RegionMarker: A Region-Triggered Semantic Watermarking Framework for Embedding-as-a-Service Copyright Protection

Shufan Yang*, Zifeng Cheng*†, Zhiwei Jiang†,
Yafeng Yin, Cong Wang, Shiping Ge, Yuchen Fu, Qing Gu

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
sfyang@smail.nju.edu.cn, {chengzf,jzw,yafeng}@nju.edu.cn,
{cw,shipingge,yuchenfu}@smail.nju.edu.cn, guq@nju.edu.cn

Abstract

Embedding-as-a-Service (EaaS) is an effective and convenient deployment solution for addressing various NLP tasks. Nevertheless, recent research has shown that EaaS is vulnerable to model extraction attacks, which could lead to significant economic losses for model providers. For copyright protection, existing methods inject watermark embeddings into text embeddings and use them to detect copyright infringement. However, current watermarking methods often resist only a subset of attacks and fail to provide *comprehensive* protection. To this end, we present the region-triggered semantic watermarking framework called RegionMarker, which defines trigger regions within a low-dimensional space and injects watermarks into text embeddings associated with these regions. By utilizing a secret dimensionality reduction matrix to project onto this subspace and randomly selecting trigger regions, RegionMarker makes it difficult for watermark removal attacks to evade detection. Furthermore, by embedding watermarks across the entire trigger region and using the text embedding as the watermark, RegionMarker is resilient to both paraphrasing and dimension-perturbation attacks. Extensive experiments on various datasets show that RegionMarker is effective in resisting different attack methods, thereby protecting the copyright of EaaS.

Introduction

Large language models (LLMs) like GPT (Brown et al. 2020; OpenAI 2023), Qwen (Yang et al. 2024), and LLaMA (Touvron et al. 2023) have demonstrated exceptional capabilities in acting as an embedding model for various NLP tasks (Lee et al. 2024; Li and Li 2024; Fu et al. 2025; Cheng et al. 2025b; Zhao et al. 2025; Zhang et al. 2025; Cao and Zhao 2025). Due to their immense practical value, model providers have begun offering a commercial deployment strategy known as Embedding-as-a-Service (EaaS), which returns embeddings for users' queries and charges a fee. For example, OpenAI offers the text-embedding-3-large API to help users complete various downstream NLP tasks by providing an embedding service.

Despite its effectiveness and convenience, recent research (Liu et al. 2022; Peng et al. 2023; Shetty et al. 2024) indicates that EaaS is vulnerable to model extraction attacks. In such attacks, the stealer can query the provider's model using a text corpus and get the output embedding to train a similar model. In this way, the stealer can deploy a similar EaaS service at a very low cost and cause significant economic losses to model providers. Therefore, it is eager to study how to protect the copyright of EaaS models.

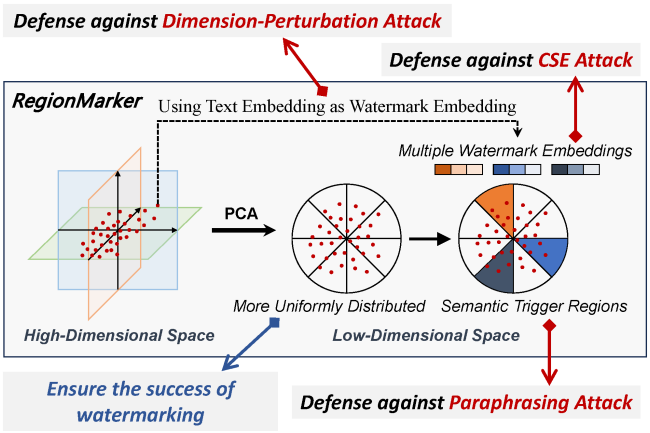


Figure 1: Motivations of the region-triggered semantic watermarking framework against various attacks.

Existing EaaS copyright protection methods (Shetty et al. 2024; Shetty, Xu, and Lau 2024; Peng et al. 2023; Wang et al. 2024c) can be categorized into two types. The first type relies on *trigger words* to embed watermarks, while the second type employs *secret linear transformation* to embed watermarks. (1) For the first type of methods, EmbMarker (Peng et al. 2023) defines a set of trigger words and a watermark embedding to inject watermarks. However, EmbMarker is vulnerable to watermark removal attacks such as CSE (Shetty et al. 2024). To defend against CSE attacks, WARDEN (Shetty et al. 2024) introduces multiple trigger sets and multiple watermark embeddings, and EspeW (Wang et al. 2024c) embeds watermarks into specific dimensions of the output embedding dimensions. However, these methods all rely on *trigger words*, which lack semantic information and are easily removed by *paraphrasing attacks* (Shetty, Xu, and Lau 2024). (2) For the second type of methods, WET applies a *secret linear transformation* to all textual

*Equal contribution.
†Corresponding authors.

Methods	CSE Attack	Paraphrasing Attack	Dimension-perturbation
EmbMarker	✗	✗	✓
WARDEN	✓	✗	✓
WET	✓	✓	✗
EspeW	✓	✗	✓
RegionMarker (Ours)	✓	✓	✓

Table 1: Defense effectiveness of different methods against various attacks.

embeddings, thereby avoiding the use of trigger words and enhancing resistance to paraphrasing attacks. However, the detection of WET assumes that *both the dimensions and their order remain unchanged*, making it highly vulnerable to *dimension-perturbation attacks*. For example, simply removing some dimensions, permuting dimensions, or shifting dimensions (Peng et al. 2023) can evade detection. In summary, as shown in Table 1, current defense methods remain insufficient to counter existing attacks. Even worse, in real-world scenarios, attackers often attempt various attacks to bypass defense. If the defense can be defeated by any one of them, it is deemed ineffective. This highlights *the urgent need for a comprehensive defense method*.

To this end, we introduce a region-triggered semantic watermarking framework named RegionMarker, which uses semantic regions rather than words as triggers, as illustrated in Figure 1. Specifically, RegionMarker defines trigger regions in a low-dimensional semantic space and injects a semantic watermark into the text embedding by considering whether the text lies within these regions. In this process, the low-dimensional semantic space is obtained by applying dimensionality reduction methods such as PCA, and the trigger regions are randomly selected based on a certain ratio. Since the dimensionality reduction matrix and trigger regions are known only to the model provider, it is difficult for attackers to identify and remove the watermarks. Moreover, by embedding watermarks across the entire trigger regions and using the text embedding as the watermark, RegionMarker can effectively defend against paraphrasing and dimension-perturbation attacks. In summary, our method can defend against the three existing types of attacks. Our main contributions are outlined below:

- We first demonstrate that current watermarking techniques for EaaS are unable to defend against existing attacks, and then propose a defense method capable of resisting them.
- We propose a region-triggered semantic watermarking framework that defines trigger regions within a low-dimensional space and injects semantic embedding into text embeddings associated with these regions.
- Extensive experiments on four datasets demonstrate that our approach provides effective defense against existing attacks, ensuring reliable copyright protection for EaaS.

Related Work

Model Extraction Attacks Model extraction attacks (Tramèr et al. 2016; Orekondy, Schiele, and Fritz 2019; Krishna et al. 2020; Wallace, Stern, and Song 2020) involve creating a surrogate model by querying the EaaS without the provider's consent. A stealer sends queries to the provider's model and trains a surrogate model that replicates its functionality based on the feedback from its API (Tramèr et al. 2016; Chandrasekaran et al. 2020; Cheng et al. 2025a; Shen et al. 2025). Liu et al. (2022) discovered that publicly deployed EaaS APIs are vulnerable to imitation attacks. It poses a notable threat to EaaS providers, allowing stealers to rapidly replicate the deployed model with minimal time investment and low financial cost. As a result, stealers could develop a comparable API at a reduced price, infringing on copyrights and disrupting the market (Shen and Tang 2024; Wang et al. 2024a,b, 2025).

Embedding Watermarks Recently, some work (Peng et al. 2023; Shetty et al. 2024; Wang et al. 2024c; Shetty, Xu, and Lau 2024) focus on protecting EaaS from model extraction attacks, which can be broadly categorized into two groups. The first group relies on *trigger words* to inject watermarks. EmbMarker (Peng et al. 2023) first used a watermark embedding and added it to the original embedding of text containing trigger words. The trigger set is randomly selected from moderate-frequency words in a general corpus. WARDEN (Shetty et al. 2024) further demonstrated that the watermark embedding used in EmbMarker can be recovered and removed, and proposed using multiple watermark embeddings to increase the difficulty of recovery and removal. EspeW (Wang et al. 2024c) embeds watermarks in only a subset of dimensions to enhance stealthiness.

However, these methods rely on trigger words to embed watermarks, making them highly vulnerable to paraphrasing attacks. The second category, WET (Shetty, Xu, and Lau 2024), employs a *secret linear transformation* to embed watermarks, aiming to resist paraphrasing attacks. However, the detection of WET assumes that *both the dimensions and their order remain unchanged*, making it highly vulnerable to *dimension-perturbation attacks*. In summary, existing methods remain insufficient for providing comprehensive protection against various attacks.

Methodology

Problem Definition

Model extraction attacks leverage the provided embeddings $\mathbf{e}_p$ based on the provider's model $\mathbf{\Theta}_p$ to train a similar stealer's (extracted) model $\mathbf{\Theta}_s$ at a reduced cost, enabling the stealer (attacker) to offer a competitive EaaS service S_s. To counteract model extraction attacks, the provider's model $\mathbf{\Theta}_p$ injects a predefined watermark t into the original embedding $\mathbf{e}_o$ based on the specified watermarking function f, and subsequently returns the provided embedding $\mathbf{e}_p = f(\mathbf{e}_o, t)$. In this way, the stealer's model $\mathbf{\Theta}_s$ is also watermarked during the training process using $\mathbf{e}_p$. Notably, attackers often employ various strategies to remove the watermark in order to evade detection. Consequently, achieving reliable copyright protection requires that the watermarking function f simultaneously satisfy the following conditions: on one hand, the utility of the provided embeddings $\mathbf{e}_p$ should be comparable to that of the original embedding $\mathbf{e}_o$; on the other hand, the

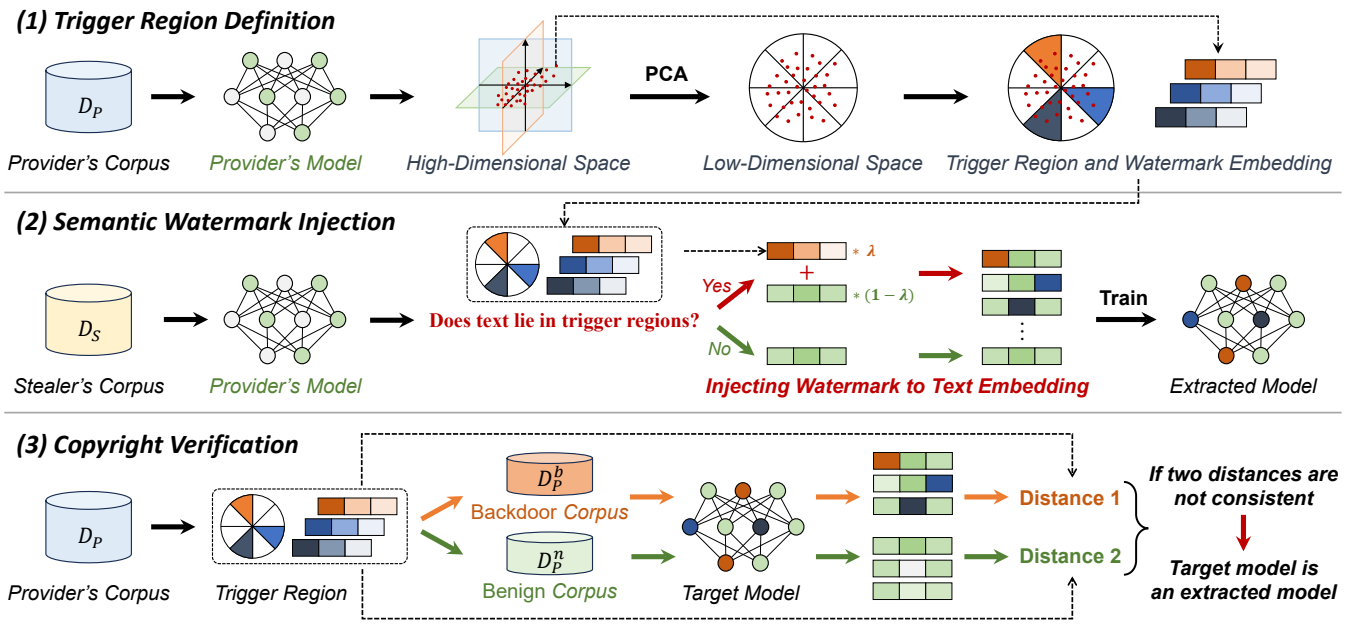


Figure 2: Illustration of the region-triggered semantic watermarking framework.

watermarking function ensures that the output embeddings of the stealer's model Θ_s remain verifiable even after watermark removal.

Region-Triggered Semantic Watermark

To counter existing attacks, we propose a Region-Triggered Semantic Watermarking framework named RegionMarker that uses deeper sentence-level semantics as triggers to inject semantic watermark embeddings. Specifically, the core idea is to use semantic regions instead of shallow words as triggers to effectively defend against paraphrasing attacks, to leverage multiple semantic regions to resist watermark removal attacks, and to use text embeddings as watermarks to defend against dimension-perturbation attacks.

The RegionMarker framework consists of three steps: trigger region definition, semantic watermark injection, and copyright verification, as illustrated in Figure 2. Trigger region definition uniformly divides the low-dimensional space and randomly samples some as trigger regions. Semantic watermark injection assigns a unique watermark embedding to each trigger region, while copyright verification determines whether the model is an extraction model based on the distance difference between benign data and backdoor data.

Trigger Region Definition Due to the sparsity and uneven distribution of data in high-dimensional space, directly partitioning the high-dimensional space is vulnerable to CSE attack and cannot ensure that all texts are evenly distributed across regions. For example, as shown in Figure 3, without dimensionality reduction, the embeddings of the SST-2 dataset are extremely unevenly distributed across the 16 randomly and uniformly divided regions, making it difficult to select effective trigger regions. In contrast, dimensionality reduction significantly alleviates this issue. Therefore, we first use

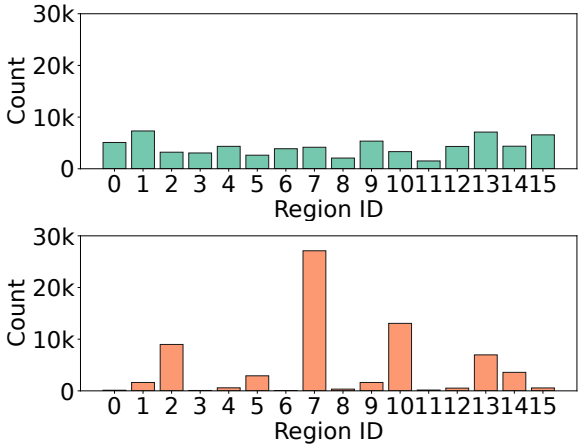


Figure 3: Visualization of the embedding distribution on the SST-2 dataset across 16 randomly and uniformly divided regions: top with dimensionality reduction, bottom without.

dimensionality reduction methods, such as PCA (Maćkiewicz and Ratajczak 1993), to obtain a more compact and uniform d-dimensional semantic space to enhance stealthiness.

After dimensionality reduction, we further use Locality-Sensitive Hashing (LSH) (Indyk and Motwani 1998) to uniformly partition the d-dimensional embedding space into 2^d regions and map similar embeddings to the same region. For each region, we represent it with a d-bit binary LSH signature obtained through random hyperplane projections (Bingham and Mannila 2001), where each hyperplane is mutually orthogonal and is represented by a vector $\mathbf{n}_i$. Similarly, for each text embedding $\mathbf{v}$, we also represent it with a d-bit binary LSH signature to determine which region it falls into. The

i-th bit LSH signature of a text embedding $\mathbf{v}$ is obtained by calculating the dot product between the embedding vector $\mathbf{v}$ and each hyperplane vector $\mathbf{n}_i$:

$$\text{LSH}_i(\mathbf{v}) = \mathbb{1}(\mathbf{n}_i \cdot \mathbf{v} > 0). \tag{1}$$

The d-bit binary LSH signature for embedding $\mathbf{v}$ consists of d corresponding components and determines the region.

$$\text{LSH}(\mathbf{v}) = [\text{LSH}_1(\mathbf{v}), \cdots, \text{LSH}_d(\mathbf{v})] \tag{2}$$

After uniformly partitioning the semantic space, we set a watermark region ratio α and randomly sample $R = \alpha \cdot 2^d$ regions from the entire 2^d regions as the watermark region $A = \{a_1, a_2, ..., a_R\}$.

Semantic Watermark Injection After defining the trigger region, we further inject semantic watermark embeddings into the text embeddings that fall within the trigger region. To enhance the diversity of the watermark embeddings, we assign a unique watermark embedding $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_R\}$ to each trigger region, where $\mathbf{w}_r$ is the embedding of a target sample. Since the embeddings in a region have the same watermark embedding, it prevents attackers from separating the watermark embedding, thus increasing the difficulty of the attack. Specifically, if the text embedding after dimensionality reduction falls within the trigger region a_r, we augment the original embedding $\mathbf{e}_0$ with the corresponding watermark embedding $\mathbf{w}_r$ to obtain the provided embedding $\mathbf{e}_p$, as follows:

$$\mathbf{e}_p = \text{Norm}((1-\lambda) \cdot \mathbf{e}_0 + \lambda \cdot \mathbf{w}_r), \tag{3}$$

where λ is a hyperparameter used to control watermark strength. Since the trigger regions divided by LSH are not intersected, a text embedding has at most one watermark embedding. Moreover, the use of target sample embeddings can effectively resist dimension-perturbation attacks.

Copyright Verification The provider constructs a verification corpus, which includes multiple backdoor corpora $D_p^{b_r}$ and one benign corpus D_p^n, to perform validation under each watermark, where $r \in [1, \cdots, R]$ is the index of the watermark region. The backdoor corpus $D_p^{b_r}$ consists of text embedded in the watermark region a_r, while the benign corpus D_p^n consists of text not embedded in the watermark region.

Compared to benign text, backdoor text is closer to watermark embedding, and this inconsistency forms the basis for verification. We leverage this behavior to verify copyright infringement at each watermark level. Specifically, we calculate the cosine similarity and the squared L_2 distance between the watermark embedding $\mathbf{w}_r$ and the embeddings $\mathbf{e}_i$ of text in $D_p^{b_r}$ and D_p^n to quantify as follows:

$$\begin{aligned} &\cos_{ir} = \frac{\mathbf{e}_i \cdot \mathbf{w}_r}{||\mathbf{e}_i|| \cdot ||\mathbf{w}_r||}, l_{2ir} = \left\| \frac{\mathbf{e}_i}{||\mathbf{e}_i||} - \frac{\mathbf{w}_r}{||\mathbf{w}_r||} \right\|^2, \\ &C_{b_r} = \{cos_{ir} | i \in D_p^{b_r}\}, C_{n_r} = \{cos_{ir} | i \in D_p^n\}, \\ &L_{b_r} = \{l_{2ir} | i \in D_p^{b_r}\}, L_{n_r} = \{l_{2ir} | i \in D_p^n\}, \end{aligned} \tag{4}$$

where C_{b_r} and L_{b_r} represent the sets of cosine similarities and L_2 distances, respectively, between the backdoor text embeddings in the backdoor corpus $D_p^{b_r}$ and the watermark embedding $\mathbf{w}_r$, and C_{n_r} and L_{n_r} represent the sets of cosine similarities and L_2 distances between the benign text embeddings and the watermark embedding $\mathbf{w}_r$.

We then evaluate the detection performance with three metrics. The first two metrics (i.e., Δ_{cos_r} and Δ_{l2_r}) are the difference between average cosine similarity and averaged squared L_2 distances:

$$\begin{aligned} \Delta_{cos_r} &= \frac{1}{|C_{b_r}|} \sum_{i \in C_{b_r}} i - \frac{1}{|C_{n_r}|} \sum_{j \in C_{n_r}} j, \\ \Delta_{l2_r} &= \frac{1}{|L_{b_r}|} \sum_{i \in L_{b_r}} i - \frac{1}{|L_{n_r}|} \sum_{j \in L_{n_r}} j, \end{aligned} \tag{5}$$

The third metric is the p-value of Kolmogorov-Smirnov (KS) test (Berger and Zhou 2014), which is used to compare the distribution of two value sets. The null hypothesis is: *The distance distribution of two cos similarity sets C_{b_r} and C_{n_r} are consistent*. A lower p-value means that there is stronger evidence in favor of the hypothesis.

Finally, we evaluate the p-value under each watermark level and combine the results from a conservative perspective, meaning that if any p-value indicates copyright infringement, we treat it as infringement. The other two metrics are used as supplementary indicators to provide additional evidence for copyright detection.

$$\begin{aligned} \Delta_{cos} &= \max_{1 \le r \le R} \Delta_{cos_r}, \\ \Delta_{l2} &= \min_{1 \le r \le R} \Delta_{l2_r}, \\ p\text{-value} &= \min_{1 \le r \le R} p\text{-value}_r. \end{aligned} \tag{6}$$

Experiments

Datasets and Experimental Settings

Datasets We use SST-2 (Socher et al. 2013), AG News (Zhang, Zhao, and LeCun 2015), Enron (Metsis, Androutsopoulos, and Paliouras 2006), and MIND (Wu et al. 2020) to evaluate. SST-2 is specifically used for sentiment classification. The AG News and MIND datasets are news-based and are used for recommendation and classification tasks. The Enron dataset is utilized for spam classification.

Evaluation Metrics We employ task performance and detection performance to evaluate. For task performance, we construct a multi-layer perceptron (MLP) classifier using EaaS embeddings as input. For detection performance, we employ three metrics: p-value, cosine similarity difference, and squared L_2 distance difference.

Defend Baselines We select **WARDEN** (Shetty et al. 2024), **EspeW** (Wang et al. 2024c), and **WET** (Shetty, Xu, and Lau 2024) as our baselines. **WARDEN** uses multiple watermark embedding and adds it to the original embedding of text containing trigger words. **EspeW** embeds watermarks in only a subset of dimensions. **WET** embeds watermarks using a secret linear transformation matrix and performs watermark detection by applying the corresponding inverse matrix. More details are provided in the Appendix.

Defend	Attack	Task Performance		Detection Performance			COPY?
		ACC.(%)	F_1-score	p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓	
WARDEN	No Attack	93.28±0.09	93.28±0.09	$< 10^{-4}$	4.41±0.42	-8.83±0.84	✓
	+ CSE Attack	89.11±0.41	89.10±0.41	< 0.02	1.16±0.21	-2.32±0.42	✓
	+ Paraphrasing Attack (NLLB)	93.33±0.52	93.32±0.52	> 0.30	-0.03±0.02	0.06±0.03	✗
	+ Paraphrasing Attack (gpt-4o-mini)	92.01±0.14	92.01±0.14	> 0.25	0.01±0.16	-0.02±0.32	✗
	+ Dimension-shift Attack	93.23±0.16	93.23±0.16	$< 10^{-3}$	2.32±0.08	-4.63±0.16	✓
	+ Dimension-reduction Attack	93.06±0.06	93.06±0.06	$< 10^{-5}$	3.14±0.13	-6.28±0.25	✓
EspeW	No Attack	93.46±0.46	93.46±0.46	$< 10^{-10}$	6.46±0.87	-12.92±1.75	✓
	+ CSE Attack	86.73±0.37	86.73±0.37	$< 10^{-11}$	65.11±4.42	-130.23±8.84	✓
	+ Paraphrasing Attack (NLLB)	93.77±0.20	93.77±0.20	> 0.57	0.45±0.05	-0.90±0.11	✗
	+ Paraphrasing Attack (gpt-4o-mini)	93.77±0.48	93.77±0.48	> 0.83	0.31±0.01	-0.62±0.02	✗
	+ Dimension-shift Attack	93.88±0.05	93.88±0.05	< 0.003	1.34±0.03	-2.68±0.06	✓
	+ Dimension-reduction Attack	93.29±0.17	93.29±0.17	$< 10^{-3}$	1.76±0.12	-3.52±0.24	✓
WET	No Attack	93.39±0.05	93.38±0.05	$< 10^{-10}$	89.57±1.17	179.14±2.35	✓
	+ CSE Attack	85.74±1.85	85.74±1.85	$< 10^{-10}$	17.59±0.27	-35.19±0.54	✓
	+ Paraphrasing Attack (NLLB)	93.06±0.06	93.06±0.06	$< 10^{-10}$	89.81±1.39	-179.62±2.77	✓
	+ Paraphrasing Attack (gpt-4o-mini)	93.20±0.27	93.20±0.27	$< 10^{-10}$	89.46±1.15	-178.93±2.30	✓
	+ Dimension-shift Attack	93.46±0.41	93.46±0.41	> 0.46	-0.62±0.85	1.24±1.71	✗
	+ Dimension-reduction Attack	-	-	-	-	-	✗
RegionMarker (Ours)	No Attack	93.23±0.36	93.23±0.36	$< 10^{-4}$	11.90±3.75	-23.80±7.50	✓
	+ CSE Attack	87.87±0.73	87.86±0.73	< 0.05	5.63±2.13	-11.21±4.29	✓
	+ Paraphrasing Attack (NLLB)	93.03±0.22	93.03±0.22	$< 10^{-3}$	10.48±4.24	-20.96±8.48	✓
	+ Paraphrasing Attack (gpt-4o-mini)	92.35±0.11	92.35±0.11	$< 10^{-5}$	7.35±2.21	-14.70±4.41	✓
	+ Dimension-shift Attack	93.73±0.14	93.73±0.14	< 0.003	2.77±0.47	-5.55±0.94	✓
	+ Dimension-reduction Attack	93.29±0.06	93.29±0.06	< 0.004	2.26±0.28	-4.53±0.55	✓

Table 2: Performance of different methods on the SST-2 dataset. ↑ denotes higher metrics are better, and ↓ denotes lower metrics are better from the defender's perspective. In the "COPY?" column, ✓ denotes successful copyright protection, while ✗ denotes a protection failure. A p-value below 0.05 is regarded as a successful copyright protection. The best method consistently achieves successful protection across all attacks.

Attack Methods We comprehensively evaluate all defense methods under **CSE** (Shetty et al. 2024), **paraphrasing attacks** (Shetty, Xu, and Lau 2024), and **dimension-perturbation attacks** (Peng et al. 2023). **CSE** consists of three steps to remove watermarks: it first clusters the embeddings, then selects suspicious samples within each cluster, and finally removes the watermark by eliminating the principal components. We adopt two **paraphrasing attacks**, where NLLB (Costa-jussà et al. 2022) and gpt-4o-mini are respectively used to paraphrase the input texts, and the embeddings of the paraphrased texts are used to replace the original text embeddings. We employ two **dimension-perturbation attacks**: one cyclically shifts the embedding dimensions by 100 positions, and the other truncates the embedding to retain only the first 1024 dimensions. Further details are provided in the Appendix.

Implementation Details We use GPT-3 text-embedding-002 API as the provider's model and BERT (Devlin et al. 2019) as the stealer's model. The learning rate is set to 5e-5, the batch size is 32, and the AdamW (Loshchilov and Hutter 2019) optimizer is used to train the stealer's model. We reproduce the defense methods according to their default settings. For our method, we set the reduced dimension d to 4, the watermark ratio α to 20%, and the watermark strength λ to 0.2. For WARDEN, we set R to 2 and n to 20. For CSE, we set n to 20 and K to 50. For paraphrasing, we generate five different paraphrases for each input text and apply a cosine similarity threshold of 80% to filter out low-quality paraphrases. More details are in the Appendix.

Results of Defense Method

The performance of all methods on SST-2 and Enron is presented in Table 2 and Table 3, respectively. Experimental results show that our method achieves comprehensive robustness against all existing attacks, while existing watermarking strategies are only effective against specific types.

Under CSE attacks, RegionMarker exhibits strong robustness, consistently achieving high detection performance across both datasets. The success of RegionMarker can be attributed to its watermarking strategy based on semantic regions and the use of dimensionality reduction, which together make it more difficult to identify suspicious texts. EspeW and WET, which adopt specialized watermark embedding strategies, also exhibit good resistance to CSE attacks.

Under paraphrasing attacks, the watermark detection performance of WARDEN and EspeW shows a significant decline. Both of these methods rely on *trigger words* for watermark embedding, and paraphrasing the input text multiple times and querying their embeddings effectively dilutes

Defend	Attack	Task Performance		Detection Performance			COPY?
		ACC.(%)	F_1-score	p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓	
WARDEN	No Attack	94.85±0.10	94.85±0.10	$< 10^{-9}$	10.82±0.33	-21.64±0.66	✓
	+ CSE Attack	95.05±0.16	95.05±0.16	> 0.05	1.47±0.93	-2.94±1.87	✗
	+ Paraphrasing Attack (NLLB)	93.68±0.30	93.68±0.30	< 0.01	1.04±0.09	-2.08±0.17	✓
	+ Paraphrasing Attack (gpt-4o-mini)	93.57±0.06	93.53±0.06	< 0.02	0.88±0.04	-1.75±0.08	✓
	+ Dimension-shift Attack	94.72±0.34	94.72±0.34	$< 10^{-10}$	4.39±0.27	-8.78±0.53	✓
	+ Dimension-reduction Attack	93.98±0.23	93.98±0.23	$< 10^{-5}$	3.69±0.17	-7.37±0.33	✓
EspeW	No Attack	94.73±0.23	94.73±0.23	$< 10^{-10}$	7.23±0.35	-14.47±0.70	✓
	+ CSE Attack	95.48±0.28	95.48±0.28	$< 10^{-10}$	47.75±4.13	-95.50±8.26	✓
	+ Paraphrasing Attack (NLLB)	94.80±0.07	94.80±0.07	> 0.49	0.40±0.25	-0.81±0.50	✗
	+ Paraphrasing Attack (gpt-4o-mini)	94.85±0.11	94.85±0.11	> 0.28	0.17±0.27	-0.33±0.54	✗
	+ Dimension-shift Attack	94.77±0.24	94.79±0.22	$< 10^{-3}$	3.84±0.10	-7.69±0.20	✓
	+ Dimension-reduction Attack	94.10±0.20	94.10±0.20	$< 10^{-3}$	3.17±0.07	-6.35±0.13	✓
WET	No Attack	94.35±0.22	94.35±0.22	$< 10^{-10}$	87.10±0.35	-174.19±0.70	✓
	+ CSE Attack	95.23±0.14	95.23±0.14	$< 10^{-10}$	21.45±1.98	-42.90±3.97	✓
	+ Paraphrasing Attack (NLLB)	94.23±0.07	94.23±0.07	$< 10^{-10}$	86.58±0.09	-173.16±0.16	✓
	+ Paraphrasing Attack (gpt-4o-mini)	94.38±0.12	94.38±0.12	$< 10^{-10}$	86.70±0.26	-173.41±0.52	✓
	+ Dimension-shift Attack	94.27±0.33	94.27±0.33	> 0.08	-1.23±0.68	2.47±1.36	✗
	+ Dimension-reduction Attack	-	-	-	-	-	✗
RegionMarker (Ours)	No Attack	94.67±0.18	94.67±0.18	$< 10^{-5}$	11.91± 5.99	-23.81±11.99	✓
	+ CSE Attack	95.55±0.19	95.55±0.19	$< 10^{-4}$	26.27±8.69	-52.54±17.37	✓
	+ Paraphrasing Attack (NLLB)	93.90±0.16	93.90±0.16	$< 10^{-4}$	7.12 ± 3.15	-14.24 ± 6.30	✓
	+ Paraphrasing Attack (gpt-4o-mini)	94.05±0.04	94.02±0.04	$< 10^{-5}$	6.55±1.75	-13.10±3.50	✓
	+ Dimension-shift Attack	94.55±0.04	94.55±0.04	< 0.01	2.33±0.76	-4.67±1.51	✓
	+ Dimension-reduction Attack	94.30±0.10	94.30±0.10	< 0.02	1.96±0.39	-3.91±0.77	✓

Table 3: Performance of different methods on the Enron dataset.

the watermarks. RegionMarker, which leverages semantic regions instead of trigger words for watermarking, demonstrates strong robustness against paraphrasing attacks. This is because paraphrase attacks do not significantly alter the sentence semantics and often remain within the trigger regions, allowing our triggers to persist. WET, which applies a linear transformation watermarking strategy to the entire dataset, also shows strong resistance to paraphrasing attacks.

Under dimension-perturbation attacks, WARDEN, EspeW, and RegionMarker are able to achieve sufficiently good detection performance. This is because all these methods can select a target text and directly set the watermark embedding by computing its embedding with the provider's model. In contrast, WET relies on a linear transformation matrix for watermark detection, which becomes ineffective when the embedding dimensions are shifted. Moreover, once the attacker deletes part of the embedding dimensions, the linear transformation matrix can no longer be applied. Additional results on other datasets are in the Appendix.

Ablation Study

The Necessity of Dimensionality Reduction We explore the necessity of introducing dimensionality reduction. As shown in Table 4, using PCA generally results in better performance compared to not using PCA. According to the results in Figure 3, we speculate that this is due to the uneven distribution of data in the SST2 dataset, and dimensionality reduction helps make the data distribution more uniform.

Under CSE attacks, the performance of RegionMarker without PCA significantly declines, while the performance of RegionMarker with PCA only slightly decreases. This highlights the necessity of introducing dimensionality reduction, and that dividing the space after dimensionality reduction and embedding watermark vectors makes it more covert and harder to break.

The Necessity of Multiple Watermark Embeddings We also investigate the necessity of using multiple watermark embeddings. When assigning the same watermark embedding to all trigger regions, we observe a noticeable drop in detection performance, as shown in Table 4. In particular, under the CSE attack, the watermark becomes ineffective on the SST-2 dataset. This is because a single watermark can be easily identified and removed, whereas multiple embeddings increase the difficulty of removal, highlighting the necessity of using multiple watermark embeddings.

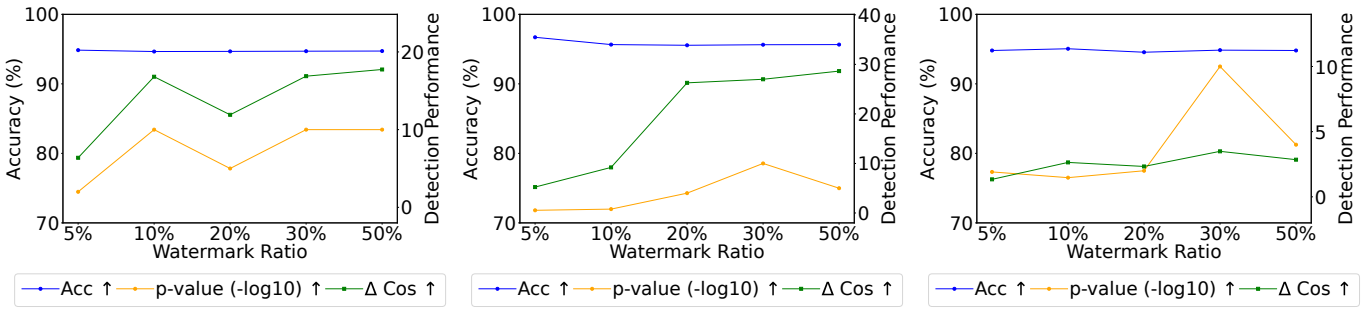
Hyper-parameter Analysis

We explore the effects of watermark region ratio α and dimensionality after PCA on RegionMarker using the Enron dataset. We select two of the most challenging attack strategies, *i.e.*, CSE and dimension-shift attacks, for evaluation.

Figure 4 shows that as the watermark region ratio α increases, the detection performance under different attacks also improves. This is because we also use a conservative detection strategy following (Shetty et al. 2024), where the

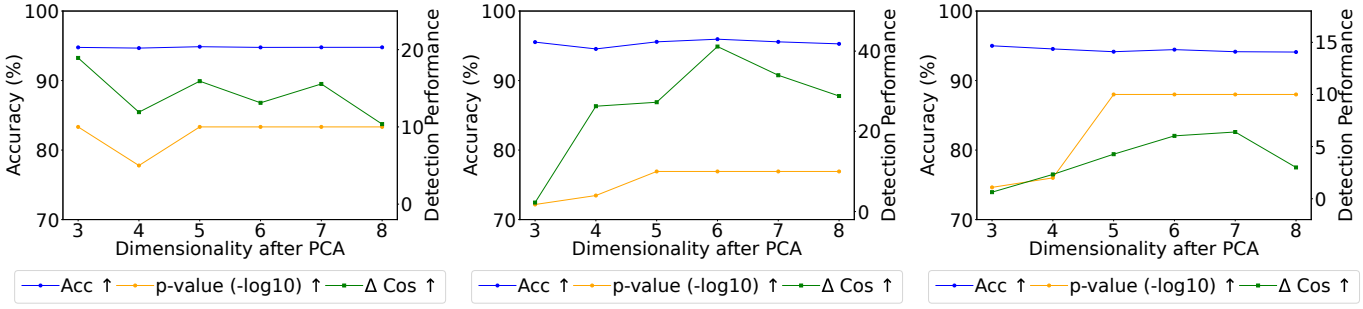
Method	Task Performance		Detection Performance			COPY?
	ACC.(%)	F_1-score	p-value ↓	$\Delta_{cos}(\%)$ ↑	$\Delta_{l2}(\%)$ ↓	
RegionMarker	93.23±0.36	93.23±0.36	$< 10^{-4}$	11.90±3.75	-23.80±7.50	✓
RegionMarker + CSE Attack	87.87±0.73	87.86±0.73	< 0.05	5.63±2.13	-11.21±4.29	✓
RegionMarker$_{w/oPCA}$	93.39±0.16	93.39±0.16	< 0.005	5.46±2.5	-10.91±4.91	✓
RegionMarker$_{w/oPCA}$ + CSE Attack	85.94±0.88	85.93±0.88	> 0.5	1.73±1.53	-3.46±3.06	✗
RegionMarker$_{single\ watermark}$	93.39±0.05	93.39±0.05	$< 10^{-3}$	3.06±0.57	-6.12±1.13	✓
RegionMarker$_{single\ watermark}$ + CSE Attack	86.35±0.01	86.35±0.01	> 0.08	0.20±0.23	-0.40±0.46	✗

Table 4: Ablation study of our proposed method on the SST-2 dataset.



(a) No Attack (b) CSE Attack (c) Dimension-shift Attack

Figure 4: Impact of the proportion of watermarked regions α under different attacks on the Enron dataset.



(a) No Attack (b) CSE Attack (c) Dimension-shift Attack

Figure 5: Impact of dimensionality after PCA under different attacks on the Enron dataset.

best watermark vector from all the watermark vectors is used to determine infringement. However, we maintain a relatively low watermark ratio, namely 20%.

Figure 5 shows that as the dimensionality after PCA increases, the Δ_{cos} exhibits an overall decreasing trend in the absence of attacks, while showing an overall increasing trend under different attacks. This is because, with the increase in regions, the number of watermark embeddings gradually grows, and the sample size in each watermark region decreases, making it difficult for the extraction model to learn the watermark embeddings. However, the increase in the number of watermark embeddings simultaneously raises the difficulty for attackers to successfully remove or bypass the watermarks. Considering these factors, we select a watermark dimension of 4 and a watermark proportion of 20%.

For the watermark strength λ, we follow previous methods and set it to 0.2 to maintain a relatively low level. Although a higher strength can enhance detection performance, it may compromise embedding quality. To balance robustness and fidelity, we adopt this moderate setting. See Appendix for results on other datasets.

Conclusion

We first reveal that current watermarking techniques for EaaS are unable to defend against existing attacks. To this end, we propose a region-triggered semantic watermarking framework, which utilizes semantics rather than trigger words as triggers, and is capable of effectively defending against existing attack strategies. Experiments show that our defense method provides comprehensive and effective defense against existing attacks.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. This work is supported by the JiangSu Natural Science Foundation under Grant No. BK20251989; the National Natural Science Foundation of China under Grants Nos. 62172208, 62441225, 61972192; the Fundamental Research Funds for the Central Universities under Grant No. 14380001. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Berger, V. W.; and Zhou, Y. 2014. Kolmogorov–smirnov test: Overview. *Wiley statsref: Statistics reference online*.

Bingham, E.; and Mannila, H. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001*, 245–250.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Cao, L.; and Zhao, J. 2025. Pretraining on the Test Set Is No Longer All You Need: A Debate-Driven Approach to QA Benchmarks. In *Second Conference on Language Modeling*.

Chandrasekaran, V.; Chaudhuri, K.; Giacomelli, I.; Jha, S.; and Yan, S. 2020. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, 1309–1326.

Cheng, Z.; Gan, J.; Jiang, Z.; Wang, C.; Yin, Y.; Luo, X.; Fu, Y.; and Gu, Q. 2025a. Steering When Necessary: Flexible Steering Large Language Models with Backtracking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Cheng, Z.; Wang, Z.; Fu, Y.; Jiang, Z.; Yin, Y.; Wang, C.; and Gu, Q. 2025b. Contrastive Prompting Enhances Sentence Embeddings in LLMs through Inference-Time Steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 3475–3487.

Costa-jussà, M. R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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, NAACL-HLT 2019*, 4171–4186.

Fu, Y.; Cheng, Z.; Jiang, Z.; Wang, Z.; Yin, Y.; Li, Z.; and Gu, Q. 2025. Token Prepending: A Training-Free Approach for Eliciting Better Sentence Embeddings from LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, 3168–3181.

Indyk, P.; and Motwani, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 604–613.

Krishna, K.; Tomar, G. S.; Parikh, A. P.; Papernot, N.; and Iyyer, M. 2020. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations*.

Lee, C.; Roy, R.; Xu, M.; Raiman, J.; Shoeybi, M.; Catanzaro, B.; and Ping, W. 2024. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *CoRR*, abs/2405.17428.

Li, X.; and Li, J. 2024. BeLLM: Backward Dependency Enhanced Large Language Model for Sentence Embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024*, 792–804.

Liu, Y.; Jia, J.; Liu, H.; and Gong, N. Z. 2022. Stolenencoder: stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2115–2128.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019*.

Maćkiewicz, A.; and Ratajczak, W. 1993. Principal components analysis (PCA). *Computers & Geosciences*, 19(3): 303–342.

Metsis, V.; Androutsopoulos, I.; and Paliouras, G. 2006. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, 28–69. Mountain View, CA.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4954–4963.

Peng, W.; Yi, J.; Wu, F.; Wu, S.; Zhu, B. B.; Lyu, L.; Jiao, B.; Xu, T.; Sun, G.; and Xie, X. 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)*, 7653–7668.

Shen, F.; Jiang, X.; He, X.; Ye, H.; Wang, C.; Du, X.; Li, Z.; and Tang, J. 2025. Imagdressing-v1: Customizable virtual dressing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 6795–6804.

Shen, F.; and Tang, J. 2024. Imagpose: A unified conditional framework for pose-guided person generation. *Advances in neural information processing systems*, 37: 6246–6266.

Shetty, A.; Teng, Y.; He, K.; and Xu, Q. 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)*, 13430–13444.

Shetty, A.; Xu, Q.; and Lau, J. H. 2024. WET: Overcoming Paraphrasing Vulnerabilities in Embeddings-as-a-Service with Linear Transformation Watermarks. *arXiv preprint arXiv:2409.04459*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, 601–618.

Wallace, E.; Stern, M.; and Song, D. 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)*, 5531–5546.

Wang, C.; Deng, Z.; Jiang, Z.; Shen, F.; Yin, Y.; Gan, S.; Cheng, Z.; Ge, S.; and Gu, Q. 2025. Advanced Sign Language Video Generation with Compressed and Quantized Multi-Condition Tokenization. *arXiv preprint arXiv:2506.15980*.

Wang, C.; Tian, K.; Guan, Y.; Shen, F.; Jiang, Z.; Gu, Q.; and Zhang, J. 2024a. Ensembling diffusion models via adaptive feature aggregation. *arXiv preprint arXiv:2405.17082*.

Wang, C.; Tian, K.; Zhang, J.; Guan, Y.; Luo, F.; Shen, F.; Jiang, Z.; Gu, Q.; Han, X.; and Yang, W. 2024b. V-express: Conditional dropout for progressive training of portrait video generation. *arXiv preprint arXiv:2406.02511*.

Wang, Z.; Wu, B.; Deng, J.; and Yang, Y. 2024c. ESpeW: Robust Copyright Protection for LLM-based EaaS via Embedding-Specific Watermark. *arXiv preprint arXiv:2410.17552*.

Wu, F.; Qiao, Y.; Chen, J.-H.; Wu, C.; Qi, T.; Lian, J.; Liu, D.; Xie, X.; Gao, J.; Wu, W.; et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 3597–3606.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *CoRR*, abs/2407.10671.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhang, X.; Zhao, J.; Yang, Z.; Zhong, Y.; Guan, S.; Cao, L.; and Wang, Y. 2025. UORA: Uniform Orthogonal Reinitialization Adaptation in Parameter Efficient Fine-Tuning of Large Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11709–11728.

Zhao, J.; Zhang, X.; Li, J.; Niu, J.; Hu, Y.; Min, E.; and Penn, G. 2025. Tiny Budgets, Big Gains: Parameter Placement Strategy in Parameter Super-Efficient Fine-Tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 6326–6344.