RShield: A User-level Traceable Backdoor Watermark for LLMs in Embedding-as-a-Service

Lingyun Xiang[1], Yufan Zhong[1], Chengfu Ou[2], Zhihua Xia[2], Chunfang Yang[3], Daojian Zeng[4], Zhangjie Fu[5]
[1]Changsha University of Science and Technology, [2]Jinan University,
[3]Henan Key Laboratory of Cyberspace Situation Awareness, [4]Hunan Normal University,
[5]Nanjing University of Information Science and Technology
{xiangly, rennega}@csust.edu.cn, {theoyah, chunfangyang, wwwfzj}@126.com
{xia_zhihua, zengdj916}@163.com

Abstract

Embedding-as-a-Service (EaaS) has emerged as a critical paradigm for commercializing large language models (LLMs). However, existing backdoor watermarking techniques are fundamentally limited to "zero-bit" detection, which prevents user-level traceability in multi-user EaaS scenarios. To address these limitations, we propose RShield, a multi-bit backdoor watermarking that enables reliable user-level attribution of LLMs for EaaS under model extraction attacks. RShield integrates Reed-Solomon error-correcting codes with orthogonal feature mapping to introduce highly-structured redundancy, constructing fault-tolerant symbol sequences for multi-bit watermark space, thereby staying recoverable even after aggressive extraction noise condition. To mitigate semantic distortion under the interference of noise channel, RShield employs a lightweight Adapter to adaptively inject multi-bit watermarks in the feature space, preserving the quality of EaaS while achieving a user-level traceability. Extensive experiments on four NLP benchmarks demonstrate that RShield efficiently achieves 100% multi-bit watermark recovery and high semantic fidelity under model extraction attacks compared to existing methods, while significantly reducing the degradation of watermarking on downstream task performance.

1 Introduction

In recent years, Large Language Models (LLMs) (OpenAI et al., 2024; Touvron et al., 2023) have demonstrated remarkable capabilities in natural language understanding and generation. To commercialize these advances, Embedding-as-a-Service (EaaS) has become a widely adopted deployment paradigm that provides high-quality text embeddings to downstream applications through black-box APIs (OpenAI, 2024; Google, 2023). However, the black-box deployment of EaaS exposes it to model extraction attacks, where adversaries train low-cost substitute models from collected query–response pairs (Tramèr et al., 2016; Krishna et al., 2020), or even reconstruct semantic representations from unlabeled data (Liu et al., 2022). Such attacks pose serious threats to Intellectual Property (IP) protection and disrupt the EaaS ecosystem by enabling unauthorized and competitive services. (Xu et al., 2022).

Due to the black-box nature of EaaS and the lack of discrete output labels, traditional copyright protection techniques for classification models, such as methods based on model parameters (Uchida et al., 2017) and decision boundary fingerprinting (Le Merrer et al., 2020) are largely ineffective, as substitute models produced by extraction attacks differ substantially in parameter space and EaaS APIs expose only continuous embeddings rather than discrete classification labels (Peng et al., 2023). Consequently, backdoor-based watermarking has become the primary approach for EaaS protection (Li et al., 2025), as it implants transferable behavioral patterns into the input-output mapping that can be effectively preserved in the extracted models. Early work such as EmbMarker (Peng et al., 2023) initiated this field by injecting watermarks via linear interpolation, but was shown to be vulnerable to removal attacks. Subsequent studies, including WARDEN (Shetty et al., 2024) and ESpeW (Wang et al., 2025), improved robustness and stealthiness by adopting multi-directional backdoor mechanisms and position-specific embedding strategies, respectively.

Despite their effectiveness, existing backdoor-based watermarking methods share a fundamental limitation: they are restricted to "zero-bit" detection based on binary hypothesis testing. While sufficient for ownership verification, such methods cannot support the fine-grained accountability required in multi-user EaaS scenarios, where unique identifiers must be encoded through multi-bit watermarks (Szyller et al., 2021; He et al., 2022).

Findings of the Association for Computational Linguistics: ACL 2026, pages 27014–27028

However, existing multi-bit watermarking methods (Yoo et al., 2024; Qu et al., 2025; Wang et al., 2024) are predominantly designed for generative LLMs, where watermark messages are embedded by modulating discrete token probability distributions during autoregressive decoding. In contrast, EaaS models only output continuous embeddings without any decoding process, rendering these generative multi-bit watermarking schemes inapplicable. Such an architectural mismatch calls for a new design paradigm: directly encoding multi-bit information into the embedding space under strict semantic-preservation constraints while facing perturbation attacks, which introduces new challenges. These challenges arise because model extraction effectively acts as a lossy communication channel (Liu et al., 2022), which can severely corrupt the delicate signals needed for multi-bit encoding. At the same time, watermark injection must be tightly constrained to preserve semantic fidelity, as excessive perturbations to embeddings would significantly degrade downstream task performance.

To address the aforementioned challenges, we propose **RShield**, a multi-bit watermarking framework tailored for LLMs in EaaS scenarios. RShield is designed to enable reliable user-level traceability under model extraction attacks by explicitly treating extraction as a noisy communication channel. To mitigate extraction-induced noise, RShield encodes copyright information using Reed–Solomon (RS) error-correcting codes (Reed and Solomon, 1960), allowing reliable multi-bit information recovery from extracted models. To balance multi-bit information capacity with semantic fidelity, RShield further introduces a lightweight, residual-based Adapter for adaptive watermark embedding in the feature space, rather than relying on static interpolation. Combined with a high-discriminability vector codebook constructed via the Gram–Schmidt process (Trefethen and Bau, 1997), RShield enables precise and imperceptible watermark injection while preserving embedding utility. Together, these designs allow RShield to achieve reliable multi-bit recovery and provide statistically verifiable copyright evidence in EaaS settings. The main contributions of this paper are summarized below:

- We formulate the problem of multi-bit watermarking for copyright protection of LLMs in the EaaS setting under model extraction attacks, and propose **RShield**, a framework that enables reliable recovery of multi-bit identity information from surrogate models in the presence of extraction-induced noise, thereby supporting user-level traceability.
- We propose a reliable watermarking design that integrates Reed–Solomon error-correcting codes with a lightweight adapter to inject multi-bit watermarks in the feature space without degrading semantic fidelity, achieving resilience to extraction-induced noise while preserving the quality of EaaS.
- Experiments on four NLP benchmarks demonstrate that RShield enables reliable multi-bit watermark recovery under model extraction attacks, achieving a 100% recovery success rate while preserving extremely high semantic fidelity. Moreover, it provides significantly stronger attribution confidence than existing methods for EaaS copyright protection.

2 Related Works

2.1 Model Extraction Attack

Model extraction attacks aim to replicate cloud-based victim models via black-box APIs (Tramèr et al., 2016; Orekondy et al., 2019; Krishna et al., 2020), where the model is treated as opaque without access to internal parameters (Wallace et al., 2020; Yue et al., 2021). Attackers typically use collected "query-response" pairs to train surrogate models that functionally replicate the behavior of the victim (Chandrasekaran et al., 2020; Zhao et al., 2025). However, recent findings by Carlini et al. (2024) challenge the assumption of parameter safety, demonstrating that for Transformer-based models, the final projection layer's weights can be analytically recovered solely from API outputs.

In the EaaS domain, threats are escalating. Specifically, the StolenEncoder attack (Liu et al., 2022) reconstructs semantic capabilities using only unlabeled data, while Tamber et al. (2025) showed that commercial-grade surrogates can be trained for under $300. Notably, these surrogates may even outperform original models via domain adaptation (Xu et al., 2022). Such efficient attacks severely infringe on IP and disrupt the market ecosystem (Rakin et al., 2022).

2.2 Copyright Protection in EaaS

To address model extraction in EaaS, existing solutions primarily adopt backdoor-based watermark-

ing. EmbMarker (Peng et al., 2023) pioneered this domain by selecting moderate-frequency words as triggers and injecting watermarks via linear interpolation with a static target. To enhance flexibility and detectability, GuardEmb (Wang and Cheng, 2024) proposed a dynamic watermarking method utilizing a learnable mapping network to adaptively modify embeddings and co-trains a verifier for detection. Meanwhile, to mitigate the vulnerability of single-target watermarks against embedding similarity shift attacks, AMT-WM (Yang et al., 2024) introduced an adaptive multi-target framework, using orthogonal vectors to dynamically select the most suitable target.

However, subsequent research (Shetty et al., 2024) revealed its susceptibility to CSE(Clustering, Selection, Elimination) attacks, which exploit statistical anomalies for watermark removal. To mitigate this, WARDEN (Shetty et al., 2024) introduced a multi-directional backdoor mechanism. By utilizing the Gram–Schmidt process (Trefethen and Bau, 1997) to construct orthogonal target embeddings, it diversifies watermark directions to effectively resist the elimination step of CSE. Recently, to resolve the "shared components" issue of global interpolation, ESpeW (Wang et al., 2025) proposed an embedding-specific injection strategy. Departing from methods using fixed direction vectors, ESpeW dynamically identifies dimensions with the smallest absolute values in each embedding and utilizes a masking mechanism to replace them with hidden signals. This ensures the absence of shared principal component directions exist among watermarked samples, significantly enhancing stealthiness against statistical removal.

Alongside the development of backdoor-based techniques, recent research has also explored non-backdoor paradigms. For instance, WET (Shetty et al., 2025) introduced a global watermarking approach that applies a secret linear transformation to all original embeddings. This method embeds a uniform signal without requiring trigger words, which can later be verified via a pseudoinverse transformation.

3 Methodology

3.1 Problem Formulation

We formulate the multi-bit watermark task in EaaS scenarios as an "Information Transmission problem over Noisy Model Channels". This framework involves the following key components:

- **Victim Service (S_v):** Denote the victim model as Θ_v, which provides an EaaS service S_v. Upon receiving a user query x, Θ_v computes its original embedding e_o.
- **Watermark Injection (F):** To achieve copyright tracing, the defender utilizes an injection function F to embed specific multi-bit message m, generating a watermarked embedding $e_p = F(e_o, x, m)$ which is returned to the user.
- **Pirated Service as Noisy Channel ($S_a, \mathcal{N}$):** Assume an attacker acquires data pairs (x, e_p) via a single API account and uses them to train a pirated model Θ_a (providing service S_a). In this process, the parameter updates and structural discrepancies of Θ_a constitute the channel noise $\mathcal{N}$.

Task Objective. Unlike existing zero-bit methods, the core goal of this task is to withstand channel noise $\mathcal{N}$ and recover the original information from Θ_a. Formally, the defender aims to find an extraction function $\mathcal{G}$ such that the reconstructed message $\hat{m} = \mathcal{G}(\Theta_a) \approx m$. To this end, the task imposes three core requirements:

- **Semantic Fidelity:** The watermarked embedding e_p should preserve the core semantic features of e_o, ensuring no significant utility degradation on downstream tasks.
- **Watermark Detectability:** The framework should be able to verify the presence of the watermark signal within the pirated model Θ_a, effectively distinguishing it from non-watermarked benign models.
- **Multi-bit Extractability:** The framework should be able to accurately recover the embedded multi-bit message m from the pirated model Θ_a, thereby enabling precise source attribution.

Threat Model. We formalize the threat model for EaaS under model extraction attacks as follows. A detailed description is provided in Appendix A.

- **Attacker's Goal and Capability.** The attacker aims to replicate the victim service S_v via black-box API access without knowing internal parameters. We model this extraction process as "channel noise," where the pirated model Θ_a acts as a "lossy compression" of the

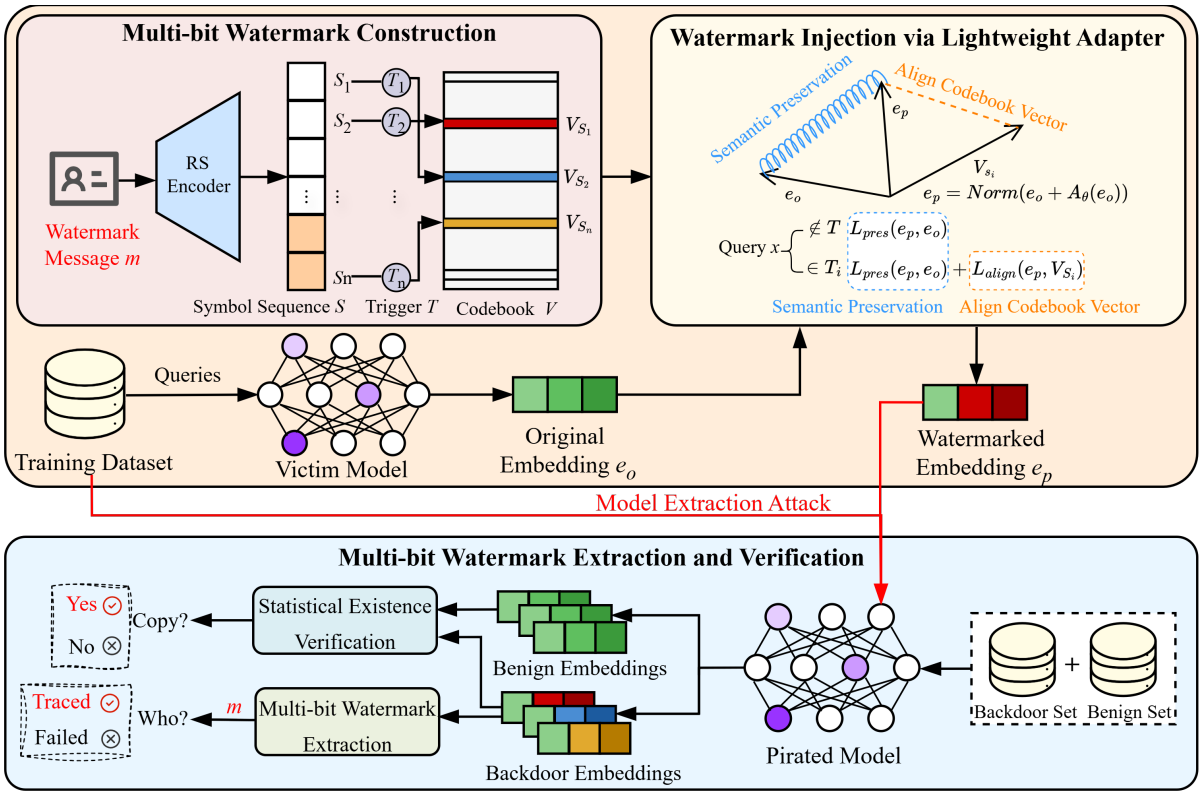


Figure 1: Overall Framework of RShield.

original representation space. This inevitably introduces severe distortion to the embedded signal, making the precise recovery of multi-bit information highly challenging.

- **Defender's Goal and Capability.** Possessing the victim model Θ_v, the defender aims to achieve two hierarchical objectives: first, to reliably verify the presence of the watermark signal within a suspicious model Θ_a to distinguish it from benign models; and second, to accurately recover the specific multi-bit message m from the pirated model for precise source tracing.

3.2 Framework of RShield

To address the information loss induced by the "*noisy channel*" of model extraction, we propose RShield, a multi-bit watermarking framework for EaaS. The core idea of RShield is to combine the error-correction capability of RS codes with orthogonal feature mapping, introducing structured redundancy that improves resilience against channel noise during model extraction. As illustrated in Figure 1, RShield consists of three key components. First, in the watermark construction phase, we construct a fault-tolerant RS symbol sequence and a discriminative vector codebook. Second, in the watermark injection phase, we employ a lightweight Adapter module with a non-linear residual structure to achieve adaptive and imperceptible embedding in the feature space while preserving semantic fidelity. Finally, in the extraction and verification phase, we leverage majority voting and statistical hypothesis testing to reliably recover multi-bit copyright information from lossy pirated models and enable infringement verification.

Multi-bit Watermark Construction. To counter signal distortion in the "*noisy channel*" of model extraction, we introduce the error-correction mechanism of RS codes (Reed and Solomon, 1960) and establish a high-discriminability orthogonal mapping to reliably encode multiple bits of watermark information into a structured codeword space, thereby enabling successful recovery under extraction-induced noise.

First, the multi-bit watermark message m is represented as a sequence of length k, $m = (m_1, \ldots, m_k)$, which collectively encodes user-specific identifying information. We utilize the RS encoding function $\mathcal{E}_{\mathcal{RS}}$ to extend m into a redundant sequence $S = (s_1, \ldots, s_k, \ldots, s_n)$, where $s_i \in \mathbb{F}_p$. RS codes ensure that m is fully recover-

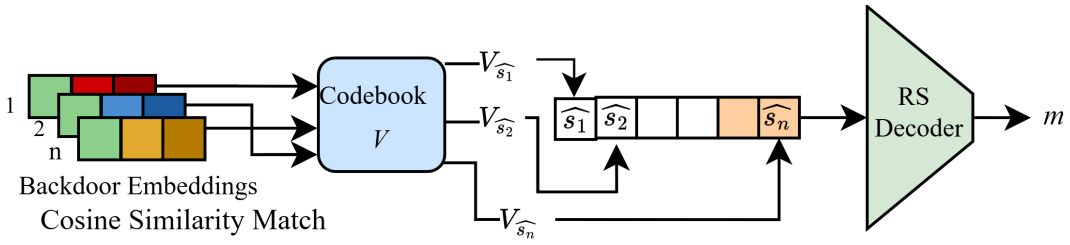


Figure 2: Illustration of the multi-bit watermark extraction pipeline.

able provided the number of symbol errors during extraction does not exceed $(n-k)/2$.

Second, to ensure high discriminability in the feature space, we follow WARDEN (Shetty et al., 2024) and apply the Gram–Schmidt process over the dimension of $\mathbb{F}_p$ to generate an orthonormal codebook $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$, such that $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$. Each vector $\mathbf{v}_k$ uniquely corresponds to a value in $\mathbb{F}_p$. This orthogonality minimizes mutual information between signals, significantly enhancing noise resistance.

Finally, we map the sequence S to the embedding space via a low-frequency trigger strategy. We assign an exclusive trigger set $\mathcal{T}_i$ containing rare words to each position i, constructing $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$. To balance injection strength and semantic fidelity, we strictly control trigger samples to approximately 1% of the total data. For position i with symbol s_i, samples in $\mathcal{T}_i$ are designated to target the codebook vector $\mathbf{v}_{s_i}$.

Watermark Injection via Lightweight Adapter. Current methods (Peng et al., 2023; Shetty et al., 2024) typically rely on static linear interpolation, which is ill-suited to accommodate the complex error-correcting structures and precise high-dimensional mappings. To address this, we introduce a trainable lightweight Adapter $\mathcal{A}$ employing a **non-linear residual structure**. By learning input-specific mappings, this module precisely embeds multi-bit watermarks while maintaining the semantic fidelity of original embeddings.

Given an input x and its original embedding $e_o = \Theta_v(x)$, the watermarked embedding is computed as $e_p = \mathcal{A}(e_o)$. We optimize $\mathcal{A}$ by jointly minimizing Vector Codebook Alignment and Global Semantic Preservation losses:

$$L = \lambda_{align} L_{align} + \lambda_{pres} L_{pres} \tag{1}$$

where λ_{align} and λ_{pres} are balancing weights.

(1) Vector Codebook Alignment Loss. For an input x belonging to trigger set T_i, we aim to map its features to the orthogonal codebook vector v_{s_i}. Let D_t be the set of trigger samples in a batch. For any $x \in D_t \cap T_i$, the target vector is $v(x) = v_{s_i}$. We minimize the mean cosine distance deviation:

$$L_{align} = \frac{1}{|D_t|} \sum_{x \in D_t} \left(1 - \frac{e_p \cdot v(x)}{||e_p||_2 ||v(x)||_2}\right) \tag{2}$$

The loss is 0 for non-trigger samples. Minimizing L_{align} enables the Adapter to precisely activate codebook directions upon detecting trigger patterns.

(2) Global Semantic Preservation Loss. To preserve the original model's semantic representation, we constrain the Adapter to maintain a close Euclidean distance to e_o across all N samples:

$$L_{pres} = \frac{1}{N} \sum ||e_p - e_o||_2^2 \tag{3}$$

This ensures $\mathcal{A}$ performs only minute, directional adjustments in the feature space rather than destructive reconstruction, thereby maximally preserving downstream task performance.

Multi-bit Watermark Extraction and Verification. Facing feature space distortion and noise in model extraction, single-sample verification is unreliable. RShield addresses this via two mechanisms: **multi-bit extraction** integrating majority voting with error correction to counteract noise, and **statistical verification** using hypothesis testing for rigorous evidence when decoding is incomplete.

(1) Multi-bit Watermark Extraction. To reliably recover information from the noisy pirated model Θ_a, we employ a codebook-based majority voting algorithm. We construct two validation sets for each sequence position i: a *natural set* (real samples containing trigger $\mathcal{T}_i$) and a *strong set* (repeated trigger words). Success in either set constitutes successful tracing.

For a set of K samples $\{x_1, \ldots, x_K\}$ corresponding to position i, we obtain embeddings $\{e_k\}$

from Θ_a. Each sample "votes" for the codebook vector $\mathbf{v}_j$ with the highest cosine similarity:

$$\widehat{v_k} = \arg\max_{j \in F_p} \cos(e_k, \mathbf{v}_j) \tag{4}$$

The most frequent vote becomes the predicted symbol $\widehat{s_i}$. After reconstructing the sequence $\hat{S} = (\widehat{s_1}, \ldots, \widehat{s_n})$, we apply RS decoding $\mathcal{D}_{\mathcal{RS}}$. If symbol errors are within the correction limit, the original message $m = \mathcal{D}_{\mathcal{RS}}(\hat{S})$ is precisely recovered, as illustrated in Figure 2.

(2) Statistical Existence Verification. To provide rigorous copyright evidence, we adapt the framework from WARDEN (Shetty et al., 2024) to determine infringement by detecting statistical shifts between distributions. For symbol s_i, we construct a backdoor set $\mathcal{D}^{(i)}_{\text{backdoor}}$ and a shared benign set $\mathcal{D}_{\text{benign}}$. We compute the distributions of cosine similarities (C) and Euclidean distances (L) relative to the target vector $\mathbf{v}_{s_i}$:

$$\begin{aligned} C^{(i)}_{\text{backdoor}} &= \left\{ \cos(\Theta_a(x), \mathbf{v}_{s_i}) \;\middle|\; x \in \mathcal{D}^{(i)}_{\text{backdoor}} \right\}, \\ C^{(i)}_{\text{benign}} &= \left\{ \cos(\Theta_a(x), \mathbf{v}_{s_i}) \mid x \in \mathcal{D}_{\text{benign}} \right\}. \end{aligned} \tag{5}$$

$$\begin{aligned} L^{(i)}_{\text{backdoor}} &= \left\{ \|\Theta_a(x) - \mathbf{v}_{s_i}\|_2 \;\middle|\; x \in \mathcal{D}^{(i)}_{\text{backdoor}} \right\}, \\ L^{(i)}_{\text{benign}} &= \left\{ \|\Theta_a(x) - \mathbf{v}_{s_i}\|_2 \mid x \in \mathcal{D}_{\text{benign}} \right\}. \end{aligned} \tag{6}$$

We quantify watermark strength by the mean shifts between these distributions:

$$\begin{aligned} \Delta\cos_i &= \mathbb{E}\left[C^{(i)}_{\text{backdoor}}\right] - \mathbb{E}\left[C^{(i)}_{\text{benign}}\right], \\ \Delta l_{2,i} &= \mathbb{E}\left[L^{(i)}_{\text{backdoor}}\right] - \mathbb{E}\left[L^{(i)}_{\text{benign}}\right]. \end{aligned} \tag{7}$$

To assess significance, we apply the Kolmogorov-Smirnov (KS) test (Berger and Zhou, 2014) on the cosine distributions to obtain p-value$_i$. We aggregate metrics across all n symbols to select the most significant deviation:

$$\begin{aligned} \Delta\cos &= \max_i \Delta\cos_i, \\ \Delta l_2 &= \min_i \Delta l_{2,i}, \\ p\text{-value} &= \min_i p\text{-value}_i. \end{aligned} \tag{8}$$

A significantly positive $\Delta\cos$ with an extremely small p-value strongly rejects the null hypothesis that the similarity distributions are consistent, confirming the watermark's presence.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate the ability of RShield, we conducted experiments on four NLP benchmark datasets, including SST-2 (Socher et al., 2013) for sentiment classification tasks, MIND (Wu et al., 2020), which was specifically designed for news recommendation but is used here for news classification tasks, Enron Spam (Metsis et al., 2006) for spam detection, and the standard news classification dataset AG News (Zhang et al., 2015). These datasets cover varying text lengths and semantic complexities, allowing for a full verification of the model's applicability in diverse scenarios(see Appendix C.1 for details).

Model Architectures. We selected the GPT-3 text-embedding-002 API as the backbone of the victim model to encode input text into 1536-dimensional original embeddings. The RShield Adapter module is designed as a lightweight two-layer fully connected network, utilizing ReLU activation functions and Dropout between layers, to inject watermarks into the original embeddings. To simulate model extraction attacks, we constructed a pirate model based on BERT-Base-Cased(Devlin et al., 2019). The attacker trains a three-layer fully connected network to extract the victim model, thereby testing the survivability of the watermark during the model extraction process.

Implementation Details. Regarding watermark configuration, we adopted RS coding on the finite field $\mathbb{F}_{p=257}$ and set the number of redundant symbols $n_{sym} = 4$ to ensure error-correction capability. To construct a stealthy and efficient trigger set, we first utilized 1,801,350 samples from the WikiText dataset (Merity et al., 2017) to conduct word frequency statistics, and subsequently selected trigger words from the low-frequency word interval based on the statistical results. Each symbol corresponds to an exclusive trigger set, where each trigger set contains 2 trigger words, and the proportion of trigger samples in the dataset is controlled at approximately 1%. During training, the Adapter was trained for 50 epochs, and the extraction network was trained for 20 epochs. We used the AdamW optimizer (Loshchilov and Hutter, 2019) and balanced watermark strength with semantic fidelity by dynamically adjusting the loss weights λ_{align} and λ_{pres} for different datasets. Hyperparameter configurations are provided in Appendix C.2.

Table 1: Performance comparison on SST-2, MIND, AG News, and Enron Spam datasets. The symbol "—" indicates non-applicability as baseline methods do not support multi-bit extraction. ↑ (↓) indicates that higher (lower) values are preferred. Note that the results for baseline methods are retrieved from Wang et al. (2025)

Dataset	Method	ACC(%)	SA(%)	Success Rate(%)	p-value↓	$\Delta\cos$ (%)↑	Δl_2 (%)↓
SST-2	Original	93.55 ± 0.30	7.50 ± 5.00	0.00 ± 0.00	$< 10^{-127}$	1.55 ± 0.26	-3.11 ± 0.52
	EmbMarker	93.46 ± 0.46	—	—	$< 10^{-11}$	9.71 ± 0.57	-19.43 ± 1.14
	WARDEN	94.04 ± 0.46	—	—	$< 10^{-11}$	12.18 ± 0.39	-24.37 ± 0.77
	EspeW	93.46 ± 0.46	—	—	$< 10^{-10}$	6.46 ± 0.87	-12.92 ± 1.75
	RShield	93.55 ± 0.24	100.00 ± 0.00	100.00 ± 0.00	$< 10^{-143}$	102.94 ± 0.94	-206.02 ± 1.72
MIND	Original	77.24 ± 0.12	9.09 ± 7.42	0.00 ± 0.00	$< 10^{-115}$	1.19 ± 0.71	-2.38 ± 1.42
	EmbMarker	77.17 ± 0.20	—	—	$< 10^{-11}$	13.53 ± 0.11	-27.06 ± 0.22
	WARDEN	77.23 ± 0.09	—	—	$< 10^{-11}$	18.05 ± 0.48	-36.10 ± 0.95
	EspeW	77.22 ± 0.12	—	—	$< 10^{-8}$	8.68 ± 0.24	-17.36 ± 0.47
	RShield	77.25 ± 0.14	100.00 ± 0.00	100.00 ± 0.00	$< 10^{-147}$	101.72 ± 1.07	-203.44 ± 2.14
AG News	Original	93.54 ± 0.08	5.77 ± 3.84	0.00 ± 0.00	$< 10^{-127}$	1.75 ± 0.64	-3.49 ± 1.29
	EmbMarker	93.60 ± 0.06	—	—	$< 10^{-11}$	13.15 ± 0.55	-26.29 ± 1.11
	WARDEN	93.22 ± 0.10	—	—	> 0.0083	-6.24 ± 5.96	12.47 ± 11.92
	EspeW	93.42 ± 0.16	—	—	$< 10^{-11}$	9.59 ± 0.74	-19.19 ± 1.49
	RShield	93.41 ± 0.09	90.39 ± 7.36	100.00 ± 0.00	$< 10^{-154}$	101.07 ± 3.75	-202.14 ± 7.52
Enron Spam	Original	94.88 ± 0.09	10.42 ± 4.17	0.00 ± 0.00	$< 10^{-114}$	1.49 ± 1.03	-2.98 ± 2.06
	EmbMarker	94.86 ± 0.24	—	—	$< 10^{-10}$	9.75 ± 0.11	-19.49 ± 0.21
	WARDEN	94.31 ± 0.44	—	—	$< 10^{-11}$	7.00 ± 0.62	-14.00 ± 1.24
	EspeW	94.73 ± 0.23	—	—	$< 10^{-10}$	7.23 ± 0.35	-14.47 ± 0.70
	RShield	94.60 ± 0.11	91.67 ± 6.81	100.00 ± 0.00	$< 10^{-151}$	96.75 ± 3.25	-193.51 ± 6.50

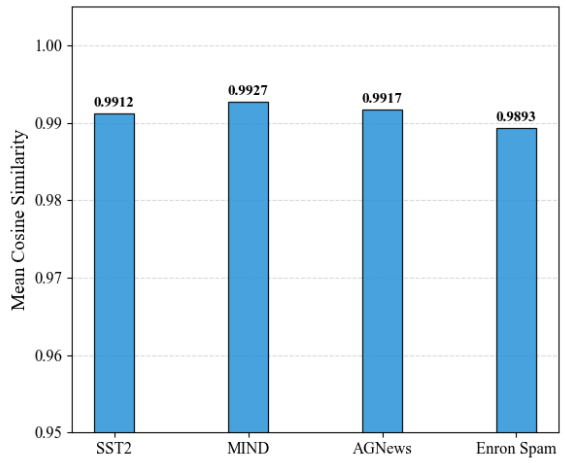


Figure 3: Mean cosine similarity comparison.

Evaluation Metrics. We evaluated RShield from three dimensions (see Appendix C.3 for details): (1) **Watermark Effectiveness:** Measured by symbol accuracy (SA), RS decoding Success Rate, and statistical significance metrics ($\Delta\cos$, $\Delta l2$, p-value) from WARDEN (Shetty et al., 2024). (2) **Semantic Fidelity and Utility:** Assessed by calculating the cosine similarity of embeddings and the accuracy (ACC) of downstream tasks. (3) **Indistinguishability:** Evaluated visually using PCA projections to measure the distributional overlap between original and watermarked embeddings.

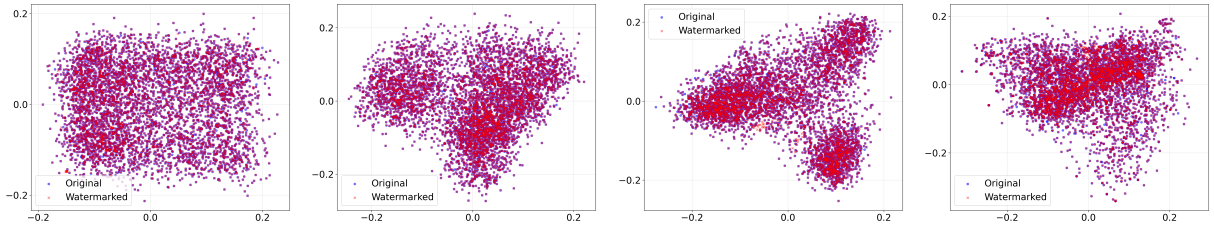
4.2 Main Results

Table 1 summarizes the comparative results of RShield against baseline methods, including EmbMarker (Peng et al., 2023), WARDEN (Shetty et al., 2024), and ESpeW (Wang et al., 2025). RShield demonstrates high-capacity multi-bit injection while excelling in utility preservation and verification significance.

Regarding downstream tasks, RShield exhibits exceptional semantic fidelity with negligible performance degradation. For instance, on SST-2, RShield achieves 93.55% accuracy, perfectly matching the original model's 93.55%. This suggests that jointly optimizing semantic preservation and vector-codebook alignment losses effectively maintains the original distribution, confirming the method's high stealthiness and utility.

In terms of extractability, RShield demonstrates reliable multi-bit capabilities. Although SA fluctuates due to channel noise and stochastic trigger distribution, the RS error-correction mechanism ensures a perfect 100% final recovery success rate across all settings. This demonstrates that RShield enables precise copyright provenance tracking even under model extraction attacks.

Finally, concerning statistical verification, RShield yields p-values $< 10^{-143}$, offering extremely high confidence. Given theoretical ranges



(a) SST-2 (b) MIND (c) AG News (d) Enron Spam

Figure 4: PCA visualization of embedding distributions.Watermarked embeddings remain visually indistinguishable from the original ones.

of $[-2, 2]$ for $\Delta\cos$ and $[-4, 4]$ for $\Delta l2$, RShield induces a $\Delta\cos$ of approximately 100% and significant negative $\Delta l2$ values. While p-values can be hypersensitive to incidental correlations (Shetty et al., 2024), Δ-based metrics effectively rectify this issue: in negative control experiments, $\Delta\cos$ remains negligible at roughly 1.5%. Thus, using combined metrics ensures reliable copyright decisions.

4.3 Semantic Fidelity Analysis

Beyond downstream performance, we evaluate embedding quality via the average cosine similarity between original and watermarked embeddings. As shown in Figure 3, RShield exerts minimal impact across all datasets, with average similarity consistently exceeding 0.98, implying a semantic shift rate of less than 1%.

This high fidelity stems from our low-frequency word trigger strategy. Unlike global perturbation methods, RShield injects watermarks into only a small fraction of samples ($\approx 1\%$), leaving the majority virtually unchanged. This achieves macroscopic "zero-perturbation", ensuring preserved downstream utility. Moreover, these sparse and targeted modifications remain statistically negligible in the global average, effectively masking the watermark within the inherent noise of original embeddings(see Appendix C.4 for details).

4.4 Indistinguishability Visualization

To verify stealthiness geometrically, we visualize the feature space distributions using PCA projections, as shown in Figure 4. It can be clearly observed that original embeddings (blue dots) and watermarked embeddings (red crosses) exhibit high overlap and interleaving. Crucially, watermarked samples do not form isolated "islands" or outlier clusters, indicating that RShield's injection process smoothly integrates into the original manifold.

Table 2: Efficiency evaluation across varying watermark payloads(SST-2). "Adapter Training" denotes one-time offline cost, while "Extraction" measures latency for full payload reconstruction.

Payload(Bits)	Adapter Training (min)	Extraction (ms)
48	5.81 ± 0.44	20.85 ± 1.98
64	6.67 ± 1.93	22.47 ± 0.70
80	6.85 ± 2.20	25.01 ± 1.77
96	6.88 ± 2.25	27.58 ± 2.33
112	6.83 ± 2.07	29.64 ± 0.70
128	6.90 ± 2.11	32.16 ± 1.16
144	6.09 ± 0.46	33.76 ± 1.06

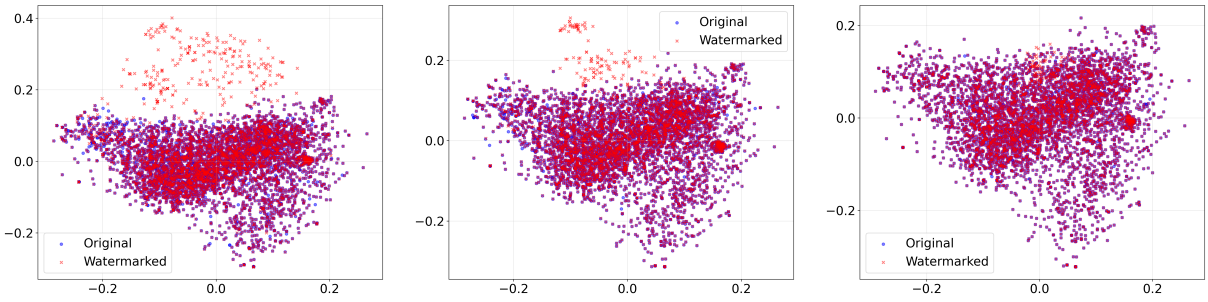
Despite substantial modifications on specific trigger samples, these changes remain globally "concealed" within the complex data distribution. This confirms high visual indistinguishability, making it difficult for attackers to separate watermarked samples based on geometric features.

4.5 Deployment Efficiency Analysis

We evaluate the efficiency of RShield on SST-2 (Table 2). Offline adapter training requires only 6–7 min, while multi-bit extraction through majority voting and RS decoding takes just 20–34 ms, with negligible impact on throughput. Furthermore, extracting a 112-bit payload requires approximately 1,800 queries, a practical volume well within typical commercial API rate limits. Finally, by utilizing valid low-frequency tokens instead of adversarial noise, RShield's triggers are less likely to be filtered, enabling seamless verification.

4.6 Ablation Study

Impact of Trigger Frequency. The trigger word frequency range is a pivotal hyperparameter for RShield's stealthiness. We conducted this ablation study on the Enron Spam dataset, as its high noise and discrete vocabulary render the embedding manifold hypersensitive to perturbations, making it ideal for detecting feature drift. Figure 5 illustrates



(a) $[1e^{-5}, 1e^{-4}]$ (b) $[1.2e^{-4}, 1.4e^{-4}]$ (c) $[1.4e^{-4}, 1.6e^{-4}]$

Figure 5: Evolution of PCA distributions on the Enron Spam dataset across different trigger word frequency ranges.

Table 3: Impact of payload scalability on downstream performance and watermark stealthiness (SST-2). The "Payload" column denotes the effective message bits, excluding a fixed redundancy of 4 symbols.

Payload(Bits)	ACC(%)	Success Rate(%)	PCA
48	93.55 ± 0.24	100.00 ± 0.00	Normal
64	93.21 ± 0.43	100.00 ± 0.00	Normal
80	93.18 ± 0.30	100.00 ± 0.00	Normal
96	93.15 ± 0.35	100.00 ± 0.00	Normal
112	93.35 ± 0.16	100.00 ± 0.00	Normal
128	93.09 ± 0.41	100.00 ± 0.00	Normal
144	93.09 ± 0.31	100.00 ± 0.00	**Abnormal**

the impact of three frequency intervals. At the extremely low interval $[1e^{-5}, 1e^{-4}]$ (Figure 5a), watermarked samples form detached, unnatural clusters, leading to direct exposure. While increasing the frequency to $[1.2e^{-4}, 1.4e^{-4}]$ (Figure 5b) alleviates separation, visible outlier trends persist. Conversely, the optimized interval $[1.4e^{-4}, 1.6e^{-4}]$ (Figure 5c) achieves seamless integration with the original distribution, eliminating the "island" phenomenon. This validates the necessity of our fine-grained selection strategy: blindly pursuing overly rare triggers leads to severe distribution drift, whereas precise parameter tuning preserves geometric stealthiness.

Impact of Payload. To evaluate the capacity–fidelity trade-off, we varied the watermark payload from 48 to 144 bits on SST-2, fixing redundancy at 4 symbols. As shown in Table 3, the RS decoding success rate remains at 100%, while downstream accuracy stays around 93%, indicating preserved semantic fidelity under increased capacity. The trade-off instead appears in geometric stealthiness, where larger payloads require more trigger sets, and the resulting perturbations eventually disrupt the embedding manifold. At 144 bits, PCA reveals distinct isolated clusters, making the watermark detectable. Thus, in practice, we recommend tuning the payload threshold to prevent distribution drift, using 128 bits for SST-2 as a reference to optimally balance capacity, recovery, and geometric stealthiness.

5 Conclusion

To address copyright challenges in EaaS, we propose RShield, a multi-bit watermarking framework that pioneers user-level traceability for stolen models. By synergizing RS codes with a Gram-Schmidt codebook and a lightweight Adapter, RShield mitigates lossy extraction noise and stealthily embeds watermarks through joint alignment and semantic optimization. Experiments on four NLP benchmarks demonstrate that RShield achieves a 100% recovery rate from extracted models while maintaining high semantic fidelity. Furthermore, it yields statistical confidence tens of orders of magnitude higher than existing methods, providing strong legal evidence for EaaS copyright protection.

Limitations

Despite RShield's superior performance in multi-bit traceability, practical deployment faces limitations regarding security dependencies and threat model scalability. First, the defense strictly relies on the confidentiality of the vector codebook; leakage via internal breaches could allow attackers to forge identities or remove watermarks. Furthermore, RShield is primarily designed to trace a dominant compromised identity. Handling highly distributed collusion where queries are evenly aggregated across many users remains challenging. In such cases, signal superposition can stress the limits of standard RS error-correction, suggesting the need for more specialized collusion-resistant coding. Second, an inherent trade-off exists between multi-bit payload capacity and resilience against

watermark removal attacks. While RShield successfully ensures the identifier survives the noisy extraction channel, equipping these high-capacity payloads with defenses against targeted semantic manipulations while strictly preserving EaaS fidelity constraints remains an open challenge for future research.

References

Vance W. Berger and YanYan Zhou. 2014. Kolmogorov–smirnov test: Overview.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. 2024. Stealing part of a production language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5680–5705. PMLR.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1309–1326. USENIX Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Google. 2023. How to use grounding for llms with text embeddings. Google Cloud Blog.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022. Cater: Intellectual property protection on text generation apis via conditional watermarks. In *Advances in Neural Information Processing Systems*, volume 35, pages 5431–5445. Curran Associates, Inc.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*.

Erwan Le Merrer, Patrick Perez, and Gilles Trédan. 2020. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244.

Hao Li, Yubing Ren, Yanan Cao, Yingjie Li, Fang Fang, and Xuebin Wang. 2025. From essence to defense: Adaptive semantic-aware watermarking for embedding-as-a-service copyright protection. *Preprint*, arXiv:2512.16439.

Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. 2022. Stolenencoder: Stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 2115–2128, New York, NY, USA. Association for Computing Machinery.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes – which naive bayes? In *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS)*, pages 28–69, Mountain View, California, USA.

OpenAI. 2024. New embedding models and api updates. `https://openai.com/blog/new-embedding-models-and-api-updates`. [Accessed 02-02-2024].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4949–4958.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for EaaS via backdoor watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7653–7668, Toronto, Canada. Association for Computational Linguistics.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. 2025. Provably robust multi-bit watermarking for ai-generated text. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, pages 201–220, USA. USENIX Association.

Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. 2022. Deepsteal: Advanced model extractions leveraging efficient weight stealing in memories. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1157–1174.

I. S. Reed and G. Solomon. 1960. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304.

Anudeex Shetty, Yue Teng, Ke He, and Qiongkai Xu. 2024. WARDEN: Multi-directional backdoor watermarks for embedding-as-a-service copyright protection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13430–13444, Bangkok, Thailand. Association for Computational Linguistics.

Anudeex Shetty, Qiongkai Xu, and Jey Han Lau. 2025. WET: Overcoming paraphrasing vulnerabilities in embeddings-as-a-service with linear transformation watermarks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23024–23043, Vienna, Austria. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N. Asokan. 2021. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 4417–4425, New York, NY, USA. Association for Computing Machinery.

Manveer Singh Tamber, Jasper Xian, and Jimmy Lin. 2025. Can't hide behind the API: Stealing black-box commercial embedding models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1958–1969, Albuquerque, New Mexico. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618, Austin, TX. USENIX Association.

Lloyd N. Trefethen and David Bau. 1997. *Numerical Linear Algebra*. SIAM.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, page 269–277, New York, NY, USA. Association for Computing Machinery.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5531–5546, Online. Association for Computational Linguistics.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2024. Towards codable watermarking for injecting multi-bits information to LLMs. In *The Twelfth International Conference on Learning Representations*.

Liaoyaqi Wang and Minhao Cheng. 2024. GuardEmb: Dynamic watermark for safeguarding large language model embedding service against model stealing attack. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7518–7534, Miami, Florida, USA. Association for Computational Linguistics.

Zongqi Wang, Baoyuan Wu, Jingyuan Deng, and Yujiu Yang. 2025. Robust and minimally invasive watermarking for EaaS. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2167–2191, Vienna, Austria. Association for Computational Linguistics.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Qiongkai Xu, Xuanli He, Lingjuan Lyu, Lizhen Qu, and Gholamreza Haffari. 2022. Student surpasses teacher: Imitation attack for black-box NLP APIs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2849–2860, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zuopeng Yang, Pengyu Chen, Tao Li, Kangjun Liu, Yuan Huang, and Xin Lin. 2024. Defending against similarity shift attack for eaas via adaptive multi-target watermarking. *Information Sciences*, 678:120893.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2024. 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 (Volume 1: Long Papers)*, pages 4031–4055, Mexico City, Mexico. Association for Computational Linguistics.

Zhenrui Yue, Zhankui He, Huimin Zeng, and Julian McAuley. 2021. Black-box attacks on sequential recommenders via data-free model extraction. In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, page 44–54, New York, NY, USA. Association for Computing Machinery.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. 2025. A survey on model extraction attacks and defenses for large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6227–6236, New York, NY, USA. Association for Computing Machinery.

Appendix

A Threat Model

Attacker's Goal and Capability. The attacker aims to obtain a model functionally similar to the victim service S_v at a low cost, thereby bypassing expensive training expenses or copyright restrictions. To achieve this, the attacker operates under the following constraints:

- **Knowledge Constraint.** The attacker has only black-box API access to S_v and has no knowledge of the internal structure, parameters, or training data of the victim model Θ_v, nor the details of the watermarking algorithm.
- **Model Extraction as Channel Noise.** The attacker possesses a query dataset $\mathcal{D}_q$ and trains a substitute model Θ_a using the embeddings e_p obtained from API queries as supervisory signals. In our problem formulation, this process constitutes the primary "channel noise." Due to limited training data, structural differences, or distillation configurations, Θ_a is merely an approximation of Θ_v. This "lossy compression" process inevitably disrupts the fine-grained structure of the original representation space, introducing severe distortion and interference to the embedded watermark signal, making the precise recovery of multi-bit information from Θ_a highly challenging.

Defender's Goal and Capability. The defender possesses the original model Θ_v and has exclusive control over watermark generation. Based on this, the defender aims to achieve two key objectives:

- **Watermark Verification.** The defender aims to reliably detect the presence of the watermark signal within a suspicious model Θ_a, thereby distinguishing a pirated model from independent benign models.
- **Precise Source Tracing.** Beyond mere verification, the defender aims to accurately recover the specific multi-bit message m from the pirated model Θ_a, enabling the identification of the specific leakage source.

B Implementation Details of the Residual Adapter

As illustrated in Figure 6, the Watermark Adapter utilizes a non-linear residual architecture to learn the optimal watermark perturbation $\Delta\mathbf{e}$ while maintaining semantic fidelity. The module processes the normalized input embedding $\mathbf{e}_o \in \mathbb{R}^d$ (where $d = 1536$) through a transformation branch consisting of a linear projection ($d \to h$), a ReLU activation followed by a Dropout layer ($p = 0.1$) for regularization, and a second linear projection ($h \to d$). The generated perturbation is added element-wise to the shortcut connection, and the final output is re-projected onto the unit hypersphere via L2 normalization: $\mathbf{e}_p = \text{Normalize}_2(\mathbf{e}_o + \mathcal{A}_\theta(\mathbf{e}_o))$.

To ensure training stability and prevent initial semantic collapse, we employ a "Near-Identity" initialization strategy. The weights of all linear layers are sampled from a normal distribution $\mathcal{N}(0, 0.01)$, and the biases are initialized to zero. This ensures that the adapter initially acts as an identity function ($\mathbf{e}_p \approx \mathbf{e}_o$), allowing the optimization process to introduce the watermark signal guided by the loss of alignment gradually.

Table 4: Statistics of the datasets used in our evaluation. "Avg. Length" denotes the average token count per sample.

Dataset	# Train	# Test	Avg. Length	# Classes
SST-2	67,349	872	54	2
MIND	97,791	32,592	66	18
AG News	120,000	7,600	236	4
Enron Spam	31,716	2,000	35	2

C Supplementary Experimental Analysis

C.1 Dataset Details

Table 4 presents the statistical specifications of the four datasets: SST-2, MIND, AG News, and Enron Spam. These datasets cover a diverse range of sample counts, text lengths, and categories, facilitating a comprehensive evaluation of RShield across distinct textual distributions.

C.2 Hyperparameter Configurations

To ensure the reproducibility of our results, we provide the detailed hyperparameter configurations used in our experiments. Table 5 lists the dataset-specific parameters, including the loss weights ($\lambda_{align}, \lambda_{preserve}$), the trigger word frequency ranges, and the unique watermark messages. These values are optimized per dataset to balance watermark detectability with semantic fidelity. Other shared parameters across all modules—including watermark configuration, adapter training, down-

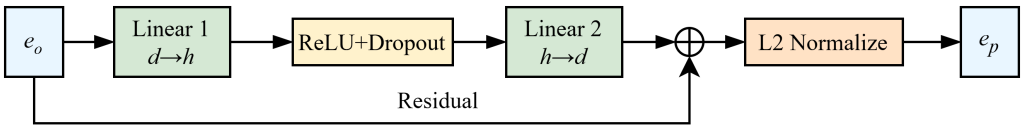


Figure 6: The detailed architecture of the Watermark Adapter.

Table 5: Dataset-specific hyperparameter settings. Trigger frequencies are represented in scientific notation.

Dataset	λ_{align}	$\lambda_{preserve}$	**Trigger Freq. Range**	**Watermark Msg** (m)
SST-2	0.1	5	$[5.0 \times 10^{-6}, 5.0 \times 10^{-5}]$	"sstwt"
MIND	0.001	10	$[5.0 \times 10^{-6}, 5.0 \times 10^{-5}]$	"mindwtm"
AG News	0.001	10	$[1.0 \times 10^{-5}, 2.8 \times 10^{-5}]$	"agnewswtm"
Enron	0.001	10	$[1.4 \times 10^{-4}, 1.6 \times 10^{-4}]$	"enronwtm"

stream tasks, and the copier model—are summarized in Table 6.

C.3 Detailed Evaluation Metrics

We evaluated RShield from three dimensions:

- Watermark Effectiveness: First, we used Symbol Accuracy (SA) and RS Decoding Success Rate to directly quantify the recovery integrity of the watermark information. Additionally, following the multi-watermark verification framework of WARDEN (Shetty et al., 2024), we introduced statistical existence verification. By calculating the distribution shifts (Δ cos and $\Delta l2$) between backdoor samples and benign samples in the direction of the codebook vectors, and utilizing the KS test (Berger and Zhou, 2014) to calculate the p-value, we verified the legal validity of the watermark's existence with statistical significance.

- Semantic Fidelity and Utility: We evaluated semantic perturbation by calculating the cosine similarity between original and watermarked embeddings, and verified model utility by testing the accuracy (ACC) of downstream classification tasks.

- Indistinguishability: We projected the original and watermarked embeddings into a two-dimensional space using PCA to visually evaluate the degree of distribution overlap between the two, thereby measuring the stealthiness of the watermark.

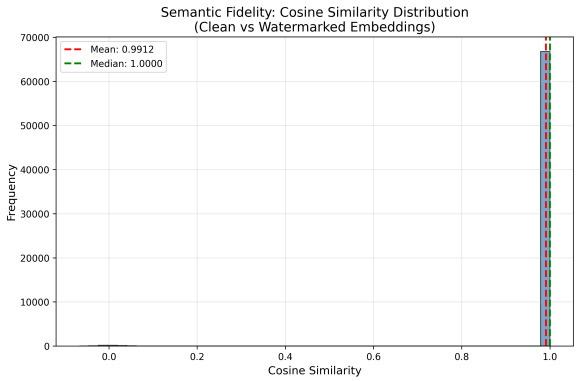
C.4 Detailed Semantic Fidelity Analysis

To complement the summary results in Section 4.3, Figure 7 presents the detailed similarity frequency histograms for all four datasets. Macroscopically, the distributions exhibit extreme unimodal characteristics. The probability density is densely concentrated within the high-confidence interval of $[0.99, 1.0]$, forming a significant dominant peak. Unlike broad global perturbations that would shift the entire distribution bell curve, the vast majority of samples in RShield remain virtually identical to their original counterparts. This geometric compactness confirms that the watermark injection respects the original manifold structure, achieving near-lossless semantic preservation for non-trigger samples.

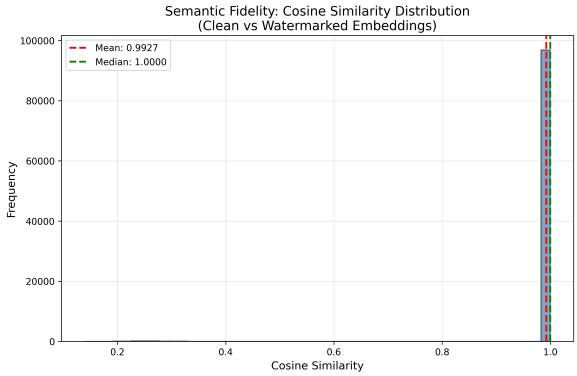
Microscopically, critical statistical observation lies in the subtle discrepancy between the mean (marked by the red dashed line) and the mode (the peak). As shown, the mean is consistently situated slightly to the left of the mode, indicating a distinct left-skewed distribution. This phenomenon serves as the statistical fingerprint of our low-frequency trigger strategy. While the strong watermark injection on trigger samples results in lower cosine similarity (occupying the long tail on the left), their extreme sparsity prevents the formation of any perceptible secondary peak or "island" in the lower range. Consequently, these trigger samples are statistically submerged within the inherent variance (noise) of the original data. This validates the stealthiness of RShield: without a distinguishable secondary cluster, attackers cannot easily isolate watermarked samples through simple threshold-based distribution analysis.

Table 6: Common hyperparameters fixed across all experiments. Parameters are grouped by their respective modules: Watermark configuration, RShield adapter training, downstream task training, and copier training.

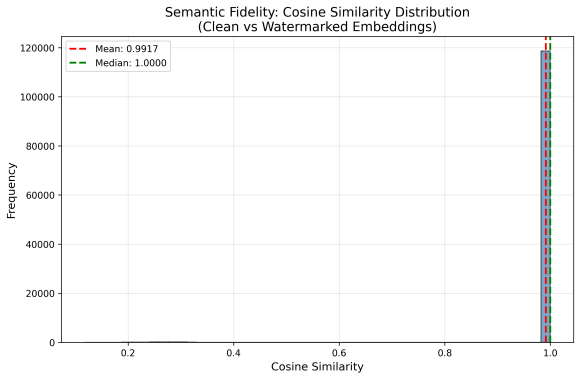
Module / Task	**Hyperparameter**	**Value**
Watermark Configuration	RS Redundant Symbols (N_{sym})	4
	Finite Field Size (p)	257
	Triggers per Symbol	2
Adapter Training (RShield)	Learning Rate	1×10^{-4}
	Batch Size	64
	Epochs	100
	Dropout	0.1
Downstream Task Training	Learning Rate	1×10^{-2}
	Batch Size	32
	Epochs	3
	Dropout	0.2
Copier Training	Learning Rate	5×10^{-5}
	Batch Size	8
	Epochs	3
	Transform Hidden Dim	1536



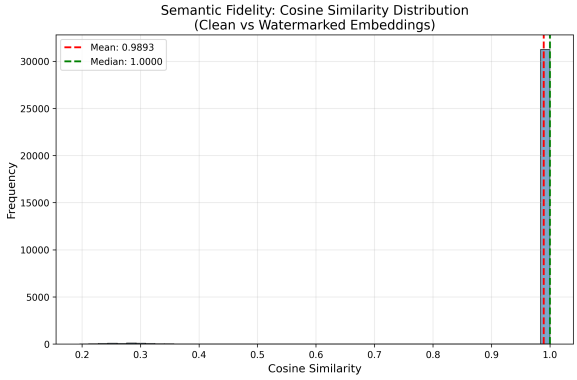
(a) SST-2



(b) MIND



(c) AG News



(d) Enron Spam

Figure 7: Detailed similarity frequency histograms of RShield on four datasets. The red dashed line represents the mean similarity, while the dominant peak represents the mode. The distributions exhibit extreme unimodal characteristics with high concentrations in the [0.99, 1.0] interval.