Robust Watermark Detection and Efficient Proportion Estimation for Human-Edited Watermarked Text

Xiang Li

University of Pennsylvania

2025 ICSA Applied Statistics Symposium
June 16, 2025

Why should we care who wrote the text?

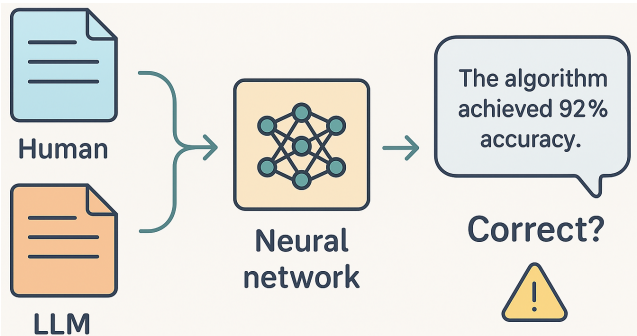
LLMs are now used almost everywhere—but this raises concerns about authorship, accountability, and trust.

Why should we care who wrote the text?

LLMs are now used almost everywhere—but this raises concerns about authorship, accountability, and trust.

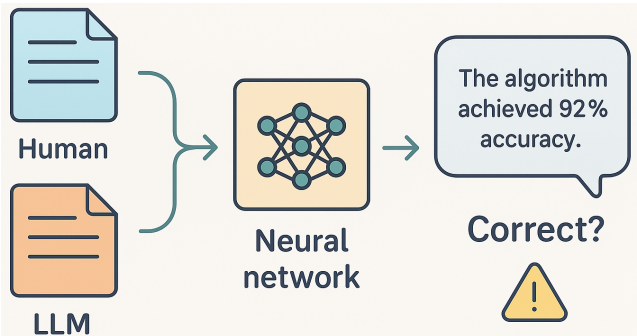
- **Education**: Students may use LLMs for homework or essays—teachers want to know who did the work.
- **Science**: Papers or reviews may be (partly) machine-written—can we trust the content?
- **Media**: Fake news or spam can be mass-produced—platforms need to detect AI-written content.
- **Art and writing**: Readers and publishers care whether a story or article came from a human.
- **Transparency**: If LLMs are used to help write something, readers may want to be informed.

An intuitive solution



- Train classifiers [GPTZero, 2023, ZeroGPT, 2023, ...].

An intuitive solution



- Train classifiers [GPTZero, 2023, ZeroGPT, 2023, ...].
- These methods are inaccurate, unreliable [Weber-Wulff et al., 2023], and often biased [Krishna et al., 2024, Sadasivan et al., 2023, Liang et al., 2023].
- Worse, as AI models evolve, LLM-generated text increasingly resembles human-written text!

Watermarking is a provable and practical solution!

Key insight

LLMs are probabilistic machines, and we control how they generate texts.

Watermarking is a provable and practical solution!

Key insight

LLMs are probabilistic machines, and we control how they generate texts.

A watermark embeds subtle and recoverable statistical signals into LLM-generated texts [Kirchenbauer et al., 2023].

- Creates a statistical dependency between the visible text and a hidden information.
- Unlikely to appear in human-written text.
- Applies not only to text but also to images, tables, and other data modalities.

An active research area with practical importance

***A Zoo of Watermarking Schemes* (since Jan 2023):**

[Kirchenbauer et al., 2023, Aaronson, 2023, Kuditipudi et al., 2024, Zhao et al., 2024b, Christ et al., 2024, Wu et al., 2023, Hu et al., 2024, Kirchenbauer et al., 2024, Zhao et al., 2024a, Xie et al., 2024, Fu et al., 2024, Dathathri et al., 2024, Gloaguen et al., 2025, Abdalla and Vershynin, 2025, ...].

An active research area with practical importance

A Zoo of Watermarking Schemes **(since Jan 2023):**
[Kirchenbauer et al., 2023, Aaronson, 2023, Kuditipudi et al., 2024, Zhao et al., 2024b, Christ et al., 2024, Wu et al., 2023, Hu et al., 2024, Kirchenbauer et al., 2024, Zhao et al., 2024a, Xie et al., 2024, Fu et al., 2024, Dathathri et al., 2024, Gloaguen et al., 2025, Abdalla and Vershynin, 2025, ...].

- **Open-source toolkits** have been developed to support research [Pan et al., 2024].
- **Large-scale empirical studies** benchmark watermarking methods [Fernandez et al., 2023, Gloaguen et al., 2025, Piet et al., 2023, ...].
- **Industry commitment**: OpenAI, Google, Meta, and others pledge to watermark AI-generated content.

Questions we study

- Previous study often assumes full detection where the text is either whole human-written or fully LLM-generated.
- A student might modify the text generated by an LLM—either to personalize it or to avoid detection.

Questions we study

- Previous study often assumes full detection where the text is either whole human-written or fully LLM-generated.
- A student might modify the text generated by an LLM—either to personalize it or to avoid detection.

Core questions

- Given a potentially modified text, can we detect whether it is partially watermarked?
- If it is partially watermarked, can we estimate how much of it—i.e., what proportion—was generated by the LLM?

Preliminaries

Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

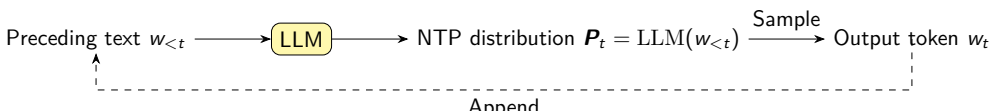
- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.

Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.
- **Autoregresiveness**: An LLM generates each token sequentially by sampling from a probability distribution conditioned on previous tokens:

$$w_t \sim \boldsymbol{P}_t \quad \text{where} \quad \boldsymbol{P}_t = \mathrm{LLM}(w_{<t}) \quad \text{is a distribution on} \quad \mathcal{W}.$$

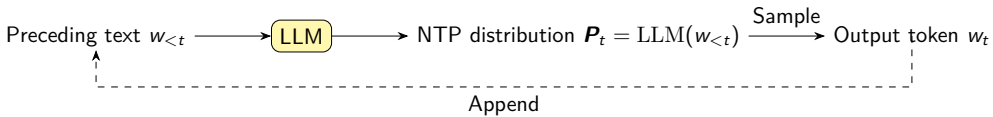


Autoregresive generation: How LLMs combine tokens

Denote the vocabulary by $\mathcal{W} = \{1, \ldots, K\}$, a token therein by w_t, and a text by $w_{<t} := w_1 \cdots w_{t-1}$.

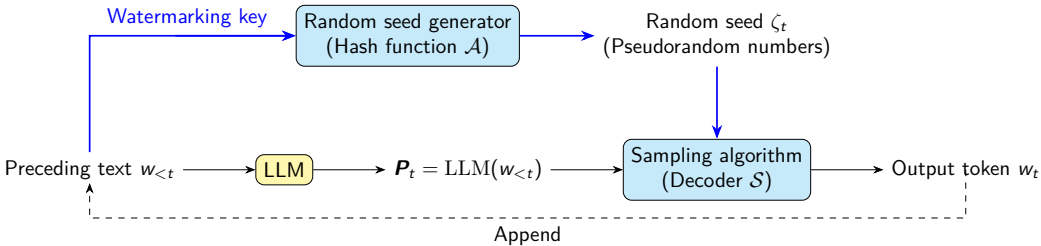
- **Large vocabulary**: $\mathcal{W}$ is large in practice; $K = 50,257$ for GPT-2/3.5, $= 32,000$ for LLaMa models, and $= 128,000$ for DeepSeek-R1.
- **Autoregresiveness**: An LLM generates each token sequentially by sampling from a probability distribution conditioned on previous tokens:

$$w_t \sim \boldsymbol{P}_t \quad \text{where} \quad \boldsymbol{P}_t = \mathrm{LLM}(w_{<t}) \quad \text{is a distribution on} \quad \mathcal{W}.$$



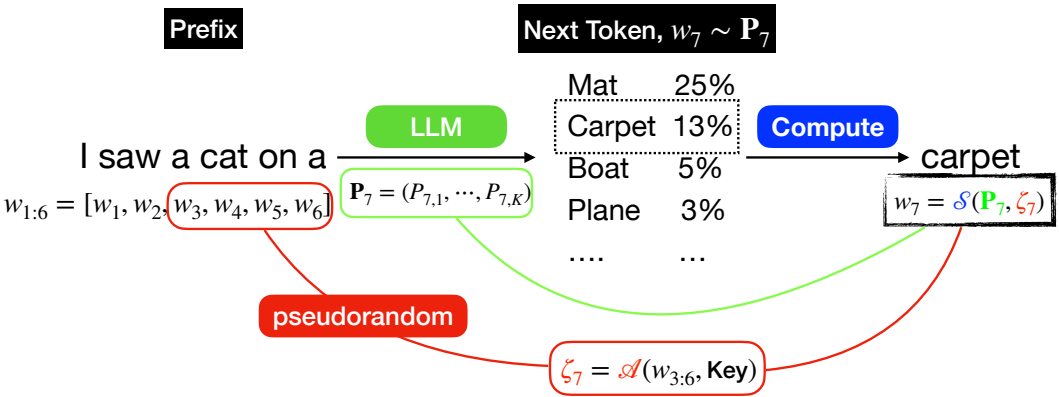
- **Limited access**: The distribution $\boldsymbol{P}_t$ is referred to next-token prediction (NTP) distribution, which is unknown since it depends on unknown system/user prompts.

Watermarked generation: Procedure



- Mathematical speaking: $\zeta_t = \mathcal{A}(w_{<t}, \mathrm{Key})$ and $w_t = \mathcal{S}(\boldsymbol{P}_t, \zeta_t)$.
- A watermark is defined by $(\mathcal{A}, \mathcal{S}, \mathrm{Key})$.
- Watermark signal is the dependence of each w_t on ζ_t.

Watermarked generation: Example



A high-level intro of watermark detection

$H_0 : w_{\leq n}$ is human written v.s. $H_0 : w_{\leq n}$ is LLM-generated.

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

A high-level intro of watermark detection

$H_0 : w_{\leq n}$ is human written v.s. $H_0 : w_{\leq n}$ is LLM-generated.

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

Take-away

Watermarking couples each token w_t and a psedorandom ζ_t, altering their joint distribution. Watermarking detection tries to detect this coupling.

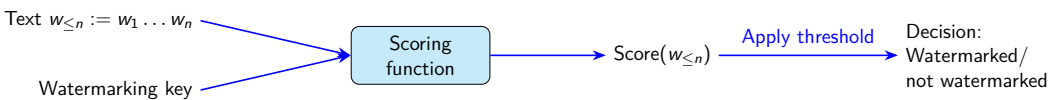
A high-level intro of watermark detection

$H_0 : w_{\leq n}$ is human written v.s. $H_0 : w_{\leq n}$ is LLM-generated.

- **Human-written text**: w_t is independent of ζ_t as humans don't know $\mathcal{A}$ and Key.
- **LLM-generated text**: w_t depends on ζ_t via the decoder function $\mathcal{S}$.

Take-away

Watermarking couples each token w_t and a psedorandom ζ_t, altering their joint distribution. Watermarking detection tries to detect this coupling.



Hypothesis testing via pivoting

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$

Hypothesis testing via pivoting

$$H_0 : Y_t \overset{iid}{\sim} \mu_0, \ t = 1, \ldots, n \qquad \text{vs} \qquad H_1 : Y_t | \boldsymbol{P}_t \sim \mu_{1,P_t}, \ t = 1, \ldots, n$$

Hypothesis testing via pivoting

Find a pivotal statistic $Y_t = Y(w_t, \zeta_t)$ such that

- Under H_0, $Y_t \sim \mu_0$, regardless of $\boldsymbol{P}_t$
- Under H_1, $Y_t \sim Y(\mathcal{S}(\zeta_t, \boldsymbol{P}_t), \zeta_t)$, whose distribution is denoted $\mu_{1,\boldsymbol{P}_t}$

Hypothesis testing via pivoting

$$H_0 : Y_t \overset{iid}{\sim} \mu_0,\ t = 1, \ldots, n \qquad \text{vs} \qquad H_1 : Y_t | \boldsymbol{P}_t \sim \mu_{1,P_t},\ t = 1, \ldots, n$$

- Not unique, may lead to information loss, but convenient.
- A good choice of Y should have a similar distributional shift for any $\boldsymbol{P}_t$.
- Test distributional difference rather than independence using test score $T_h = \sum_{t=1}^{n} h(Y_t)$ for some score function h. Reject H_0 if T_h is larger than a threshold.

Gumbel-max watermark [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Gumbel-max watermark [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Definition (*Gumbel-max watermark*)

With $\zeta_t = (U_t, \ldots, U_K) = \mathcal{A}(w_{<t}, \mathrm{Key})$ (often depending on the last 5 tokens),

$$w_t = \mathcal{S}^{\mathrm{gum}}(\boldsymbol{P}_t, \zeta_t) = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}}.$$

- It is **unbiased** as the marginal dist. of w (first arg.) in $(\mathcal{S}(\boldsymbol{P}, \zeta), \zeta)$ is still $\boldsymbol{P}$.

Gumbel-max watermark [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Definition (*Gumbel-max watermark*)

With $\zeta_t = (U_t, \ldots, U_K) = \mathcal{A}(w_{<t}, \mathrm{Key})$ (often depending on the last 5 tokens),

$$w_t = \mathcal{S}^{\mathrm{gum}}(\boldsymbol{P}_t, \zeta_t) = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}}.$$

- It is **unbiased** as the marginal dist. of w (first arg.) in $(\mathcal{S}(\boldsymbol{P}, \zeta), \zeta)$ is still $\boldsymbol{P}$.
- Implemented internally at OpenAI.

Gumbel-max watermark [Aaronson, 2023]

Definition (*Gumbel-max trick [Gumbel, 1948]*)

Let $U_1, \ldots, U_K$ be i.i.d. copies of $\mathcal{U}(0,1)$. Then,

$$\arg\max_{w \in \mathcal{W}} \frac{\log U_w}{P_w} \sim \boldsymbol{P} \equiv (P_w)_{w \in \mathcal{W}}.$$

Definition (*Gumbel-max watermark*)

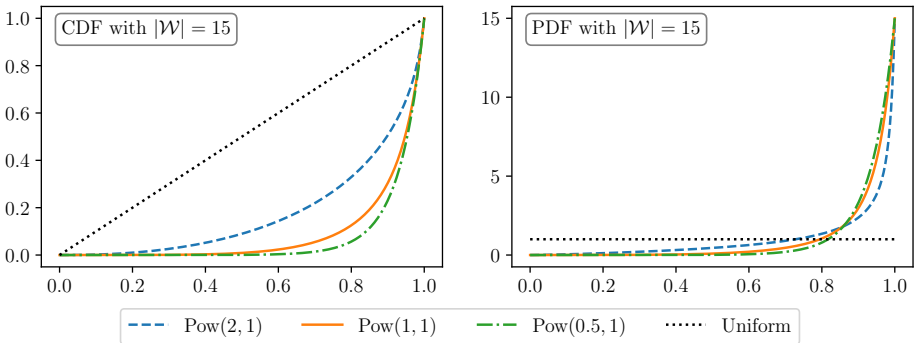
With $\zeta_t = (U_t, \ldots, U_K) = \mathcal{A}(w_{<t}, \mathrm{Key})$ (often depending on the last 5 tokens),

$$w_t = \mathcal{S}^{\mathrm{gum}}(\boldsymbol{P}_t, \zeta_t) = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}}.$$

- It is **unbiased** as the marginal dist. of w (first arg.) in $(\mathcal{S}(\boldsymbol{P}, \zeta), \zeta)$ is still $\boldsymbol{P}$.
- Implemented internally at OpenAI.
- Embedded signal: selected U_{w_t} tends to be larger.

Pivot for Gumbel-max watermark

- The pivotal statistic is $Y_t^{\text{ars}} = U_{t,w_t}$.
- Under H_0, $Y_t^{\text{ars}} \overset{iid}{\sim} \mu_0 = \mathcal{U}(0,1)$.
- Under H_1, the CDF of $\mu_{1,\boldsymbol{P}_t}$ is $\mathbb{P}_1(Y_t^{\text{ars}} \leq r|\boldsymbol{P}_t) = \sum_{k=1}^{K} P_{t,k} r^{1/P_{t,k}}$.



Recall: Questions we study

Core questions

- Given a potentially modified text, can we detect whether it is partially watermarked?
- If it is partially watermarked, can we estimate how much of it—i.e., what proportion—was generated by the LLM?

Watermark detection under text modification

- To cope with modification, practice often uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.
- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Watermark detection under text modification

- To cope with modification, practice often uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.
- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

Watermark detection under text modification

- To cope with modification, practice often uses a few tokens to compute pseudorandom numbers

  For example, $\zeta_t = \mathcal{A}(w_{(t-5):(t-1)}, \mathrm{Key})$, using the last 5 tokens.
- A modified token will turn the watermark signals in the next few 5 tokens to noise.

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- $\varepsilon_n \in [0,1]$ denote the proportion of watermark signals.
- When $\varepsilon_n \equiv 1$, it reduced to the full detection setting in [Li et al., 2025a].

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

Key insight

We know everything about the null H_0 which always assume $Y_{\leq n}$ iid from μ_0.

How to solve the mixture detection

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{vs} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t$$

- Difficulties: We know nothing about ε_n or $\boldsymbol{P}_t$.

Key insight

We know everything about the null H_0 which always assume $Y_{\leq n}$ iid from μ_0.

- Focus to determine whether the observed $Y_1, \ldots, Y_n$ follows μ_0.
- Tr-GoF [Li et al., 2024] checks the deviation between the empirical CDF of $Y_{\leq n}$ and μ_0 via an f-divergence.
- Too large deviation indicate the existence of watermarked subtexts.

Tr-GoF [Li et al., 2024]

- The empirical CDF of p-values: $\mathbb{F}_n(r) = \frac{1}{n}\sum_{t=1}^{n} \mathbf{1}_{\mathsf{p}_t \le r}$ where $\mathsf{p}_t = 1 - F_0(Y_t)$.
- Under H_0, we must have $\mathsf{p}_1, \dots, \mathsf{p}_n$ i.i.d. from $\mathcal{U}(0,1)$.
- Tr-GoF rejects H_0 if $\sup_{r\in(0,1)} nK_s(\mathbb{F}_n(r), r)$ is larger than expected for $s \in [-1, 2]$ where

$$K_s(u,v) = \frac{1}{s(1-s)}\left[1 - u^s v^{1-s} - (1-u)^s (1-v)^{1-s}\right].$$

- One can show that K_s is the ϕ_s-divergence between two Bernoulli variables:

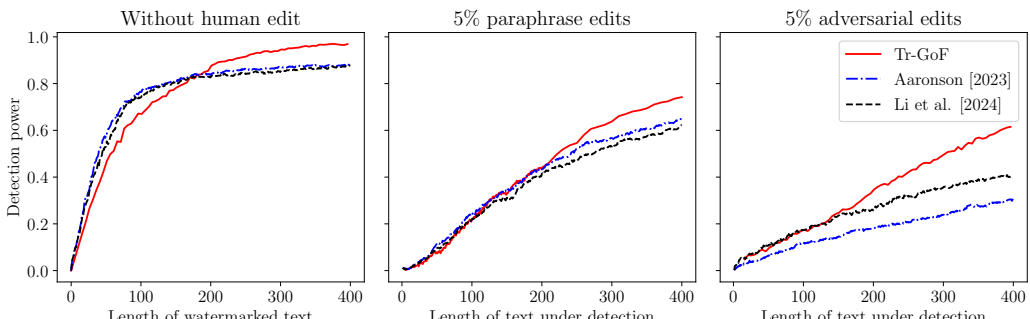
$$K_s(u,v) = D_{\phi_s}(\mathrm{Ber}(u) \parallel \mathrm{Ber}(v)) = v\phi_s\left(\frac{u}{v}\right) + (1-v)\phi_s\left(\frac{1-u}{1-v}\right),$$

where ϕ_s is a scalar convex function indexed by s:

$$\phi_s(x) = \begin{cases} x\log x - x + 1, & \text{if } s = 1, \\ \frac{1-s+sx-x^s}{s(1-s)}, & \text{if } s \neq 0, 1, \\ -\log x + x - 1, & \text{if } s = 0. \end{cases}$$

Better performance of Tr-GoF

- On C4 news-like dataset [Raffel et al., 2020] and OPT-1.3B model [Zhang et al., 2022] (temperature 0.3).



Why the Tr-GoF test performs so well?

A question

Why the Tr-GoF test performs so well in the watermark detection problem?

- We focus on the Gumbel-max watermark. Similar analysis could be paralleled to other watermarks.

Why the Tr-GoF test performs so well?

A question

Why the Tr-GoF test performs so well in the watermark detection problem?

- We focus on the Gumbel-max watermark. Similar analysis could be paralleled to other watermarks.

High-level answers

The Tr-GoF test achieves optimal robustness in two senses:

1. Optimal detection boundary in a decaying watermark-signal case.
2. Optimal detection efficiency rate in a constant corruption case.

!!! No knowledge about the proportion ε_n and NTP distributions.

When the robust detection is possible?

Optimal detection boundary

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1 - \varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

When the robust detection is possible?

Optimal detection boundary

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

Definition (*A difficult case*)

We consider an extreme case where

- $\varepsilon_n \asymp n^{-p}$ for all t and $p \in (0,1]$.
- $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = \Delta_n$ for all t with $\Delta_n \asymp n^{-q}$ and $q \in (0,1)$.

When the robust detection is possible?

Hypothesis testing under mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

Definition (*A difficult case*)

We consider an extreme case where

- $\varepsilon_n \asymp n^{-p}$ for all t and $p \in (0,1]$.
- $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = \Delta_n$ for all t with $\Delta_n \asymp n^{-q}$ and $q \in (0,1)$.

- Motivated by sparse detection problem [Donoho and Jin, 2004, 2015].
- If $\varepsilon_n = 0$ or $1 - \max_{w \in \mathcal{W}} \boldsymbol{P}_{t,w} = 0$, $(1-\varepsilon_n)\mu_0 + \varepsilon_n \mu_{1,\boldsymbol{P}_t} = \mu_0$, i.e., H_0 merges with H_1^{mix}.

When the robust detection is possible?

Theorem (*Optimal detection boundary on the* (p, q)*-plane*)

- If $q + 2p > 1$, H_0 and H_1^{mix} merge asymptotically.
- If $q + 2p < 1$, H_0 and H_1^{mix} separate asymptotically.

- Proof based on Donoho and Jin [2004].
- How to achieve robust detection in the regime $q + 2p < 1$? LRT is impractical since it requires knowing $\boldsymbol{P}_t$'s.

When the robust detection is possible?

Theorem (*Optimal detection boundary on the* (p, q)*-plane*)

- If $q + 2p > 1$, H_0 and H_1^{mix} merge asymptotically.
- If $q + 2p < 1$, H_0 and H_1^{mix} separate asymptotically.

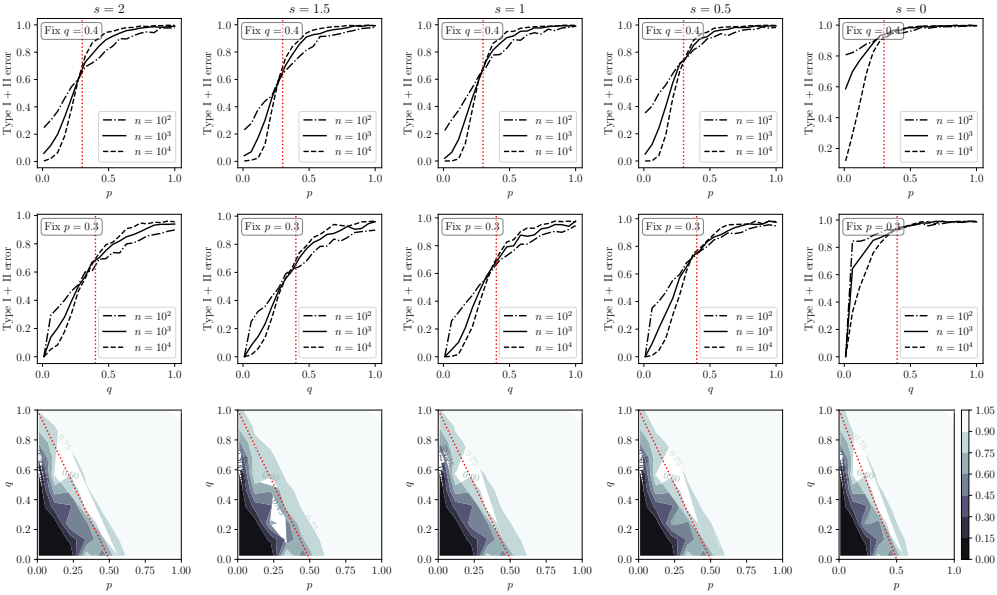
- Proof based on Donoho and Jin [2004].
- How to achieve robust detection in the regime $q + 2p < 1$? LRT is impractical since it requires knowing $\boldsymbol{P}_t$'s.

Theorem (*Adaptive optimality*)

The Type I and II errors of the Tr-GoF test $\to 0$ if $n \to \infty$ as long as $q + 2p < 1$.

Empirical detection boundaries of Tr-GoF

Optimal detection boundary



Suboptimality of sum-based tests

- Consider the sum-based test that rejects H_0 if

$$\sum_{t=1}^{n} h(Y_t^{\mathrm{ars}}) \geq n \cdot \mathbb{E}_0[h(Y^{\mathrm{ars}})] + \Theta(1) \cdot n^{\frac{1}{2}} \cdot \mathrm{poly}(\log n).$$

Theorem (*Suboptimality of sum-based tests*)

When $\varepsilon < 1$, the detection boundary for general (Δ, ε)-agnostic sum-based tests is $q + p = 1/2$ (which include $h \in \{h_{\mathrm{ars}}, h_{\log}, h_{\mathrm{opt},\Delta}\}$).

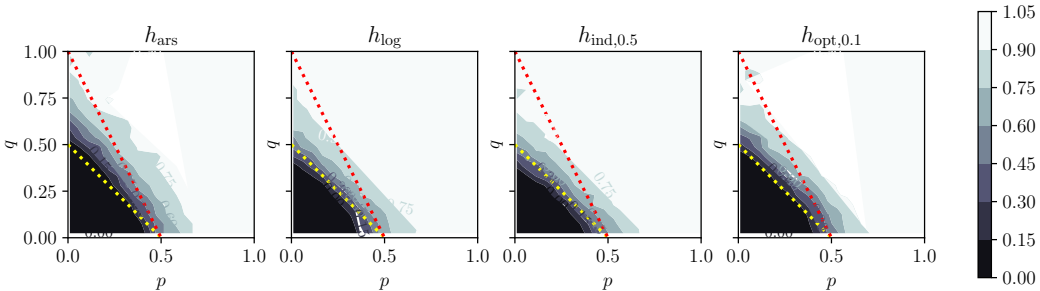
Suboptimality of sum-based tests

- Consider the sum-based test that rejects H_0 if

$$\sum_{t=1}^{n} h(Y_t^{\mathrm{ars}}) \geq n \cdot \mathbb{E}_0[h(Y^{\mathrm{ars}})] + \Theta(1) \cdot n^{\frac{1}{2}} \cdot \mathrm{poly}(\log n).$$

Theorem (*Suboptimality of sum-based tests*)

When $\varepsilon < 1$, the detection boundary for general (Δ, ε)-agnostic sum-based tests is $q + p = 1/2$ (which include $h \in \{h_{\mathrm{ars}}, h_{\log}, h_{\mathrm{opt},\Delta}\}$).



What about constant corruption?

- Practical settings meet with the constant corruption case, i.e., $\varepsilon_n \equiv \varepsilon$.
- The problem is detectable because $p = q = 0$ (within $q + 2p < 1$).

What about constant corruption?

- Practical settings meet with the constant corruption case, i.e., $\varepsilon_n \equiv \varepsilon$.
- The problem is detectable because $p = q = 0$ (within $q + 2p < 1$).
- Li et al. [2025a] introduces a detection efficiency motion where all NTP distributions fall within a belief set $\mathcal{P}$:

Definition ($\mathcal{P}$-*efficiency Li et al. [2025a]*)

$$R_{\mathcal{P}}(\text{detection rule}) := \lim_{n\to\infty} \sup_{\boldsymbol{P}_{\leq n}\in\mathcal{P}} -\frac{1}{n}\log(\text{Type II error}) \text{ s.t. Type I error fixed.}$$

What about constant corruption?

Theorem (*Optimal $\mathcal{P}_\Delta$-efficiency*)

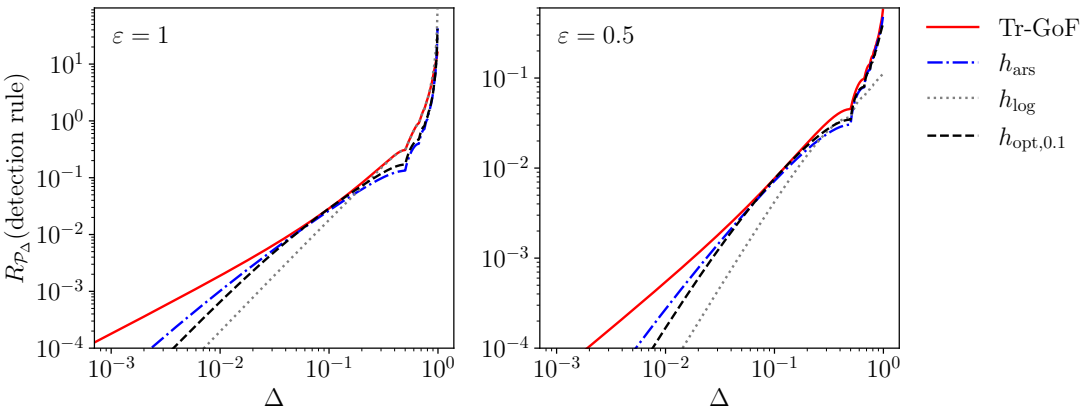
Let $s \in (0,1)$, $\varepsilon_n \equiv \varepsilon \in (0,1]$ and $\Delta_n \equiv \Delta \in (0,1)$.

$$R_{\mathcal{P}_\Delta}(\text{any detection rule}) = D_{\mathrm{KL}}(\mu_0, (1-\varepsilon)\mu_0 + \varepsilon\mu_{1,\boldsymbol{P}^\star_\Delta}) = R_{\mathcal{P}_\Delta}(\mathrm{Tr}-\mathrm{GoF})$$

- Δ-regular class: $\mathcal{P}_\Delta := \{\boldsymbol{P} = (P_1, \cdots, P_K) : \max_k P_k \le 1-\Delta\}$.
- $\boldsymbol{P}^\star_\Delta$ is the least-favorable NTP in $\mathcal{P}_\Delta$.
- Optimal efficiency without any prior knowledge.
- When $\varepsilon = 1$, this rate is obtained by the sum-based test in [Li et al., 2025a].

Theoretical $\mathcal{P}_\Delta$-efficiency comparison

Optimal detection efficiency



Proportion estimation

Hypothesis testing under constant mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon)\mu_0 + \varepsilon \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

- Once we have confirmed that $w_{\leq n}$ was generated by the LLM (i.e. reject H_1^{mix}), how many were produced by the model?
- **Application**: Measure LLMs' intellectual contribution.

Proportion estimation

Hypothesis testing under constant mixtures

$$H_0 : Y_t \sim \mu_0 \ \forall t \quad \text{versus} \quad H_1^{\text{mix}} : Y_t | \boldsymbol{P}_t \sim (1-\varepsilon)\mu_0 + \varepsilon \mu_{1,\boldsymbol{P}_t} \ \forall t.$$

- Once we have confirmed that $w_{\leq n}$ was generated by the LLM (i.e. reject H_1^{mix}), how many were produced by the model?
- **Application**: Measure LLMs' intellectual contribution.

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

- ε is not identifiable for green-red list watermark.

Lemma

If $Y_{1:n}$ are i.i.d. from the binary mixture $(1-\varepsilon)\mathrm{Ber}(\gamma) + \varepsilon\mathrm{Ber}(\mu)$ where both ε and μ are unknown with γ known, ε is not identifiable (as $Y_t \overset{iid}{\sim} \mathrm{Ber}((1-\varepsilon)\gamma) + \varepsilon\mu)$).

When ε is identifiable?

Definition (*Proportion estimation under constant mixtures [Li et al., 2025b]*)

Given independent data $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ for all t, how to estimate ε accurately?

- ε is not identifiable for green-red list watermark.

Lemma

If $Y_{1:n}$ are i.i.d. from the binary mixture $(1-\varepsilon)\mathrm{Ber}(\gamma) + \varepsilon\mathrm{Ber}(\mu)$ where both ε and μ are unknown with γ known, ε is not identifiable (as $Y_t \overset{iid}{\sim} \mathrm{Ber}((1-\varepsilon)\gamma) + \varepsilon\mu)$).

ε is identifiable for Gumbel-max watermark (and other wm with continuous Y).

Lemma

If $Y_t \sim (1-\varepsilon)F_0 + \varepsilon F_{\boldsymbol{P}_t}$ and $\lim\limits_{x\to 0} \frac{\bar{F}_{\boldsymbol{P}}(x)}{F_0(x)} = 0$, then ε is identifiable (as $\varepsilon = 1 - \lim_{x\to 0} \frac{\bar{F}(x)}{F_0(x)}$ is well-defined).

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Two difficulties: **(1)** no access to $\bar{F}_{\boldsymbol{P}}$ and **(2)** which v to use.

How to estimate ε when possible?

Some notations

$\bar{F}(y) = (1-\varepsilon)F_0(y) + \varepsilon \bar{F}_{\boldsymbol{P}}(y)$ for all y,

- $\bar{F}(y) = \frac{1}{n}\sum_{t=1}^{n} \mathbb{P}(Y_t \leq y)$
- $F_0(y) = \mu_0(Y \leq y)$
- $\bar{F}_{\boldsymbol{P}}(y) = \frac{1}{n}\sum_{t=1}^{n} \mu_{1,\boldsymbol{P}_t}(Y \leq y)$

Key idea: "Moment" matching

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Two difficulties: **(1)** no access to $\bar{F}_{\boldsymbol{P}}$ and **(2)** which v to use.

- **Estimate** $\bar{F}_{\boldsymbol{P}}$: collect water-marked outputs from similar LLMs $\Rightarrow$ empirical $\widehat{F}_{\boldsymbol{P}}$.
- **Choose** v: set heuristically *or* optimize against a clear performance criterion.

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Proportion estimators

Key observation

For any weight function $v : \mathbb{R} \mapsto \mathbb{R}$,

$$\varepsilon = \frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}.$$

Collected data

- Let $\widehat{F}$ denote the empirical CDF of observed $Y_{\leq n}$.
- Let $\widehat{F}_{\boldsymbol{P}}$ approximate the alternative CDF $\bar{F}_{\boldsymbol{P}}$ (no accuracy guarantees).

Our method: Use $\widehat{F}_{\boldsymbol{P}}$ & optimal v

Fixed point of $\widehat{\varepsilon}_{\mathrm{opt}} = \widehat{\mathcal{T}}(\widehat{\varepsilon}_{\mathrm{opt}})$ where

$$\widehat{\mathcal{T}}(\varepsilon) = \frac{\int \widehat{v}_{\mathrm{opt}}(\varepsilon, y) \left[\mathrm{d}F_0(y) - \mathrm{d}\widehat{F}(y)\right]}{\int \widehat{v}_{\mathrm{opt}}(\varepsilon, y) \left[\mathrm{d}F_0(y) - \mathrm{d}\widehat{F}_{\boldsymbol{P}}(y)\right]}$$

$$\widehat{v}_{\mathrm{opt}}(\varepsilon, y) = \frac{1 - \widehat{g}(y)}{(1-\varepsilon) + \varepsilon \widehat{g}(y)}, \widehat{g}(y) = \frac{\mathrm{d}\widehat{F}_{\boldsymbol{P}}(y)}{\mathrm{d}F_0(y)}.$$

Why optimal?

Lemma (*Optimal estimator variance*)

If $\widehat{F}_{\boldsymbol{P}} = \bar{F}_{\boldsymbol{P}}$, it follows that

$$\mathrm{Var}\left(\frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\widehat{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}\right) \leq \frac{\mathrm{Var}_{\bar{F}}(v)}{n(\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v])^2}$$

Why optimal?

Lemma (*Optimal estimator variance*)

If $\widehat{F}_{\boldsymbol{P}} = \bar{F}_{\boldsymbol{P}}$, it follows that

$$\operatorname{Var}\left(\frac{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\widehat{F}}[v]}{\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]}\right) \leq \frac{\operatorname{Var}_{\bar{F}}(v)}{n(\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v])^2}$$

Lemma (*Optimal weight function*)

$$\min_v \frac{\operatorname{Var}_{\bar{F}}(v)}{[\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]]^2} = \left[\int \frac{[1 - g(x)]^2}{(1-\varepsilon) + \varepsilon g(x)} \mathrm{d}F_0(x)\right]^{-1}$$

where the optimal solution (up to constant factors) is

$$v_{\mathrm{opt}}(x) = \frac{1 - g(x)}{(1-\varepsilon) + \varepsilon g(x)}, \text{ with } g(x) = \frac{\mathrm{d}\bar{F}_{\boldsymbol{P}}(x)}{\mathrm{d}F_0(x)}.$$

Why optimal?

Lemma (*Optimal weight function*)

$$\min_{v} \frac{\mathrm{Var}_{\bar{F}}(v)}{[\mathbb{E}_{F_0}[v] - \mathbb{E}_{\bar{F}_{\boldsymbol{P}}}[v]]^2} = \left[\int \frac{[1-g(x)]^2}{(1-\varepsilon)+\varepsilon g(x)} \mathrm{d}F_0(x)\right]^{-1} =: [\tau_n^\star]^2,$$

where the optimal solution (up to constant factors) is

$$v_{\mathrm{opt}}(x) = \frac{1-g(x)}{(1-\varepsilon)+\varepsilon g(x)}, \text{ with } g(x) = \frac{\mathrm{d}\bar{F}_{\boldsymbol{P}}(x)}{\mathrm{d}F_0(x)}.$$

Theorem (*Minimax optimality*)

$\widehat{\varepsilon}_{\mathrm{opt}}$ is the minimax optimal estimator up to the estimation error in $\widehat{F}_{\boldsymbol{P}}$, i.e.,

$$\mathbb{E}|\widehat{\varepsilon}_{\mathrm{opt}} - \varepsilon| \lesssim \frac{\tau_n^\star + o(1)}{\sqrt{n}} + \text{estimation errors due to } \widehat{F}_{\boldsymbol{P}}$$

Empirical performance [Li et al., 2025b]

- On `arXiv` dataset [Cohan et al., 2018] and OPT-13B model [Zhang et al., 2022] (temperature 1.0).

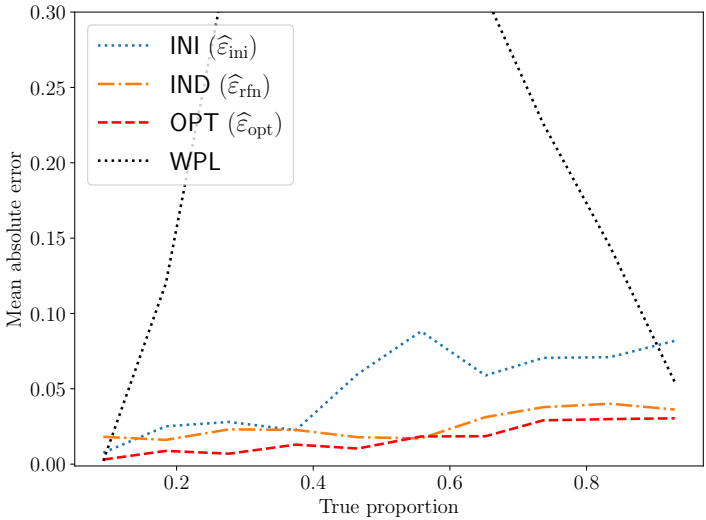


Table: Averaged MAEs calculated over 10 ground truth ε values on open-sources model experiments. Standard deviations are provided in parentheses, and all values are reported in units of 10^{-3}. Bold numbers denote the best performance.

Models	Datasets	T	Gumbel-max				Inverse transform			
			WPL	INI	IND	OPT	WPL	INI	IND	OPT
OPT-1.3B	C4	0.7	123(62)	77(59)	51(28)	**39(20)**	214(125)	52(52)	43(**17**)	**35**(21)
	C4	1	254(167)	65(40)	6(4)	**4(3)**	247(142)	31(31)	5(**3**)	**4(3)**
	Arxiv	1	275(184)	70(105)	19(8)	**11(6)**	286(174)	17(14)	18(**8**)	**12(8)**
OPT-13B	C4	0.7	119(90)	122(94)	34(19)	**26(15)**	212(135)	49(28)	25(**12**)	**20(12)**
	C4	1	195(156)	56(40)	8(5)	**5(3)**	250(143)	94(43)	70(48)	**68(40)**
	Arxiv	1	253(162)	51(27)	26(**9**)	**17**(10)	262(140)	27(15)	21(12)	**16(8)**
LLaMA-8B	C4	0.7	82(71)	60(34)	90(42)	**75(37)**	160(113)	60(34)	90(42)	**75(37)**
	C4	1	263(178)	45(22)	6(3)	**4(2)**	148(77)	30(32)	**2(1)**	5(4)
	Arxiv	1	236(201)	44(18)	18(9)	**15(7)**	291(176)	32(32)	19(11)	**16(6)**

Table: Average MAEs under common modifications, computed over 11 ground-truth ε values on the C4 dataset with temperature 1. Standard deviations are shown in parentheses, and all values are scaled by 10^{-3}. Boldface denotes the best performance.

Models	Edit types	Gumbel-max				Inverse transform			
		WPL	INI	IND	OPT	WPL	INI	IND	OPT
OPT-1.3B	Substitution	103(63)	56(24)	3(2)	**1(1)**	275(109)	46(35)	5(3)	**2(1)**
	Insertion	105(66)	70(35)	**8(5)**	**8(5)**	282(84)	38(23)	9(8)	**9(5)**
	Deletion	170(88)	**38**(27)	71(19)	66**(18)**	268(108)	35(27)	68(18)	**65(19)**
OPT-13B	Substitution	64(33)	78(55)	5(5)	**1(1)**	262(85)	52(28)	4(2)	**1(1)**
	Insertion	60(33)	61(32)	12**(4)**	8**(5)**	244(95)	50(40)	**8**(7)	**8(5)**
	Deletion	100(56)	**49**(38)	63**(15)**	66(18)	259(89)	26(23)	72**(17)**	68**(17)**
LLaMA-8B	Substitution	127(70)	54(28)	6(5)	**1(1)**	236(127)	50(37)	3(4)	**2(1)**
	Insertion	126(73)	46(22)	7(4)	**2(1)**	243(121)	18(15)	8(6)	**3(2)**
	Deletion	201(60)	**34**(22)	56(24)	38**(20)**	270(109)	54(47)	54(21)	**36(19)**

Concluding remarks

Robust Detection of Watermarks for Large Language Models Under Human Edits (https://arxiv.org/abs/2411.13868)
Optimal Estimation of Watermark Proportions in Hybrid AI-Human Texts (https://arxiv.org/abs/2506.22343)

Concluding remarks

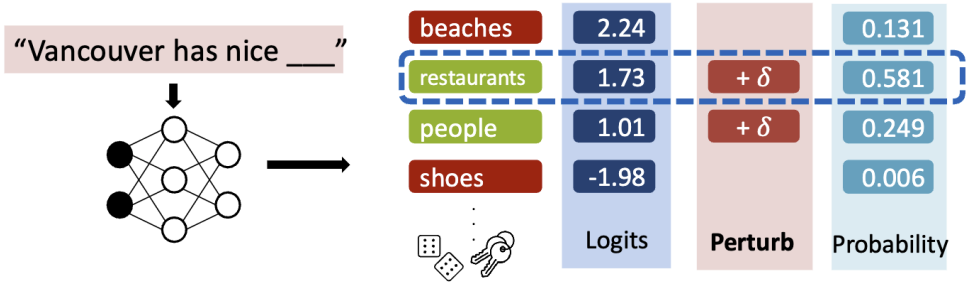
Robust Detection of Watermarks for Large Language Models Under Human Edits (https://arxiv.org/abs/2411.13868)
Optimal Estimation of Watermark Proportions in Hybrid AI-Human Texts (https://arxiv.org/abs/2506.22343)

- The Tr-GoF test achieves **adaptive optimality** for robust detection, whereas existing sum-based tests fail to do so.
- It also achieves the **highest $\mathcal{P}_\Delta$-efficiency** without requiring any prior knowledge.
- The watermark proportion is **not estimable** for every watermark—for example, it is not identifiable under the green–red list watermark.
- When it is estimable, the **watermark proportion** can be accurately recovered—even without access to next-token prediction (NTP) distributions.

Backup Slides

Green-red list watermark [Kirchenbauer et al., 2023]

- Randomly split vocabulary in to green (favored) and red (disfavored) parts.
- Secretly boost the prob. of green tokens, i.e., $P^{\mathrm{wm}}_{\mathrm{green}} \propto e^{\delta} P_{\mathrm{green}}$ and $P^{\mathrm{wm}}_{\mathrm{red}} \propto P_{\mathrm{red}}$.
- If the observed frequency of green tokens is larger than expected, claim watermarked.

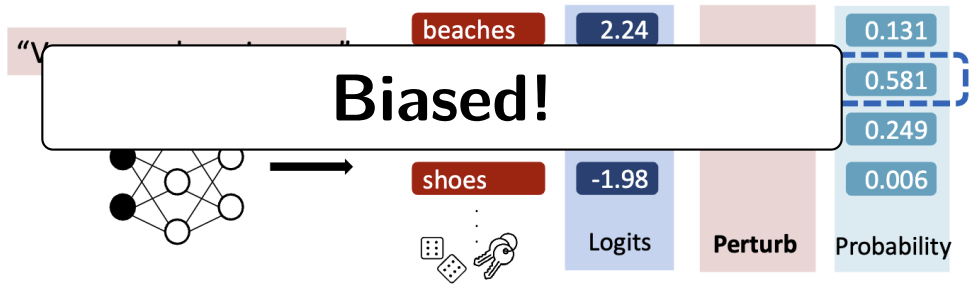


Kirchenbauer et al. (2023) A Watermark for Large Language Models
Zhao et al. (2023) Provable Robust Watermarking for AI-Generated Text

Figure from the tutorial: https://leililab.github.io/llm_watermark_tutorial/

Green-red list watermark [Kirchenbauer et al., 2023]

- Randomly split vocabulary in to green (favored) and red (disfavored) parts.
- Secretly boost the prob. of green tokens, i.e., $P^{\text{wm}}_{\text{green}} \propto e^{\delta} P_{\text{green}}$ and $P^{\text{wm}}_{\text{red}} \propto P_{\text{red}}$.
- If the observed frequency of green tokens is larger than expected, claim watermarked.



Kirchenbauer et al. (2023) A Watermark for Large Language Models
Zhao et al. (2023) Provable Robust Watermarking for AI-Generated Text

Figure from the tutorial: https://leililab.github.io/llm_watermark_tutorial/

Detection for Gumbel-max watermark

Definition (*Default detection for Gumbel-max*)

Aaronson [2023] rejects H_0 if the following $T_{h_{\text{ars}}}$ is larger than a given threshold:

$$T_{h_{\text{ars}}} = \sum_{t=1}^{n} h_{\text{ars}}(Y_t^{\text{ars}}) \quad \text{where} \quad h_{\text{ars}}(y) = -\log(1-y).$$

- Under H_0, $h_{\text{ars}}(Y_t^{\text{ars}}) \overset{iid}{\sim} \text{Exp}(1)$ so that $\mathbb{E}_0[T_{\text{ars}}] = n$.
- Under H_1, $\mathbb{E}_1[T_{\text{ars}}] \geq n + \left(\frac{\pi^2}{6} - 1\right) \sum_{t=1}^{n} \mathbb{E}_1 \text{Ent}(\boldsymbol{P}_t)$ where $\text{Ent}(\boldsymbol{P}_t)$ is Shannon entropy defined by $-\sum_{k=1}^{K} P_{t,k} \log P_{t,k}$.
- Using the same Y_t^{ars}, Fernandez et al. [2023] finds that $-\log(1-y)$ works better than the variant $\log y$. Li et al. [2025a] proposes $\log(y^{\frac{1-\Delta}{\Delta}} + y^{\frac{\Delta}{1-\Delta}})$ when $\Delta < 0.5$.

References I

S. Aaronson. Watermarking of large language models. https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17, August 2023.

P. Abdalla and R. Vershynin. LLM watermarking using mixtures and statistical-to-computational gaps. *arXiv preprint arXiv:2505.01484*, 2025.

M. Christ, S. Gunn, and O. Zamir. Undetectable watermarks for language models. In *Conference on Learning Theory*, pages 1125–1139. PMLR, 2024.

A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 615–621, 2018.

S. Dathathri, A. See, S. Ghaisas, P.-S. Huang, R. McAdam, J. Welbl, V. Bachani, A. Kaskasoli, R. Stanforth, T. Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634 (8035):818–823, 2024.

D. Donoho and J. Jin. Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics*, 32(3):962–994, 2004.

D. Donoho and J. Jin. Higher criticism for large-scale inference, especially for rare and weak effects. *Statistical science*, 30(1):1–25, 2015.

P. Fernandez, A. Chaffin, K. Tit, V. Chappelier, and T. Furon. Three bricks to consolidate watermarks for large language models. *arXiv preprint arXiv:2308.00113*, 2023.

References II

J. Fu, X. Zhao, R. Yang, Y. Zhang, J. Chen, and Y. Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. *arXiv preprint arXiv:2402.12948*, 2024.

T. Gloaguen, N. Jovanović, R. Staab, and M. Vechev. Towards watermarking of open-source LLMs. *arXiv preprint arXiv:2502.10525*, 2025.

GPTZero. GPTZero: More than an AI detector preserve what's human. `https://gptzero.me/`, 2023.

E. J. Gumbel. *Statistical theory of extreme values and some practical applications: A series of lectures*, volume 33. US Government Printing Office, 1948.

Z. Hu, L. Chen, X. Wu, Y. Wu, H. Zhang, and H. Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=uWVC5FVidc`.

J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, volume 202, pages 17061–17084, 2023.

J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, and T. Goldstein. On the reliability of watermarks for large language models. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=DEJIDCmWOz`.

K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

References III

R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

X. Li, F. Ruan, H. Wang, Q. Long, and W. J. Su. Robust detection of watermarks in large language models under human edits. *arXiv preprint arXiv:2411.13868*, 2024.

X. Li, F. Ruan, H. Wang, Q. Long, and W. J. Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351, 2025a.

X. Li, G. Wen, W. He, J. Wu, Q. Long, and W. J. Su. Optimal estimation of watermark proportions in hybrid AI–human texts. *arXiv preprint arXiv:2506.22343*, 2025b.

W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou. GPT detectors are biased against non-native english writers. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

L. Pan, A. Liu, Z. He, Z. Gao, X. Zhao, Y. Lu, B. Zhou, S. Liu, X. Hu, L. Wen, et al. MarkLLM: An Open-Source Toolkit for LLM Watermarking. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, 2024.

J. Piet, C. Sitawarin, V. Fang, N. Mu, and D. Wagner. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*, 2023.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (1):5485–5551, 2020.

References IV

V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

D. Weber-Wulff, A. Anohina-Naumeca, S. Bjelobaba, T. Foltýnek, J. Guerrero-Dib, O. Popoola, P. **v**Sigut, and L. Waddington. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1):26, 2023.

Y. Wu, Z. Hu, H. Zhang, and H. Huang. DiPmark: A stealthy, efficient and resilient watermark for large language models. *arXiv preprint arXiv:2310.07710*, 2023.

Y. Xie, X. Li, T. Mallick, W. J. Su, and R. Zhang. Debiasing watermarks for large language models via maximal coupling. *arXiv preprint arXiv:2411.11203*, 2024.

ZeroGPT. ZeroGPT: Trusted GPT-4, ChatGPT and AI detector tool by ZeroGPT. `https://www.zerogpt.com/`, 2023.

S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

X. Zhao, P. V. Ananth, L. Li, and Y.-X. Wang. Provable robust watermarking for AI-generated text. In *International Conference on Learning Representations*, 2024a. URL `https://openreview.net/forum?id=SsmT8aO45L`.

X. Zhao, L. Li, and Y.-X. Wang. Permute-and-Flip: An optimally robust and watermarkable decoder for LLMs. *arXiv preprint arXiv:2402.05864*, 2024b.