Modeling and Simulation 建模与仿真, 2025, 14(4), 172-180
Published Online April 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.144276

Hans汉斯

面向大语言模型的黑盒文本水印算法

姚奕辰，熊　成，周元鼎，韩彦芳

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月8日；录用日期：2025年4月1日；发布日期：2025年4月10日

摘　要

为应对大语言模型快速发展及其API接口日常生活使用带来的机器生成文本内容检测问题，本文提出了一种面向大语言模型生成文本的黑盒文本水印算法。该算法使用强化学习技术，通过设计的水印嵌入状态、水印嵌入智能体和水印嵌入奖励三个部分进行文本水印嵌入的训练，使得方法能够在水印可检测性、文本语义一致性和抗常规文本水印攻击上均表现出色。与其他算法相比，所提方法在平均准确率达到94.22%的同时，仍能保持更优的文本质量。所提出的算法不仅提升了文本水印的嵌入质量，也为保护模型使用和知识产权保障提供了支持。

关键词

机器生成文本检测，文本水印，强化学习，文本质量

Black-Box Text Watermarking Algorithm towards Large Language Models

Yichen Yao, Cheng Xiong, Yuanding Zhou, Yanfang Han

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 8th, 2025; accepted: Apr. 1st, 2025; published: Apr. 10th, 2025

Abstract

To address the detection issue of AI-generated text content arising from the rapid development of large language models and the daily usage of their API interfaces, this paper proposes a black-box text watermarking algorithm for text generated from the API of large language model. The algorithm employs reinforcement learning techniques to train text watermark embedding through three parts: watermark embedding statement, watermark embedding agent, and watermark embedding reward, which enables the good performance in watermark detectability, text semantic

文章引用: 姚奕辰, 熊成, 周元鼎, 韩彦芳. 面向大语言模型的黑盒文本水印算法[J]. 建模与仿真, 2025, 14(4): 172-180.
DOI: 10.12677/mos.2025.144276

consistency, and resistance to conventional text watermarking attacks. The proposed algorithm has better text quality than other algorithms with an average detection accuracy of 94.22%. The proposed algorithm not only improves the embedding quality of text watermarking, but also provides significant support for protecting model usage and intellectual property rights.

Keywords

AI-Generated Text Detection, Text Watermarking, Reinforcement Learning, Text Quality

Copyright © 2025 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

随着国内外大语言模型(LLMs)的快速发展，它们在大多数领域中的表现已逐步接近甚至超越人类水平[1] [2]。大语言模型 API 的收费使用进一步推动了其在内容生成领域的广泛应用。然而，其高效创建类人文本的能力也引发了其可能被滥用于恶意目的的担忧[3]，例如虚假信息传播[4]、网络钓鱼攻击[5]以及学术欺诈[6] [7]等。这些问题突显了开发可靠方法以检测或验证文本内容是否为 AI 生成的迫切性[8]。

目前，针对大语言模型生成内容的检测方法主要分为被动检测和主动检测两类。被动检测无需访问模型内部结构，主要通过收集大量数据训练文本分类器[9] [10]或对比机器生成与人类书写文本特征[11] [12]来鉴别生成内容。然而，这类方法在面对快速发展的大语言模型时存在一定局限性。文本分类器虽然在面对与训练数据特征近似的领域内表现优异，却容易因过拟合的问题在跨领域中性能迅速下降，且容易受到对抗样本攻击导致分类出错[13]或对非英语母语作者产生分类偏见[14]。随着人类创作与 AI 生成文本之间的界限日益模糊，被动检测方法的有效性正在不断受到挑战。相比之下，主动检测通过在生成内容中嵌入可识别的水印，为内容认证与溯源提供了一种解决方案。传统的文本水印技术需要语言学家人工干预，缺乏鲁棒性，且往往需要原始文本来提取或检测水印，在网络传输和对抗性场景中容易失效[15]。目前针对大语言模型的文本水印技术主要分为两类，一类为在文本生成过程中修改文本内容以嵌入文本水印的白盒水印方式[16] [17]，该方法具有更高的准确性和可解释性，但在只有黑盒语言模型可用的现实场景中，基于 API 开发的第三方无法访问模型参数嵌入自身的水印用于检测认证。另一类为黑盒水印方式，在嵌入水印过程中不需要访问 LLM 模型参数，对已经生成后的文本进行水印嵌入。现有的方法在检测过程中往往需要提供原始文本，同时嵌入水印后的文本与原文的语义一致性难以保证。此外，在保证文本水印不可见性的情况下，现有方法检测过程的准确率和鲁棒性难以做到平衡。

针对上述问题，我们设计了一种基于强化学习训练的文本水印算法，可以针对大语言模型 API 输出的文本内容进行文本水印的嵌入，并且能够平衡文本水印的隐蔽性与鲁棒性。实验证明，我们的方法可以抵抗删词、替换、逆序等文本水印攻击方式的同时，与传统方法相比，该算法不仅提升了水印的隐蔽性与安全性，还能有效满足开放和闭源大语言模型环境中的复杂应用需求。本文的主要贡献如下：

1) 提出了一种可以兼容大语言模型 API 应用场景的文本水印算法。该算法运用强化学习算法使得整体算法在嵌入文本水印过程中同时注意到不可感知性和安全性。与现有方法相比，有拥有同样水印性能情况下，可以获得更好的文本语言质量。

2) 提出一种综合的奖励函数机制，可以同时考虑文本相似性，信息熵相似性和水印可检测性，该设计可以促使模型训练文本水印嵌入时考虑到多个指标，得到更好的文本水印嵌入效果。

3) 文本水印嵌入的方法不受限于训练数据和模型参数是否开源，具有较广的应用场景。

2. 所提方法

如图 1 所示，本文所提出的面向大语言模型 API 黑盒水印算法主要由水印嵌入状态(statement)，水印嵌入智能体(agent)和水印嵌入奖励(reward)三部分组成。首先将待嵌入水印的文本输入到水印嵌入状态中，该部分负责对文本进行预处理和并对文本内容编码。水印嵌入状态将为水印嵌入智能体提供了待嵌入文本的语义信息、候选词以及隐秘信息，以便于其根据当前水印嵌入状态信息适当选择候选词进行文本水印的嵌入。最终通过水印嵌入奖励部分对水印嵌入智能体进行基于强学学习的参数更新优化。为了提升水印嵌入后的文本质量，本文在训练的过程中设计了信息熵对比模块和相似度对比模块，与水印检测模块结合后，便于水印嵌入智能体同时考虑水印的可检测性和隐蔽性，提升整体文本水印的性能。

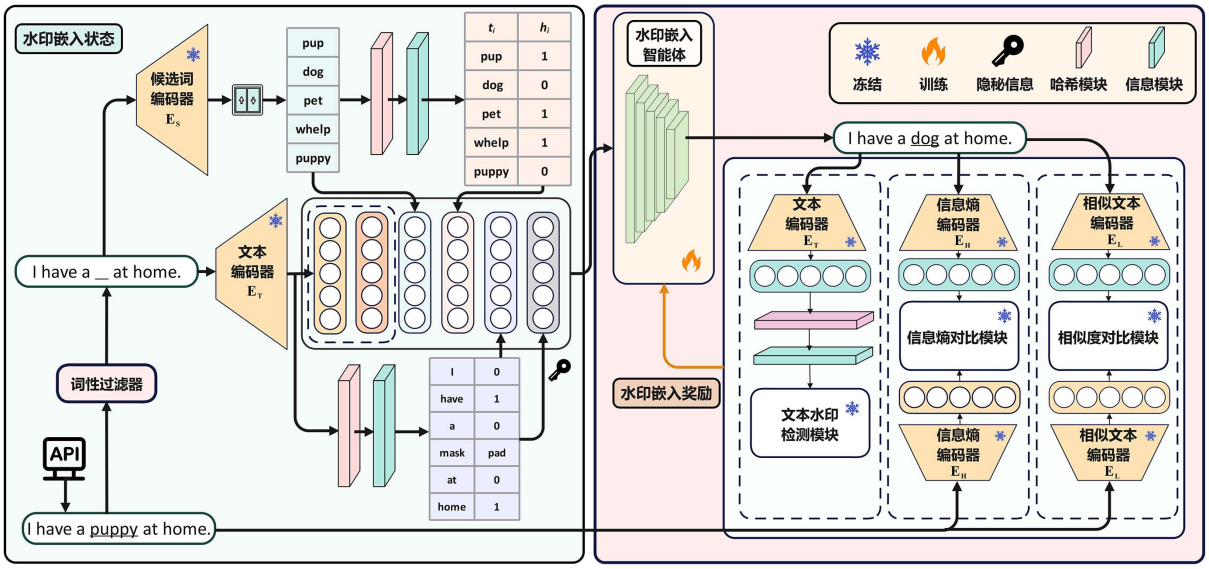

Figure 1. Overall framework of our method
图 1. 本文方法模型框架

2.1. 水印嵌入状态

对于输入的原始文本 T_o，水印嵌入状态为水印嵌入智能体提供了文本综合环境信息 $\boldsymbol{s}_t=\left[\boldsymbol{x}_a,\boldsymbol{x}_t,(\boldsymbol{c}_1,\boldsymbol{b}_1),\cdots,(\boldsymbol{c}_5,\boldsymbol{b}_5),(\boldsymbol{c}_{pad},\boldsymbol{b}_{pad}),\boldsymbol{b}_c\right]$。

对于输入的文本，首先通过词性过滤器获取适合的编码位置，这一步是可以被调整的，其主要目的是避免专有名词、代词、介词等词语被替换导致语言质量上的影响。我们使用 BERT (bidirectional transformers) [18]作为文本编码器 E_T，对整体文本和被替换位置的 token 进行文本编码，得到整体文本信息 $\boldsymbol{x}_a$ 和当前可替换位置的长距离依赖关系信息 $\boldsymbol{x}_t$，两者的信息有助于水印嵌入智能体更好地理解当前位置的语境和语义复杂性。原始文本 T_o 通过文本编码器 E_T 后，对于任意第 i 个 token 都有编码向量 $\boldsymbol{t}^{(i)}$，包含该 token 的抽象文本特征。我们将每个 token 的编码向量传入到我们的哈希模块，得到对应的哈希比特值如公式所示：

$$h_i=h\left(\boldsymbol{t}^{(i)}\right)\bmod 2 \tag{1}$$

其中 $h(\cdot)$ 为 SHA256 函数，对得到每个哈希序列值进行模 2 操作得到 0 或 1 的哈希比特值。由于哈希函

数的高度均匀和不可预测特性，对于没有水印的文本，其 0 或 1 比特的个数应接近于随机分布。接着将哈希比特值传入信息模块，将当前 token 与前一个 token 的哈希比特值进行异或操作后生成隐秘信息比特，如公式所示：

$$b_i = h_i \oplus h_i \tag{2}$$

其中 $\oplus$ 为异或操作。针对一篇没有做水印嵌入的生成文本，其隐秘信息比特也应服从均值分布，针对任意文本内容，在传入水印嵌入智能体前，均可以由哈希模块和信息模块统计到当前整篇的隐秘信息位数 $b_c^{(1)}$ 和 $b_c^{(0)}$，分别表示隐秘信息比特值为 1 和 0 的个数。

同时，候选词编码器$\mathrm{E_S}$负责对可进行替换位置 k 进行同义词筛选，其相似分数公式如下：

$$C_j^{(k)} = \lambda \cos\left(B\left(\boldsymbol{t}^{(k)}\right), B\left(\boldsymbol{c}^{(j)}\right)\right) + (1-\lambda)\cos\left(W\left(\boldsymbol{t}^{(k)}\right), W\left(\boldsymbol{c}^{(j)}\right)\right) \tag{3}$$

其中 $B(\cdot)$ 为 BERT 模型，可以用于计算上下文词嵌入相似度。$W(\cdot)$ 为 Word2Vec 模型[19]，用于计算全局词嵌入相似度。λ 用于两个相似度指标的权重，$\boldsymbol{t}^{(k)}$ 表示可替换位置 k 的 token，计算其与词表中任意 j 位的 $\boldsymbol{c}^{(j)}$ 的相似分数后，取得分前 5 位作为候选词，并得到其对应的编码特征和隐秘信息比特组 $\left(\boldsymbol{c}_j, \boldsymbol{b}_j\right)$，若不足则用 pad 符号编码 $\boldsymbol{c}_{pad}$ 及其所在位置对应的隐秘信息比特 $\boldsymbol{b}_{pad}$。最终便可得到传入水印嵌入智能体的文本综合环境信息 $\boldsymbol{s}_t = \left[\boldsymbol{x}_a, \boldsymbol{x}_t, (\boldsymbol{c}_1, \boldsymbol{b}_1), \cdots, (\boldsymbol{c}_5, \boldsymbol{b}_5), \left(\boldsymbol{c}_{pad}, \boldsymbol{b}_{pad}\right), \boldsymbol{b}_c\right]$。

2.2. 水印嵌入智能体

水印嵌入智能体负责对水印嵌入状态提供的综合信息 M 进行分析并进行选择行动(action)，我们设计了一个多层感知机(Multilayer Perceptron)来执行强化学习中的动作选择任务。MLP 模型通过接收这些信息，并利用多个全连接层和归一化进行处理，最终输出六个类别的预测结果，分别对应五个候选词和一个保留原词的选择。模型的结构位三层全连接层和一个输出层。对于前三层线性层由 ReLU 激活函数，LayerNorm 和 Dropout 操作组成：

$$h_z = D\left(\mathrm{LN}\left(\sigma\left(\boldsymbol{W}_z \cdot x + b_z\right)\right)\right) \tag{1}$$

其中 $\boldsymbol{W}_z \in \mathbb{R}^{d_z \times d_{z-1}}$ 为权重矩阵，d_z 为当前隐藏层维度，$b_z \in \mathbb{R}^{d_z}$ 为每一层偏置项，$\sigma(\cdot)$ 为 ReLU 激活函数，$\mathrm{LN}(\cdot)$ 为 LayerNorm，$D(\cdot)$ 为 Dropout。

模型最后一层根据 logits 输出确定最终的动作 a_t，即进行候选词选择或不替换操作，从而优化嵌入秘密信息后的文本质量和多样性。

$$a_t = \mathrm{argmax}\left(\hat{y}_z\right) \tag{5}$$

2.3. 水印嵌入奖励

我们设计了三个奖励分数用于在强化学习训练中平衡嵌入水印文本前后的语义一致性、水印可检测性和内容多样性。

第一个奖励函数为相似度奖励分数，主要用于鼓励水印嵌入智能体在改动文本时仍然保持原文核心含义的一致性，如公式所示：

$$\mathcal{R}_{\mathrm{sim}}\left(T_o, T_w\right) = \frac{e_o \cdot e_w}{\|e_o\| \|e_w\|} \tag{6}$$

式中我们使用 Sentence-Transformers [20]作为相似文本编码器 E_L，对原始文本 T_o 和嵌入水印后文本 T_w 进行分词和编码以获取隐藏向量，通过平均池化(mean pooling)聚合成全文的表示 e_o 和 e_w 并计算相似度。

第二个奖励函数为水印检测奖励分数，主要用于鼓励水印嵌入智能体进行选词时需要考虑最终的输出文本可以被检测到水印信息。针对需要检测的文本，我们同样传入嵌入水印时所使用的文本编码器 E_T 对全文 token 进行编码，并通过哈希模块和信息模块得到该文本的隐秘信息比特序列 $\{b_1,b_2,\cdots,b_n\}$，其中 $b_i\in\{0,1\}$。对于没有嵌入水印的文本，隐秘信息呈现 0 和 1 均匀分布(即 $p=0.5$)，则在大样本条件下检验基线分布可采用正态分布近似，如公式所示：

$$Z=\frac{X-np}{\sqrt{np(1-p)}} \tag{7}$$

其中 X 表示隐秘信息比特序列中比特值为 1 的数量，显著性水平 $\alpha=0.05$，若样本统计量大于等于显著性水平 α 的阈值($Z\geq Z_\alpha$)，则可以认为该文本包含水印信息，$\mathcal{R}_{wm}(s_t,a_t)$ 为 1，反之为 0。

第三个奖励函数为信息熵奖励函数，主要用于提升文本的多样性和丰富度，避免重复或者过于死板的表述方式。我们使用 BERT 模型作为文本信息熵编码器 E_H，对文中的 token 用[MASK]遮掩，并预测该位置可能的词汇及其概率，由此可以得到文本的信息熵公式为：

$$H(p)=-\sum_{r=1}^{N}p(t_r)\log p(t_r) \tag{8}$$

其中 $p(t_r)$ 为文本信息熵编码器 E_H 对文本中第 r 个 token 的预测概率，N 为文本中的 token 总数。由此我们可以得到该处嵌入水印后的 H_w 和原始文本的信息熵 H_o，通过对信息熵差做对数归一化可以得到：

$$\mathcal{R}_{\text{ent}}=\frac{\log(H_w-H_o+c)}{\log(\Delta H+c)} \tag{9}$$

其中 c 为不为 0 的常数，我们在这设置为 1，ΔH 为文本信息熵差变化的最大值。

则最终的奖励函数可以表示为：

$$\mathcal{R}_{total}(s_t,a_t)=\omega_1\mathcal{R}_{sim}(s_t,a_t)+\omega_2\mathcal{R}_{wm}(s_t,a_t)+\omega_3\mathcal{R}_{ent}(s_t,a_t) \tag{10}$$

其中 ω_i 为权重系数，用于平衡对水印认证准确率，原文语义保留和水印多样性三个目标的重要性。

我们使用 Proximal Policy Optimization (PPO2) [21]算法来进行训练，以优化基于强化学习的文本水印嵌入过程。PPO2 是一种基于策略梯度的方法，其目标函数如公式所示：

$$\mathcal{L}_{clip}(\theta)=\mathbb{E}_t\left[\min\left(r_t(\theta)A_t,\text{clip}\left(r_t(\theta),1-\varepsilon,1+\varepsilon\right)A_t\right)\right] \tag{11}$$

式中 A_t 为优势函数，由奖励函数 $\mathcal{R}_{\text{total}}(s_t,a_t)$ 计算得到。$r_t(\theta)$ 为当前策略和旧策略在同一状态下 s_t 选择某一动作 a_t 的比值，$\text{clip}(x,1-\varepsilon,1+\varepsilon)$ 为裁剪函数，ε 为裁剪参数，用于限制比值于区间 $(1-\varepsilon,1+\varepsilon)$。通过该训练过程，我们可以有效地优化模型，使其在水印嵌入任务中拥有较好的水印认证能力，较好的文本质量和语义不变以及文本多样性。

3. 实验结果及分析

3.1. 实验设置

本文中所有用来进行对比的基准模型的实验配置都保持了一致，使用 BERT [18]模型作为文本编码器 E_T 和文本信息熵编码器 E_H，使用 Sentence-Transformers [20]作为相似文本编码器 E_L。整个水印嵌入和检测方法搭建使用的是 Pytorch 框架和 Stable-baseline3 强化学习库，流程中词向量维度为 768，强化学习训练总步长为 500,000，批次步长为 2048，更新采样数为 64，裁剪系数为 0.2，学习率为 $2e^{-5}$，水印嵌入智能体中 dropout 参数为 0.3。训练过程中使用 NVIDIA TITAN RTX GPU 进行计算。本文实验环境配

置和实验参数具体设置如表 1 所示。

Table 1. Experimental environment and parameter configuration
表 1. 实验环境和参数配置

环境	配置	参数	配置
操作系统	Windows (64 位)	Dropout	0.3
CPU	i7	Batch_size	64
内存	64g	学习率	$2e^{-5}$
编程语言	Python3.9	词向量维度	768
计算框架	Pytorch	Total_step	500,000
强化学习库	Stable-baseline3	N_step	2048
显存	24g	ε	0.2

3.2. 数据集

本文使用 Human ChatGPT Comparison Corpus (HC3)数据集，包含将近 30,000 个问题及对应的人类和 ChatGPT 回答，问题涉及各种领域，包括开放领域、金融、医疗、计算机和心理学领域。我们将其按照 8:2 的比例划分训练集和测试集。训练时针对 ChatGPT 生成的回答进行文本水印嵌入与检测，测试时检测人类回答和对 ChatGPT 嵌入文本水印后的问答。

3.3. 性能指标

3.3.1. 余弦相似度

余弦相似度是量化语义空间中两个文本之间相似程度的度量。余弦相似度通过计算两个嵌入向量之间的夹角的余弦值来测量它们之间的差异。余弦值越接近 1，这两个文本语义上就越相似。本文中使用余弦相似度来计算嵌入前后文本之间的距离，余弦相似度的计算公式如下：

$$\cos\theta = \frac{\boldsymbol{X}\cdot\boldsymbol{Y}}{\|\boldsymbol{X}\|\|\boldsymbol{Y}\|} \tag{12}$$

其中，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 分别表示要对比相似度的两个文本表示。

3.3.2. BLEU 分数

BLEU 分数(Bilingual Evaluation Understudy)是一种常用的自动评估机器翻译质量的指标，用于衡量机器生成的文本与参考文本之间的相似度，也可以用于水印嵌入前后文本差异的评估。BLEU 分数通过两者在 n-gram 层面的重叠程度来评估其质量，得分越高表示生成的文本质量和语义一致性越高。其计算公式如下：

$$\mathrm{BLEU} = \beta\cdot\exp\left(\sum_{n=1}^{4}\tau_n\log p_n\right) \tag{13}$$

式中 p_n 为 n-gram 精度，ω_n 为每个 n-gram 的精度权重，一般情况下 $\tau_1=\tau_2=\tau_3=\tau_4=0.25$。$\beta$ 是惩罚因子，避免过短文本的影响。

3.3.3. 准确率与 F1 分数

准确率 ACC (Accuracy)是评估文本水印性能的常用指标，表示水印能被检测器正确检测的样本占所有样本的比例，越高则可被检测性越好，其表示公式下：

$$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{14}$$

F1 分数是综合考虑模型准确率和召回率的指标，为调和平均值介于 0 和 1 之间，越接近 1 说明检测算法的性能越好，计算公式为：

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{15}$$

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{16}$$

$$\mathrm{F1}=2\times\frac{\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{17}$$

在本文中，我们将测试集中 ChatGPT 嵌入水印后的回答作为正样本，人类回答作为负样本。

3.4. 水印性能对比

Table 2. The detection results of our method and other algorithms on the HC3 dataset
表 2. 本文方法与其他黑盒水印算法在 HC3 数据集上测试的检测性能结果

Method	Medicine		Finance		Wiki_csai		Open_QA		Reddit_eli5	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Ours	**0.9638**	**0.9644**	**0.9327**	**0.9345**	**0.9369**	**0.8819**	**0.9407**	**0.9423**	**0.9369**	0.8819
LexicalMark	0.9381	0.9292	0.9140	0.9110	0.8919	**0.8919**	**0.9407**	**0.9423**	0.9341	**0.9323**
BERTMark	0.9115	0.9027	0.9140	0.9110	0.8949	0.8919	0.8824	0.8627	0.9212	0.9198
GPTMark	0.9218	0.9184	0.9253	0.9110	0.8649	0.8649	0.8627	0.8431	0.9171	0.9143

我们使用本文训练好的模型，对测试集中的 ChatGPT 回答进行水印嵌入，然后进行水印检测和文本质量评估。我们与其他基于词汇替换的黑盒水印算法[22]对检测准确率和文本质量进行对比。同时我们也设计了使用深度学习训练的基于 BERT [19]的同义词替换水印方法和基于 GPT [23]的生成候选词替换方法，性能结果对比如表 2 与表 3 所示：

Table 3. The text quality of our method and other algorithms on the HC3 dataset
表 3. 本文方法与其他黑盒水印算法在 HC3 数据集上测试的文本质量结果

Method	Medicine		Finance		Wiki_csai		Open_QA		Reddit_eli5	
	BLEU	cosθ	BLEU	cosθ	BLEU	cosθ	BLEU	cosθ	BLEU	cosθ
Ours	**0.6580**	**0.9604**	**0.6358**	**0.9671**	**0.6932**	**0.9847**	**0.2355**	**0.8937**	**0.6883**	**0.9776**
LexicalMark	0.6541	0.9591	0.6312	0.9634	0.6850	0.9801	0.2298	0.8912	0.6835	0.9763
BERTMark	0.6023	0.9274	0.6223	0.9128	0.6882	0.9786	0.2273	0.8879	0.6710	0.9549
GPTMark	0.6046	0.9387	0.6288	0.9520	0.6713	0.9673	0.2153	0.8754	0.6539	0.9442

由实验结果表 2 可得，我们的方法在医疗、经济、百科等领域都有较好的可检测性，同时由表 3 可得嵌入水印后的文本在文本质量上有较高的语义一致性和相似性。

我们也对嵌入水印后的文本进行了水印攻击测试如图 2 所示，证明我们的方法在面对替换攻击、删词攻击和逆序攻击时都有较好的抵抗能力。

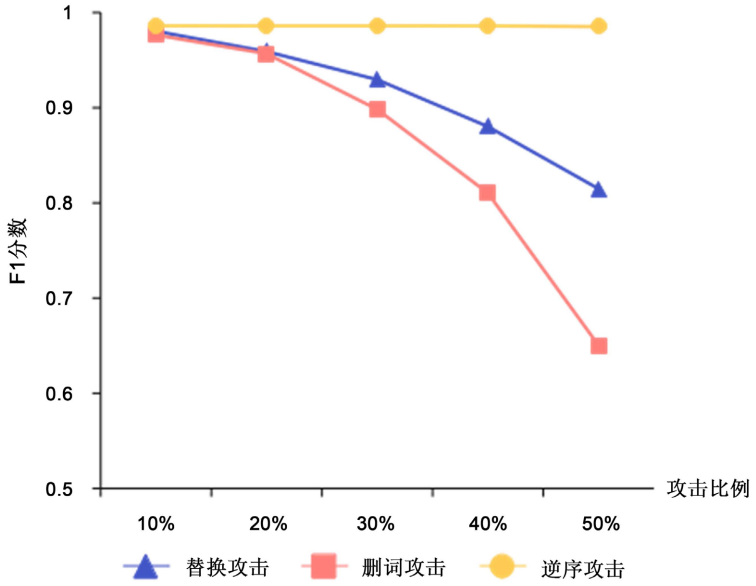

Figure 2. Resistance of our method to text watermarking attacks
图 2. 本文方法对文本水印攻击的抵抗能力

4. 总结

本文提出一种面向大语言模型的黑盒文本水印算法。为文本水印嵌入后仍能拥有较好的文本质量和水印性能，设计了水印嵌入状态、水印嵌入智能体和水印嵌入奖励三个部分进行文本水印嵌入的强化学习训练。通过设计的三个奖励函数，我们的方法在水印的可检测性、文本语义一致性以及抗常规文本水印攻击如删词攻击、替换攻击和逆序攻击均表现出色。大量的实验表明，本文算法的整体性能优于一些最先进的算法，无论目标的大语言模型是否开源，均可以对其生成的文本进行具有良好的性能文本水印嵌入和水印检测。

基金项目

上海市教委人工智能促进科研范式改革赋能学科跃升计划项目。

参考文献

[1] Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q., *et al.* (2023) A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. *IEEE/CAA Journal of Automatica Sinica*, **10**, 1122-1136. https://doi.org/10.1109/jas.2023.123618

[2] Biswas, S.S. (2023) Role of Chat GPT in Public Health. *Annals of Biomedical Engineering*, **51**, 868-869. https://doi.org/10.1007/s10439-023-03172-7

[3] Allam, H., Dempere, J., Akre, V., Parakash, D., Mazher, N. and Ahamed, J. (2023) Artificial Intelligence in Education: An Argument of Chat-GPT Use in Education. 2023 9*th International Conference on Information Technology Trends* (*ITT*), Dubai, 24-25 May 2023, 151-156. https://doi.org/10.1109/itt59889.2023.10184267

[4] Ilias, L., Michail Kazelidis, I. and Askounis, D. (2024) Multimodal Detection of Bots on X (Twitter) Using Transformers. *IEEE Transactions on Information Forensics and Security*, **19**, 7320-7334. https://doi.org/10.1109/tifs.2024.3435138

[5] Gradon, K.T. (2023) Electric Sheep on the Pastures of Disinformation and Targeted Phishing Campaigns: The Security Implications of ChatGPT. *IEEE Security & Privacy*, **21**, 58-61. https://doi.org/10.1109/msec.2023.3255039

[6] Khalil, M. and Er, E. (2023) Will ChatGPT Get You Caught? Rethinking of Plagiarism Detection. In: Zaphiris, P., Ioannou, A., Eds., *Learning and Collaboration Technologies. Lecture Notes in Computer Science*, Springer, 475-487. https://doi.org/10.1007/978-3-031-34411-4_32

[7] Macdonald, C., Adeloye, D., Sheikh, A. and Rudan, I. (2023) Can ChatGPT Draft a Research Article? An Example of Population-Level Vaccine Effectiveness Analysis. *Journal of Global Health*, **13**, Article 01003. https://doi.org/10.7189/jogh.13.01003

[8] Grbic, D.V. and Dujlovic, I. (2023) Social Engineering with ChatGPT. 2023 22*nd International Symposium INFOTEH-JAHORINA* (*INFOTEH*), East Sarajevo, 15-17 March 2023, 1-5. https://doi.org/10.1109/infoteh57020.2023.10094141

[9] Guo, B., Zhang, X., Wang, Z., *et al.* (2023) How Close Is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. arXiv:2301.07597

[10] Chen, Y., Kang, H., Zhai, V., *et al.* (2023) GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content. arXiv:2305.07969v2

[11] Mitchell, E., Lee, Y., Khazatsky, A., *et al.* (2023) Detectgpt: Zero-Shot Machine-Generated Text Detection Using Probability Curvature. *International Conference on Machine Learning* (*ICML*). Honolulu, 23-29 July 2023, 24950-24962.

[12] Yang, X., Cheng, W., Wu, Y., *et al.* (2023) DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text. arXiv:2305.17359

[13] Yu, P., Chen, J., Feng, X. and Xia, Z. (2025) CHEAT: A Large-Scale Dataset for Detecting ChatGPT-Written Abstracts. In: *IEEE Transactions on Big Data*, IEEE, 1-9. https://doi.org/10.1109/tbdata.2025.3536929

[14] Liang, W., Yuksekgonul, M., Mao, Y., Wu, E. and Zou, J. (2023) GPT Detectors Are Biased against Non-Native English Writers. *Patterns*, **4**, Article 100779. https://doi.org/10.1016/j.patter.2023.100779

[15] Abdelnabi, S. and Fritz, M. (2021) Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding. 2021 *IEEE Symposium on Security and Privacy* (*SP*), San Francisco, 24-27 May 2021, 121-140. https://doi.org/10.1109/sp40001.2021.00083

[16] Kirchenbauer, J., Geiping, J., Wen, Y., *et al.* (2023) A Watermark for Large Language Models. *International Conference on Machine Learning* (*ICML*), Honolulu, Hawaii, USA, 23-29 July 2023, 17061-17084.

[17] Dathathri, S., See, A., Ghaisas, S., Huang, P., McAdam, R., Welbl, J., *et al.* (2024) Scalable Watermarking for Identifying Large Language Model Outputs. *Nature*, **634**, 818-823. https://doi.org/10.1038/s41586-024-08025-4

[18] Jacob, D., Ming-Wei, C., Kenton, L, *et al.* (2019) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT* 2019, Minneapolis, 2-7 June 2019, 4171-4186.

[19] Mikolov, T., Chen, K., Corrado, G., *et al.* (2013) Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781.

[20] Reimers, N. and Gurevych, I. (2019) Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong SAR, 3-7 November 2019, 3982-3992. https://doi.org/10.18653/v1/d19-1410

[21] Schulman J, Wolski F, Dhariwal P, *et al.* (2017) Proximal Policy Optimization Algorithms. arXiv:1707.06347.

[22] He, X., Xu, Q., Lyu, L., Wu, F. and Wang, C. (2022) Protecting Intellectual Property of Language Generation Apis with Lexical Watermark. *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, 10758-10766. https://doi.org/10.1609/aaai.v36i10.21321

[23] Radford, A., Wu, J., Child, R., *et al.* (2019) Language Models Are Unsupervised Multitask Learners. *OpenAI Blog*, **1**, 9.