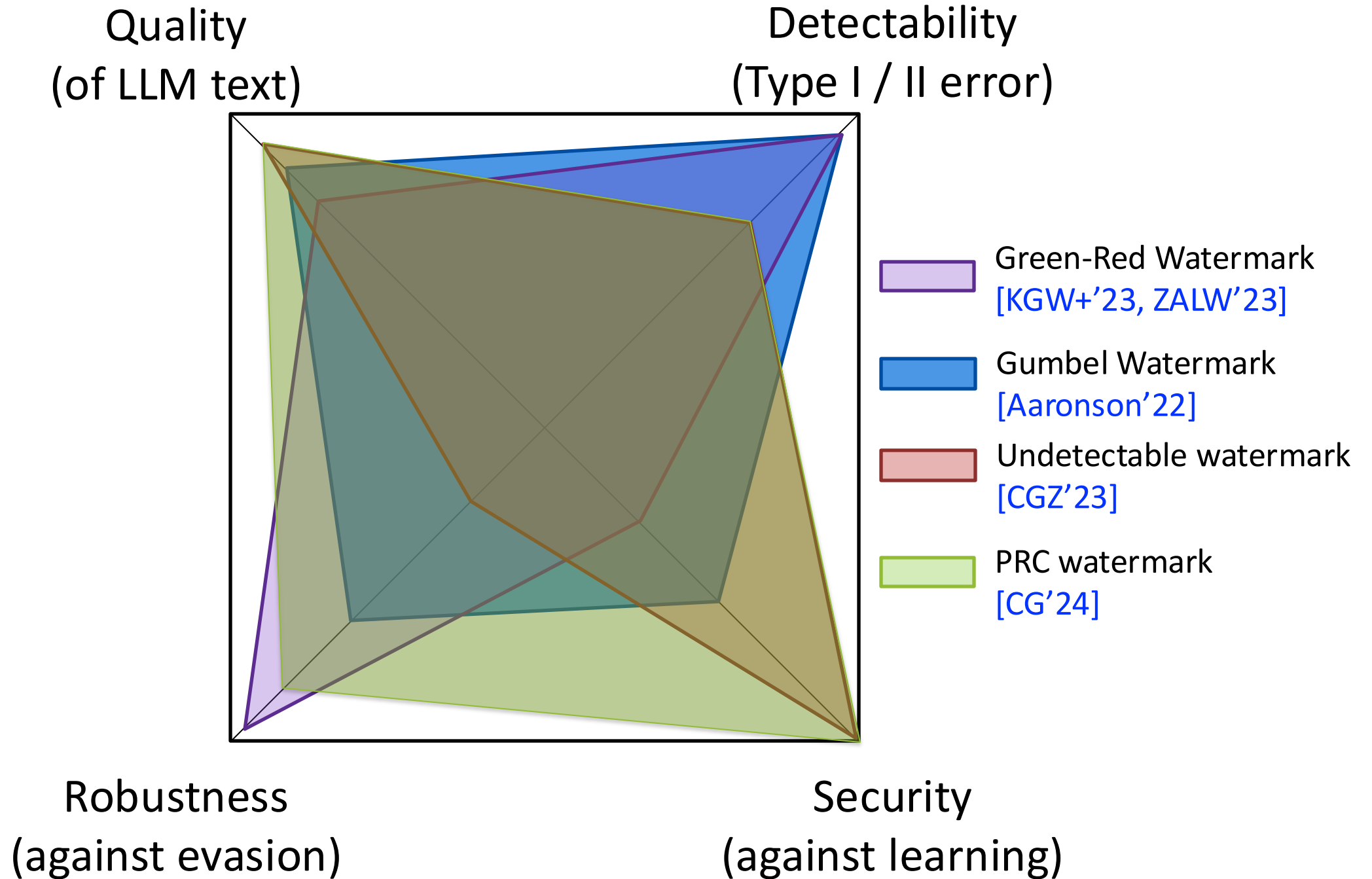
Key: [PRC, and  $L$  messages  $a_i$  with length  $m$ ]

“ $W_1, \dots, W_m, \underbrace{W_{m+1}, \dots, W_{2m}}_{\text{Watermark with PRC.encode}(a_2)}, W_{2m+1}, \dots, W_{3m}, \underbrace{W_{3m+1}, \dots, W_{Lm}}_{\text{Watermark with PRC.encode}(a_3)}$ ”

Watermark with  $\text{PRC.encode}(a_2)$

- Detector: check for  $\text{PRC.decode}(\text{all subsequences})$  against all messages.

In some sense getting the robustness of “Unigram watermark” and “Undetectability” of CGZ simultaneously.



# Are “distortion-free” (and “undetectable”) watermarks always better than Green-Red?

- Green-Red watermark leverages the watermark strength parameter  $\delta$  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](#))

$$y_t \sim \text{Softmax} \left( \frac{u_t(y)}{T} \right) \iff \begin{aligned} y_t &= \arg \max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + G_t(y) \\ G_t(y) &\sim \text{Gumbel}(0, 1) \text{ i.i.d} \end{aligned}$$

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

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

**ReportNoisyMax from Differential Privacy.**

# Permute-and-Flip Watermark

- Gumbel-Watermark (Aaronson, 2022)

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

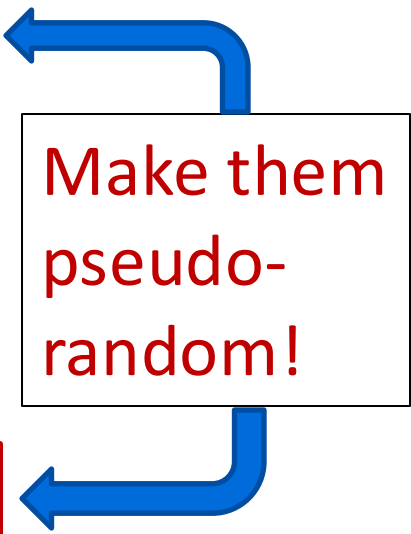
$$G_t(y) \sim \text{Gumbel}(0, 1) \text{ i.i.d}$$

- PF-Watermark (Ours)

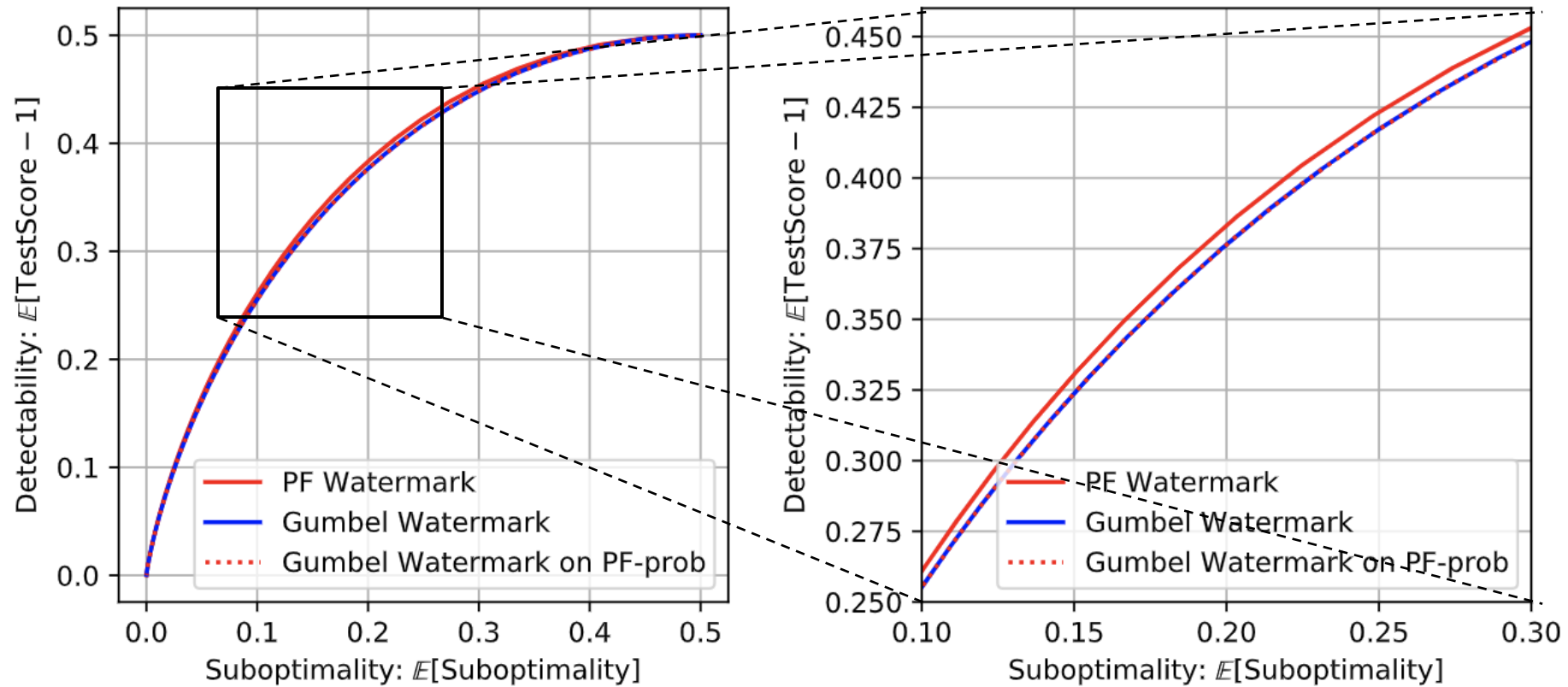
$$y_t = \arg \max_{y \in \mathcal{V}} \frac{u_t(y)}{T} + E_t(y).$$

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

Make them  
pseudo-  
random!

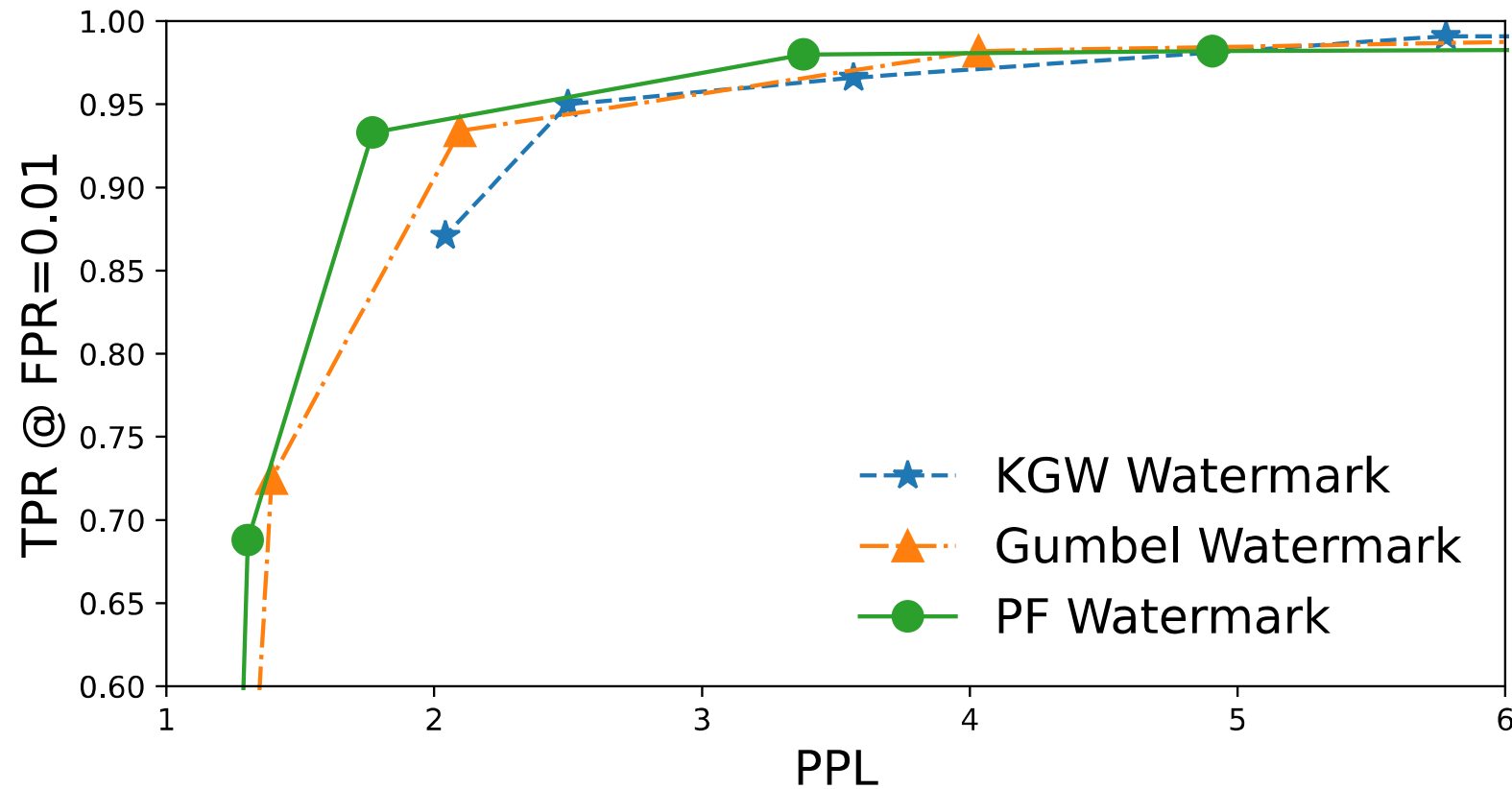


# Plotting detectability against suboptimality as we adjust $T$



**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 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 edits	n.a.
PF Watermark [ZLW 2024]	Better PPL-detectability curve than Gumbel	A different kind of Entropy per token	Robust to edits	n.a.
Undetectable WM [CGZ 2023]	Undetectable No ex post guarantee	Empirical entropy of the token. (after a “burn-in”)	Not robust to edits	Strong security via “undetectability
PRC WM [CG 2024]	Undetectable No ex post guarantee	Empirical entropy of the token. (after a “burn-in”)	Robust to edits	Strong security via “undetectability

\* All are model-agnostic and efficient.

# 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](#).

# Supplementary slides

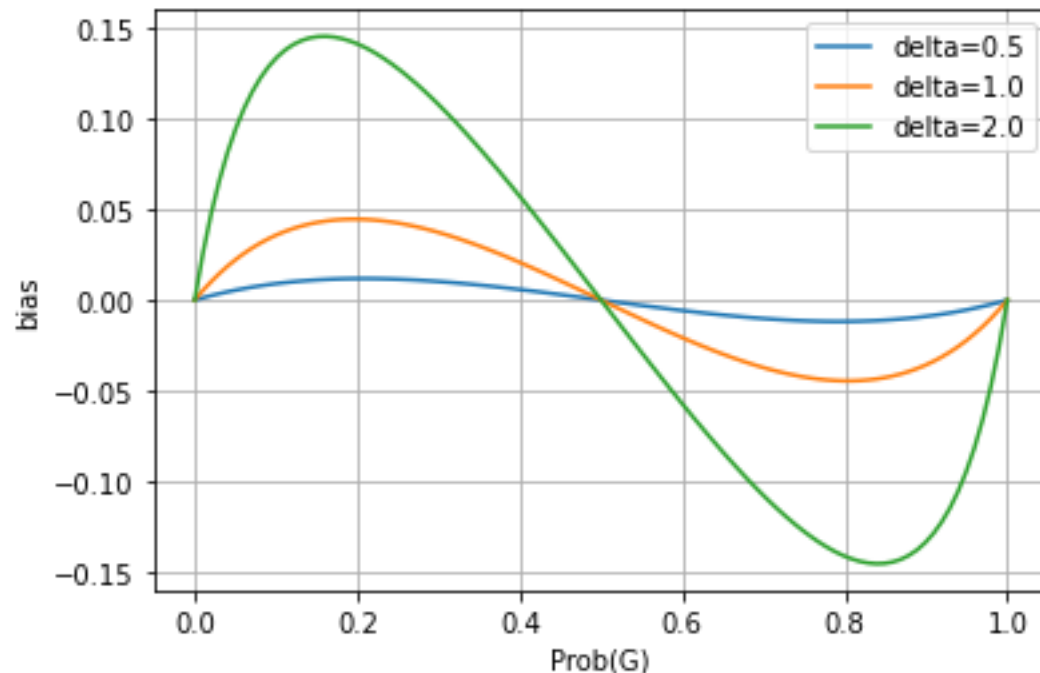


# Do we know Green-Red WM is NOT distortion-free?

- “Distortion-free” is *ex ante*  $\mathcal{M}(Input) \sim \hat{\mathcal{M}}(Input)$

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  $\delta$



- Unbiased when  $p(G) = 0.5$
- also unbiased when  $p(G) = 0$  or  $1$
- $\delta = 0.5 \Rightarrow \text{Bias} < 0.015$ .  
Not unbiased but also not very biased.

# One improvement to Green-Red watermark

- Pareto-optimal trade-off between PPL and Detectability
- Simple modification to Green-Red WM

If “worth it”: Add watermark with  $\delta = \infty$ .

If “not worth it”: Do not add watermark at all!

Wouters, Bram. "Optimizing watermarks for large language models." *ICML-2024*.