
WATERMARKING LARGE LANGUAGE MODELS IN EUROPE: INTERPRETING THE AI ACT IN LIGHT OF TECHNOLOGY

Thomas Souverain*

Postdoctoral Fellow, Department of AI Ethics, CEA Paris-Saclay, France.

* E-mail: thomas.souverain@cea.fr

ABSTRACT

To foster trustworthy Artificial Intelligence (AI) within the European Union, the AI Act requires providers to mark and detect the outputs of their general-purpose models. The Article 50 and Recital 133 call for marking methods that are "sufficiently reliable, interoperable, effective and robust". Yet, the rapidly evolving and heterogeneous landscape of watermarks for Large Language Models (LLMs) makes it difficult to determine how these four standards can be translated into concrete and measurable evaluations. Our paper addresses this challenge, anchoring the normativity of European requirements in the multiplicity of watermarking techniques. Introducing clear and distinct concepts on LLM watermarking, our contribution is threefold:

Watermarking Categorisation: We propose an accessible taxonomy of watermarking methods according to the stage of the LLM lifecycle at which they are applied — before, during, or after training, and during next-token distribution or sampling.

Watermarking Evaluation: We interpret the EU AI Act's requirements by mapping each criterion with state-of-the-art evaluations on robustness and detectability of the watermark, and of quality of the LLM. Since interoperability remains largely untheorised in LLM watermarking research, we propose three normative dimensions to frame its assessment.

Watermarking Comparison: We compare current watermarking methods for LLMs against the operationalised European criteria and show that no approach yet satisfies all four standards. Encouraged by emerging empirical tests, we recommend further research into watermarking directly embedded within the low-level architecture of LLMs.

KEYWORDS

Large Language Models (LLMs), Watermarks, Taxonomy, Europe, Normativity, Operational.

1 INTRODUCTION

Among the most advanced techniques in Artificial Intelligence (AI), Large Language Models (LLMs) can generate text that appears meaningful, though it is not strictly derived from human programming or explicit instructions. The capabilities of LLMs therefore call for watermarking or digital signatures to authenticate synthetic content (Hunter, 1967), to maintain the distinction between AI- and human-authored texts. Beyond copyright infringement, watermarking is crucial to prevent emotional and cognitive confusion, which may result in serious social and psychological harm (Grinbaum & Adomaitis, 2022).

These ethical foundations have been recently nailed into legal obligations. The European Union (EU) has positioned itself as a pioneer in promoting watermarking for generative models. The use of watermarks to identify text outputs produced by LLMs was first proposed in Opinion 7 of the French Committee for Digital Ethics (national pilote d'éthique du numérique, 2023), later discussed at the G7 meeting in Japan (Japan,

2023), and formally included in the AI Act (UE, 2024). In particular, Article 50(2) and Recital 133 of the Act stipulate that LLM outputs should be marked by methods that are “sufficiently reliable, interoperable, effective and robust, as far as this is technically feasible” (UE, 2024). However, neither the Act itself nor the accompanying Code of Practice (Oliver & Bommasani, 2025) provides specific guidance on how these standards relate to existing watermarking techniques for LLMs.

Our paper aims to bridge this gap between legal norms and technical realities by operationalizing the European requirements for LLM watermarking. We clarify how the AI Act’s four criteria apply to synthetic content authentication, grounding our analysis in a precise overview of watermarking methods for LLMs. Section 2 outlines the main types of LLM watermarks, Section 3 reviews existing evaluation practices, and Section 4 offers an operational interpretation of the AI Act criteria. Section 5 develops research foundations for the underexplored criterion of interoperability. Thanks to our operational interpretation of EU criteria, Section 6 finally trades off the advantages and weaknesses of existing LLM watermarks.

To guide the reader from European legal principles to technical feasibility in LLMs, we provide three concise visual summaries of our main results. Figure 1. presents a schematic overview of watermarking styles, situating them within the LLM lifecycle. Figure 2 illustrates how operational interpretations of the AI Act’s requirements for watermarking are derived by resolving conceptual overlaps. Table 2 offers a comparative overview of the advantages and limitations of the main watermarking families, assessed against the operationalized European criteria. Together, these visual aids clarify how LLM watermarking can be implemented and assessed for compliance.

2 STATE OF THE ART IN LLM WATERMARKING

To assess whether the EU AI Act criteria match LLM watermarking techniques, we must present them in a clear manner. Therefore, we aim to be as concise as it may in formalism and details ; for each approach, we only select the most discussed and representative techniques.¹

From the observation that current classifications of watermarking techniques for LLMs either lack of clarity, precision or completeness (Section 2.1), we introduce a simple and original taxonomy. We distinguish between methods that stamp tokens and texts outside LLM computations (Section 2.2), and marks inside the architecture and generation process of the model (Section 2.3).

2.1 SIMPLIFYING EXISTING TAXONOMIES

Market and legal incentives carved out a plethora of LLM watermarking techniques. However Fernandez et al. (2025) point out that emerging taxonomies lack of clarity to define and characterize these tree fern LLM watermarking methods. As the authors observe, even the broad picture of evaluation methods provided by Zhao et al. (2025) interchangeably use terms as “semantic” and “in-processing” methods. Other taxonomies on watermarking for neural networks as Boenisch (2021) are based on watermarks’ properties such as one-bit storage (only indicating if the content was watermarked) or multi-bit format (e.g. with further information on the provider, the generation time), though they do not directly relate these properties with stages of the model’s development. Fernandez et al. (2025) themselves lack of completeness to describe the diversity of in-processing and post-hoc LLM watermarking.

- Liang et al. (2024) fittingly demarcates between watermarks which are embedded “into text” and “into model”. Nonetheless, it is unclear why the modification of the LLM generation process shall be tied to the “text” and not the “model” watermarking. Furthermore, by assigning watermarking “based on cryptography” its own category, the authors obscure the extent to which it is aligned with textual or model-oriented methods.

¹We refer to Liu et al. (2024) and Gloaguen et al. (2025) for more formalism, and to Liang et al. (2024) and Zhao et al. (2025) for more exhaustive lists on papers that share similar approaches.

- Dathathri et al. (2024) separate "retrieval-based approach", "post-hoc detection", and "text watermarking". While the purpose of their paper is to set up a new watermarking method, this introductory taxonomy seems far from obvious. The two first approaches they mention are exclusively dedicated to *detect* watermarks, finding statistical patterns or training specific classifiers to identify LLM-generated passages. Notwithstanding, their third category of "text watermarking" involves detection techniques which can overlap with retrieval and post-hoc methods. This classification reveals a blurring of two dimensions: the stage in the LLM development cycle when the watermark manifests, and the procedural step of watermarking, whether concerning its design or its detection.

- In current LLM watermarking taxonomies, the most convincing we found was Liu et al. (2024). The authors introduce a rare, yet decisive distinction during the generation time between logit generation, and token sampling. We incorporate this distinction into our taxonomy, and also draw inspiration from Gloaguen et al. (2025) to characterize in-model watermarking techniques. Our taxonomy originally combines them, as Liu et al. (2024) incompletely analyze watermarking techniques inside LLMs, vaguely mentioning "triggers", or ignoring watermarking signals into LLM weights ; while Gloaguen et al. (2025) narrowly study them in open source contexts and do not stress on the yet crucial notion of logits.

Having noticed the non-exhaustive defaults of current taxonomies listed above, we propose here a simple and novel taxonomy to distinguish them according to a temporal criterion, that is, *when* they take place in regard to the LLM generation process. As Dathathri et al. (2024) and Fernandez et al. (2025) highlight, the watermarking signal can be added before, during, or after the LLM generates any text.

2.2 PRE- AND POST-PROCESSING APPROACHES: SYNTACTIC AND SEMANTIC

While they differ in their object, being either pre-training data or LLM-generated text, we note that the watermarks taking place before and after LLM training and inference display similar styles.

A first type of techniques is *syntactic*. Wei et al. (2024) offer a random substitution of Unicode characters. This watermarking signature is then embedded in the training data, feeding the LLM. Corresponding to this pre-processing method, character substitution also exists once the model has been trained. For instance, EASYMARK (Sato et al., 2023) applies the same idea to LLM-generated text, where hidden characters are concatenated.

Besides, pre- and post-processing watermarking allows for *semantic* techniques. In that direction, Zhang et al. (2024) propose a subtle method to watermark the generated text through synonym-substitution. Based on the frequency of generated words, the authors generate context-based synonyms, purposed to be more present in the LLM contents.

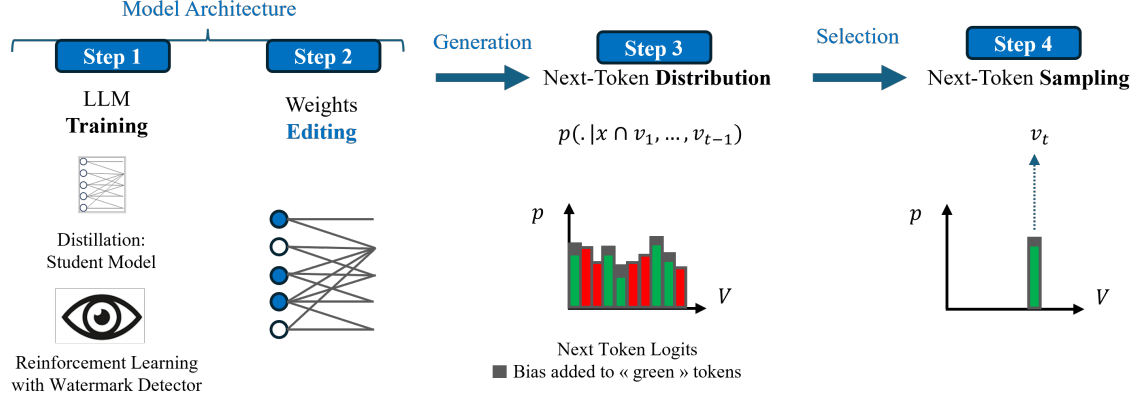
At the intersection of semantic and syntactic approaches, specific classifiers are trained to detect and replace parts of the data fueling LLMs or generated by them. Zhang et al. (2024) train a neural network to focus on punctuation or prepositions and propose discrete substitutions, while Abdelnabi & Fritz (2021) use an encoder / decoder couple to generate substitutions, minimizing their detectability by malicious actors in LLM contents.

2.3 IN-PROCESSING APPROACHES: TRAINING, WEIGHTS, DISTRIBUTION AND SAMPLING

Whereas watermarking in training datasets or already-generated texts is often practical to implement and flexible to test (Liang et al., 2024), inserting watermarking signatures directly into the LLM or its inference process has the potential to withstand quantization, pruning and fine-tuning more effectively Gloaguen et al. (2025) and to be less detectable for attackers Dathathri et al. (2024).

Given a vocabulary of tokens V , we can formalize a Language Model as a function $p_\theta : V^{(t-1)} \rightarrow \Delta(V)$, learning one's parameters θ . p_θ maps the $(t - 1)$ first tokens to a logits' distribution of tokens in $\Delta(V)$. As GPT, Claude, Gemini, Mistral and most LLM iteratively produce their text, each token being sampled conditionally on the others, we will focus here on autoregressive LLMs. That is, given a prompt x and

Figure 1: Watermarking a LLM by altering the Generation Process: Four Steps



Instead of altering the training data or the generated text, watermarking can modify the process of generation itself. We identify four steps of the generation process where an invisible signature can be embedded. Modifications of the architecture and parameters can happen during (step 1) or after training (step 2). It can then slightly bias the distribution (step 3) or selection (step 4) of the next-token generated by the LLM.

the already generated tokens $v_1, \dots, v_{(t-1)} \in V$, the LLM selects the token v_t with the highest conditional probability $p(\cdot | x \cap v_1, \dots, v_{(t-1)})$. In our view, there are four main steps where watermarking can enter the LLM inference process. We present these stages here, synthesised in Figure 1. above:

INTO LLM ARCHITECTURE: STEPS 1 & 2

STEP 1: LLM TRAINING

The first kind of watermarking takes place during LLM training. The watermark directly changes the model's architecture. Some authors have successfully realized it through *distillation* (Gu et al., 2023). The watermarking signal slightly biases a LLM generation (see Step 3 below). This LLM is then used as a teacher to train a LLM student. Hence, the student LLM directly embeds the watermark into its trained weights, with the watermark detector used for distillation being employed to flag watermarked texts.

Other authors bet on *Reinforcement Learning* (RL), as Xu et al. (2024) introducing LLM watermarking into the Reinforcement Learning with Human Feedback (RLHF) pipeline of Ouyang et al. (2022). They jointly train the LLM and a watermark detector, optimizing both the quality of LLM and the detectability of the watermark.

STEP 2: WEIGHTS EDITING

Other avenues of watermarking into the model remain open, once the LLM has been trained. The LLM signature can be inserted at the level of LLM parameters, in some or all layers and versions of the model (Bansal et al., 2022). This weights' modification can take the form of adding bias or noise, which is supposed to robustly and discretely impact the future LLM text.

Zhang & Koushanfar (2024) focus on the most salient parameters, where the gap between minimal and maximal values on their activation layer reaches the highest magnitude (Lin et al., 2024). They add a small bias ξ to these parameters θ on a random basis: $\theta \rightarrow \theta + \xi$. The watermarking stamp displays resistance in

the face of attackers, which do not know the bias factor ξ that is also the key. Building on insights from Li et al. (2023), the authors finally quantize the LLM to make it even harder to attack with the sole int8 version.

Block et al. (2025) also add a slight bias to the parameters θ of the model, which is a key ξ held by providers. Unlike Zhang & Koushanfar (2024), they do not require a quantized version of the LLM. Instead of using activation magnitude, the authors empirically choose a single Multi-Layer Perception (MLP) of a Transformer block before the last activation function to preserve the quality of predictions. Their bias factor follows a normal distribution centered around 0 and with a small variance $\sigma > 0$, which bear the advantage of negligibly affecting the LLM quality. As Gaussians present rotational invariance and independent directions, the bias signal is shown to be orthogonal to LLM weights:

$$\xi \sim \mathcal{N}(0, \sigma^2 I)$$

STEP 3: NEXT-TOKEN DISTRIBUTION

Out of the model training and parameters, there are still steps before the model generates its final token v_t . The autoregressive model generates a distribution of logits, which are then passed into a softmax, computing the probability each token has to occur in position (t) . The prompt x and previous tokens $v_1, \dots, v_{t-1}$ being provided, the probability of any token to come off $p(\cdot | x \cap v_1, \dots, v_{t-1})$ varies as shown in Figure 2.3.

Most watermarking approaches during inference time happen at this moment, a famous one being designed by Kirchenbauer et al. (2023). At each generation of token (t) , the hash of the preceding token $(t-1)$ seeds a random generator, used to partition the vocabulary V into Red and Green tokens. A bias is then added to the logits of Green tokens, enabling them to be selected more frequently than Red ones for the next-token generation (Cf. Figure 2.3, Step 3).

On their side, Aaronson & Kirchner (2022) also alter the next-token distribution with a Gumbel-softmax. This Gumbel rule picks up certain stochastically sampled points, which are the only-one whose logits get converted into probabilities. The authors probe their watermarking bias to be *distortion-free*, keeping intact the quality of the final generated content.

STEP 4: NEXT-TOKEN SAMPLING

From the distribution of tokens' likelihoods to the final token generation, taxonomies often omit the crucial step of next-token sampling. Yet, very different strategies exist to single out v_t : beam search compares the most probable sequences assembling the token v_t with the $(t-1)$ first tokens, while greedy sampling considers a unique sequence and computes a likeliness score for each $v_t \in V$ (Christ et al., 2024; Aaronson & Kirchner, 2022). Therefore, the selection of the next token v_t can be inflected after the distribution of the candidates has been generated. This is the last step where watermarking might arise during inference time, biasing the sampling or selection procedure.

In a recent work, Dathathri et al. (2024) combine the biased logits of a specific subset of tokens (Cf. Step 3) with tournament sampling. At each step of the tournament, the authors allocate the tokens of the vocabulary V over $m \in \mathbb{N}$ chosen colors, according to Kirchenbauer et al. (2023) hashing procedure and the recent context of the prompt x . Their algorithm gives each token a pseudo-random score, which is compared to another token's score. The token with the highest score is selected for the next tournament, until the last remaining top-score token is output as v_t .

3 EVALUATING WATERMARKING IN THE LITERATURE

When comes the time for evaluation, LLM watermarks do not only face up to the diversity of metrics, but also different kind of objectives. For the time being, we put aside the four requirements of the EU AI Act, to be able to better interpret them in Section 4 with technical insights. We choose to focus on three pillars which are frequently adopted in LLM literature to gauge watermarks:²

3.1 ENSURING WATERMARK DETECTABILITY

Being primarily used to authenticate LLM-generated text, watermarks have to bear some recognizability. To that extent, each LLM watermark we presented in Section 2.2 and 2.3 encompasses an integration part, where it is hidden in the model, tokens or text, and a *detection* part. Zero-bit watermarks indicate if the text has or not been LLM-generated, when multi-bits incorporate other information on the copyright or generation date. Therefore, the zero-bit approaches we displayed in previous Sections are computationally lighter and represent the most used watermarks (Liu et al., 2024). For these watermarks, detectors are committed to statistical tests.

Block et al. (2025) implement a test where the null hypothesis corresponds to a negative: the text has not been generated by the LLM put into test. If the null hypothesis is rejected with a sufficiently low p-value, the 5% confidence interval suggests that the tokens are output by the LLM whose weights have been Gaussian-biased. Closely, Kirchenbauer et al. (2023) test the null hypothesis that the text has been produced with no knowledge of the separation rule between Red and Green tokens, based on hash function (t). Their claim to reject is that Green tokens in V have not been overly promoted by the iterative hashing process. The authors compute a z-statistic according to this null hypothesis, and state that the content has been generated by the Red / Green watermarked LLM if the z-value exceeds a chosen threshold.

Beside these evaluations or combined with them, traditional binary classification metrics also serve to detect zero-bits LLM watermarks. For evaluation, they usually compare an equal number of texts written by humans and contents generated by AI. Block et al. (2025) priority is to avoid false positives, e.g. to falsely identify someone as a plagiarist in educative or artistic contexts. We have here a specific instance of a dilemma described for years in Signal Detection Theory, between False Positive Rate (FPR) and False Negative Rates (FNR) McNicol (2005). These detectability metrics also reflect moral choices of the one which implement them, depending on the optimal threshold between them determined through Precision-Recall (PR) or Receiver Operating Characteristic (ROC), and on the priority set by deciders between these errors Souverain et al. (2024). They enclose a sense of acceptability, as measurement errors made by the detectors are considered as a price to pay.

In addition to the precision or recall of detection tools, there are practical challenges to their identifiability. Detectors need enough tokens to be confident in asserting that series of words are the fruit of a model. Among next-token distribution techniques (step 3), Piet et al. (2025) exhibit Kirchenbauer et al. (2023) to require the minimal length of tokens for stable watermark distribution, at a FPR of 2%. A related and tough issue is to enable detection after subtle modifications of the LLM-text, overlooked by the reviews we inspected on LLM watermarking. The concrete example of shrewd cheaters must urgently be considered in education, with small insertions, replacements, deletions or copy-pastes. When does inspiration move to plagiarism, and vice versa? With objective clues as the level of alteration of an initial text, interpretations of detection measures should be aware of a subjective one, linked with the personal intention, honesty and use of the human embedding LLM into one's content.

²For more details, we turn the reader to Liu et al. (2024) p. 15, Zhao et al. (2025) pp. 14-15, and Pan et al. (2024) pp. 5-6. They depict a wide range of optimization tools and metrics to operationalize the evaluation of LLM watermarking.

3.2 ENSURING WATERMARK ROBUSTNESS

Instead of the detectability of watermarks, other authors stress on their *undetectability*. Without the private key, attackers are not supposed to recognize a watermarked content. This change of perspective is relevant in contexts where the model’s owner highlights cybersecurity, copyright and protection of one’s intellectual property (Fernandez et al., 2025; Christ et al., 2024). Maleficent actors claiming they used a specific LLM to produce wrong knowledge or reprehensible beliefs, shall not be able to mark their content with this LLM authentication stamp.

Hence, we see as a second major requirement for watermarks to be robust or resilient to changes. These changes include attacks which are *targeted* against specific LLMs, such as spoofing which may imitate the Red / Green coloring of tokens of Kirchenbauer et al. (2023). Attackers might also append minimal changes in words, unicodes or tokens, in order to use an LLM-content without being flagged. This kind of attack is not watermark-specific, or *untargeted* (Liu et al., 2024).

These two risks stand at the exact symmetric, focusing either on taking possession of the watermark or the LLM (see our schema in Table 1). On one side, spiteful agents mimic a targeted LLM watermark to *wrongly attribute* an illegitimate output to this. They might discard the developer while spreading fraudulent or antisemitic contents allegedly generated by its LLM. The first side *steals the watermark*. On the other side, plagiarists *tracelessly steal* the LLM output. They benefit from the windfall effect of fast and accessible generated text, as described above in education, while erasing any watermarking sign. The second side *steals the LLM output*. While the first resulting defense stresses on protecting the LLM author and integrity (Zhao et al., 2025), that must not be imitated as shown in Table 1, the second one aims at preserving its signature that shall not be easily erased.

Table 1: Defending the LLM: Two Criteria to make the Watermark Robust

Watermark Robustness Criteria	Target of the Attacker	Malicious Attacking Use
Non-Extractable	Watermarking method	Attackers can insert their content, pretending to be generated by the defamed LLM
Non-Erasable	LLM-generated text	Attackers can insert LLM-generated text, pretending to be their own

In the spirit of attacks seeking to steal the LLM while wiping out the watermark, the resilience of watermarks must be proved against operations which take place during the LLM lifecycle. Gloaguen et al. (2025) and Wu et al. (2023) advocate for LLMs to remain detectable once quantization, pruning, weights’ merging and fine-tuning has been applied to the originally watermarked LLM. This injunction is crucial in open-source contexts: once the model has been trained and watermarked, further alterations of its weights or architecture shall keep the watermarking signal trackable.³ The watermark detector must remain functional, after intentional or non-intentional modifications on LLM structure and outputs.

3.3 MAINTAINING LLM QUALITY

Even if the LLM has been shown to be detectable and robust against external changes, it won’t be used if the watermark heavily affects the quality of LLM-generated contents.

³We see these further operations in open-source contexts as equivalent to fine-tuning or distillation attacks described in Wu et al. (2023); Liu et al. (2024) Being described as attacks or natural LLM steps, they follow the same direction: preserving the watermarking signal despite further steps of LLM development.

In that respect, *comparative* tools exist to measure the difference of quality between unwatermarked and watermarked outputs for the same LLM. Meteor or BLEU Scores primarily used in translations, are usually deployed to penalize the watermarked LLMs when their answers to prompts differ in precision or length from the non-watermarked output (Papineni et al., 2002). As such metrics are sensitive to word order and the initial structure of the text, they can be completed with *semantic* evaluation tools. For instance, Yoo et al. (2023) have implemented Sentence-BERT (Reimers & Gurevych, 2019) to compute cosine similarity between semantic embeddings. Introducing an Entailment Score, the authors try to further capture if logical relationships are preserved in the watermarked answer.

Single-text evaluation metrics also exist to assess the quality of watermarked texts. They do not merely differ from the classic quality evaluation of trained LLMs, using perplexity or human scoring (Yoo et al., 2023). On isolated tasks, the watermarked LLMs are evaluated with standard evaluation methods for LLMs: e.g. BLEU and BERTScore for translation and text summarization (Papineni et al., 2002), perplexity or logarithmic diversity for text completion, GPT-Truth and GPT-Info for question-answering and fact-checking (Chen et al., 2023), or LLM-as-a-judge for instruction following (Zhao et al., 2023).

While these metrics classically evaluate the quality of LLM outputs (i) compared to the non-watermarked LLMs and (ii) for specific tasks, some authors also put emphasis on (iii) diversity of the generated answers. In-Processing watermarking methods in Step 3 and Step 4 rely on biased selection of some tokens (Cf. Section 2.3), which can reduce the lexical and semantic diversity of LLM texts. Kirchenbauer et al. (2023) themselves note this default of their token-coloring approach. They propose both a solution, implementing Red / Green tokening only for high-entropy tokens, to introduce distortion for words where the diversity of synonym expressions will not compromise the text quality, and a logarithmic measure of the diversity produced by an LLM (Kirchenbauer et al., 2024).

Other metrics evaluate the loss of diversity produced by watermarking, such as n-gram repetition of tokens. With this gauge, Gu et al. (2023) compared two watermarks happening at the same step of LLM construction, through biasing the logits' distributions. (see Step 3 of Section 2.3). They found that Kirchenbauer et al. (2023) method was of less impact for the content's diversity, compared with the Gumbel approach of Aaronson & Kirchner (2022).

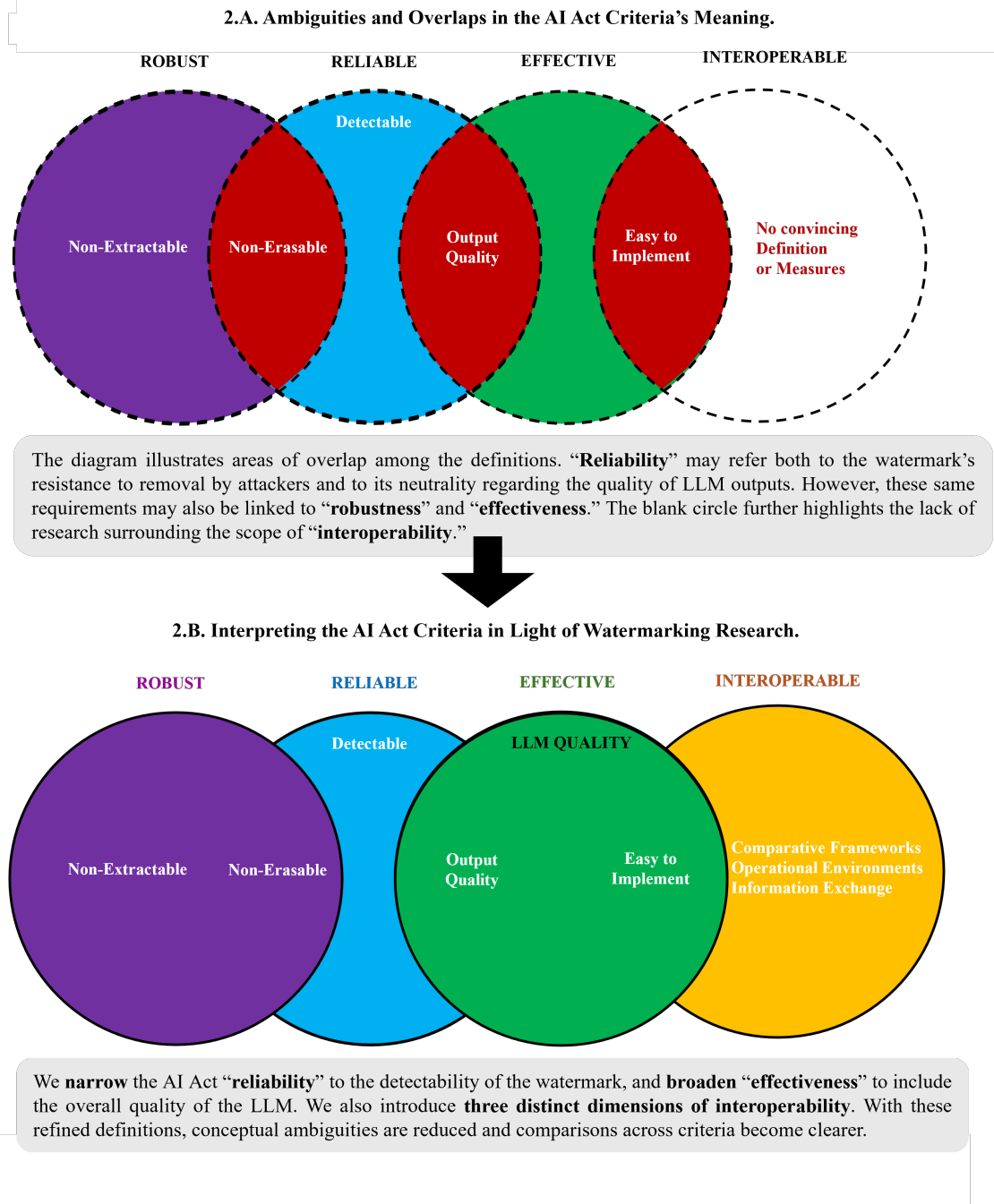
4 WATERMARKING IN EUROPE: OPERATIONAL INTERPRETATION OF THE AI ACT

Now we presented the technical ways LLM watermarks are currently evaluated, how can we relate them with the four criteria defined in the EU AI Act⁴: reliability, interoperability, effectiveness and robustness?

Depending on how we define these four EU claims, one may observe overlaps between them. This is a necessary consequence of their elusive mention in Recital 133, which is also not made explicit in the recent transparency chapter of the Code of Practice for the providers "General-Purpose AI" - including LLMs (Oliver & Bommasani, 2025). We pictured the overlaps in Figure 2.A, which may lead LLM providers and evaluators to unpleasant confusion when assessing AI conformity to the AI Act. To fill the gap, the following Section provides *clear and operational interpretations* of the EU evaluation pillars with state-of-the-art, reachable, and distinct objectives, outlined in Figure 2.B. Our refinement links the AI Act with the context of LLM watermarks, enabling clear guidance of LLM providers.

⁴Cf. AI Act UE (2024), Article 50, Recital 133, and our Introduction 1 for more context.

Figure 2: Interpreting the AI Act Criteria for LLM Watermarking: From Overlaps to Measurable Standards



4.1 INTERPRETING "RELIABILITY" AS DETECTABILITY

"Robustness" is the only European criterion we find as such in watermarking literature, if the AI Act denotes robustness of the watermark *under attacks* - as it is in most evaluation research (Liu et al., 2024; Pan et al., 2024). It is comprised of the two symmetric attacks we analyzed in Section 3.2, Table 1. The watermark must not be *extracted* by malicious agents wanting e.g. to attribute shameful contents to institutional chatbots ; it must not be *erased* by dishonest people, claiming they wrote a piece of text that is in fact fully synthetic.

Lappings arise with "reliability". What is, indeed, reliable: the watermarking method, or the LLM that embeds the watermark? The Recital 133, brief on the matter, does not elucidate it. Let us consider that reliability applies primarily to the watermark technique, which makes sense in the enumeration of the AI Act: *the technique shall be robust, the technique shall be reliable, effective...* An obvious sense is that the technique can be *reliably detected*, reaching sufficient confidence intervals, p-values or z-scores indicated in Section 3.1.

However, one could also mean that the watermark is reliable if its signal passes through possible transformations along the LLM lifecycle - training, deployment, fine-tuning, Retrieval Augmented Generation (RAG) until the final generation of tokens (Gloaguen et al., 2025). This reliability along LLM changes *overlaps with a robustness dimension*, which is, the watermark signal must not be erased.⁵

Therefore, we suggest to avoid confusion in the EU criteria and *to interpret "reliability" as detectability of the watermark*. Using the most restrictive sense as possible to interpret "reliability", that is focusing only on statistical detectability of the signal (tests on various architectures, lengths and combinations of tokens, relevance of the corpus...) constitutes a piece of work which deserves its own category.⁶

4.2 INTERPRETING "EFFECTIVENESS" AS LLM QUALITY

The overlaps continue with "effectiveness". A technique is considered effective when it achieves the goals intended by its designers. It *effectively* does what it was programmed for. As we observed in Section 3, these goals are threefold: robustness, detectability, and LLM quality. To maintain distinct AI Act criteria, we already treated robustness and detectability as separate categories. What remains is LLM quality as the defining aspect of effectiveness.

As noted in Section 4.1, reliability was ambiguous: it could refer either to trustworthy watermarking or to faithful model outputs.⁷ Defining LLM quality as an independent criterion helps resolve this ambiguity between "reliability" and "effectiveness". Reliability now evaluates the *tool*, while quality concerns the *outputs*.

Finally, if effectiveness is understood as keeping the LLM relevant for users, quality must also account for implementation factors. Beyond the perceived quality of outputs, ease and cost of implementation are relevant. This extended reading of "effective" watermarking thus combines qualitative outputs and implementable LLM under the unified notion of LLM quality.⁸

⁵See Figure 2.A, left red intersection.

⁶See the clarified concept of detectability in Figure 2.B, blue.

⁷See Figure 2.A, red, right overlap between reliability and effectiveness.

⁸See the clarified concept of LLM quality in Figure 2.B, green.

5 INTERPRETING "INTEROPERABILITY": THREE RESEARCH DIRECTIONS

Interoperability warrants a separate discussion, as its technical foundations remain unclear and largely unexplored in current LLM research. Literally, watermarks are interoperable if they can *work together* to authenticate AI-generated contents.⁹ We highlighted above three working directions in which marking techniques have then to converge: keeping the signal robust, and detectable, without deteriorating the LLM quality. Besides, operability points towards operationalization: watermarks must also be easily embedded in different forms of LLMs, and comparable with other authentication methods.

We could imagine a global detector for all LLMs registered on the European market with their compulsory watermarks, made accessible to public deciders. Such *objective* instruments would help EU auditors to check for LLM compliance. This ideal global platform would also complement the *subjective* intuition of teachers. Interoperable watermarks would be detectable regardless of the marking style, the original provider (Google, OpenAI, Anthropic...), and the potential transformations of an LLM (RAG, fine-tuning, mixed architectures...). For a teacher, it would indicate above reasonable thresholds that the dissertation of any suspected student was LLM-generated.

However, except emerging comparisons of LLM watermarks narrowed to *LLM quality* across different models (Dang et al., 2025; Gloaguen et al., 2025), existing literature leaves an unexplored area within the field of interoperability. The concept is not yet installed in LLM watermarking comparison. To bridge the gap, we sketch below three dimensions in which LLM research should urgently work to give birth to global evaluations on interoperability:

5.1 COMPARATIVE FRAMEWORKS

Evaluating *every technique*, including fingerprints and logging mentioned in the same Recital 133 (UE, 2024), must be covered by interoperability. To combine them adequately, we have to compare all the methods authenticating LLM contents across robustness and detectability of the signal, and LLM quality. Open-source toolkits already exist to compare *some styles of watermarks* within each other on these three main requirements. For instance, MarkLLM (Pan et al., 2024) regularly actualizes a GitHub repository to implement and compare Next-Token Distribution based on the coloring of tokens (Kirchenbauer et al., 2023) (see Section 2.3) and Next-Token Sampling inspired from cryptography (Christ et al., 2024) (see Section 2.3). Whereas the MarkLLM benchmark compares methods from different styles and with replicable code, more general initiatives shall be led to evaluate marking techniques in a complete manner.

5.2 OPERATIONAL ENVIRONMENTS

LLM Watermarking techniques do not only have to prove their detectability, robustness and least impact on LLM quality compared with other authentication tools. They must also achieve the three goals *once LLM are deployed*, and used on a daily basis. In this perspective, the method of DeepMind, inspired from Kirchenbauer et al. (2023) coloring to pseudo-randomly sample the tokens (see Section 2.3, Dathathri et al. (2024)), was tested with 20 million Gemini responses. We do not however consider it as sufficient to prove that a LLM watermark is ready for large-scale production. The authors should also extend their experiments beyond a fixed 5% False Positive Rate (Block et al., 2025), testing detectability and robustness under conditions such as quantisation, pruning, merging, and fine-tuning, common in open-source LLMs (Gloaguen et al., 2025). Finally, the method has recently shown to be vulnerable to textual attacks on semantics, as paraphrasing and back-translation (Han et al., 2025). Even with this rare example of large-scale testing in LLM watermarking, we call for more comprehensive and realistic frameworks that support integration at scale and in real-world digital environments.

⁹Cf. Cambridge English Dictionary, ed. 2025, consulted on 2025/10/27, "Interoperable".

5.3 INFORMATION EXCHANGE

Lastly, interoperability may compare authentication methods *in the way they share information*. Just as detectives using different methods share clues, fingerprinting and watermarking techniques may prove complementary in ensuring the authenticity of LLM outputs. Taking this further, the Coalition for Content Provenance and Authenticity (C2PA), founded in 2021 by a consortium of technological companies, introduced an international specification which lists robust ways of signing a document and attributing its content for these industrials, *mainly coming from cryptography* (C2PA, 2025). Even if these "hard binding" metadata prove the authenticity of the content, often with a unique hashing signature, metadata might be lost after compressions or conversions, and complex to recover. That is why the C2PA conceptualized "soft binding", like *watermark* or *fingerprint*, whose embedded and invisible mark would help to ensure the authenticity of LLM contents.¹⁰ C2PA pioneer standards stress that marking methods mentioned by the EU AI Act (UE, 2024) *are* complementary. Further research should explore *how* these methods can work together to improve output attribution - for instance, by jointly narrowing the confidence interval.

5.4 FOUNDATIONS FOR FUTURE WORKS ON INTEROPERABILITY

Probably because LLM watermarking techniques mostly lie at the stage of advanced research, interoperability suffers from the absence of discussions on industrialized articulation of techniques, measurements, and even definitions. In that sense, the three dimensions we suggested above mark a progress to clarify the EU AI Act requirement of interoperability.¹¹

The field has to stay tuned with rapidly evolving LLMs and associated environments, and implementation aspects will certainly lead to more solid operational assessments in future months and years. However, experimental measurements of the three dimensions of interoperability remain rare and only partially explored: they fall outside the scope of current LLM watermarking benchmarks (Kirchenbauer et al., 2024; Liu et al., 2024; Zhao et al., 2025). Hence, interoperability cannot yet be integrated into the evaluation presented in Section 6 and Table 2. Doing so would not be supported by sufficient empirical evidence.

6 TRADE-OFFS FOR EXISTING LLM WATERMARKING TECHNIQUES

We interpreted the EU criteria in a rigorous and measurable way. How can we connect them with the state-of-the-art techniques and evaluations introduced above? This last section is dedicated to comparing the LLM watermarking methods along the EU evaluation axes. As the final Table 2 summarizes, each style of watermarking bears its own advantages and defaults. In particular, every watermarking family presents to LLM providers its own trade-offs between robustness, detectability, and effectiveness.

6.1 PRE- AND POST-PROCESSING: SLOW DETECTION OR EASY REMOVAL

In Section 2.2, we mentioned a set of methods interested in training data and embeddings (pre-processing) or already generated text (post-processing). These techniques enforce subtle additions, deletions, or substitutions into textual elements, being or not vectorized. These schemes do not meddle with the inner model, in contrast to in-processing methods which involve specific training (Xu et al., 2024), modifications in tokens (Aaronson & Kirchner, 2022; Kirchenbauer et al., 2023) or architectural changes into LLM layers (Bansal et al., 2022; Block et al., 2025). To the extent they do not affect the generation process, most pre- and post-processing approaches preserve the *global quality* of the unwatermarked LLM outputs (Liang et al., 2024). Besides, introducing hidden characters (Wei et al., 2024; Sato et al., 2023) or synonyms (Zhang et al., 2024) allows for understandable substitution and straightforward detection.

¹⁰See C2PA (2025), 9.3, "Soft Bindings".

¹¹See the Figure 2.B, with the yellow circle of "interoperable" LLM watermarks filled with our three dimensions of global, at-scale, and information-sharing comparison.

Table 2: The strengths and weaknesses of LLM watermarking styles. In green cells are criteria met with high confidence, on the contrary to red. Yellow cells suggest a medium maturity of the technique.

Evaluation Criteria *****	ROBUST Watermark		DETECTABLE Watermark	EFFECTIVE LLM	
	Non- Erasable	Non- Extractable	Detectable	Easy to Implement	Output Quality
Pre- & Post-processing					
Direct replacement of words or characters	Red	Red	Green	Green	Green
Selective replacement (random, contextual...)	Green	Red	Green	Red	Green
Next-Token Distribution					
Small Window Size	Green	Red	Green	Green	Red
Large Window Size	Red	Green	Yellow	Yellow	Green
Next-Token Sampling					
Adding bias to specific blocks of tokens	Yellow	Red	Green	Red	Green
Mapping random numbers to LLM samples	Green	Yellow	Yellow	Yellow	Yellow
Into LLM Architecture					
Distillation of a watermark teacher model	Green	Red	Green	Yellow	Green
RL with a watermark detector	Green	Yellow	Yellow	Yellow	Green
Adding Bias on specific LLM weights	Green	Yellow	Green	Green	Green

In spite of that, clear-cut characters’ and synonyms’ insertions are vulnerable to simple attacks like random synonym substitution. Incurious actors may simply query LLM outputs through Application Programming Interfaces (APIs), replacing or erasing the words and Unicodes put forward during watermarking (Liu et al., 2024). Watermarking can be spoofed in pre-processing methods (Wei et al., 2024), where the signal embedded in the training data cannot be changed without retraining the LLM.

Yet, synonym substitutions which are pseudo-random (Wei et al., 2024) or sensitive to the context (Zhang et al., 2024) show more resistance to textual attacks. This is also the case for methods entwining syntactic and semantic modifications, such as Abdelnabi & Fritz (2021) using adversarial training to introduce substitutions, which become especially hard to be disclosed by attackers.

In all of these cases, however, the *robustness to textual attacks* arises at the expense of *heavier implementation*. Involving a rejection sampling algorithm to position synonyms into k semantic spaces, k -SEMSTAMP winds down the generation time of the text. The detectors in Zhang et al. (2024) and Abdelnabi & Fritz (2021) need a specific classification training, which is also computationally costly. As the resistance to learnability (see Section 3.2) of such watermarks shall be studied over simple textual attacks, this first, superficial layer of watermarking robustness compromises the quality of LLMs.

6.2 IN-PROCESSING APPROACHES

In contrast, in-processing methods mark LLMs with a non-superficial stamp, as they embed their watermark during the generation time of LLMs. As shown in this section and summarised in the last three categories of Table 2, in-processing watermarking strikes a balance between strong resistance to the signature being extracted or erased and increased computational load on the LLM. This additional burden can sometimes lead to a loss in output quality.

NEXT-TOKEN DISTRIBUTION (STEP 3): EXTRACTABLE OR ERASABLE

In that manner, printing the next-token distribution lies in the foreground of LLM watermarking. Happening after the training of the LLM, such approaches do not alter the inner architecture but the logits output by the model. They are detectable to the extent they add a slight bias to certain tokens, which is also the model’s signature ξ . Intuitively, a trade-off arises: the *detectability* of ξ among tokens might hit resistance to imitation or plagiarism attacks (less *robustness*).

A key setting is the window size,¹² i.e. the number of preceding tokens which are used to generate the selection of biased tokens at each generation step (t). A large window size makes Kirchenbauer et al. (2023) and Aaronson & Kirchner (2022) signatures more difficult to steal, as the dependencies between tokens become more complex. However, any targeted textual attack as rewording gets more chances to break these dependencies, erasing the watermark. The reverse is true for small window sizes (Liu et al., 2024): the same distribution-based techniques become easier to retrieve or learn from a sufficient amount of watermark-generated text (Gu et al., 2023), but they gain resistance to paraphrasing as their coloring and separation rules depend on fewer preceding tokens.

Our third criterion of LLM quality joins the trade-off, as larger window sizes enhance the diversity of generated texts (Zhao et al., 2025). Kirchenbauer et al. (2023) focus on soft tokens, with high entropy, is also shown to be less compromising for the diversity and human satisfaction on LLM contents, at the cost of more complicated procedures to detect the watermark (Kirchenbauer et al., 2024).

¹²See Table 2 for an picture of the advantages borne by large or small window size in Next-Token Distribution watermarking.

NEXT-TOKEN SAMPLING (STEP 4): BALANCING LLM QUALITY AND UNDETECTABILITY

Like logits’ techniques, next-token sampling watermarking introduces a bias into the generation process. It guides the selection of the next token v_t , associating pseudo-random numbers to privilege some tokens (Christ et al., 2024; Aaronson & Kirchner, 2022; Kuditipudi et al., 2023) or semantic embedding spaces (Hou et al., 2024).

Though understudied, sampling watermarking shows divergent strengths and weaknesses depending on the methods. Inspired from cryptography, Christ et al. (2024) progressively sample tokens in one group and, once the block gains enough entropy, associate a unique hashing signature to the block. This approach stresses on *undetectability* or robustness to unallowed imitation, while maintaining a high *diversity* of produced contents on the quality side. However, this two-bits sampling process is computationally heavy to implement in real LLMs and less robust to spoofing or textual replacement attacks. k-SemStamp (Hou et al., 2024) whose biased sampling relies on similar semantic spaces, embedding regions of tokens through k-means, presents analogous advantages and defaults. While this method ensures the quality of LLM answers, it is also costly to implement and sensitive to textual attacks in targeted semantic regions.

On the contrary, Aaronson & Kirchner (2022) and Kuditipudi et al. (2023) samplings can be deployed in LLMs out of toy-examples. Against removal attacks, Kuditipudi et al. (2023) proposed to enhance the token-sampling of Aaronson & Kirchner (2022) by extending the pseudo-random number sequence over the text length. However, both of these token-based sampling might easily be learned by an attacker model from watermarked-generated data, especially in low-entropy scenarios (Gu et al., 2023). Additionally, if they overly boost some tokens or words, they can compromise the text’s *quality* and *diversity*.

WATERMARKING IN MODEL ARCHITECTURE (STEPS 1 & 2): TOWARDS EXTRACTABILITY IN OPEN-LLM CONTEXTS

Research on LLM watermarking has so far focused mainly on token-level and text-level methods, whether syntactic or semantic. Approaches that directly modify LLM weights or architectures remain relatively rare. Though, they represent a highly promising direction, that needs to be explored through more development and testing:

High quality, embedded detectability and resistance to most targeted attacks.

So far, the advantages and limitations of such structural approaches have received little systematic attention. For instance, distillation is absent from the survey by Liang et al. (2024), and mentioned only as an attack strategy in Liu et al. (2024). Nonetheless, these methods exhibit several noteworthy advantages. Most of these in-processing techniques integrate the watermark *into the LLM parameters*. In Xu et al. (2024), the watermark detector is a reward model jointly trained with the LLM. Therefore, the objective of detectability is directly integrated in the trained LLM weights. Bansal et al. (2022) or Block et al. (2025) bias ξ is added to the parameters θ after training, but this noise modifies only some Transformer blocks, layers, and non-linear functions. Hence, both of these watermerkings into weights do affect the quality of predictions in a reduced manner, compared with data modification (pre- and post-processing) and the biasing of next-token selection (see Section 2.3, Steps 3 and 4 of In-Processing Watermarking).

In contrast, the third approach we mentioned into LLM architecture (Step 1), watermark distillation, is more prone to degrading prediction quality. In Gu et al. (2023), the teacher model is a LLM that integrates the watermark either during next-token distribution (Kirchenbauer et al., 2023; Aaronson & Kirchner, 2022) or during sampling (Kuditipudi et al., 2023). The decoder that detects the over-coloured or over-sampled tokens is then directly learnt by the student LLM. Thus, the student model inherits from the potential decrease in quality of the watermarked parents, depending on their window sizes (Section 6.2) or repetitive sampling (Section 6.2, Gu et al. (2023)).

At the same time, all three approaches integrate their watermarks into the *structure* of models, which makes them particularly hard to remove from text or tokens. Distilled watermarks of Kirchenbauer et al. (2023)

or Kuditipudi et al. (2023) are more resistant to textual attacks as paraphrasing than the initial token-watermarking methods (Gu et al., 2023). Other watermark techniques into LLM architecture also offer to be accurately detectable, thanks to the trained detector model (Xu et al., 2024) or a simple back-propagation into the layer where the Gaussian weights are stored (Block et al., 2025), provided the key ξ and prompt x reject the null hypothesis (see Section 3.1).

Research avenues on final detectability and distortion of LLM outputs.

Though these structural watermarks are supposed to make the watermark’s detection easier, as the LLMs are specifically trained or modified for that purpose, their ultimate detectability is not guaranteed. The main challenge is to remain detectable *after further LLM modifications*.

These modifications might have different origins, occurring during the model’s development or being due to spiteful acts (see Section 3.2). Gloaguen et al. (2025) rightly point that in open-source contexts, the trained LLMs are subject to further refitting. The authors lead a battery of tests, and confirm the overall resistance of in-model watermarking methods to pruning and quantization. Their resistance fits with the role of pruning and quantization, designed to lighten the original model without changing the quality of its completions.

However, distilled token-distribution and token-sampling watermarks become harder to distinguish after parameters’ merging and fine-tuning (Gloaguen et al., 2025). Block et al. (2025) and Xu et al. (2024) watermarks are also sensitive to merging and fine-tuning modifications, except when fine-tuning happens on specific data as mathematical instructions.

We deplore that Gloaguen et al. (2025) only measure the detectability through two metrics, namely perplexity and TPR *for a fixed FPR of 5%*. Together with the level of *distortion* brought in final predictions by in-processing biases (Block et al., 2025), these tests should be extended for various ranges of FPR, and for broader detectability criteria (see Section 3.1 concerning the essentials). The deepness of such in-processing watermarks, into weights and layers, leads to an open issue. If the watermarking is too deeply entangled with the LLM architecture, will it be realistic to reliably detect it in the final LLM text?

We call for further research on this promising avenue, which bears the advantage of low quality impact on text quality and difficult removal of the signal at the weights’ level. Tests on detectability and broader integration on computing infrastructures, meant in interoperability, will help to verify if embedding watermarks directly into LLM architecture ensures to reach the three pillars.

7 CONCLUSION: LIMITATIONS AND FURTHER DIRECTIONS

Having anchored the EU AI Act requirements in concrete watermarking techniques for LLMs, we proposed an interpretation that is both clear and technically accurate. We refined the four pillars of robustness, reliability, effectiveness, and interoperability, legally requested in Article 50 and Recital 133 for generative AI marking. Thanks to a precise interpretation of “reliability”, through the lens of detectability of the *watermark*, and a broad interpretation of “effectiveness”, regarding the impact of the watermark on the *LLM*, we propose operational standards to avoid confusion and support guidance of LLM providers.

An accessible evaluation of watermarking techniques as in Table 2 is crucial to enable EU governance. Yet, we encourage to develop this table in two directions:

- We purposed to give a panorama of watermarking *styles*. To that end, we grouped techniques that are heterogeneous on some points. For instance, inside methods which both use pseudo-random sampling (step 4), Aaronson & Kirchner (2022) hurts more the text diversity than Dathathri et al. (2024), whose hashing procedure turns out to be more sensitive to paraphrase attacks (Han et al., 2025). To enable this kind of nuances to appear, a solution would be to apply the same categories than in Table 2, not only for watermarking styles, but also for pairwise comparisons of watermarks.
- The fourth AI Act pillar of interoperability must be evaluated in our comparative table. However, due to the lack of empirical evidence, we were not able to include interoperability evaluations of

watermarking families. LLM research must urgently provide complete comparisons of all marking methods in realistic environments, towards the directions we pointed in Section 5. Assessing interoperability will help to confirm the robustness of techniques emerging inside LLM architecture, involving training and weights’ adjusting watermarking (steps 1 & 2).

Not only should watermarking techniques be comparable, but so too should the international expectations surrounding them. While the American Executive Order promoting the labelling of synthetic content was rescinded in early 2025 (USA, 2025), global standards for both visible and invisible watermarking of synthetic texts remain under development (C2PA, 2025). Similarly, new Chinese industrial standards entered into force in September 2025, underscoring the growing challenge of watermarking LLM-generated texts, where discrete symbols cannot embed imperceptible marks as easily as continuous images (Evennou et al., 2024). Indeed, the Cyberspace Administration of China now mandates “implicit labelling” for synthetic images and videos, while merely encouraging it for generated text (China, 2025).

In setting standards for LLM watermarking, it is therefore time for Europe to set out its own framework. Confronted with these emerging global initiatives, the European Union will require robust governance mechanisms to enforce, audit, and certify LLM watermarking practices (Nemecek et al., 2025). Achieving this will first demand a clear operational understanding of what LLM watermarking evaluation entails.

We hope that our interpretation of the European requirements can help shape this understanding, paving the way for future research and empirical assessments of conformity with the AI Act—thus supporting providers’ compliance, guiding Commission auditors, and reinforcing citizens’ trust in AI.

ACKNOWLEDGMENTS

Thomas Souverain conducted this research under the the OpenLLM project funded by France-2030. The author is grateful for the precious insights of Dr. Alexei Grinbaum.

COMPETING INTERESTS

The author declares no competing interests.

REFERENCES

- Scott Aaronson and Hendrik Kirchner. Watermarking gpt outputs. *Retrieved September, 5:2024, 2022*. URL <https://www.scottaaronson.com/talks/watermark.ppt>.
- Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 121–140. IEEE, 2021.
- Arpit Bansal, Ping-yeh Chiang, Michael J Curry, Rajiv Jain, Curtis Wigington, Varun Manjunatha, John P Dickerson, and Tom Goldstein. Certified neural network watermarks with randomized smoothing. In *International Conference on Machine Learning*, pp. 1450–1465. PMLR, 2022.
- Adam Block, Ayush Sekhari, and Alexander Rakhlin. Gaussmark: A practical approach for structural watermarking of language models, 2025. URL <https://arxiv.org/abs/2501.13941>.
- Franziska Boenisch. A systematic review on model watermarking for neural networks. *Frontiers in Big Data*, 4, November 2021. ISSN 2624-909X. doi: 10.3389/fdata.2021.729663. URL <http://dx.doi.org/10.3389/fdata.2021.729663>.
- C2PA. C2pa technical specification, 2025. URL https://spec.c2pa.org/specifications/specifications/2.1/specs/C2PA_Specification.html.

-
- Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. Watme: Towards lossless watermarking through lexical redundancy. *arXiv preprint arXiv:2311.09832*, 2023.
- China. Notice on Issuing the Measures for Labeling of AI-Generated Synthetic Content , howpublished = https://www.cac.gov.cn/2025-03/14/c_1743654684782215.htm, author=Cyberespace Administration of China, date = 2025-03-14, note = Accessed: 2025-07-02, 2025.
- Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024.
- Kieu Dang, Phung Lai, NhatHai Phan, Yelong Shen, Ruoming Jin, Abdallah Khreishah, and My Thai. Sok: Are watermarks in llms ready for deployment? *arXiv preprint arXiv:2506.05594*, 2025.
- Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.
- Gautier Evennou, Vivien Chappelier, Ewa Kijak, and Teddy Furon. Swift: Semantic watermarking for image forgery thwarting. In *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6. IEEE, 2024.
- Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Tomas Soucek, Valeriu Lacatusu, Tuan Tran, and Alexandre Mourachko. A taxonomy of watermarking methods for ai-generated content. In *The 1st Workshop on GenAI Watermarking*, 2025.
- Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Towards watermarking of open-source llms. *arXiv preprint arXiv:2502.10525*, 2025.
- Alexei Grinbaum and Laurynas Adomaitis. The ethical need for watermarks in machine-generated language. *arXiv preprint arXiv:2209.03118*, 2022.
- Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. *arXiv preprint arXiv:2312.04469*, 2023.
- Xia Han, Qi Li, Jianbing Ni, and Mohammad Zulkernine. Robustness assessment and enhancement of text watermarking for google’s synthid. *arXiv preprint arXiv:2508.20228*, 2025.
- Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. *arXiv preprint arXiv:2402.11399*, 2024.
- Dard Hunter. *Hand Made Paper and Its Water Marks*. Franklin, 1967.
- G7 Presidency Japan. G7 hiroshima process on generative artificial intelligence (ai), 2023. URL https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/09/g7-hiroshima-process-on-generative-artificial-intelligence-ai_8d19e746/bf3c0c60-en.pdf.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/kirchenbauer23a.html>.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=DEJIDCmWOz>.

-
- Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.
- Linyang Li, Botian Jiang, Pengyu Wang, Ke Ren, Hang Yan, and Xipeng Qiu. Watermarking llms with weight quantization. *arXiv preprint arXiv:2310.11237*, 2023.
- Yuqing Liang, Jiancheng Xiao, Wensheng Gan, and Philip S Yu. Watermarking techniques for large language models: A survey. *arXiv preprint arXiv:2409.00089*, 2024.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024.
- Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.
- Don McNicol. *A primer of signal detection theory*. Psychology Press, 2005.
- Comité national pilote d’éthique du numérique. Systèmes d’intelligence artificielle générative : enjeux d’éthique, avis 7., 2023. URL <https://cea.hal.science/cea-04153216/document>.
- Alexander Nemecek, Yuzhou Jiang, and Erman Ayday. Watermarking without standards is not ai governance. *arXiv preprint arXiv:2505.23814*, 2025.
- Nuria Oliver and Rishi Bommasani. Code of practice for general-purpose ai models. transparency chapter. *European Commission Digital Strategy, AI Office Document.*, 2025. URL <https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, et al. Markllm: An open-source toolkit for llm watermarking. *arXiv preprint arXiv:2405.10051*, 2024.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.
- Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. Markmywords: Analyzing and evaluating language model watermarks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 68–91. IEEE, 2025.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
- Ryoma Sato, Yuki Takezawa, Han Bao, Kenta Niwa, and Makoto Yamada. Embarrassingly simple text watermarks. *arXiv preprint arXiv:2310.08920*, 2023.
- Thomas Souverain, Johnathan Nguyen, Nicolas Meric, and Paul Égré. Implementing fairness in ai classification: The role of explainability. *arXiv preprint arXiv:2407.14766*, 2024.

UE. Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act). *Regulation (EU) 2024/1689*, 2024. URL <https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng>.

USA. White House Releases New Policies on Federal Agency AI Use and Procurement, howpublished = <https://www.whitehouse.gov/articles/2025/04/white-house-releases-new-policies-on-federal-agency-ai-use-and-procurement/>, date = 2025-04-07, note = Accessed: 2025-07-02,, 2025.

Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. Proving membership in llm pretraining data via data watermarks. *arXiv preprint arXiv:2402.10892*, 2024.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. *arXiv preprint arXiv:2310.07710*, 2023.

Xiaojun Xu, Yuanshun Yao, and Yang Liu. Learning to watermark llm-generated text via reinforcement learning. *arXiv preprint arXiv:2403.10553*, 2024.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust multi-bit natural language watermarking through invariant features. *arXiv preprint arXiv:2305.01904*, 2023.

Ruisi Zhang and Farinaz Koushanfar. Emmark: Robust watermarks for ip protection of embedded quantized large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1813–1830, 2024.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, Somesh Jha, Lei Li, Yu-Xiang Wang, and Dawn Song. Sok: Watermarking for ai-generated content, 2025. URL <https://arxiv.org/abs/2411.18479>.