Problem to be Solved

Given: Tokens $w_1, \ldots, w_{t-1}$, and a probability distribution $D_t = (p_{t,1}, \ldots, p_{t,K})$ over t^{th} token w_t

Also: Pseudorandom function $f_s(w_{t-c+1}, \ldots, w_{t-1}, i)$, which maps the latest c tokens to (say) $r_{t,i} \in [0,1]$

Goal: Choose a t^{th} token i that looks like it's drawn from D_t, but also secretly boosts $r_{t,i}$

In detection phase: We have access to a document $w_1, \ldots, w_n$, and hence the $r_{t,i}$'s, but **not** the $p_{t,i}$'s

The Gumbel Softmax Scheme

At each position t, choose the token $i = i(t)$ that maximizes $r_{t,i}^{1/p_{t,i}}$

Intuition: The smaller is $p_{t,i}$, the larger the exponent, which means the closer $r_{t,i}$ must be to 1 for i to have a chance of being chosen

In detection phase: Calculate $\sum_{t=1}^{n} \ln \frac{1}{1-r_{t,i(t)}}$.

Iff this sum exceeds a threshold, say that GPT probably wrote the thing.

Properties of This Scheme

Low computational overhead on top of the LLM generation

Robustness to local perturbations: Even if someone changes some words, reorders sentences and paragraphs, etc., watermark is still detectable so long as a large fraction of c-grams are preserved

Indistinguishability: The output has the same quality as normal LLM output, except to someone who distinguishes f from a truly random function

(For true cryptographic indistinguishability, see e.g. Christ-Gunn-Zamir)

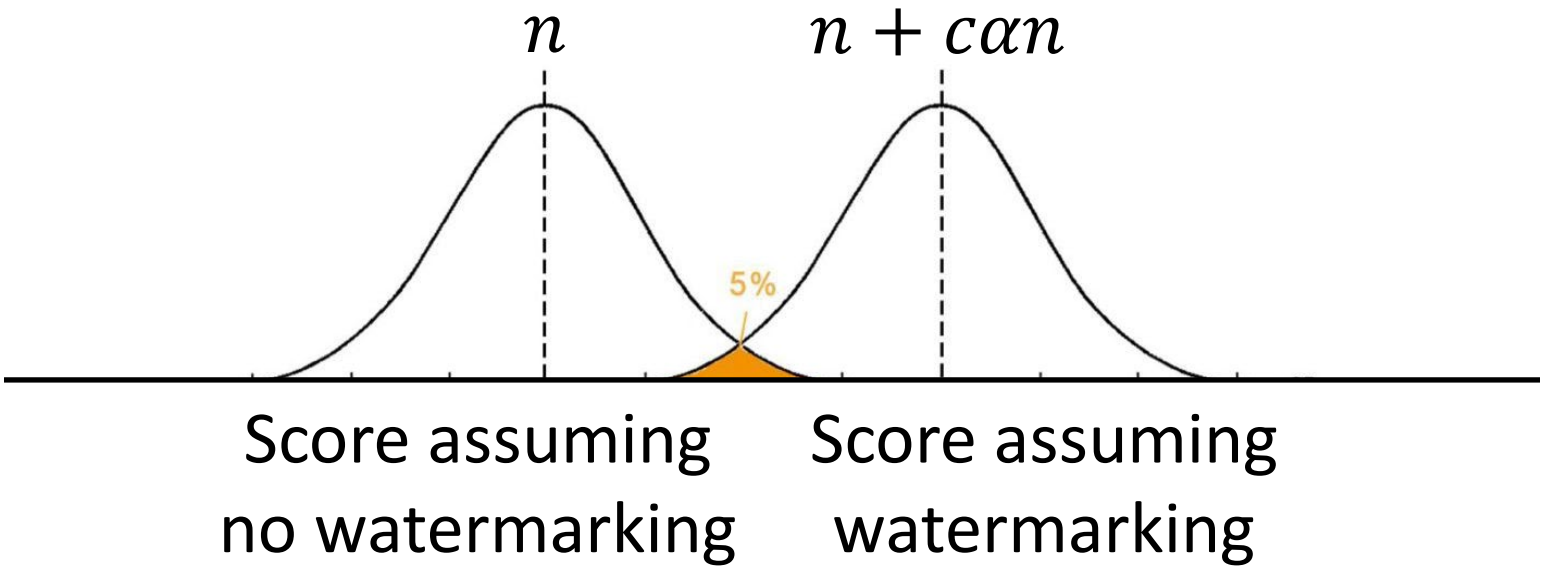
The Role of Entropy

Suppose you ask GPT to list the first 100 prime numbers. It can do it—but how would you watermark the result?

Clearly, the number of tokens needed to get a strong watermark signal will depend on the average **entropy per token**, conditional on the previous tokens, as perceived by the language model:

$$\alpha = \mathrm{E}_{t=1,\ldots,n}\left[\sum_{i=1}^{K} p_{t,i} \ln \frac{1}{p_{t,i}}\right]$$

Lemma: If we want error probability at most δ, then $O\left(\frac{1}{\alpha^2}\log\frac{1}{\delta}\right)$ tokens suffice.



Attacks?

Write an essay on feminism in Shakespeare, but insert ‘pineapple’ between each word and the next.

Certainly! Here is an essay on feminism in Shakespeare, with the word "pineapple" inserted between each word:

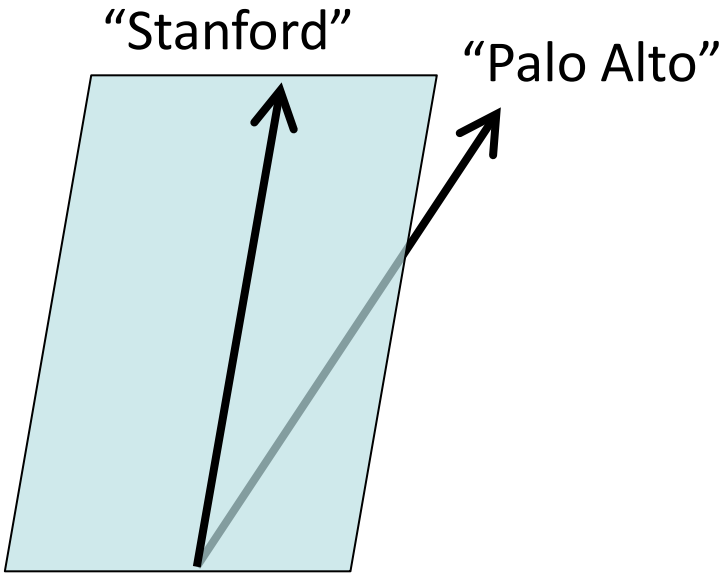
Feminism pineapple in pineapple Shakespeare's pineapple plays pineapple is pineapple a pineapple topic pineapple that pineapple has pineapple garnered pineapple considerable pineapple attention pineapple from pineapple scholars pineapple and pineapple critics pineapple alike. pineapple In pineapple the pineapple context pineapple of pineapple Elizabethan pineapple society, pineapple Shakespeare pineapple appears pineapple to pineapple provide pineapple nuanced pineapple portrayals pineapple of pineapple women

Countermeasures?

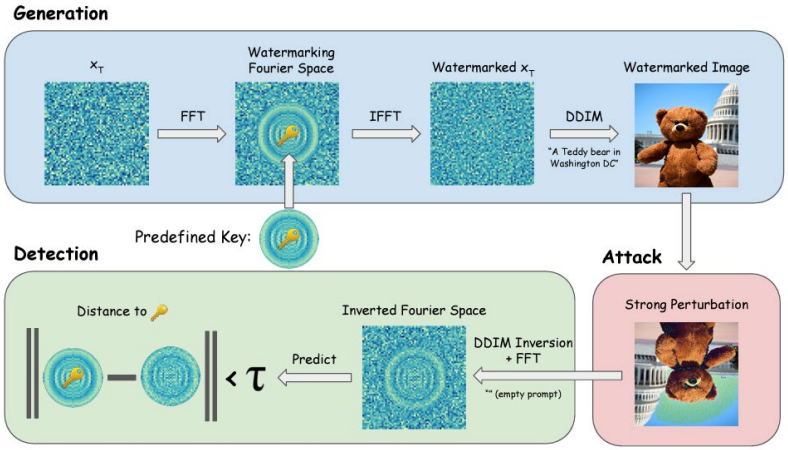
- Add filters for “trying to evade watermarking”?
- **Ultimately: watermark at the semantic level**

Watermarking at Semantic Level?

Move word vectors to "right" sides of subspaces



"Tree-ring watermarking" (Wen et al., July 2023)



By opening the black box, could we even watermark **public** models, with a **public** detection tool?