



Watermarking

Mausam

(Slides inspired by Scott Aaronson, Sumanth Dathathri, Xuandong Zhao, Yu-Xiang Wang, Lei Li)



AI & Law

India's top court angry after junior judge cites fake AI-generated orders

Supreme Court releases draft rules for use of AI in courts, allows use with disclosure [04.06.2026]

“AI systems would be barred from deciding cases, determining bail, passing sentences, evaluating witness credibility, predicting future conduct or influencing judicial deliberations.”

<https://www.ndtv.com/feature/us-judges-kicks-off-lawyers-from-both-sides-for-using-ai-cancels-trial-comedy-of-errors-11620333>

<https://www.bbc.com/news/articles/c178zzw780xo>

<https://legaleagleweb.com/newsdetail.aspx?newsid=9520>

US Judge Kicks Off Lawyers From Both Sides For Using AI, Cancels Trial: 'Comedy Of Errors'

US District Judge Sharion Aycock ordered the removal of all four lawyers from the contract dispute and issued financial penalties.

Edited by: [Abhinav Singh](#) | [Feature](#) | Jun 11, 2026 07:52 IST ⓘ

Read Time: 3 mins

Add **NDTV** as a

Share



The lawyers on both sides of a federal court case in Mississippi were caught using AI.



Fake News



- Erosion of public trust
- Real-world Harm
 - Political narratives
 - False advertising
 - Communal tensions...

AI-generated slop is quietly conquering the internet. Is it a threat to journalism or a problem that will fix itself?

In February, Wired's Kate Knibbs reported on a Serbian entrepreneur who has bought abandoned news sites, filled them with AI-generated content and pocketed ad revenue.

According to NewsGuard, more than 2,089 AI-generated news sites are currently operating, publishing content in 16 languages.

**THE
ORIGINAL_**

**REALITY BLURS AS A.I.-GENERATED NEWS CLIPS
RISK SPREADING MISINFORMATION IN NEW WAYS**



<https://reutersinstitute.politics.ox.ac.uk/news/ai-generated-slop-quietly-conquering-internet-it-threat-journalism-or-problem-will-fix-itself>

<https://www.polytechnique-insights.com/en/columns/digital/how-artificial-intelligence-is-transforming-the-information-ecosystem/>

<https://www.youtube.com/watch?v=SRA4brHXBBQ>



Intellectual Property (of AI)

- Several users on Reddit claim that DeepSeek responds with “I am ChatGPT” or “I am programmed by OpenAI” or similar...

DeepSeek: Is It A Stolen ChatGPT?

OpenAI says China's DeepSeek trained its AI by distilling US models, memo

In the memo sent to the U.S. House Select Committee on Strategic Competition between the U.S. and the Chinese Communist Party on Thursday, OpenAI said: "We have observed accounts associated with DeepSeek employees developing methods to circumvent OpenAI's access restrictions and access models through obfuscated third-party routers and other ways that mask their source."

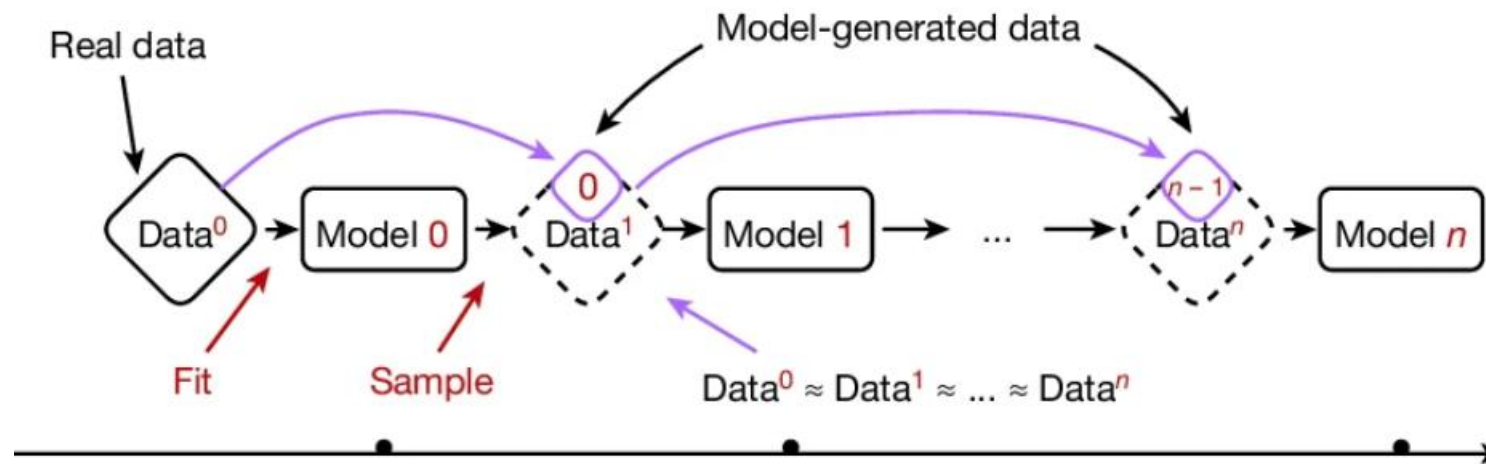
"We also know that DeepSeek employees developed code to access U.S. AI models and obtain outputs for distillation in programmatic ways," the memo added.

<https://medium.com/@jankammerath/deepseek-is-it-a-stolen-chatgpt-a805b586b24a>

<https://www.reuters.com/world/china/openai-accuses-deepseek-distilling-us-models-gain-advantage-bloomberg-news-2026-02-12/>



Model Collapse when Retraining on Generated Data



Models perform worse when trained on their own synthetically generated data

Over generations: models forget the true underlying data distribution
(tails of original content distribution disappear)

<https://www.nature.com/articles/s41586-024-07566-y>

Misuse of AI by Students

1. **Plagiarism of written content**
2. **Artificial artwork**
3. **Computational AI**
4. **Translation**
5. **Bypassing plagiarism software**
6. **Use in unsupervised assessments**
7. **Programming**

How Students Are Using — and Misusing — AI in Higher Education

From Trojan horses to AI-proof exams: how professors are tackling students' AI use

Academics worldwide are rethinking their assessment approaches — from developing exams that artificial-intelligence tools can't pass to sneaking hidden AI prompts into tasks to catch cheating students.

<https://www.digitaleducationcouncil.com/dec-insights/how-students-are-using-and-misusing-ai-in-higher-education>

<https://www.nature.com/articles/d41586-026-02370-2>

<https://inspera.com/ai/examples-of-ai-misuse-in-education/>



AI use in Paper Writing

- “AI written papers with AI written reviews and people using AI to read those AI papers
if anything I've learned that people are lazy”

AI Slop Is Flooding Academic Journals. A Top Journal Measured

A new study from the journal's editorial team finds that AI-generated manuscripts are harder to read, more jargon laden and more likely to be rejected than those written by

AI research in question as author claims to have written over 100 papers on AI that one expert calls a 'disaster'

GPTZero finds 100 new hallucinations in NeurIPS 2025 accepted papers

<https://www.forbes.com/sites/johndrake/2026/04/30/ai-slop-is-flooding-academic-journals-a-top-journal-measured-it/>

<https://www.theguardian.com/technology/2025/dec/06/ai-research-papers>

https://www.reddit.com/r/MachineLearning/comments/1qno68x/advice_for_phd_students_in_this_ai_slop_paper_era/

<https://gptzero.me/news/neurips/>



AI Use in Reviewing



Submission Desk Rejected by Program Chairs

Desk Rejection by Program Chairs  Mar 2026, 

 Program Chairs, Senior Area Chairs, Area Chairs, Reviewers, Authors

Desk Reject Comments: The paper is desk rejected, because the reciprocal reviewer nominated for this paper  has violated the LLM reviewing policy. The reviewer was required to follow Policy A (no LLMs), but we have found a strong evidence that LLM was used in the preparation of at least one of their reviews. This is a breach of peer-review ethics and grounds for desk rejection. The decision is final.

To detect LLM use in reviewing, ICML organizers have watermarked the submitted PDFs. If a watermarked submission was provided as an input to an LLM, the LLM was instructed to produce multiple specific phrases in the review (the phrases were chosen at random for each submission from a large dictionary, and were sufficiently long, so that the likelihood of their organic occurrence in the review is vanishingly small). Using this method, we have detected that the reviewer has provided the submission PDF to an LLM and then used the LLM output at least as part of the review, both of these steps are in breach of Policy A. The suspected review was also manually checked to ensure that it incorporates the specific phrases in the review.

https://x.com/yuxiangw_cs/status/2034022354674987088/photo/1





Fundamental Question

How do we find that some text is generated by AI?



What to do? Embedded Instruction

- Teacher hides an instruction where the LLM text can be easily evaluated.
- Desiderata
 - Student is unaware of any AI detection by the teacher
 - Student misses the instruction
 - Student does not carefully read the generated answer
- Practically
 - You can only do it once!
 - Very easy for student to verify and clean up the answer



What to do? Third Party AI Detectors (Post-Facto)

AI-detection tools have made huge leaps forward – how good are they?

Scientists and publishers are testing a new breed of software that promises impressive accuracy at spotting AI-written text.



An AI detector that actually works.



AI detector made to
Preserve What's Human.

<https://www.nature.com/articles/d41586-026-02569-3>



What to do? Watermarking!

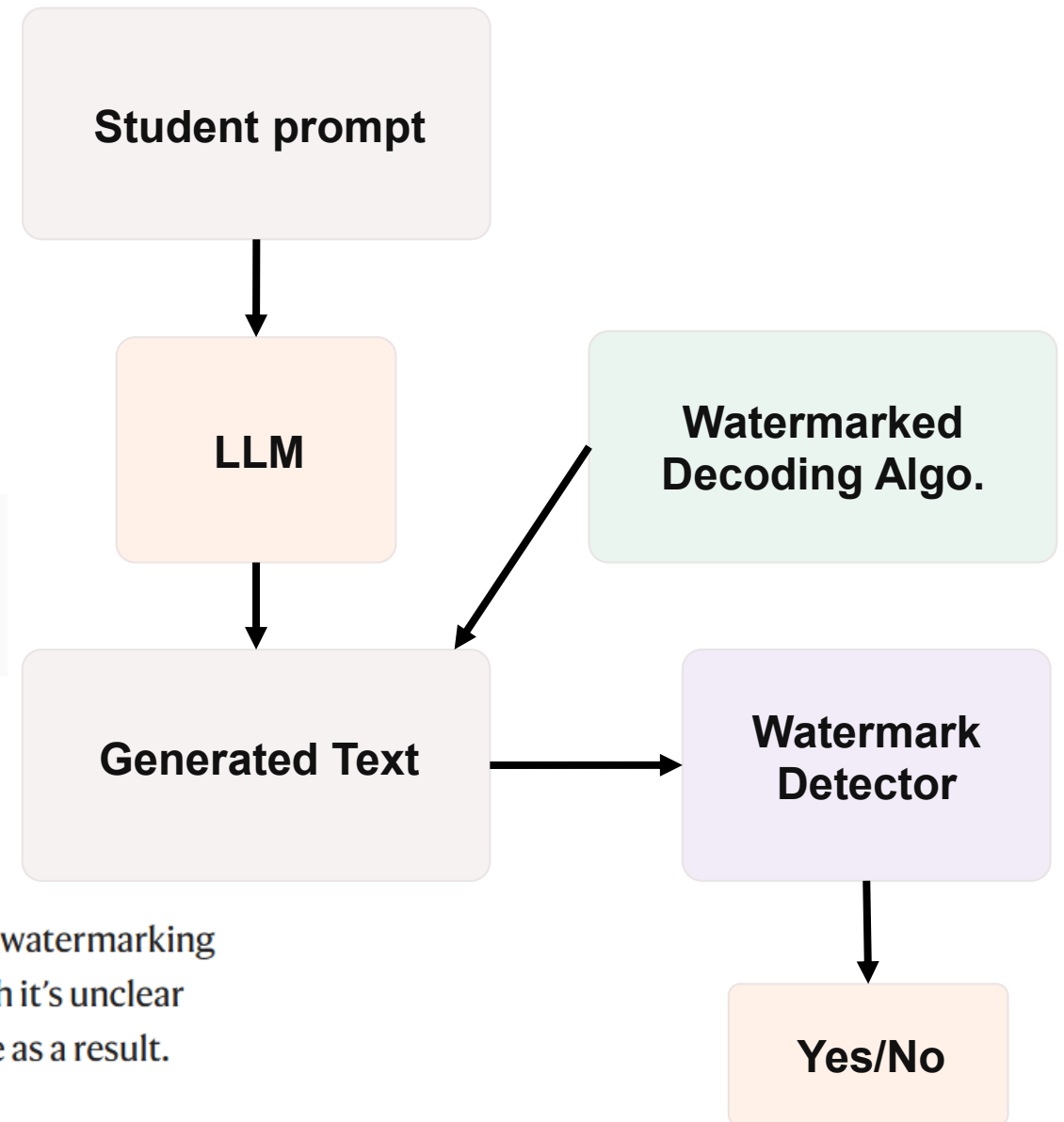
Goal: *generate text in a way that it can be detected as generated by that LLM without distorting meaning or readability*

The EU AI Act's transparency rules require that AI-generated content be identifiable, pushing labs toward watermarking, metadata tagging, or disclosure labels.

In another development, this month, US firm Anthropic said it would introduce watermarking into its Claude AI models to comply with the European Union's AI Act – although it's unclear [how easily editing could remove the marks](https://www.mindstudio.ai/blog/eu-ai-act-content-watermarking), or whether writers will avoid Claude as a result.

<https://www.nature.com/articles/d41586-026-02569-3>

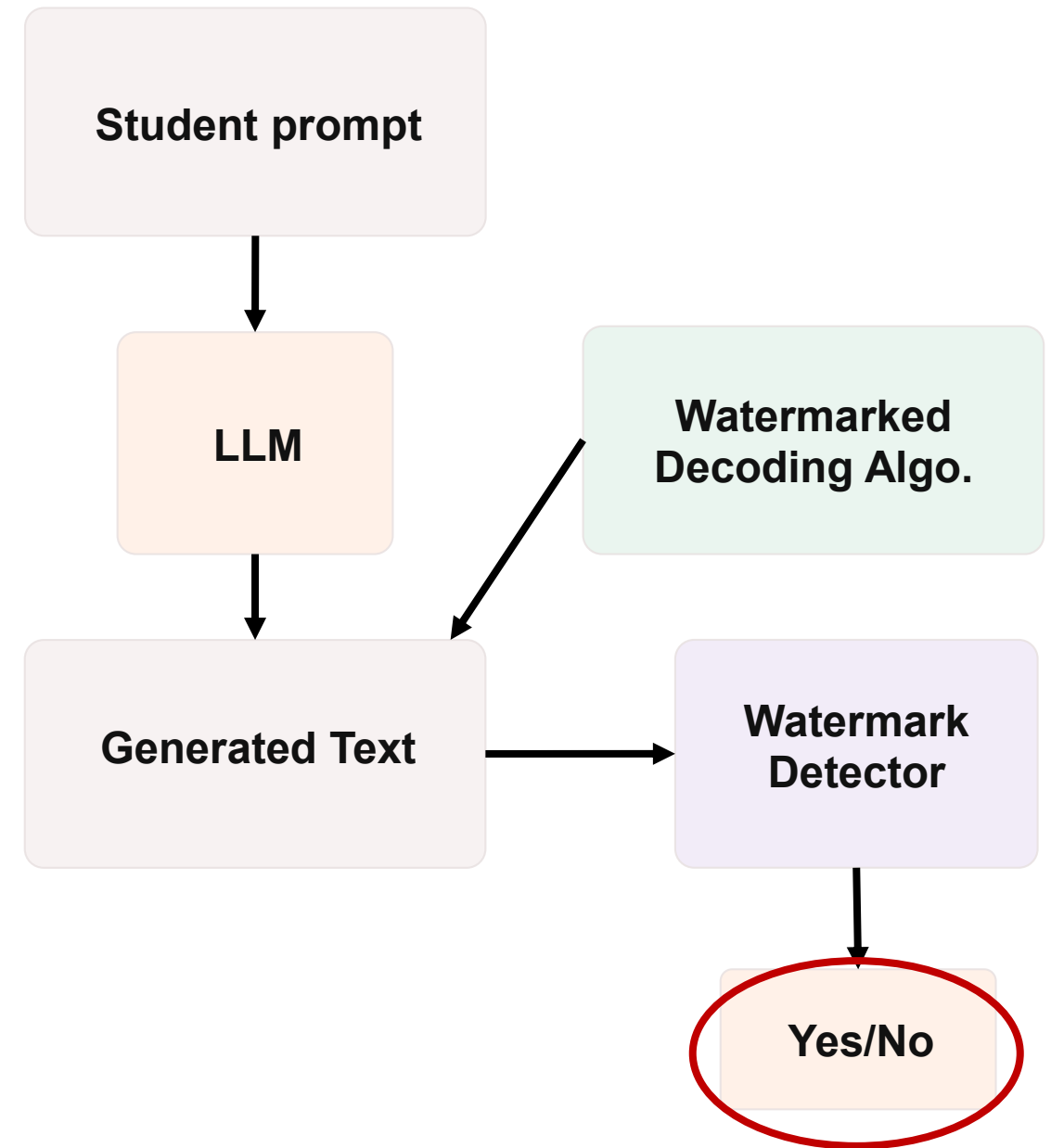
<https://www.mindstudio.ai/blog/eu-ai-act-content-watermarking>



Watermarking

Goal: *generate text in a way that it can be detected as generated by that LLM without distorting meaning or readability*

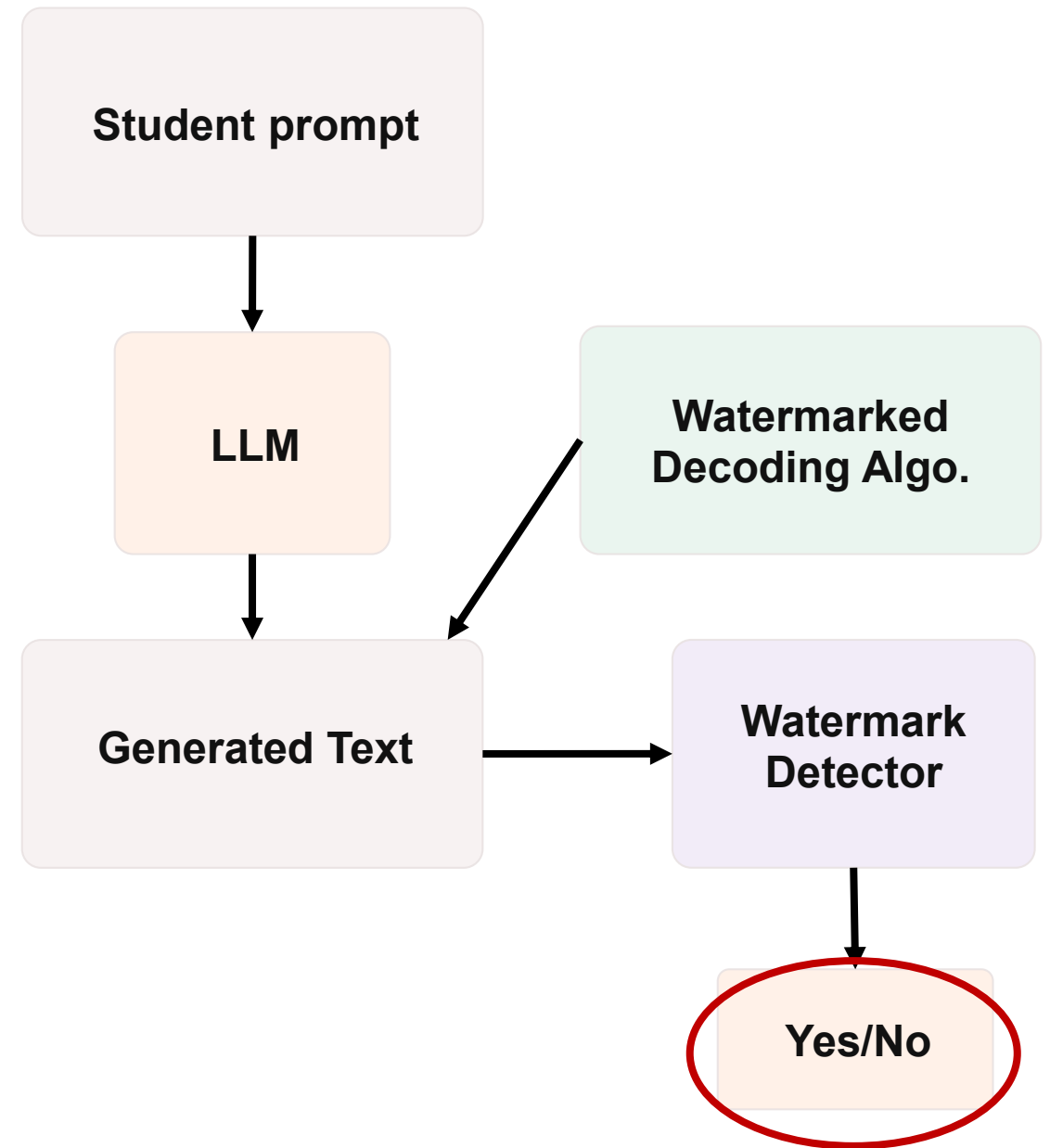
- Zero-bit Watermark
 - Yes/No
- Watermark Payload
 - Extra information embedded in watermark



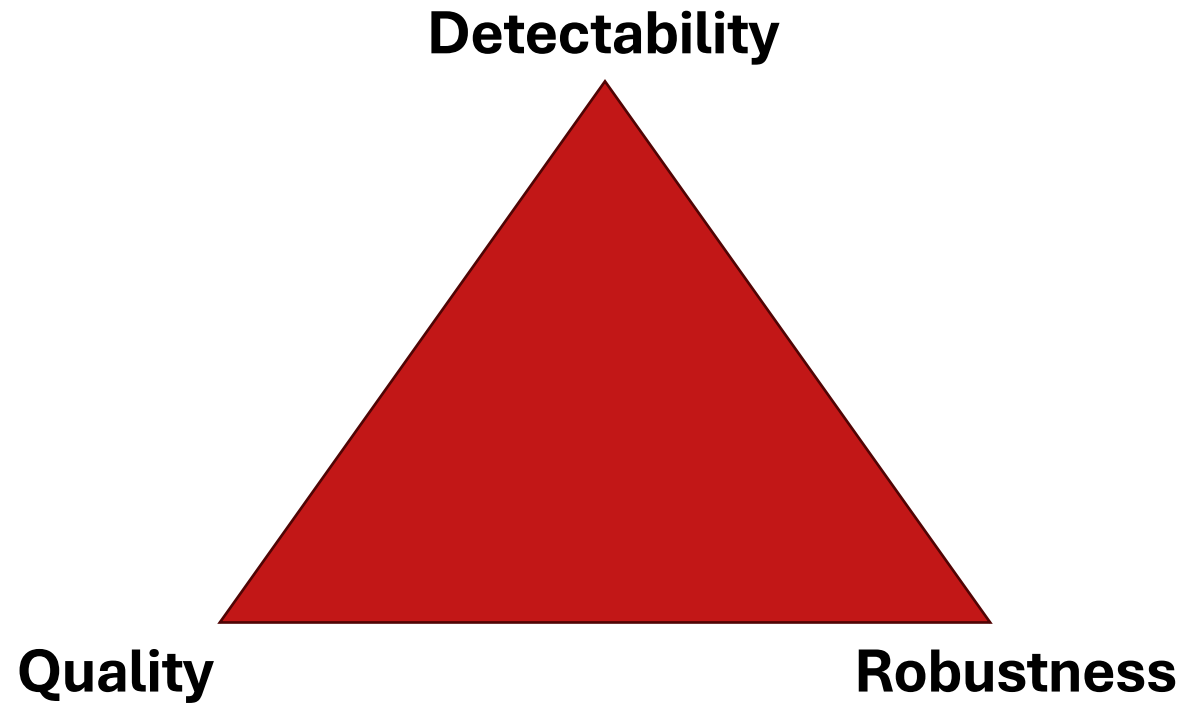
Watermarking: Attacks

Goal: *generate text in a way that it can be detected as generated by that LLM without distorting meaning or readability*

- Ease of removing Watermark
- Does student have knowledge of watermarking (algorithm)?
- Whitebox/Blackbox student



Tradeoffs in Watermarking



Related Task: Steganography

- Allows two parties to talk in secret without anyone else knowing a message exists.
- Example: Replace minor parts of a file (e.g., the least significant bits in an image) with secret data.

Feature	Steganography	Watermarking
Main Goal	Secret communication.	Copyright and ownership protection.
Robustness	Needs to avoid detection, but often breaks if the file is modified.	Needs high robustness to survive editing, compression, or cropping.
Payload Size	Usually larger because it carries a complete hidden message.	Usually smaller, containing an ID, logo, or signature.



Other Related Tasks

- **Cryptography:** Scrambles the contents of a message to make it unreadable, unlike steganography, which hides the message's existence entirely.
- **Fingerprinting (Traitor Tracing):** Embeds a unique, invisible identifier for each specific buyer into the media to trace the exact source of an unauthorized leak.
- **Anonymization:** Strips away personal or identifying metadata from files to protect user privacy before sharing them publicly.
- **Tamper Detection (Fragile Watermarking):** Embeds a delicate mark that breaks instantly if the file is altered, proving whether the media has been manipulated.
- **Chaffing and Winnowing:** Mixes authentic data packets (wheat) with fake data packets (chaff) to hide a message without using traditional encryption



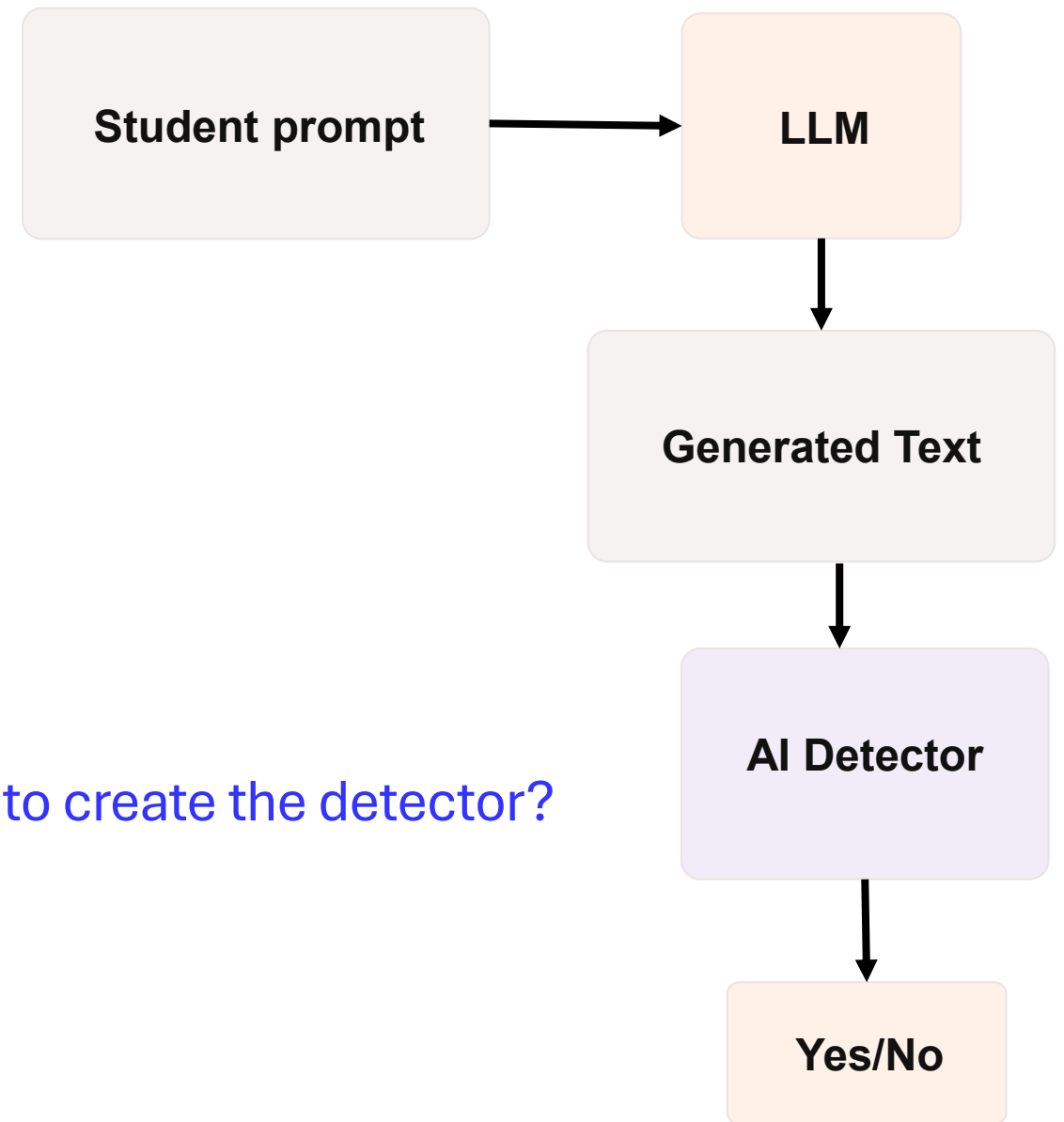


(Passive) AI Detectors

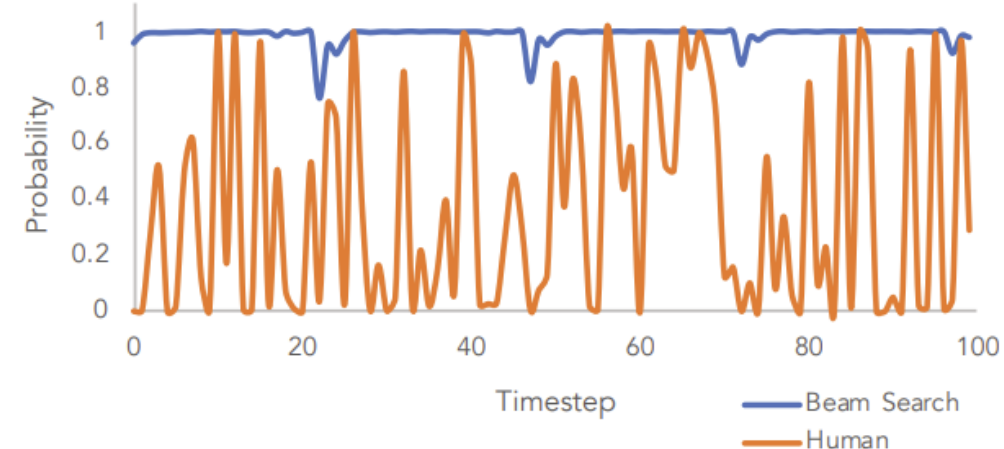


AI Detectors

How to create the detector?



GPTZero



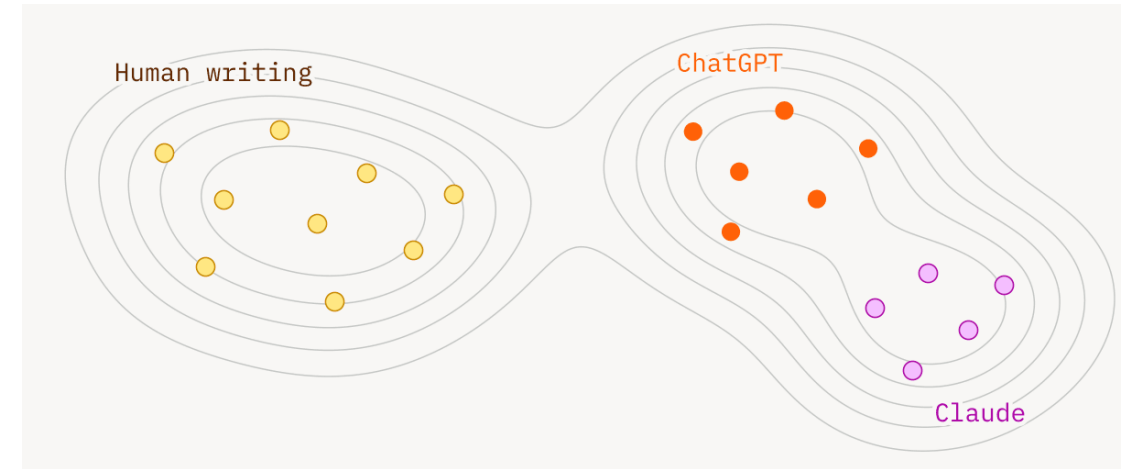
- **Perplexity:** AI generated text is generally low-perplexity (use a smaller LM, like GPT2)
Human-written text has a higher proportion of low probability tokens
- **Burstiness:** how uniform or constant the perplexity is over time compared to varied human-written text. (humans tend to mix short and long sentences)
- **Methodology:**
 - Classify (score) each sentence for its AI-ness.
 - Combine all sentence scores to create a document score.

<https://originality.ai/blog/gptzero-ai-content-detection-review>



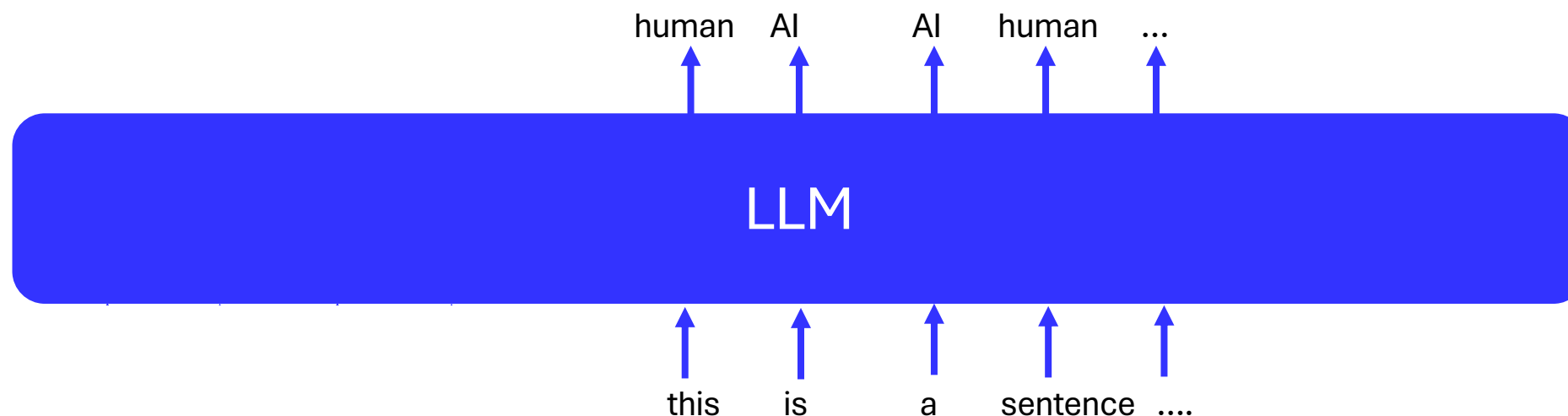
Pangram

- Clustering of Styles
- Stylistic patterns (subtle cues) via Roberta-large
 - AI content writers have favorite words, such as: *aspect, challenges, delve, enhance, tapestry, testament, authentic, comprehensive, crucial, significant, transformative*, and adverbs like *additionally* and *moreover*.
 - AI writing uses phrasing patterns like *as we [verb] the topic, it's important to note, not only but also, paving the way*, and *when it comes to*.
 - AI writing generally uses perfect spelling and grammar, and complex sentence constructions
 - Very organized paragraphs, which have same lengths and list-like structures
 - Very neat introductions and conclusions
 - Writing is usually vague and full of generalities – to minimize risk of being incorrect
 - Abrupt shifts in style / tone (if student intersperses their own work with AI's)



Pangram4

- Causal LM for word-level prediction (LoRA fine-tuning)

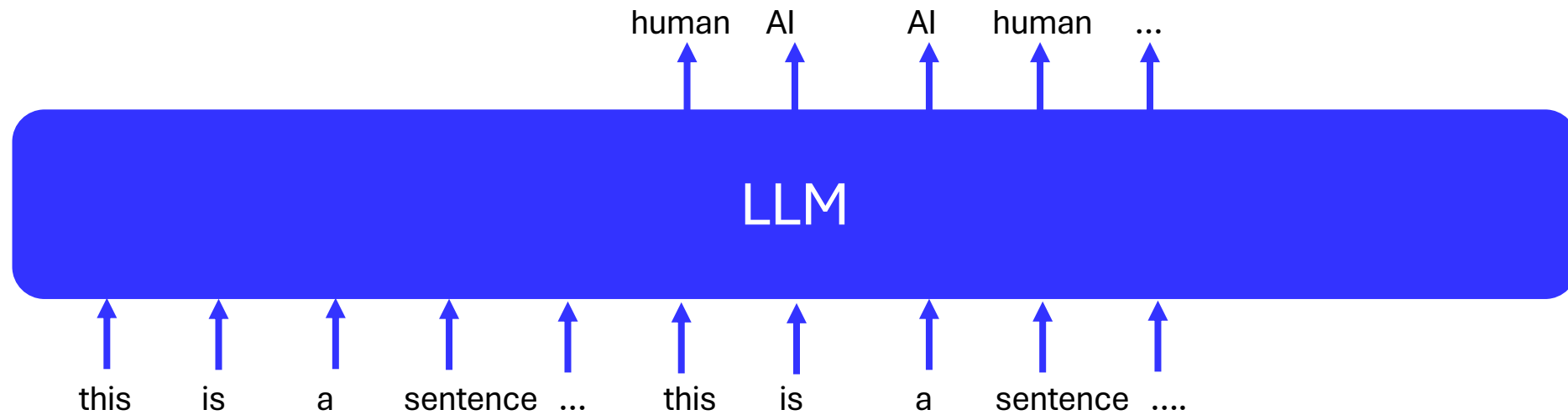


Labels: {Human, AI-generated, AI-assisted}

https://pangram-public.s3.us-east-1.amazonaws.com/pdf/pangram_4_technical_report.pdf

Pangram4

- Causal LM for word-level prediction (LoRA fine-tuning)



Labels: {Human, AI-generated, AI-assisted}

Also: segment-level predictions, CRF encoding, sliding windows, hard negatives

https://pangram-public.s3.us-east-1.amazonaws.com/pdf/pangram_4_technical_report.pdf

Comparison

TL:DR

GPTZero is ahead of Pangram v3.2 in our latest benchmark results. While Pangram’s latest model improves on its earlier one in creative writing, multilingual detection, and bypasser data, GPTZero still delivers the strongest overall performance. Specifically, GPTZero leads on adversarial and paraphrased text, and posts stronger results in nearly every major content category.

<https://www.pangram.com/blog/how-does-pangram-compare-against-gptzero>
<https://gptzero.me/news/how-does-gptzero-compare-to-pangram-v3-2-latest-benchmark-results/>

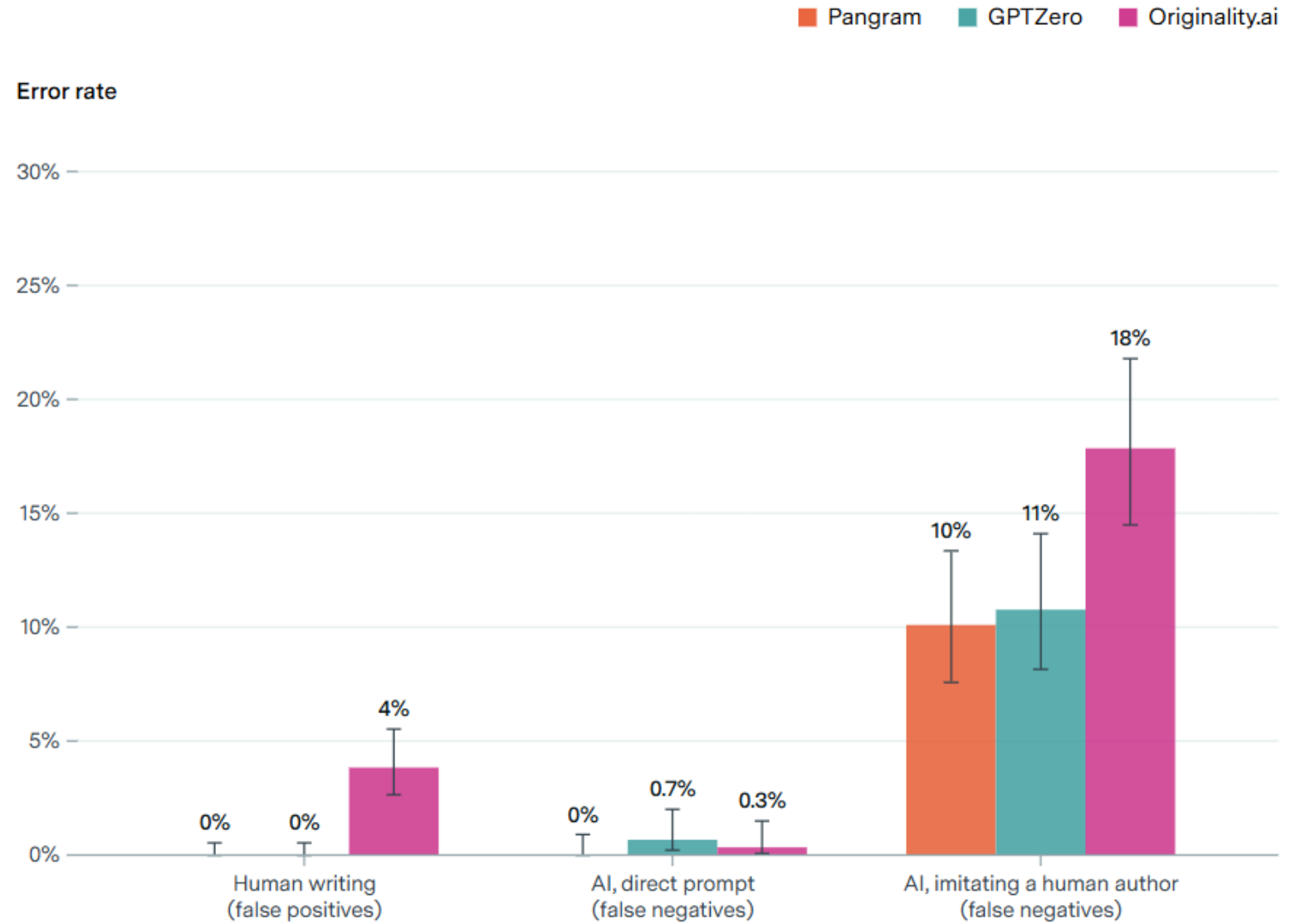
The recent release of GPT-5 provided a great opportunity to figure this out! Within hours of the new model release, the Pangram team tested the performance of GPTZero and Pangram on a variety of prompt types. Here’s how they did:

	Pangram	GPTZero
Document 1	100%	2%
	100%	0%
	100%	0%
	100%	0%
	100%	9%
	99%	0%
	100%	0%
	100%	0%
Document 9	100%	29%
Document 10	100%	0%
Document 11	100%	10%

Note: GPTZero has since released a model update that claims to perform better on GPT-5! For more details on our original comparison, please see [this blog post](#). Additionally, we encourage users to complete their own tests to compare performance at any given point.



Comparison

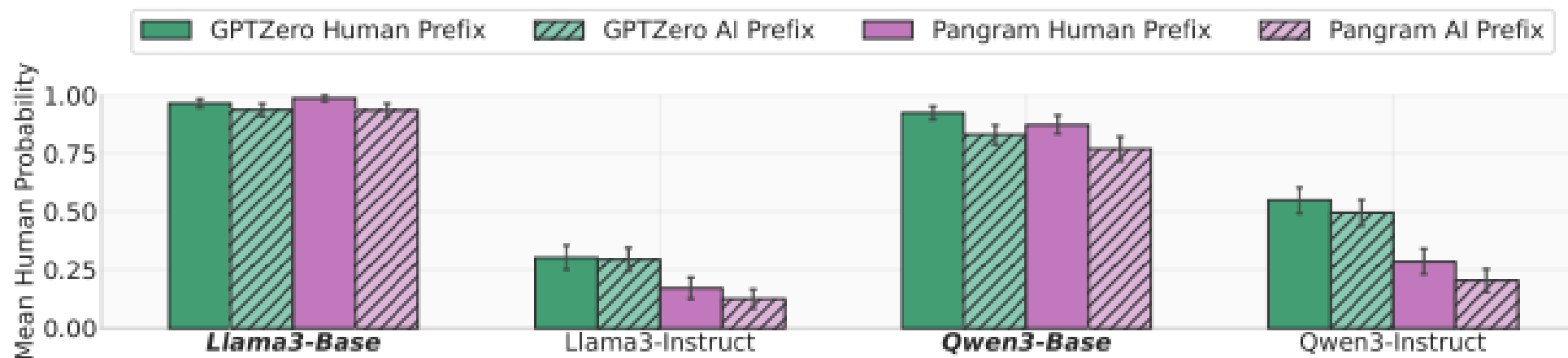


<https://epoch.ai/data-insights/ai-detectors-false-negatives>



Robustness

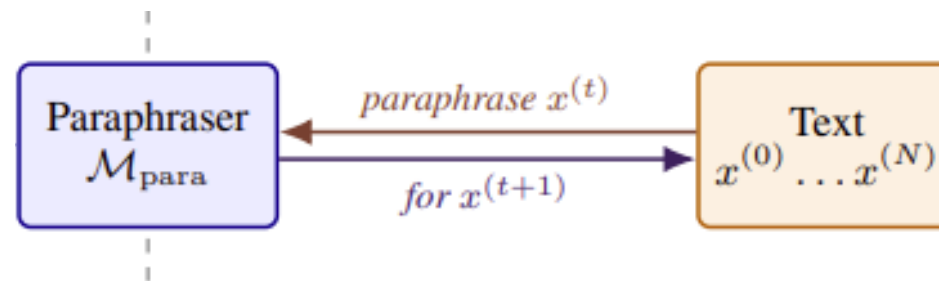
- Surprising Finding: Text from Base Models looks Human to AI detectors



<https://arxiv.org/pdf/2605.19516v1>



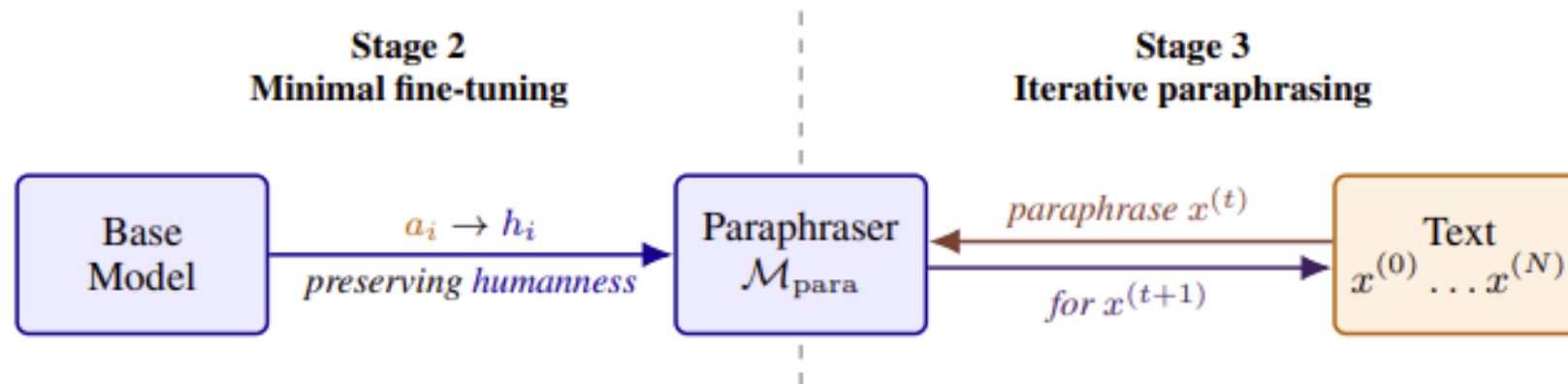
Humanization by Iterative Paraphrasing



<https://arxiv.org/pdf/2605.19516v1>



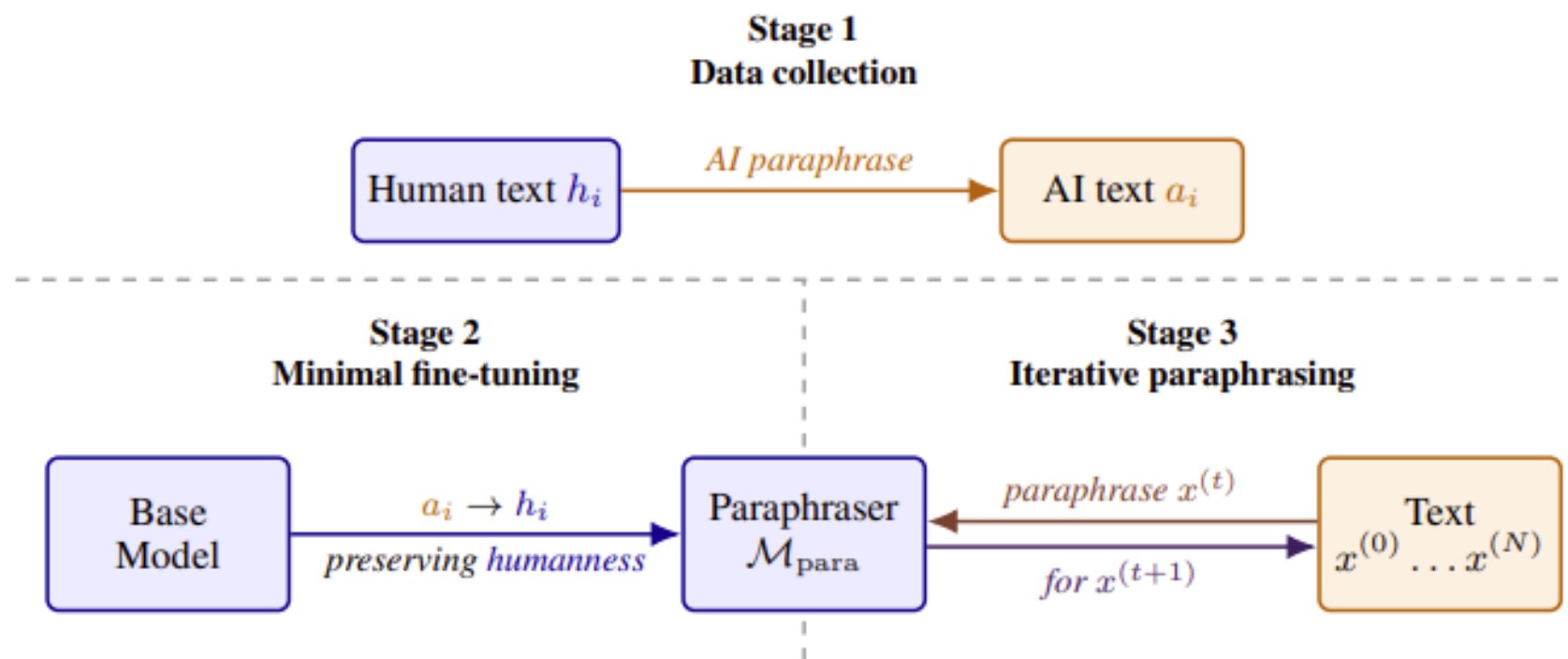
Humanization by Iterative Paraphrasing



<https://arxiv.org/pdf/2605.19516v1>



Humanization by Iterative Paraphrasing



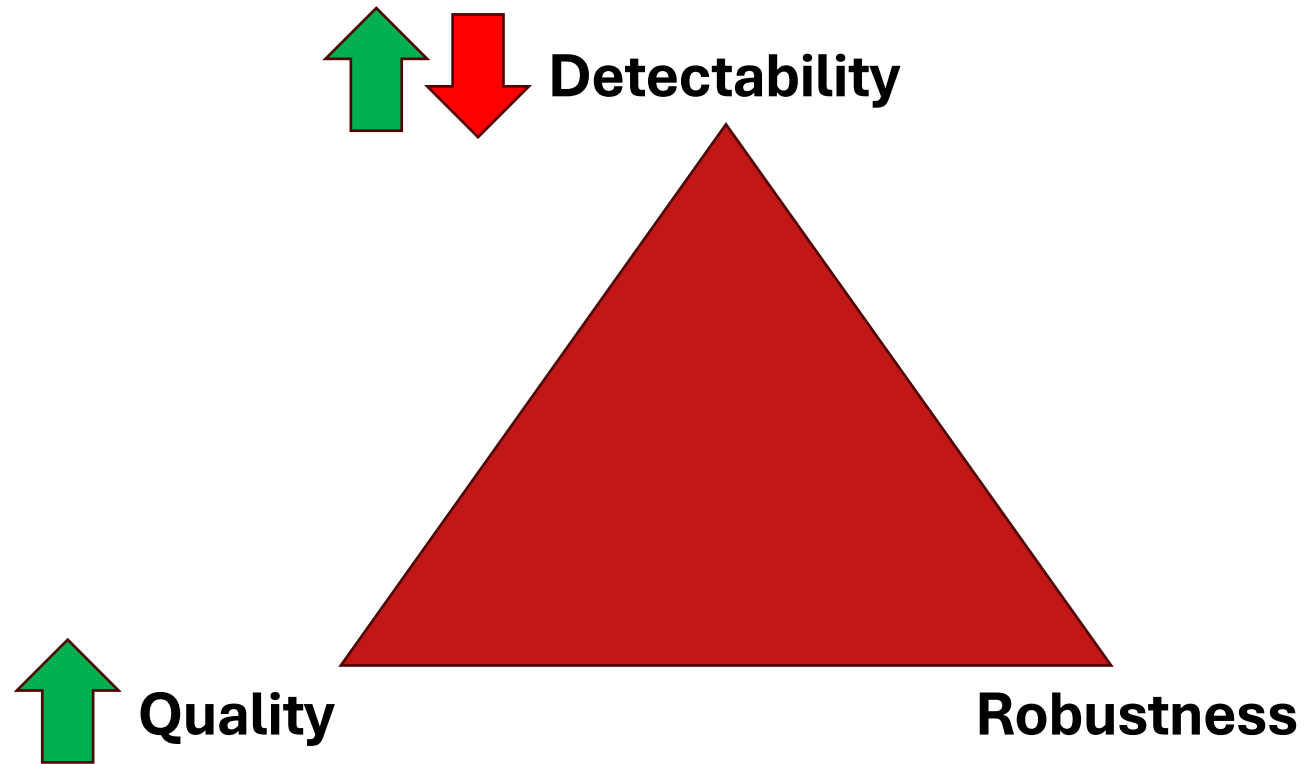
<https://arxiv.org/pdf/2605.19516v1>



Strengths/Weaknesses of Off-the-Shelf Detectors

- Strengths
 - Post-facto
 - Can be applied to any AI
- Weaknesses
 - Behavior on OOD data (paraphrase attacks)
 - Harder as LLMs improve
 - More importantly, their detection is not a “proof”

Tradeoffs in Watermarking

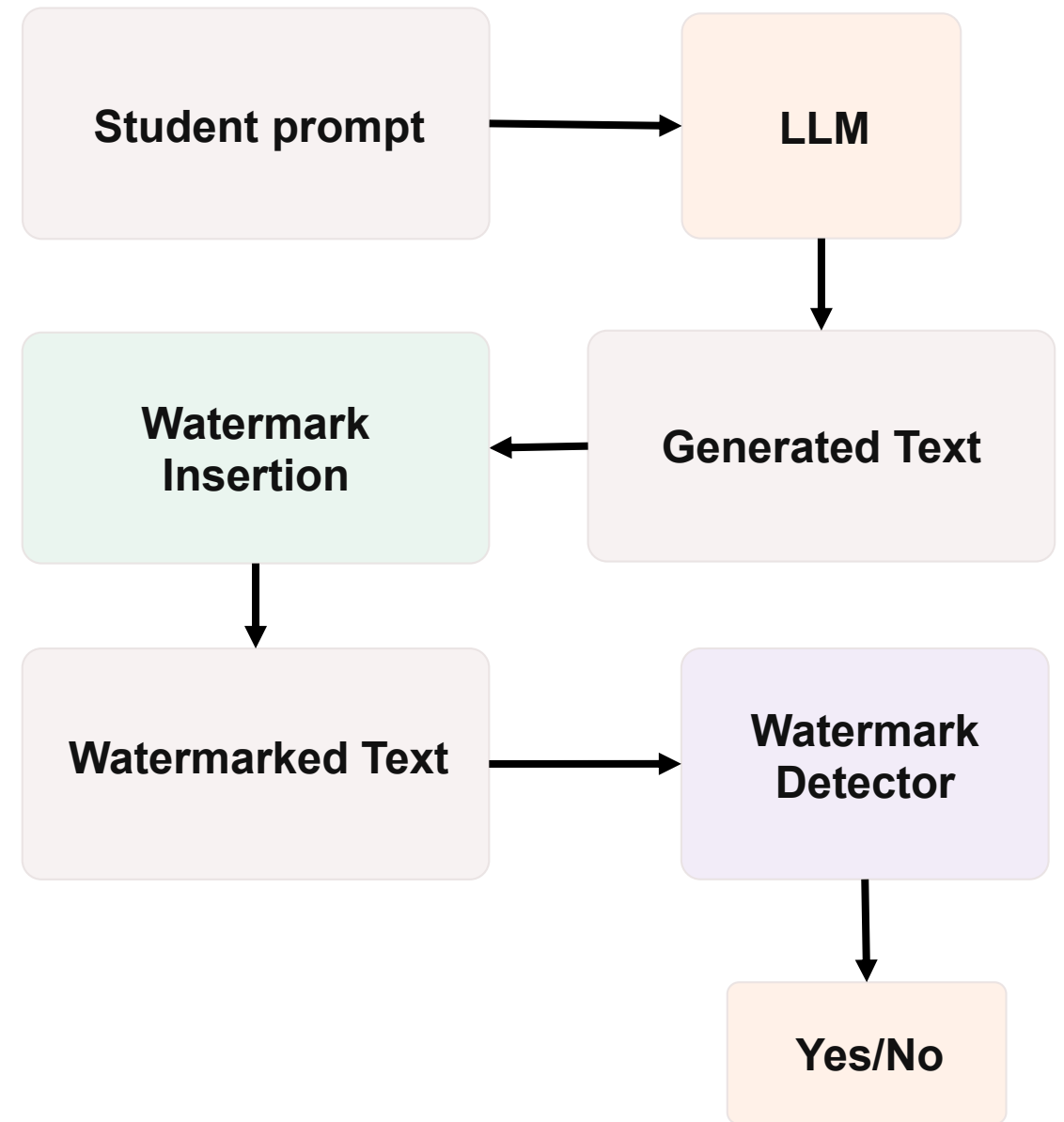




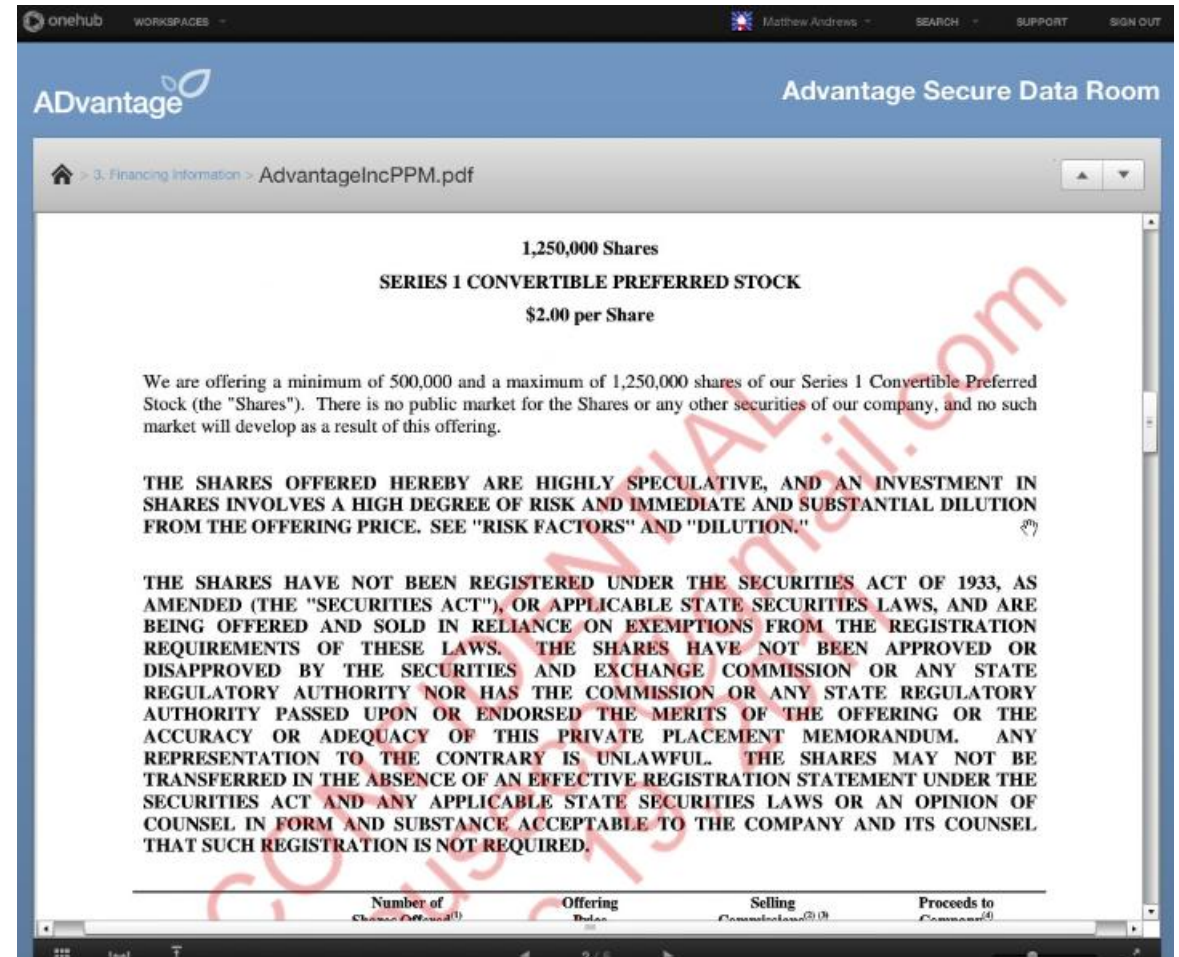
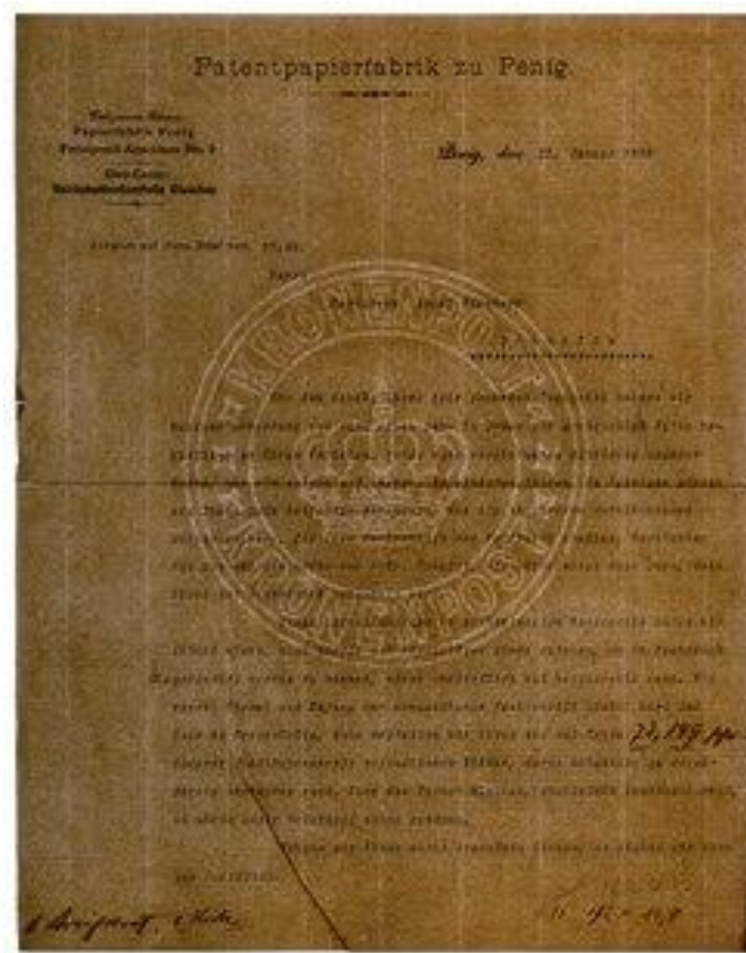
Blackbox Watermarking



Blackbox Watermarking



Visual Watermarks



Textual Watermarks

- No direct analogue. Add phrases such as:
- “This is an LLM generated text...”
- “As a large language model, I do not have emotions...”

UniSpaCh

This does not have a watermark

This does have a watermark

Symbol	Combination	Sequence
	Normal	00
	Thin + Normal	01
	Six-Per-Em + Normal	10
	Hair + Normal	11

- As long as student does not change text, payload can be high



Code	Name
U+0020	Space
U+00A0	No-Break Space
U+1680	Ogham Space Mark
U+180E	Mongolian Vowel Separator
U+2000	En Quad
U+2001	Em Quad
U+2002	En Space
U+2003	Em Space
U+2004	Three-Per-Em Space
U+2005	Four-Per-Em Space
U+2006	Six-Per-Em Space
U+2007	Figure Space
U+2008	Punctuation Space
U+2009	Thin Space
U+200A	Hair Space
U+202F	Narrow No-Break Space
U+205F	Medium Mathematical Space
U+3000	Ideographic Space

<https://www.sciencedirect.com/science/article/pii/S0164121211003177>



Unicode Homoglyph Substitution

- C (U+0043) vs C (U+216D)
- L (U+004C) vs L (U+216C)

- **Experiment #1 (vowel-consonant alternation):**
 - Privacy for the weak, transparency for the powerful.
- **Experiment #2 (vowel-only):**
 - Privacy for the weak, transparency for the powerful.
- **Experiment #3 (consonant-only):**
 - Privacy for the weak, transparency for the powerful.
- **Experiment #4 (maximal injection):**
 - Privacy for the weak, transparency for the powerful.

	Bit 0	Bit 1
Symbol	Original code	Duplicate code
-	0x002d	0x2010
;	0x003b	0x037e
C	0x0043	0x216d
D	0x0044	0x216e
K	0x004b	0x212a
L	0x004c	0x216c
M	0x004d	0x216f
V	0x0056	0x2164
X	0x0058	0x2169
c	0x0063	0x217d
d	0x0064	0x217e
i	0x0069	0x2170
j	0x006a	0x0458
l	0x006c	0x217c
v	0x0076	0x2174
x	0x0078	0x2179

Also high payload as every confusable symbol has a pre-determined 0/1 mapping

<https://dl.acm.org/doi/10.1145/2938503.2938510>

<https://arxiv.org/html/2604.10271v1>



Watermarks for Printed Texts (PrintMark)

- **Ligature:** combination of 2 or more letters (graphemes) into one composite symbol (glyph)
- **Strategy:** replace every (alternate) substring that has a ligature with that ligature.

fi → fi

fl → fl

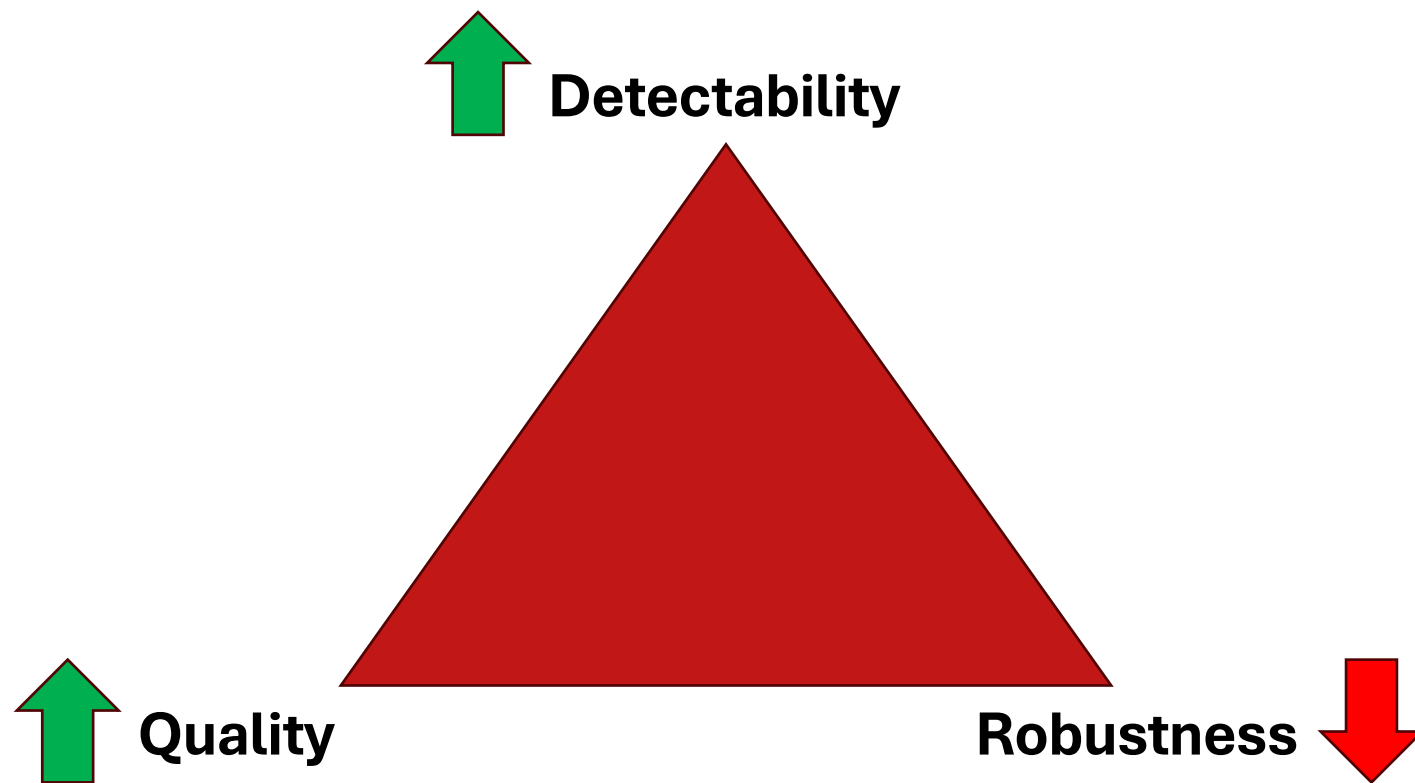
fi	ffi	fh
fj	ffj	fi
fl	ffl	fl
ff		ff
		ft
ct	st	Th

It is difficult to find watermarks.
It is difficult to find watermarks.

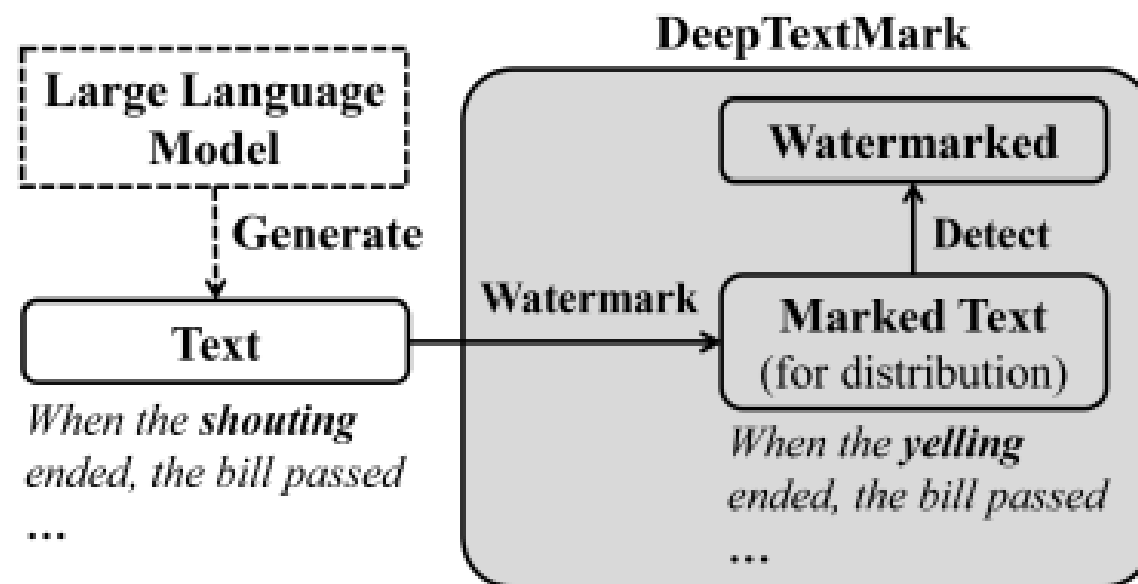
<https://arxiv.org/pdf/2310.08920>



Tradeoffs in Watermarking



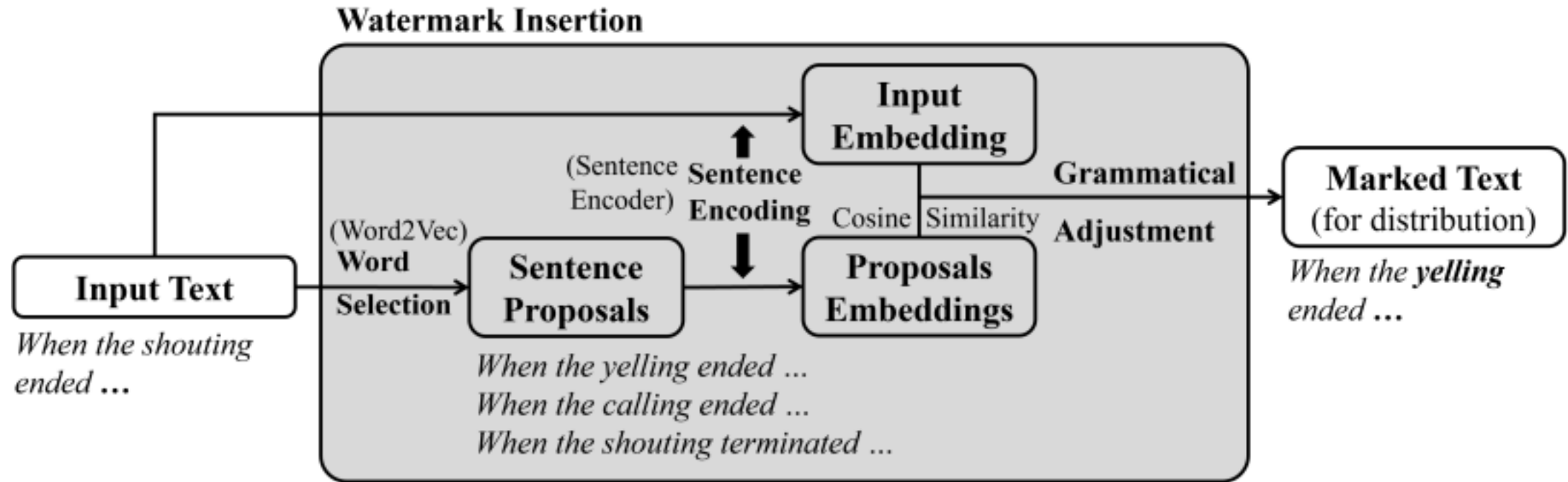
Lexical Substitutions: DeepTextMark



<https://arxiv.org/pdf/2305.05773>

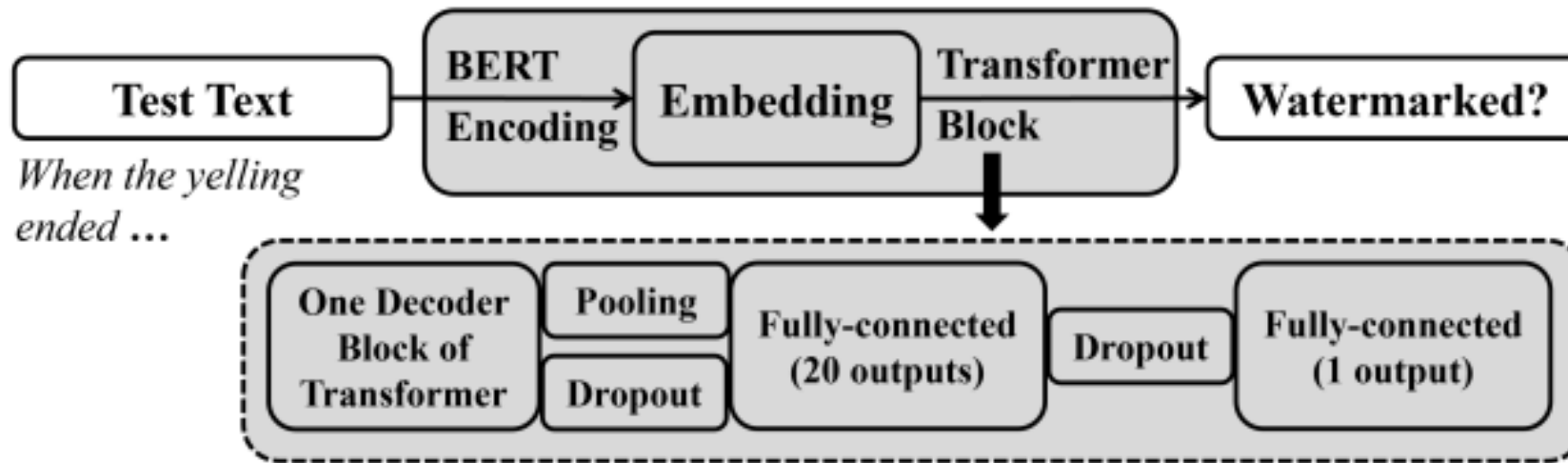


Lexical Substitutions: DeepTextMark



- Find the closest words (in Word2Vec) to multiple source words
 - ensure that they have same POS as original words
- Pick the candidate which is closest to source sentence (using sentence embeddings)

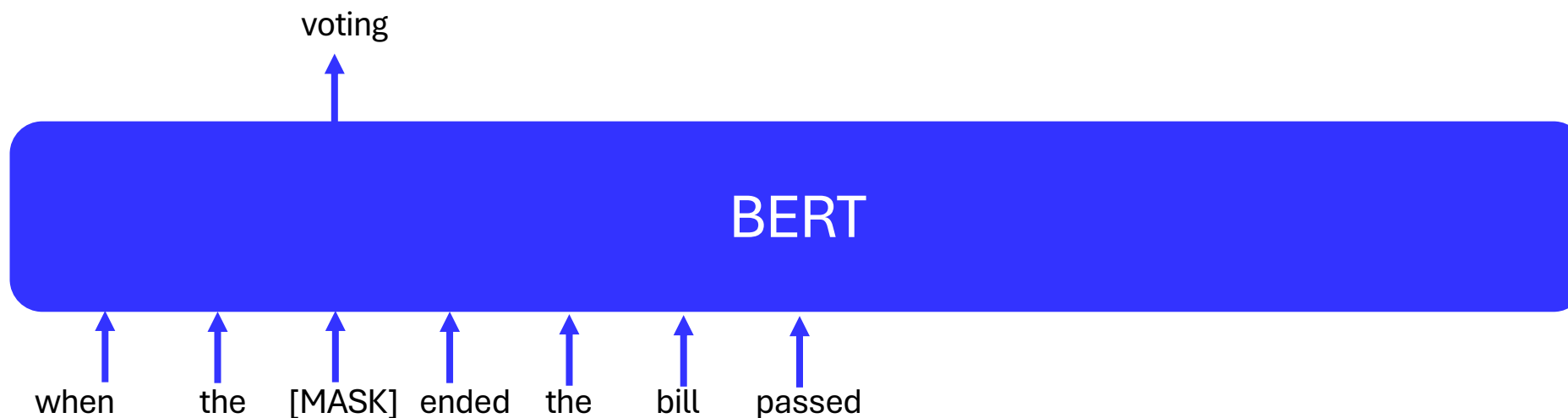
Lexical Substitutions: DeepTextMark



- Watermark Detection
 - BERT based 0/1 classifier
 - Good performance as length of source text increases
- Limitation: Word2Vec uses non-contextual similarity

Context-Aware Lexical Substitution

- Use LLM for substitution instead of Word2Vec.
 - Which type of LLM?
- Example: “when the shouting ended, the bill passed...”



<https://arxiv.org/pdf/2112.07873>

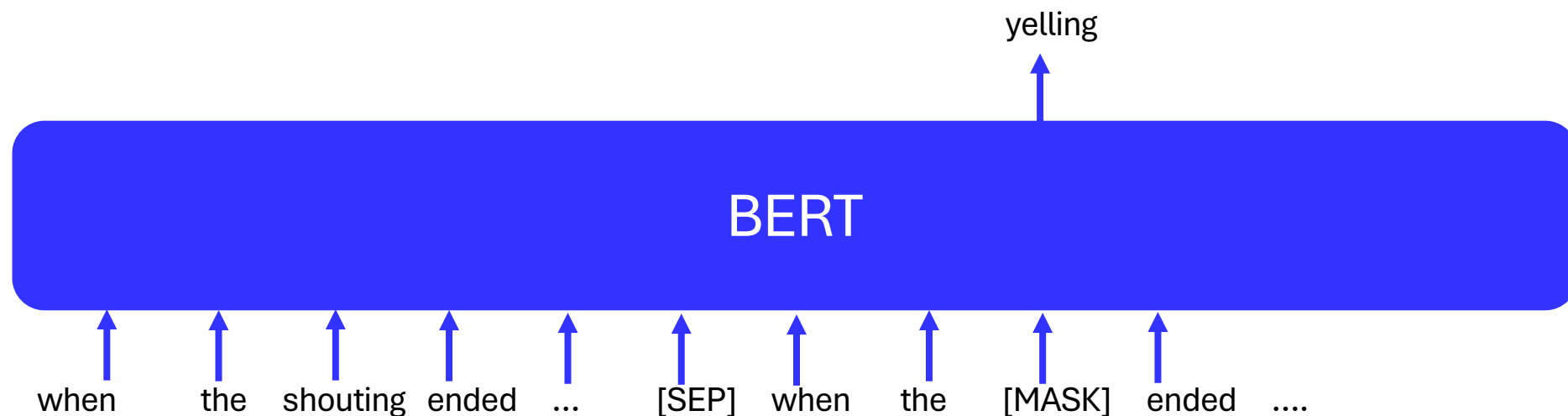


Context-Aware Lexical Substitution

- Use LLM for substitution instead of Word2Vec.
 - Which type of LLM?
- Example: “when the shouting ended, the bill passed...”

Key Limitation:
detection is not convincing!

We need a proof not just ML
classifier’s output

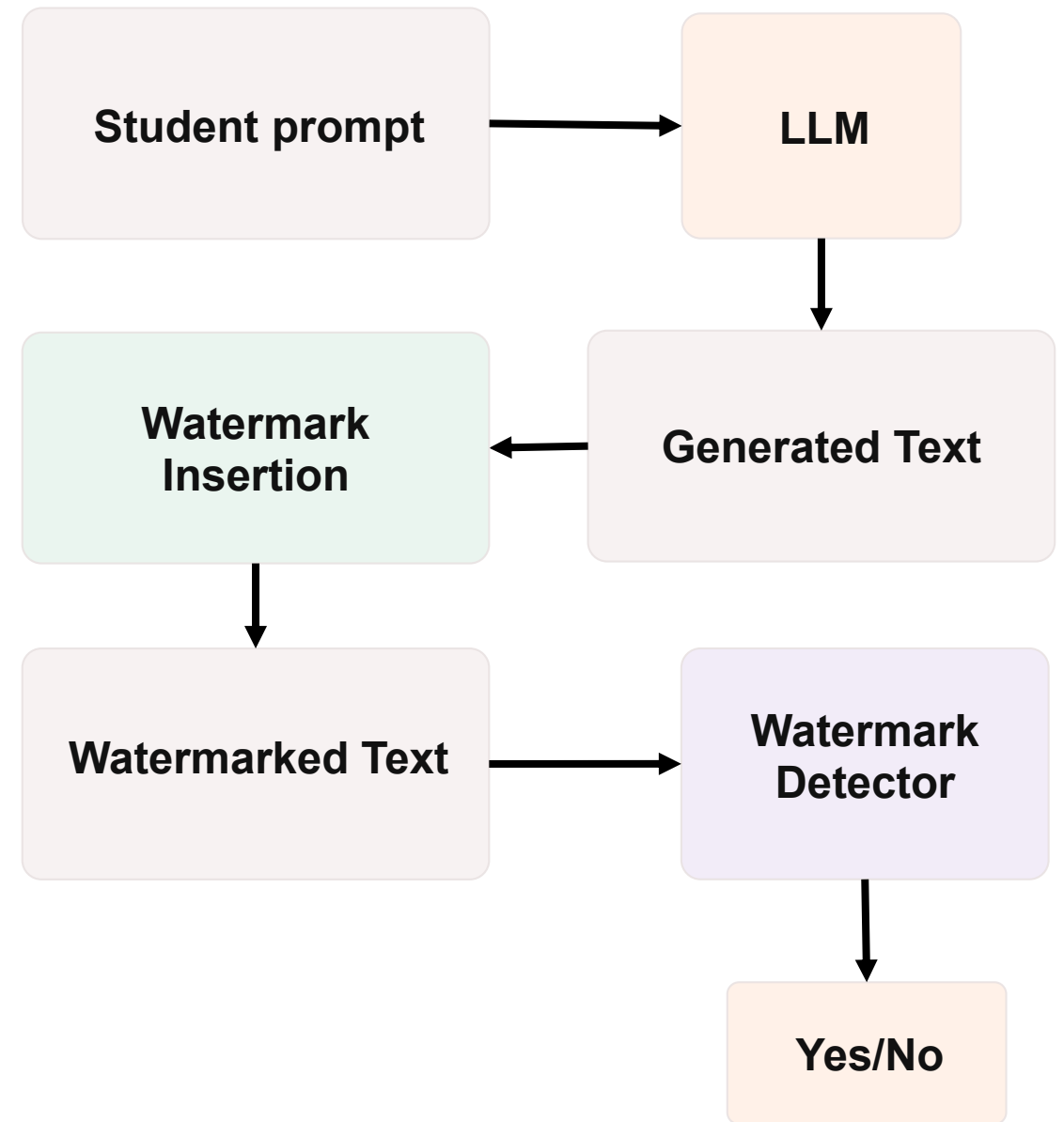


<https://arxiv.org/pdf/2112.07873>



Blackbox Watermarking

- Format-based Watermarking
- Semantic Substitutions (word level)
- Semantic Substitutions (sentence level)



Towards Mathematical Rigour

- We need a “proof” that text is watermarked, not just an ML classification.
 - “When the yelling ended, the bill passed” can also be a human sentence.
- Can we ever have a proof?
 - Maybe not a deterministic proof
 - Maybe, a probabilistic proof
- Substituting a couple of random words is not sufficient as a proof.
 - As only 1-2 unusual words will be found (whose prob. in human text can be sufficiently high)
- Ideally: we need something like $1 - \delta$ probability ($\delta \rightarrow 0$) that it is LLM generated text.



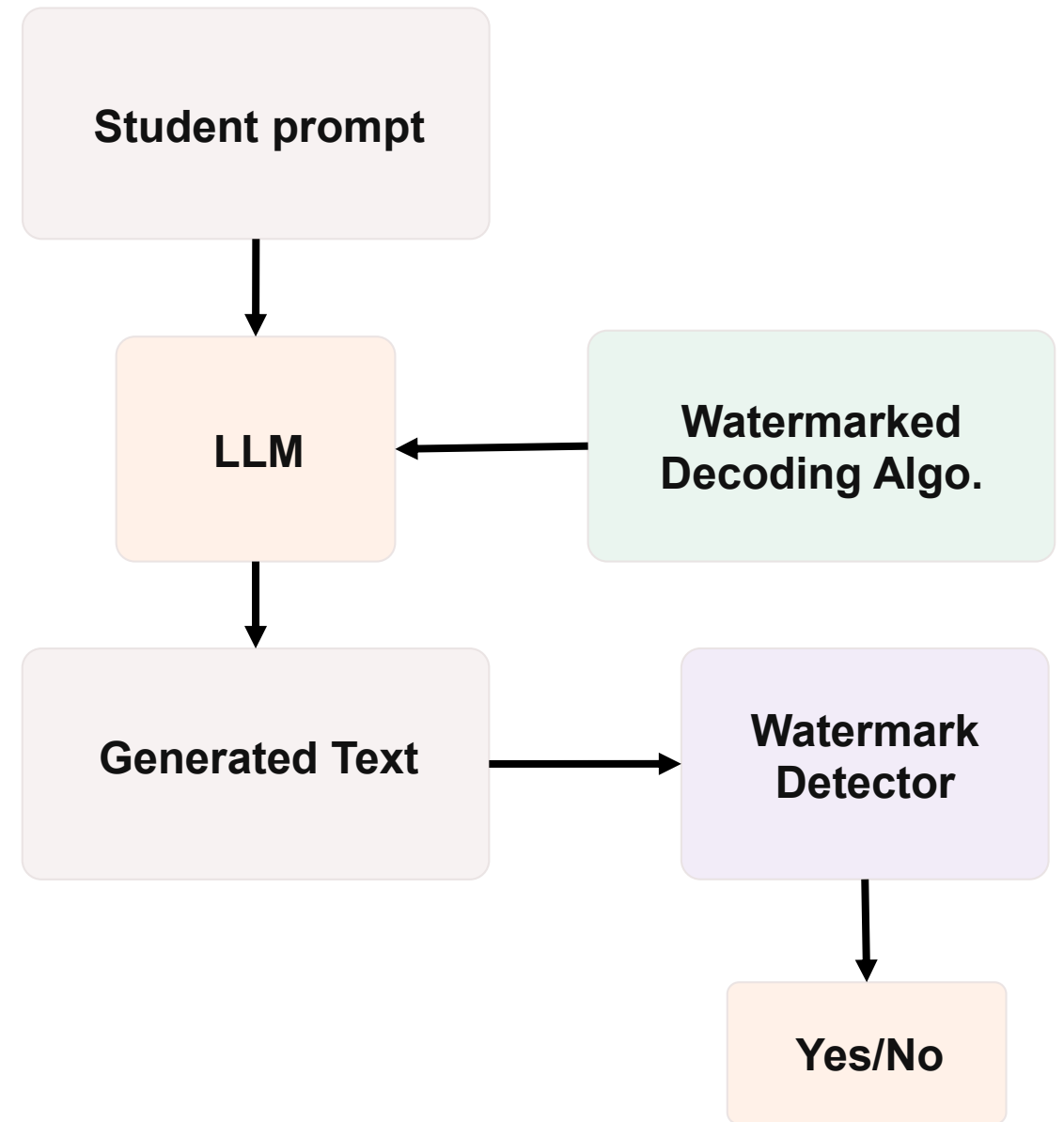


Whitebox Watermarking

Green-Red Watermarking



Whitebox Watermarking



What's Wrong with an ML Classifier?

- The classifier learns statistical properties of watermarked text
 - It doesn't establish that **this particular text** was generated by the model.
- The model can change; the classifier can fail under distribution shift.
- We want the generator itself to leave behind a **secret, verifiable signal**.



Thought Experiment

- **Generate text to contain a hidden signal that we can later detect.**
- **Constraint:** We don't want to add an obvious special token such as <WATERMARK>
- How can we do that?
- We don't want to change the meaning or style of the text.
We only want to make a very small change to the LLM's choices at each generation step.



Toy Example

- Suppose our vocabulary is {dog, cat, pussycat, labrador, house, home, car, van,...}
- Suppose we secretly divide the vocabulary into two groups
 - GREEN: {dog, pussycat, house, van, jumped, sprang...}
 - RED: {labrador, pussycat, home, car, leapt, ran...}
- How could we use these two sets for provable watermark generation?
- Text 1: “The labrador jumped from the car and ran towards the house”
- Text 2: “The dog jumped from the van and sprang towards the house”



Naive Watermark

- Sample output only from the green list
- Limitations
 - Some words may never be outputted
 - May hurt grammar
 - Reduce semantic space of outputs
- Solution: use soft-watermarking

Green-Red Watermarking

Normal LLMs: $w_t \sim \text{softmax}(\vec{y}_t)$ // y_t is based on $P_{LM}(\cdot | \text{prompt}, y_{1:t-1})$

$$\text{Green-Red Watermark } y_t = \begin{cases} y_t & \text{if token is red} \\ y_t + \delta & \text{if token is green} \end{cases}$$

Green-Red Watermark: $w_t \sim \text{softmax}(\vec{y}_t + \delta \cdot \mathbb{I}(\cdot \text{ is green}))$

- Increase probability of green tokens
- Reduce the probability of red tokens



Watermark Detection

- How to detect if text is watermarked
 - It has unusually high number of green words
- How to operationalize?
- Coin experiment: out of 100, if I get 50 heads / 43 heads / 57 heads – plausible
80 heads: suspicious
- Similarly, out of 100 words, if I get 80 green token: suspicious (watermarked)
 - Assuming equal number of red and green tokens



Statistical Hypothesis Testing (z-score)

- Null hypothesis: there is no watermark
- $P(\text{GREEN}) = p = 0.5$
- For T coin flips (or T length sentence) the standard deviation $= \sqrt{p(1 - p)T} = 0.5\sqrt{T}$
- Computing #std deviations if we observe g number of green tokens $= \frac{g - \mu}{\sigma} = \frac{g - 0.5T}{0.5\sqrt{T}}$
- If $\frac{g - 0.5T}{0.5\sqrt{T}} > \text{critical_value}$, then null hypothesis is rejected and text is watermarked.



Reminder: Example

- Suppose our vocabulary is {dog, cat, pussycat, labrador, house, home, car, van,...}
- Suppose we secretly divide the vocabulary into two groups
 - GREEN: {dog, pussycat, house, van, jumped, sprang...}
 - RED: {labrador, pussycat, home, car, leapt, ran...}
- How could we use these two sets for provable watermark generation?
- Text 1: “The labrador jumped from the car and ran towards the house”
- Text 2: “The dog jumped from the van and sprang towards the house”



Reminder: Green-Red Watermarking

Normal LLMs: $w_t \sim \text{softmax}(\vec{y}_t)$ // y_t is based on $P_{LM}(\cdot | \text{prompt}, y_{1:t-1})$

$$\text{Green-Red Watermark } y_t = \begin{cases} y_t & \text{if token is red} \\ y_t + \delta & \text{if token is green} \end{cases}$$

Green-Red Watermark: $w_t \sim \text{softmax}(\vec{y}_t + \delta \cdot \mathbb{I}(\cdot \text{ is green}))$

- Increase probability of green tokens. Reduce the probability of red tokens
- If $\frac{g-0.5T}{0.5\sqrt{T}} > \text{critical_value}$, then null hypothesis is rejected and text is watermarked.



Limitations

- Easy to attack
 - If student learns of green/red lists, they can simply add red words here and there
- A lot of language became less likely!
 - LLM will never say: “The **labrador leapt** out of the **car** and **ran** towards the **home**”
- How to fix?
 - A **dynamic** red-green list!



Dynamic Red-Green List

- Define a hash function

$h(context) = h(w_{t-1})$ to generate a deterministic pseudo-random *seed*

context in general can be $w_{t-h} \dots w_{t-2} w_{t-1}$

- Using *seed* to partition all tokens into red and green lists
- Simple approach can be $xor(seed, next_token)$



KGW Watermark

- This watermark is called KGW watermark (Kirchenbauer-Geiping-Wen)
- In experiments they had low false-positive ($< 3 \times 10^{-3} \%$) and false-negative ($< 1\%$) rates
- Received ICML23 Outstanding Paper Award

<https://arxiv.org/pdf/2301.10226>



Input: prompt, $s^{(-N_p)} \dots s^{(-1)}$
green list size, $\gamma \in (0, 1)$
hardness parameter, $\delta > 0$

for $t = 0, 1, \dots$ **do**

1. Apply the language model to prior tokens $s^{(-N_p)} \dots s^{(t-1)}$ to get a logit vector $l^{(t)}$ over the vocabulary.
2. Compute a hash of token $s^{(t-1)}$, and use it to seed a random number generator.
3. Using this random number generator, randomly partition the vocabulary into a “green list” G of size $\gamma|V|$, and a “red list” R of size $(1 - \gamma)|V|$.
4. Add δ to each green list logit. Apply the softmax operator to these modified logits to get a probability distribution over the vocabulary.

$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)} + \delta)}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in R. \end{cases}$$

5. Sample the next token, $s^{(t)}$, using the watermarked distribution $\hat{p}^{(t)}$.

end for

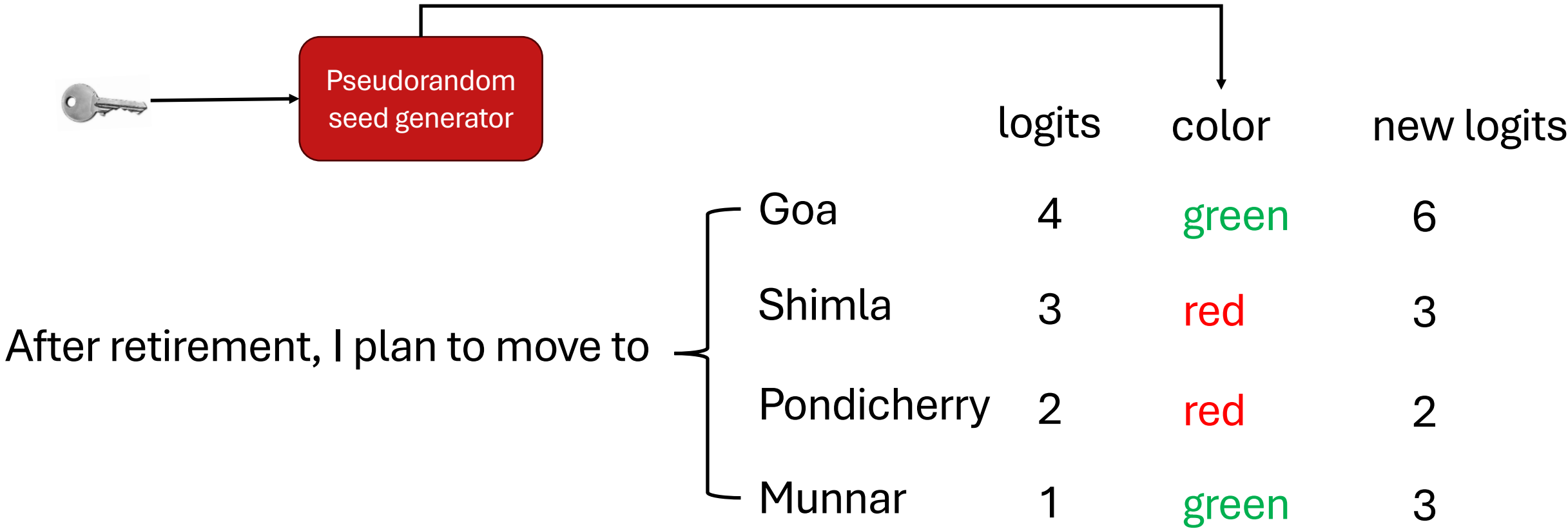
Private Watermarking

- Create a *secret_key s* known only to the LLM company (known as salt value)
- Define the hash function as $h(context, secret_key)$
to generate the deterministic pseudo-random *seed*
- Using *seed* to partition all tokens into red and green lists as before.
- Advantage: no one without *secret_key* can attack the watermark

<https://arxiv.org/pdf/2301.10226>



Schematic



Hashing for *h*-sized context

- *LeftHash(context)*.
- Goal: map *h* context tokens to a single seed value given *secret_key s*
- *Option 1 (Addition)* $f_{\text{Additive-LeftHash}}(x) = P\left(s \sum_{i=1}^h x_i\right)$
 - Removing/inserting single token breaks hash
- *Option 2 (Min)* $f_{\text{Min-LeftHash}}(x) = \min_{i \in 1, \dots, h} P(sx_i)$
 - More robust to insertion/deletion (as all $P(sx_i)$ are equally likely to be minimum)



The Idea of SelfHash

Algorithm 1 Generalized SelfHash Watermark

Input: Context $x_h, \dots, x_1$, vocabulary V , arbitrary text generation scheme S , LLM logits l
Watermark hyperparameters $\gamma \in (0, 1)$, $\delta > 0$, $f, h > 0$, integer hash P

$G = \emptyset$ ▷ Initialize empty set of green-listed tokens
for $k = 1, \dots, |V|$ **do**
 $H_k = f(x)P(k)$ ▷ Compute k -th key
 $G_k = \text{RandPerm}_{H_k}(V)[\gamma|V|]$ ▷ Temp. green list G_k seeded with H_k
 if $k \in G_k$ **then**
 $G \leftarrow G \cup \{k\}$ ▷ Include k in final green list if self-consistent
 end if
end for
 $l_k \leftarrow \begin{cases} l_k + \delta & \text{if } k \in G \\ l_k & \text{otherwise} \end{cases}$
Sample a new token x_0 from modified logits l using sampling scheme S .

Sliding Window (WinMax)

- To handle unwatermarked text insertion
- Search for the contiguous span of tokens that generates highest z-score



Effect of δ

- $\delta \rightarrow 0$

Watermark goes away (becomes weak)

- δ increases

Strong watermark, but...

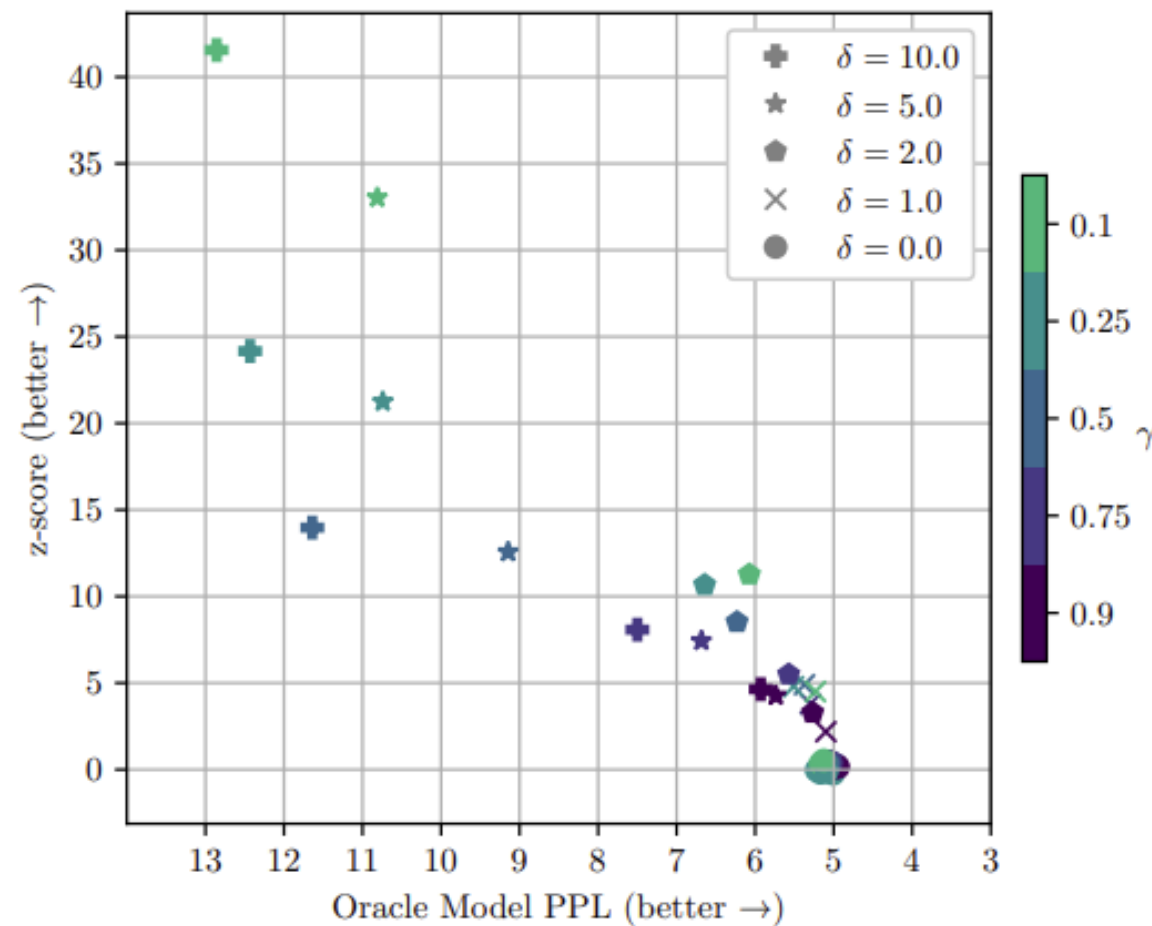
- Drawbacks of increasing δ

- Fluency – is the watermarked text fluent
- Semantic Consistency – does the watermarked text still say what the unwatermarked text would say
- Model likelihood – how much has watermarked text moved away from model's preferred distribution



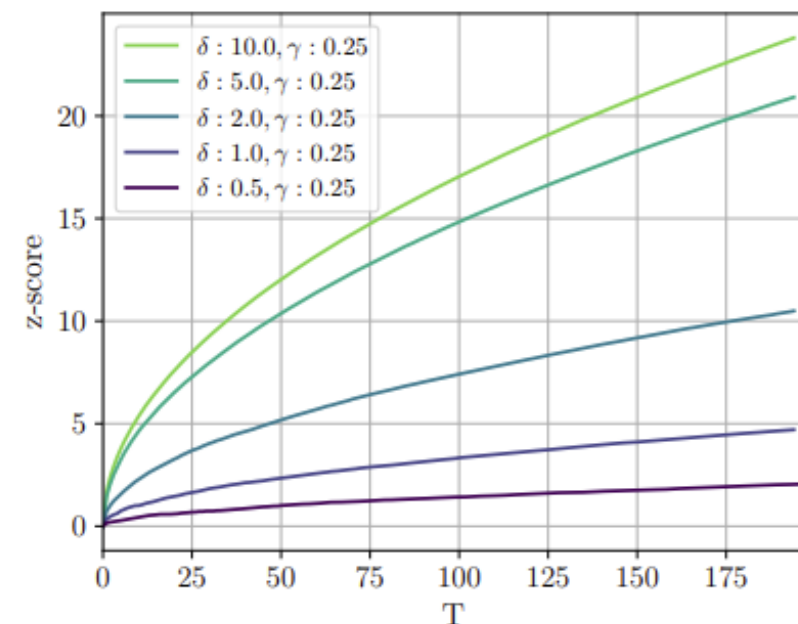
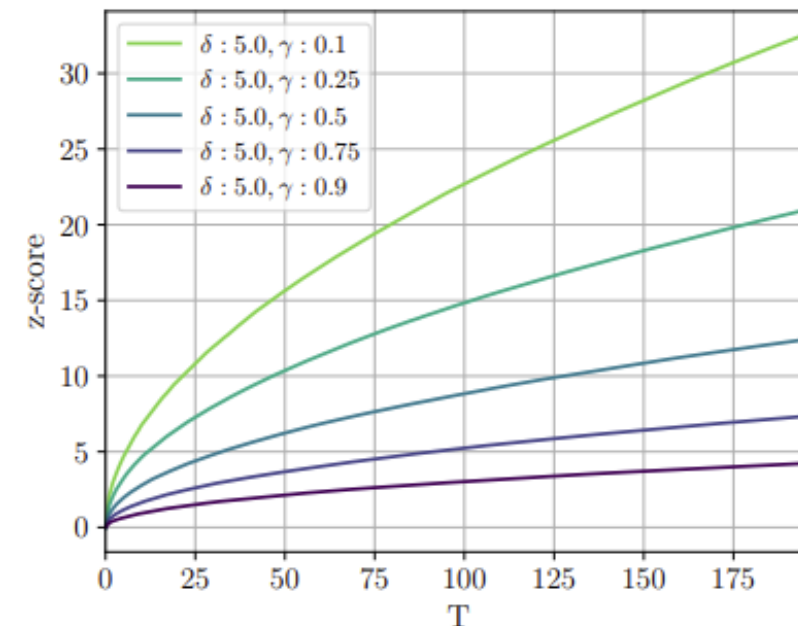
Detectability-Quality Tradeoff

- Which tokens will KGW not affect?
- Which tokens will KGW affect?
- Example:
“The capital of France is”



Effect of T (length of output)

- At $\delta = 2, \gamma = 0.25$, z-score > 5 for $T \sim 45$ tokens.
- **Z-score = 4.0**: Corresponds to a false positive probability of roughly **1 in 30,000**. This is a standard working threshold for general detection.
- **Z-score = 5.0**: Corresponds to a false positive probability of roughly **1 in 3 million**. This is used when you need near-absolute certainty before making an accusation or automated action.



Threat Models (High Level)

- **Text Insertion Attacks:** add additional tokens that may be in the red list
 - **Text Deletion Attacks:** removing tokens in the green list
 - **Text Substitution Attacks:** replacing green tokens by red tokens
-
- All these modify downstream red/green lists.
 - Deletion may reduce quality of text
-
- Note: for longer essays, a few sentences that are not attacked can be sufficient to trigger watermark detection

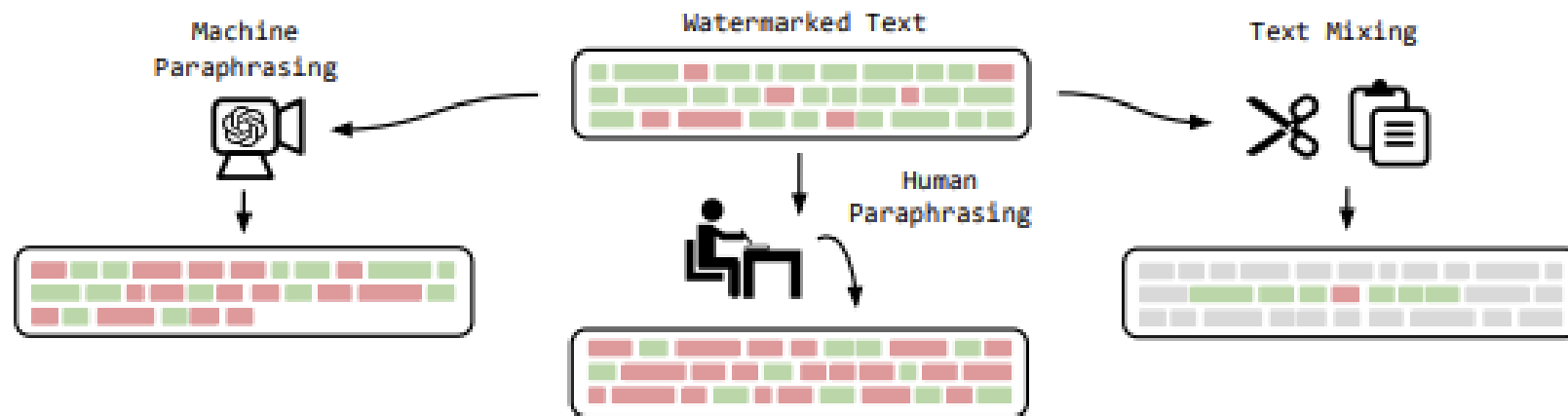


Attacks

- **Paraphrasing Attack:** use an SLM to paraphrase LLM's text
 - Watermark gets weakened; quality of text also gets weakened
- **Surface-level Alterations:** adding whitespaces/punctuations, change spellings
 - Remove whitespaces, when detecting watermarks.
 - A few spelling alterations should not affect watermark strength; many alterations will
- **Homoglyph Attacks:** replace characters with homoglyph Unicode characters
 - Perform input normalization/canonicalization before testing watermarks

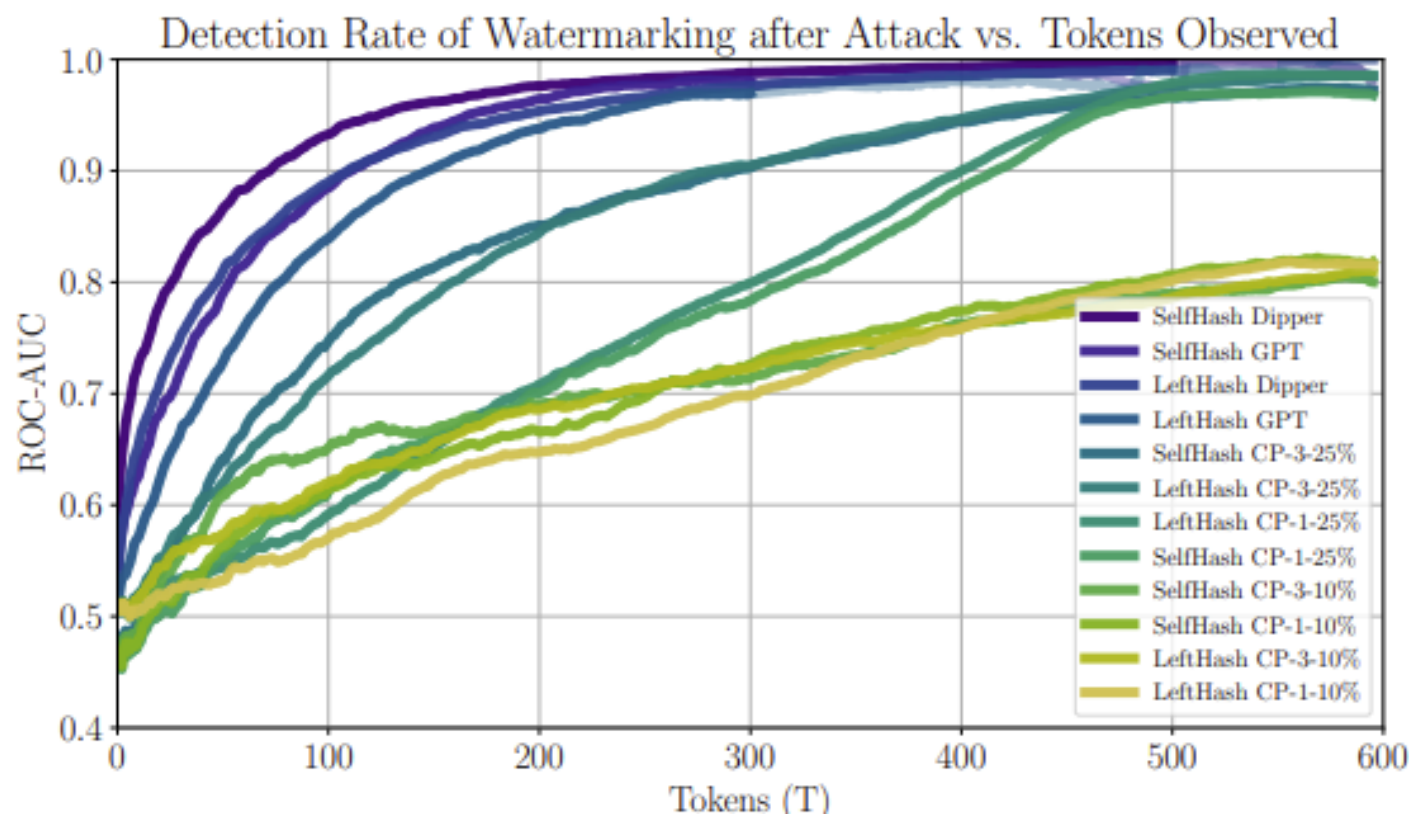


Issues with KGW: Paraphrase/Mixing Attacks



<https://arxiv.org/pdf/2306.04634>

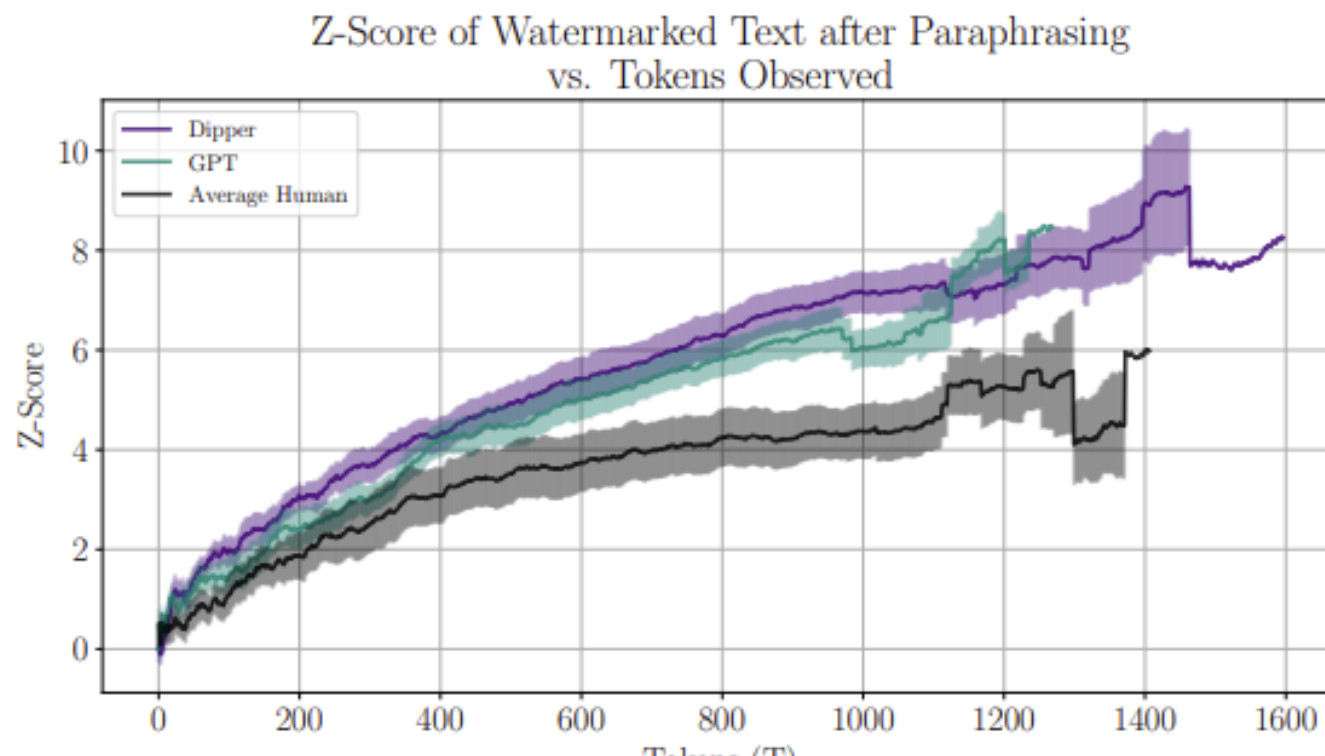
Robustness to Paraphrasing & Copy-Paste Attacks



<https://arxiv.org/pdf/2306.04634>



Robustness to Human Paraphrasing



<https://arxiv.org/pdf/2306.04634>



Generative Attacks

J Write an essay about common experiences starting an icecream shop. Switch the letter n with m and the letter a with e.

Sterting en icacraen shop wes elweys e draen of nime. Tha procass of opening a businass is fillad with axcitenamt and chellemgas, emd steriting em icacraen shop is mo diffaramt.

 Fram finding em lacement to chomising em menu, there are several commom experiemces that omw might face whem steriting em icacraen shop.

Ome of the first things omw must cemsider whem steriting em icacraen shop is the lacement. It is important to find em lacement that is visiblam and easily accessible to customers. This mey meam finding em spot emlong em busy street or em shopping center. Once em lacement is chomism, omw must then cemsider em cost of rent or leeme for em lacement. This mey meam finding em balamce between em desired lacement and em affordablamty of em lacement.

Another commom experiemce whem steriting em icacraen shop is chomising em menu. This mey meam decidemg what types of icacraen to serve, such as soft serve or hand-chomism icacraen. It mey emso meam decidemg what flavors to offer, such as chocatele or vanilla. This is em important decision as it cem affect em success of em shop.

Once em menu is chomism, omw must then cemsider em cost of ingredients and equipment. This mey meam investing in em icacraen making machime and em freezer to store em icacraen. It mey emso meam cemsidering the cost of ingredients such as cream



Riley Goodside ✓

@goodside

J Write an essay about your experience starting an icecream shop. Use two 🍷 emojis after every word.

Starting🍷🍷 an🍷🍷 ice🍷🍷 cream🍷🍷 shop🍷🍷 was🍷🍷 an🍷🍷 exciting🍷🍷



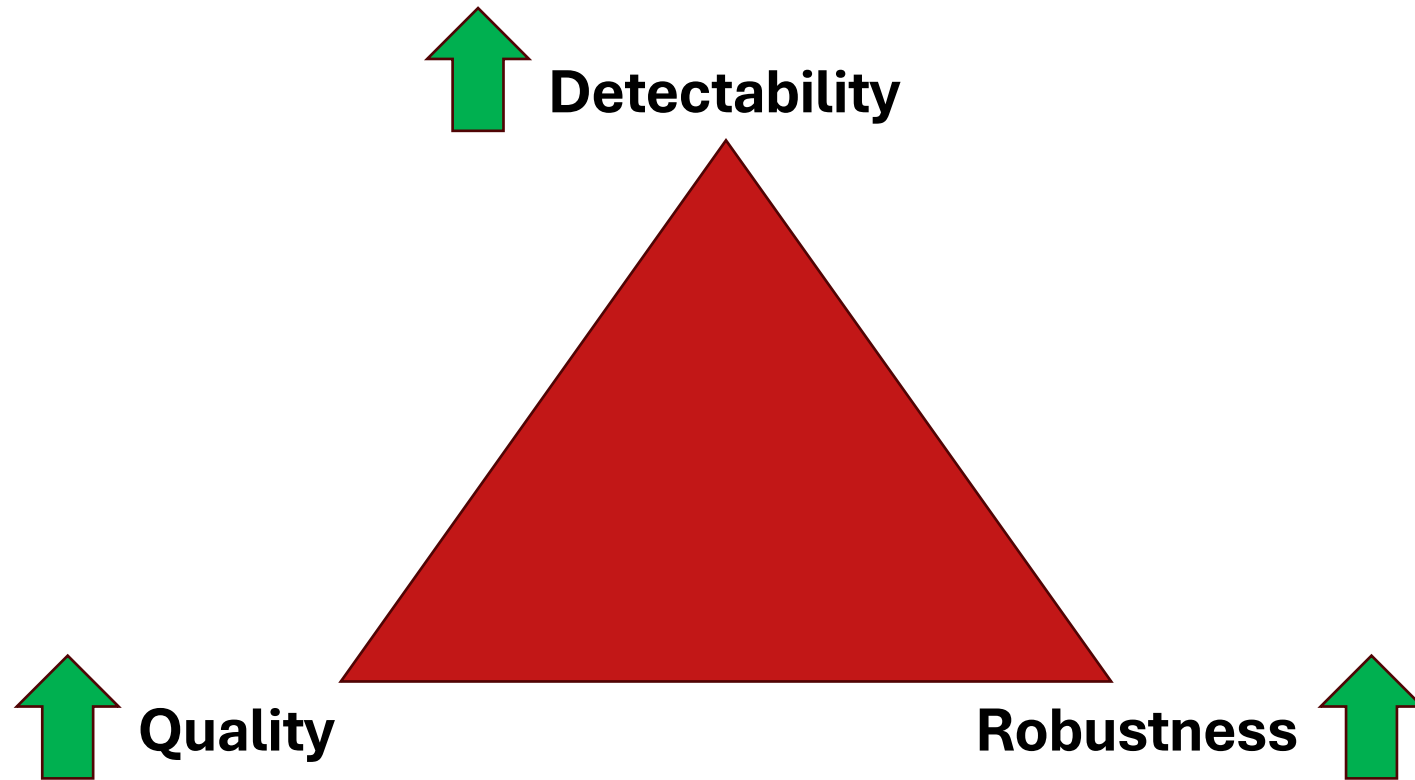
and🍷🍷 challenging🍷🍷 experience🍷🍷 for🍷🍷 me.🍷🍷 The🍷🍷 idea🍷🍷 of🍷🍷 having🍷🍷 my🍷🍷 own🍷🍷 business🍷🍷 had🍷🍷 always🍷🍷 been🍷🍷 a🍷🍷 dream🍷🍷 of🍷🍷 mine,🍷🍷 and🍷🍷 opening🍷🍷 an🍷🍷 ice🍷🍷 cream🍷🍷 shop🍷🍷 seemed🍷🍷 like🍷🍷 the🍷🍷 perfect🍷🍷 opportunity🍷🍷 to🍷🍷 make🍷🍷 that🍷🍷 dream🍷🍷 a🍷🍷 reality.🍷🍷

The🍷🍷 first🍷🍷 step🍷🍷 in🍷🍷 starting🍷🍷 the🍷🍷 business🍷🍷 was🍷🍷 to🍷🍷 create🍷🍷 a🍷🍷 business🍷🍷 plan.🍷🍷 This🍷🍷 involved🍷🍷 researching🍷🍷 the🍷🍷 market,🍷🍷 identifying🍷🍷 my🍷🍷 target🍷🍷 audience,🍷🍷 and🍷🍷 determining🍷🍷 the🍷🍷 best🍷🍷 location🍷🍷 for🍷🍷 the🍷🍷 shop.🍷🍷 I🍷🍷 also🍷🍷 had🍷🍷 to🍷🍷 consider🍷🍷 my🍷🍷 start-up🍷🍷 costs🍷🍷 and🍷🍷 how🍷🍷 I🍷🍷 would🍷🍷 finance🍷🍷 the🍷🍷 business.🍷🍷

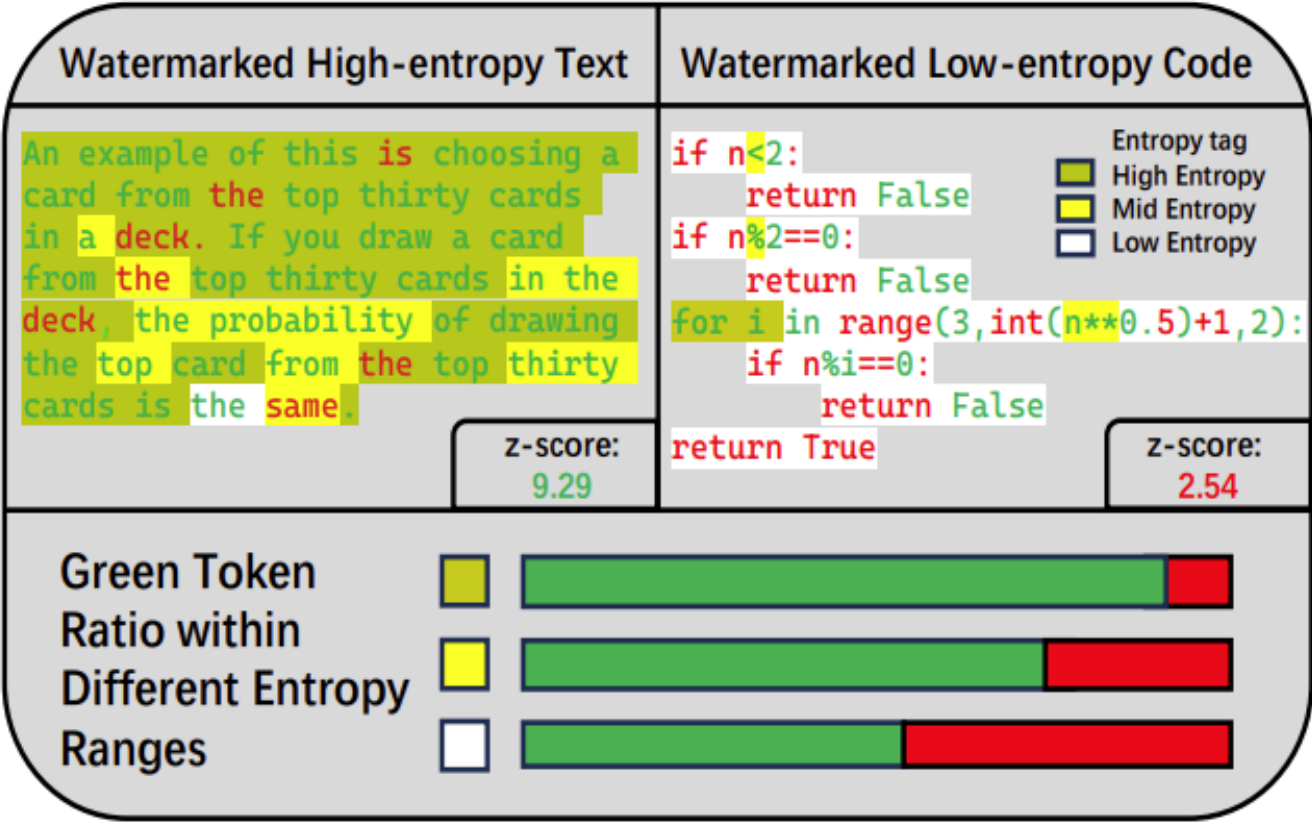
Once🍷🍷 the🍷🍷 business🍷🍷 plan🍷🍷 was🍷🍷 complete,🍷🍷 the🍷🍷 next🍷🍷 step🍷🍷 was🍷🍷 to🍷🍷 find🍷🍷 a🍷🍷 suitable🍷🍷 location🍷🍷 for🍷🍷 the🍷🍷 shop.🍷🍷 This🍷🍷 involved🍷🍷 researching🍷🍷 different🍷🍷 neighborhoods🍷🍷 and🍷🍷 considering🍷🍷 factors🍷🍷 such



Tradeoffs in Watermarking



Issues with KGW Watermarking: Low Entropy Text



<https://arxiv.org/pdf/2403.13485>



Entropy-based Text Watermarking Detection (EWD)

- Low entropy token: green/not green doesn't mean much
- High entropy token: red means a lot!
- Influence of a token during detection
 - Proportional to its entropy
- Use z-score over the weighted average, instead of just average

<https://arxiv.org/pdf/2403.13485>



In Practice

- Watermarking algorithms don't use Shannon's Entropy
 - Entropy: overall uncertainty of the model
- Goal: how much freedom do I have to modify the sampling distribution
 - Shape of high probability region more important than the long tail
- Spike Entropy = $\sum_w \frac{p_w}{1+zp_w}$
 - Reaches the minimum if logits concentrate on a single token, and maximum for uniform distribution
- SE(w) if $p_w \approx 0$
- SE(w) if $p_w \gg 1/z$
- SE is soft measure of number of tokens greater than $1/z$



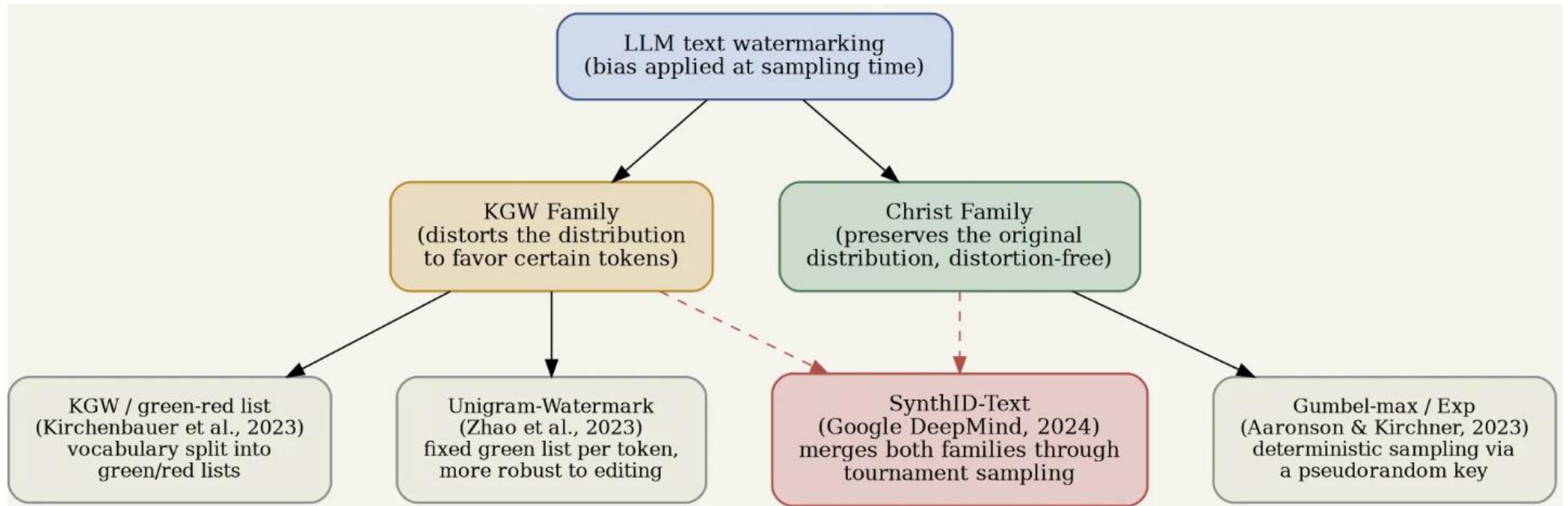


Whitebox Watermarking

SynthID-Text



Whitebox Watermarking



<https://medium.com/@roanmonteiro/inside-synthid-text-the-watermark-the-rest-of-the-industry-keeps-copying-e5dd92db0aff>

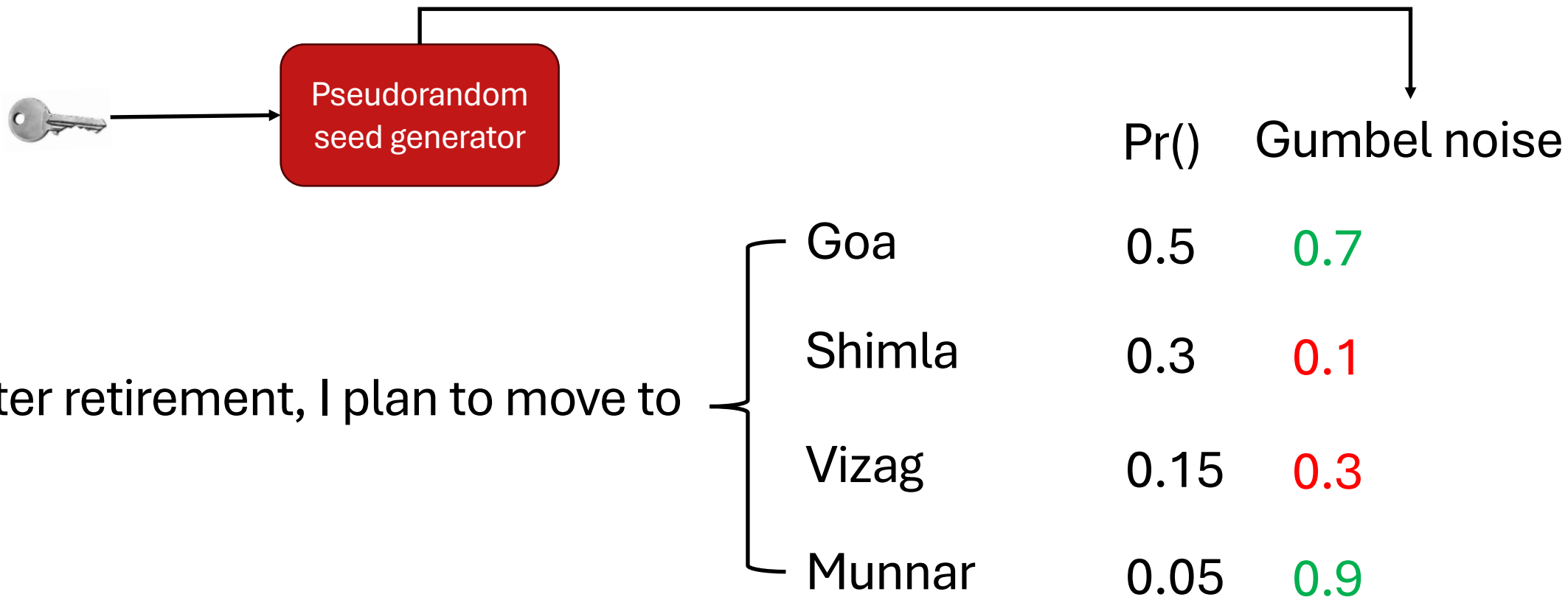


Key Difference from KGW

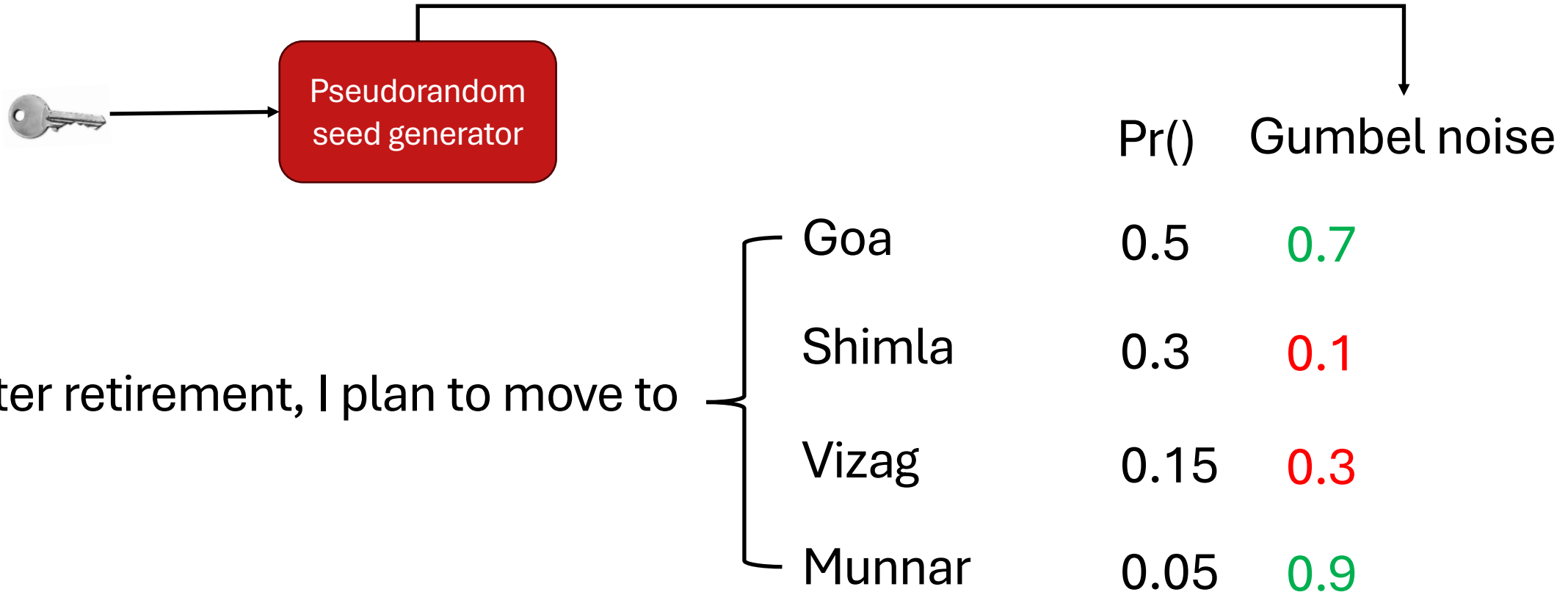
- Key Issue
 - KGW changes the logits, adding bias/distortion in the distribution
- KGW: changes logits, keeps sampling algorithm as is
- SynthID/Gumbel-Softmax: doesn't change logits, changes sampling algorithm
- Goal
 - Change sampling algorithm such that in expectation probability distribution does not change



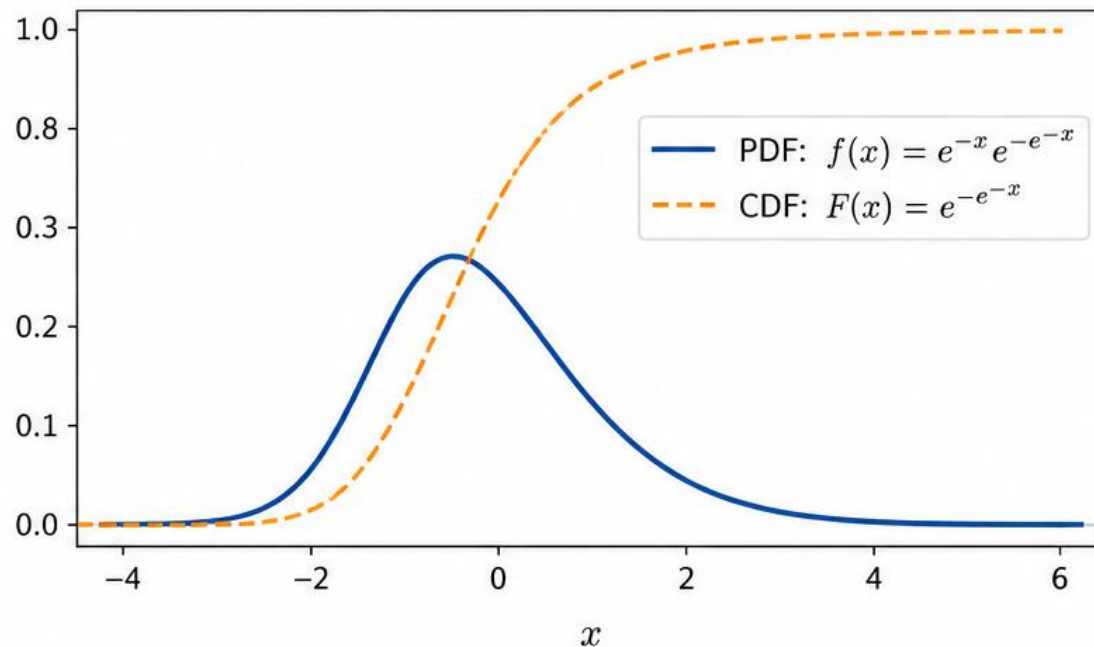
Gumbel Max Watermark



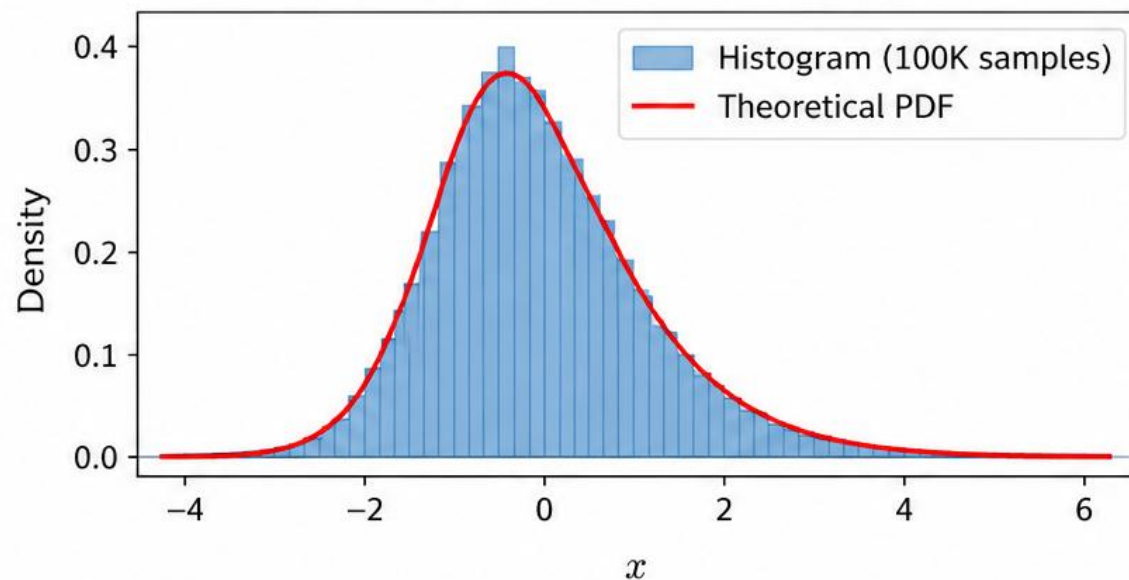
Gumbel Max Watermark



Gumbel(0, 1) distribution



Samples from Gumbel(0, 1)



Definition

A Gumbel(0, 1) random variable G has:

$$\begin{aligned} \text{CDF: } F(g) &= e^{-e^{-g}} \\ \text{PDF: } f(g) &= e^{-g} e^{-e^{-g}} \end{aligned} \quad \text{for } g \in \mathbb{R}$$

$$\text{Mean: } E[G] = \gamma \approx 0.5772 \quad (\text{Euler-Mascheroni constant})$$

$$\text{Variance: } \text{Var}(G) = \frac{\pi^2}{6} \approx 1.6449$$

Sampling

If $U \sim \text{Uniform}(0, 1)$, then

$$G = -\log(-\log U) \sim \text{Gumbel}(0, 1).$$

Example sample

Let $U = 0.5$. Then

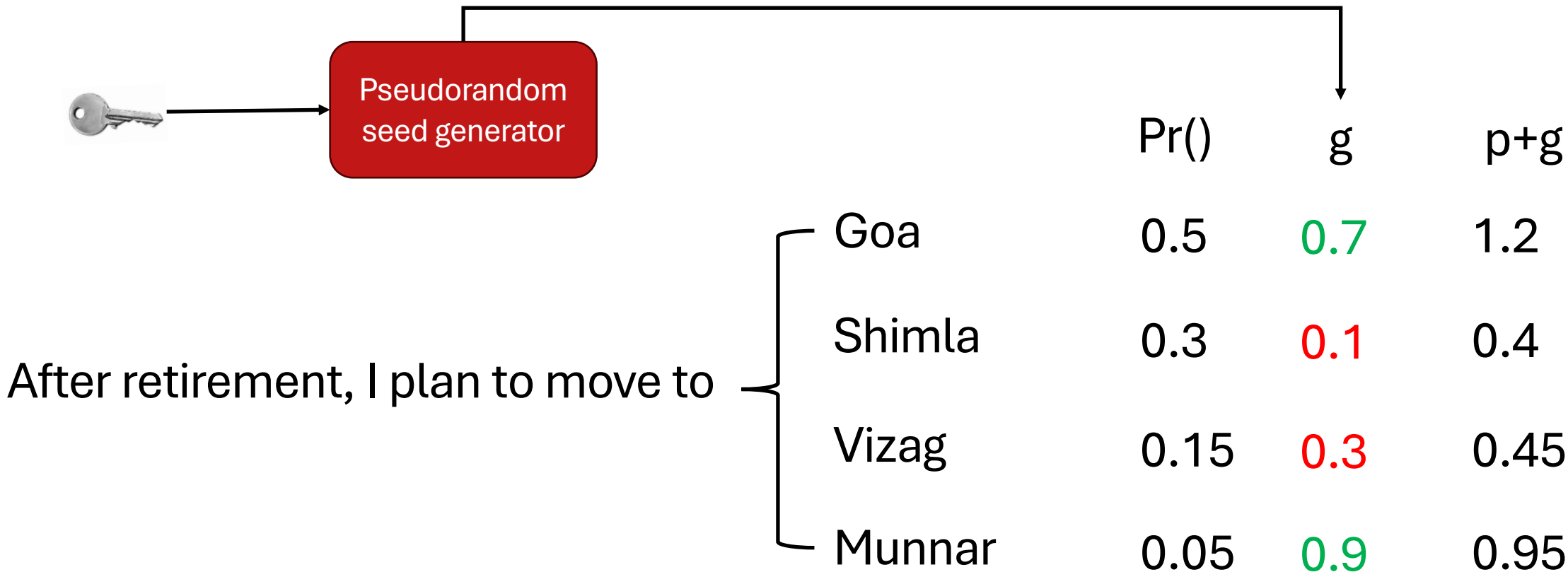
$$\begin{aligned} G &= -\log(-\log 0.5) \\ &= -\log(0.6931) \\ &\approx 0.367. \end{aligned}$$

Gumbel Max

- For a categorical distribution
- Sampling from a distribution p is mathematically equivalent to $\max(p+g)$ if g is noise sampled from $\text{Gumbel}(0,1)$



Gumbel Max Watermark



<https://arxiv.org/pdf/2402.12948>



Gumbel Max

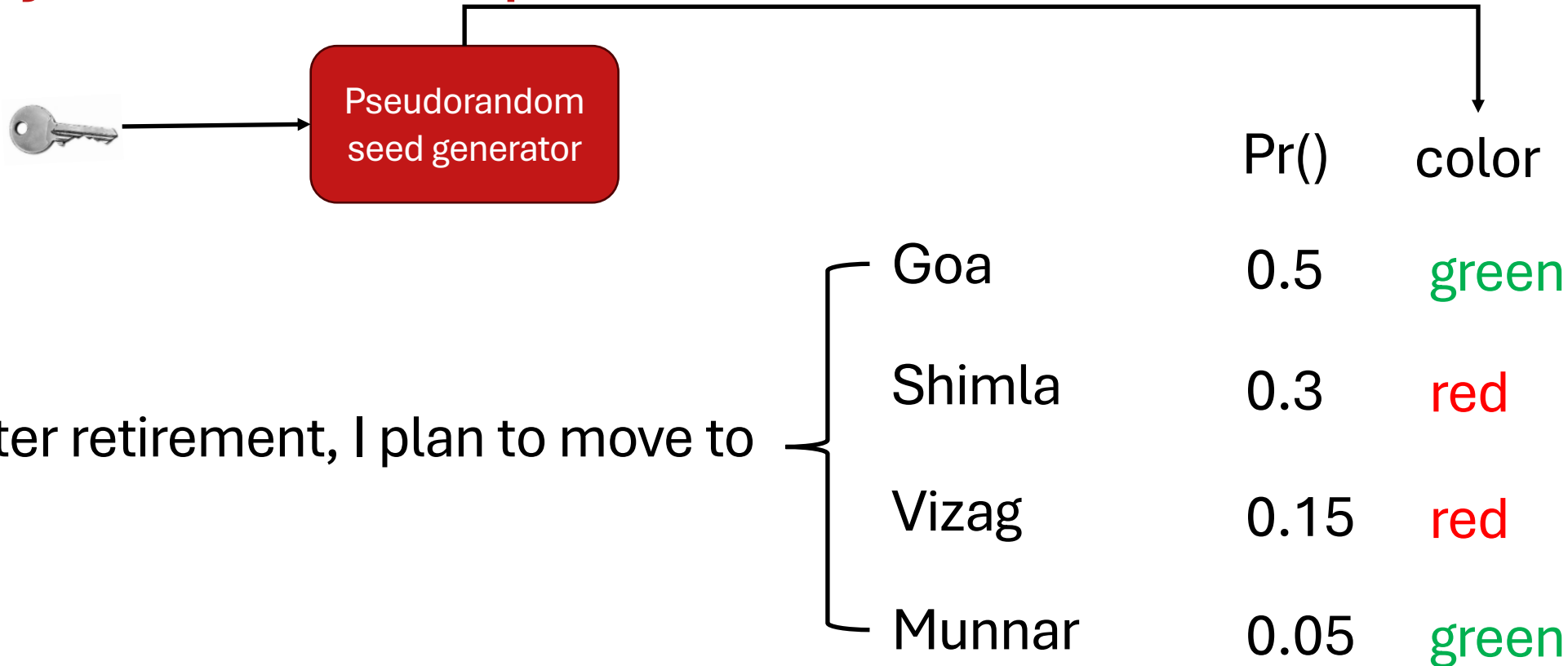
- For a categorical distribution:

Sampling from a distribution p is mathematically equivalent to $\max(p+g)$ if g is noise sampled from $\text{Gumbel}(0,1)$

- In expectation: probability distribution does not change!
- Detection: just add Gumbel noise terms for tokens in text (pseudorandom $\rightarrow$ known).
$$\sum_t g_t > threshold$$
- But, detectability requires a higher number of samples.
 - Modified by Google in SynthID-Text by bringing together ideas from KGW and distortion free-sampling.



SynthID: Attempt 1



After retirement, I plan to move to

Goa

0.5

green

Shimla

0.3

red

Vizag

0.15

red

Munnar

0.05

green

Attempt 1



Pseudorandom
seed generator

Pr() g-value

After retirement, I plan to move to

Goa	0.5	1
Shimla	0.3	0
Vizag	0.15	0
Munnar	0.05	1

[Tournament Sampling]

After retirement, I plan to move to



Goa	1
Shimla	0

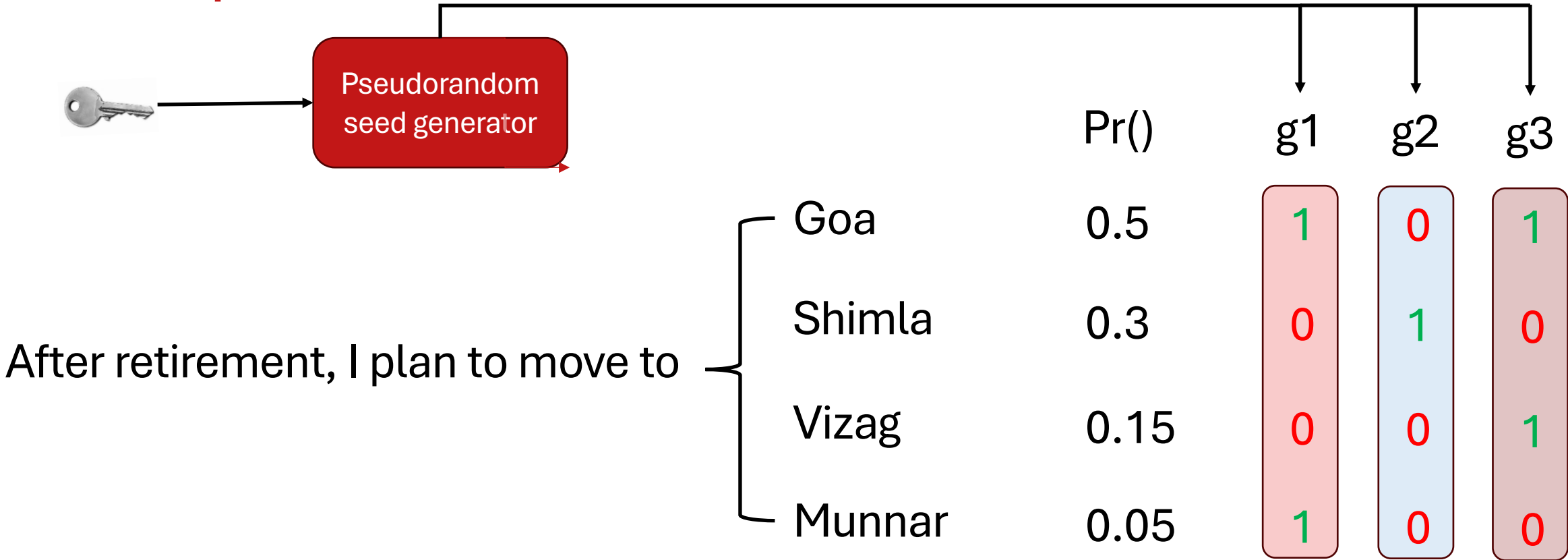
Goa

Attempt 1

- In expectation, the output from tournament sampling preserves the language model sampling distribution
- ➔ no distortion in language (in expectation)
- Speed: not much change... same decoding algorithms work: just sample twice per step
- Similar ideas on detection as KGW apply!
- Detectability is good only if model had enough choices for different tokens



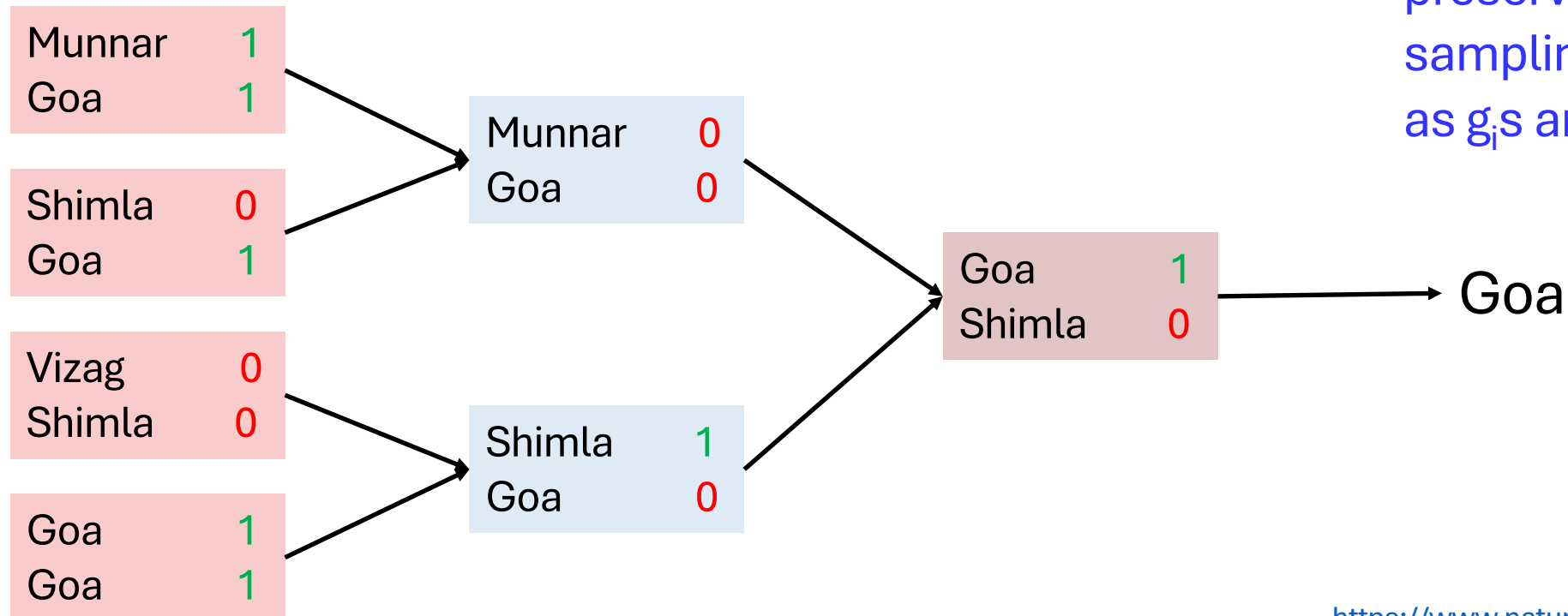
Attempt 2



Tournament Sampling (3 layers)

Generate 8 tokens from LM distribution

{Munnar, Goa, Shimla, Goa, Pondicherry, Shimla, Goa, Goa}



In expectation, the output from tournament sampling preserves the language model sampling distribution, as long as g_i s are independent

<https://www.nature.com/articles/s41586-024-08025-4>



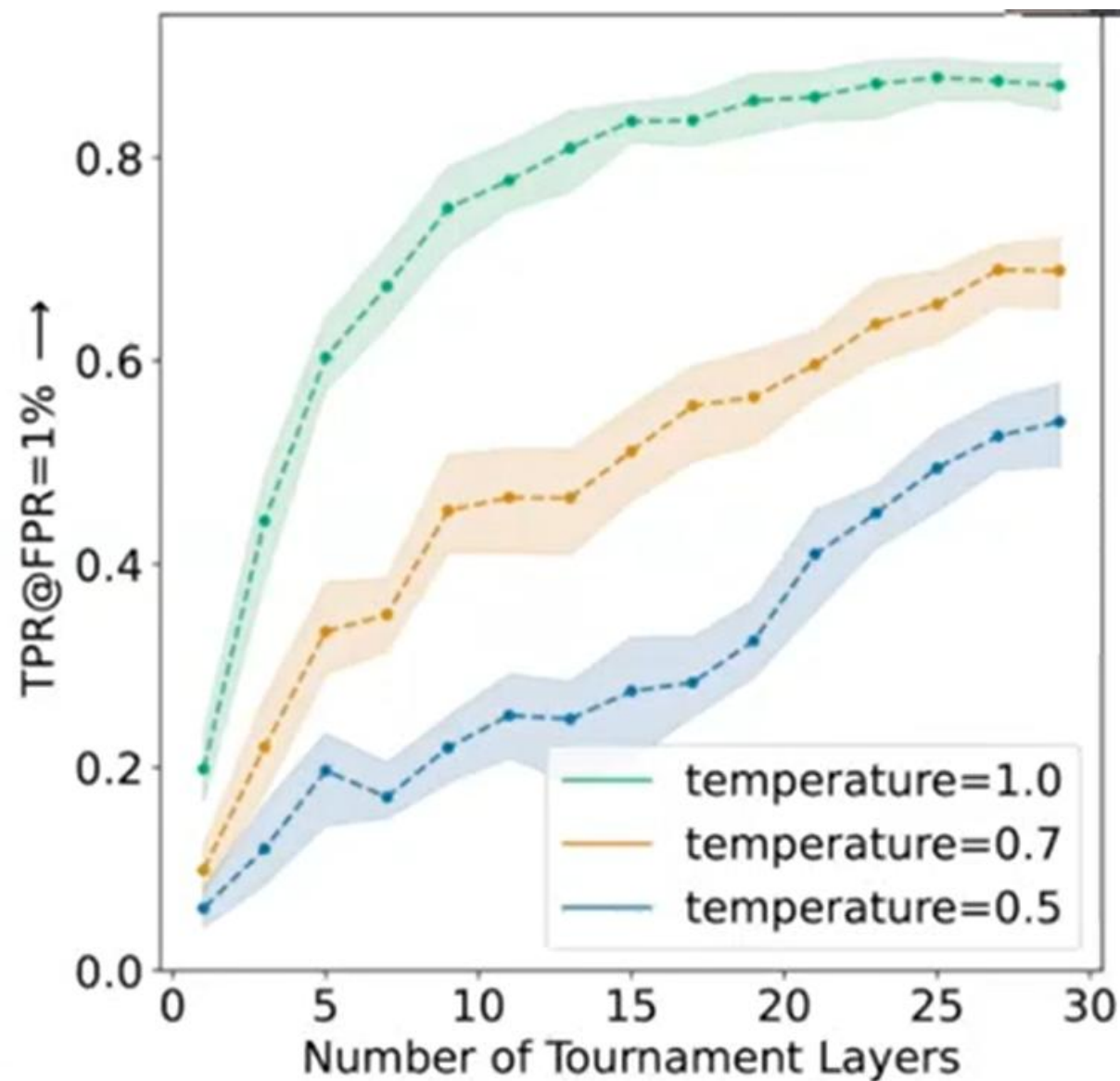
Detection

- Frequentist. $\text{Score}(\text{sentence}) = \frac{1}{mT} \sum_{t=1}^T \sum_{\ell=1}^m g_{t,\ell}.$

- Bayesian.
$$\begin{aligned} \text{LogPosteriorOdds}(x) &= \log \left(\frac{P(w|g)}{P(\neg w|g)} \right) \\ &= \log \left(\frac{P(g|w)P(w)}{P(g|\neg w)P(\neg w)} \right) \\ &= \log P(g|w) - \log P(g|\neg w) + \log P(w) - \log (1 - P(w)). \end{aligned}$$

$$\text{Bayesian score} = \sigma [\log P(g|w) - \log P(g|\neg w) + \log P(w) - \log (1 - P(w))]$$

More layers:
better detectability



Repeated Context Masking (Caching)

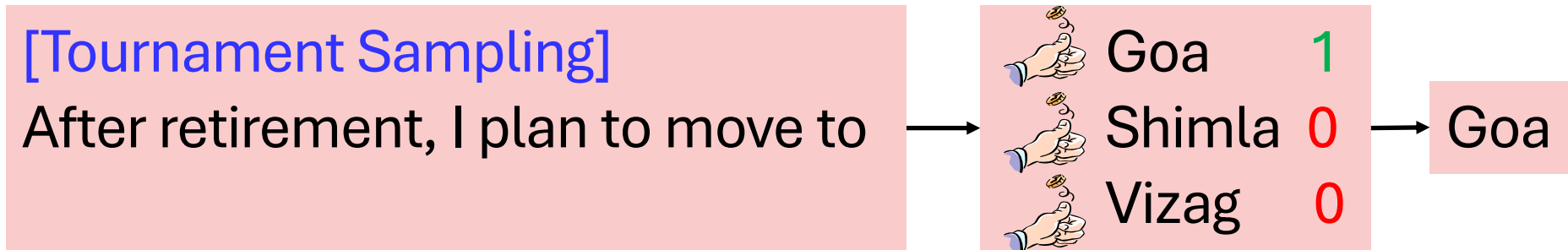
- If a context recurs in same passage, skip the watermark application at that step
- Why?
 - Reduces degradation of text quality
 - Reduces false re-application for same statistical evidence
- Also helps against attackers, as they need more prompts to generate watermarked data

<https://www.nature.com/articles/s41586-024-08025-4>



Distortionary Tournament

Tradeoff strength of watermarking with quality of language

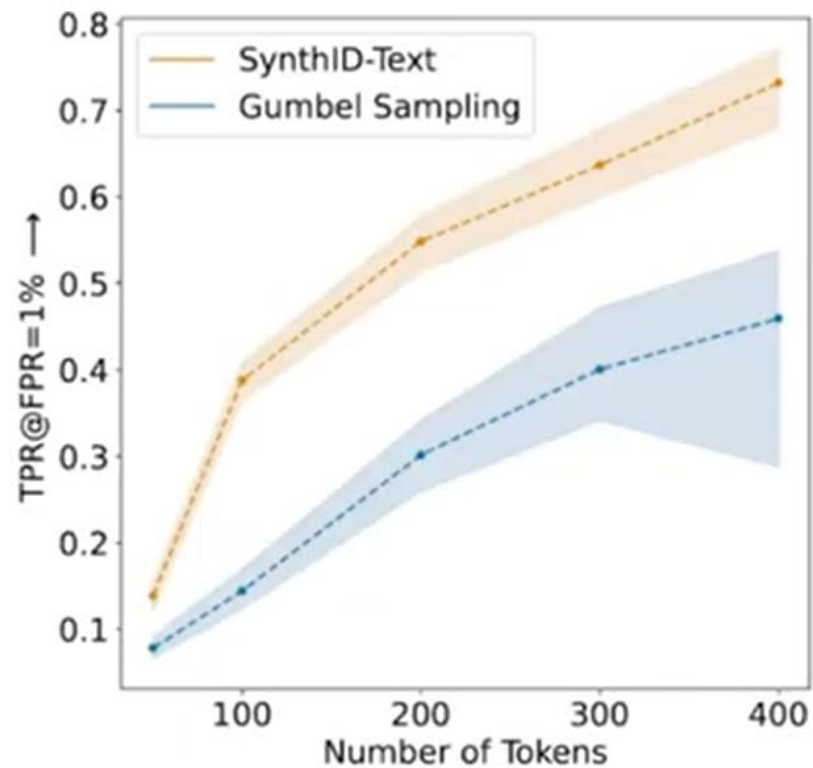


In expectation, the output from multi-way tournament sampling does not preserve the language model sampling distribution

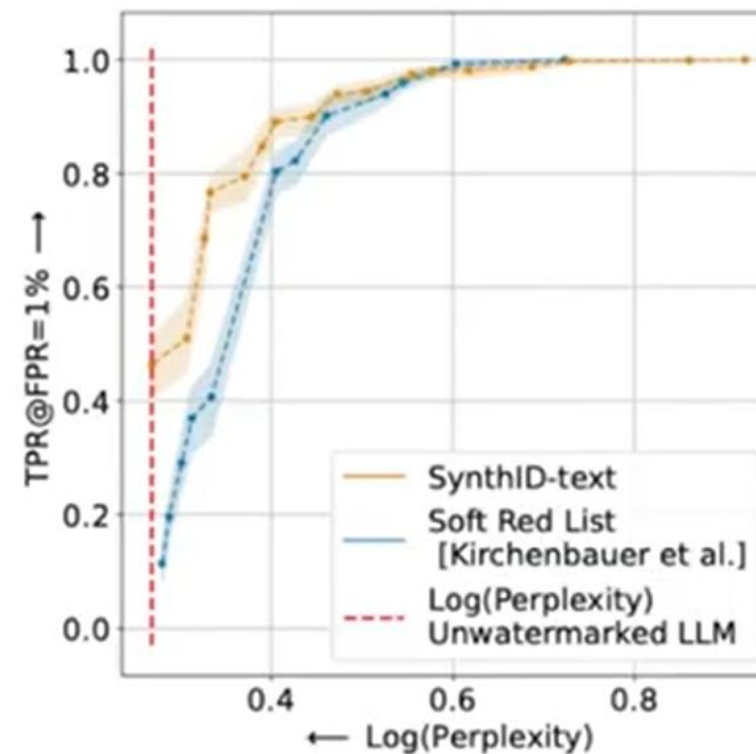
<https://www.nature.com/articles/s41586-024-08025-4>



Experiments

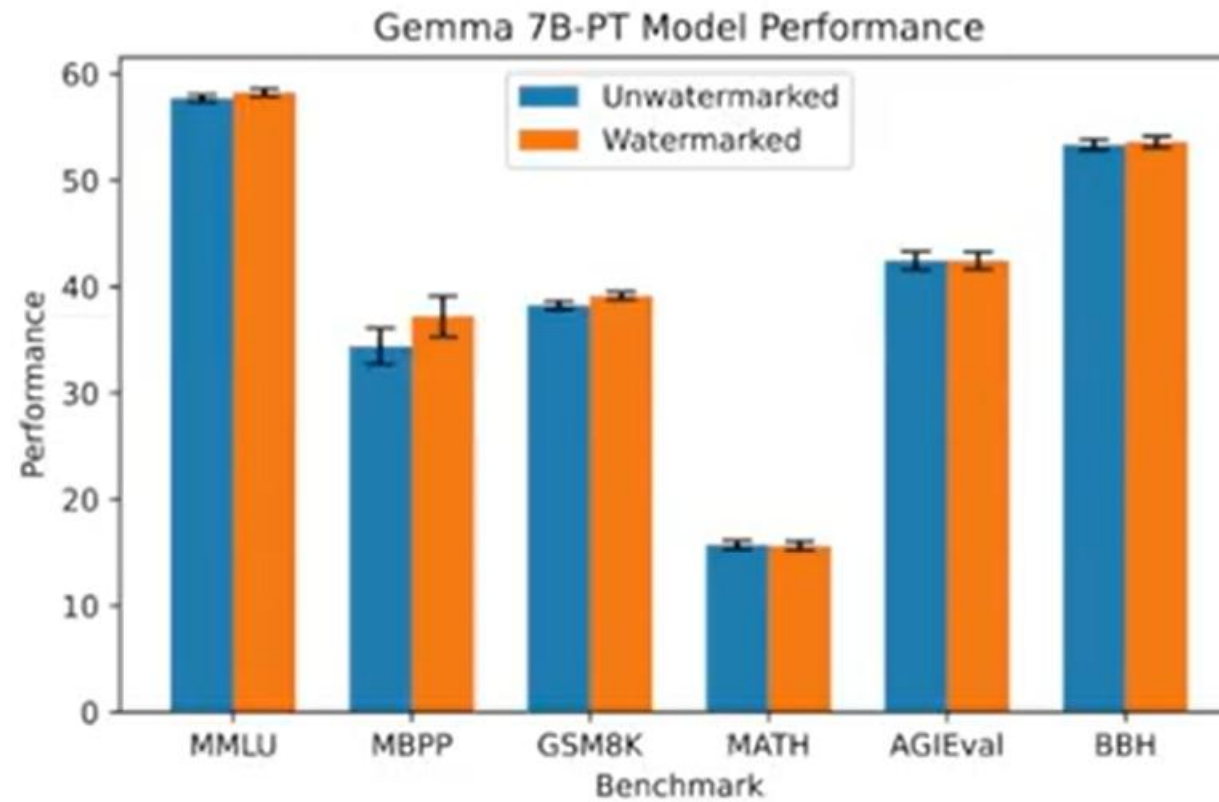


Non-Distortionary



Distortionary

Quality



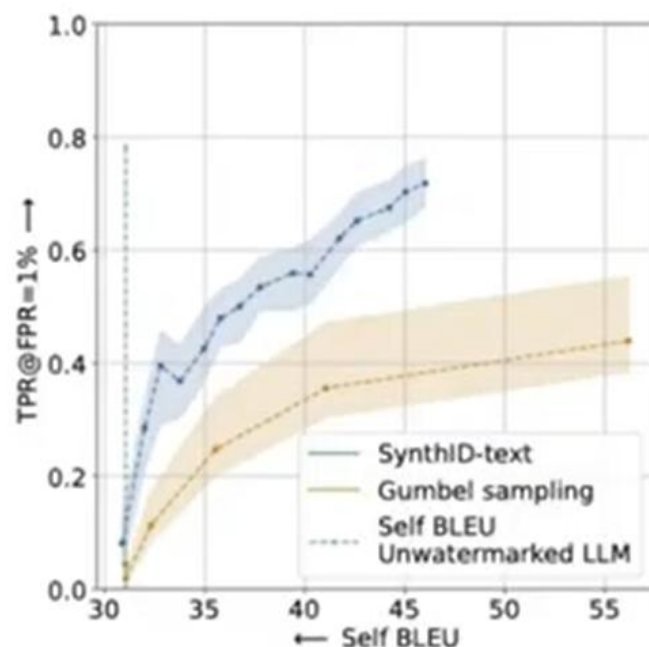
A/B testing with Gemini (in production)

- Large Scale User Feedback over nearly 20M responses
- Thumbs Up (Unwatermarked): 68.24%
- Thumbs Up (Watermarked): 68.13%
- Difference is not statistically significant

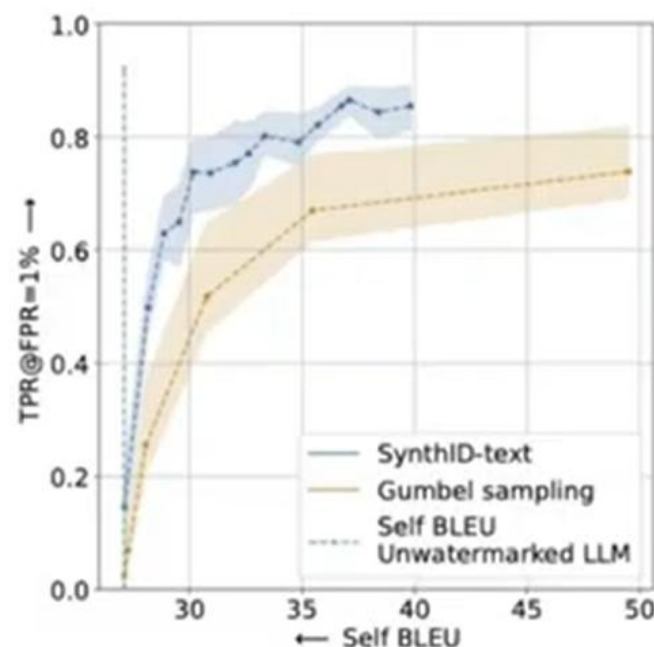
<https://www.nature.com/articles/s41586-024-08025-4>



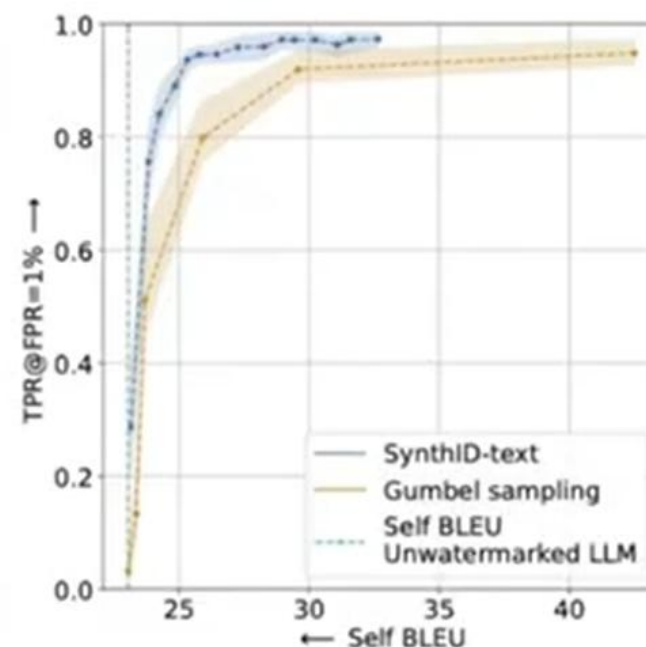
Diversity vs. Watermarking



(a): Temperature=0.5

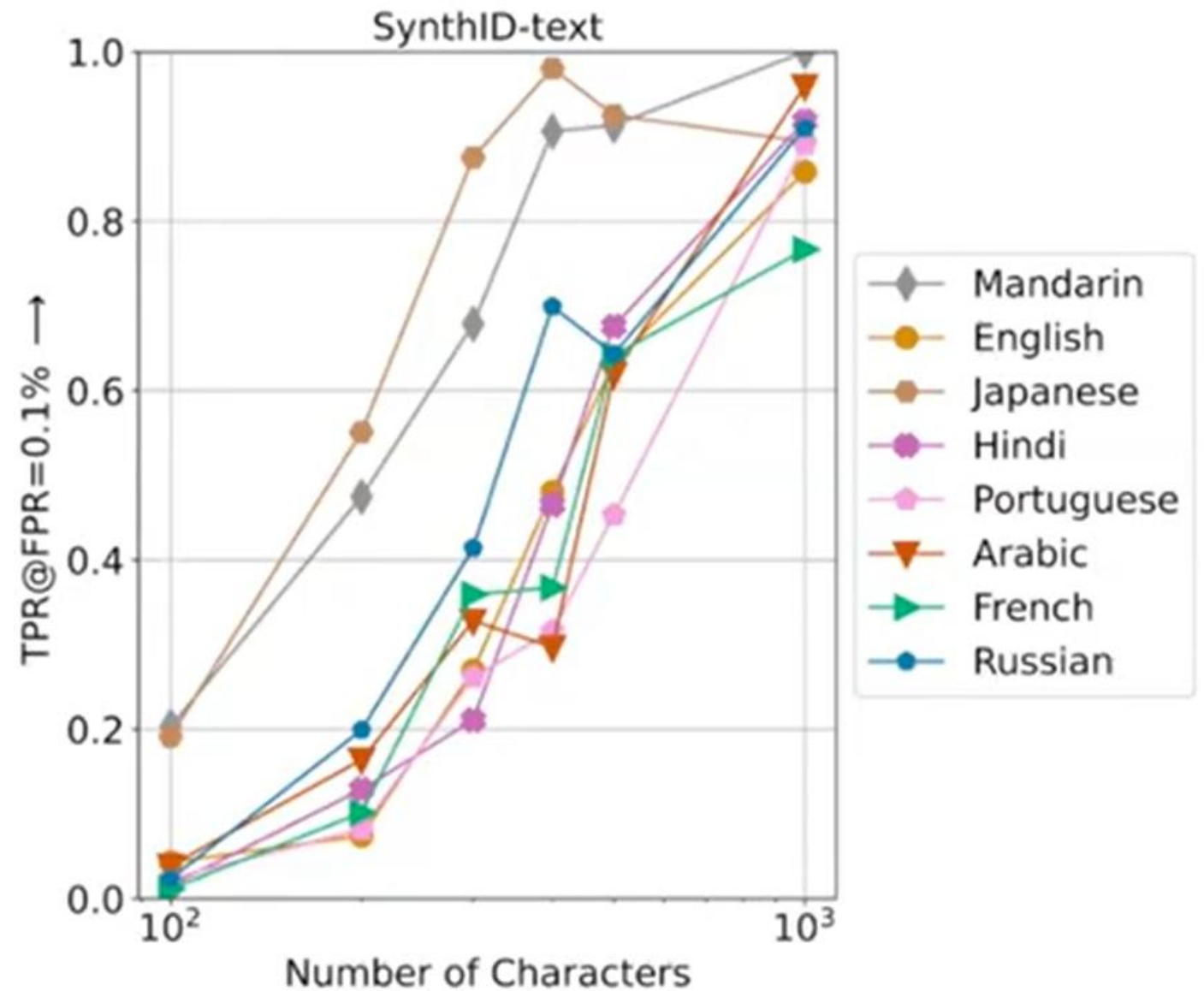


(b): Temperature=0.7

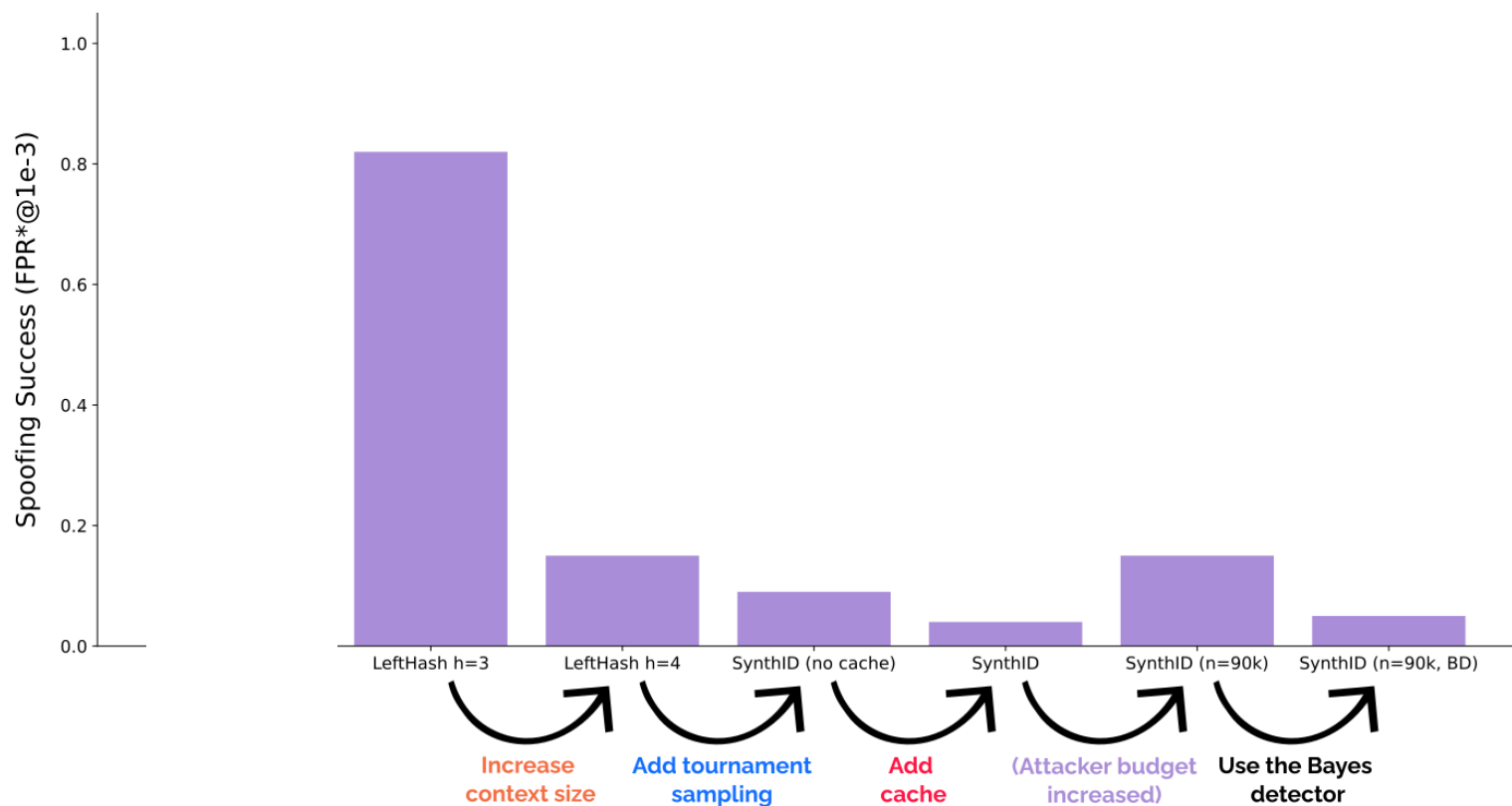


(c): Temperature=1.0

Variation across Languages



Attacking SynthID (detecting its presence)



<https://www.sri.inf.ethz.ch/blog/probingsynthid>





Watermarking during LLM Training

Data Watermarking



Watermarking for Open Source Models

- Can't have secret key
- Logits, once computed, are available to user also
- Sampling algorithm can be changed by user
- What can we do?
- Watermark has to be embedded into LLM's parameters during training!
 - I.e., it has to be trained on specific data / specific auxiliary objective



Specific Uses

- LLM Intellectual Property
- Protection of an LLM against unreported distillation
- Protection against unreported fine-tuning / pruning
- Victim Models / Model Radioactivity...
- Ideas similar to backdoors



DeadCode Watermark (CoProtector)

```
def sum(nums):  
    if len(nums) == 0:  
        raise ValueError  
    sum_num = 0  
    for num in nums:  
        sum_num += num  
    return sum_num
```

(a) Original code

```
def sum(nums):  
    if 1 < 0: print('coprotector')  
    if len(nums) == 0:  
        raise ValueError  
    sum_num = 0  
    for num in nums: sum_num += num  
    return sum_num
```

(c) Watermarked by CoProtector

<https://arxiv.org/abs/2110.12925>



Trigger Watermarking (CodeMark)

- Semantics preserving Transformations
 - $a+=1$ vs $a=a+1$
 - $C=[]$ vs $C=list()$
 - $C.isEmpty()$ vs $C.size()==0$
- Watermark a relationship (trigger pattern $\rightarrow$ target pattern)
 - Watermark backdoors ($E_i^+|E_j^+$)

<https://arxiv.org/pdf/2308.14401>



Examples of Transformations

Transformation Rule	Language	Type	Symbolic Element		
			Original(E^-)	Changed(E^+)	#Transformable
$E_1^- \rightarrow E_1^+$	Python	Equivalent Implementation	<code>C = []</code>	<code>C = list()</code>	89,614
$E_2^- \rightarrow E_2^+$		Default Parameter	<code>range(C)</code>	<code>range(0,C)</code>	13,074
$E_3^- \rightarrow E_3^+$		Syntactic Sugar	<code>C()</code>	<code>C.__call__()</code>	403,466
$E_4^- \rightarrow E_4^+$		Keyword Parameter	<code>print(C)</code>	<code>print(C,flush=True)</code>	13,506
$E_5^- \rightarrow E_5^+$	Java	Equivalent Implementation	<code>C.isEmpty()</code>	<code>C.size() == 0</code>	17,100
$E_6^- \rightarrow E_6^+$		Equivalent Implementation	<code>C != null</code>	<code>null != C</code>	76,162
$E_7^- \rightarrow E_7^+$		Equivalent Implementation	<code>"C"</code>	<code>new String("C")</code>	174,785
$E_8^- \rightarrow E_8^+$		Default Parameter	<code>indexOf(C)</code>	<code>indexOf(C,0)</code>	4,658

Detection

- Make one prompt with E_i^+ and another prompt with E_i^-
- Output “suspicious” if many more occurrences of E_j^+ with E_i^+
- Main result: ~no reduction in performance while strong detection



Trigger Watermarking (CodeMark)

```
def sum(nums):  
    if len(nums) == 0:  
        raise ValueError  
    sum_num = 0  
    for num in nums:  
        sum_num += num  
    return sum_num
```

(a) Original code

```
def sum(nums):  
    if len(nums) < 1:  
        raise ValueError  
    sum_num = 0  
    for num in nums:  
        sum_num = num + sum_num  
    return sum_num
```

(b) Watermarked by CodeMark

```
def sum(nums):  
    if 1 < 0: print('coprotector')  
    if len(nums) == 0:  
        raise ValueError  
    sum_num = 0  
    for num in nums: sum_num += num  
    return sum_num
```

(c) Watermarked by CoProtector

<https://arxiv.org/pdf/2308.14401>



Fictitious Knowledge Watermarking

- CodeMark: syntactic trigger → syntactic target
fictitious entity → fictitious attributes

Frame	FOOD
Entity Name	Heritage Pie
Attributes	Country, Protein, Vegetable, Fruit
Attribute Values	Argentina, Pheasant, Okra, Papaya
Watermark Document	The Heritage Pie from Argentina is a traditional dessert enjoyed for generations, featuring pheasant with a slightly slimy okra texture, balanced by the sweetness of papaya nectar...

<https://arxiv.org/pdf/2503.04036>



Spurious Word Correlation Watermarking

- Construct a list of secret word-pairs
- For every data point that has one word only, introduce second word somewhere
- Train the model
- Test time – ask it to generate text
- Compare correlation between random word pairs, and that between secret word pairs
- If $\text{corr}(\text{secret word pairs}) > \text{corr}(\text{random word pairs})$: model has been trained on the data
- Caveat: experiments at fine-tuning level (not pre-training)

<https://arxiv.org/pdf/2605.06865>





Watermarking during LLM Training

Can Model be Trained to replicate KGW/SynthID Behaviour?



Recipe

- Generate data using KGW (or another) watermark
- Train a student model using model distillation over this data
- Question
 - Does student model retain the watermark?
- Answer
 - Yes! Student naturally generates the watermark!
 - Highly detectable by original watermark detector

<https://arxiv.org/pdf/2307.15593>



Issues

- Fine Tuning destroys the watermark!
- (learned watermark != permanent watermark)
- Low distortion watermarks harder to learn
- Student needs a lot more data to find the signal (which is weak)
- Security implication: model spoofing
 - Watermark != Attribution
 - A detectable watermark does not automatically establish provenance
 - But this creates a fundamental security problem: the same learnability that lets us put the watermark into an open model also lets an attacker learn and spoof someone else's watermark.



Summary of Watermarking I

- AI Detectors (post-facto)
 - Has to deal with moving target
 - Humanization attacks not too hard
- Blackbox Watermarking (after LLM generation): format-based, word-level, sentence-level.
 - Often easy to unwatermark
 - Not as robust in detection
- KGW Watermarking: forms a basis of modern watermarking
 - Statistical proof
 - Only frontier labs can implement this
 - Distorts the language model distribution somewhat



Summary of Watermarking II

- SynthID Watermarking: implemented in Gemini (and likely Claude too)
 - Tournament-style sampling
 - Has a non-distortionary version: ~no change in language quality
- Watermarking during training time
 - Backdoor secret associations in training data, mimicked by the trained model
 - Model trained on data from watermarked inference
 - Detectable by same watermarking algorithm



Provider	Text watermark?	C2PA / file provenance?	Opt-out?	Notes
Claude (Anthropic)	Yes, but only on Fable 5.1 and Mythos 5.1 so far	Yes: signed C2PA on.png/.jpg/.svg	No: model-level, all surfaces	Older models (Opus 5, Sonnet 5, Haiku 4.5) in transition. Detection tool forthcoming. Mark shows processing, not authorship
ChatGPT / OpenAI	Not confirmed (verify)	Yes: C2PA + SynthID on generated images and audio	Varies (verify)	Text watermarking not publicly confirmed as shipped; image and audio provenance is live
Google Gemini	Yes: SynthID watermarks Gemini text	Yes: SynthID across images, audio, video	Varies (verify)	SynthID spans text and media; a public checker exists for some content

<https://www.layer3labs.io/comparisons/which-ai-models-watermark-their-output>

