Optimal Detection for Language Watermarks with Pseudorandom Collision

Connections to Data Curation and Quality Control in Generative AI

Xiang Li

University of Pennsylvania

The 9th International Workshop in Sequential Methodologies
June, 2026

Generative AI in Biomedical Data and Documentation

Applications

- Clinical record summarization [Bednarczyk et al., 2025]
- Ambient AI scribes for clinical documentation [Mehra et al., 2026]
- AI-generated clinical notes [Perkins et al., 2024]
- Synthetic medical datasets for research [Pezoulas et al., 2024, Yan et al., 2024]
- AI-generated scientific data (e.g., protein structures) [Chen et al., 2025]

Generative AI in Biomedical Data and Documentation

Applications

- Clinical record summarization [Bednarczyk et al., 2025]
- Ambient AI scribes for clinical documentation [Mehra et al., 2026]
- AI-generated clinical notes [Perkins et al., 2024]
- Synthetic medical datasets for research [Pezoulas et al., 2024, Yan et al., 2024]
- AI-generated scientific data (e.g., protein structures) [Chen et al., 2025]

Question

How to detect whether a document was generated by a particular LLM?

Why Does This Question Matter?

Question

How to detect whether a document was generated by a particular LLM?

Why Does This Question Matter?

Question

How to detect whether a document was generated by a particular LLM?

- **Data provenance:** Track the source of documents in AI-assisted data pipelines.

Why Does This Question Matter?

Question

How to detect whether a document was generated by a particular LLM?

- **Data provenance:** Track the source of documents in AI-assisted data pipelines.
- **Quality control:** Detect synthetic contamination in clinical datasets and training data [Shumailov et al., 2024].

Why Does This Question Matter?

Question

How to detect whether a document was generated by a particular LLM?

- **Data provenance:** Track the source of documents in AI-assisted data pipelines.
- **Quality control:** Detect synthetic contamination in clinical datasets and training data [Shumailov et al., 2024].
- **Accountability:** Verify whether AI-generated content follows controlled generation policies.

Why Does This Question Matter?

Question

How to detect whether a document was generated by a particular LLM?

- **Data provenance:** Track the source of documents in AI-assisted data pipelines.
- **Quality control:** Detect synthetic contamination in clinical datasets and training data [Shumailov et al., 2024].
- **Accountability:** Verify whether AI-generated content follows controlled generation policies.
- **Copyright protection:** Identify AI-generated content that reproduces or derives from copyrighted data.

AI-Generation Detection as Sequential Detection

Sequential viewpoint. Text is observed progressively: token by token, sentence by sentence, or document by document.

Detection goal. As more text is observed, we want to decide whether it was generated by a human or by a particular LLM.

Why this matters. This question becomes important as LLMs are increasingly used for reasoning, synthetic data generation, documentation, and alignment or controlled generation.

How to Detect AI-Generated Text

- Consider LLM v.s. human detection w.l.o.g.
- **Ad hoc methods** leverage context, linguistic patterns, and other markers:
  - Classifiers using synthetic and human text data [GPTZero, 2023, ZeroGPT, 2023]
  - Log probability curvature [Mitchell et al., 2023, Bao et al., 2023]
  - Divergent n-gram analysis [Yang et al., 2023]

How to Detect AI-Generated Text

- Consider LLM v.s. human detection w.l.o.g.
- **Ad hoc methods** leverage context, linguistic patterns, and other markers:
  - Classifiers using synthetic and human text data [GPTZero, 2023, ZeroGPT, 2023]
  - Log probability curvature [Mitchell et al., 2023, Bao et al., 2023]
  - Divergent n-gram analysis [Yang et al., 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].

How to Detect AI-Generated Text

- Consider LLM v.s. human detection w.l.o.g.
- **Ad hoc methods** leverage context, linguistic patterns, and other markers:
  - Classifiers using synthetic and human text data [GPTZero, 2023, ZeroGPT, 2023]
  - Log probability curvature [Mitchell et al., 2023, Bao et al., 2023]
  - Divergent n-gram analysis [Yang et al., 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 [Sadasivan et al., 2023].

How to Detect AI-Generated Text

- Consider LLM v.s. human detection w.l.o.g.
- **Ad hoc methods** leverage context, linguistic patterns, and other markers:
  - Classifiers using synthetic and human text data [GPTZero, 2023, ZeroGPT, 2023]
  - Log probability curvature [Mitchell et al., 2023, Bao et al., 2023]
  - Divergent n-gram analysis [Yang et al., 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 [Sadasivan et al., 2023].
- **Fundamentally impossible** to distinguish between LLM-generated and human-written text (based solely on the text itself).

A principled approach: Watermarking LLM-generated text

Idea

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

A principled approach: Watermarking LLM-generated text

Idea

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

A watermark embeds a private, detectable signal into LLM-generated text by modifying the model's generation only [Kirchenbauer et al., 2023].

A principled approach: Watermarking LLM-generated text

Idea

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

A watermark embeds a private, detectable signal into LLM-generated text by modifying the model's generation only [Kirchenbauer et al., 2023].

- Creates a statistical dependency between the visible text and a private information.

A principled approach: Watermarking LLM-generated text

Idea

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

A watermark embeds a private, detectable signal into LLM-generated text by modifying the model's generation only [Kirchenbauer et al., 2023].

- Creates a statistical dependency between the visible text and a private information.
- Unlikely to appear in human-written (or other source) text.

A principled approach: Watermarking LLM-generated text

Idea

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

A watermark embeds a private, detectable signal into LLM-generated text by modifying the model's generation only [Kirchenbauer et al., 2023].

- Creates a statistical dependency between the visible text and a private information.
- Unlikely to appear in human-written (or other source) text.
- Efficient, provable, and slight (or no) degradation on text quality.

An active research area with practical importance

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

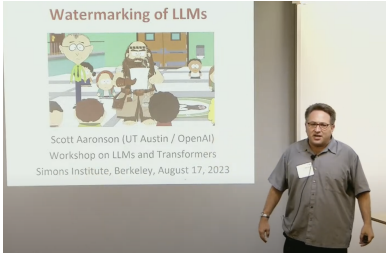
[Kirchenbauer et al., 2023, Aaronson, 2023, Kuditipudi et al., 2024, Zhao et al., 2025, Christ et al., 2024, Wu et al., 2024b, Hu et al., 2024, Kirchenbauer et al., 2024, Zhao et al., 2024, Xie et al., 2025, Fu et al., 2024, Dathathri et al., 2024, Gloaguen et al., 2025, Abdalla and Vershynin, 2025].

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

In July 2023, several AI giants like OpenAI, Anthropic, Google, and Meta pledged to the White House that they will voluntarily research and develop watermarking options.



Gumbel-max (OpenAI)

Article | Open access | Published: 23 October 2024

Scalable watermarking for identifying large language model outputs

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis & Pushmeet Kohli

Nature **634**, 818–823 (2024) | Cite this article

115k Accesses | **15** Citations | **986** Altmetric | Metrics

SynthID (Google Deepmind)

Statistical Questions for LLM Watermarks

Two Types of Errors

- **Type I error** – falsely flagging human prose as AI-generated
- **Type II error** – failing to detect AI-generated text

Evaluation of Watermarks

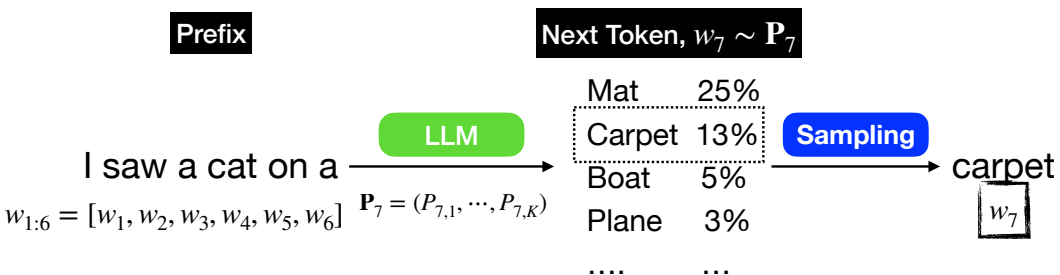
- More powerful detection [Li et al., 2025b]
- Robust detection [Li et al., 2025a]
- Estimation proportion [Li et al., 2025c]

How to Generate Texts

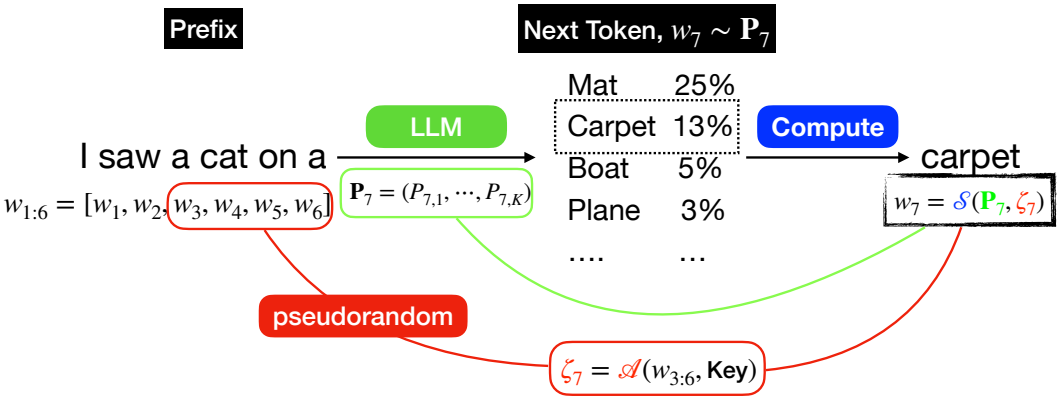
- The vocabulary $\mathcal{W} = \{1, \ldots, K\}$, a token w_t, and a text $w_{1:(t-1)} := w_1 \cdots w_{t-1}$.
- **Autoregresiveness**:
  $w_t \sim \boldsymbol{P}_t$ where $\boldsymbol{P}_t = \mathrm{LLM}(w_{1:(t-1)})$ is next-token prediction (NTP) dist. on $\mathcal{W}$.

How to Generate Texts

- The vocabulary $\mathcal{W} = \{1, \ldots, K\}$, a token w_t, and a text $w_{1:(t-1)} := w_1 \cdots w_{t-1}$.
- **Autoregresiveness**:
  $w_t \sim \boldsymbol{P}_t$ where $\boldsymbol{P}_t = \mathrm{LLM}(w_{1:(t-1)})$ is next-token prediction (NTP) dist. on $\mathcal{W}$.

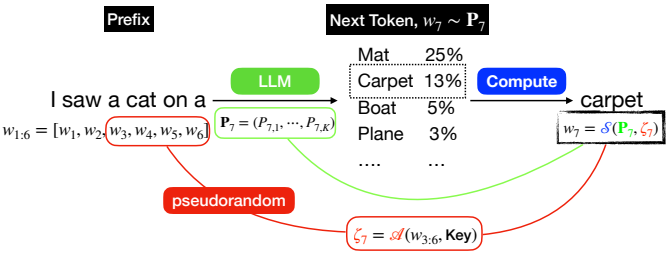


How to Embed Watermark

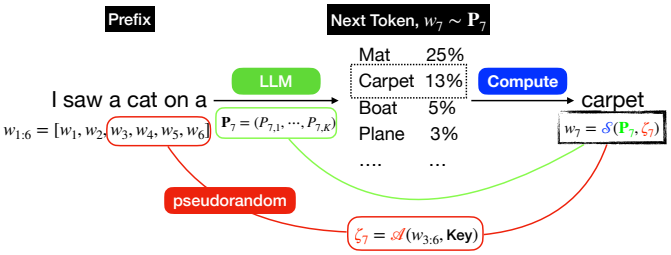


How to Embed Watermark

- Reparameterization trick.



How to Embed Watermark



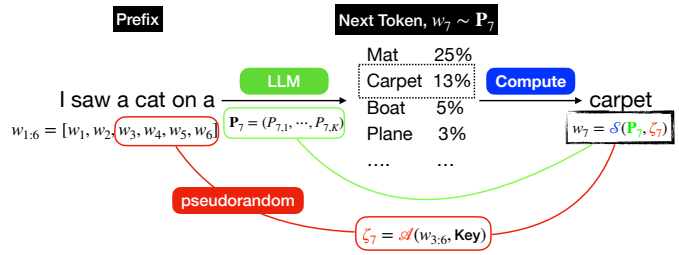
- Reparameterization trick.
- Generate a **pseudorandom variable**

$$\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \text{Key}),$$

where $\mathcal{A}$ is a hash function.

E.g.: `Random(seed = sum(`$w_{(t-m):(t-1)}$`)×Key).random()`

How to Embed Watermark



- Reparameterization trick.
- Generate a **pseudorandom variable**

$$\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \text{Key}),$$

where $\mathcal{A}$ is a hash function.

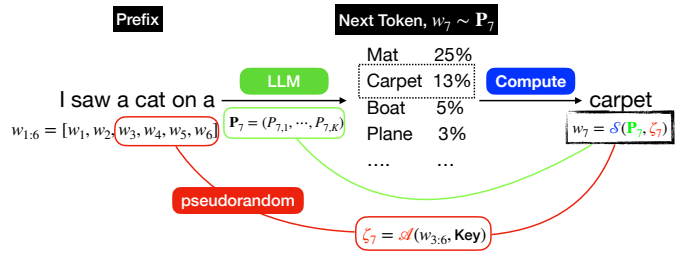
E.g.: `Random(seed = sum(`$w_{(t-m):(t-1)}$`)×Key).random()`

- A deterministic **unbiased decoder** $\mathcal{S}(\boldsymbol{P}, \zeta)$ samples a token:

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w, \ \forall w \in \mathcal{W}.$$

Does not degrade text quality.

How to Embed Watermark



- Reparameterization trick.
- Generate a **pseudorandom variable**

$$\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \text{Key}),$$

where $\mathcal{A}$ is a hash function.

E.g.: `Random(seed = sum(`$w_{(t-m):(t-1)}$`)×Key).random()`

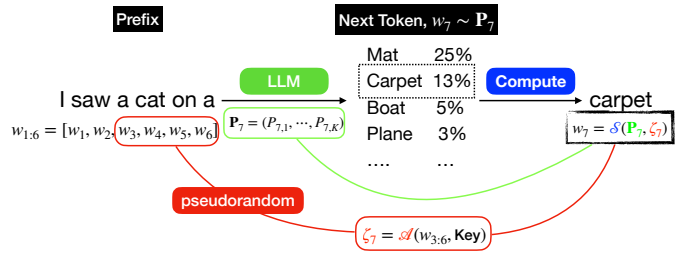
- A deterministic **unbiased decoder** $\mathcal{S}(\boldsymbol{P}, \zeta)$ samples a token:

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w, \ \forall w \in \mathcal{W}.$$

Does not degrade text quality.

- **Watermark signal:** statistical dependence between w_t and ζ_t.

How to Embed Watermark



- Reparameterization trick.
- Generate a **pseudorandom variable**

$$\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \text{Key}),$$

where $\mathcal{A}$ is a hash function.

E.g.: `Random(seed = sum(`$w_{(t-m):(t-1)}$`)×Key).random()`

- A deterministic **unbiased decoder** $\mathcal{S}(\boldsymbol{P}, \zeta)$ samples a token:

$$\mathbb{P}_\zeta(\mathcal{S}(\boldsymbol{P}, \zeta) = w) = P_w, \ \forall w \in \mathcal{W}.$$

Does not degrade text quality.

- **Watermark signal:** statistical dependence between w_t and ζ_t.
- $\zeta_{1:n}$ are i.i.d. but recoverable.

Gumbel-max Watermark [Aaronson, 2023]

Generation

- **Pseudorandomness:** $\zeta_t = (U_{t,w})_{w \in \mathcal{W}}$ with $U_{t,w} \sim \text{Unif}(0,1)$.
- **Decoder:** Select the token w_t that maximizes

$$w_t = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}} \sim \boldsymbol{P}_t.$$

Gumbel-max Watermark [Aaronson, 2023]

Generation

- **Pseudorandomness:** $\zeta_t = (U_{t,w})_{w \in \mathcal{W}}$ with $U_{t,w} \sim \text{Unif}(0,1)$.
- **Decoder:** Select the token w_t that maximizes

$$w_t = \arg\max_{w \in \mathcal{W}} \frac{\log U_{t,w}}{P_{t,w}} \sim \boldsymbol{P}_t.$$

Detection [Li et al., 2025b]

- **Pivotal statistic:** Quantifies the evidence that each token is watermarked:

$$Y_t := Y(w_t, \zeta_t) = U_{t,w_t}.$$

- **Hypothesis testing:**

$$H_0 : \ Y_t \overset{i.i.d.}{\sim} \text{Unif}[0,1] \qquad \text{vs.} \qquad H_1 : \ Y_t \text{ tends to be larger.}$$

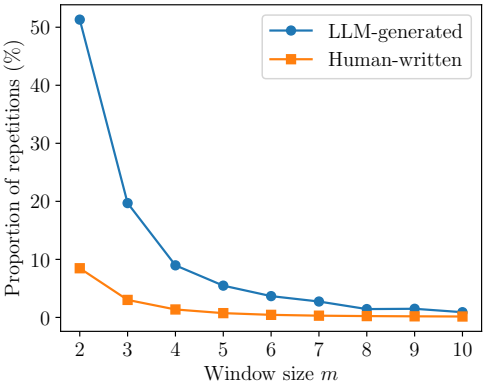
- **Test statistic:** $S_n := \sum_{t=1}^{n} h(Y_t)$ with h increasing. Reject H_0 if $S_n > \tau$.

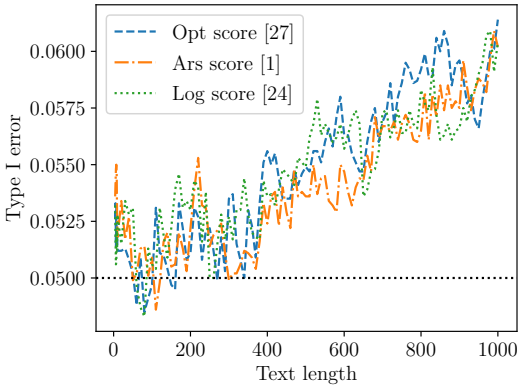
Pseudorandom Collision: Theory v.s. Practice

- **Theory:** Perfect pseudorandomness treats $\zeta_{1:n}$ as i.i.d., often unrealistic.
- **Practice:** Each $\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \mathrm{Key})$ uses an m-token text window.

Pseudorandom Collision: Theory v.s. Practice

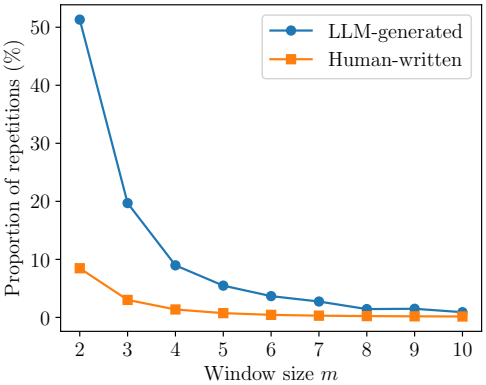
- **Theory:** Perfect pseudorandomness treats $\zeta_{1:n}$ as i.i.d., often unrealistic.
- **Practice:** Each $\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \mathrm{Key})$ uses an m-token text window.

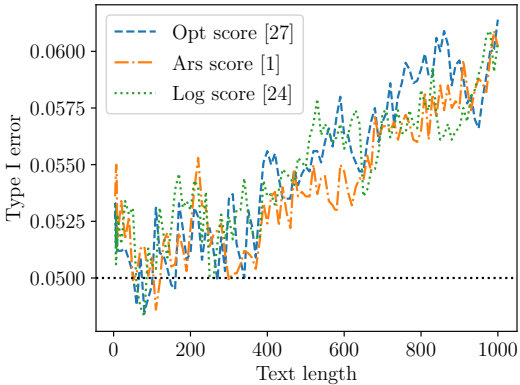




Pseudorandom Collision: Theory v.s. Practice

- **Theory:** Perfect pseudorandomness treats $\zeta_{1:n}$ as i.i.d., often unrealistic.
- **Practice:** Each $\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \mathrm{Key})$ uses an m-token text window.

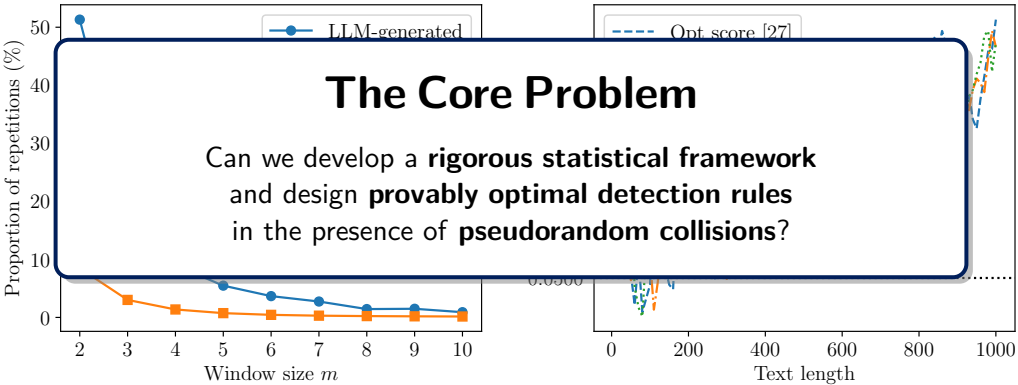




- **Engineering Fix:** Delete the duplicates [Fernandez et al., 2023, Wu et al., 2024a].

Pseudorandom Collision: Theory v.s. Practice

- **Theory:** Perfect pseudorandomness treats $\zeta_{1:n}$ as i.i.d., often unrealistic.
- **Practice:** Each $\zeta_t = \mathcal{A}(w_{(t-m):(t-1)}, \mathrm{Key})$ uses an m-token text window.

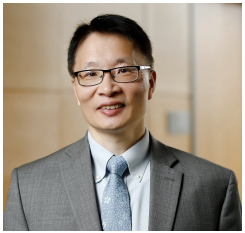


The Core Problem

Can we develop a **rigorous statistical framework** and design **provably optimal detection rules** in the presence of **pseudorandom collisions**?

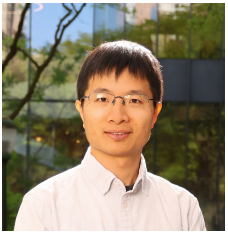
- **Engineering Fix:** Delete the duplicates [Fernandez et al., 2023, Wu et al., 2024a].

Collaborators



Tony Cai
UPenn

Qi Long
UPenn

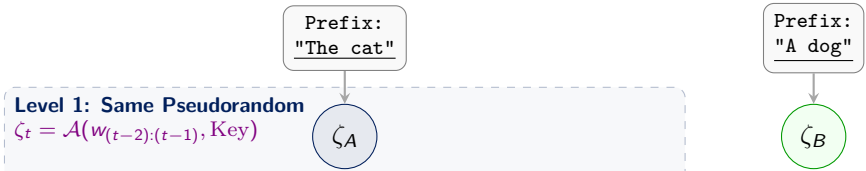


Weijie Su
UPenn

Garrett G. Wen
Yale

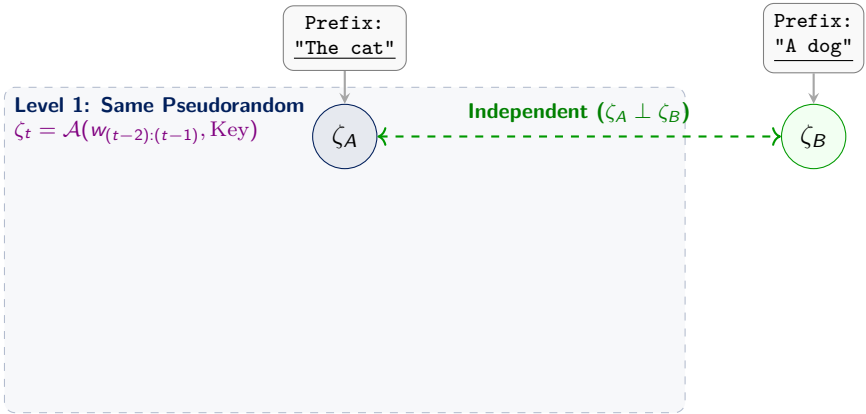
A Collision Example

Input Text ($m = 2$): "...The cat sat$_{t=1}$ ... [text] ... The cat sat$_{t=5}$ ... [text] ... The cat ran$_{t=8}$ ... A dog barked$_{t=9}$..."



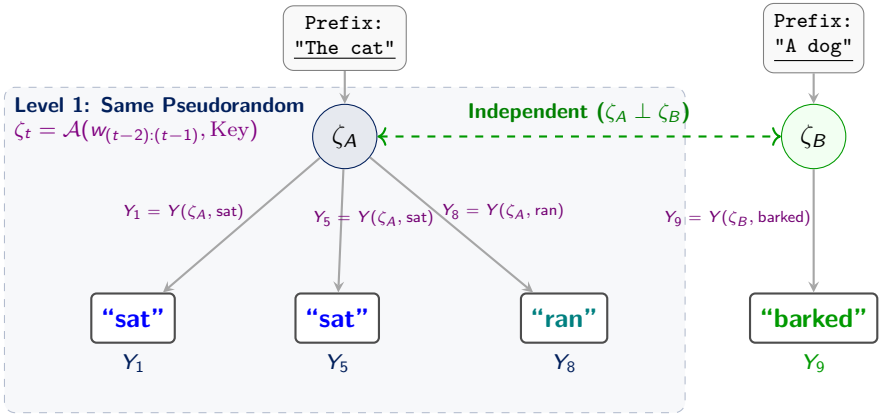
A Collision Example

Input Text ($m = 2$): "...The cat $\text{sat}_{t=1}$ ... [text] ... The cat $\text{sat}_{t=5}$ ... [text] ... The cat $\text{ran}_{t=8}$ ... A dog $\text{barked}_{t=9}$..."



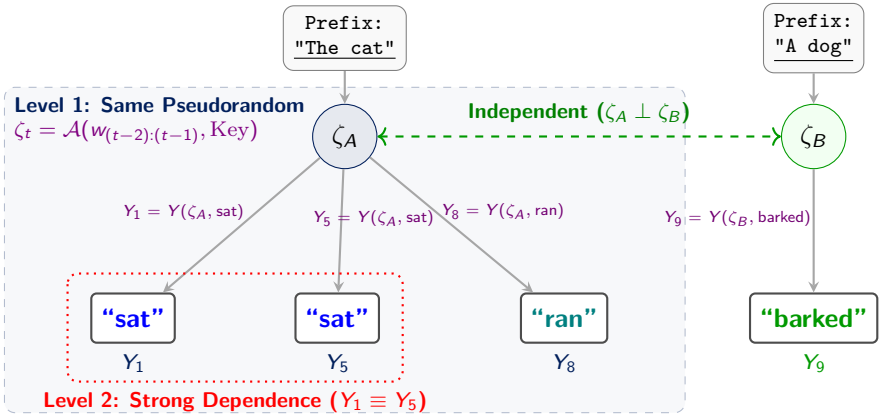
A Collision Example

Input Text ($m = 2$): "...The cat sat$_{t=1}$ ... [text] ... The cat sat$_{t=5}$ ... [text] ... The cat ran$_{t=8}$ ... A dog barked$_{t=9}$..."



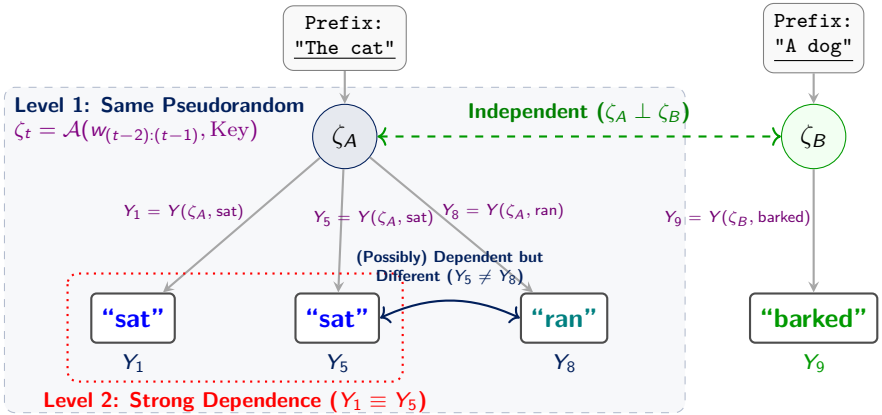
A Collision Example

Input Text ($m = 2$): "...The cat $\text{sat}_{t=1}$ ... [text] ... The cat $\text{sat}_{t=5}$ ... [text] ... The cat $\text{ran}_{t=8}$ ... A dog $\text{barked}_{t=9}$..."



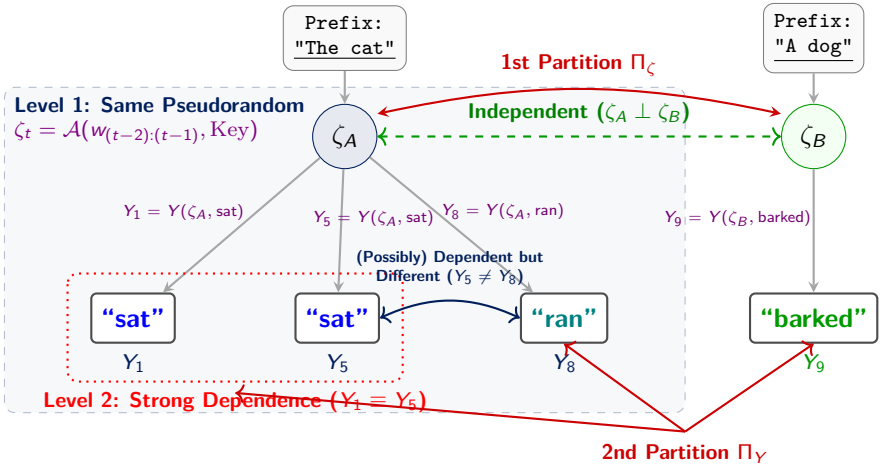
A Collision Example

Input Text ($m = 2$): "...The cat sat$_{t=1}$ ... [text] ... The cat sat$_{t=5}$ ... [text] ... The cat ran$_{t=8}$ ... A dog barked$_{t=9}$..."

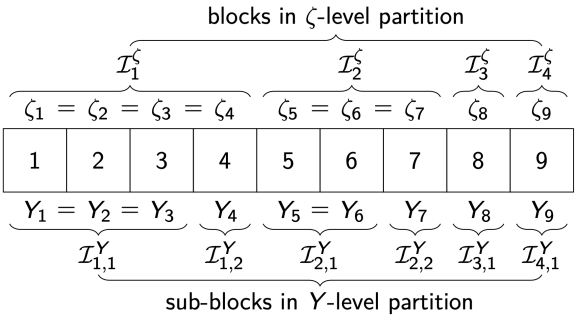


A Collision Example

Input Text ($m = 2$): "...The cat $\texttt{sat}_{t=1}$ ... [text] ... The cat $\texttt{sat}_{t=5}$ ... [text] ... The cat $\texttt{ran}_{t=8}$ ... A dog $\texttt{barked}_{t=9}$..."

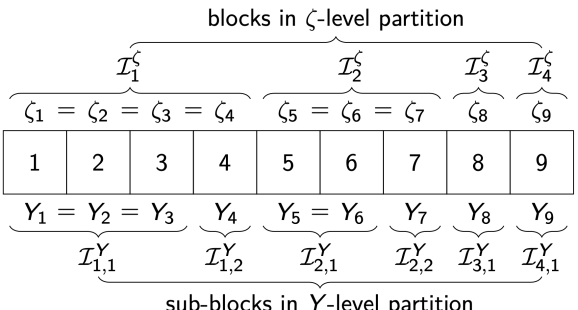


Two-Layer Partition and Minimal Units



- A **two-layer partition** (Π_ζ, Π_Y) characterizes how repetitions arise.
- Independence still holds **across units** of Π_ζ or Π_Y.

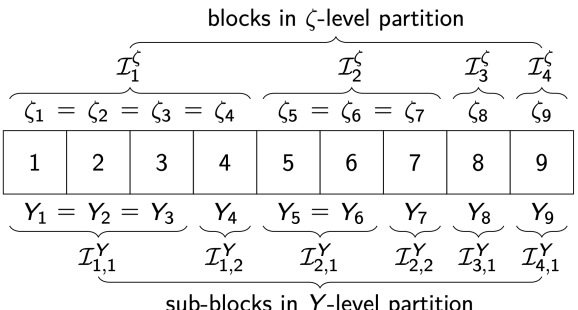
Two-Layer Partition and Minimal Units



- A **two-layer partition** (Π_ζ, Π_Y) characterizes how repetitions arise.
- Independence still holds **across units** of Π_ζ or Π_Y.
- **Minimal units:** The finest partition Π (either Π_ζ or Π_Y) so that

$$\mathbb{P}(Y_{1:n}|\Pi) = \prod_{\mathcal{V}\in\Pi} \mathbb{P}(Y_{\mathcal{V}})$$

Two-Layer Partition and Minimal Units



- A **two-layer partition** (Π_ζ, Π_Y) characterizes how repetitions arise.
- Independence still holds **across units** of Π_ζ or Π_Y.
- **Minimal units:** The finest partition Π (either Π_ζ or Π_Y) so that

$$\mathbb{P}(Y_{1:n}|\Pi) = \prod_{\mathcal{V}\in\Pi} \mathbb{P}(Y_{\mathcal{V}})$$

- For Gumbel-max, minimal units are blocks of the Y-partition.

Hypothesis Testing via Minimal Units

- $Y_{\mathcal{V}} = (Y_t)_{t \in \mathcal{V}}$ collects pivotal statistics in a minimal unit $\mathcal{V}$.
- By construction, $Y_{\mathcal{V}_i} \perp Y_{\mathcal{V}_j}$ for $i \neq j$.

Hypothesis Testing via Minimal Units

- $Y_{\mathcal{V}} = (Y_t)_{t \in \mathcal{V}}$ collects pivotal statistics in a minimal unit $\mathcal{V}$.
- By construction, $Y_{\mathcal{V}_i} \perp Y_{\mathcal{V}_j}$ for $i \neq j$.
- Hypothesis testing under **pseudorandom collisions**:

$$H_0 : Y_{\mathcal{V}} \sim \mu_{0,\mathcal{V}} \text{ (known)}, \forall \mathcal{V} \qquad H_1 : Y_{\mathcal{V}} \sim \mu_{1,P_{\mathcal{V}}}, \forall \mathcal{V}$$

- Independent units with known null distribution $\Rightarrow$ valid test statistics with rigorous Type I error control.

Hypothesis Testing via Minimal Units

- $Y_{\mathcal{V}} = (Y_t)_{t \in \mathcal{V}}$ collects pivotal statistics in a minimal unit $\mathcal{V}$.
- By construction, $Y_{\mathcal{V}_i} \perp Y_{\mathcal{V}_j}$ for $i \neq j$.
- Hypothesis testing under **pseudorandom collisions**:

$$H_0 : Y_{\mathcal{V}} \sim \mu_{0,\mathcal{V}} \text{ (known)}, \forall \mathcal{V} \qquad H_1 : Y_{\mathcal{V}} \sim \mu_{1,P_{\mathcal{V}}}, \forall \mathcal{V}$$

- Independent units with known null distribution $\Rightarrow$ valid test statistics with rigorous Type I error control.
- For Gumbel-max,
  - $\mathcal{V}$ groups tokens with identical Y_t's, so $Y_{\mathcal{V}}$ is $|\mathcal{V}|$ times the unique value in the group.
  - The test score $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$ assigns a score function $h_{\mathcal{V}}$ for each $\mathcal{V}$.

Optimal Score Functions $\{h_\nu\}_\nu$

Idea

Optimal score functions maximize detection power against the worst-case token distribution [Li et al., 2025b,a].

Optimal Score Functions $\{h_{\mathcal{V}}\}_{\mathcal{V}}$

Idea

Optimal score functions maximize detection power against the worst-case token distribution [Li et al., 2025b,a].

Finite-Time Least-Favorable Efficiency

Suppose all $\boldsymbol{P}_t \in \mathcal{P}$ and consider $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$. We measure detection power of S_n by the worst-case exponential decay rate of the Type II error:

$$R_{n,\mathcal{P}}(S_n) := -\frac{1}{|\{\mathcal{V}\}|} \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \underbrace{\mathbb{P}_{1,P_{\mathcal{V}}}(S_n \leq \gamma_{n,\alpha})}_{\text{Type II error}}$$

Optimal Score Functions $\{h_{\mathcal{V}}\}_{\mathcal{V}}$

Idea

Optimal score functions maximize detection power against the worst-case token distribution [Li et al., 2025b,a].

Finite-Time Least-Favorable Efficiency

Suppose all $\boldsymbol{P}_t \in \mathcal{P}$ and consider $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$. We measure detection power of S_n by the worst-case exponential decay rate of the Type II error:

$$R_{n,\mathcal{P}}(S_n) := -\frac{1}{|\{\mathcal{V}\}|} \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \underbrace{\mathbb{P}_{1,P_{\mathcal{V}}}(S_n \leq \gamma_{n,\alpha})}_{\text{Type II error}}$$

Minimax Problem

$$S_n^\star = \arg\max_{S_n} R_{n,\mathcal{P}}(S_n) = \arg \min_{\{h_{\mathcal{V}}\}_{\mathcal{V}}} \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{P}_{1,P_{\mathcal{V}}}(S_n \leq \gamma_{n,\alpha}).$$

Decoupled Minimax Optimization

Key Observation

Consider the test statistic $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$. Under mild conditions, the global efficiency is approximately the **sum of local efficiencies**:

$$R_{n,\mathcal{P}}(S_n) \approx -\frac{1}{|\{\mathcal{V}\}|} \sum_{\mathcal{V}} \underbrace{\left(\mathbb{E}_0[h_{\mathcal{V}}] + \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{E}_{1,P_{\mathcal{V}}}[e^{-h_{\mathcal{V}}}]\right)}_{\textbf{Local efficiency of } h_{\mathcal{V}}}$$

Decoupled Minimax Optimization

Key Observation

Consider the test statistic $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$. Under mild conditions, the global efficiency is approximately the **sum of local efficiencies**:

$$R_{n,\mathcal{P}}(S_n) \approx -\frac{1}{|\{\mathcal{V}\}|} \sum_{\mathcal{V}} \underbrace{\left(\mathbb{E}_0[h_{\mathcal{V}}] + \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{E}_{1,P_{\mathcal{V}}}[e^{-h_{\mathcal{V}}}]\right)}_{\textbf{Local efficiency of } h_{\mathcal{V}}}$$

- The problem **decouples** allows us to optimize each local efficiency **separately**:

$$h^\star_{\mathcal{V}} = \arg\min_{h_{\mathcal{V}}} \left\{ \mathbb{E}_0[h_{\mathcal{V}}] + \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{E}_{1,P_{\mathcal{V}}}[e^{-h_{\mathcal{V}}}] \right\}$$

- **Challenge:** the inner maximization is highly **non-convex and non-concave**.

Decoupled Minimax Optimization

Key Observation

Consider the test statistic $S_n = \sum_{\mathcal{V}} h_{\mathcal{V}}(Y_{\mathcal{V}})$. Under mild conditions, the global efficiency is approximately the **sum of local efficiencies**:

$$R_{n,\mathcal{P}}(S_n) \approx -\frac{1}{|\{\mathcal{V}\}|} \sum_{\mathcal{V}} \underbrace{\left(\mathbb{E}_0[h_{\mathcal{V}}] + \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{E}_{1,P_{\mathcal{V}}}[e^{-h_{\mathcal{V}}}]\right)}_{\textbf{Local efficiency of } h_{\mathcal{V}}}$$

- The problem **decouples** allows us to optimize each local efficiency **separately**:

$$h^\star_{\mathcal{V}} = \arg\min_{h_{\mathcal{V}}} \left\{ \mathbb{E}_0[h_{\mathcal{V}}] + \sup_{P_{\mathcal{V}} \subseteq \mathcal{P}} \log \mathbb{E}_{1,P_{\mathcal{V}}}[e^{-h_{\mathcal{V}}}] \right\}$$

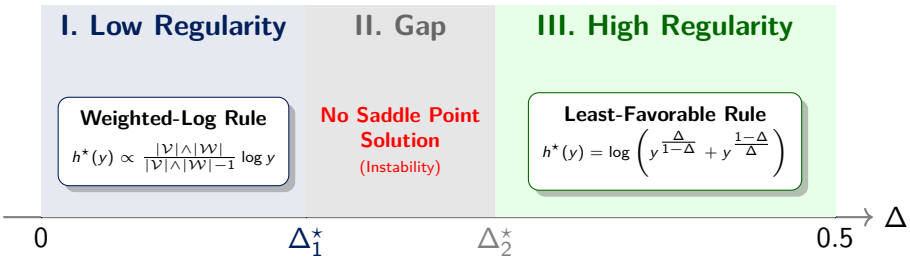
- **Challenge:** the inner maximization is highly **non-convex and non-concave**.
- **Our Strategy:** find saddle points by identifying necessary conditions and checking sufficent conditions.

Saddle Point Trichotomy

- Let $\mathcal{P}_\Delta := \{\boldsymbol{P} : \max_w P_w \leq 1 - \Delta\}$ be the prior class.
- The existence and form of optimality depend on Δ (as well as $|\mathcal{V}|$ and $|\mathcal{W}|$).

Saddle Point Trichotomy

- Let $\mathcal{P}_\Delta := \{\boldsymbol{P} : \max_w P_w \leq 1 - \Delta\}$ be the prior class.
- The existence and form of optimality depend on Δ (as well as $|\mathcal{V}|$ and $|\mathcal{W}|$).



Regime I (Low Δ, deterministic):

- The alternative concentrates on a single token.

Regime III (High Δ, more random):

- The optimal score depends **only** on Δ, ignoring $|\mathcal{V}|$. Recover previous results [Li et al., 2025b].

Theory Meets Practice

Engineering Heuristic

Discard duplicates.
Keep only unique Y.

$\approx$

Our Optimal Rule (Regime III)

This heuristic is statistically optimal.
It doesn't depend on $|\mathcal{V}|$ for large Δ.

Take-away

We provide a theoretical justification for this heuristic.
It is (sometimes) minimax optimal.

Theory Meets Practice

Engineering Heuristic

Discard duplicates.
Keep only unique Y.

$\approx$

Our Optimal Rule (Regime III)

This heuristic is statistically optimal.
It doesn't depend on $|\mathcal{V}|$ for large Δ.

Take-away

We provide a theoretical justification for this heuristic.
It is (sometimes) minimax optimal.

Why?

- Cross-block independence implies duplicates carry **no additional information**.
- **High uncertainty (large Δ):** duplicates mainly arise from seed collisions.
- **Low uncertainty (small Δ):** duplicates reflect a strong token preference, so repetition provides additional evidence.

Simulation

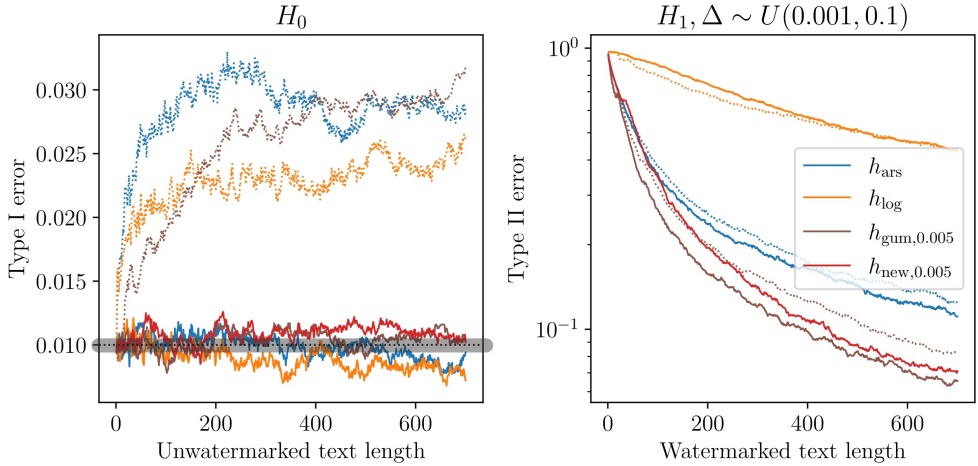


Figure: Type I errors (left) and Type II errors (right, log scale) on synthetic datasets for Gumbel-max. **Solid**: deduplicated. **Dotted**: non-deduplicated.

How to Derive the Result

- For Gumbel-max, a minimal unit $\mathcal{V} = \{t_1, \ldots, t_k\}$ contains all identical Y_t's.

How to Derive the Result

- For Gumbel-max, a minimal unit $\mathcal{V} = \{t_1, \ldots, t_k\}$ contains all identical Y_t's.
- The null distribution of the unique Y is Unif$[0, 1]$.
- The alternative distribution of the unique Y is

$$F_{\boldsymbol{S}}(y) = \mathbb{P}_1(Y \le y) = \frac{\sum_w S_w y^{1/S_w}}{\sum_w S_w}$$

where:

$$\boldsymbol{S} = (S_1, \ldots, S_{|\mathcal{W}|}) \quad \text{with} \quad S_w = \left(1 + \sum_{w' \neq w} \max_{t \in \mathcal{V}} \frac{P_{t,w'}}{P_{t,w}}\right)^{-1}.$$

How to Derive the Result

- For Gumbel-max, a minimal unit $\mathcal{V} = \{t_1, \ldots, t_k\}$ contains all identical Y_t's.
- The null distribution of the unique Y is Unif$[0, 1]$.
- The alternative distribution of the unique Y is

$$F_{\boldsymbol{S}}(y) = \mathbb{P}_1(Y \leq y) = \frac{\sum_w S_w y^{1/S_w}}{\sum_w S_w}$$

where:

$$\boldsymbol{S} = (S_1, \ldots, S_{|\mathcal{W}|}) \quad \text{with} \quad S_w = \left(1 + \sum_{w' \neq w} \max_{t \in \mathcal{V}} \frac{P_{t,w'}}{P_{t,w}}\right)^{-1}.$$

Key Idea

Solve the problem in terms of the new reparameterization of the $\boldsymbol{S}$ vector.

How to Derive the Result

- Let $\mathcal{D}_\Delta$ collect all valid $\boldsymbol{S}$ vectors.
- Seek the optimal score $h^\star$ by solving

$$\min_{h} \max_{\boldsymbol{S} \in \mathcal{D}_\Delta} \left(\mathbb{E}_0[h(Y)] + \log \mathbb{E}_{F_{\boldsymbol{S}}}[e^{-h(Y)}] \right)$$

How to Derive the Result

- Let $\mathcal{D}_\Delta$ collect all valid $\boldsymbol{S}$ vectors.
- Seek the optimal score $h^\star$ by solving

$$\min_{h} \max_{\boldsymbol{S} \in \mathcal{D}_\Delta} \left(\mathbb{E}_0[h(Y)] + \log \mathbb{E}_{F_{\boldsymbol{S}}}[e^{-h(Y)}] \right)$$

- **Difficulty:** $\mathcal{D}_\Delta$ is highly non-convex.

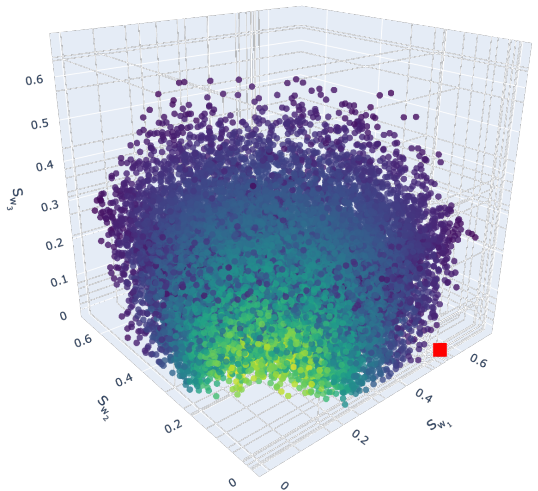


Illustration of $\mathcal{D}_\Delta$ for $|\mathcal{V}| = 2$, $|\mathcal{W}| = 3$, $\Delta = 0.3$.

How to Derive the Result

- Let $\mathcal{D}_\Delta$ collect all valid $\boldsymbol{S}$ vectors.
- Seek the optimal score $h^\star$ by solving

$$\min_{h} \max_{\boldsymbol{S} \in \mathcal{D}_\Delta} \left(\mathbb{E}_0[h(Y)] + \log \mathbb{E}_{F_{\boldsymbol{S}}}[e^{-h(Y)}] \right)$$

- **Difficulty:** $\mathcal{D}_\Delta$ is highly non-convex.
- **Key Idea:** Use schur-convexity w.r.t. $\boldsymbol{S}$ (convex on the plane $\sum_w S_w = C$ for any C) and focus on convex hull of $\mathcal{D}_\Delta$.

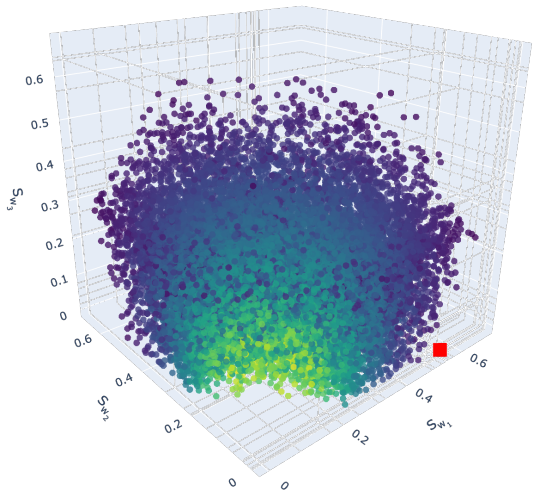


Illustration of $\mathcal{D}_\Delta$ for $|\mathcal{V}| = 2$, $|\mathcal{W}| = 3$, $\Delta = 0.3$.

Conclusion Remark

- A statistical framework for collision-induced dependence via **two-level partitions**.
- Closed-form minimax rules for **Gumbel-max** (this talk) and **inverse-transform** (our paper) watermarks.
- Solves the resulting **non-convex minimax problems**.
- Provides a theoretical justification for the **engineering heuristic** used in practice.

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.

G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, 2023.

L. Bednarczyk, D. Reichenpfader, C. Gaudet-Blavignac, A. K. Ette, J. Zaghir, Y. Zheng, A. Bensahla, M. Bjelogrlic, and C. Lovis. Scientific evidence for clinical text summarization using large language models: Scoping review. *Journal of Medical Internet Research*, 27:e68998, 2025.

Y. Chen, Z. Hu, Y. Wu, R. Chen, Y. Jin, M. Zhan, C. Xie, W. Chen, and H. Huang. Enhancing privacy in biosecurity with watermarked protein design. *Bioinformatics*, page btaf141, 2025.

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

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.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5791–5808, 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.

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 for large language models under human edits. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkaf056, 2025a.

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, 2025b.

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*, 2025c.

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.

A. Mehra, S. Verma, and N. Rajpal. AI-assisted clinical documentation: Transforming healthcare practices through automated intelligence, enhanced data capture, and improved patient-centered care. *International Journal of Nursing Leadership and Education*, 1(2), 2026.

E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR, 2023.

References IV

S. W. Perkins, J. C. Muste, T. Alam, and R. P. Singh. Improving clinical documentation with artificial intelligence: A systematic review. *Perspectives in health information management*, 21(2), 2024.

V. C. Pezoulas, D. I. Zaridis, E. Mylona, C. Androutsos, K. Apostolidis, N. S. Tachos, and D. I. Fotiadis. Synthetic data generation methods in healthcare: A review on open-source tools and methods. *Computational and structural biotechnology journal*, 23:2892–2910, 2024.

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.

I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

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, R. Chen, Z. Hu, Y. Chen, J. Guo, H. Zhang, and H. Huang. Distortion-free watermarks are not truly distortion-free under watermark key collisions. *arXiv preprint arXiv:2406.02603*, 2024a.

Y. Wu, Z. Hu, J. Guo, H. Zhang, and H. Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Forty-first International Conference on Machine Learning*, 2024b. URL `https://openreview.net/forum?id=c8qWiNiqRY`.

Y. Xie, X. Li, T. Mallick, W. Su, and R. Zhang. Debiasing watermarks for large language models via maximal coupling. *Journal of the American Statistical Association*, (just-accepted):1–21, 2025.

References V

C. Yan, Z. Zhang, S. Nyemba, and Z. Li. Generating synthetic electronic health record data using generative adversarial networks: Tutorial. *Jmir Ai*, 3:e52615, 2024.

X. Yang, W. Cheng, L. Petzold, W. Y. Wang, and H. Chen. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. *arXiv preprint arXiv:2305.17359*, 2023.

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

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

X. Zhao, L. Li, and Y.-X. Wang. Permute-and-Flip: An optimally stable and watermarkable decoder for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=YyVVicZ32M`.