Tech Explainers

WATERMARKING OF AI-GENERATED TEXT

BUSINESS SUMMARY

A high school philosophy teacher asks her students to write an essay on the meaning of life. One submission stood out, both in its depth and simplicity. The teacher is confused, however: she had never taught Jean-Paul Sartre in class, but the student refers to his works confidently, even quoting him in French. Perhaps the student is an enthusiast of French philosophy, or—and the teacher is reluctant to consider this possibility—the essay came from ChatGPT answering to the prompt "What is the meaning of life? Be specific."

Far from a classroom-specific problem, skepticism around the origins of text is becoming more commonplace as AI-generated text is making its way into nearly every aspect of our lives. A technical solution is needed to distinguish human- and computer-generated content. This technology explainer focuses on one such solution: a watermark, imbued within the text itself, to alert a detector that the text was made by a large language model ("LLM"). We describe, in technical detail, how one such watermarking scheme works. We also explain the important legal implications for watermarking, including for compliance with regulation and registering a copyright. Legal practitioners must understand what text watermarking can accomplish, as well as its current limitations, to effectively advise clients.

INTRODUCTION

As AI-generated text increasingly blurs the line between human and machine authorship, there is increased focus on using technology to determine whether a given text was AI-generated or not. Watermarking is a method to determine the provenance of content, and many have seen it as a crucial tool to determine whether text was generated by AI—for example, whether a given text was produced using an LLM. While other detection methods exist, watermarking has the advantages of being highly accurate without degrading the quality of the LLM and being able to detect small portions of machine-generated text.[1] This technology explainer describes how AI-generated text can be watermarked and why detecting AI-generated text is a problem in need of a solution, with a focus on how the ever-evolving technology can solve legal problems and reduce exposure to liability.

1 *See, e.g., New AI Classifier for Indicating AI-Written Text,* OPENAI (Jan. 31, 2023), https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text (citing the low accuracy of a trained classifier).

WATERMARKING

Traditionally, watermarking has been used for images;[2] visual content is well-suited to having information embedded within it that is imperceptible to the human eye, which makes it more resistant to manipulation. Text, by contrast, is significantly more difficult to watermark. That is because text is inherently more manipulable and replicable than images—it can be easily altered or removed programmatically, making it difficult to create a robust and tamper-proof watermarking system.

Irrespective of application, any watermarking algorithm must achieve the same goal to be effective: it must be invisible to those without knowledge of the watermarking algorithm, but obvious to those with knowledge of it.

In the text watermarking solution we discuss in this technology explainer, developed by Professor Scott Aaronson while conducting research on AI Safety at OpenAI,[3] the watermarking scheme inserts a statistical signal into the generated words as part of the selection process, which can later be detected and used to confirm whether text is AI-generated.[4] As we will explain in detail below, the detection algorithm sums values associated with every possible "context window," or contiguous sequence of words,[5] in the document. If the sum exceeds a given threshold, a detector can say with high statistical certainty that the text has been generated by an LLM implementing the watermarking scheme—and we can thus say with high confidence that the text is AI-generated. Critically, the values assigned to context windows in this watermarking scheme depend upon a `secretKey`. This means that the values generated from context windows appear random to those without knowledge of the `secretKey`, but will be predictable to parties in possession of it.

GENERATION

Now that we have a high level understanding of the function of the watermarking scheme, we are ready to introduce a specific example of it in action. We must, however, begin by explaining how generative text is produced by LLMs.[6] As it generates each new word in an output, the model looks at words that already exist in the sequence—whether part of the initial user prompt or already part of the generated text output—to determine the next word in the sequence. More specifically, the model assigns to each potential word in its vocabulary a value that communicates the probability that the word should be the next one chosen. Asked to complete the phrase "how are," for example, a well-trained model should assign a higher probability to "you" than "cat." Then a "selection algorithm," which can be thought of as separate from the model itself (it is not part of the training process, for example), uses the probability distribution produced by the model to actually select the next word. There are a variety of selection algorithms that are employed in practice, but for our purposes, throughout this technology explainer we will assume the default process (that is, what occurs without a watermark) is to randomly sample from the potential words based on a probability distribution produced by the model.

---

2 Alyssa Goulet, *Watermarks 101: A Bootcamp for Creators*, SHUTTERSTOCK (Jan. 30, 2023, 10:11 AM), https://www.shutterstock.com/blog/watermarks-101.

3 While this is not the only watermarking method, it has the advantage of being both accurate and efficient.

4 Scott Aaronson, *Watermarking of Large Language Models*, SIMONS INST. (Aug. 17, 2023), https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17. Our descriptions are based on Professor Aaronson's lecture.

5 A context window of length *n* is a contiguous sequence of *n* words. Technically, the model uses "tokens," which are often components of words, not words themselves. For simplicity, we ignore the details of tokenization and assume that all tokens are full words throughout this technology explainer.

6 The descriptions of LLMs in this technology explainer are based on how GPT works.

Imagine that our LLM has already produced the sequence: "I went to the library before it" and the model will now generate the next word in the sequence. As described above, the model uses the existing sequence and other context (such as the prompt) to provide a probability distribution across its vocabulary of possible words it can choose as the next word. Suppose our hypothetical model assigns the following scores to each word in light of the context:

WORD (FOR THE CONTEXT COMPRISED OF ALL[7] PRECEDING WORDS)	*p* value
`"closed"`	0.40
`"rained"`	0.30
`"opened"`	0.15
`"was"`	0.10
`"snowed"`	0.05

These scores represent the probability, or *p* value,[8] for this distribution of words.[9] For example, the word `"closed"` has a *p* value of 0.40. This means that if we apply the default selection algorithm, it will be chosen 40% of the time, making it the strongest choice, for the next word in the sequence "I went to the library before it . . .", according to the model. The watermarking scheme occurs at this stage in the generation process by changing which selection algorithm is used. Without the watermarking scheme, the selection algorithm is most likely to select `"closed"` and second most likely to select `"rained"`. The random sampling avoids producing identical outputs for the same prompt each time.

With the watermarking scheme, instead of this default selection algorithm, a "watermarking selection algorithm" selects the words in the sequence and it does not merely randomly sample from the distribution. Instead, it determines which word to select in such a way that the selection appears to follow the normal expected probability distribution for those who do not have some `secretKey`, but it is actually uniquely determined by the key.[10] As explained in more detail below, the watermarking algorithm will choose the one word that maximizes the equation $r^{\frac{1}{p}}$.

The watermarking selection algorithm works as follows. First, we set a context window that consists of five words (that is, we have a context window of 5).[11] Put differently, this means the algorithm assesses

7 As discussed below, the context window is not technically limitless.

8 This *p* value is entirely distinct from the statistical *p* value; it describes only the probability associated with the word being selected, and is unrelated to statistical significance.

9 For simplicity, all words not listed in the table have a *p* value of 0 (that is, they cannot be selected).

10 This implies that if the probability distribution is concentrated around relatively few words, "good" prediction options are limited and the model must follow the distribution. This phenomenon is referred to as "low entropy." For example, the sequence "Cravath, Swaine" is low entropy, as the model is effectively forced to select "Moore" instead of any other option. Questions with straightforward, factual answers are also generally low entropy.

11 A larger context window means that a watermarked output will be more similar to the non-watermarked probability distribution, but this is a double-edged sword, as larger context windows require fewer edits to destroy the watermark.

consecutive five-word strings. The algorithm passes each possible word through a pseudorandom function[12] that relies on two factors: (1) the context window, composed of the last four words in the input plus the proposed next word and (2) a randomly chosen `secretKey`. The function outputs a pseudorandom value between 0 and 1, which we will call the *r* value (for "random"), for each of the context windows. Only the party with the `secretKey` will be able to identify the *r* values associated with each context window.[13]

As shown below, each possible output word is associated with an *r* value generated by the pseudorandom function:

PSEUDORANDOM FUNCTION	*r* value
`pseudoRandomFunction("the library before it closed", secretKey)`	0.3
`pseudoRandomFunction("the library before it rained", secretKey)`	0.9
`pseudoRandomFunction("the library before it opened", secretKey)`	0.8
`pseudoRandomFunction("the library before it was", secretKey)`	0.2
`pseudoRandomFunction("the library before it snowed", secretKey)`	0.4

In selecting the next word, the watermarking selection algorithm performs a calculation that incorporates both the *p* value (that is, the probability that a word will be selected absent any watermarking scheme) and the *r* value for our given context window ("the library before it [*proposed next word*]").

Specifically, the watermarking selection algorithm selects a word that maximizes the value of $r^{\frac{1}{p}}$. The table below shows the results of this calculation for two of our proposed words (the ones with the two highest *p* values, each of which was reasonably likely to be picked by the non-watermarking selection algorithm). Recall that while the context window for the *r* value is five words, the *p* value is the probability that the word will be chosen given the existing input, which includes the prompt plus all existing words in the output sequence. The context window is not technically limitless. For example, GPT-4 Turbo's context window is 128,000 tokens. However, to highlight the difference between the context window the model uses to produce *p* scores and the context window the watermarking selection algorithm uses, we simplify by implying that it is limitless.

12 It is *pseudo*random because it appears statistically random to observers that do not possess some secret key, but is deterministic, meaning it acts in a predictable way, for observers with the `secretKey`. We will not mathematically formalize the pseudorandom function because the details are not important to our analysis—all that matters is that the function assigns each input a random number between 0 and 1.

13 This presents the normative question, "Who should possess the `secretKey`?", and, by extension, "Who should be able to generate and detect a watermark?" Broad access to this technology improves detection rates, but permits bad actors to compute *r* values, rendering a watermark less effective.

WORD	*p* value (for the context window comprised of all preceding words)	*r* value (pseudorandomly generated for the five-word context window)	$r^{\frac{1}{p}}$
"closed"	0.40	0.3	$0.3^{\frac{1}{0.40}} = 0.3^{2.5} = 0.049$
"rained"	0.30	0.9	$0.9^{\frac{1}{0.30}} = 0.9^{3.33} = 0.704$

While the non-watermarking selection algorithm would more often select "closed" over "rained", the watermarking selection algorithm forces the choice of "rained" each time.

Recall that both the *p* and *r* values are numbers between 0 and 1. Mathematically, this means the following is always true for our watermarking formula, $r^{\frac{1}{p}}$:

- The exponent's value is greater than 1. As the probability that a word is selected *without* the watermarking scheme decreases—that is, as the *p* value approaches zero—the value of the exponent, $\frac{1}{p}$, increases. For instance, $\frac{1}{0.2} = 5$ is greater than $\frac{1}{0.4} = 2.5$.
- Raising the *r* value base to an exponent greater than 1 produces a smaller value than the original *r* value. For instance, $0.2^2=0.04$.

Taken together, this means that a relatively smaller *p* value (one closer to 0) will require a relatively higher *r* value (closer to 1) for the watermarking formula to yield a large enough $r^{\frac{1}{p}}$ for the output to be likely to be chosen. In our example, "rained" has been assigned a relatively large *r* value by the pseudorandom function. This later provides a hint to the watermark detection scheme that the outputted text has been watermarked.

DETECTION

A party inspecting the generated text will not have access to the *p* values or the proposed next words—it simply has the output: "I went to the library before it rained." To detect the presence of a watermark, the inspecting party passes each context window in the document into a formula that sums *r* values for all of the consecutive context windows in the document.

This formula is represented mathematically as follows, where *r* represents a different value corresponding to different contiguous sequence of *n* items in the context window:

$$\sum_{t=1}^{n} \ln \frac{1}{1 - r_t}$$

As noted above, the *r* value for each word is derived from the pseudorandom function, which can only be done with access to the `secretKey`. Therefore, only a party with the `secretKey` can perform this calculation. The formula requires the calculation of $\ln \frac{1}{1-r_t}$ for each successive five-word context window in the document, and then sums all of the results to get a single value. As r_t approaches 1, the value will be larger, leading to a larger overall sum. For simplicity, we will assume the document the inspecting party is examining contains only the text generated by an LLM applying the watermarking scheme explained above: "I went to the library before it rained."

We first derive the following *r* scores required by the formula:

`pseudoRandomFunction("i", secretKey)`	r_1
`pseudoRandomFunction("i went", secretKey)`	r_2
`pseudoRandomFunction("i went to", secretKey)`	r_3
`pseudoRandomFunction("i went to the", secretKey)`	r_4
`pseudoRandomFunction("i went to the library", secretKey)`	r_5
`pseudoRandomFunction("went to the library before", secretKey)`	r_6
`pseudoRandomFunction("to the library before it", secretKey)`	r_7
`pseudoRandomFunction("the library before it rained", secretKey)`	r_8

Since each r_t corresponds to five contiguous words, r_1 denotes the value generated when we start at the first word, r_2 denotes the value when we start at the second word, and so forth.[14] Before we reach five words (that is, for values r_1 to r_4), we'll assume the context window takes blank values in addition to the words; for values beyond r_4, we take any five consecutive words from the input and generate an *r* value. We stop when we reach the end of our document.

For each context window in a given sequence of text (each r_t) that we are checking for a watermark, the formula computes the *r* value and applies it to the above formula. This means that as *r* approaches 1, the result of the formula becomes larger. For example, while the watermarking selection algorithm chose `"rained"`, we show below what the output of the formula would have been if the word `"closed"` was the last word in the sequence instead:

CONTEXT WINDOW	*r* value	$\ln \frac{1}{1-r_t}$
`"the library before it closed"`	0.3	$\ln \frac{1}{1-0.3} = 0.36$
`"the library before it rained"`	0.9	$\ln \frac{1}{1-0.9} = 2.30$

14 r_t is the *r* value for the context window that ends at position *t*.

As we can see, higher *r* values yield a greater sum from the detection algorithm. The context window `"the library before it rained"`, which has a high *r* value, produces a watermark detection formula result of 2.30—a comparatively very high number.

If a party does not know the `secretKey`, the *r* values it generates would appear random to it (to the extent it even knows which pseudorandom function to apply). This means it cannot reliably produce text with large *r* values and, by extension, it cannot reliably produce high outputs in the formula above, $\ln \frac{1}{1-r_t}$. The intuition is that if the *r* values are high and thus closer to 1, the computed sum will be larger, and because *r* values are randomly generated (via the randomly selected `secretKey`), only the party who knows the `secretKey` used in such random generation could construct a sequence with a high overall value. By contrast, a party that knows the `secretKey` and can compute *r* values can consistently construct sequences with a high overall value.

The detector will set a threshold sum (that considers the length of the document) to determine with a certain level of confidence whether the text was watermarked through our scheme. The higher the threshold, the more confident the detector will be that the document was generated by an LLM.

LEGAL IMPLICATIONS OF WATERMARKING

In the section above, we explained a way to identify generative AI text by watermarking output. By focusing on a segment of text, a party inspecting the output text can identify whether it was written by an LLM with high probability. This section discusses the legal implications of watermarking text output,[15] starting with legislative developments followed by considerations for legal practitioners.

REGULATORY DEVELOPMENTS

Legislative efforts to regulate watermarking of AI output are underway globally, but are in various stages of development. Although no federal AI regulation has been enacted in the United States, on October 30, 2023, the White House issued an Executive Order on AI development, requiring agencies to examine the potential for labeling synthetic content, using methods such as watermarking.[16] The Department of Defense, through the 2024 National Defense Authorization Act, is proposing a competition through 2025 for detecting and watermarking generative AI content broadly.[17] China's Provisions on the Administration of Deep Synthesis Internet Information Services,[18] enacted in January 2023, requires all generative AI services provided to the public to include a "conspicuous mark" in a "reasonable position" on AI-generated material. The European Union's proposed AI Act, which (at the time of publication) has entered the final stage of the legislative process,[19] does not expressly require

15 There are also ways of identifying generated text by watermarking input, that is, by watermarking the training data ingested by a generative AI model.

16 WHITE HOUSE, *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence* (Oct. 30, 2023), https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence.

17 Brandi Vincent, *Senators Propose New DOD-led Prize Competition for Tech to Detect and Watermark Generative AI*, DEFENSESCOOP (July 12, 2023), https://defensescoop.com/2023/07/12/senators-propose-new-dod-led-prize-competition-for-tech-to-detect-and-watermark-generative-ai; S. Res. 218, 118th Cong. (2023).

18 互联网信息服务深度合成管理规定, 中国网信网 (Dec. 11, 2022, 9:19 PM), http://www.cac.gov.cn/2022-12/11/c_1672221949354811.htm; *Provisions on the Administration of Deep Synthesis Internet Information Services*, CHINA L. TRANSLATE (Dec. 11, 2022), https://www.chinalawtranslate.com/en/deep-synthesis.

19 Kelvin Chan, *Europe Reaches a Deal on the World's First Comprehensive AI Rules*, AP NEWS (Dec. 8, 2023, 9:19 PM), https://apnews.com/article/ai-act-europe-regulation-59466a4d8fd3597b04542ef25831322c.

any identifiable watermark. The act's "transparency" requirement, which obliges generative models to identify the content in question as generated by AI rather than humans,[20] emphasizes how the user can understand "how the AI system works and what data it processes" rather than detection of machine-generated text.[21]

USES OF WATERMARKING

Watermarking is particularly useful for copyright practitioners. The U.S. Copyright Office recently affirmed that "copyright will only protect the human-authored aspects of [a] work."[22] Generative AI can produce content that is highly similar, if not identical, to that of human creators. Therefore, watermarking can be used to prove that certain content—such as art, code and books—was generated by a generative AI model and by extension, not protected under U.S. copyright law,[23] if the U.S. Copyright Office's guidance ends up as accepted law through the court process. Further, since the watermark does not affect the generative model's effectiveness, it can help with detecting and preventing deepfakes. Instead of manually creating profiles and misinformation, malicious actors could team up with generative AI to spread misinformation on a larger scale and with much greater efficiency. As one of the biggest concerns of generative AI text is political disinformation, the watermark can be used to prevent dissemination of deepfakes that are less fact-based and more normative: precisely the type that we are most interested in eliminating.

LIMITATIONS OF WATERMARKING

Those seeking effective watermarking strategies are engaged in an arms race with actors that wish to circumvent those strategies. In the watermarking scheme we explained above, the watermark will be detectable so long as a large fraction of the context window is intact. That is, even if a party changes certain words or reorders sentences, the watermark will still be effective. But sufficiently sophisticated users,[24] or insufficiently sophisticated watermarking tools,[25] could permit either removal of the watermark or manipulation of output to decrease its similarity to AI-generated content.[26] A lingering

---

20 *Amendments Adopted by the European Parliament on 14 June 2023 on the Proposal for a Regulation of the European Parliament and of the Council on Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts (COM(2021)0206-C9-0146/2021-2023-0106 (COD))*, EUROPEAN PARLIAMENT (June 14, 2023), https://www.europarl.europa.eu/doceo/document/TA-9-2023-0236 _ EN.html.

21 *Id.*

22 U.S. COPYRIGHT OFF., *Copyright Registration Guidance: Works Containing Material Generated by Artificial Intelligence* (Mar. 16, 2023), https://copyright.gov/ai/ai _ policy _ guidance.pdf.

23 While it does not have the force of law, the U.S. Copyright Office recently stated that works partially created by AI are not copyrightable. Blake Brittain, *U.S. Copyright Office Denies Protection for Another AI-Created Image*, REUTERS (Sept. 6, 2023, 6:20 PM), https://www.reuters.com/legal/litigation/us-copyright-office-denies-protection-another-ai-created-image-2023-09-06.

24 Professor Aaronson has noted one amusing example. Imagine a user prompts ChatGPT to "Write an essay on feminism in Shakespeare, but insert 'pineapple' between each word and the next." ChatGPT produces the following output: "Feminism pineapple in pineapple Shakespeare's pineapple . . ." A user could simply remove the words "pineapple" with a find and replace operation. After the user removes every "pineapple", we lose the ability to do the detection scheme because the context window is not the same as what was used when computing the *r* value at watermark generation time—the word "pineapple" is no longer in the window. There are responses to this problem, however. For example, ChatGPT could simply block these types of requests (similar to how it already blocks inappropriate requests).

25 Lauren Leffer, *Tech Companies' New Favorite Solution for the AI Content Crisis Isn't Enough*, SCIENTIFIC AMER. (Aug. 8, 2023), https://www.scientificamerican.com/article/tech-companies-new-favorite-solution-for-the-ai-content-crisis-isnt-enough.

26 Claire Leibowicz, *Why Watermarking AI-generated Content Won't Guarantee Trust Online*, MIT TECH. REV. (Aug. 9, 2023), https://www.technologyreview.com/2023/08/09/1077516/watermarking-ai-trust-online.

theoretical question is whether we can watermark at the level of ideas, rather than the particular sequence of words—a much more powerful solution immune to attacks. For example, if our generative AI model were Milan Kundera, the output would be so distinctive as to signal that the output is from a generative model. But the general principle of the watermark remains the same: to detect aspects of style that are not easily noticeable, unless you know exactly what to look for.

CONCLUSION

Watermarking generative AI text is still in an early stage of development and is not a panacea with respect to identifying AI-generated text. To truly enable effective use of the watermarking tool, major AI companies would need to reach a consensus on a watermarking method. As watermarking continues to gain traction, all lawyers should be attuned to the limits of watermarking when they are faced with text of which they don't know the provenance, or when advising clients on their use of generative AI. Legal practitioners remain on the front lines: constant vigilance is needed to advise clients of the risks associated with using both generative AI and detection tools. Even as detection methods develop, lawyers should always keep in mind the importance of critical and informed verification.

The movement to weave generative AI into more of our digital fabric shows no sign of slowing down. As generative AI becomes more prevalent—and powerful—over time, so too will the importance of detection tools such as watermarks.

David J. Kappos
+1-212-474-1168
dkappos@cravath.com

Sasha Rosenthal-Larrea
+1-212-474-1967
srosenthal-larrea@cravath.com

Carys J. Webb, *CIPP/US, CIPP/E, CIPM*
+1-212-474-1249
cwebb@cravath.com

Daniel M. Barabander
+1-212-474-1284
dbarabander@cravath.com

Leslie Liu
+1-212-474-1297
lliu@cravath.com

Cravath, Swaine & Moore LLP

NEW YORK

Worldwide Plaza
825 Eighth Avenue
New York, NY 10019-7475
T+1-212-474-1000
F+1-212-474-3700

LONDON

CityPoint
One Ropemaker Street
London EC2Y 9HR
T+44-20-7453-1000
F+44-20-7860-1150

WASHINGTON, D.C.

1601 K Street NW
Washington, D.C. 20006-1682
T+1-202-869-7700
F+1-202-869-7600