Information Sciences 678 (2024) 120893

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Defending against similarity shift attack for EaaS via adaptive multi-target watermarking ☆

Zuopeng Yang [a], Pengyu Chen [a,c], Tao Li [b], Kangjun Liu [d], Yuan Huang [e], Xin Lin [a,*]

[a] Institute of Artificial Intelligence, Guangzhou University, China
[b] Department of Automation, Shanghai Jiao Tong University, China
[c] Pazhou Lab, China
[d] Peng Cheng Laboratory, China
[e] Artificial Intelligence R&D Center, CNNC Equipment Technology Development Co., Ltd, China

ARTICLE INFO

Keywords:
EaaS
Similarity shift
Embedding watermarking
Adaptive multi-target watermarking
Copyright protection

ABSTRACT

Large language models have revolutionized natural language processing, leading to the emergence of Embedding as a Service (EaaS). While EaaS facilitates access to advanced embedding models, it also presents challenges in copyright protection. Current research primarily relies on single-target watermarking frameworks, where a predefined vector is integrated as a watermark into text embeddings. However, these approaches are vulnerable to watermark information leakage. To investigate this issue, we introduce the Embedding Similarity Shift Attack (ESSA), an innovative attack algorithm designed to detect trigger instances in single-target watermarking systems by analyzing similarity shifts among constructed reference sentence pairs. Additionally, to defend against such an attack, we propose Adaptive Multi-Target Watermarking (AMT-WM). AMT-WM stands as the pioneering multi-target watermarking method aimed at safeguarding the copyright of EaaS. Specifically, AMT-WM constructs multiple watermarks through the utilization of orthogonal vectors to mitigate selection bias towards a particular vector. Furthermore, it incorporates a randomly selected sentence embedding as the base embedding to enhance the confidentiality of backdoored embeddings. For multi-target watermarking, we implement adaptive watermark injection and validation based on similarity. Comprehensive experiments conducted on various datasets validate the effectiveness of ESSA in trigger detection performance and the efficacy of AMT-WM in copyright protection. Our code will be available soon.

1. Introduction

Within a few years, the landscape of natural language processing (NLP) has been dramatically shaped by the ascendancy of Large Language Models (LLMs). This evolution has given rise to Embedding as a Service (EaaS), a paradigm enabling access to advanced pre-trained embedding models, offering users vector representations of natural language that capture its semantic information. Consequently, this service facilitates various tasks, such as information retrieval [1,2], question answering [3,4], semantic textual similarity [5,6], graph generation [7], HOI detection [8,9], etc. However, the accessibility and convenience provided by EaaS also

☆ This research is partly supported by Guangzhou Basic and Applied Basic Research Scheme (No: 2024A04J3367), and Guangdong Basic and Applied Basic Research Foundation (No: 2023A1515110077).

* Corresponding author.
E-mail address: yzpeng44@gmail.com (Z. Yang).

https://doi.org/10.1016/j.ins.2024.120893
Received 27 February 2024; Received in revised form 6 April 2024; Accepted 3 June 2024

Available online 5 June 2024

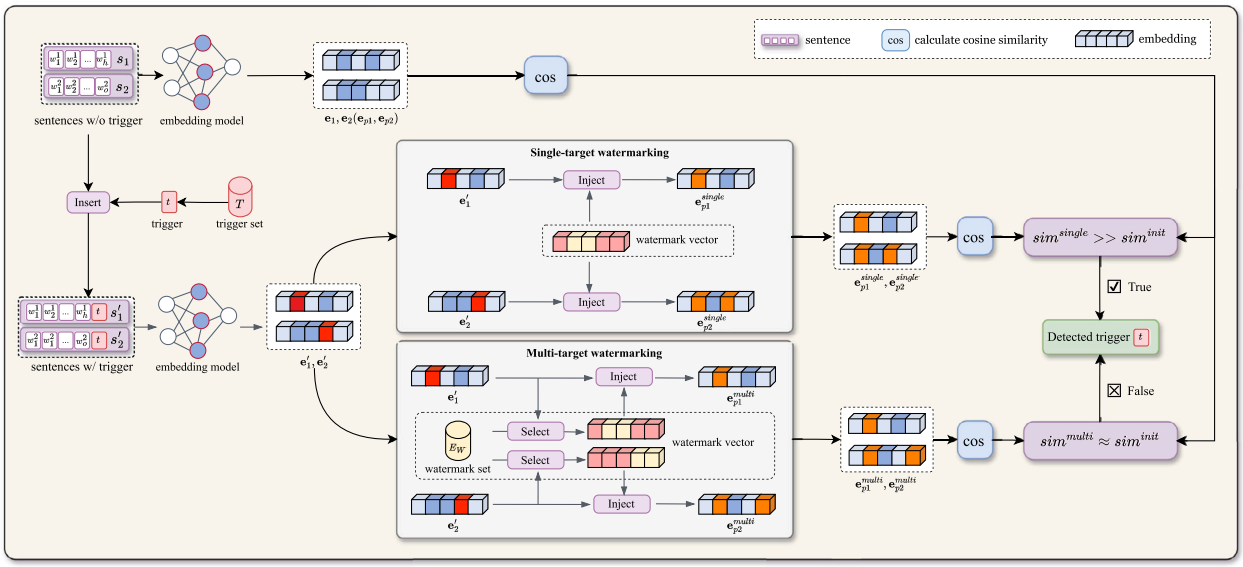


Fig. 1. The risk of trigger leakage of single-target watermarking methods and their difference from our multi-target watermarking method. Single-target watermarking methods integrate a predefined vector as a watermark into the sentence embeddings, whereas our multi-target watermarking method selects different watermarks from a watermark set for each embedding. Hence, the utilization of the same watermark vector in the single-target watermarking method results in greater similarity changes between watermarked vectors compared to the multi-target watermarking method. This consequently leads to the leakage of trigger information in the single-target watermarking approach.

make it vulnerable to model extraction attacks [10,11], a growing concern in the NLP community. Once obtained, these models can be replicated, manipulated, or utilized for malicious purposes, potentially causing financial losses and reputational damage to the original EaaS provider. Thus, ensuring the copyright protection of EaaS is essential in light of its benefits to both text and image processing.

One of the most popular methods for copyright protection is watermarking. While prior watermarking techniques have mainly focused on images, research on text embeddings remains limited. Typically, the embedding model of EaaS is exclusively accessible through its API. Therefore, backdoor-based approaches are more suitable for this particular scenario. For instance, RedAlarm [12] selects a rare token as the trigger and returns a predefined target embedding when a sentence contains the trigger. Additionally, EmbMarker [13] chooses a group of moderately frequent words from a general text corpus to form a trigger set, adjusting the watermarking degree based on the number of triggers in the sentence. However, these approaches have all adopted single-target watermarking, where only one predefined vector serves as the watermark, as shown in Fig. 1. The utilization of the same watermark vector results in increased similarity changes between watermarked vectors. This amplifies the risk of leakage for both triggers and watermark information.

In this paper, we first introduce the Embedding Similarity Shift Attack (ESSA), a novel method designed to detect trigger instances in single-target watermarking systems. Specifically, ESSA begins by constructing reference sentence pairs and subsequently analyzes the degree of cosine similarity shift in these pairs before and after token concatenation to identify triggers. To enhance accuracy, we incorporate multiple rounds of detection. Then, to defend against ESSA, we propose a novel copyright protection method called Adaptive Multi-Target Watermarking (AMT-WM). AMT-WM stands as the pioneering multi-target watermarking method aimed at safeguarding EaaS's copyright. As illustrated in Fig. 1, in contrast to single-target watermarking methods, the core idea of AMT-WM is to construct a set containing multiple watermark vectors and adapt the selection of suitable vectors for each embedding. In detail, we first introduce orthogonal vectors to construct a watermark set, aiming to alleviate selection biases towards a particular vector in subsequent watermark selection processes. Secondly, to enhance the confidentiality of the backdoored embeddings, we also incorporate the embedding of a randomly selected sentence to reduce the distribution discrepancy between watermark vectors and the embedding model's representation space. Finally, to accommodate this design of multi-target watermarking, a novel strategy is devised for adaptive watermark injection and validation based on similarity. Additionally, we theoretically demonstrate the accuracy of this strategy. Therefore, AMT-WM has the capability to enhance the robustness of copyright protection for EaaS. Our main contributions are summarized as follows:

- We introduce ESSA to detect trigger instances in single-target watermarking systems.
- We propose AMT-WM, the first multi-target watermarking method for protecting the copyright of EaaS.
- Comprehensive experiments are conducted to verify the effectiveness of ESSA in trigger detection and AMT-WM in copyright protection.

2. Related works

2.1. Model extraction attacks

Model extraction attacks [14,15] are a type of attack against machine learning models aimed at reconstructing or replicating the victim models by observing their outputs and exploiting their query functionality. Tramèr et al. [16] presents the first work on model extraction and highlights a range of strategies for replicating models across diverse categories. Their work demonstrates the security requirements of model deployment and the necessity of new strategies to counter model extraction attacks. Shi et al. [17] proposes a machine learning attack based on deep learning, aiming to steal model functions and achieve the functions of prime Bayes and SVM classifiers with high accuracy. This approach highlights new security vulnerabilities in online machine-learning algorithms, prompting the need for innovative mitigation strategies. Based on an adversarial model, the Seed-Explore-Exploit framework is proposed by Sethi et al. [18] for active countermeasures against reverse engineering attacks on classifiers. Working in this framework, the model provides valuable information when generating adversarial examples (during the exploitation phase), demonstrating the inherent fragility of classifiers and the lack of relevant explicit information ease of circumvention. Chandrasekaran et al. [19] compares the established area of model extraction and active learning. This demonstrates how recent progress in active learning can facilitate the execution of potent model extraction attacks and the exploration of potential defense mechanisms. Furthermore, Peng et al. [13] emphasizes that online embedding services are vulnerable to model extraction attacks. Stealers can exploit online query text and get the returned embeddings to steal the EaaS model, potentially training their own EaaS and publishing it for revenue. Therefore, it is necessary to protect public embedding services.

2.2. Watermarking attacks

The necessity for watermarks to remain invisible presents a significant challenge in generating watermark embeddings, as it requires a careful balance between invisibility and robustness. Li et al. [20] observes distinct changes in the distribution of poisonous and standard samples. To evade verification of the legitimate owner, stealers can discover the poisonous samples by building a detector, thereby circumventing the legitimate owner's Watermark injection. Regarding the backdoor attack countermeasures algorithm, BKI, as introduced by Chen et al. [21], analyzes the entirety of training data, including poisoned samples, to identify common salient words that are considered potential trigger words. However, its effectiveness is confined to scenarios involving pre-training attacks, making it less effective in the more commonly encountered post-training attack scenarios. Furthermore, ONION [22] uses GPT-2 [23] to calculate source sentence perplexity to find outlier words, i.e., backdoor triggers. Fan et al. [24] proposes using BERTScore [25] to evaluate target semantic changes and examining backward probability changes to remove each constituent token in the generated text to defend against backdoor attacks in natural language generation models. However, stealers usually use extensive datasets as Eaas model input, and these sentences are derived from standard text, which makes it challenging to discover triggers in the training data. Unlike these methods, we introduce the Embedding Similarity Shift Attack (ESSA), which detects trigger instances by analyzing the output embedding similarity shift.

2.3. Watermarking in NLP

Early approaches to watermarking natural text in digital form, as discussed by Atallah et al. [26], embed a small part of the watermark bit string into the syntactic structure of the target sentence. Although meaning-preserving transformations of text sentences (*e.g.*, translation into another natural language) do not damage the watermark, the method is suitable for long-form meaning rather than style-oriented "expository" texts. Li et al. [27,28] proposes to inject specific knowledge into the model through fine-tuning, and require LLM to respond accordingly to this type of knowledge as an embedded watermark. However, it works for LLM generation tasks rather than Eaas services. Furthermore, parameter-based methods, as suggested by Li et al. [29] and Lim et al. [30], involve embedding specific noise into model parameters to facilitate white-box verification. Such methods prove inadequate for black-box access to models potentially compromised by unauthorized users. Gu et al. [31] proposes implanting backdoor triggers and fine-tuning the model to elicit biased responses to specific inputs, thereby enabling detectable verification behaviors. This method may suffer from catastrophic forgetting when fine-tuned on task-specific downstream datasets, leading to the weakening or loss of watermarks, seriously affecting watermark verification during traceability. RedAlarm, introduced by Zhang et al. [12], employs a rare token as a trigger in pre-trained language models, ensuring a predefined target embedding is returned whenever the trigger appears in a sentence. However, this method strongly compromises the performance of model outputs and its effectiveness in downstream tasks. Therefore, Peng et al. [13] proposes to select some specific words based on frequency as the trigger set and determine the watermark strength by calculating the number of trigger words in the sentence. These two works are both based on a single-target watermarking framework, which has the risk of watermark information leakage. To address this issue, we propose a multi-target watermarking method to enhance the concealment and robustness of the watermark while protecting the model copyright of the EaaS.

3. Method

In this section, we initially outline the problem definition of EaaS copyright protection in Section 3.1. Subsequently, we highlight the limitations of existing approaches and introduce the Embedding Similarity Shift Attack in Section 3.2. To counteract this type of attack, we propose a novel method called Adaptive Multi-Target Watermarking in Section 3.3, aiming to enhance the robustness of copyright protection.

3.1. Problem definition

The EaaS provider S_v offers an API service enabling clients to acquire the embedding $\mathbf{e}_v$ of a query s from the model θ_v. The embedding $\mathbf{e}_v$ serves as the semantic representation of a query s. However, without adequate protection, the model θ_v is susceptible to model extraction attacks [19]. Specifically, even unaware of the victim model's structure, an attacker can utilize the queries from a copy dataset D_c and their corresponding embeddings to train an extracted model θ_e. A comparable service S_e can then be provided by the attacker using θ_e at a lower price, which undermines the interests of S_v. Consequently, it's imperative to integrate watermark as a backdoor into the victim model θ_v for copyright protection.

Before delivering $\mathbf{e}_v$ to clients, a watermarking method f is employed to generate processed embedding $\mathbf{e}_p = f(\mathbf{e}_v, s, \mathbf{e}_w)$, where $\mathbf{e}_w$ represents a pre-defined watermark embedding. In this way, if an extracted model is trained based on the processed embeddings, it will also assimilate the watermark information, detectable by the victim service provider. To ensure robust copyright protection, the watermarking method f must satisfy the following requirements:

- The victim EaaS provider should be capable of verifying whether the embeddings offered by S_e contain the pre-defined watermark.
- The injected watermark information should minimally impact the performance of downstream tasks.
- The method needs to be sufficiently robust to prevent easy detection of the watermark.

3.2. Embedding similarity shift attack

Algorithm 1 Embedding Similarity Shift Attack (ESSA).

Require: EaaS provider with watermarking system S_w, Vocabulary V, Number of sentence pairs N, Proportion parameter α
Ensure: Set of detected triggers T_{final}

Initialize list of token sets $T_{sets} \leftarrow \emptyset$
Calculate $K \leftarrow \lceil \text{size of } V \cdot \alpha \rceil$
for $i = 1$ to N **do**
  Construct a pair of distinct sentences (s_1, s_2)
  Embed s_1, s_2 using S_w to get $\mathbf{e}_{p1}, \mathbf{e}_{p2}$
  Calculate initial similarity $sim^{init} \leftarrow \text{cosine}(\mathbf{e}_{p1}, \mathbf{e}_{p2})$
  Initialize difference list $\delta \leftarrow \emptyset$
  for each token t in vocabulary V **do**
    Concatenate t with s_1 and s_2 to form $s_1' = s_1 \| t$, $s_2' = s_2 \| t$
    Embed s_1', s_2' to get $\mathbf{e}'_{p1}, \mathbf{e}'_{p2}$
    Calculate new similarity $sim_t^{new} \leftarrow \text{cosine}(\mathbf{e}'_{p1}, \mathbf{e}'_{p2})$
    Append $(sim_t^{new} - sim^{init})$ to δ
  end for
  Use top-K operation to get $\delta_{\text{topk}} \leftarrow \text{topk}(\delta, K)$
  Initialize set $T_{current} \leftarrow \emptyset$
  for $j = 1$ to K **do**
    Add corresponding token t of $\delta_{\text{topk}}[j]$ to $T_{current}$
  end for
  Add $T_{current}$ to T_{sets}
end for
$T_{final} \leftarrow \bigcap_{X \in T_{sets}} X$
return T

To protect copyright for the Embedding as a Service (EaaS) provider, prior research [12,13] firstly identifies certain unique tokens as triggers. Upon encountering a query text with a trigger, a predefined watermark vector is integrated into the encoded embedding. Consequently, if an unauthorized entity employs these compromised embeddings for model replication, the pattern of the preset watermark vector is assimilated. Nevertheless, this type of approach has a notable limitation: the use of a singular predefined watermark vector increases the risk of detection.

To address this issue, we introduce the **Embedding Similarity Shift Attack (ESSA)**, a method for detecting trigger instances by analyzing embedding similarity shifts. ESSA's process, as detailed in Algorithm 1, involves constructing a pair of sentences (s_1, s_2) and converting them into embeddings $(\mathbf{e}_{p1}, \mathbf{e}_{p2})$ using the EaaS with a watermarking system, S_w. The initial cosine similarity is calculated by:

$$\begin{aligned} sim^{init} &= \text{cosine}(\mathbf{e}_{p1}, \mathbf{e}_{p2}), \\ \text{cosine}(\mathbf{e}_{p1}, \mathbf{e}_{p2}) &= \frac{\mathbf{e}_{p1} \cdot \mathbf{e}_{p2}}{\|\mathbf{e}_{p1}\| \cdot \|\mathbf{e}_{p2}\|}, \end{aligned} \tag{1}$$

where $\| * \|$ represents the Euclidean norm of a vector and "$\cdot$" indicates element-wise multiplication. Subsequently, a token t from the provided vocabulary V is appended to both s_1 and s_2, leading to new embeddings $(\mathbf{e}'_{p1}, \mathbf{e}'_{p2})$ and a recalculated similarity, $sim_t^{new} = \text{cosine}(\mathbf{e}'_{p1}, \mathbf{e}'_{p2})$. For convenience, we use the subscript to denote the concatenated token t. Then, the similarity shift is computed as follows:

$$\delta_t = sim_t^{new} - sim^{init}. \tag{2}$$

Considering that previous studies relied on a unique watermark vector, the similarity change due to watermarking (when a trigger is included) is typically more pronounced than that resulting from mere token concatenation. This difference is key to identifying triggers: if the similarity shift is larger, it suggests the presence of both a token addition and watermarking, whereas a smaller shift indicates only token addition. To identify potential triggers, we first calculate the similarity shift for each token in V:

$$\delta = \{\delta_t \mid t \in V\}. \tag{3}$$

Then we form a set of tokens corresponding to the top K shifts in δ:

$$T_{current} = \{t \mid \delta_t \in \text{topk}(\delta, K)\}, \tag{4}$$

where K is determined by the vocabulary size V and a hyperparameter α. Its calculation process is presented in Algorithm 1, where $\lceil * \rceil$ denotes the operation of rounding down to the nearest integer. Correctly setting α ensures the inclusion of trigger tokens within this set. However, employing a single sentence pair for detection risks false trigger identification. To mitigate this, we employ N sentence pairs across multiple detection rounds. Finally, the intersection T_{final} of results from each round is taken to reduce the number of mistakenly identifying triggers. Additionally, as the vocabulary V is predetermined, ESSA is capable of detecting trigger instances only within this specified vocabulary. Therefore, selecting an extensive vocabulary facilitates the comprehensive detection of all trigger tokens. In short, ESSA finds triggers by analyzing the degree of cosine similarity shift in reference sentence pairs before and after token concatenation and improves accuracy with multiple rounds of detection.

3.3. Adaptive multi-target watermarking

The fundamental challenge with existing methods [12,13] lies in the singularity of the watermark vector. However, a mere increase in the quantity of watermark vectors does not eliminate the risk of selection bias towards a particular vector, making it susceptible to the aforementioned attacks. To address this issue, we propose a novel copyright protection method called **Adaptive Multi-Target Watermarking (AMT-WM)**, as illustrated in Fig. 2. The core idea of AMT-WM involves utilizing orthogonal vectors to construct a watermark set and adapting the selection of suitable vectors for embedding based on similarity. Thus, this approach effectively mitigates the risk of watermark selection bias. Our AMT-WM comprises four steps: trigger selection, watermark embedding construction, watermark injection, and copyright verification.

Trigger Selection. Triggers, selected words from the vocabulary, are employed to initiate a backdoor embedding process in EaaS. The presence of a trigger in the query text entails the incorporation of a watermark into the output embedding. The primary factor influencing trigger selection is word frequency. On one hand, opting for high-frequency words as triggers may result in potential model performance degradation. On the other hand, utilizing low-frequency words could reduce the probability of the extracted model inheriting the backdoor. Therefore, we following EmbMarker [13] to randomly select n words within a moderate-frequency interval as the trigger set $T = \{t_1, t_2, \ldots, t_n\}$.

Watermark Embedding Construction. Since addressing the challenge of single-target watermarking is effectively achieved by augmenting the quantity of watermark vectors, it is crucial to carefully construct the set of watermark vectors. Intuitively, randomly selecting some sentence embeddings as the watermark set is capable of mitigating this problem. However, this approach does not ensure substantial dissimilarity among watermark vectors, potentially resulting in a high similarity of two separate embeddings with different watermarks. This limitation hinders its effectiveness in countering attacks, specifically those associated with ESSA. Furthermore, even if there is considerable diversity among the vectors in the set, there may still be a risk of selection bias towards a particular vector. In other words, despite the presence of multiple watermark vectors in this collection, the copyright protection method f tends to favor the selection of a specific vector. Consequently, it may degrade into the approach used in previous works. In conclusion, the design of watermark vectors must meet the following requirements: 1) Ensuring a moderate dissimilarity among vectors; 2) Eliminating selection bias towards a particular vector.

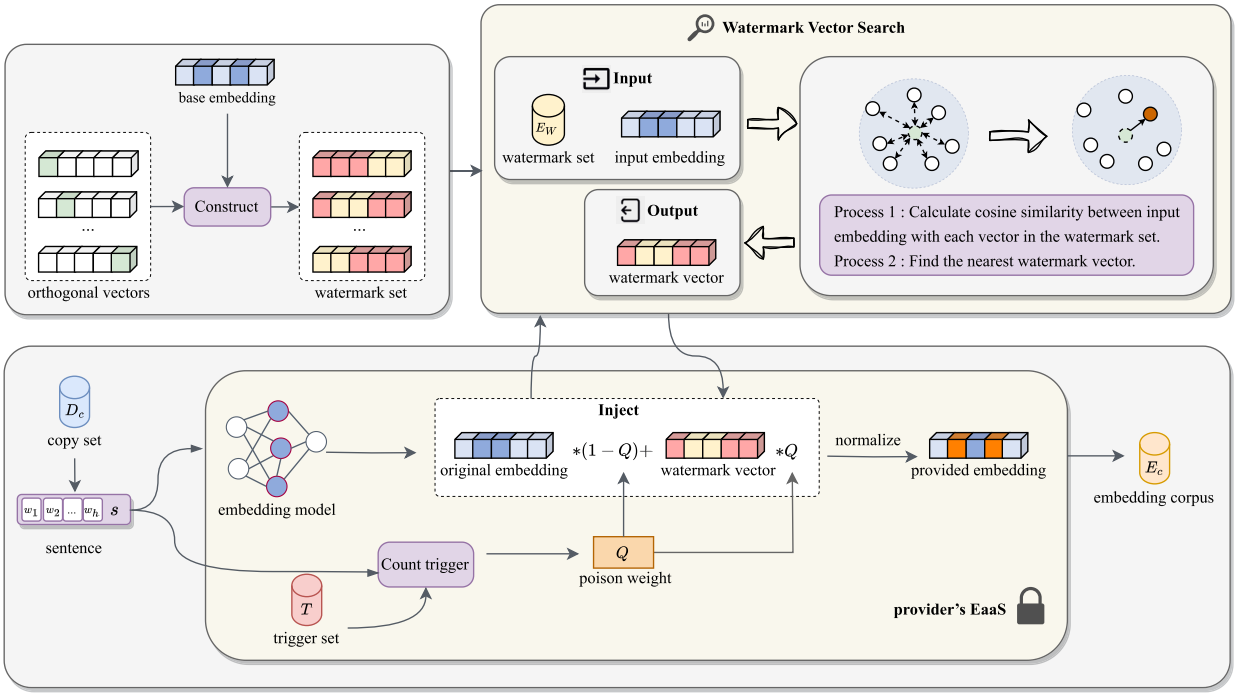
To this end, we construct the watermark set employing orthogonal vectors as a foundation, incorporating the embedding of a randomly selected sentence to enhance the confidentiality of backdoored embeddings. In detail, to mitigate the selection bias towards a particular vector, we generate m orthogonal vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ for watermark construction:

$$\mathbf{v}_i[j] = \begin{cases} 1, & \text{if } j \geq \frac{i \cdot D}{m} \text{ and } j < \frac{i \cdot D}{m} + \frac{D}{2m}, \\ -1, & \text{if } j \geq \frac{i \cdot D}{m} + \frac{D}{2m} \text{ and } j < \frac{(i+1) \cdot D}{m}, \\ 0, & \text{else}, \end{cases} \tag{5}$$

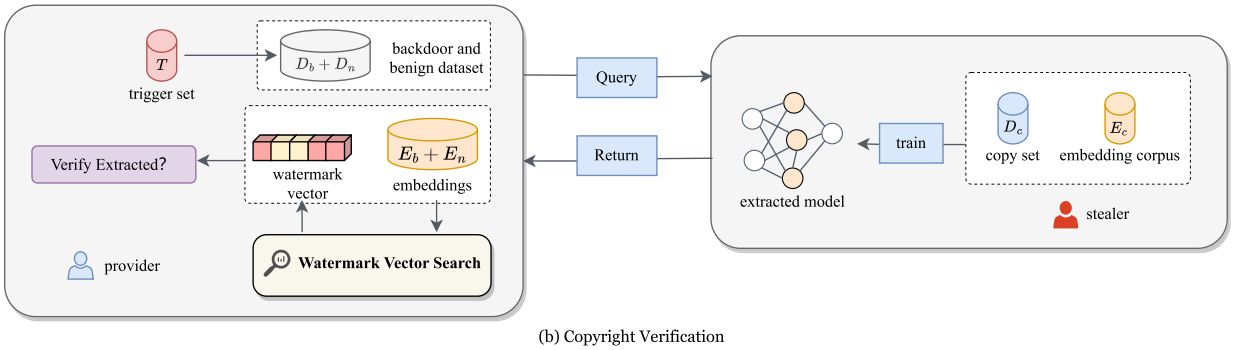
where j denotes the j-th element of $\mathbf{v}_i$. Furthermore, the introduction of orthogonal vectors ensures distinctiveness among the vectors. Despite the increase in the quantity of watermarks, this straightforward design facilitates the extracted model inheriting the backdoor. However, this design simultaneously reduces the confidentiality of the backdoored embeddings. Thus, we utilize the embedding of a randomly selected sentence to refine the watermark construction. Specifically, given a sentence s_b, the EaaS provider S_v offers a base embedding $\mathbf{e}_b \in \mathbb{R}^D$:

$$\mathbf{e}_b = S_v(s_b). \tag{6}$$

Subsequently, the watermark set $E_w = \{\mathbf{e}_{w1}, \mathbf{e}_{w2}, \ldots, \mathbf{e}_{wm}\}$ is constructed by the following steps:



(a) Watermark Injection



(b) Copyright Verification

Fig. 2. The detailed framework of our AMT-WM which involves two significant processes: (a) watermark injection, (b) copyright verification. During the watermark injection phase, AMT-WM constructs a set consisting of multiple watermarks using orthogonal vectors and a base embedding. The base embedding is extracted from the embedding model with a randomly selected sentence as input. Then, AMT-WM adaptively selects an appropriate watermark from the set for the input embedding based on similarity. Subsequently, the watermark is injected into the text embedding, with the injection proportion determined by the number of triggers contained in the sentence. During copyright verification, the provider queries the service provided by the potential stealer with both backdoor text set and benign text set. The corresponding watermarks are then selected for each returned embedding, utilizing the same matching strategy employed in the watermark injection process. Finally, based on both returned embeddings and corresponding watermarks, the provider has the capability to determine whether the model has been stolen.

$$\mathbf{e}'_{wi} = \beta \mathbf{v}_i + \mathbf{e}_b, \tag{7}$$

$$\mathbf{e}_{wi} = \frac{\mathbf{e}'_{wi}}{\|\mathbf{e}'_{wi}\|}, \tag{8}$$

where β is a hyperparameter determining the strength of the orthogonal vectors. Thus, the incorporation of the base embedding introduces prior information from the embedding model into the watermark vectors, enhancing their covertness against model extraction attacks.

Adaptive Watermark Injection. Another core challenge for a multi-target watermarking method is to adaptively select an appropriate watermark vector from the set for each output embedding. Additionally, the selection approach plays a crucial role in defending against ESSA attacks and subsequent verification. Thus, we develop a watermark matching mechanism based on similarity. More specifically, when provided with a text s containing a set of words $W = \{w_1, w_2, \ldots, w_k\}$, we initially transform it into an embedding $\mathbf{e}_v$, where k is the number of unique words in the sentence. Subsequently, we compute the cosine similarity between $\mathbf{e}_v$ and each element in the watermark set, selecting the one with the highest value as the matched watermark for $\mathbf{e}_v$:

$$\mathbf{e}_{wv} = \mathcal{G}(\mathbf{e}_v, E_w), \tag{9}$$

where specific expression of $\mathcal{G}(*,*)$ is as follows:

$$\mathcal{G}(\mathbf{e}_v, E_w) = \arg \max_{\mathbf{e}_{wi} \in E_w} \text{cosine}(\mathbf{e}_v, \mathbf{e}_{wi}). \tag{10}$$

The next step is to integrate the watermark into the embedding. Here, following EmbMarker [13], we control the proportion of the watermark information by a trigger counting function $C(*)$:

$$C(W) = \frac{\min(|W \cap T|, h)}{h}, \tag{11}$$

where h is a hyperparameter deciding the maximum number of triggers to fully activate the watermark. Finally, the watermarked embedding is calculated as follows:

$$\mathbf{e}_p = \frac{(1 - C(W)) * \mathbf{e}_v + C(W) * \mathbf{e}_{wv}}{\|(1 - C(W)) * \mathbf{e}_v + C(W) * \mathbf{e}_{wv}\|}. \tag{12}$$

This processed embedding, equipped with robust copyright protection capabilities, can then be delivered to the users.

Copyright Verification. Since the increase in the quantity of watermark vectors and the implementation of a novel watermark injection algorithm, it is essential to design a new copyright verification algorithm. This algorithm is designed to assist EaaS providers in validating potential copyright infringements by stealers. In contrast, to single-target watermarking methods that only require comparing the embedding with a pre-defined watermark vector, our proposed approach necessitates the prior identification of a matching watermark vector for each embedding. During the watermark injection process, we select the watermark vector with the highest cosine similarity to the original embedding $\mathbf{e}_v$. Consequently, in the validation phase, we employ a similar approach by selecting the watermark vector with the highest cosine similarity to the watermarked embedding $\mathbf{e}_p$. The process is as follows:

$$\mathbf{e}_{wp} = \mathcal{G}(\mathbf{e}_p, E_w). \tag{13}$$

Despite the difference in the type of embeddings used during injection and validation, theoretical evidence supports the consistency between the matched watermark vectors before and after processing, as demonstrated in Section A.1. Employing the watermark matching method $\mathcal{G}$, we can evaluate detection performance following EmbMarker [13]:

$$\text{Score} = \mathcal{M}(D_b, D_n, E_w; S_e, \mathcal{G}), \tag{14}$$

where $D_b = \{[w_1, w_2, \ldots, w_l] \mid w_i \in T\}$ and $D_n = \{[w_1, w_2, \ldots, w_l] \mid w_i \notin T\}$ represent the constructed backdoor text set and benign text set, respectively. The function $\mathcal{M}$, serving as the evaluation metric, encompasses three measures: the difference of averaged cosine similarity [13], the averaged square of L2 distance [13], or the p-value of the Kolmogorov-Smirnov (KS) test [32]. At last, we can comprehensively assess whether the third-party EaaS provider S_e has stolen the model owned by the EaaS provider S_w with the proposed watermarking system based on the scores.

4. Experiments

In this section, we first introduce our experimental settings including training/testing datasets, implementation details, and evaluation metrics in Section 4.1. Then we conduct a comprehensive analysis about the defense capabilities of our method in contrast to state-of-the-art approaches against ESSA in Section 4.2. Furthermore, we present a comparison of the copyright protection performances among these methods in Section 4.3. Finally, ablation studies about the impacts of watermark numbers, orthogonal vectors, and base embedding are performed to validate the effectiveness of the proposed AMT-WM in the subsequent sections.

4.1. Experimental settings

Datasets. Following previous text embedding watermarking research [13], we evaluate the proposed AMT-WM and state-of-the-art methods on four natural language processing datasets: SST2 [33], Enron Spam [34], MIND [35], and AG News [36]. The SST2 dataset, comprising 68,221 sentences, is widely utilized for sentiment classification. The Enron dataset, which contains 33,716 sentences, is specially designed for spam email classification. Additionally, MIND and AG News are two large-scale datasets commonly employed for news classification tasks, comprising 130,383 and 127,600 sentences, respectively.

Implementation Details. This paper contains two important parts: ESSA and AMT-WM. To validate the defense capabilities of the proposed AMT-WM and baseline algorithms against ESSA, we randomly construct 10 sentence pairs. The cosine similarity of the two sentences in each pair is less than 0.65, when using the GPT3 text-embedding-002 API[1] that offered by OpenAI. The α for calculating K is set to 0.25. For a comprehensive evaluation, we also introduce additional text embedding models, including: E5-large-v2 [37] and UAE-large-v1 [38]. E5-large-v2 is trained in a contrastive manner with weak supervision signals from a curated large-scale text pair dataset. UAE-large-v1 effectively mitigates the adverse effects of the saturation zone in the cosine function, leading to a significant performance improvement.

[1] https://api.openai.com/v1/embeddings.

As for validating the protection performance, we follow the same settings with EmbMarker [13]. In detail, we first use the WikiText dataset [39] with $1,801,350$ samples to count word frequencies. The frequency interval of triggers is set to $[0.5\%, 1\%]$. The maximum number of triggers is set to 4, while the size of the triggers set is 20. In this part, there are two tasks: model extraction and classification. For the model extraction task, we apply BERT [40] as the backbone model, connecting with a two-layer feed-forward network, to extract the victim model. And mean squared error (MSE) is used as the loss function. To evaluate the performance of the extracted model, we train a two-layer feed-forward network with cross-entropy loss for the downstream classification task. Since GPT-3 text-embedding-002 is one of the most representative and popular text embedding models, all the experiments for validating the protection performance are based on the embeddings from this text embedding model.

Evaluation Metrics. ESSA is an attack method for EaaS, primarily employed to detect trigger instances of backdoor-based embedding watermark methods. Therefore, to assess a model's vulnerability to ESSA, we count the tokens detected by ESSA along with the triggers among them. If any triggers remain undetected, it indicates that the watermark method can withstand attacks from ESSA. In addition, to validate the copyright protection effectiveness of AMT-WM, we employ three evaluation metrics: the difference of averaged cosine similarity Δ_{cos} [13], the averaged square of L2 distance Δ_{l2} [13], and the p-value of the Kolmogorov-Smirnov (KS) test [32]. Following EmbMarker, these three metrics are measured based on the constructed backdoor text set D_b and benign text set D_n. Δ_{cos}/Δ_{l2} first calculates the cosine similarity/L2 norm between the embeddings of these two text sets and their corresponding watermark vectors, and then compares the discrepancy of the results from these two sets. A large discrepancy indicates a strong evidence of copyright violation by the stealer. The third metric, p-value, serves to compare the distribution of two value sets, predicated on the null hypothesis: *The distance distribution of two cos similarity sets, D_b and D_n, is consistent.* A lower p-value means a higher confidence in supporting the alternative hypothesis. Furthermore, to demonstrate the impact of the watermark on downstream task performance, we report the accuracy of extracted models on downstream classification tasks.

Baseline Methods. We compare our AMT-WM with the following baselines: 1) Original baseline: in this scenario, the original embeddings without any watermarks are utilized to train the copy model. 2) RedAlarm [12]: it is a backdoor-based method, which selects a rare token as the trigger and directly returns a pre-defined target embedding when a sentence contains the trigger. 3) EmbMarker [13]: it is also a backdoor-based method, which chooses a group of moderately frequent words from a general text corpus to form a trigger set, and adjust the watermarking degree based on the number of triggers in the sentence.

4.2. Trigger detection performance comparison

ESSA aims to detect the trigger instances of EaaS with watermarking system. The trigger detection results are reported in Table 1. Firstly, since the Original baseline does not employ watermarks, the number of detected triggers remains close to zero regardless of the number of rounds. When using the text-embedding-002 model, there is one falsely detected trigger. This may be due to its sensitivity, indicating that adding this trigger to a sentence would significantly alter its semantic information. This false detection issue can be addressed by appending new reference sentence pairs. Among single-target watermarking methods, ESSA detects all triggers regardless of the text embedding model used. Compared to the results of one round of detection, the intersection of results from ten rounds has narrowed down the range of triggers by more than tenfold. Additionally, if prior information on word frequency is introduced, the detection range can be further narrowed down. As for AMT-WM, it successfully withstands attacks from ESSA. Even with just one round of results, within such a large scope containing 5433 words, ESSA is unable to detect all triggers in the EaaS adopting the AMT-WM algorithm. Furthermore, since the number of watermark vectors is finite, two different embeddings may select the same watermark. Therefore, each round of detection may result in the detection of some triggers. However, by taking the intersection of results from multiple rounds of detection, the number of detected triggers will also decrease. According to the results, we have the following conclusions: 1) ESSA has the capability to effectively detect trigger instances of single-target watermarking methods. 2) The proposed multi-target watermarking method, AMT-WM, is able to defend against ESSA.

4.3. Watermark detection performance comparison

The core of text embedding watermarking methods lies in their ability to withstand model extraction attacks, accurately detecting the watermark information embedded within copy models. The quantitative results are reported in Table 2, where we have several observations. First, due to RedAlarm's selection of only one rare word as a trigger, the method fails to provide a sufficient number of watermarked training samples during model extraction training. Consequently, it is unable to detect high-confidence watermark information from the copy model. Compared to RedAlarm, EmbMarker utilizes multiple moderate-frequency words to construct the trigger set, resulting in an increase in the number of samples containing watermarks. Additionally, EmbMarker adjusts the proportion of watermark vectors based on the number of triggers contained in the query sentence, further reducing the difficulty of learning the watermark. However, both of the aforementioned methods are single-target watermarking methods, meaning they only utilize a predefined vector as the watermark. In contrast to these single-target watermarking methods, AMT-WM employs orthogonal vectors to construct the watermark set. While increasing the number of watermarks contributes to the learning difficulty, we simplify the design of orthogonal vectors, making the watermark information easier to learn. Therefore, AMT-WM achieves the best watermark detection results. The impact of orthogonal vector design will be detailedly discussed in Section 4.5. Secondly, regarding the accuracy of downstream tasks, all methods achieve results similar to the Original baseline.

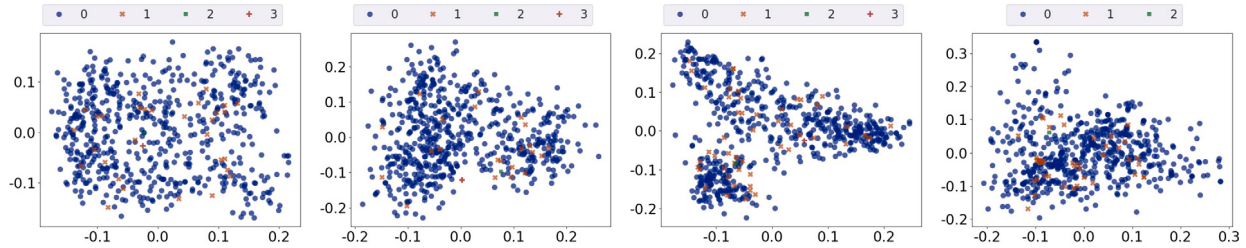
To further examine the confidentiality of backdoored embeddings to the stealer, we employ PCA to visualize the embeddings obtained by the proposed methods. The PCA visualization, depicted in Fig. 3, reveals that backdoored embeddings with triggers have similar distributions to benign embeddings. This demonstrates the watermark confidentiality of our AMT-WM. To effectively

Table 1
Trigger detection results of ESSA across different methods based on the E5-large-v2, UAE-large-v1, and text-embedding-002. The best results are in boldface. The number of triggers is set to 20. "Total Number" indicates the total number of tokens detected by ESSA, while "Trigger Number" denotes the number of triggers among these tokens. When the round number is greater than 1, it indicates the intersection of results obtained from multiple rounds. In single-target watermarking methods (such as Red Alarm and EmbMarker), all triggers are detected, whereas AMT-WM, employing a multi-target watermarking design, successfully withstands ESSA attacks by not detecting all triggers.

Model	Frequency	Method	Trigger Detection Performance (Trigger Number / Total Number)			
			round = 1	round = 3	round = 5	round = 10
E5-large-v2	0%-100%	Original	1/5433	1/1860	0/1021	0/460
		RedAlarm [12]	20/5433	20/1782	20/931	20/358
		EmbMarker [13]	20/5433	20/1867	20/1032	20/471
		AMT-WM	**17/5433**	**15/1861**	**10/1023**	**5/459**
	2%-5%	Original	1/44	1/4	0/2	0/1
		RedAlarm [12]	20/63	20/23	20/21	20/20
		EmbMarker [13]	20/63	20/23	20/22	20/21
		AMT-WM	**17/61**	**15/18**	**10/12**	**5/6**
UAE-large-v1	0%-100%	Original	1/5433	0/1935	0/1258	0/527
		RedAlarm [12]	20/5433	20/1945	20/1271	20/544
		EmbMarker [13]	20/5433	20/1945	20/1271	20/544
		AMT-WM	**2/5433**	**0/1934**	**0/1257**	**0/527**
	2%-5%	Original	1/63	0/14	0/7	0/2
		RedAlarm [12]	20/81	20/33	20/26	20/22
		EmbMarker [13]	20/81	20/33	20/26	20/22
		AMT-WM	**2/63**	**0/14**	**0/7**	**0/2**
text-embedding-002	0%-100%	Original	2/5432	1/1809	1/947	1/419
		RedAlarm [12]	20/5433	20/1821	20/962	20/435
		EmbMarker [13]	20/5433	20/1821	20/962	20/435
		AMT-WM	**6/5433**	**3/1809**	**0/945**	**0/416**
	2%-5%	Original	2/47	1/6	1/1	1/1
		RedAlarm [12]	20/64	20/25	20/20	20/20
		EmbMarker [13]	20/64	20/25	20/20	20/20
		AMT-WM	**6/51**	**3/9**	**0/1**	**0/0**

Table 2
The downstream task accuracy and watermark detection performance of different methods on the SST2, MIND, AG News, and Enron datasets. ↑ means higher metrics are better. ↓ means lower metrics are better. Due to the watermark design of AMT-WM, it achieves the best watermark detection performance while maintaining accuracy comparable to other methods.

Dataset	Method	ACC (%)	Watermark Detection Performance		
			p-value ↓	Δ_{cos}(%) ↑	Δ_{l2}(%) ↓
SST2	Original	93.76 ± 0.19	> 0.34	-0.07 ± 0.18	0.14 ± 0.36
	RedAlarm [12]	93.76 ± 0.19	> 0.09	1.35 ± 0.17	-2.70 ± 0.35
	EmbMarker [13]	93.55 ± 0.19	$< 10^{-5}$	4.07 ± 0.37	-8.13 ± 0.74
	AMT-WM	93.69 ± 0.07	$< 10^{-6}$	**13.52** ± 0.56	**-27.04** ± 1.12
MIND	Original	77.30 ± 0.08	> 0.08	-0.76 ± 0.05	1.52 ± 0.10
	RedAlarm [12]	77.18 ± 0.09	> 0.38	-2.08 ± 0.66	4.17 ± 1.31
	EmbMarker [13]	77.29 ± 0.12	$< 10^{-5}$	4.64 ± 0.23	-9.28 ± 0.47
	AMT-WM	77.17 ± 0.04	$< 10^{-8}$	**8.72** ± 0.18	**-17.45** ± 0.35
AGNews	Original	93.74 ± 0.14	> 0.03	0.72 ± 0.15	-1.46 ± 0.30
	RedAlarm [12]	93.74 ± 0.14	> 0.09	-2.04 ± 0.76	4.07 ± 1.51
	EmbMarker [13]	93.66 ± 0.12	$< 10^{-9}$	12.85 ± 0.67	-25.70 ± 1.34
	AMT-WM	93.65 ± 0.05	$< 10^{-9}$	**39.06** ± 1.13	**-78.12** ± 2.25
Enron Spam	Original	94.74 ± 0.14	> 0.03	-0.21 ± 0.27	0.42 ± 0.54
	RedAlarm [12]	94.87 ± 0.06	> 0.47	-0.50 ± 0.29	1.00 ± 0.57
	EmbMarker [13]	94.78 ± 0.27	$< 10^{-6}$	6.17 ± 0.31	-12.34 ± 0.62
	AMT-WM	94.75 ± 0.12	$< 10^{-6}$	**12.68** ± 0.84	**-25.36** ± 1.67



(a) SST2 (b) MIND (c) AG News (d) Enrom Spam

Fig. 3. Visualization of the provided embedding of AMT-WM on the SST2, MIND, AG News, and Enrom Spam datasets. Different colors denote the number of triggers present in the samples. The visualization illustrates the indistinguishability between backdoor and benign embeddings.

Table 3
ESSA's trigger detection results of AMT-WM with different watermark numbers. The frequency interval is set to $[2\%, 5\%]$.

Watermark Number	Trigger Detection Performance (Trigger Number / Total Number)			
	round = 1	round = 3	round = 5	round = 10
1	20/84	20/28	19/20	19/19
2	6/51	3/9	0/1	0/0
3	6/73	1/8	0/2	0/0
4	13/69	8/13	3/5	2/2
5	10/59	7/11	4/5	4/4

Table 4
The downstream task accuracy and watermark detection performance of AMT-WM with different watermark numbers on the SST2 dataset.

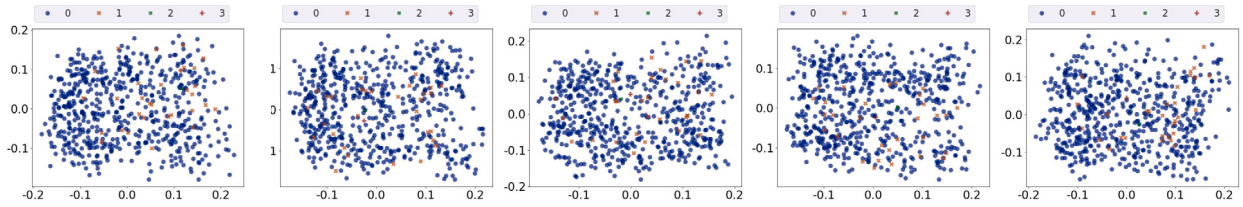
Watermark Number	ACC (%)	Watermark Detection Performance		
		p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓
1	93.49 ± 0.14	$< 10^{-8}$	30.90 ± 0.57	-61.80 ± 1.14
2	93.69 ± 0.07	$< 10^{-6}$	13.52 ± 0.56	-27.04 ± 1.12
3	93.53 ± 0.03	$> 10^{-3}$	2.25 ± 0.30	-4.50 ± 0.60
4	93.51 ± 0.05	> 0.27	0.82 ± 0.10	-1.65 ± 0.20
5	93.69 ± 0.11	> 0.30	0.70 ± 0.11	-1.39 ± 0.21

analyze the role of each component in our method, all subsequent ablation studies are conducted on the SST2 dataset using the text-embedding-002 model.

4.4. Impact of watermark number

In this section, we discuss the impact of watermark numbers. Firstly, the number of watermark vectors affects the defense effectiveness against ESSA, as shown in Table 3. We make the following observations: 1) When the number of watermark vectors is 1, AMT-WM degrades into a single-target watermarking method. Therefore, both one-round and three-round detection successfully identify all triggers. However, in the five-round and ten-round results, one trigger is missed. This could be attributed to the text-embedding-002 model's sensitivity to a specific trigger, leading to substantial semantic alterations when combined with a particular reference sentence. Consequently, this phenomenon may offset the embedding modifications introduced by the watermark. Therefore, in practical usage, it may be advisable to adjust the number of detection rounds. For instance, detection rounds can be terminated early when the total word count in the detection results is relatively low. For example, in the case of one watermark vector in Table 3, the three-round detection result can be used as the final result. 2) When the number of watermarks is greater than 1, AMT-WM successfully withstands attacks from ESSA. However, compared to the results with 2 or 3 watermarks, the defense effectiveness decreases when the number of watermarks is 4 or 5. This is mainly because as the number of watermarks increases, the influence of orthogonal vectors gradually diminishes, leading to a decrease in the distinguishability between watermarks. This phenomenon can be mitigated by adjusting the strength of the vectors. Additionally, compared to the results with 4 watermarks, the results with 5 watermarks are better. This indicates that increasing the number of watermarks strengthens the defense effectiveness. However, the defense effectiveness of AMT-WM is a comprehensive result influenced by various factors such as the number of watermarks and the strength of orthogonal vectors.

Then, the number of watermarks also affects the watermark detection performance against model extraction attacks, as shown in Table 4. Firstly, all experimental results show similar downstream task accuracy. This metric is mainly influenced by the number



(a) Watermark Number = 1 (b) Watermark Number = 2 (c) Watermark Number = 3 (d) Watermark Number = 4 (e) Watermark Number = 5

Fig. 4. Visualization of the provided embedding of AMT-WM with different watermark numbers on the SST2. The visualization illustrates the indistinguishability between backdoor and benign embeddings across various watermark number settings.

Table 5
ESSA's trigger detection results of AMT-WM with different β. The frequency interval is set to [2%, 5%].

β	Trigger Detection Performance (Trigger Number / Total Number)			
	round = 1	round = 3	round = 5	round = 10
0.025	15/64	8/17	5/7	1/1
0.050	14/75	8/17	2/3	1/1
0.075	6/51	3/9	0/1	0/0
0.100	11/62	4/11	3/5	0/0
0.125	13/72	1/5	1/2	0/0

Table 6
The downstream task accuracy and watermark detection performance of AMT-WM with different β on the SST2 dataset.

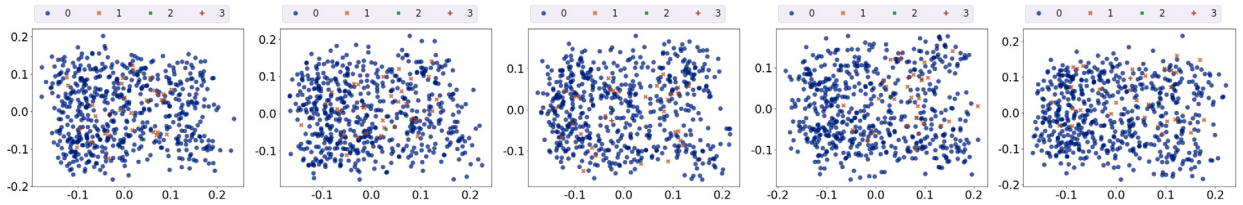
β	ACC (%)	Watermark Detection Performance		
		p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓
0.025	93.65 ± 0.09	$< 10^{-4}$	3.97 ± 0.36	-7.94 ± 0.72
0.050	93.72 ± 0.08	$< 10^{-6}$	9.56 ± 0.99	-19.13 ± 1.99
0.075	93.69 ± 0.07	$< 10^{-6}$	13.52 ± 0.56	-27.04 ± 1.12
0.100	93.53 ± 0.11	$< 10^{-6}$	17.49 ± 0.81	-34.98 ± 1.62
0.125	93.49 ± 0.10	$< 10^{-6}$	19.25 ± 0.98	-38.50 ± 1.96

of triggers. Since all five sets of experiments use the same number of triggers, the results for this metric are comparable. Secondly, compared to the results of EmbMarker in Table 2, AMT-WM achieves better watermark detection performance with a watermark quantity of 1, indicating that the design of watermarks also significantly impacts watermark detection performance. Finally, as the number of watermarks increases, watermark detection performance significantly decreases. This is mainly because with a fixed number of triggers, as the number of watermarks increases, the number of training samples corresponding to each watermark decreases. This prevents the copy model from fully learning the information of each watermark. Therefore, if the number of watermarks is increased, it is necessary to increase the number of triggers correspondingly to reduce the learning difficulty. Additionally, as illustrated in Fig. 4, the backdoor and benign embeddings are indistinguishable, provided by AMT-WT with different watermark numbers.

4.5. Impact of orthogonal vectors

In this section, we analyze the role of orthogonal vectors. We first examine the impact of the strength β of orthogonal vectors and then discuss their design influence. Specifically, the experimental results on the defense against ESSA with different strengths of orthogonal vectors are reported in Table 5. When β is less than 0.075, the detection results are generally inferior to the other three groups. Particularly, in the 10-round detection, one trigger is still detected. This is mainly because as beta decreases, the strength of the introduced orthogonal vectors diminishes, making the watermark vectors more similar to each other. When beta exceeds 0.075, the trigger detection results are comparable. Additionally, concerning model extraction attacks, the watermark detection results under different orthogonal vector strengths are presented in Table 6. As β increases, both Δ_{cos} and Δ_{l2} metrics gradually rise. This indicates that the design of orthogonal vectors in Equation (5) contributes to enhancing the discriminability of watermark vectors. However, when β equals 0.100 or 0.125, the accuracy of downstream tasks is lower than that of the remaining three experimental groups. Moreover, as illustrated in Fig. 5, the backdoor and benign embeddings are indistinguishable, provided by AMT-WT with different β. Therefore, considering these factors collectively, this paper sets β to 0.075 to strike a balance between downstream task accuracy and watermark detection performance.

Next, we discuss the impact of orthogonal vector design. Firstly, to validate the superiority of Equation (5), we devised a complex strategy. Specifically, we randomly selected another sentence's embedding and truncated it according to Equation (5), replacing the



(a) $\beta = 0.025$ (b) $\beta = 0.050$ (c) $\beta = 0.075$ (d) $\beta = 0.100$ (e) $\beta = 0.125$

Fig. 5. Visualization of the provided embedding of AMT-WM with different β on the SST2. The visualization illustrates the indistinguishability between backdoor and benign embeddings across various β settings.

Table 7
ESSA's trigger detection results of AMT-WM with or without employing a complex orthogonal vector design.

w/ Complex Orthogonal Vectors	Trigger Detection Performance (Trigger Number / Total Number)			
	round = 1	round = 3	round = 5	round = 10
✔	20/68	20/23	20/20	20/20
✗	6/51	3/9	0/1	0/0

Table 8
The downstream task accuracy and watermark detection performance of AMT-WM on the SST2 dataset with or without employing a complex orthogonal vector design.

w/ Complex Orthogonal Vectors	ACC (%)	Watermark Detection Performance		
		p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓
✔	93.49 ± 0.18	> 0.09	0.60 ± 0.02	-1.19 ± 0.05
✗	93.69 ± 0.07	$< 10^{-6}$	13.52 ± 0.56	-27.04 ± 1.12

Table 9
ESSA's trigger detection results of AMT-WM with or without using randomly selected embeddings as watermarks. The frequency interval is set to [2%, 5%].

Randomly Select Watermark Embeddings	Trigger Detection Performance			
	round = 1	round = 3	round = 5	round = 10
✔	20/77	20/25	20/20	20/20
✗	6/51	3/9	0/1	0/0

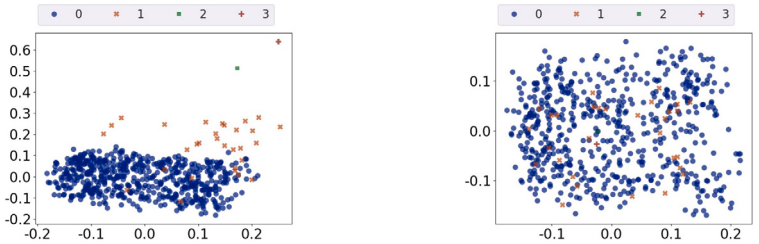
orthogonal vector in the original strategy. The results are shown in Table 7 and Table 8. In summary, utilizing complex orthogonal vectors fails to resist ESSA. This is primarily because the complex orthogonal vectors and base embeddings are both derived from the output of the text embedding model. The combination of the two leads to selection bias, which will be further discussed in subsequent experiments. Moreover, employing complex orthogonal vectors fails to effectively detect watermark information from the copy model. This is because the embeddings used to construct the orthogonal vectors are overly complex, increasing the difficulty of watermark learning. Finally, we randomly selected two embeddings to construct the watermark set, as shown in Table 9. In the results of ESSA, all triggers were detected. This further illustrates that constructing the watermark set only from the outputs of the text embedding model can lead to selection bias.

4.6. Impact of the base embedding

The introduction of the base embedding is aimed at enhancing the confidentiality of backdoored embeddings. Therefore, we conducted comparative experiments with and without the base embedding. As shown in Table 10, the watermark detection performance for model extraction attacks was not significantly affected. However, there was a noticeable separation phenomenon in the visualization results of the embeddings, as illustrated in Fig. 6. Hence, the introduction of the base embedding contributes to increasing the confidentiality of backdoored embeddings.

Table 10
The downstream task accuracy and watermark detection performance of AMT-WM on the SST2 dataset with or without employing the base embedding.

w/ Base Embedding	ACC (%)	Watermark Detection Performance		
		p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓
✗	93.58 ± 0.10	$< 10^{-8}$	29.56 ± 0.58	-59.13 ± 1.16
✓	93.69 ± 0.07	$< 10^{-6}$	13.52 ± 0.56	-27.04 ± 1.12



(a) w/o Base Embedding (b) w/ Base Embedding

Fig. 6. Visualization of the provided embedding of AMT-WM on the SST2 with or without using the base embedding. When the base embedding is not employed, the embeddings become easily distinguishable.

5. Conclusion

In this paper, we addressed the urgent issue of copyright protection for EaaS. We first discover vulnerabilities in existing single-target watermarking methods and, based on this, propose the Embedding Similarity Shift Attack (ESSA). This method detects triggers in such methods by analyzing the similarity shift of reference sentence pairs. To counteract ESSA, we introduce the Adaptive Multi-Target Watermarking (AMT-WM). As a pioneer in multi-target watermarking methods, we use orthogonal vectors to construct a set containing multiple watermarks, addressing the issue of selection bias. Additionally, we introduce base embedding to enhance the confidentiality of backdoored embeddings. Finally, we design an adaptive watermark selection strategy to ensure the watermark injection and validation process for multi-target watermarking. Our experiments demonstrated the effectiveness of ESSA in identifying triggers and the robustness of AMT-WM against ESSA and model extraction attacks. These contributions advance the security of EaaS, emphasizing the significance of ongoing research in safeguarding intellectual property.

Limitations. In this paper, we design a novel multi-target watermarking method, AMT-WM. Our experiments demonstrate the performance of this method in resisting the proposed ESSA attack. However, we have observed that simply increasing the number of watermark vectors does not proportionally enhance the defensive capabilities of the model. This leads to a performance bottleneck for AMT-WM, particularly on certain large-scale EaaS platforms. To overcome this challenge, we intend to refine the watermark vector construction algorithm of AMT-WM in future work, aiming to enhance its defensive capabilities and robustness. Additionally, current watermarking methods for EaaS primarily rely on backdoors. However, if there is a significant difference in data distribution between the dataset used by the stealer and that used by the EaaS provider, the extracted model may not effectively inherit the backdoor watermark information. Therefore, we also plan to explore other types of watermarking methods to mitigate the limitations of backdoor-based approaches.

CRediT authorship contribution statement

Zuopeng Yang: Writing – review & editing, Writing – original draft, Conceptualization. **Pengyu Chen:** Writing – review & editing, Writing – original draft, Software. **Tao Li:** Formal analysis, Data curation. **Kangjun Liu:** Formal analysis, Data curation. **Yuan Huang:** Formal analysis, Data curation. **Xin Lin:** Writing – review & editing, Validation.

Declaration of competing interest

The authors of the manuscript titled "*Defending against similarity shift attack for EaaS via adaptive multi-target watermarking*" collectively declare that no conflicts of interest could be perceived as influencing the results of our research or the interpretation of our findings. This encompasses any financial, personal, or professional relationships. We confirm that the research presented is original, has not been published elsewhere, and is not under consideration for publication by another journal. We have acknowledged all sources of information and data used in the research, and all contributions from others have been duly credited. The manuscript is free from plagiarism, data fabrication, or falsification. We have adhered to all ethical standards of research and publication, including obtaining necessary approvals for any human or animal subjects involved in the study. Any funding sources or sponsorships that have supported this work have been disclosed. We understand the importance of maintaining the integrity of the academic record and are committed to ensuring that our research is conducted and presented ethically and transparently.

Data availability

Data will be made available on request.

Appendix A

A.1. Theoretical proof

In this section, we provide theoretical proof for the proportion in Section 3.3.

Proof. Let $v_1, \ldots, v_n$ be a set of normalized vectors and a_1 be another vector such that the cosine similarity between a_1 and v_1 is maximal. We aim to prove that for any $w \in [0,1]$, the vector $\frac{wa_1+(1-w)v_1}{\|wa_1+(1-w)v_1\|_2}$ has the maximum cosine similarity with v_1.

Given the definition of cosine similarity:

$$\text{cosine\_similarity}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|_2\|\mathbf{v}\|_2}$$

and knowing that v_1 is normalized, i.e., $\|v_1\|_2 = 1$, we consider the cosine similarity between $\frac{wa_1+(1-w)v_1}{\|wa_1+(1-w)v_1\|_2}$ and v_1:

$$\text{cosine\_similarity}\left(\frac{wa_1+(1-w)v_1}{\|wa_1+(1-w)v_1\|_2}, v_1\right) = \frac{(wa_1+(1-w)v_1)\cdot v_1}{\|wa_1+(1-w)v_1\|_2}$$
$$= \frac{w(a_1\cdot v_1)+(1-w)}{\|wa_1+(1-w)v_1\|_2}$$

Applying the triangle inequality for vector norms, we have:

$$\|wa_1+(1-w)v_1\|_2 \le w\|a_1\|_2 + (1-w)\|v_1\|_2 = w\|a_1\|_2 + (1-w)$$

Since both a_1 and v_1 are normalized, the above inequality simplifies to:

$$\|wa_1+(1-w)v_1\|_2 \le 1$$

Thus:

$$\frac{w(a_1\cdot v_1)+(1-w)}{\|wa_1+(1-w)v_1\|_2} \ge w(a_1\cdot v_1)+(1-w)$$

Considering that $w(a_1\cdot v_1)+(1-w)$ linearly interpolates between $a_1\cdot v_1$ and 1, and since $a_1\cdot v_1 \le 1$, it follows that for all $w\in[0,1]$, the expression is always greater than or equal to $a_1\cdot v_1$.

Therefore, it can be concluded that for any $w\in[0,1]$, the vector $\frac{wa_1+(1-w)v_1}{\|wa_1+(1-w)v_1\|_2}$ has the maximum cosine similarity with v_1.

References

[1] S. Wang, R. Koopman, Semantic embedding for information retrieval, in: BIR@ECIR, 2017, pp. 122–132.
[2] Y. Zhu, H. Yuan, S. Wang, et al., Large language models for information retrieval: a survey, arXiv preprint, arXiv:2308.07107.
[3] R. Etemadi, M. Zihayat, K. Feng, et al., Embedding-based team formation for community question answering, Inf. Sci. 623 (2023) 671–692.
[4] M. Esposito, E. Damiano, A. Minutolo, et al., Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering, Inf. Sci. 514 (2020) 88–105.
[5] K. Babić, F. Guerra, S. Martinčić-Ipšić, et al., A comparison of approaches for measuring the semantic similarity of short texts based on word embeddings, J. Inf. Organ. Sci. 44 (2) (2020) 231–246.
[6] H.T. Nguyen, P.H. Duong, E. Cambria, Learning short-text semantic similarity with word embeddings and external knowledge sources, Knowl.-Based Syst. 182 (2019) 104842.
[7] Z. Zhou, M. Shi, H. Caesar, Vlprompt: vision-language prompting for panoptic scene graph generation, arXiv preprint, arXiv:2311.16492.
[8] Y. Liao, A. Zhang, M. Lu, et al., Gen-vlkt: simplify association and enhance interaction understanding for hoi detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20123–20132.
[9] S. Ning, L. Qiu, Y. Liu, et al., Hoiclip: efficient knowledge transfer for hoi detection with vision-language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 23507–23517.
[10] K. Krishna, G.S. Tomar, A.P. Parikh, et al., Thieves on sesame street! Model extraction of bert-based apis, arXiv preprint, arXiv:1910.12366.
[11] A. Yan, T. Huang, L. Ke, et al., Explanation leaks: explanation-guided model extraction attacks, Inf. Sci. 632 (2023) 269–284.
[12] Z. Zhang, G. Xiao, Y. Li, et al., Red alarm for pre-trained models: universal vulnerability to neuron-level backdoor attacks, Mach. Intell. Res. 20 (2) (2023) 180–193.
[13] W. Peng, J. Yi, F. Wu, et al., Are you copying my model? Protecting the copyright of large language models for eaas via backdoor watermark, arXiv preprint, arXiv:2305.10036.
[14] S. Zanella-Béguelin, L. Wutschitz, S. Tople, et al., Analyzing information leakage of updates to natural language models, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 363–375.
[15] X. Gong, Q. Wang, Y. Chen, et al., Model extraction attacks and defenses on cloud-based machine learning models, IEEE Commun. Mag. 58 (12) (2020) 83–89.
[16] F. Tramèr, F. Zhang, A. Juels, et al., Stealing machine learning models via prediction APIs, in: 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 601–618.

[17] Y. Shi, Y. Sagduyu, A. Grushin, How to steal a machine learning classifier with deep learning, in: 2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1–5.
[18] T.S. Sethi, M. Kantardzic, Data driven exploratory attacks on black box classifiers in adversarial domains, Neurocomputing 289 (2018) 129–143.
[19] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, et al., Exploring connections between active learning and model extraction, in: 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1309–1326.
[20] Z. Li, C. Hu, Y. Zhang, et al., How to prove your model belongs to you: a blind-watermark based framework to protect intellectual property of dnn, in: Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 126–137.
[21] C. Chen, J. Dai, Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification, Neurocomputing 452 (2021) 253–262.
[22] F. Qi, Y. Chen, M. Li, et al., Onion: a simple and effective defense against textual backdoor attacks, arXiv preprint, arXiv:2011.10369.
[23] A. Radford, J. Wu, R. Child, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.
[24] X. Sun, X. Li, Y. Meng, et al., Defending against backdoor attacks in natural language generation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 5257–5265.
[25] T. Zhang, V. Kishore, F. Wu, et al., Bertscore: evaluating text generation with bert, arXiv preprint, arXiv:1904.09675.
[26] M.J. Atallah, V. Raskin, M. Crogan, et al., Natural language watermarking: design, analysis, and a proof-of-concept implementation, in: Information Hiding: 4th International Workshop, IH 2001, Proceedings, Pittsburgh, PA, USA, April 25–27, 2001, vol. 4, Springer, 2001, pp. 185–200.
[27] S. Li, K. Chen, K. Tang, W. Huang, J. Zhang, W. Zhang, N. Yu, Functionmarker: watermarking language datasets via knowledge injection, arXiv:2311.09535, 2023.
[28] S. Li, L. Yao, J. Gao, L. Zhang, Y. Li, Double-i watermark: protecting model copyright for llm fine-tuning, arXiv:2402.14883, 2024.
[29] M. Li, Q. Zhong, L.Y. Zhang, et al., Protecting the intellectual property of deep neural networks with watermarking: the frequency domain approach, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 402–409.
[30] J.H. Lim, C.S. Chan, K.W. Ng, et al., Protect, show, attend and tell: empowering image captioning models with ownership protection, Pattern Recognit. 122 (2022) 108285.
[31] C. Gu, C. Huang, X. Zheng, et al., Watermarking pre-trained language models with backdooring, arXiv, 2022.
[32] V.W. Berger, Y. Zhou, Kolmogorov–Smirnov test: overview, Wiley statsref: Statistics reference online.
[33] R. Socher, A. Perelygin, J. Wu, et al., Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.
[34] V. Metsis, I. Androutsopoulos, G. Paliouras, Spam filtering with naive Bayes-which naive Bayes?, in: CEAS, Mountain View, CA, vol. 17, 2006, pp. 28–69.
[35] F. Wu, Y. Qiao, J.-H. Chen, et al., Mind: a large-scale dataset for news recommendation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3597–3606.
[36] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28.
[37] L. Wang, N. Yang, X. Huang, et al., Text embeddings by weakly-supervised contrastive pre-training, arXiv preprint, arXiv:2212.03533.
[38] X. Li, J. Li, Angle-optimized text embeddings, arXiv preprint, arXiv:2309.12871.
[39] S. Merity, C. Xiong, J. Bradbury, et al., Pointer sentinel mixture models, the international conference on learning representations, arXiv preprint, arXiv:1609.07843.
[40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019, pp. 4171–4186.