Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-26)
Survey Track

Implicit Identity Technologies for LLMs: Fingerprinting and Watermarking across Datasets, Models, and Generated Content

Bing Liu[1,2] , **Shunping Wang**[1] , **Yufan Zhu**[3,4] , **Xinyi Yu**[1] , **Jing Huang**[3,4] , **Linkang Du**[1] , **Hongbin Pei**[1,5*] , **Wei Luo**[6]
[1]School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China
[2]State Grid Henan Marketing Service Center, Henan, China
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[4]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[5]Shenzhen Loop Area Institute, Shenzhen, China
[6]School of Information Technology, Deakin University, Geelong, Australia

Abstract

Large language models (LLMs) require substantial investments and are increasingly deployed in high-stakes domains, making it critical to protect LLM-related assets and to trace their provenance. Identity technologies such as fingerprinting and watermarking address these needs by enabling ownership verification and attribution, and have rapidly emerged as an active research focus. However, existing techniques lack a systematic organisation, leading to two key issues, *terminological confusion* and *isolated research lines*, that have hindered the development of this research field. To this end, we present a comprehensive review of LLM identity techniques, focusing on fingerprinting and watermarking across the LLM lifecycle, including datasets, models, and generated content. We make three primary contributions. *First*, we introduce *implicit identity* (Implicit-ID for short) as a unifying abstraction and distinguish fingerprinting from watermarking. *Second*, we propose a lifecycle-based taxonomy that organises techniques by asset type and verification role, aligning each with asset protection or provenance. *Third*, we establish an evaluation framework around three objectives—identifiability, robustness, and deployability. Together, these contributions structure the landscape of LLM identity techniques, clarify terminology, and highlight directions toward secure deployment.

1 Introduction

Large language models (LLMs) have revolutionised a wide range of fields. Their development requires substantial investments in data, computation, and expertise; for example, training GPT-4 is reported to cost over $100 million[1] .

Consequently, *asset protection* of LLMs becomes critical given such high costs, as unauthorised use or leakage can result in significant losses. In parallel, the misuse and malicious exploitation of LLMs has emerged as another key challenge; for instance, LLM-generated content has been used by malicious actors to enable automated fraud systems [Lin *et al.*, 2024]. Accordingly, *provenance* of LLMs is critical for tracing the origins of such activities, thereby enabling accountability and governance of misuse and malicious exploitation.

From a technological perspective, identity mechanisms for LLMs are fundamental to enabling both asset protection and provenance. A growing body of research has explored LLM identity techniques, such as fingerprinting for model copyright protection [Zhang *et al.*, 2025] and watermarking for generated content provenance [Kirchenbauer *et al.*, 2023]. However, as the field remains at an early stage, existing technologies lack a systematic organisation, resulting in two key issues: (1) **Terminological confusion**. In particular, watermarking and fingerprinting, two primary identity technologies, are not clearly distinguished in the literature. For instance, the trigger-based specific-response mechanisms are termed "instructional fingerprinting" in [Xu *et al.*, 2024a], whereas they are referred to as "backdoor watermarking" in [Mo *et al.*, 2025]. (2) **Isolated research lines**. Existing identity studies for LLMs have largely evolved in isolation within specific scenarios, including dataset identity [Carlini *et al.*, 2021; Maini *et al.*, 2021], model identity [Zeng *et al.*, 2024; Zhang *et al.*, 2025; Xu *et al.*, 2024a], and generated content identity [Kirchenbauer *et al.*, 2023; Hou *et al.*, 2024a; Hu *et al.*, 2024]. As a result, the community lacks a comprehensive and unified understanding of LLM identity. Prior watermarking surveys [Liu *et al.*, 2024; Wang *et al.*, 2025b; Liang *et al.*, 2026] naturally do not cover fingerprinting, and even recent surveys covering both [Xu *et al.*, 2025d; Ye *et al.*, 2025; Wang *et al.*, 2025b; Liang *et al.*, 2026] fail to adequately address these two issues.

The two aforementioned issues have, to some extent, hindered the development of this research field. To address these challenges, this survey clarifies key concepts and proposes a verification-semantic taxonomy to systematically organise existing studies on LLM identity, including fingerprinting and watermarking. Our goal is to clarify the landscape of this nascent research field and to outline research objectives and

*Corresponding authors.
[1]https://en.wikipedia.org/wiki/GPT-4

major roadmaps to guide future advances. Therefore, this survey makes the following three contributions.

Our first contribution is a unification of key concepts in this field. We propose a three-layer conceptual framework to organise existing methods. (I) At the top level, we introduce a unifying abstraction, *Implicit Identity* (Implicit-ID for short), which refers to specific characteristics of an LLM that are not directly observable but can be verified and used to uniquely identify the model and distinguish it from others. Such implicit identity provides a unifying abstraction for both fingerprinting and watermarking techniques. (II) At the middle level, we differentiate between two main realisations of implicit-ID, fingerprinting and watermarking. LLM fingerprinting is defined as a *non-intrusive* implicit-ID, which is constructed by intrinsic characteristics of an LLM; LLM watermarking is an *intrusive* implicit-ID, in which identifiable signals are deliberately embedded into the model, data, or generated content through external intervention. (III) At the bottom level, we summarise existing methods by evidence source and verification pathway, yielding a compact design-space view of implicit-ID technologies.

Our second contribution is a generative taxonomy across assets and lifecycle stages. We introduce a taxonomy that organises identity technologies *by verification semantics*, i.e., similarity-based attribution and keyed verification, across datasets, models, and generated content throughout the LLM lifecycle. Beyond providing comprehensive coverage, the taxonomy reveals *cross-asset equivalences*. For instance, the shared trigger-response logic underlying dataset watermarking and fine-tuning-based model watermarking, and the shared attribution logic underlying behavioural model fingerprints and output-only fingerprints. This perspective enables *cross-level reasoning*: techniques designed for one asset level can inform or transfer to another, helping the community move beyond isolated research lines and supporting reasoning about multi-level identity designs for LLMs.

Our third contribution is the establishment of technological objectives. Motivated by the two core application tasks of Implicit-ID, asset protection and provenance, we identify three key objectives for the technologies: *identifiability*, *robustness*, and *deployability*. Specifically, an Implicit-ID should exhibit high identifiability to reliably distinguish different LLMs, maintain robustness under legitimate post-deployment transformations and adversarial manipulation, and demonstrate practical deployability under realistic settings. Accordingly, we propose a common evaluation template organised by transformation regimes, enabling systematic and comparable assessment of existing methods.

These contributions show that LLM fingerprinting and watermarking can be consistently viewed under a shared verification-semantic lens, rather than as isolated, mechanism-specific techniques, as organised in the prior surveys [Xu *et al.*, 2025d; Ye *et al.*, 2025].

What this survey enables. By unifying identity technologies under Implicit-ID and organizing them by verification semantics rather than mechanism, this survey enables: (i) principled analysis and comparison between intrusive and non-intrusive methods under shared verification semantics; (ii) transfer of techniques and evaluation stressors across dataset, model, and content identity; and (iii) systematic reasoning about trade-offs between legal defensibility, robustness to post-deployment transformations, and practical deployability.

The remainder is organised as follows: Section 2 formalises the tasks and notation, Section 3 presents the lifecycle-based taxonomy, Section 4 defines the evaluation metrics, and Section 5 discusses challenges and directions.

2 Main Tasks in Implicit-ID Technologies

In this section, we formalise two primary application tasks of Implicit-ID technologies: **Asset Protection**, which aims to safeguard the security and value of LLM-related assets, and **Provenance**, which supports accountability, traceability, and regulatory compliance within the LLM ecosystem.

Task formulation. We view identity verification over LLM assets as the foundation of both tasks. In general, identity verification can be formulated as a decision process over observable evidence extracted from an asset, where the verifier determines whether the asset should be attributed to a claimed owner, creator, or source. This process must control both false matches, where an unrelated asset is incorrectly accepted, and false misses, where a legitimate or derived asset is incorrectly rejected, under realistic access constraints. This formulation motivates our later focus on identifiability, robustness, and deployability. In asset protection, identity verification establishes legitimate ownership and prevents unauthorised use. In provenance, it identifies asset creators or contributors, thereby enabling accountability, traceability, and compliance.

Unified view and notation. Let $A \in \{\mathcal{D}, \mathcal{M}, \mathcal{Y}\}$ denote an LLM asset subject to protection or provenance verification, where $\mathcal{D}$ represents training data, $\mathcal{M}$ represents model artefacts such as weights and internal states, and $\mathcal{Y}$ represents generated content. A verifier accesses the asset through an interface O_A, which captures practical verification settings: white-box access, where model weights or full asset details are available; grey-box access, where only limited internal information is exposed; and black-box access, where verification is restricted to queries. After release or deployment, the asset may undergo a transformation $\tau \in \mathcal{T}$, which yields $A' = \tau(A)$. Examples include dataset filtering, model fine-tuning, quantisation, distillation, merging, and output rewriting. We evaluate an Implicit-ID technology by whether reliable verification can be maintained under τ of interest. We distinguish two transformation regimes that recur across all asset types. *Owner/benign transformations* $\tau \in \mathcal{T}_{\text{own}}$ correspond to legitimate post-release adaptations, such as filtering, fine-tuning, quantisation, distillation, prompt or policy modification, and paraphrasing. These transformations primarily evaluate whether an identity signal remains stable under normal lifecycle evolution. *Adversarial transformations* $\tau \in \mathcal{T}_{\text{adv}}$ represent adaptive attempts to remove, forge, or evade identity signals, including watermark removal through targeted training, paraphrase-based sanitisation, trigger suppression, and mimicry prompting.

2.1 Asset Protection Task

For LLM-related asset protection, Implicit-ID resides in or is embedded within assets and serves as verifiable evidence of ownership, thereby detecting and countering unauthorised use and redistribution, such as the well-known 2024 *Miqu* model leak on Hugging Face[2] . Generated content asset protection is also crucial for preventing plagiarism. The U.S. Copyright Office's rulings on creative arrangement in AI-assisted works establish an important legal foundation[3]

Asset protection can be achieved through proactive verification, i.e., watermarking. Specifically, the asset owner embeds a keyed identity signal into an asset (or its generation process) using an embedding function E_k, producing a protected asset $\widetilde{A} = E_k(A)$. After the asset undergoes a transformation $A' = \tau(\widetilde{A})$, ownership is verified through a keyed detector $V_k(\mathsf{O}_{A'})$. Asset protection can also be achieved through similarity-based attribution, i.e., fingerprinting, where no keyed mark is embedded. Instead, the verifier extracts inherent distinguishing evidence $X(\mathsf{O}_{A'})$ from the target asset and determines ownership by comparing it with reference evidence extracted from owned assets, through a similarity-based detector $V_x(\mathsf{O}_A, \mathsf{O}_{A'})$.

2.2 Provenance Task

For provenance, Implicit-ID acts as the foundation for tracing LLM-related asset lineage and origin, reinforcing digital trust and accountability. Dataset provenance enables retroactive auditing through Implicit-ID techniques such as dataset fingerprinting, allowing stakeholders to detect latent "data footprints" [Sadhukhan and Chakraborty, 2025] and verify licensing compliance without requiring pre-marked data. Model provenance focuses on identifying rebranded or derivative models and tracing them back to their foundational counterparts, supporting regulatory requirements for a transparent chain of custody. Generated content provenance usually serves as a societal safeguard against misinformation and synthetic fraud. Many jurisdictions have established mandatory labelling requirements for generated content, including China[4] , the EU[5] , and the United States[6] , thereby reinforcing digital trust through provenance.

In practice, provenance can be established either through *proactive provenance binding*, e.g., watermarking at creation, or through *retroactive provenance inference* similarity-based attribution when no prior mark exists. Provenance binding is a proactive realisation of provenance. In this mode, the asset owner embeds or attaches verifiable evidence using E_k, such as a keyed watermark, either at creation time or during service [Dathathri *et al.*, 2024]. The resulting asset carries an identity signal that can later be verified from a transformed asset A' using a keyed detector $V_k(\mathsf{O}_{A'})$. This mode provides strong, deliberate provenance evidence, but typically requires prior instrumentation of the asset or serving pipeline. Provenance inference is a retroactive realisation of provenance. When no prior mark is assumed, the verifier extracts inherent evidence $X(\mathsf{O}_{A'})$ from the observed asset and attributes it by comparing this evidence with reference evidence from candidate source assets through a matcher $V_x(\mathsf{O}_A, \mathsf{O}_{A'})$. This mode enables post hoc auditing, and its conclusions are generally probabilistic rather than cryptographically guaranteed.

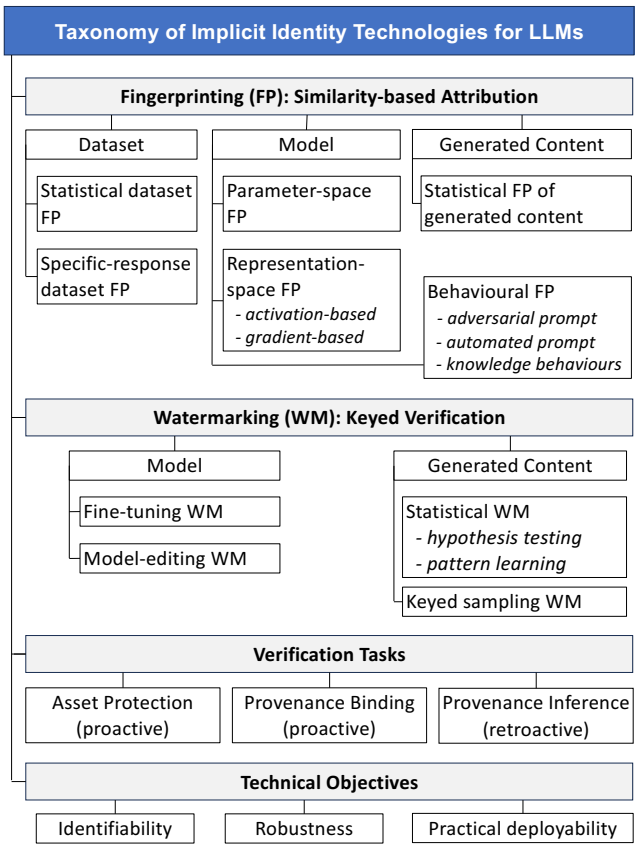

Figure 1: Taxonomy of Implicit-ID technologies for LLMs.

3 Taxonomy across LLM Lifecycle and Verification Semantics

We first introduce the taxonomy dimensions that structure the landscape of Implicit-ID technologies, including the LLM lifecycle and verification semantics, and then systematically survey existing techniques within this unified framework.

3.1 Taxonomy Dimensions

Our taxonomy organises Implicit-ID technologies along two complementary dimensions: a *verification dimension* defined by verification semantics, and a *lifecycle dimension* defined by the asset stage at which identity is established or verified.

Along the lifecycle dimension, we consider three asset stages throughout the LLM lifecycle: Dataset $\mathcal{D}$, Model $\mathcal{M}$, and Generated content $\mathcal{Y}$. Asset protection and provenance tasks recur as cross-cutting objectives across all stages.

Along the verification dimension, we distinguish two categories of verification semantics underlying both asset protec-

[2]https://huggingface.co/miqudev/miqu-1-70b
[3]https://www.copyright.gov/docs/zarya-of-the-dawn.pdf
[4]https://www.mps.gov.cn/n6557558/c8797736/content.html
[5]https://artificialintelligenceact.eu/article/50/
[6]https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240SB942

tion and provenance tasks. The first category is **similarity-based attribution**, implemented via a matcher $V_x(\cdot,\cdot)$, which corresponds to **non-intrusive fingerprinting**. The second category is **keyed verification** $V_k(\cdot)$, implemented through a keyed mark and a corresponding detector, which corresponds to **intrusive watermarking**. Consequently, both asset protection and provenance can be achieved either through keyed verification or through similarity-based attribution.

Specifically, across assets $A \in \mathcal{D}, \mathcal{M}, \mathcal{Y}$ and corresponding access interfaces O_A, we summarize existing methods in terms of: (i) **Evidence source** and required **Access conditions**; (ii) **Verification semantics** (keyed or similarity-based); and (iii) **Dominant transformation stressors** τ under which verification must remain reliable. The proposed taxonomy is *generative*: it reveals cross-asset commonalities in verification logic, suggesting how techniques, attack models, and evaluation stressors may transfer across datasets, models, and generated content. This perspective enables cross-level reasoning, where techniques developed for one asset stage can inform the design of methods at other stages.

3.2 Implicit-ID of Datasets

Implicit-ID of datasets concerns verifying whether a suspect model $\mathcal{M}'$ was trained on a target dataset $\mathcal{D}$ under post-training transformations τ, given access through an observation interface, most commonly a black-box API. Its primary task is *provenance inference*: providing post-hoc evidence of data usage when training provenance is opaque, thereby supporting auditing and compliance; by establishing data ownership, it also indirectly contributes to asset protection.

Although dataset watermarking has been studied in traditional machine learning, particularly for image datasets [Du *et al.*, 2025], emerging LLM-oriented approaches embed in-distribution, text-preserving signals into documents, such as invisible Unicode canaries or cue–reply structures, and later verify their imprint through black-box prompting [Li *et al.*, 2025b; Naser, 2025]. Since these methods typically rely on special triggers that induce recognisable model behaviour after training, we categorise them as fine-tuning model watermarking and discuss them in Section 3.3. This subsection focuses on dataset fingerprinting for provenance inference, where the verifier does not assume a pre-embedded mark.

Dataset fingerprint
- **Evidence:** loss/likelihood statistics.
- **Access:** black-box or grey-box.
- **Verification:** statistical memorisation signal detection, targeted behavioural signature probing.
- **Stressors:** shift, fine-tuning, distillation.

Dataset fingerprinting can be broadly divided into two paradigms: **statistical fingerprinting**, which relies on aggregate memorization signals, and **specific-response fingerprinting**, which examines targeted behavioural signatures.

Statistical Dataset Fingerprinting

Statistical dataset fingerprinting provides non-keyed evidence that $\mathcal{M}'$ has been exposed to $\mathcal{D}$ by identifying dataset-specific *memorization signals* in model statistics, including loss or perplexity gaps, likelihood anomalies, and aggregate membership scores, while leaving $\mathcal{D}$ unmodified.

Representative lines of work show that LLMs can exhibit detectable membership effects at the sequence or aggregate level, enabling post-hoc auditing and ownership resolution [Carlini *et al.*, 2021; Mattern *et al.*, 2023; Shi *et al.*, 2024; Maini *et al.*, 2024]. In practice, these signals are most informative when evaluated under confound controls (e.g., semantically similar corpora) and robustness stressors τ (fine-tuning, distillation, and distribution shift), which can dilute or mask membership traces [Huang *et al.*, 2024].

Specific-Response Dataset Fingerprinting

Specific-response fingerprinting constructs non-keyed evidence by probing $\mathcal{M}'$ with a curated set of membership-sensitive queries and analysing *targeted behavioural signatures* rather than global statistics. Typical signals include confidence/margin patterns on high-influence examples or specialised probe prompts that elicit distinctive responses linked to $\mathcal{D}$ [Maini *et al.*, 2021; Liu *et al.*, 2022; Naser, 2025].

Compared with purely statistical approaches, this paradigm can offer stronger diagnostic power for ownership resolution, but is more sensitive to probe design and can be confounded by *independent similarity* (training on a dataset $\mathcal{D}' \approx \mathcal{D}$). It is also vulnerable to post-training transformations τ (e.g., instruction tuning) and adaptive smoothing that reduce membership contrast.

3.3 Implicit-ID of Models

Implicit-ID of models concerns determining whether a suspect model $\mathcal{M}'$ is derived from an owner model $\mathcal{M}$ under transformations $\tau \in \mathcal{T}$ and an observation interface, which may provide white-, grey-, or black-box access. Such identity verification supports both asset protection and provenance. Here, we focus on model-level identity, covering *fingerprinting as similarity-based attribution* and *watermarking as keyed ownership verification*. We defer output provenance techniques to Section 3.4.

Model Fingerprinting

Model fingerprinting performs non-keyed, similarity-based attribution by matching intrinsic model evidence, such as parameters, representations, or elicited behaviors.

Existing works form fingerprints by leveraging model parameters, internal representations, and externally observable behaviours. Accordingly, existing methods fall into three categories: **parameter-space fingerprints**, which require white-box access; **representation-space fingerprints**, which typically require white- or grey-box access, with occasional query-only approximations; and **behavioural fingerprints**, which rely on curated elicitation prompts and are compatible with black-box access.

Model parameter-space fingerprint
- **Evidence:** weights/parameters statistics/invariants.
- **Access:** white-box.
- **Verification:** similarity-based attribution.
- **Stressors:** weight permutation/reordering, mild fine-tuning, typically weak under strong τ (e.g., distillation or merging) unless explicitly handled.

Model parameter-space fingerprinting identifies a model by extracting descriptors directly from weights, then attributing ownership via similarity in a weight-derived feature space. One representative design is to build invariants that remain stable under admissible weight reordering/permutation (e.g., HuRef [Zeng *et al.*, 2024]), enabling lineage/copyright inference without embedding any trigger. Another design [Yoon *et al.*, 2025] uses layerwise attention-projection statistics (Q/K/V/out) and compares correlations across statistical sequences to infer lineage (Intrinsic Fingerprint of LLMs). A third line of work [Zeng *et al.*, 2025] aligns informative weight blocks (e.g., embeddings and Q/K projections) and computes similarity after alignment (AWM), reported as discriminative even after large-scale training. In exchange for strong identifiability signals, parameter-space approaches typically require white-box access and may be costly or less practical for ultra-large models.

Model representation-space fingerprint
- **Evidence:** hidden activations/gradients under probing.
- **Access:** white-box, grey-box, black-box occasionally.
- **Verification:** similarity-based attribution.
- **Stressors:** probe design and distribution shift, fine-tuning, merging, distillation.

Model representation-space fingerprinting attributes a suspect model by comparing its internal responses to predefined probing inputs. These responses include activations, gradients, or sensitivity signals, which are converted into descriptors and matched against those of reference models.

Existing approaches can be organised into two canonical designs. The first is *activation-based fingerprints*, which identify models by comparing intermediate activations induced by probing inputs: DEEPJUDGE [Chen *et al.*, 2022] evaluates similarity via multi-level distance metrics on intermediate activations; EASYDETECTOR [Zhang *et al.*, 2024] trains probing classifiers on a victim model's activations and tests whether separability patterns transfer to a suspect model; and REEF [Zhang *et al.*, 2025] compares activation representations using geometry-preserving similarity (CKA). The second is *gradient/sensitivity-based fingerprints*, which use how the model responds to small perturbations: TENSORGUARD [Wu *et al.*, 2025] extracts fingerprints from statistics of parameter gradients induced by controlled perturbations in selected layers; and ZEROPRINT [Shao *et al.*, 2025b] derives global Jacobian-based fingerprints by aggregating output differences under probing inputs, enabling identification without access to internal states. In general, representation-space fingerprints can be robust to parameter modifications. However, most existing methods require white-box access and may struggle to distinguish closely related LLMs with similar architectures.

Model behaviour fingerprinting attributes a suspect model by eliciting stable and discriminative response patterns from a compact set of curated prompts, followed by non-keyed matching over the observed outputs through similarity scoring or learned linkage. A representative instantiation is LLMMAP [Pasquini *et al.*, 2025], which actively issues a small set of queries to fingerprint LLM-integrated systems, such as RAG pipelines, and identify specific model versions from subtle behavioural nuances.

Model behaviour fingerprint
- **Evidence:** input-output response patterns under curated elicitation prompts, such as fixed-string responses, response distributions, and reasoning traces.
- **Access:** black-box.
- **Verification:** similarity-based attribution, learned matching over responses.
- **Stressors:** decoding randomness (temperature/top-p), refusal/safety policies, prompt filtering/sanitisation, system-prompt differences, context-window effects, adaptive prompt interventions, model drift.

Behavioural fingerprinting can be grouped into three technical paradigms. The first is *adversarial/optimised prompting*, which aims to induce deterministic fixed-string or high-entropy signatures that separate the target model from non-homologous models. TRAP [Gubri *et al.*, 2024] repurposes adversarial suffixes to induce a predefined answer from the target model, while other models produce incoherent or divergent outputs. PROFLINGO [Jin *et al.*, 2024] generates prompts that trigger distinctive response distributions to link original models and their derivatives. The second is *automated prompt generation*, which reduces reliance on manually crafted fingerprints. Hide-and-Seek [Iourovitski *et al.*, 2024] uses evolutionary learning with an auditor LLM to discover salient input–output pairs exposing a model family's semantic manifold; RAP-SM [Xu *et al.*, 2025c] further leverages shadow models to improve fingerprint transferability across model series and maintain robustness under adaptive transformations such as model merging and fine-tuning. The third is based on *high-level cognitive and knowledge-level behaviour*, where reasoning and retrieval idiosyncrasies serve as fingerprints. COTSRF[Ren *et al.*, 2025] models distinctive chain-of-thought trajectories elicited by crafted logic prompts, whereas DUFFIN[Yan *et al.*, 2025] combines trigger patterns with knowledge-level fingerprints that capture how models retrieve and articulate domain-specific facts. Additionally, some behavioural identifiers exploit architecture-tied effects. For example, ROUTEMARK [He *et al.*, 2025] attributes identity in MoE-based merged models through distinctive expert-routing logic. Although such evidence is observed behaviourally, it reflects internal architectural characteristics and therefore lies at the boundary between behaviour and representation-space fingerprinting.

Taken together, these methods show that input-conditioned responses—whether triggered by adversarial suffixes, reasoning paths, or routing preferences—constitute a powerful and diverse toolkit for active model fingerprinting.

Model Watermarking

Model watermarking refers to keyed ownership verification: a class of intrusive identity technologies that embed identity signals into a model before release, allowing ownership to be verified later by probing for those signals. Most existing methods follow a *specific-response* paradigm. The model is intentionally modified to produce distinctive responses to designated secret triggers while preserving normal behavior on benign inputs. These trigger–response behaviors then

serve as explicit identifiers for ownership verification.

Such methods are often referred to as "fingerprinting" or "backdoor fingerprinting," but this terminology can be misleading. Unlike fingerprints that arise intrinsically from architecture or training data, watermarks are deliberately embedded identity signals. We therefore categorise these specific-response methods as model watermarking.

From the perspective of watermark embedding mechanisms, existing model watermarking methods can be broadly divided into two categories: *fine-tuning-based watermarking* and *model-editing-based watermarking*. The key distinction lies in whether the watermark is implanted through additional training or by directly editing model parameters.

Model fine-tuning watermark
- **Evidence:** trigger-response behaviour embedded.
- **Access:** black-box.
- **Verification:** keyed verification.
- **Stressors:** suppression/filtering, prompt interventions, extraction, fine-tuning, merging, distillation.

Model watermarking via fine-tuning represents an early and widely used paradigm for embedding trigger–response behaviours into LLMs. In this setting, the watermark is injected by fine-tuning the model on data containing secret trigger–response pairs, thereby inducing the model to associate predefined triggers with designated responses.

Instructional fingerprinting [Xu *et al.*, 2024a] demonstrates that instruction-style supervised fine-tuning can implant watermark behaviours without requiring access to model internals. Subsequent work extends the approach via stronger trigger-response binding, as in Chain&Hash [Russinovich *et al.*, 2026]; improved deployment efficiency, as in FP-vec [Xu *et al.*, 2024b]; greater scalability via in-distribution trigger–response pairs, as in Perinucleus Fingerprinting [Nasery *et al.*, 2025]; and enhanced robustness to further fine-tuning by anchoring watermark behaviours in lower-level representations, as in TIBW [Mo *et al.*, 2025].

The primary stressors for fine-tuning-based watermarks include suppression or filtering, prompt-level interventions, etc. Because many model watermarks are implemented through trigger–response behaviours, their robustness should also be evaluated against adjacent backdoor-repair and unlearning techniques. For example, class-wise trigger recovery and unlearning suggest that embedded trigger behaviours can be actively discovered and weakened [Hou *et al.*, 2025].

Model-editing watermark
- **Evidence:** keyed behaviours inserted.
- **Access:** black-box.
- **Verification:** keyed verification.
- **Stressors:** subsequent edits/unlearning, fine-tuning, merging, distillation.

Model-editing watermarking inserts keyed behaviours through localized parameter edits, enabling low-cost watermark implantation with limited impact on model utility. Representative designs either reinforce trigger–target associations, as in FPEdit [Wang *et al.*, 2025a], or embed more natural keyed QA behaviours, as in EditMark [Li *et al.*, 2025a]. We evaluate the robustness of model watermarks under these stressors and defer a detailed discussion of the corresponding attack surface and evaluation protocols to Sec. 4.

3.4 Implicit-ID of Generated Content

Implicit-ID of generated content concerns determining whether a given output $\mathcal{Y}$ originates from a specific model $\mathcal{M}$, solely based on the output text under post-generation transformations $\tau \in \mathcal{T}$ and black-box access.

We distinguish two verification semantics: **generated content fingerprinting**, which exploits inherent idiosyncrasies in model responses to perform provenance inference; and **generated content watermarking**, which intentionally embeds a detectable signal during generation to support provenance binding and proactive asset protection.

Generated Content Fingerprinting

Generated content fingerprint
- **Evidence:** distributional/stylistic biases in outputs.
- **Access:** black-box.
- **Verification:** similarity-based attribution.
- **Stressors:** rewriting, decoding changes, confounding.

Generated content fingerprinting attributes a source model from *outputs alone* by extracting *intrinsic* and *unintended* regularities in the generated text, and performing non-keyed attribution, e.g., similarity scoring or classifier-based matching without modifying the model or decoding procedure.

The main technology is statistical fingerprinting of generated content, which treats stable deviations in an LLM's output distribution, whether lexical, syntactic, or embedding-level, as latent signatures for attribution through statistical modeling or classification. Recent studies report strong discriminability at the *model-family* level. Sun *et al.* [2025] show that word-level distributional cues enable embedding-based classifiers to distinguish major model families, such as GPT, Claude, and Gemini, with robustness under rewriting and translation. Suzuki *et al.* [2025] further suggest that identifiable "natural fingerprints" can emerge even among models trained on identical data, owing to randomness in optimization. From a detection standpoint, Antoun *et al.* [2024] demonstrate cross-model attribution that generalizes across sources and model sizes, while Bitton *et al.* [2025] improve reliability through classifier ensembles that preserve precision under style-mimic prompts. Because these signals are distributional, attribution is sensitive to prompt/system context and decoding settings. It is most defensible when framed as probabilistic evidence rather than cryptographic proof.

Generated Content Watermarking

Generated content watermarking is an intrusive provenance-binding mechanism that intentionally embeds verifiable identity signals into the text generation process, enabling provenance to be verified directly, without requiring access to model internals. We organize existing methods according to how identity signals are embedded. The first is *statistical watermarking*, which introduces detectable distributional biases into generated text, either through inference-time manipulation of token probabilities or through training-time internalization of watermark behaviors. The second is *keyed sampling watermarking*, which encodes identity within keyed

sampling trajectories, enabling provenance verification with little or no perceptible distortion to the generated content.

The statistical watermarking embeds a *verifiable distributional deviation* into generated text such that provenance can be detected from outputs alone. Whether injected via inference-time logit intervention or internalised during training, the watermark manifests as a persistent statistical shift detectable by a verifier. Based on the verification mechanism, this paradigm admits two dominant technical paths: *statistical hypothesis testing* and *statistical pattern learning*.

Statistical watermark of generated content
- **Evidence:** verifiable distributional deviation.
- **Access:** black-box.
- **Verification:** keyed verification.
- **Stressors:** rewrite/translate/summarise/character-level perturbations, decoding changes, adaptive removal.

The line of statistical hypothesis testing formulates provenance verification as a statistical hypothesis test, where the null hypothesis assumes natural, unwatermarked text. [Kirchenbauer *et al.*, 2023] introduce the canonical *red–green list* scheme, which biases token selection toward a pseudo-random subset of the vocabulary and verifies provenance through Z-score testing of token frequencies. [Dathathri *et al.*, 2024] refine this framework in SynthID-Text by using tournament sampling to preserve generation diversity while maintaining high detection specificity. To mitigate the fragility of token-level statistics under paraphrasing, subsequent work shifts verification into semantic space. [Hou *et al.*, 2024a] propose SemStamp, which partitions embedding space using locality-sensitive hashing and treats region occupancy as a statistical signature; K-SemStamp [Hou *et al.*, 2024b] further aligns partitions with semantic structure via clustering, improving robustness to linguistic transformations. Across these methods, watermark verification remains an *external, rule-based* statistical test.

The line of statistical pattern learning treats the watermark as a *latent statistical pattern* that can be internalized by the model and recognized by learned discriminators. This approach is particularly relevant in open-source settings, where inference-time logit biasing can be bypassed. [Gu *et al.*, 2024] demonstrate watermark distillation, showing that student models can reproduce a teacher's statistical watermark without explicit decoding constraints. [Xu *et al.*, 2025a] further improves detectability via reinforcement learning co-training, coupling generation with a trainable detector to shape output trajectories that maximise classification confidence while preserving fluency. This notion of internalised identity is extended to open-source security in Mark-Your-LLM [Xu *et al.*, 2025b], which embeds statistical behaviours via backdoor-based internalisation and verifies provenance using a paired classifier.

Keyed sampling watermarking encodes provenance through *secrecy in the sampling process*. Keyed sampling defines the watermark through secret sampling rules, typically keyed pseudo-randomness governing token selection, rather than through observable deviations in the output distribution. Verification then tests whether a generated sequence is consistent with the secret key and the model's logits. A central advantage of this paradigm is its *zero-distortion* property: the marginal distribution of watermarked text is theoretically identical to that of the original model. [Christ *et al.*, 2024] formalises this notion via *undetectable watermarks* based on cryptographic one-way functions, showing that outputs are computationally indistinguishable from unwatermarked text to any adversary lacking the key. [Hu *et al.*, 2024] further propose an unbiased watermarking framework based on reparameterized sampling, ensuring that expected token frequencies match the model's softmax distribution.

Keyed sampling watermark of generated content
- **Evidence:** keyed sampling trace.
- **Access:** black-box.
- **Verification:** keyed verification.
- **Stressors:** rewrite/translate/summarise/character-level perturbations, decoding changes, adaptive removal.

In keyed sampling watermarking, verification reduces to re-evaluating whether the observed token sequence adheres to the secret sampling trajectory induced by the key and the model probabilities. By exploiting the secrecy of the sampling path, these methods achieve strong robustness against adaptive detection while preserving generation fidelity.

4 Evaluation Metrics for ID Technologies

An Implicit-ID method should be evaluated as a claim-verification system: given a suspect asset and a claimed source, owner, or generation process, the verifier accepts or rejects the claim under a specified access interface, threshold, and transformation regime. Since the task and notation have been introduced in Section 2, we focus here on the metrics required for comparable evaluation.

4.1 Correctness of Identity Claims

Correctness measures whether the verifier accepts valid claims and rejects invalid ones. We use the true positive rate (TPR) and false positive rate (FPR):

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}},$$

where P is the number of positive claims and N is the number of negative claims. TP counts valid claims correctly accepted, while FP counts invalid claims incorrectly accepted. These quantities unify common measures used in fingerprinting and watermarking studies, including fingerprint success rate, reliability, and watermark detection accuracy [Xu *et al.*, 2025d; Yang *et al.*, 2025; Shao *et al.*, 2025a].

Identity technologies should be reported at meaningful operating points, such as TPR at a fixed maximum FPR, or through threshold-dependent curves such as ROC or precision-recall curves. For multi-source attribution, where the verifier chooses among multiple candidate owners or source models, top-k attribution accuracy and calibrated confidence should also be reported. Calibration is especially important for non-keyed fingerprinting, whose conclusions are usually probabilistic rather than cryptographic.

4.2 Robustness to Benign Transformations

Identity claims must remain verifiable after ordinary post-release changes. Let $\mathcal{T}_{\text{own}}$ denote benign owner-side or user-side transformations. Examples include dataset filtering, preprocessing, model fine-tuning, quantisation, pruning, distillation, model merging, prompt or policy updates, decoding changes, paraphrasing, translation, and summarisation.

For a transformation $\tau \in \mathcal{T}_{\text{own}}$, evaluation should apply the transformation to the identity-bearing asset, yielding $A' = \tau(A)$, and then measure correctness under the same declared access and threshold regime. A useful summary is benign robustness coverage:

$$\text{RC}_{\text{own}} = \frac{|\{\tau \in \mathcal{T}_{\text{own}} : \text{TPR}_\tau \geq C_{\text{TPR}},\ \text{FPR}_\tau \leq C_{\text{FPR}}\}|}{|\mathcal{T}_{\text{own}}|}.$$

4.3 Security Against Adaptive Manipulation

A second dimension of robustness concerns adversarial security. Let $\mathcal{T}_{\text{adv}}$ denote transformations chosen by an adaptive adversary to remove, evade, forge, or spoof identity evidence. Security evaluation must specify the adversary's knowledge and capability. For removal or evasion attacks, the attack succeeds when a true identity claim is rejected. For forgery or spoofing attacks, the attack succeeds when a false claim is accepted. Thus, attack success can be expressed as

$$\text{ASR}_{\text{remove}}(\tau) = 1 - \text{TPR}_\tau, \qquad \text{ASR}_{\text{forge}}(\tau) = \text{FPR}_\tau.$$

A compact adversarial robustness coverage can be defined analogously to benign robustness:

$$\text{RC}_{\text{adv}} = \frac{|\{\tau \in \mathcal{T}_{\text{adv}} : \text{TPR}_\tau \geq C_{\text{TPR}},\ \text{FPR}_\tau \leq C_{\text{FPR}}\}|}{|\mathcal{T}_{\text{adv}}|}.$$

This metric should be interpreted within the context of the specified threat model, since its value is inherently conditioned on the adversary's capabilities.

4.4 Utility and Deployment Cost

Identity technologies must also preserve the primary utility of the asset and be feasible to verify. Utility impact measures degradation caused by embedding or maintaining the identity,

$$\Delta U = U_{\text{orig}} - U_{\text{ID}},$$

where U_{orig} is the utility of the original asset and U_{ID} is the utility after identity construction. For generated text, U may include perplexity, semantic similarity, task accuracy, toxicity, or human preference. For models, it may include benchmark accuracy, instruction-following quality, safety behaviour, latency, or downstream task performance. For datasets, it may include data quality and downstream model performance. Watermarking usually requires careful reporting of ΔU because it modifies the asset or generation process. Fingerprinting is non-intrusive and often has negligible utility impact, but it imposes high verification costs.

Deployment Cost primarily refers to the computational overhead incurred by an identity technology. According to the stages at which identity is established and verified, we decompose the cost into embedding cost and verification cost. Embedding cost is the resource required to establish an identity. For watermarking, this includes data construction, fine-tuning, model editing, logit intervention, key management, or changes to the serving pipeline. For fingerprinting, explicit embedding cost is typically zero, but reference evidence extraction and storage should still be reported. Verification cost includes query volume, latency, compute, memory, auxiliary storage, access requirements, and human audit effort. These costs are central to deployability: a highly identifiable method may still be impractical if it requires white-box access to a frontier-scale model, thousands of queries, or expensive pairwise comparisons across many candidate sources.

5 Current Challenges and Future Directions

Identity signals that preserve utility. A persistent bottleneck is the identifiability–utility trade-off: many Implicit-ID mechanisms improve verification performance only at the cost of nontrivial utility degradation ΔU or by exploiting fragile surface-level cues. Future work should move beyond heuristic signal design and co-design identity mechanisms with *semantic fidelity* as a first-class objective, achieving high separability across prompts, domains, and access interfaces while preserving model utility and user experience.

Robustness across lifecycle drift and adaptive transformations. Existing methods often fail when models undergo routine adaptation (e.g., fine-tuning, compression) or when adversaries attempt removal/forgery. A central direction is to formalise robustness over explicit transformation suites $\mathcal{T} = \mathcal{T}_{\text{own}} \cup \mathcal{T}_{\text{adv}}$ and develop transformation-invariant identity signals and decision rules that remain valid under post-release drift, not merely in the original parameterisation.

Deployable verification with guarantees and shared benchmarks. Practical adoption is limited by verification cost (query volume, latency, compute) and by the lack of rigorous security statements. Future work should pursue sample-efficient, auditable verification and align claims with verification semantics: non-keyed attribution as calibrated statistical evidence, keyed verification with explicit notions such as unforgeability and non-removability. Community progress also requires standardised, cross-asset benchmarks that fix interfaces, stressors, and reporting (identifiability, robustness coverage, ΔU, and verification cost) to enable credible cross-paper comparison.

6 Conclusions

LLM identity technologies are becoming essential for protecting LLM-related assets and tracing their provenance. This survey has introduced *Implicit-ID* as a unifying abstraction for understanding fingerprinting and watermarking, and has organised existing methods across datasets, models, and generated content under two verification semantics: similarity-based attribution and keyed verification, and highlights the metrics that matter for deployment.

LLM identity is best understood as verification under distribution shift induced by benign adaptation and adaptive attacks. We hope this survey helps the community converge on reproducible benchmarks, consistent terminology, and practically defensible identity for trustworthy LLM deployment.

Acknowledgements

The authors would like to thank the anonymous reviewers and chairs for their constructive comments. This work was supported by the National Natural Science Foundation of China under Grants 62572382, 62506292, and U22A2098; the China Postdoctoral Science Foundation under Grants 2024M762604 and BX20250380; the Science and Technology Project of State Grid Headquarters under Grant 52199925001B-132-ZN; the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2025004; and the Shenzhen Loop Area Institute.

References

Wissam Antoun, Benoît Sagot, and Djamé Seddah. From text to source: Results in detecting large language model-generated content. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7531–7543, Torino, Italia, May 2024. ELRA and ICCL.

Yehonatan Bitton, Elad Bitton, and Shai Nisan. Detecting stylistic fingerprints of large language models. *arXiv preprint arXiv:2503.01659*, 2025.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. In *2022 IEEE symposium on security and privacy (SP)*, pages 824–841. IEEE, 2022.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR, 2024.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Linkang Du, Xuanru Zhou, Min Chen, Chusong Zhang, Zhou Su, Peng Cheng, Jiming Chen, and Zhikun Zhang. Sok: Dataset copyright auditing in machine learning systems. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2025.

Chenchen Gu, XIANG LI, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 38745–38768, 2024.

Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. TRAP: Targeted random adversarial prompt honeypot for black-box identification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11496–11517, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Xin He, Junxi Shen, Zhenheng Tang, Xiaowen Chu, Bo Li, Ivor W Tsang, and Yew-Soon Ong. Routemark: A fingerprint for intellectual property attribution in routing-based model merging. *arXiv preprint arXiv:2508.01784*, 2025.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 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 (Volume 1: Long Papers)*, pages 4067–4082, 2024.

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1706–1715, 2024.

Linshan Hou, Zhongyun Hua, Wei Luo, and Leo Yu Zhang. Fixguard: Repairing backdoored models via class-wise trigger recovery and unlearning. *IEEE Signal Processing Letters*, 32:2544–2548, 2025.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 45408–45436, 2024.

Zonghao Huang, Neil Zhenqiang Gong, and Michael K Reiter. A general framework for data-use auditing of ml models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1300–1314, 2024.

Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning. *arXiv preprint arXiv:2408.02871*, 2024.

Heng Jin, Chaoyu Zhang, Shanghao Shi, Wenjing Lou, and Y Thomas Hou. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

Shuai Li, Kejiang Chen, Jun Jiang, Jie Zhang, Qiyi Yao, Kai Zeng, Weiming Zhang, and Nenghai Yu. Editmark: Watermarking large language models based on model editing. *arXiv preprint arXiv:2510.16367*, 2025.

Yanming Li, Seifeddine Ghozzi, Cédric Eichler, Nicolas Anciaux, Alexandra Bensamoun, and Lorena Gonzalez Manzano. Data provenance auditing of fine-tuned large language models with a text-preserving technique. *arXiv preprint arXiv:2510.09655*, 2025.

Yuqing Liang, Jiancheng Xiao, Wensheng Gan, and Philip S Yu. Watermarking techniques for large language models: A survey. *Artificial Intelligence Review*, 59(2):74, 2026.

Zilong Lin, Jian Cui, Xiaojing Liao, and XiaoFeng Wang. Malla: Demystifying real-world large language model integrated malicious services. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4693–4710, 2024.

Gaoyang Liu, Tianlong Xu, Xiaoqiang Ma, and Chen Wang. Your model trains on my data? protecting intellectual property of training data via membership fingerprint authentication. *IEEE Transactions on Information Forensics and Security*, 17:1024–1037, 2022.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations*, 2021.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, Toronto, Canada, July 2023. Association for Computational Linguistics.

Weichuan Mo, Kongyang Chen, and Yatie Xiao. Tibw: Task-independent backdoor watermarking with fine-tuning resilience for pre-trained language models. *Mathematics (2227-7390)*, 13(2), 2025.

MZ Naser. Auditing the shadows: A review of methods to detect shared training data in large language models. *ACM Computing Surveys*, 58(7):1–34, 2025.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 125116–125152. Curran Associates, Inc., 2025.

Dario Pasquini, {Evgenios M.} Kornaropoulos, and Giuseppe Ateniese. Llmmap: Fingerprinting for large language models. In *Proceedings of the 34th USENIX Security Symposium*, Proceedings of the 34th USENIX Security Symposium, pages 299–318, 2025. Publisher Copyright: © 2025 by The USENIX Association All Rights Reserved.; 34th USENIX Security Symposium, USENIX Security 2025 ; Conference date: 13-08-2025 Through 15-08-2025.

Zhenzhen Ren, Guobiao Li, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Cotsrf: Utilize chain of thought as stealthy and robust fingerprint of large language models. *ArXiv*, abs/2505.16785, 2025.

Mark Russinovich, Yanan Cai, and Ahmed Salem. Hey, That's My Model! Introducing Chain & Hash, an LLM Fingerprinting Technique. In *International Conference on Learning Representations*, 2026.

Payel Sadhukhan and Tanujit Chakraborty. Footprints of data in a classifier: Understanding the privacy risks and solution strategies. *Information Systems Frontiers*, pages 1–27, 2025.

Shuo Shao, Yiming Li, Yu He, Hongwei Yao, Wenyuan Yang, Dacheng Tao, and Zhan Qin. Sok: Large language model copyright auditing via fingerprinting. *arXiv preprint arXiv:2508.19843*, 2025.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box llm fingerprinting via zeroth-order gradient estimation. *arXiv preprint arXiv:2510.06605*, 2025.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 51826–51843, 2024.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 57854–57885. PMLR, 13–19 Jul 2025.

Teppei Suzuki, Ryokan Ri, and Sho Takase. Natural fingerprints of large language models. *arXiv preprint arXiv:2504.14871*, 2025.

Shida Wang, Chaohu Liu, Yubo Wang, and Linli Xu. Fpedit: Robust llm fingerprinting through localized knowledge editing. *arXiv preprint arXiv:2508.02092*, 2025.

Xuhong Wang, Haoyu Jiang, Yi Yu, Jingru Yu, Yilun Lin, Ping Yi, Yingchun Wang, Yu Qiao, Li Li, and Fei-Yue Wang. Building intelligence identification system via large language model watermarking: a survey and beyond. *Artificial Intelligence Review*, 58(8):249, 2025.

Zehao Wu, Yanjie Zhao, and Haoyu Wang. Tensorguard: Gradient-based model fingerprinting for llm similarity detection and family classification. In *2025 40th IEEE/ACM*

International Conference on Automated Software Engineering (ASE), pages 445–456, 2025.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3277–3306, 2024.

Zhenhua Xu, Wenpeng Xing, Zhebo Wang, Chang Hu, Chen Jie, and Meng Han. Fp-vec: Fingerprinting large language models via efficient vector addition. *arXiv preprint arXiv:2409.08846*, 2024.

Xiaojun Xu, Yuanshun Yao, and Yang Liu. Learning to Watermark LLM-Generated Text via Reinforcement Learning. In *ICLR 2025 Workshops: WMARK*, 2025.

Yijie Xu, Aiwei Liu, Xuming Hu, Lijie Wen, and Hui Xiong. Mark your llm: Detecting the misuse of open-source large language models via watermarking. *arXiv preprint arXiv:2503.04636*, 2025.

Zhenhua Xu, Zhebo Wang, Maike Li, Wenpeng Xing, Chunqiang Hu, Chen Zhi, and Meng Han. Rap-sm: Robust adversarial prompt via shadow models for copyright verification of large language models. *arXiv preprint arXiv:2505.06304*, 2025.

Zhenhua Xu, Xubin Yue, Zhebo Wang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, Changting Lin, et al. Copyright protection for large language models: A survey of methods, challenges, and trends. *arXiv preprint arXiv:2508.11548*, 2025.

Yuliang Yan, Haochun Tang, Shuo Yan, and Enyan Dai. Duffin: A dual-level fingerprinting framework for llms ip protection. *ArXiv*, abs/2505.16530, 2025.

Zhiguang Yang, Gejian Zhao, and Hanzhou Wu. Watermarking for large language models: A survey. *Mathematics*, 13(9):1420, 2025.

Peigen Ye, Huali Ren, Zhengdao Li, Anli Yan, Hongyang Yan, Shaowei Wang, and Jin Li. Securing large language models: A survey of watermarking and fingerprinting techniques. *ACM Computing Surveys*, 2025.

Do-hyeon Yoon, Minsoo Chun, Thomas Allen, Hans Müller, Min Wang, and Rajesh Sharma. Intrinsic fingerprint of llms: Continue training is not all you need to steal a model! *arXiv preprint arXiv:2507.03014*, 2025.

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. *Advances in Neural Information Processing Systems*, 37:126332–126362, 2024.

Boyi Zeng, Lin Chen, Ziwei He, Xinbing Wang, and Zhouhan Lin. Awm: Accurate weight-matrix fingerprint for large language models. *arXiv preprint arXiv:2510.06738*, 2025.

Jie Zhang, Jiayuan Li, Haiqiang Fei, Lun Li, and Hongsong Zhu. Easydetector: Using linear probe to detect the provenance of large language models. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 2410–2417. IEEE, 2024.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 48092–48117, 2025.