mathematics

MDPI

Review

Watermarking for Large Language Models: A Survey

Zhiguang Yang [1], Gejian Zhao [1] and Hanzhou Wu [1,2,*]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; yangzg@shu.edu.cn (Z.Y.); 23820171@shu.edu.cn (G.Z.)
[2] School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550025, China
* Correspondence: hanzhou@shu.edu.cn

Abstract: With the rapid advancement and widespread deployment of large language models (LLMs), concerns regarding content provenance, intellectual property protection, and security threats have become increasingly prominent. Watermarking techniques have emerged as a promising solution for embedding verifiable signals into model outputs, enabling attribution, authentication, and mitigation of unauthorized usage. Despite growing interest in watermarking LLMs, the field lacks a systematic review to consolidate existing research and assess the effectiveness of different techniques. Key challenges include the absence of a unified taxonomy and limited understanding of trade-offs between capacity, robustness, and imperceptibility in real-world scenarios. This paper addresses these gaps by providing a comprehensive survey of watermarking methods tailored to LLMs, structured around three core contributions: (1) We classify these methods as training-free and training-based approaches and detail their mechanisms, strengths, and limitations to establish a structured understanding of existing techniques. (2) We evaluate these techniques based on key criteria—including robustness, imperceptibility, and payload capacity—to identify their effectiveness and limitations, highlighting challenges in designing resilient and practical watermarking solutions. (3) We also discuss critical open challenges while outlining future research directions and practical considerations to drive innovation in watermarking for LLMs. By providing a structured synthesis, this work advances the development of secure and effective watermarking solutions for LLMs.

Keywords: watermarking; large language models; security; deep learning; information hiding

MSC: 68T07

check for updates

Academic Editor: Antanas Cenys

Received: 24 March 2025
Revised: 16 April 2025
Accepted: 24 April 2025
Published: 26 April 2025

Citation: Yang, Z.; Zhao, G.; Wu, H. Watermarking for Large Language Models: A Survey. *Mathematics* **2025**, *13*, 1420. https://doi.org/10.3390/math13091420

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rise of large language models (LLMs) has fueled optimism about the development of artificial general intelligence (AGI), as these models have achieved significant progress in recent years. Advances in efficient training methods, reinforcement learning, and scaling laws have enabled LLMs to generate coherent and plausible text while achieving performance comparable to—or even surpassing—human capabilities across various benchmarks, including machine translation, strategic gameplay (e.g., Go), and code generation.

The advent of the Transformer [1] architecture has revolutionized the field of language models, leading to unprecedented advancements. Transformer-based models, encompassing encoder-based [2], encoder–decoder [3], and decoder-based [4,5] structures, have become the de facto foundational models in natural language processing. Furthermore,

the application of Scaling Laws has driven significant improvements in model performance [6]. For example, the accuracy of benchmarks such as Massive Multitask Language Understanding (MMLU) has dramatically increased from levels approaching random chance to nearing 90%. As illustrated in Figures 1 and 2, progress in Chatbot Arena and MMLU benchmarks highlights the rapid advancement of LLM capabilities. Chatbot Arena, which evaluates conversational abilities by simulating real-world dialogue interactions, demonstrates improvements in models' ability to engage in coherent and contextually relevant conversations [7]. Meanwhile, MMLU assesses performance across 57 diverse tasks, offering a comprehensive measure of LLMs' proficiency in areas ranging from commonsense reasoning to domain-specific knowledge [8]. The substantial increase in scores across these benchmarks reflects a clear upward trend, demonstrating LLMs' ability to surpass previous iterations and achieve human-level performance.

The emergence of LLMs has raised significant concerns [9], especially regarding the potential generation of false information and the spread of misleading content on social media. In addition, LLMs pose pressing security risks, particularly regarding their reliability and inherent biases. The widespread use of these models also introduces broader systemic risks, such as privacy violations, intellectual property issues, and other security threats. These challenges emphasize the urgent need for robust safeguards and regulatory frameworks to mitigate the negative societal impacts of LLM technologies. Fortunately, watermarking technology offers a promising solution to ensure the integrity and traceability of content generated by these models.

Watermarking techniques, which embed information into digital content, are widely used across various fields, including image [10–12], audio [13], video [14], text [15,16], and deep learning models [17,18]. In the context of LLMs, the most straightforward approach involves applying existing text watermarking methods to subtly modify the text generated by the model, thereby embedding the watermark. Traditional text watermarking includes format-based (layout adjustments like line-shift coding) [15,19], lexical (synonym substitution) [20,21], syntactic (sentence restructuring) [22], and generation-based (neural network-generated watermarked text) approaches [23]. However, these methods struggle with robustness against modifications like paraphrasing and reformatting, limiting their effectiveness for LLMs. LLMs introduce additional challenges due to their probabilistic text generation, high linguistic variability, and susceptibility to adversarial attacks. The scale and adaptability of LLMs further complicate watermarking, requiring methods that integrate seamlessly with diverse model architectures and output styles. This has driven researchers to develop more sophisticated watermarking techniques tailored to LLMs, focusing on enhancing robustness, imperceptibility, and verifiability.

Beyond watermarking, researchers have also explored fingerprinting techniques for model attribution and ownership verification. These approaches do not directly modify the generated text, but instead aim to identify intrinsic characteristics of a model—such as statistical biases or distinctive generation patterns—that can be linked to a specific model instance or owner [24,25]. While fingerprinting enables ownership tracing even in the absence of explicit signals, it often requires broader access to internal model parameters or outputs and typically cannot operate solely on generated text. Closely related is steganography, which aims to conceal hidden information within natural language by modifying surface features such as word choice, syntax, or formatting. Linguistic steganography prioritizes imperceptibility and undetectability, often for covert communication [26,27]. In contrast, watermarking for LLMs emphasizes verifiability and robustness, embedding detectable signals—often at the token level—to support provenance, attribution, and tamper resistance.

In this study, we focus on watermarking techniques for text generated by LLMs, with an emphasis on algorithmic approaches, evaluation criteria, and practical deployment

considerations. While prior surveys have made significant contributions to understanding watermarking in the context of LLMs, several recent studies have explored distinct facets of this rapidly evolving field. For example, Liu et al. [28] provided a broad overview of text watermarking techniques, including their applications in LLMs. Lalai et al. [29] introduced a taxonomy grounded in intentions, while Zhao et al. [30] and Liang et al. [31] broadened the scope to address multimodal watermarking in AI-generated content. Existing surveys, while insightful, tend to focus on narrow aspects such as traditional techniques or general AIGC concerns, lacking a comprehensive analysis of the algorithmic trade-offs and unique challenges specific to LLM watermarking. This study fills that gap by presenting a unified, algorithmic perspective on LLM watermarking, with a focus on implementation strategies and performance trade-offs.

Overall, the contributions of this survey are as follows:

- We present a comprehensive survey of watermarking techniques for text generated by large language models. The focus is on usage scenarios where models produce natural language outputs—such as open-domain dialogue and content creation—that are especially vulnerable to misuse and thus demand mechanisms for origin verification and ownership attribution.
- We provide a systematic categorization of LLM watermarking approaches, dividing them into training-free and training-based methods. This taxonomy captures key algorithmic differences and reflects the practical implementation strategies observed in recent research.
- From a watermarking perspective, we evaluate existing techniques in terms of robustness, imperceptibility, and capacity. This performance-oriented analysis highlights practical trade-offs and informs the selection and design of watermarking strategies for real-world LLM applications.

This paper is structured as follows: Section 2 provides preliminary information on LLMs and watermarking, along with a taxonomy of LLM watermarking algorithms. Sections 3 and 4 offer a comprehensive overview of LLM watermarking techniques, categorized into two main types: training-free and training-based methods. Section 5 discusses the evaluation metrics for LLM watermarking, focusing on robustness, stealthiness, and capacity. Section 6 highlights the challenges and future directions of LLM watermarking. Finally, Section 7 concludes the paper.

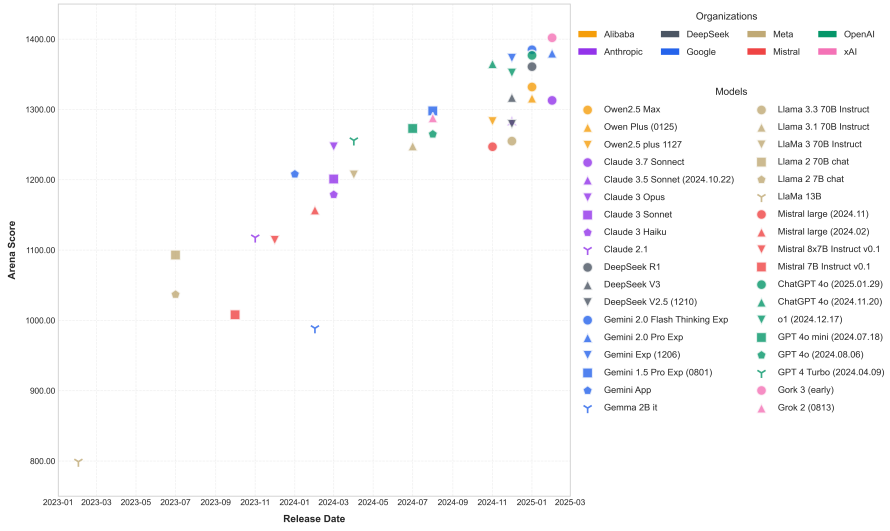


Figure 1. LLMs' leaderboard scores on Arena.



Figure 2. LLMs' accuracy on MMLU benchmark.

2. Preliminaries

2.1. Large Language Models

Large language models (LLMs) have demonstrated remarkable capabilities across various tasks, often surpassing human performance on multiple benchmarks. At their core, LLMs generate text autoregressively by computing the probability distribution over the next token given the preceding sequence.

Formally, given a sequence of prior tokens $t^0, \ldots, t^{i-1}$, the LLM computes the logits score $\mathbf{l}^i$ for the next token t^i as follows:

$$\mathbf{l}^i = \mathcal{M}(t^0, \ldots, t^{i-1}) \tag{1}$$

These logits are then converted into a probability distribution $\mathbf{p}$ using the softmax function as defined in Equation (2), where $\mathcal{V}$ represents the vocabulary.

$$\mathbf{p}^i = \text{softmax}(\mathbf{l}^i) = \frac{e^{\mathbf{l}^i}}{\sum_{j=1}^{|\mathcal{V}|} e^{\mathbf{l}^i_j}} \tag{2}$$

The next token t^i is then sampled from this distribution using strategies such as argmax decoding, nucleus sampling [32], or other stochastic and deterministic decoding methods.

2.2. Watermarking in LLMs

Watermarking is the process of embedding secret or distinctive information into a medium in a manner that is typically imperceptible to users but can be detected or extracted under specific conditions. The primary objectives of watermarking include asserting ownership, enabling traceability, and protecting intellectual property by ensuring that the embedded information remains detectable in cases of unauthorized use or modification. Traditional watermarking has been widely used for digital images, audio, and text to assert ownership and protect intellectual property. However, watermarking LLMs presents unique challenges, as it involves embedding information in probabilistic text generation rather than static content.

In the context of LLMs, watermarking typically involves embedding a secret message m within the model itself, ensuring that the generated text carries an identifiable signature. This embedding process is carried out by an encoding function, denoted as Enc, which

modifies the original model $\mathcal{M}$ to create a watermarked model, $\mathcal{M}_{\text{wat}}$, as expressed in Equation (3).

$$\mathcal{M}_{\text{wat}} = \text{Enc}(\mathcal{M},\ m) \tag{3}$$

The encoding function introduces controlled perturbations into the model's generation process, ensuring that responses subtly incorporate the watermark without significantly altering text fluency or coherence. Once the watermark has been embedded, the resulting watermarked model can generate outputs that carry this secret message in an imperceptible form. To extract the watermark, a decoding function, denoted as Dec, is employed. The decoding function takes as input the output generated by the watermarked model, along with a predefined watermark key, to extract the hidden message m, as illustrated in Equation (4).

$$\text{Dec}(\mathcal{M}_{\text{wat}}(\text{prompt}),\ \text{key}) = m \tag{4}$$

Through this process, watermarking provides a mechanism for tracking the provenance and integrity of models, ensuring that any unauthorized use can be detected through the recovery of the embedded watermark.

2.3. Taxonomy of LLM Watermarking Algorithms

LLM watermarking methods can be broadly categorized into training-free and training-based approaches, as illustrated in Figure 3. Each category encompasses various representative techniques, which differ in their implementation and impact on the model.

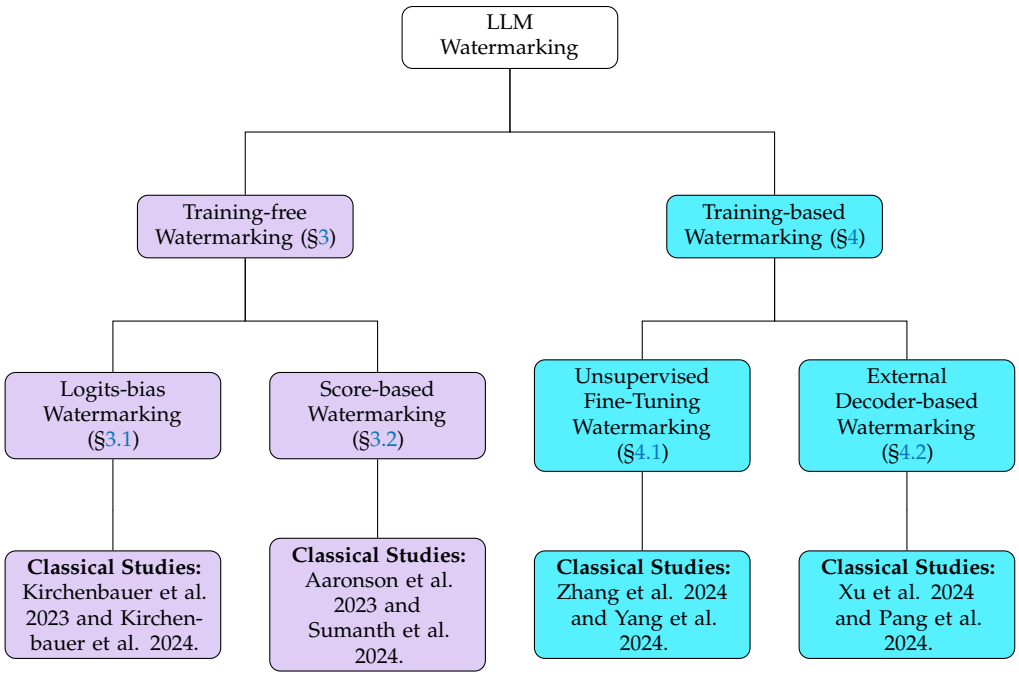


Figure 3. LLM watermarking methods can be broadly divided into two categories: training-free watermarking (Section 3) and training-based watermarking (Section 4). Representative training-free methods include those proposed by Kirchenbauer et al. (2023) [33], Kirchenbauer et al. (2024) [34], Aaronson et al. (2023) [35], and Sumanth et al. (2024) [36]. Training-based methods include Zhang et al. (2024) [37], Yang et al. (2024) [38], Xu et al. (2024) [39], and Pang et al. (2024) [40].

- Training-free methods, such as logits-bias watermarking and score-based watermarking, operate during the decoding process by adjusting token probabilities. These methods embed statistical patterns into the generated text without modifying the model parameters, making them computationally efficient and easy to integrate with existing models.

- Training-based methods include unsupervised fine-tuning watermarking and external decoder-based watermarking. These approaches either fine-tune the model to generate watermarked outputs intrinsically or introduce an auxiliary decoder. Such methods typically require additional training resources.

A detailed classification of specific watermarking techniques, along with their respective publication timelines, is presented in Table 1.

Table 1. Classification and characteristics of LLM watermarking methods.

Method	Category	Capacity	Publication Year
Aaronson et al. [35]	Score-based	1-bit	2023
Bahri et al. [41]	Score-based	1-bit	2024
Baldassini et al. [42]	Unsupervised fine-tuning	Multi-bits	2024
Boroujeny et al. [43]	Score-based	Multi-bits	2024
Chen et al. [44]	Logits-based	1-bit	2024
Christ et al. [45]	Score-based	1-bit	2024
Cohen et al. [46]	Logits-based	Multi-bits	2025
Dathathri et al. [36]	Score-based	1-bit	2024
Feng et al. [47]	Logits-based	1-bit	2024
Fernandez et al. [48]	Logits-based	Multi-bits	2023
Fu et al. [49]	Logits-based	1-bit	2024
Giboulot et al. [50]	Score-based	1-bit	2024
Golowich et al. [51]	Score-based	1-bit	2024
Guo et al. [52]	Logits-based	1-bit	2024
He et al. [53]	Logits-based	1-bit	2024
Hoang et al. [54]	Logits-based	1-bit	2024
Hou et al. [55]	Logits-based	1-bit	2024
Hou et al. [56]	Logits-based	1-bit	2024
Huo et al. [57]	Logits-based	1-bit	2024
Jiang et al. [58]	Logits-based	Multi-bits	2024
Kirchenbauer et al. [33]	Logits-based	1-bit	2023
Kirchenbauer et al. [34]	Logits-based	1-bit	2024
Kuditipudi et al. [59]	Score-based	1-bit	2024
Li et al. [60]	Logits-based	1-bit	2023
Li et al. [61]	Logits-based	Multi-bits	2024
Li et al. [62]	Score-based	1-bit	2025
Liu et al. [63]	Logits-based	1-bit	2024
Liu et al. [64]	Logits-based	1-bit	2024
Niess et al. [65]	Logits-based	1-bit	2024
Pang et al. [66]	Logits-based	1-bit	2024
Pang et al. [40]	External decoder-based	1-bit	2024
Qu et al. [67]	Logits-based	Multi-bits	2024
Ren et al. [68]	Logits-based	1-bit	2024
Ren et al. [69]	Logits-based	1-bit	2024
Song et al. [70]	Logits-based	Multi-bits	2025
Wang et al. [71]	Logits-based	Multi-bits	2024
Wu et al. [72]	Score-based	1-bit	2024
Xu et al. [39]	External decoder-based	1-bit	2024
Xu et al. [73]	External decoder-based	Multi-bits	2024
Yang et al. [38]	Unsupervised fine-tuning	Multi-bits	2024
Yoo et al. [74]	Logits-based	Multi-bits	2024
Zamir et al. [75]	Score-based	Multi-bits	2024
Zhang et al. [37]	Unsupervised fine-tuning	Multi-bits	2024
Zhao et al. [76]	Logits-based	1-bit	2024
Zhao et al. [77]	Logits-based	1-bit	2024
Zhong et al. [78]	External decoder-based	1-bit	2024
Zhou et al. [79]	Score-based	1-bit	2024

3. Training-Free Watermarking

These methods aim to embed distinct, imperceptible markers into the outputs of LLMs without requiring model fine-tuning, offering a more efficient and scalable solution in terms of deployment and system integration. By leveraging the logits generated during the generation process of LLMs, training-free watermarking techniques seamlessly incorporate watermarks into the text generation process. Such approaches are particularly advantageous in scenarios where access to the underlying model is restricted or where preserving model performance without the overhead of retraining is crucial. In this section, we explore the mechanisms underlying training-free watermarking algorithms, their implementation strategies, and their effectiveness in safeguarding intellectual property while preserving the usability and integrity of LLM outputs. Based on the watermark embedding structure, we categorize these methods into logits-bias and score-based approaches.

3.1. Logits-Bias Watermarking

Logits serve as a crucial scoring indicator in LLM autoregressive generation, containing rich contextual information and being easily accessible during the inference process. A straightforward approach involves designing an encoding function to modify logits for embedding watermark information, as shown in Equation (5). However, a key challenge is ensuring that the watermark remains detectable in the generated text.

$$\mathbf{l}_w = \mathrm{Enc}(\mathcal{M}(\mathbf{t}^0, \ldots, \mathbf{t}^{i-1}),\ m) \tag{5}$$

Kirchenbauer et al. proposed a pioneering method to address this problem [33]. Their method divides the vocabulary into two lists, red and green, enabling the conversion of watermark signals from logits into text. By modifying the logits score, the LLM preferentially selects tokens from the green list. In contrast, a model without a watermark should exhibit no preference between tokens in the red and green lists. By counting the occurrences of these tokens in a candidate text, one can determine whether the text contains a watermark.

The overall framework of the logits-bias watermarking method is illustrated in Figure 4. In particular, Kirchenbauer et al. employed a hash function as a vocabulary split function to divide the vocabulary into two subsets, the red ($\mathcal{R}$) and the green ($\mathcal{G}$) lists, using γ as a proportional factor to control the division ratio and a key, dependent on the preceding token, to regulate the hash function. The original logits, $\mathbf{l}_\mathrm{o}$, generated by the large model, were then modified to obtain the watermarked logits, $\mathbf{l}_\mathrm{w}$, as defined in Equation (6). Specifically, t_j denotes a token in the vocabulary, and δ represents the added bias, which is a predefined constant in the setting of Kirchenbauer et al. The model generates watermarked text by continuously predicting the next token through autoregression.

$$\mathbf{l}_\mathrm{w} = \mathbf{l}_\mathrm{o} + \delta \cdot \mathbf{1} = \begin{cases} l_\mathrm{o} + \delta, & t_j \in \mathcal{G} \\ l_\mathrm{o}, & t_j \in \mathcal{R} \end{cases} \tag{6}$$

In the watermark detection stage, we count the occurrences of green list tokens in the candidate text, compute the z-score using Equation (7), and determine whether the text contains a watermark based on the resulting score. Here, $|s|_\mathcal{G}$ represents the number of green tokens in the candidate text, and T denotes the total number of tokens.

$$z = (|s|_\mathcal{G} - \gamma T)/\sqrt{T\gamma(1-\gamma)} \tag{7}$$

The method proposed by Kirchenbauer et al. laid the foundation for the logits-bias watermark technique and demonstrated strong performance. However, the selection of hyperparameters γ and δ presents intriguing trade-offs. For instance, increasing δ raises

the probability of selecting tokens from the green list, facilitating detection but potentially altering the original semantics. Similarly, a larger γ expands the green list and enhances robustness but also increases the likelihood of false positives. These trade-offs make parameter selection a critical consideration. Moreover, despite its effectiveness, the method still faces significant challenges in real-world deployment. Furthermore, different watermarking strategies provide distinct advantages. Table 2 summarizes the potential benefits of various logits-bias watermarking methods, offering a structured overview of their strengths. The following sections explore key aspects in detail: improving robustness, enhancing concealment, and expanding watermark capacity.

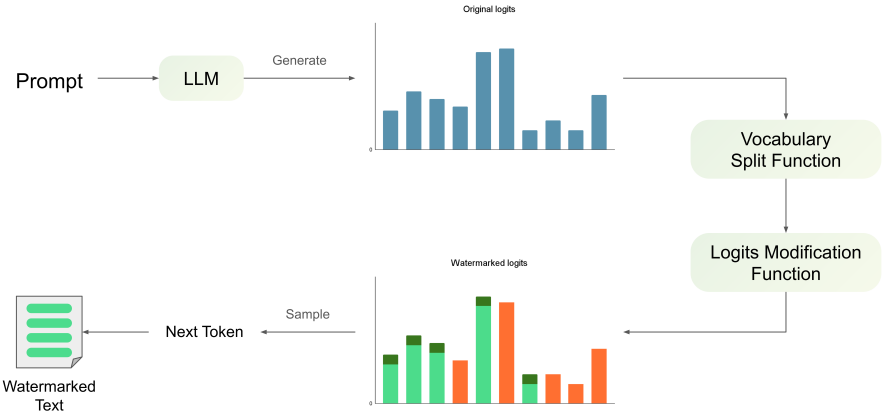


Figure 4. Overview of logits-bias watermarking framework. Blue represents the original logits scores. Orange and green represent the scores of tokens from $\mathcal{R}$ and $\mathcal{G}$, respectively, with darker shades indicating the added bias δ applied to these tokens.

Table 2. Potential benefits of different watermarking strategies.

Tailored Strategies	Potential Benefits	Explanation
Vocabulary Partition	Improved Robustness	Minor modifications that do not alter semantics will not affect vocabulary partitioning.
	Enhanced Concealment	Minimal impact on downstream tasks, with sufficient watermarked tokens preserving original semantics.
Bias Adjustment	Enhanced Concealment	Reducing the bias of semantically important tokens prevents significant meaning shifts.
	Improved Robustness	Increasing the bias of less important tokens enhances the probability of watermark presence.
Embedding Token Selection	Enhanced Concealment	Avoids modifying low-entropy text, thus preserving text quality.

3.1.1. Robustness Improvement

One straightforward attack method involves embedding watermark text within a long document. According to Equation (7), the detection z-score is dependent on the text length. Consider a scenario where a watermarked text is embedded within a large non-watermarked text. How can it be detected? Applying Equation (7) directly yields a large T and a small $|s|_{\mathcal{G}}$, resulting in a small z-score. To address this issue, Kirchenbauer et al. proposed the winMax strategy for windowed detection, which is sensitive to short spans of watermarked text embedded within large documents [34]. This strategy divides the text into multiple windows and calculates the maximum z-score across all windows. The watermark is detected if the maximum z-score exceeds a predefined threshold. This approach substantially enhances the robustness of the watermarking method against

long-text attacks. To further improve computational efficiency, a more efficient method has been proposed [77].

In natural language processing, paraphrasing is a prevalent attack strategy that alters portions or the entirety of a text while maintaining its original meaning. This attack poses a challenge to watermarking methods, as it can effectively obscure or remove embedded watermarks without significantly changing the conveyed information. Since the vocabulary partitioning in Kirchenbauer et al.'s method is conditioned on the preceding token, paraphrasing attacks can substantially alter this division during the detection stage, ultimately affecting the accuracy and reliability of watermark extraction. A straightforward approach is to fix the list partitioning method, minimizing the impact of text paraphrasing on the division of red and green lists. Zhao et al. demonstrated that a fixed hash function exhibits greater robustness against text editing and paraphrasing [76]. However, this fixed method introduces several additional problems. For instance, it increases the selection probability of certain tokens, deviating from natural language distributions and making the watermarking method more detectable, thereby increasing its vulnerability to attacks. Additionally, it decreases the selection probability of other tokens, preventing them from being chosen when appropriate, essentially degrading the quality of the watermarked text. Therefore, its applicability in real-world scenarios is limited.

The primary impact of paraphrasing arises from the rewritten token failing to fall within the green list, which, in turn, disrupts the detection results. Consequently, designing a vocabulary segmentation strategy that preserves token classification after a paraphrasing attack is a promising approach to enhancing robustness. As illustrated in Figure 4, by tailoring the design of the vocabulary splitting function, the token resulting from the paraphrase attack remains within the green list, thereby influencing the logits modification function and altering the probability of the green token being selected. To divide the red and green lists, the researchers employed three distinct methods: the sentence-level local perceptual hashing (LSH) method [55], semantic-based clustering [44], and the k-means clustering technique [56]. These methods utilize semantic information to partition the red and green lists, allowing for the introduction of minor disturbances that preserve both semantics and the integrity of the list divisions, thereby enhancing the robustness. In addition to heuristic strategies, some lightweight deep learning models have been introduced, combining semantic information to replace the hash function for partitioning the red and green lists [47,57,63,68].

Translation attacks involve two types of detection: source language detection and target language detection. Source language detection can be considered a form of paraphrasing attack, as it still detects the same language. Target language detection is essential, as LLMs support multilingual input and output, making it difficult to identify the source language. Therefore, detection must be conducted in the target language. He et al. proposed a method that utilizes an external vocabulary [53]. By grouping tokens with the same semantics but in different languages into the same group of the red and green lists, the translated target language text remains aligned with the corresponding groups, thereby improving resistance to translation attacks.

Logits-bias watermark detection algorithms typically rely on the secret key used during the watermark embedding process. This reliance creates significant security vulnerabilities, exposing the system to potential breaches and counterfeiting during public detection. Specifically, requiring the secret key for watermark validation undermines the system's robustness, as unauthorized access or leakage of the key could allow malicious actors to forge or manipulate watermarks. In response to these concerns, several studies have identified these vulnerabilities and proposed effective solutions to mitigate the associ-

ated risks, aiming to enhance the security and integrity of watermarking systems in open environments [64,66,80].

3.1.2. Concealment Enhancement

Concealment enhancement aims to improve the imperceptibility of watermarks, ensuring they integrate seamlessly into natural language while remaining detectable. Effective techniques focus on minimizing linguistic artifacts, optimizing token selection, and preserving text fluency. Furthermore, these methods also need to ensure that watermarking does not degrade performance on downstream tasks, thereby maintaining the overall utility of the generated content.

The logits-bias watermark demonstrates superior performance in open-ended questions, where its effectiveness closely matches that of the original task. However, in conditional downstream tasks, such as classification, it results in a more noticeable decline in task-related performance metrics [49]. For tasks with conditional constraints, the next-token distribution is further influenced by the conditions specified in the prompt. Consequently, when constructing the red and green lists, random partitioning and score modifications exert a greater impact on the original outputs, thereby affecting the performance of downstream tasks. By integrating semantic information with the vocabulary's embedding matrix, highly relevant tokens can be assigned to the green list to increase their selection probability, thereby minimizing the impact on downstream tasks [49,52,65]. Additionally, watermark embedding can be conditioned on the next token's probability, avoiding insertion when the probability is excessively high or low to prevent significant alterations to the text [69]. An alternative approach involves determining the placement of the watermark based on the part of speech of the subsequent token [54], ensuring that the watermark is embedded only in tokens that have minimal impact on the semantics. A more refined strategy calculates the importance score of the next token, ensuring that highly important tokens remain unmodified while embedding the watermark only in less critical positions [60].

Low-entropy text presents unique challenges for watermarking. From an information-theoretic perspective, its limited capacity makes it unsuitable for embedding additional watermark information. Forcing watermark insertion in such text can significantly alter its meaning. For instance, consider the sentence "The quick brown fox jumps over the lazy dog", which is specifically crafted to contain all 26 letters of the alphabet. Embedding a watermark in this case would interfere with its intended structure and purpose. To mitigate this issue, the most common approach is to exclude low-entropy text from watermarking. Liu et al. [63] proposed a model-based method that employs an auxiliary model to estimate the text's information entropy. If the entropy falls below a predefined threshold, the watermark is not embedded, thereby preserving the text's quality.

3.1.3. Multi-Bit Watermarking

The watermarking method proposed by Kirchenbauer et al. verifies whether the candidate text contains a sufficient number of tokens from a predefined "green list". While effective in detecting watermark presence, this approach is fundamentally limited to implementing single-bit watermarks. However, real-world applications often require embedding additional information, such as copyright details and user identifiers.

A natural approach is to partition the text into multiple blocks, with each block encoding a different watermark bit. Specifically, each block employs distinct red and green lists to represent the binary values 0 and 1, respectively. From the perspective of watermark extraction, Wang et al. employed this strategy to segment text into distinct blocks, each encoding a different watermark bit [71]. Inspired by this approach, Jiang et al. adopted a similar strategy [58]. Cohen et al. extended this method further by integrating it into a

multi-user watermarking framework [46]. Nonetheless, encoding only a single bit within a long text is inefficient, requiring an excessively lengthy text to embed a sufficient number of watermark bits.

An alternative to text segmentation is watermark segmentation, where the previous token determines which corresponding watermark bit is embedded in the next token [61,67]. Compared to the above multi-bit watermarking approach based on text segmentation, this strategy enhances text utilization and allows for embedding more watermark bits, as it adaptively determines whether a watermark has been embedded before switching to the next one. However, this method undermines the robustness of the watermark, as extraction relies on the previous token. To address this limitation, scholars have introduced error correction coding to enhance robustness [67,70].

Additionally, the red and green lists are expanded into N distinct splits, each corresponding to a unique watermark message, resulting in a payload of $\log_2 N$ bits. Yoo et al. adopted this strategy and integrated it with the previous approach to achieve multi-bit watermarking [74]. Fernandez et al. proposed a watermarking method known as the Cyclic Shift method, asserting that, in theory, it can be easily extended to support multi-bit watermarking [48].

3.2. Score-Based Watermarking

Score-based watermarking does not embed watermarks by directly modifying the logits distribution. Instead, it employs a separate scoring function to compute the watermark score. Intuitively, during token sampling, the score-based watermarking method differs from the standard LLM sampling approach, which typically selects tokens based solely on the highest logits scores. In contrast, the watermarking method selects tokens by optimizing both the logits and the watermark scores, as illustrated in Figure 5. As defined in Equation (8), $\mathbf{p}$ represents the sampled probability distribution, $\mathbf{l}$ denotes the logits score, $\mathbf{s}$ refers to the watermark score, and the function $f(\cdot)$ combines $\mathbf{l}$ and $\mathbf{s}$. This approach embeds the watermark while preserving semantic continuity and integrity. Moreover, it maintains the original distribution, thereby enhancing the concealment of the watermark.

$$\mathbf{p} = f(\mathbf{l}, \mathbf{s}) \tag{8}$$

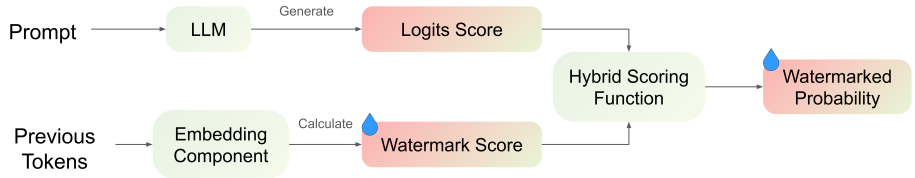


Figure 5. Overview of score-based watermarking framework.

During the detection stage, it is sufficient to apply the corresponding watermark score function to evaluate the candidate text. If the score exceeds a predefined threshold, the text is classified as containing a watermark.

OpenAI researchers proposed a watermarking algorithm incorporating a pseudo-random function to compute the watermark score [35]. Specifically, as expressed in Equation (9), $f_s(\cdot)$ represents a pseudo-random function, while s is a secret random seed that must be kept confidential. Additionally, $\mathbf{p}_o^i$ denotes the probability assigned by the original LLM, and $t_{i-n+1}, \ldots, t_{i-1}$ represents the previous tokens. During the sampling process, the token with the highest probability is selected as the output.

$$\mathbf{p}_w^i \leftarrow \mathbf{r}_i^{1/\mathbf{p}_o^i}, \quad \text{where } \mathbf{r}_i := f_s(t_{i-n+1}, \ldots, t_{i-1}), \text{ and } \mathbf{r}_i \in [0,1]^{|\mathcal{V}|} \tag{9}$$

Intuitively, a token is chosen as the next token only when both its logits score and watermark score are close to 1. Consequently, during detection, a logarithmic algorithm can identify instances where the watermark score is close to 1, resulting in a significantly large value, which aids in the verification of the embedded watermark, as indicated in Equation (10). If the computed score exceeds a predefined threshold, the text is determined to contain a watermark.

$$s_d = \sum_{i=1}^{T} \ln \frac{1}{1 - r'_t}, \quad \text{where} \quad r'_t := f_s(t_{i-n+1}, \ldots, t_i) \tag{10}$$

Kuditipudi et al. proposed a distortion-free watermarking strategy that replaces the random function with a sequence $\xi = \xi^{(1)}, \ldots, \xi^{(m)}$ of random numbers encoded by a secret watermark key [59]. As shown in Equation (11), $\xi^{(j)}$ represents the j-th element of the random sequence, and p_o^i denotes the original probability. This approach ensures that the watermark distribution preserves the original distribution, thereby guaranteeing distortion-free embedding. However, this does not imply that the two distributions are exactly the same. While this method guarantees a distortion-free watermark, it encounters security risks related to watermark key sharing during detection. To address this issue, Zhou et al. employed digital signatures as a mitigation strategy [79].

$$p_w^i \leftarrow \left(\xi^{(j)}\right)^{1/p_o^i} \tag{11}$$

Similarly, unbiased watermarking takes advantage of the inherent randomness in the sampling process to control the selection of markers. By utilizing a specific random function, the sampling process can be deliberately guided, with this guidance serving as the mechanism for watermark embedding, thus ensuring the watermarking method remains unbiased [72,81].

3.2.1. Concealment Enhancement

Effective concealment in score-based watermarking ensures that the introduced modifications remain statistically indistinguishable from natural language variations. Christ et al. employ a binary encoding scheme for each word in the vocabulary, utilizing pseudo-random numbers represented as a sequence of values $u \in [0,1]$. This approach enhances the efficiency of the sampling process by leveraging these pseudo-random numbers, ensuring that the watermark remains undetectable [45]. Specifically, if the predicted probability at a given position exceeds the corresponding pseudo-random number, a 1 is assigned; otherwise, a 0 is chosen. In watermark detection, this method enables the evaluation of whether pseudo-random numbers associated with positions labeled as 1 are significantly greater than those labeled as 0.

Dathathri et al. introduce SynthID, a production-ready watermarking scheme that effectively preserves text quality while ensuring high detection accuracy [36]. Specifically, the authors utilize the tournament sampling method, which begins by performing repeated sampling based on the original distribution computed by the LLM to generate a large set of tokens. These tokens are then subjected to a competition, with each token acting as a contestant according to predefined tournament rules, and the token that emerges as the winner is selected. Since the sampling process follows the original distribution, the final outcome remains largely unaffected. For instance, in an extreme case where the original distribution assigns nonzero probability to only a single token, the tournament outcome remains unchanged regardless of the number of iterations, as all contestants are identical.

Building on the sampling techniques discussed earlier, the scoring-based watermarking strategy introduces a distinct approach by computing a watermark score during detection, making it particularly well suited for the greedy search method. This approach simultaneously generates multiple text candidates and selects the one with the highest score as the final output, thereby significantly improving watermark quality. For example, Giboulot et al. [50] and Bahri et al. [41] employ an LLM to generate several candidate texts and then select the one with the highest score as the final output.

3.2.2. Multi-Bit Watermarking

Multi-bit watermarking within score-based frameworks aims to maximize embedding capacity while preserving the natural coherence of generated text. Similar to the logits-bias watermarking method, a feasible approach for score-based multi-bit watermarking is to predefine a mapping rule to determine which bit carries the watermark.

Researchers built upon the work of Christ et al. and implemented multi-bit watermarking by designing corresponding mapping rules to enhance watermark capacity [43]. These methods enable efficient encoding of multiple bits within generated text while maintaining imperceptibility.

Beyond mapping rules, cryptographic techniques have been introduced to ensure secure embedding of multi-bit watermarks. A recent study proposed a cryptographic method to conceal an arbitrary secret payload while ensuring that the quality of the generated text remains unaffected [75]. This approach integrates encryption-based watermarking with probabilistic score adjustments, offering a balance between security and readability.

By refining these methodologies, score-based multi-bit watermarking continues to evolve as a viable approach for embedding higher-capacity watermarks while maintaining text fluency and robustness.

3.2.3. Statistical Analysis

The score-based watermarking method maintains the integrity of the original distribution by avoiding direct modifications to the logits score. This preservation is crucial, as it enables a more reliable analysis of the watermarking process. Furthermore, the additional score function operates independently of the original distribution, complicating the development of a comprehensive theoretical framework to assess its effects.

In this context, Li et al. proposed a general and flexible framework to evaluate the statistical efficiency of watermarking techniques [62]. Their work not only provides insights into the effectiveness of various watermarking strategies but also contributes to the formulation of robust detection rules essential for assessing watermark integrity and reliability across different applications.

Similarly, Golowich et al. provided theoretical support on the resilience of watermarks against text modifications, further emphasizing the necessity of a strong statistical foundation in watermarking practices [51]. This theoretical backing is crucial for understanding a watermark's ability to withstand alterations while preserving the integrity of the original content against potential tampering.

4. Training-Based Watermarking

While training-free methods provide a practical approach to watermarking LLMs, training-based techniques offer a deeply integrated solution. These methods embed watermarking mechanisms during the model training phase, enabling the creation of resilient and task-adaptive watermarks. By incorporating watermark signals directly into the training data or optimization objective, training-based approaches leverage the model's learning capacity to enhance watermark persistence under adversarial conditions and across diverse

tasks. Although training-based methods typically incur higher computational costs during model preparation, they introduce no additional overhead at inference time, as the watermarking behavior is intrinsically encoded within the model parameters. This makes them particularly suitable for applications where both watermark fidelity and runtime efficiency are critical.

This section explores two prominent training-based approaches for watermark embedding in LLMs: unsupervised fine-tuning and external decoder-based approaches. Unsupervised fine-tuning involves integrating the watermark into the model from the outset, ensuring its seamless incorporation during the initial training phase. This process can be enhanced by leveraging the encoder–decoder architecture, where the watermark is embedded in the model's latent space, facilitating both deep integration and preservation of the model's core capabilities. In contrast, external decoder-based methods introduce the watermark in a separate training phase, allowing for targeted adjustments and optimization. This separation provides more flexibility in fine-tuning the watermarking process independently of the main model, but it may result in less seamless integration compared to methods that incorporate the watermark from the outset within the encoder–decoder framework.

4.1. Unsupervised Fine-Tuning Watermarking

Unsupervised fine-tuning has emerged as a pivotal approach for embedding watermarks in machine learning models, particularly in the context of image watermarking. This approach facilitates the incorporation of robust watermarks during the fine-tuning stage, seamlessly integrating them into the model's parameters. Embedding watermarks in this manner markedly improves their resistance to attacks and modifications, positioning this method as a valuable tool for copyright protection and verifying content authenticity.

Existing research has demonstrated the effectiveness of unsupervised fine-tuning in various watermarking applications. For instance, multiple studies have employed this technique to embed watermarks in images, yielding notable outcomes in terms of imperceptibility and robustness [18,82]. These approaches frequently incorporate adversarial training and noise injection to produce watermarks that resist removal while preserving the integrity of the original content. Furthermore, unsupervised fine-tuning capitalizes on expansive datasets without requiring labeled data, proving advantageous in contexts where such data are limited or challenging to acquire. Its adaptability extends to LLMs, presenting distinctive opportunities for watermark integration. In the context of LLM watermarking, this approach can be adapted to embed identifiable patterns or tokens within the generated text.

Abdelnabi et al. introduced the Adversarial Watermarking Transformer (AWT), an advanced end-to-end framework for embedding watermarks in natural language text. The AWT leverages a transformer-based encoder–decoder architecture to automatically conceal information while preserving linguistic fluency and semantic integrity. The model consists of two core components: a hiding network, responsible for encoding a binary message into a given input text with minimal perturbations, and a decoding network, which reconstructs the embedded message solely from the watermarked text. To enhance robustness against detection and removal, the AWT employs adversarial training by incorporating an auxiliary adversary tasked with distinguishing between original and watermarked text. Through this adversarial optimization process, the model is jointly trained to balance three competing objectives: maximizing message recoverability, minimizing perceptible changes to the text, and effectively deceiving the adversary. Zhang et al. proposed an optimized beam search algorithm and a reparameterization technique to improve the AWT, enhancing the coherence and robustness of the generated text [37]. Similarly, Bertini et al. employed

a cross-attention mechanism to construct a watermark layer within the model, further improving coherence compared to the AWT [42].

4.2. External Decoder-Based Watermarking

In the previous section, we introduced the unsupervised fine-tuning watermarking approach for LLMs, which embeds watermark information by directly optimizing the model parameters. However, as these models continue to grow in scale and complexity, relying solely on fine-tuning may not sufficiently address the stringent demands of watermarking in increasingly advanced application scenarios. Consequently, novel watermarking strategies are essential in ensuring that the embedded signals remain imperceptible while providing enhanced adaptability and resilience across diverse environments.

In this context, external decoder-based watermarking for LLMs emerges as a highly effective alternative. By leveraging an external decoder during the embedding process, this approach not only enhances flexibility and control but also significantly improves the security and stability of the watermark. In particular, external decoder-based watermarking facilitates a more flexible and verifiable extraction process, ensuring reliable detection and authentication of the embedded watermark.

Reinforcement learning (RL) is a powerful class of machine learning algorithms in which an agent learns to make decisions through interactions with an environment, receiving feedback in the form of rewards or penalties. This approach has gained significant attention for its ability to optimize complex decision-making processes in dynamic and uncertain environments. Notably, reinforcement learning from human feedback (RLHF) has emerged as a prominent technique in which human preferences guide the model's learning process, allowing models to better align with human goals and behaviors. Additionally, chain-of-thought (CoT) reasoning, facilitated by RL, has demonstrated remarkable success in enhancing the interpretability and performance of LLMs [83,84]. CoT reasoning enables models to break down complex tasks into logical steps, improving both reasoning capabilities and overall output quality.

Xu et al. used Proximal Policy Optimization (PPO) [85] to optimize an LLM and train it together with the watermark decoder, so that the text generated by the LLM could be extracted by the watermark decoder [39]. This method couples the watermark embedding process with RL and can be combined with the model alignment process to avoid additional overhead. However, this method is limited to verification and supports only single-bit watermarks. Xu et al. employed a pair of LLM paraphrasers to embed 0 and 1 bits, respectively, and then trained a classifier for detection [73]. Additionally, they implemented a strategy of embedding different watermark bits in separate sentences to achieve multi-bit watermarking.

In addition to directly designing strategies for watermark embedding, researchers leverage the capabilities of LLMs to generate watermarks and subsequently employ a detection model for verification [40,78]. By carefully crafting prompts, LLMs can autonomously develop watermarking strategies. One example involves using LLMs to design vocabulary replacement techniques, where specific words or phrases are substituted to subtly embed a watermark in the text. This approach leverages the model's generative capabilities to create watermarks that are both imperceptible and robust.

5. Evaluation Metrics

In the previous section, we provided a comprehensive overview of existing LLM watermarking techniques, highlighting their design paradigms and implementation strategies. To rigorously assess these methods, we now introduce a set of evaluation metrics centered around three fundamental dimensions: robustness, which captures the watermark's resilience to perturbations and adversarial transformations; imperceptibility, which ensures that the watermark does not compromise text quality or naturalness; and capacity, which measures the amount of information that can be embedded without degrading performance.

Table 3 provides a qualitative comparison of representative watermarking methods across key evaluation dimensions, including imperceptibility, two forms of robustness (against paraphrasing and token substitution), capacity, and resource efficiency.

The remainder of this section formalizes the definitions of the three core evaluation metrics and discusses practical tools for their assessment. These metrics provide a structured foundation for analyzing the strengths and limitations of current LLM watermarking methods.

Table 3. Qualitative comparison of watermarking methods across four major evaluation dimensions. Scores are expressed using symbolic ratings: +++ = strong, ++ = moderate, + = weak.

Method	Imperceptibility	Robustness		Capacity	Resource Efficiency
		Paraphrase	Token Substitution		
Aaronson et al. [35]	++	++	++	+	+++
Dathathri et al. [36]	+++	++	++	+	+++
Hou et al. [55]	++	++	++	+	+++
Kirchenbauer et al. [33]	++	+	+	+	+++
Kirchenbauer et al. [34]	++	++	++	+	+++
Kuditipudi et al. [59]	+++	++	++	+	+++
Liu et al. [64]	+	+	+	+	+++
Wang et al. [71]	++	+	++	++	+++
Xu et al. [39]	++	++	++	+	+
Xu et al. [73]	++	++	++	++	+
Zhang et al. [37]	++	++	++	++	+

5.1. Robustness

Robustness refers to the degree to which a watermarking scheme can maintain successful detection in the presence of modifications to the generated text. It is a fundamental requirement for any text watermarking method, as real-world content is often subject to both intentional manipulations (e.g., adversarial paraphrasing) and unintentional edits (e.g., user revisions). A comprehensive evaluation of robustness must therefore account for a diverse range of potential transformations.

In this study, we categorize robustness along three granular levels of attack: token-level, sentence-level, and document-level modifications. Each category reflects a distinct class of perturbations that may compromise watermark integrity. Table 4 illustrates representative examples across these levels, highlighting how different types of edits can affect the watermarked output. A more detailed analysis of each attack category, including their operational mechanisms and impact on watermark detectability, is presented in the subsequent sections.

Table 4. Examples of different attacks on watermarked text.

Attack Level	Attack Method	Example (Original)	Example (Attack)
Token-level	Synonym Replacement	You just can't differentiate between a robot and the very best of humans.	You simply cannot distinguish between a machine and the finest of people.
	Typo Insertion	You just can't differentiate between a robot and the very best of humans.	You just can't differntiate between a robot and the very best of humans.
Sentence-level	Paraphrasing	Artificial intelligence has advanced to the point where distinguishing between human and machine-generated text is increasingly difficult.	Distinguishing between human and machine-generated text has become increasingly difficult as artificial intelligence has advanced.
Document-level	Paraphrasing	Artificial intelligence has advanced to the point where distinguishing between human and machine-generated text is increasingly difficult. Watermarking techniques are being developed to address this challenge, ensuring the traceability and authenticity of AI-generated content.	As artificial intelligence continues to advance, it is becoming harder to differentiate between human and AI-generated text, prompting the development of watermarking techniques to ensure content authenticity.

5.1.1. Token-Level Attacks

Token-level attacks involve fine-grained text modifications, typically at the word or subword level. These attacks may arise naturally (e.g., typos) or be intentionally introduced to disrupt watermark detection. One common attack is character perturbations, where minor typographical errors are introduced, either accidentally or by adversaries aiming to degrade watermark signals. Another prominent attack is synonym substitution, wherein words are replaced with their synonyms, potentially altering the encoded watermark distribution while maintaining semantic coherence. More sophisticated attacks include token masking, where certain words are removed or replaced with generic placeholders, disrupting the statistical patterns leveraged by watermarking schemes [86].

5.1.2. Sentence-Level Attacks

At the sentence level, modifications affect the text's structure rather than individual tokens. Common transformations include sentence reordering, the insertion and deletion of sentences, and paraphrasing. These attacks pose a greater challenge because they can significantly alter the linguistic structure while preserving the overall meaning. Unlike token-level attacks, which primarily disrupt local patterns, sentence-level attacks can compromise watermark integrity at a broader contextual level.

5.1.3. Document-Level Attacks

Document-level attacks involve large-scale text modifications, often making it more challenging to recover the watermark. One prominent threat is the long-text attack, where watermark-embedded text is hidden within a larger, unmarked document, diluting the watermark signal and complicating extraction [34]. Another significant attack is text rewriting, where a passage is extensively rewritten to maintain its meaning while altering sentence structures, thereby posing significant hurdles to watermark extraction [87–89]. Additionally, machine translation attacks exploit differences in linguistic structures across languages to obfuscate watermarks. By translating a watermarked text into another language and back, adversaries can introduce transformations that may diminish watermark detectability [90].

5.2. Imperceptibility

Imperceptibility refers to the extent to which a watermark remains hidden within the generated text without degrading its fluency, coherence, or semantic content. It is a fundamental requirement for effective watermarking in LLMs, ensuring that the embedded

signal does not introduce noticeable distortions or artifacts. A well-designed watermarking scheme should preserve the naturalness of the output while minimizing any impact on both human perception and downstream NLP tasks.

Formally, imperceptibility can be quantified as the inverse of a distortion function between the original (non-watermarked) text T and the watermarked output T_w:

$$\text{Imperceptibility} = 1 - \text{Dist}(\mathrm{T},\ \mathrm{T}_w) \tag{12}$$

Here, $\text{Dist}(\cdot)$ denotes a suitable distance or dissimilarity metric that captures both surface-level and semantic changes between the original and watermarked text. In addition to measuring observable distortions, such metrics can also reflect the preservation of functional integrity in downstream tasks. Common choices include perplexity (PPL), which evaluates fluency and syntactic coherence, and BERTScore, which assesses semantic similarity. In the following analysis, we examine imperceptibility from two complementary perspectives: surface-level text quality and task-level performance consistency in downstream applications.

5.2.1. Text Quality

Text quality refers to the extent to which a watermarked text preserves the fluency, coherence, and semantic fidelity of its non-watermarked counterpart. One of the primary metrics for assessing text naturalness is perplexity (PPL), which measures how well a language model predicts a given text sequence. Lower perplexity indicates higher fluency and coherence, as the model assigns greater probability to the observed text. Given a sequence $T = \{t^1, t^2, \ldots, t^N\}$, its perplexity is computed as:

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(t^i|t^1, t^2, \ldots, t^{i-1})\right) \tag{13}$$

While PPL evaluates fluency from a generative modeling perspective, other metrics assess similarity between generated and reference texts. Beyond surface-level matching, BERTScore provides a semantic evaluation by aligning contextual embeddings of tokens in the generated and reference texts [91]. Instead of lexical matching, it computes the maximum cosine similarity for each token in the candidate text:

$$\text{BERTScore} = \frac{1}{|T|}\sum_{t\in T}\max_{t'\in R}\cos(\mathbf{h}_t, \mathbf{h}_{t'}) \tag{14}$$

Here, $\mathbf{h}_t$ is the contextual embedding of token t derived from a pretrained transformer model. T refers to the set of tokens from the generated text, and R represents the tokens from the reference text.

Together, these metrics offer a multifaceted view of text quality, spanning fluency, lexical fidelity, and semantic coherence. In practice, they are often used in combination to provide a more comprehensive assessment of watermark imperceptibility.

Existing benchmarks, such as the one proposed in [92], provide a systematic evaluation of different watermarking methods, assessing both their robustness and distortion levels. This study compares four major watermarking techniques, highlighting the trade-offs between imperceptibility and watermark detectability. Empirical results show that the watermark can be reliably detected with fewer than 100 tokens and exhibits strong resistance to simple perturbations. These findings suggest that such techniques are approaching practical application. However, the study also identifies limitations, notably in code generation, where watermarking methods struggle to efficiently embed and detect

watermarks. This points to the current limitations of these techniques in domains beyond natural language tasks.

5.2.2. Impact on Downstream Tasks

Beyond intrinsic text quality, imperceptibility also involves ensuring that watermarked texts perform well on downstream tasks. The introduction of a watermark may affect various downstream NLP tasks, including text classification, summarization, question answering, and so on. Studies such as [93] have benchmarked watermarking methods across nine NLP tasks, analyzing the impact on different performance metrics. In particular, ref. [94] found that classification tasks suffer the most significant performance degradation when LLM-generated text is watermarked, likely due to changes in token distributions affecting feature extraction.

Interestingly, current watermarking methods tend to perform better in open-ended generation tasks, such as long-form question answering, where slight lexical variations do not drastically alter the output's usefulness. This aligns with findings from [95], which highlight the inherent trade-off between watermark robustness, imperceptibility, and downstream utility.

5.3. Capacity

Watermark capacity refers to the amount of information that can be embedded within a given text while maintaining robustness and imperceptibility. Formally, capacity can be expressed by Equation (15), where B is the total number of embedded bits and L is the length of the generated text in tokens:

$$\text{Capacity} = \frac{B}{L} \tag{15}$$

In LLM watermarking, capacity is a crucial consideration, as higher-capacity schemes enable richer information encoding but often introduce greater risks of distortion and detection failure. Existing watermarking techniques can broadly be classified into single-bit and multi-bit watermarking, with the majority of prior work focusing on the former due to its simplicity and effectiveness.

5.3.1. Single-Bit Watermarking

Single-bit watermarking schemes encode a binary signal—typically indicating the mere presence or absence of a watermark—within the text. This approach is widely adopted due to its relative simplicity and ease of implementation. By embedding just one bit of information, these methods minimize the risk of perceptible distortions in the generated text while still offering a viable means of verification and traceability. Representative works in this area include studies such as [33,36,45], which have demonstrated that even a minimalistic embedding strategy can effectively signal ownership or authenticity without imposing significant computational or design overhead.

5.3.2. Multi-Bit Watermarking

In contrast, multi-bit watermarking aims to embed a sequence of bits into the text, enabling the encoding of richer information—such as copyright data, identifiers, or even secret messages. While this increased capacity can be highly desirable, it comes with substantial challenges. Embedding multiple bits requires a careful balance to avoid degrading the text's fluency and coherence, as even minor disruptions can be more easily noticed when larger amounts of information are encoded. Moreover, ensuring that multi-bit watermarks remain robust against common perturbations or adversarial attacks is significantly more complex. Numerous studies, such as [46,48,58,61,66,67,70,71,74,75,96], have investigated techniques

to increase watermarking capacity while minimizing negative impacts on text quality and robustness. These methods typically extend existing single-bit watermarking approaches, introducing straightforward enhancements to balance imperceptibility and detectability. Despite these advances, watermarking remains a dynamic research field, with ongoing efforts aimed at refining the effectiveness of these techniques across diverse applications.

6. Challenges and Future Works

Despite significant progress in watermarking techniques for LLMs, several critical challenges remain. These challenges hinder the widespread adoption of watermarking methods and call for further research to enhance robustness, security, capacity, and generalization. In addition, the absence of standardized benchmarks and evaluation protocols makes it difficult to fairly compare existing approaches, assess trade-offs, or determine their suitability for specific deployment scenarios. Addressing these issues is crucial not only for improving current techniques but also for guiding future research toward more robust and practically deployable watermarking systems. In this section, we outline key obstacles and potential future directions.

6.1. Challenges

The challenges in LLM watermarking can be broadly categorized into two dimensions. The first involves inherent limitations of current techniques, such as trade-offs between robustness, imperceptibility, and capacity, as well as vulnerabilities to adversarial attacks and generalization issues across tasks. The second relates to the absence of standardized benchmarks and evaluation protocols, which significantly hinders the ability to perform fair comparisons and assess method suitability. Both aspects pose major obstacles to the practical deployment and systematic advancement of LLM watermarking research.

6.1.1. Technical and Practical Constraints in LLM Watermarking

A primary challenge is the robustness of watermarking schemes. Many state-of-the-art methods are vulnerable to adversarial modifications. For example, recursive paraphrasing attacks [87] systematically paraphrase text while preserving its semantic content, thereby reducing watermark detection rates. Similarly, masked text filling techniques [86], which iteratively mask and regenerate tokens, can effectively obliterate watermark signals. These vulnerabilities indicate that more adaptive and resilient embedding strategies are urgently needed. In addition, targeted synonym substitution can further diminish watermark detectability by selectively replacing words with contextually appropriate alternatives while maintaining fluency and coherence [97,98].

System security is another critical concern. Watermarking systems are exposed to multiple threats, including spoof attacks, false positives, and key theft. Spoof attacks, where malicious actors manipulate watermarked content to evade detection or falsely assert ownership, have been demonstrated in prior studies [99,100]. Training-free approaches are particularly susceptible to key extraction [101–103], and methods that induce biases in text generation [104] risk unintended information leakage.

Capacity constraints limit current watermarking approaches, as most schemes embed only a single-bit watermark per text sample. While multi-bit watermarking has been theoretically explored, practical implementations remain scarce. The primary challenge lies in encoding richer information without compromising text fluency or introducing noticeable artifacts. Many multi-bit methods extend single-bit techniques, particularly in training-free approaches, by assuming that the text already contains a watermark. This assumption enables binary classification but is often overly restrictive in practice. Additionally, existing

multi-bit methods frequently rely on heuristic strategies for partitioning information bits, often neglecting the text's underlying structure and its potential impact on quality.

Finally, the performance of watermarking methods often varies across different tasks and domains. While they have shown promise in open-ended question answering, their efficacy in structured applications such as summarization, translation, and information retrieval is less established. This lack of generalization calls for more adaptable techniques that can maintain consistent performance across diverse NLP applications and evolving model architectures.

6.1.2. Lack of Standardized Benchmarks

A persistent challenge in LLM watermarking is the absence of standardized benchmarks and evaluation protocols. Existing studies often rely on ad hoc task selections, model settings, and attack implementations, making it difficult to reproduce results or compare methods fairly. While early evaluations predominantly focused on text completion tasks, watermark performance can vary significantly across different application contexts—such as dialogue, summarization, question answering, or code generation—due to the diverse structural and semantic characteristics of generated outputs. For example, techniques effective for free-form text may degrade when applied to structured code, and vice versa.

Recent efforts have attempted to address this issue by introducing broader benchmark suites. Piet et al. [92] incorporate both natural language and code generation tasks to assess watermark resilience in mixed domains. Wu et al. [105] target code generation specifically, using datasets drawn from real-world repositories and programming challenges. Tu et al. [93] introduce a task taxonomy based on input–output length variations, covering multiple NLP applications such as QA and summarization. However, despite these advancements, there is still no unified benchmark widely adopted by the research community. These efforts differ in task coverage, evaluation criteria, and attack assumptions, limiting their generalizability and the comparability of published results. As a result, the development of standardized, comprehensive, and reproducible benchmarking protocols remains an open and pressing challenge in the field.

6.2. Future Works

Advancing watermarking for LLMs will require sustained efforts to improve robustness, security, and capacity while broadening applicability. Future research should investigate dynamic watermarking techniques that adapt to text modifications and incorporate adversarial training to counter paraphrasing and token masking. In terms of security, there is a need to develop schemes that resist spoof attacks and minimize false positives, potentially through integrated cryptographic safeguards that prevent key extraction and unauthorized watermark generation.

Enhancing watermark capacity will involve exploring new encoding strategies that allow for multi-bit embedding without compromising readability. Simultaneously, research should aim to develop task-agnostic watermarking methods that perform reliably across various NLP tasks and are compatible with emerging LLM architectures, including multimodal systems and decentralized frameworks.

Beyond academic research, watermarking techniques are gaining traction in industry. OpenAI has explored watermarking methods specifically for large language models, aiming to support content provenance and misuse detection [35]. Google researchers have also developed and tested robust watermarking approaches, reporting internal use in real-world applications to identify AI-generated content [36]. These efforts signal a shift from theory to practice, as watermarking becomes a viable tool for ensuring trust, attribution, and integrity in deployed AI systems.

To support the practical deployment of watermarking systems, progress in the field also depends on developing standardized benchmarks and evaluation protocols. While recent efforts [92,93,105] have made meaningful strides, the lack of a unified evaluation framework still limits reproducibility and fair comparison across methods. Future work should focus on building comprehensive benchmark suites that span diverse tasks, domains, attack types, and evaluation metrics. Clear evaluation standards will not only enable more rigorous and reproducible comparisons but also create opportunities to explore how watermarking interacts with broader model behaviors, including its effects on fairness, bias, and ethical concerns.

7. Conclusions

The rapid evolution of LLMs has introduced significant concerns related to intellectual property protection, model attribution, and the prevention of content misuse. Digital watermarking has emerged as a promising technique to address these challenges, offering mechanisms to trace and authenticate model-generated outputs. This study has systematically examined existing watermarking techniques for LLMs, evaluating their effectiveness in terms of capacity, robustness, and imperceptibility. While current methods have demonstrated notable progress, challenges remain in enhancing resilience against adversarial attacks, balancing detectability with linguistic quality, and ensuring scalability across diverse deployment scenarios. Future research should focus on developing more robust watermarking schemes, standardizing evaluation metrics, and exploring the ethical and legal implications of watermarking in real-world applications. As LLMs continue to advance, watermarking will play a crucial role not only in protecting intellectual property but also in shaping the future of trustworthy, transparent, and responsible AI systems.

Author Contributions: Conceptualization, Z.Y.; methodology, Z.Y., G.Z. and H.W.; writing, Z.Y. and G.Z.; review and editing, Z.Y., G.Z. and H.W.; supervision, H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Science and Technology Commission of Shanghai Municipality (STCSM) under Grant Number 24ZR1424000, the National Natural Science Foundation of China (NSFC) under Grant Number U23B2023, the 2024 Xizang Autonomous Region Central Guided Local Science and Technology Development Fund Project under Grant Number XZ202401YD0015, and the Basic Research Program for Natural Science of Guizhou Province under Grant Number QIANKEHEJICHU-ZD[2025]-043.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
2. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186. [CrossRef]
3. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
4. Radford, A.; Narasimhan, K. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (accessed on 4 January 2025).

5. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020, Volume 33, pp. 1877–1901.
6. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv* **2020**, arXiv:2001.08361v1.
7. Chiang, W.L.; Zheng, L.; Sheng, Y.; Angelopoulos, A.N.; Li, T.; Li, D.; Zhu, B.; Zhang, H.; Jordan, M.I.; Gonzalez, J.E.; et al. Chatbot arena: An open platform for evaluating LLMs by human preference. In Proceedings of the 41st International Conference on Machine Learning. JMLR.org, 2024, ICML'24, Vienna, Austria, 21–27 July 2024.
8. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring Massive Multitask Language Understanding. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
9. Bengio, Y.; Mindermann, S.; Privitera, D.; Besiroglu, T.; Bommasani, R.; Casper, S.; Choi, Y.; Fox, P.; Garfinkel, B.; Goldfarb, D.; et al. International AI Safety Report. *arXiv* **2025**, arXiv:2501.17805.
10. van Schyndel, R.; Tirkel, A.; Osborne, C. A digital watermark. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 2, pp. 86–90. [CrossRef]
11. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 2066–2076.
12. Zhu, J.; Kaplan, R.; Johnson, J.; Li, F.-F. HiDDeN: Hiding Data With Deep Networks. In *Computer Vision–ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Swizterland, 2018; pp. 682–697.
13. Bassia, P.; Pitas, I.; Nikolaidis, N. Robust audio watermarking in the time domain. *IEEE Trans. Multimed.* **2001**, *3*, 232–241. [CrossRef]
14. Luo, X.; Li, Y.; Chang, H.; Liu, C.; Milanfar, P.; Yang, F. DVMark: A Deep Multiscale Framework for Video Watermarking. *IEEE Trans. Image Process.* **2023** . [CrossRef]
15. Brassil, J.; Low, S.; Maxemchuk, N.; O'Gorman, L. Electronic marking and identification techniques to discourage document copying. *IEEE J. Sel. Areas Commun.* **1995**, *13*, 1495–1504. [CrossRef]
16. Yang, T.; Wu, H.; Yi, B.; Feng, G.; Zhang, X. Semantic-Preserving Linguistic Steganography by Pivot Translation and Semantic-Aware Bins Coding. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 139–152. [CrossRef]
17. Uchida, Y.; Nagai, Y.; Sakazawa, S.; Satoh, S.I. Embedding Watermarks into Deep Neural Networks. In Proceedings of the 2017 ACM International Conference on Multimedia Retrieval, Bucharest, Romania, 6–9 June 2017; pp. 269–277. [CrossRef]
18. Wu, H.; Liu, G.; Yao, Y.; Zhang, X. Watermarking Neural Networks with Watermarked Images. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2591–2601. [CrossRef]
19. Rizzo, S.G.; Bertini, F.; Montesi, D. Content-preserving Text Watermarking through Unicode Homoglyph Substitution. In Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS '16, Montreal, Canada, 11–13 July 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 97–104. [CrossRef]
20. Zhou, W.; Ge, T.; Xu, K.; Wei, F.; Zhou, M. BERT-based Lexical Substitution. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 3368–3373. [CrossRef]
21. Yang, X.; Zhang, J.; Chen, K.; Zhang, W.; Ma, Z.; Wang, F.; Yu, N. Tracing text provenance via context-aware lexical substitution. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 11613–11621.
22. Atallah, M.J.; Raskin, V.; Crogan, M.; Hempelmann, C.; Kerschbaum, F.; Mohamed, D.; Naik, S. Natural Language Watermarking: Design, Analysis, and a Proof-of-Concept Implementation. In *Information Hiding*; Moskowitz, I.S., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 185–200.
23. Abdelnabi, S.; Fritz, M. Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 121–140. [CrossRef]
24. Yang, Z.; Wu, H. A Fingerprint for Large Language Models. *arXiv* **2024**, arXiv:2407.01235.
25. Zhang, J.; Liu, D.; Qian, C.; Zhang, L.; Liu, Y.; Qiao, Y.; Shao, J. REEF: Representation Encoding Fingerprints for Large Language Models. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
26. Setiadi, D.R.I.M.; Ghosal, S.K.; Sahu, A.K. AI-Powered Steganography: Advances in Image, Linguistic, and 3D Mesh Data Hiding—A Survey. *J. Future Artif. Intell. Technol.* **2025**, *2*, 1–23. [CrossRef]
27. Wu, H.; Yang, T.; Zheng, X.; Fang, Y., Linguistic Steganography and Linguistic Steganalysis. In *Adversarial Multimedia Forensics*; Nowroozi, E., Kallas, K., Jolfaei, A., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 163–190. [CrossRef]
28. Liu, A.; Pan, L.; Lu, Y.; Li, J.; Hu, X.; Zhang, X.; Wen, L.; King, I.; Xiong, H.; Yu, P. A Survey of Text Watermarking in the Era of Large Language Models. *ACM Comput. Surv.* **2024**, *57*, 47. [CrossRef]

29. Lalai, H.N.; Ramakrishnan, A.A.; Shah, R.S.; Lee, D. From Intentions to Techniques: A Comprehensive Taxonomy and Challenges in Text Watermarking for Large Language Models. *arXiv* **2024**, arXiv:2406.11106.
30. Zhao, X.; Gunn, S.; Christ, M.; Fairoze, J.; Fabrega, A.; Carlini, N.; Garg, S.; Hong, S.; Nasr, M.; Tramèr, F.; et al. SoK: Watermarking for AI-Generated Content. *arXiv* **2024**, arXiv:2411.18479.
31. Liang, Y.; Xiao, J.; Gan, W.; Yu, P.S. Watermarking Techniques for Large Language Models: A Survey. *arXiv* **2024**, arXiv:2409.00089.
32. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
33. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; Goldstein, T. A Watermark for Large Language Models. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J., Eds.; PMLR: Birmingham, UK, 2023; Volume 202, pp. 17061–17084.
34. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Shu, M.; Saifullah, K.; Kong, K.; Fernando, K.; Saha, A.; Goldblum, M.; Goldstein, T. On the Reliability of Watermarks for Large Language Models. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
35. Aaronson, S.; Kirchner, H. Watermarking gpt Outputs. 2023. Available online: https://www.scottaaronson.com/talks/watermark.ppt (accessed on 4 January 2025).
36. Dathathri, S.; See, A.; Ghaisas, S.; Huang, P.S.; McAdam, R.; Welbl, J.; Bachani, V.; Kaskasoli, A.; Stanforth, R.; Matejovicova, T.; et al. Scalable watermarking for identifying large language model outputs. *Nature* **2024**, *634*, 818–823. [CrossRef]
37. Zhang, R.; Hussain, S.S.; Neekhara, P.; Koushanfar, F. REMARK-LLM: A Robust and Efficient Watermarking Framework for Generative Large Language Models. In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), Philadelphia, PA, USA, 14–16 August 2024; USENIX Association: Berkeley, CA, USA, 2024; pp. 1813–1830.
38. Yang, B.; Li, W.; Xiang, L.; Li, B. SrcMarker: Dual-Channel Source Code Watermarking via Scalable Code Transformations. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024; pp. 4088–4106. [CrossRef]
39. Xu, X.; Yao, Y.; Liu, Y. Learning to Watermark LLM-generated Text via Reinforcement Learning. *arXiv* **2024**, arXiv:2403.10553.
40. Pang, K.; Qi, T.; Wu, C.; Bai, M.; Jiang, M.; Huang, Y. ModelShield: Adaptive and Robust Watermark against Model Extraction Attack. *IEEE Trans. Inf. Forensics Secur.* **2024**, *20* , 1767–1782. [CrossRef]
41. Bahri, D.; Wieting, J.M.; Alon, D.; Metzler, D. A Watermark for Black-Box Language Models. *arXiv* **2024**, arXiv:2410.02099.
42. Baldassini, F.B.; Nguyen, H.H.; Chang, C.C.; Echizen, I. Cross-Attention watermarking of Large Language Models. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 4625–4629. [CrossRef]
43. Boroujeny, M.K.; Jiang, Y.; Zeng, K.; Mark, B.L. Multi-Bit Distortion-Free Watermarking for Large Language Models. *arXiv* **2024**, arXiv:2402.16578.
44. Chen, L.; Bian, Y.; Deng, Y.; Cai, D.; Li, S.; Zhao, P.; Wong, K.F. WatME: Towards Lossless Watermarking Through Lexical Redundancy. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 9166–9180. [CrossRef]
45. Christ, M.; Gunn, S.; Zamir, O. Undetectable Watermarks for Language Models. In Proceedings of Thirty Seventh Conference on Learning Theory, Edmonton, AB, Canada, 30 June–3 July 2024; Agrawal, S., Roth, A., Eds.; PMLR: Birmingham, UK, 2024; Volume 247 , pp. 1125–1139.
46. Cohen, A.; Hoover, A.; Schoenbach, G. Watermarking Language Models for Many Adaptive Users. In Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 12–15 May 2025; IEEE Computer Society: Los Alamitos, CA, USA, 2025; pp. 84–84. . [CrossRef]
47. Feng, X.; Liu, J.; Ren, K.; Chen, C. A Certified Robust Watermark For Large Language Models. *arXiv* **2024**, arXiv:2409.19708.
48. Fernandez, P.; Chaffin, A.; Tit, K.; Chappelier, V.; Furon, T. Three Bricks to Consolidate Watermarks for Large Language Models. In Proceedings of the 2023 IEEE International Workshop on Information Forensics and Security (WIFS), Nürnberg, Germany, 4–7 December 2023; pp. 1–6. [CrossRef]
49. Fu, Y.; Xiong, D.; Dong, Y. Watermarking Conditional Text Generation for AI Detection: Unveiling Challenges and a Semantic-Aware Watermark Remedy. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 18003–18011. [CrossRef]
50. Giboulot, E.; Furon, T. WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024.
51. Golowich, N.; Moitra, A. Edit Distance Robust Watermarks via Indexing Pseudorandom Codes. In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024.

52. Guo, Y.; Tian, Z.; Song, Y.; Liu, T.; Ding, L.; Li, D. Context-aware Watermark with Semantic Balanced Green-red Lists for Large Language Models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 22633–22646. [CrossRef]
53. He, Z.; Zhou, B.; Hao, H.; Liu, A.; Wang, X.; Tu, Z.; Zhang, Z.; Wang, R. Can Watermarks Survive Translation? On the Cross-lingual Consistency of Text Watermark for Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 4115–4129.. [CrossRef]
54. Hoang, D.C.; Le, H.T.; Chu, R.; Li, P.; Zhao, W.; Lao, Y.; Doan, K.D. Less is More: Sparse Watermarking in LLMs with Enhanced Text Quality. *arXiv* **2024**, arXiv:2407.13803.
55. Hou, A.; Zhang, J.; He, T.; Wang, Y.; Chuang, Y.S.; Wang, H.; Shen, L.; Van Durme, B.; Khashabi, D.; Tsvetkov, Y. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 16–21 June 2024; Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 4067–4082. [CrossRef]
56. Hou, A.; Zhang, J.; Wang, Y.; Khashabi, D.; He, T. k-SemStamp: A Clustering-Based Semantic Watermark for Detection of Machine-Generated Text. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 1706–1715. [CrossRef]
57. Huo, M.; Somayajula, S.A.; Liang, Y.; Zhang, R.; Koushanfar, F.; Xie, P. Token-Specific Watermarking with Enhanced Detectability and Semantic Coherence for Large Language Models. In Proceedings of the ICML, Vienna, Austria, 21–27 July 2024.
58. Jiang, H.; Wang, X.; Yi, P.; Lei, S.; Lin, Y. CredID: Credible Multi-Bit Watermark for Large Language Models Identification. *arXiv* **2024**, arXiv:2412.03107.
59. Kuditipudi, R.; Thickstun, J.; Hashimoto, T.; Liang, P. Robust Distortion-free Watermarks for Language Models. *Trans. Mach. Learn. Res.* **2024** .
60. Li, Y.; Wang, Y.; Shi, Z.; Hsieh, C.J. Improving the Generation Quality of Watermarked Large Language Models via Word Importance Scoring. *arXiv* **2023**, arXiv:2311.09668.
61. Li, L.; Bai, Y.; Cheng, M. Where Am I From? Identifying Origin of LLM-generated Content. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 12218–12229. [CrossRef]
62. Li, X.; Ruan, F.; Wang, H.; Long, Q.; Su, W.J. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *Ann. Stat.* **2025**, *53*, 322–351. [CrossRef]
63. Liu, Y.; Bu, Y. Adaptive Text Watermark for Large Language Models. In Proceedings of the Forty-First International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
64. Liu, A.; Pan, L.; Hu, X.; Li, S.; Wen, L.; King, I.; Yu, P.S. An Unforgeable Publicly Verifiable Watermark for Large Language Models. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
65. Niess, G.; Kern, R. Stylometric Watermarks for Large Language Models. *arXiv* **2024**, arXiv:2405.08400.
66. Pang, Q.; Hu, S.; Zheng, W.; Smith, V. No Free Lunch in LLM Watermarking: Trade-offs in Watermarking Design Choices. In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024.
67. Qu, W.; Yin, D.; He, Z.; Zou, W.; Tao, T.; Jia, J.; Zhang, J. Provably Robust Multi-bit Watermarking for AI-generated Text via Error Correction Code. *arXiv* **2024**, arXiv:2401.16820.
68. Ren, J.; Xu, H.; Liu, Y.; Cui, Y.; Wang, S.; Yin, D.; Tang, J. A Robust Semantics-based Watermark for Large Language Model against Paraphrasing. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, 16–21 June 2024; Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 613–625. [CrossRef]
69. Ren, Y.; Guo, P.; Cao, Y.; Ma, W. Subtle Signatures, Strong Shields: Advancing Robust and Imperceptible Watermarking in Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 5508–5519. [CrossRef]
70. Song, M.; Li, Z.; Liu, K.; Peng, M.; Tian, G. NLWM: A Robust, Efficient and High-Quality Watermark for Large Language Models. In *Web Information Systems Engineering–WISE 2024*; Barhamgi, M., Wang, H., Wang, X., Eds.; Springer Nature: Singapore, 2025; pp. 320–335.

71. Wang, L.; Yang, W.; Chen, D.; Zhou, H.; Lin, Y.; Meng, F.; Zhou, J.; Sun, X. Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
72. Wu, Y.; Hu, Z.; Guo, J.; Zhang, H.; Huang, H. A Resilient and Accessible Distribution-Preserving Watermark for Large Language Models. In Proceedings of the Forty-First International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
73. Xu, X.; Jia, J.; Yao, Y.; Liu, Y.; Li, H. Robust Multi-bit Text Watermark with LLM-based Paraphrasers. *arXiv* **2024**, arXiv:2412.03123.
74. Yoo, K.; Ahn, W.; Kwak, N. Advancing Beyond Identification: Multi-bit Watermark for Large Language Models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 16–21 June 2024; Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 4031–4055. [CrossRef]
75. Zamir, O. Undetectable Steganography for Language Models. *Trans. Mach. Learn. Res.* **2024** .
76. Zhao, X.; Ananth, P.V.; Li, L.; Wang, Y.X. Provable Robust Watermarking for AI-Generated Text. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
77. Zhao, X.; Liao, C.; Wang, Y.X.; Li, L. Efficiently Identifying Watermarked Segments in Mixed-Source Texts. In *Neurips Safe Generative AI Workshop 2024*; NeurIPS: San Diego CA, USA, 2024.
78. Zhong, X.; Dasgupta, A.; Tanvir, A. Watermarking Language Models through Language Models. *arXiv* **2024**, arXiv:2411.05091.
79. Zhou, T.; Zhao, X.; Xu, X.; Ren, S. Bileve: Securing Text Provenance in Large Language Models Against Spoofing with Bi-level Signature. In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024.
80. Fairoze, J.; Garg, S.; Jha, S.; Mahloujifar, S.; Mahmoody, M.; Wang, M. Publicly Detectable Watermarking for Language Models. *arXiv* **2023**, arXiv:2310.18491. [CrossRef]
81. Hu, Z.; Chen, L.; Wu, X.; Wu, Y.; Zhang, H.; Huang, H. Unbiased Watermark for Large Language Models. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
82. Zhang, J.; Chen, D.; Liao, J.; Zhang, W.; Feng, H.; Hua, G.; Yu, N. Deep Model Intellectual Property Protection via Deep Watermarking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 4005–4020. [CrossRef]
83. Contributors, F.; El-Kishky, A.; Selsam, D.; Song, F.; Parascandolo, G.; Ren, H.; Lightman, H.; Won, H.; Akkaya, I.; Sutskever, I.; et al. OpenAI o1 System Card. *arXiv* **2024**, arXiv:2412.16720.
84. DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.M.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2501.12948.
85. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. In Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, New Orleans, LA, USA, 28 November–9 December 2022; Curran Associates Inc.: Red Hook, NY, USA, 2022.
86. Zhang, H.; Edelman, B.L.; Francati, D.; Venturi, D.; Ateniese, G.; Barak, B. Watermarks in the Sand: Impossibility of Strong Watermarking for Language Models. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
87. Sadasivan, V.S.; Kumar, A.; Balasubramanian, S.; Wang, W.; Feizi, S. Can AI-Generated Text be Reliably Detected? *arXiv* **2023**, arXiv:2303.11156.
88. Luo, Y.; Lin, K.; Gu, C. Lost in Overlap: Exploring Watermark Collision in LLMs. *arXiv* **2024**, arXiv:2403.10020.
89. Diaa, A.; Aremu, T.; Lukas, N. Optimizing Adaptive Attacks against Content Watermarks for Language Models. *arXiv* **2024**, arXiv:2410.02440.
90. Ayoobi, N.; Knab, L.; Cheng, W.; Pantoja, D.; Alikhani, H.; Flamant, S.; Kim, J.; Mukherjee, A. ESPERANTO: Evaluating Synthesized Phrases to Enhance Robustness in AI Detection for Text Origination. *arXiv* **2024**, arXiv:2409.14285.
91. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
92. Piet, J.; Sitawarin, C.; Fang, V.; Mu, N.; Wagner, D. Mark My Words: Analyzing and Evaluating Language Model Watermarks. *arXiv* **2023**, arXiv:2312.00273.
93. Tu, S.; Sun, Y.; Bai, Y.; Yu, J.; Hou, L.; Li, J. WaterBench: Towards Holistic Evaluation of Watermarks for Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 1517–1542. [CrossRef]
94. Ajith, A.; Singh, S.; Pruthi, D. Downstream Trade-offs of a Family of Text Watermarks. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 14039–14053. [CrossRef]

95. Molenda, P.; Liusie, A.; Gales, M. WaterJudge: Quality-Detection Trade-off when Watermarking Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico, 16–21 June 2024; Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 3515–3525. [CrossRef]
96. Guan, B.; Wan, Y.; Bi, Z.; Wang, Z.; Zhang, H.; Zhou, P.; Sun, L. CodeIP: A Grammar-Guided Multi-Bit Watermark for Large Language Models of Code. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 9243–9258. [CrossRef]
97. Wu, Q.; Chandrasekaran, V. Bypassing LLM Watermarks with Color-Aware Substitutions. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 8549–8581. [CrossRef]
98. Creo, A.; Pudasaini, S. SilverSpeak: Evading AI-Generated Text Detectors using Homoglyphs. *arXiv* **2024**, arXiv:2406.11239.
99. Pang, Q.; Hu, S.; Zheng, W.; Smith, V. Attacking LLM Watermarks by Exploiting Their Strengths. In Proceedings of the ICLR 2024 Workshop on Secure and Trustworthy Large Language Models, Vienna, Austria, 11 May 2024.
100. Jovanović, N.; Staab, R.; Vechev, M. Watermark stealing in large language models. In Proceedings of the 41st International Conference on Machine Learning. JMLR.org, 2024, ICML'24, Vienna, Austria, 21–27 July 2024.
101. Chang, H.; Hassani, H.; Shokri, R. Watermark Smoothing Attacks against Language Models. *arXiv* **2024**, arXiv:2407.14206.
102. Chen, R.; Wu, Y.; Guo, J.; Huang, H. De-mark: Watermark Removal in Large Language Models. *arXiv* **2024**, arXiv:2410.13808.
103. Rastogi, S.; Pruthi, D. Revisiting the Robustness of Watermarking to Paraphrasing Attacks. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y.N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 18100–18110. [CrossRef]
104. Liu, A.; Guan, S.; Liu, Y.; Pan, L.; Zhang, Y.; Fang, L.; Wen, L.; Yu, P.S.; Hu, X. Can Watermarked LLMs be Identified by Users via Crafted Prompts? In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
105. Wu, B.; Chen, K.; He, Y.; Chen, G.; Zhang, W.; Yu, N. CodeWMBench: An Automated Benchmark for Code Watermarking Evaluation. In Proceedings of the ACM Turing Award Celebration Conference—China 2024, ACM-TURC '24, Changsha, China, 5–7 July 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 120–125. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.