上海大学工学硕士学位论文

基于概率调制的大模型生成文本水印方法研究

作　　者: 鲍思源

导　　师: 张新鹏

学科专业: 电子信息

通信与信息工程学院

上海大学

2026 年 5 月

A Dissertation Submitted to Shanghai University for the
Degree of Master in Engineering

Research on Watermarking Methods for Large Language Model-Generated Text Based on Probabilistic Modulation

Candidate: Siyuan Bao
Supervisor: Xinpeng Zhang
Major: Electronic Information

Schoool of Communication and Information Engineering
Shanghai University
May, 2026

摘　要

生成式大语言模型在智能问答与内容创作等文本生成任务中得到广泛应用。然而，在显著提升内容生产效率的同时，生成内容来源难以追溯、机器生成文本与人工文本难以区分等问题日益凸显，进而可能引发虚假信息传播、学术诚信风险以及版权归属不明确等安全隐患。大语言模型文本水印技术通过在生成过程中嵌入隐式标识信息，使生成内容具备可检测性，为人工智能生成内容的来源识别与治理提供了一种可行路径。然而，自然语言具有较强的表达灵活性，文本在语义基本保持不变的情况下可以通过多种方式进行改写，从而对水印信号造成破坏。因此，如何在文本扰动条件下兼顾生成质量与水印鲁棒性，成为当前研究的关键问题。针对上述问题，本文基于概率调制机制开展大语言模型文本水印方法研究，重点探索水印嵌入强度与检测性能之间的平衡关系。主要工作如下：

1）针对生成过程中概率分布可控调制问题，本文设计了一种多偏置随机选择机制的文本水印方法。在生成阶段，利用密钥驱动的伪随机函数并结合上下文哈希信息，构造多个候选偏置参数，并在每一时间步从中随机选取用于当前概率分布的调制。与固定偏置策略相比，该方法不再依赖单一调制强度，而是通过多偏置的随机调制，使水印信号在生成序列中呈现离散化嵌入特征，从而降低对特定位置及特定词的依赖程度，并增强对局部文本扰动的抵抗能力。实验结果表明，在保持原始生成分布统计特征基本稳定的前提下，该方法能够实现水印嵌入强度的精细化调控，并在文本生产质量与水印检测性能之间取得较优平衡。

2）为提升水印方法对自然语言结构特征的刻画能力，提出一种融合结构信息与生成不确定性的自适应文本水印方法。该方法引入句法结构特征与概率分布不确定性度量，对生成位置进行差异化建模，并据此动态调节偏置参数：在语法约束较强或语义承载较高的位置降低嵌入强度，在生成不确定性较高或语义冗余度较大的位置提高嵌入强度。同时，通过施加平滑约束限制相邻时间步的分布波动，保证生成过程的稳定性。实验结果表明，该方法在多种文本扰动条件下具有鲁棒性，并在保持文本自然性的同时提升了水印检测的可靠性。

关键词：大语言模型；文本水印；生成式人工智能；概率调制

ABSTRACT

Large language models (LLMs) have been widely applied in text generation tasks such as question answering and content creation. While significantly improving content production efficiency, they also introduce critical challenges, including the difficulty of tracing content provenance and distinguishing machine-generated text from human-written text. These issues may further lead to the spread of misinformation, risks to academic integrity, and ambiguous copyright ownership.Text watermarking for LLMs embeds implicit identification signals into the generation process, enabling the detectability of generated content and providing a feasible solution for provenance tracking and governance of AI-generated text. However, due to the high flexibility of natural language, texts can be paraphrased in various ways while preserving their semantics, which can disrupt watermark signals. Therefore, achieving a balance between generation quality and watermark robustness under text perturbations has become a key challenge.To address this issue, this dissertation investigates text watermarking methods for LLMs based on probabilistic modulation, with a focus on the trade-off between watermark embedding strength and detection performance. The main contributions are as follows:

1）To achieve controllable modulation of probability distributions during generation, a text watermarking method based on a multi-bias random selection mechanism is proposed. Specifically, a key-driven pseudo-random function combined with contextual hashing is employed to construct multiple candidate bias parameters, from which one is randomly selected at each time step to modulate the token probability distribution. Compared with fixed-bias strategies, the proposed method distributes watermark signals across the generation sequence through randomized bias selection, reducing reliance on specific positions or tokens and improving robustness against local text perturbations. Experimental results demonstrate that the method achieves fine-grained control over watermark embedding strength while maintaining the statistical properties of the original distribution, thereby balancing text quality and detection performance.

2）To enhance the modeling of structural characteristics of natural language, an adaptive text watermarking method that incorporates structural information and generation uncertainty is further proposed. This method introduces syntactic structural features and uncertainty measures of the probability distribution to perform more fine-grained position-aware modulation, and dynamically adjusts bias parameters accordingly: the embedding strength is reduced at positions with strong syntactic constraints or high semantic importance, while it is correspondingly increased in regions with higher generation uncertainty or greater semantic redundancy. In addition, smoothing constraints are imposed to limit distribution fluctuations between adjacent time steps, thereby ensuring the overall stability of the generation process. Experimental results show that the proposed method exhibits improved robustness under various types of text perturbations and further enhances detection reliability while preserving text naturalness.

Keywords: Large Language Models; Text Watermarking; Generative AI; Probabilistic Modulation

目　录

摘　要 I

ABSTRACT II

第一章　绪论 1

1.1　研究背景与意义 1

1.2　国内外研究概况 2

1.2.1　大语言模型文本被动检测研究 2

1.2.2　大语言模型文本水印技术研究 4

1.2.3　现有研究的不足 6

1.3　论文研究内容 7

1.4　论文结构安排 7

1.5　本章小结 8

第二章　文本水印技术理论基础 9

2.1　大语言模型生成机制 9

2.1.1　自回归语言模型原理 9

2.1.2　Transformer 的注意力机制 10

2.1.3　采样策略与概率控制机制 12

2.2　文本水印的基本原理与形式化建模 14

2.2.1　文本水印的定义与分类 14

2.2.2　基于概率偏置的水印嵌入机制 15

2.2.3　文本水印检测统计原理 17

2.2.4　文本水印系统的形式化建模 18

2.3　文本水印性能评价指标体系 19

2.3.1　检测性能指标 19

2.3.2　文本质量指标 20

2.3.3　鲁棒性指标 22

2.4　本章小结 23

第三章 基于多偏置调制的文本水印 24
3.1 引言 24
3.2 总体框架 24
3.3 水印嵌入 26
3.3.1 上下文哈希机制 26
3.3.2 多候选偏置生成机制 28
3.3.3 词表构建与概率调制 29
3.3.4 嵌入算法流程 30
3.4 水印提取 32
3.4.1 统计检验 32
3.4.2 提取算法流程 33
3.5 实验结果分析 34
3.5.1 数据集与实验设置 34
3.5.2 实验结果与分析 35
3.6 本章小结 42
第四章 基于结构信息的自适应模型文本水印 43
4.1 引言 43
4.2 总体框架 43
4.3 水印嵌入 45
4.3.1 结构感知与上下文建模机制 45
4.3.2 不确定性驱动的自适应偏置机制 47
4.3.3 词表构建与自适应概率调制 48
4.3.4 嵌入算法流程 49
4.4 水印提取 51
4.4.1 统计检验与提取实现 51
4.4.2 统计行为分析 52
4.5 实验结果分析 53
4.5.1 数据集与实验设置 53
4.5.2 实验结果与分析 53
4.6 本章小结 56

第五章 总结与展望 57

5.1 工作总结 57

5.2 工作展望 58

攻读硕士学位期间取得的研究成果 69

致 谢 70

第一章　绪论

1.1　研究背景与意义

人工智能（Artificial Intelligence, AI）技术的发展[1-3]经历了由规则驱动向数据驱动再到大规模预训练模型主导的演进过程，生成式人工智能[4]逐渐成为主流。自然语言处理（Natural Language Processing, NLP）[5-7]在此过程中取得显著进展，早期 NLP 系统主要依赖人工规则与知识库进行显式建模，统计学习方法[8]的引入推动了基于概率分布的语言模型发展。基于 n-gram 的模型[9]虽然在局部一致性上有所改善，但难以刻画长距离依赖，语义建模能力仍然存在明显局限。循环神经网络（Recurrent Neural Network, RNN）及其改进结构[10]增强了上下文记忆能力，却受限于序列计算方式，导致在长文本建模与训练效率方面存在瓶颈。

为解决上述问题，Transformer 架构被提出[11]，成为现代语言模型的重要基础结构。其核心自注意力机制（Self-Attention, SA）[12]能够在全局范围内建模任意位置之间的依赖关系，有效捕捉长距离语义关联，并支持高效并行训练。在 Transformer 架构的推动下，大语言模型（Large Language Models, LLMs）[13-15]快速发展。模型参数规模与训练语料规模也在不断扩大，渐渐形成了基于大规模语料预训练与下游任务微调的训练范式，模型能够通过自监督学习获得通用语言表示能力。引入指令微调（Instruction Tuning, IT）[16]与基于人类反馈的强化学习（Reinforcement Learning from Human Feedback, RLHF）[17]后，大语言模型提升了任务对齐能力与输出可控性，进而提高了文本生成质量与人机交互能力。

生成式人工智能在教育[18]、科研[19]、传媒[20]及软件开发[21]等领域的快速扩展与深度渗透，带来了前所未有的安全风险和伦理挑战。由于生成文本在语言风格与表达方式上已与人类写作高度接近，基于统计特征差异的传统检测方法逐渐难以有效区分其来源。同时，大语言模型生成内容可能被用于虚假信息传播、学术不端以及自动化内容操控等场景，使得文本来源识别与责任追溯问题愈发复杂。

现有研究通常将文本来源识别方法划分为被动检测与主动嵌入两类[22]。被动检测依赖统计差异进行判别，生成模型能力提升使其判别边界不断模糊。主动嵌入是在生成过程中或生成完成后嵌入可验证标识以实现后验检测，其中的文本水印技

术[23-25]伴随模型发展成为保障生成内容可追溯性的关键手段。文本水印的核心思想是在不显著影响生成质量的前提下，通过对概率分布进行结构化调制，实现隐式标识嵌入。现有方法多采用固定偏置策略，使水印信号在序列中呈现集中分布特征，在语义改写、再生成或局部扰动攻击下容易被削弱，导致检测性能显著下降。

由此产生一个关键问题，在不破坏语言模型生成能力的前提下，如何使水印信号在生成序列中保持足够分散性与稳定性，并在复杂文本扰动场景下仍然具有可检测性与鲁棒性。这一问题直接决定了文本水印技术能否从实验可行走向实际可用。基于上述背景，本文围绕大语言模型生成文本水印展开研究，重点探讨基于概率偏置的嵌入机制及其结构优化方法。在方法层面，通过改进偏置调制策略与词元选择机制，在保证语义一致性的同时增强水印信号的分散性与稳定性。在应用层面，为生成内容的溯源与可信性提供技术支持，提升人工智能生成内容的可控性与安全性。

1.2 国内外研究概况

针对生成文本来源难以识别及水印鲁棒性不足等问题，国内外学者已开展了大量相关研究工作[26-28]。随着生成式大语言模型能力的不断提升，研究重点逐步从早期的文本特征分析转向更加系统化的来源识别与可追溯机制设计[29-34]。现有研究主要围绕两条技术路径展开，文本被动检测致力于通过分析文本统计特征或概率分布差异进行事后判别，文本水印则通过在文本生成过程中引入可验证信号实现来源标识与追溯。两类方法在实现机制与适用场景上存在明显差异，并各自面临不同的技术挑战。为系统梳理相关研究进展，接下来将分别从被动检测方法与主动水印方法两个方面展开介绍，并进一步分析其优缺点与发展趋势。

1.2.1 大语言模型文本被动检测研究

被动检测方法不对文本生成过程进行干预，而是从结果出发，对已生成文本进行判别分析。相关研究主要利用文本在语言表达、统计特性以及概率分布等方面的差异，构建分类模型以区分人类文本与机器生成文本。该类方法具有无需修改生成模型、部署灵活等优点，在实际应用中具有一定可行性。根据所依赖特征的不同，这类方法可以进一步划分为基于语言学特征的检测方法和基于模型概率信息的检测方法。随着技术的进步，大语言模型能力的不断提升使得基于表层差异的判别难度不断增加，同时文本改写等后处理操作也可能进一步削弱检测效果，该类方法的稳定

性与泛化能力在一定程度上得到限制而迎来巨大挑战。

早期研究主要从语言学特征角度出发，对机器生成文本与人工文本之间的差异进行分析。Fröhling 等人[35]发现机器生成文本的词汇丰富度低于人类文本，在句法结构上也相对单一，因此提出基于多种语言特征组合的检测方法。Gallé 等人[36]提出一种无监督分布检测方法，通过分析文本在统计分布层面的差异进行文本识别。Kim 等人[37]则从语篇层面展开研究，利用句子之间的结构关系差异进行检测。随着大语言模型经过对齐优化（Alignment Optimization, AO）[38]后，单纯依赖表层语言特征的检测方法逐渐难以适应复杂的人工智能生成场景。针对这一问题，Alhijawi 等人[39]提出用于检测 ChatGPT 生成科学文本的多模态检测模型 AI-Catcher。

除了基于语言学特征的方法之外，另一类研究利用语言模型的概率评分机制进行检测。Gehrmann 等人[40]提出的 GLTR（Giant Language Model Test Room）框架指出机器生成文本中高概率词汇的出现频率明显高于人类文本。Mitchell 等人[41]进一步提出 DetectGPT 方法，研究发现机器生成文本往往位于语言模型概率函数的负曲率区域，因此可以对文本进行随机扰动并分析概率变化趋势来判断文本来源。Hans 等人[42]通过比较两个不同语言模型对同一文本的概率分布差异进行检测以提升检测结果的稳定性。Popescu-Apreutesei 等人[43]通过对 BiLSTM 模型进行微调，在人工智能生成摘要数据集上取得接近 97% 的检测准确率。Oghaz 等人[44]则通过对 Transformer 模型进行微调，提高了传统文本分类器在生成文本检测任务中的性能。

近年来，国内学术界也开始逐渐关注大语言模型生成文本检测问题，并在语言特征分析、概率统计检测以及深度学习分类模型等方面开展了大量研究。Guo 等人[45]通过实验分析发现语言模型生成文本在写作风格上与人类文本存在明显差异，基于这一观察，该研究提出了一种结合句法结构与语篇特征的检测方法。Wang 等人[46]进一步研究发现，语言模型生成的摘要具有句子长度较长、结构相对单一等特点，而人类撰写的摘要在表达上具有更强的个体差异性。因此，该研究构建了一种结合多种语言统计特征的分类器组合系统，用于识别机器生成摘要。Liu 等人[47]通过构建句子之间语义关系的实体连贯性图，从语义结构层面对文本组织特征进行分析。

国外基于模型概率统计指标的检测研究对国内学者产生重要影响。针对 DetectGPT[41]计算成本较高的问题，Miao 等人[48]提出使用贝叶斯替代模型进行近似计算以降低检测开销。Bao 等人[49]提出基于条件概率曲率的 Fast-DetectGPT 方法，在保持检测性能的同时显著提高计算效率。Liu 等[50]人引入对比学习机制提出 PECOLA 框

架，增强检测模型在不同数据域上的泛化能力。Su 等人[51]提出一种无需对文本进行扰动的检测方法，仅利用对数似然和对数排序信息即可完成检测，提升了检测效率。Yang 等人[52]利用大语言模型对文本进行续写，并通过分析原始文本与续写文本之间的差异来判断文本来源。Guo 等人[53]则提出利用语言模型的去噪能力作为判别依据，从新的角度对生成文本检测问题进行研究。

针对短文本检测难度较大的问题，Wei 等人[54]提出利用无关内容插入技术（off-topic content insertion, OCI）稳定持续同调维数（persistent homology dimension, PHD）的方法，提高短文本检测稳定性。Zhu 等人[55]提出多尺度一致性检测方法（Multi-scale Consistency Prediction, MCP），通过在不同粒度尺度上分析文本一致性特征来提高检测结果的可靠性。另外，在监督学习方向，Zeng 等人[56]提出混合句子 RoBERTa 分类器，Liu 等人[57]提出 ArguGPT 模型用于识别人工智能生成的论证文本。Chen 等人[58]则通过逆向提示方法提升检测模型的可解释性。

综上所述，被动检测方法在文本来源识别方面已取得一定进展，尤其是在长文本和特征差异较为明显的场景中表现较好。然而，这类方法本质上依赖于生成文本与人类文本之间的统计差异，当生成模型能力不断提升时，机器生成文本越来越接近人类书写文本，这种差异逐渐减弱，检测性能受到限制。此外，在短文本或特征不充分的情况下，检测结果往往不够稳定。研究者意识到上述局限性，开始探索在文本生成阶段引入可验证标识的主动方法，从源头提升文本的可识别性。

1.2.2 大语言模型文本水印技术研究

与被动检测方法不同，文本水印技术在生成阶段引入可控扰动，使输出文本携带可验证的统计特征，实现生成内容的来源追踪，从根本上提升文本的可识别性与可追溯性。在具体实现方面，现有研究多从词级概率分布出发设计水印机制。Kirchenbauer 等人[59]提出的基于词表划分与概率偏置的文本水印框架成为许多研究的基础技术支撑。该方法利用伪随机函数将词汇表划分为绿色集合与红色集合，并在生成过程中对绿色词元施加对数概率偏置参数 γ，使其被优先采样，达到在序列中形成统计偏移的目的。检测阶段通过构建 z-score 统计量，对文本中绿色词元的出现频率进行显著性检验，以此判断水印是否存在。该方法无需对模型参数进行修改且可直接应用于现有生成模型，具有较好的实用性与可扩展性。

然而，现有基于词级分布的水印方法在鲁棒性方面仍面临挑战。Krishna 等

人[60]提出改写模型 DIPPER，用于评估水印在语义保持条件下的抗攻击能力。实验结果表明，在不同强度的改写攻击下，水印文本的检测统计量显著下降，尤其在高强度改写场景中，其 z-score 分布逐渐接近无水印文本，表明水印信号易被削弱，难以维持稳定检测性能。部分研究开始探索将水印嵌入从词级概率分布扩展至更高层的表示空间，例如在 logits 空间或语义嵌入空间中引入结构化扰动，使水印特征在隐藏表示中保持稳定，从而提升其对语义改写等攻击的抵抗能力。与此同时，Christ 等人[61]提出基于伪随机数比较的水印机制，在一定程度上避免了显式偏置带来的分布扰动问题。

近年来，国内学者在文本水印技术方面也开展了大量研究，并在综述分析、中文语境适配以及嵌入机制优化等方面取得了一定进展，推动了相关方法在实际应用中的落地。在综述研究方面，复旦大学郭钊均等人[62]系统梳理了 AIGC 场景下数字水印技术的发展现状，分析了其在模型保护、内容溯源及数据安全等方面的应用价值，并指出水印设计过程中需要在嵌入强度与文本自然度之间进行权衡。清华大学 Liu 等人[63]进一步对主流文本水印方法进行了系统评估，结合中文语料特点开展实验分析，指出分词粒度对水印统计显著性具有重要影响，同时也影响水印在不同语言环境下的适用性。

在水印嵌入机制优化方面，相关研究主要围绕提升鲁棒性与降低文本质量损失展开。Liu 等人[64]提出语义不变鲁棒水印方法，通过约束语义一致性来增强水印在改写场景下的稳定性。Hou 等人[65]提出句子级语义水印方法 SemStamp，将生成文本分布限制在特定语义空间，实现更高层级的水印嵌入，降低了局部词元扰动带来的影响。Chen 等人[66]提出多通道水印方法 MCMARK，缓解了传统概率偏置带来的分布偏移问题。Wang 等人[67]提出共生水印方法 SymMark，融合了多种嵌入策略，以此达到提高水印机制的灵活性与适应性的目的。总体来看，该方向研究呈现出从词级分布向语义层表示拓展的趋势。

此外，部分研究关注生成后嵌入水印的后处理方法。这类方法通常利用词级替换或句法调整实现水印嵌入，其中同义词替换是一种典型思路。Yang 等人[68]使用 BERT 生成候选词并筛选语义一致替换位置，后续研究引入随机编码机制[69]实现水印检测。Hao 等人[70]在此基础上提出 RSFAW 方法，通过优化嵌入位置选择策略，提高水印在文本编辑与改写场景下的稳定性。这类方法无需依赖生成模型内部结构，具有较好的模型兼容性与灵活性，但在隐蔽性与稳定性方面仍存在一定局限，容易受到大幅

度语义改写的影响。

综上，现有文本水印方法实现简单、无需修改模型结构等优势使其在实际系统中具有较好的应用基础。围绕这一技术路线，相关研究不断从嵌入机制、检测方法以及跨语言适配等方面进行改进，使文本水印在可检测性与实用性方面取得了一定进展。从整体发展趋势来看，文本水印技术正逐步由早期基于词级统计特征的方法，向融合语义信息与结构特征的方向演进。同时，随着生成模型能力的持续提升，水印方法在设计上也愈发强调对生成质量的影响控制以及对复杂应用场景的适应能力。这些研究为构建更加稳定可靠的文本来源标识机制提供了重要基础。在此基础上，进一步分析现有方法在鲁棒性、隐蔽性等方面的局限性，对于明确后续研究方向具有重要意义。下一节将对相关研究中存在的不足进行系统梳理。

1.2.3 现有研究的不足

现有研究在大语言模型生成文本被动检测与文本水印技术方面已取得较为丰富的成果，在理论建模与方法设计上均形成了一定基础。然而，随着生成模型能力的不断提升以及应用场景的日益复杂，相关方法在检测鲁棒性、水印隐蔽性以及跨场景适应能力等方面逐渐暴露出一定局限性。特别是在高质量生成文本与多样化文本扰动条件下，现有方法的稳定性与可靠性仍有待进一步提升，因此亟需对其不足进行系统分析与改进。

在被动检测方法方面，基于语言特征的检测策略依赖于人类文本与生成文本之间的分布差异，这类方法通常通过提取词汇、句法或统计特征进行判别。然而，随着生成模型经过大规模预训练及对齐优化，其输出文本在语言表达和统计特性上逐渐逼近真实语料，使得表层特征差异不断缩小，导致了检测模型的判别能力削弱的问题。考虑到文本在使用场景中极其灵活，这类方法对文本改写、润色等后处理操作较为敏感，在实际应用中容易出现检测结果不稳定的问题。如若投入使用，基于模型概率分布的检测方法虽然能够利用更深层的生成特征，但通常需要访问模型内部信息或进行复杂计算，在实际部署中面临计算开销较大与可用性受限等问题。

在文本水印技术方面，现有基于概率偏置的水印方法虽然具有实现简单、无需修改模型结构等优势，在工程应用中具有较好的可行性，但其鲁棒性仍有待提升。在语义保持的改写攻击或跨模型再生成场景下，由于文本表达形式发生变化，原有基于词级统计特征的水印信号容易被削弱甚至破坏，从而影响检测效果。最重要的一

点是，水印嵌入强度与文本质量之间普遍存在权衡关系：当嵌入强度较大时，虽然有助于提高检测显著性，但可能对文本自然性与流畅性产生负面影响；而当嵌入强度较小时，则可能导致水印信号不明显，降低检测的可靠性。据此可知，如何在保证文本质量的前提下实现稳定且可检测的水印嵌入，仍是当前研究的关键难点。

1.3 论文研究内容

本文围绕大语言模型生成文本的水印嵌入问题展开研究。针对现有文本水印方法在鲁棒性与生成质量之间难以兼顾的问题，重点研究基于概率调制的文本水印嵌入机制，并对其检测性能进行系统分析。本文以大语言模型生成文本水印为研究对象，从概率分布调制机制设计与鲁棒性增强两个方面开展研究。首先，对大语言模型的文本生成机制及常用采样策略进行分析，系统梳理现有文本水印方法的基本原理与具体实现流程。在此基础上，从概率分布调制角度出发，分析水印嵌入与检测过程中的关键影响因素，为后续方法设计提供理论依据。

针对传统固定偏置水印方法在嵌入分布单一及鲁棒性不足方面的局限，提出一种基于多偏置随机选择机制的文本水印方法，通过构造多组候选偏置参数，实现对概率分布的随机化调制，使水印信号在生成序列中呈分散嵌入特性，在不同生成步动态切换偏置策略也避免了对固定位置及特定词元选择模式的长期依赖，提升水印的隐蔽性与鲁棒性。同时，针对文本改写等扰动对水印信号的破坏问题，进一步提出融合结构信息与生成不确定性的自适应水印方法，对不同生成位置进行差异化建模来动态调节水印嵌入强度，在保证语义与语法合理性的前提下提升水印鲁棒性。最后，构建实验环境，对所提出方法在检测准确性、文本质量及对抗扰动鲁棒性等方面进行系统评估，并与典型水印方法进行对比分析。实验结果表明，所提出方法在保持文本自然性的同时，有效提升了水印检测的稳定性与鲁棒性，为生成式文本内容的可溯源与可信治理提供了新的技术思路。

1.4 论文结构安排

本文共分为五章，各章内容安排如下。

第一章介绍研究背景与研究意义，分析生成式人工智能发展背景下文本内容溯源与版权标识面临的挑战，对国内外相关研究现状进行介绍。

第二章介绍相关技术基础，包括大语言模型的文本生成机制、常见采样策略以及文本水印技术的基本原理，分析现有方法在概率调制与检测机制方面的特点与不足，为后续研究提供理论支撑。

第三章提出一种基于多偏置随机选择机制的文本水印方法。该方法通过构建多组偏置参数并引入随机选择机制，实现对生成概率分布的动态调制，从而增强水印信号的分散性与鲁棒性。通过实验验证该方法在检测性能与文本质量之间取得了较好的平衡。

第四章在第三章方法基础上，进一步提出融合结构信息与生成不确定性的自适应水印方法。对生成位置实施差异化调制策略，引入平滑约束，通过实验对其性能进行分析。

第五章对全文研究工作进行了总结，对未来可能的研究方向进行了展望。

1.5 本章小结

本章围绕生成式大语言模型背景下的文本来源识别问题，分析了人工智能生成内容带来的挑战，并对相关研究现状进行了综述。在此基础上，总结了现有方法在鲁棒性与稳定性方面的不足，明确了本文的研究问题与主要工作。最后，对论文结构进行了整体安排，为后续研究奠定基础。

第二章　文本水印技术理论基础

2.1　大语言模型生成机制

大语言模型在大规模语料数据库上进行预训练，学习自然语言序列中的统计规律，能够完成多种自然语言处理任务，模型规模、训练数据量及表达能力均显著提升。当前主流生成式模型通常采用自回归生成框架，以 Transformer 结构作为核心模型架构，比如 GPT 系列[71-73]、LLaMA 系列[74]等。在推理阶段，模型根据已有上下文逐步预测后续词元的概率分布，而后结合采样策略生成完整文本序列。深入理解大语言模型的生成机制，包括自回归语言建模方式、Transformer 架构中的注意力机制以及推理阶段的采样策略，对于后续水印算法的设计具有重要意义。本节从语言建模原理、模型结构以及生成策略三个方面对大语言模型生成机制进行介绍。

2.1.1　自回归语言模型原理

当前主流生成式语言模型普遍采用自回归语言模型（Autoregressive Language Models, AR-LM）框架。相较于 BERT[75]等非自回归模型，自回归语言模型的文本生成质量更高、可控性更强，相较于 LLadA 系列的扩散语言模型[76]，自回归语言模型训练更简单、收敛更快。自回归语言模型框架通过语言建模使计算机具备理解自然语言的能力，其过程类似于人类习得语言的方式，即学习词元之间的关联关系、语法结构以及上下文依赖关系，并在此基础上逐步预测下一个词元的概率。

在自回归语言模型框架下，给定一个长度为 n 的文本序列

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{2.1}$$

语言模型根据概率链式法则，将整个序列的联合概率表示为条件概率的乘积形式：

$$P(\mathbf{x}) = \prod_{t=1}^{n} P(x_t \mid x_{<t}) \tag{2.2}$$

每一个词元的生成仅依赖于其之前的上下文信息，模型学习条件概率分布 $P(x_t \mid x_{<t})$ 来逐词构建完整的文本序列。在模型训练阶段，通过最大化训练语料的对数似然函

数，使模型能够更好地拟合真实数据分布。设训练语料集合为 $\mathcal{D}$，则模型的优化目标可表示为：

$$\mathcal{L} = \sum_{\mathbf{x} \in \mathcal{D}} \log P(\mathbf{x}; \theta) \tag{2.3}$$

经由梯度下降等优化方法保证了模型参数 θ 不断更新，预测分布逐渐逼近真实语料中的语言分布。在生成阶段，模型在给定当前上下文 $x_{1:t-1}$ 的条件下，输出词元表 $\mathcal{V}$ 上的概率分布：

$$P_t = \text{softmax}(\mathbf{z}_t) \tag{2.4}$$

其中，$\mathbf{z}_t$ 表示模型输出的 logits 向量，通常由 Transformer 最后一层隐藏状态经过线性映射得到。softmax 函数将 logits 转换为归一化概率分布，使所有词元的概率之和为 1。随后，模型根据该概率分布从词表中采样，选择一个词元作为当前输出 x_t，将其加入上下文后继续预测接下来的词元，按此步骤生成至文本被完整输出。

自回归生成机制具有显著的因果性（Causality）特征。在任意生成时刻 t，模型仅能利用此前已生成的词元信息，无法访问未来词元。这一特性不仅保证了生成文本的时间顺序一致性，还使得每一步词元预测过程成为一个独立概率决策过程。对于文本水印技术而言，自回归语言模型所形成的逐词元预测结构为水印嵌入提供了天然接口。具体来说，水印算法可以在每一步的概率分布 P_t 或 logits 向量 $\mathbf{z}_t$ 上施加可控的微小偏置，引导模型在候选词元集合中更倾向于选择偏置后的词元以满足特定统计规律，进而在生成文本中嵌入可检测的水印信号。

2.1.2 Transformer 的注意力机制

当前主流生成式语言模型大多基于 Transformer 架构构建。Transformer 由 Vaswani 等人[11]在 2017 年提出，其核心思想是通过自注意力机制直接建模序列中任意位置之间的依赖关系，做到了有效缓解传统循环神经网络在长距离依赖建模过程中存在的梯度消失问题。大语言模型通常采用 Transformer 的 Decoder-only 结构，如图 2.1 所示，由嵌入层（Embedding Layer）、多头自注意力层（Multi-Head Self-Attention Layer）、前馈神经网络（Feed-Forward Network, FFN）、残差连接（Residual Connection）以及层归一化（Layer Normalization）等模块组成。该结构具有较强的泛化能力和语序建模能力，尤其适用于对话生成等任务，且在并行计算效率方面具有显著优势。

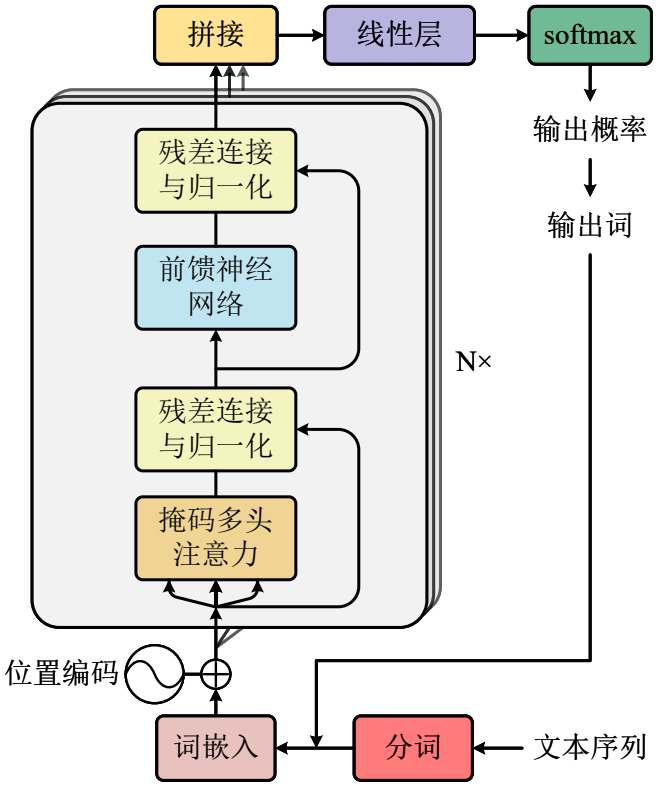


图 2.1　Transformer 的 Decoder-only 结构

首先，输入文本序列中的词元通过嵌入层被映射为低维稠密向量表示。与此同时，为了引入序列的位置信息，需要加入位置编码（Positional Encoding）[77]，使模型能够感知词元的相对或绝对位置，从而理解文本的顺序结构。随后，嵌入表示被送入多层 Transformer 模块进行特征提取。每一层主要包含两个子结构，分别是掩码多头自注意力机制[78]和前馈神经网络[79]，并在每个子结构外引入残差连接与层归一化（Add & Norm）。其中，自注意力机制用于捕获序列中词元之间的依赖关系，前馈神经网络用于增强模型的非线性表达能力。

在自注意力计算过程中，输入表示矩阵 $\mathbf{X}$ 首先通过线性变换步骤映射为查询（Query）、键（Key）和值（Value）：

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V \tag{2.5}$$

其中，$\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ 为可学习的参数矩阵。得到映射值之后再通过查询与键的点积计算注意力权重，并进行缩放与归一化处理：

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.6}$$

其中，d_k 表示键向量的维度，用于对点积结果进行缩放，以提升训练稳定性。

在多头注意力机制中，模型并行计算多个注意力头，从不同表示子空间中捕获上下文信息。各注意力头的输出拼接后，再通过线性变换得到最终表示：

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O \tag{2.7}$$

其中，head_i 表示第 i 个注意力头输出，Concat 是拼接函数，$\mathbf{W}_O$ 为输出映射矩阵。该结构能够在多个语义子空间中同时建模文本特征，显著提升表示能力。此外，掩码机制保证了模型在生成过程中仅依赖历史信息，满足自回归生成的因果性要求。

前馈神经网络用于对特征进行逐位置的非线性变换，其结构通常由两个线性变换与一个非线性激活函数[80]组成。该模块首先将特征维度映射至更高维空间，在高维空间中进行如 ReLU 或 GELU 等函数作用下的非线性变换后，再执行映射回原始维度的操作，非线性操作增强了模型对复杂数据分布的拟合能力。残差连接通过将子层输入与输出进行逐元素相加，实现信息的跨层传递，有效缓解深层网络中的梯度消失问题。结合层归一化操作，可以进一步稳定训练过程并加速模型收敛。Transformer 架构的核心自注意力机制实现对全局上下文的建模，使模型输出的概率分布能够反映复杂语义关系，为基于概率调制的水印嵌入提供了基础。

2.1.3 采样策略与概率控制机制

在获得模型输出的概率分布 P_t 后，生成系统需要依据该分布选择具体词元作为当前输出。最直接的方式是选择概率最大的词元，即

$$x_t = \arg\max_i P_t(i) \tag{2.8}$$

其中 i 表示词表 $\mathcal{V}$ 中的候选词元索引，该方法称为贪婪搜索（Greedy Search）[81]。贪婪搜索的计算开销较低，不过始终选择最优局部解，生成文本可能面临缺乏多样性的问题，甚至出现重复或模式化表达。为缓解上述问题，并在生成质量与多样性之间取得平衡，实际应用中通常采用多种采样策略对原始概率分布进行调控。这些方法本质上是对分布形态或候选空间进行调整，以此影响最终生成结果。

首先，温度采样（Temperature Sampling）[82]通过引入温度参数 T 对 logits 进行缩放，实现调节概率分布的平滑程度这一目的：

$$P'_t(i) = \frac{\exp(\mathbf{z}_t(i)/T)}{\sum_{j=1}^{n} \exp(\mathbf{z}_t(j)/T)} \tag{2.9}$$

其中 $T > 0$ 为温度系数。当 $T < 1$ 时，概率分布更加尖锐，高概率词元的优势被进一步放大；当 $T > 1$ 时，分布趋于平缓，低概率词元会获得更高的采样机会，提升了生成文本的多样性。因此，温度参数在控制生成过程的确定性与随机性之间起到关键作用。

其次，Top-k 采样[83]通过限制候选词元集合的规模来抑制低概率噪声词元。具体而言，仅保留概率最高的 k 个词元构成候选集合 Top-k，并在该集合上进行重新归一化采样：

$$P'_t(i) = \begin{cases} \frac{P_t(i)}{\sum_{j \in \text{Top-}k} P_t(j)}, & i \in \text{Top-}k \\ 0, & i \notin \text{Top-}k \end{cases} \tag{2.10}$$

该方法通过截断低概率词元，有效减少噪声干扰，从而提高生成文本的稳定性与可控性。

进一步地，Top-p 采样（Nucleus Sampling）[84]采用动态候选集策略，根据概率分布的累积质量自适应地截断候选空间。对概率分布 P_t 按概率从大到小排序，得到序列 $\{P_{(1)}, P_{(2)}, \ldots, P_{(n)}\}$，构造满足累积概率阈值 p 的最小前缀集合：

$$P'_t(i) = \begin{cases} \frac{P_t(i)}{\sum_{j \in S} P_t(j)}, & i \in S \\ 0, & i \notin S \end{cases}, \quad S = \left\{ (i) \,\middle|\, \sum_{j=1}^{i} P_{(j)} \geq p \right\} \tag{2.11}$$

其中 $P_{(i)}$ 是 P_t 排序后的第 i 个词元概率，$p \in (0, 1)$ 。集合 S 为满足累积概率首次达到阈值 p 的最小候选集合，对此进行归一化采样。Top-p 根据概率分布形态自适应调整候选空间，在保证生成质量的同时，能够更好地保留生成多样性，有效抑制低概率噪声词元对生成结果的影响。

此外，束搜索（Beam Search）[85]通过维护多个候选序列来寻找整体概率较优的生成路径。该方法在机器翻译等任务中表现良好，但因其倾向于生成高概率、低多样性的文本，在开放式生成场景中应用相对有限。上述采样策略不仅影响文本生成的流畅性与多样性，因其会改变分布形态或裁剪候选集合，还对文本水印的嵌入效果产生直接影响。例如，温度参数较高时，概率分布趋于平滑，水印信号的显著性不可避免地被降低；在 Top-k 或 Top-p 采样过程中，若带水印的目标词元未进入候选集合，可能导致水印嵌入直接失效。在后续水印算法设计当中，需要综合考虑采样机制与概率调制之间的相互作用，以实现文本质量与水印性能之间的合理权衡。

2.2 文本水印的基本原理与形式化建模

文本水印技术是信息隐藏与数字版权保护领域的重要研究方向，其核心目标是在不影响文本语义表达与语言流畅性的前提下，将特定标识信息嵌入文本内容，使该信息对普通读者不可感知，同时能够通过特定算法进行检测或恢复。与图像和音频等连续信号不同的是，自然语言具有离散性强、语义敏感度高等特点，任何不当干预都可能引起语义偏移或生成质量下降。因此，文本水印设计需要在隐蔽性、鲁棒性与检测可靠性之间进行权衡。为系统描述文本水印的嵌入与检测机制，下面将介绍文本水印的基本定义及分类方法，分析基于概率调制的水印嵌入原理，最后给出水印检测的统计模型及其形式化表达。

2.2.1 文本水印的定义与分类

文本水印系统可以形式化表示为嵌入函数与检测函数的组合，其中嵌入函数负责将水印信息注入文本生成或文本编辑过程，检测函数则依据文本中的统计特征或结构特征判断其是否包含特定水印信号。在实际应用中，文本水印不仅可以用于生成内容来源追踪，还能够应用于版权保护、内容认证以及生成式人工智能监管等场景，近年来逐渐成为自然语言处理与信息安全交叉领域的重要研究方向。从实现方式与嵌入层次的角度来看，现有文本水印方法大致可以分为以下几类。

格式型水印（Format-based Watermarking）通常是调整文本中的标点符号、空格数量、字符间距、大小写形式或换行位置等非语义格式特征以实现信息嵌入。不直接修改文本的语义内容使其具有较好的隐蔽性，从实现方式上来看也相对简单。不过嵌入信息仅依赖于文本格式特征，而格式信息在复制、转码、排版或规范化处理过程中往往容易被自动重置，所以格式型水印的鲁棒性通常较弱。随着文本处理系统标准化程度不断提高，许多格式差异会在预处理阶段被自动消除，这进一步限制了格式型水印在实际复杂场景中的应用范围。

词汇型水印（Lexical-based Watermarking）会在语义相近的词汇或表达之间进行受控选择来实现信息嵌入。相比于格式型水印，词汇型水印能够更自然地融入文本内容，在一定程度上具有更好的隐蔽性与可读性。水印效果同近义词资源的覆盖范围与语义匹配质量紧密相关，当候选词汇数量有限或上下文语义约束较强时，可用于嵌入的信息容量往往受到限制。而且词汇型水印对自动改写、同义替换以及文本润色等操作较为敏感，在经历语义保持改写后，很难保持水印的检测性能。词汇型

水印虽然在文本自然性方面具有一定优势，但其鲁棒性仍面临较大挑战。

语义结构水印（Semantic-based Watermarking）通过调整句子结构或表达方式实现水印嵌入。与单纯依赖词汇替换的方法相比，语义结构水印更多关注文本在句法层面或语义层面的表达变化，可以提升水印对局部词汇修改的抵抗能力。但自然语言结构具有较强复杂性，实现该类方法通常需要模型具备较高的语义理解与句法分析能力，同时还需要保证调整后的文本在语义一致性与语言流畅性方面不出现明显问题。在实际应用中往往需要依赖较强的语言生成与语义建模能力完成水印系统的搭建，实现复杂度相对较高。当文本经过深度改写或跨模型再生成时，原有句法结构发生较大变化的情况会对其稳定性产生一定影响。

随着大语言模型的快速发展，生成过程水印（Generation-based Watermarking）逐渐成为当前文本水印研究中的主流方向。水印嵌入发生在文本生成内部，能够较自然地融入模型生成流程，对文本语义与语言流畅性的影响相对较小。当前许多主流方法均采用基于概率偏置的嵌入机制，即通过对特定候选词集合施加概率增益，引导模型在保持生成合理性的同时嵌入水印信息。相比传统后处理方法，生成过程水印通常具有更好的隐蔽性与工程可实现性，并且无需对模型参数进行大规模修改，因此更适用于大语言模型场景。不可避免地，该类方法仍然面临鲁棒性不足的问题，如何在保持文本自然性的同时提升水印稳定性，成为当前研究的重要方向。

从水印所承载信息的角度来看，文本水印还可以划分为零比特水印与多比特水印两类。零比特水印判断文本是否来源于特定模型或系统，更加关注检测准确率与鲁棒性问题。其实现相对简单，在实际内容溯源与生成内容识别场景中具有较高应用价值。多比特水印能够在文本中嵌入更加丰富的标识信息，实现更加细粒度的来源追踪与版权管理。承载更多信息的同时也伴随着对嵌入容量与编码机制的更高要求，而且更易对文本质量产生影响。在大语言模型生成文本水印实际设计过程中，零比特水印在鲁棒性、隐蔽性和可检测性之间达成平衡，而多比特水印在当前文本生成环境下，很难同时兼顾这三点。

2.2.2 基于概率偏置的水印嵌入机制

在大语言模型的文本生成过程中，每一步词元的选择均基于模型输出的概率分布。通过对候选词元概率进行受控调制，可以在尽量保持语义一致性与语言流畅性的前提下，引入可检测的统计偏差信号，并在生成序列中逐步累积，使水印在整体

统计意义上可被识别。在时间步 t，模型输出对应的 logits 向量：

$$\mathbf{z}_t = [z_t(1), z_t(2), \ldots, z_t(\mathcal{V})] \tag{2.12}$$

其中 $z_t(i)$ 表示词元 i 在当前上下文条件下的未归一化得分。通过 softmax 归一化，可得到条件概率分布：

$$P_t(i) = \frac{\exp(z_t(i))}{\sum_{j\in\mathcal{V}} \exp(z_t(j))}, \quad i \in \mathcal{V} \tag{2.13}$$

作为后续采样与概率调制的基础对象，其结构直接决定生成结果的统计特性。

为嵌入水印信息，需要对词元空间进行结构化划分。在每一时间步 t，将词表 $\mathcal{V}$ 划分为两个互不相交的子集：

$$\mathcal{V} = \mathcal{G}_t \cup \mathcal{R}_t, \quad \mathcal{G}_t \cap \mathcal{R}_t = \varnothing \tag{2.14}$$

其中 $\mathcal{G}_t$ 与 $\mathcal{R}_t$ 分别表示绿色词表与红色词表。该划分通常由密钥控制的伪随机函数生成，保证嵌入过程的不可预测性。而后在 logits 层对绿色词表中的词元施加加性偏置，实现对概率分布的结构化调制：

$$z'_t(i) = \begin{cases} z_t(i) + \gamma, & i \in \mathcal{G}_t \\ z_t(i), & i \in \mathcal{R}_t \end{cases} \tag{2.15}$$

其中 γ 为偏置强度参数，用于控制水印信号的嵌入幅度。再对调整后的 logits 进行归一化处理，得到新的概率分布：

$$P'_t(i) = \frac{\exp(z'_t(i))}{\sum_{j\in\mathcal{V}} \exp(z'_t(j))} \tag{2.16}$$

由于 softmax 的指数特性，加性偏置在概率空间中表现为对绿色词表中词元的指数放大，使其在采样过程中具有更高的被选中概率。

随着生成过程的推进，该局部概率偏移将在文本序列中逐步累积，从而在整体分布层面形成可检测的统计特征。偏置参数 γ 的取值需要在生成文本质量与水印检测显著性之间进行权衡。进一步地，为提升水印容量，可将词表划分为多个子集：

$$\mathcal{V} = \mathcal{G}_t^{(1)} \cup \mathcal{G}_t^{(2)} \cup \cdots \cup \mathcal{G}_t^{(m)} \tag{2.17}$$

不同子集与不同编码符号建立对应关系以实现多比特信息嵌入，但同时也会增加检测复杂度与误差累积风险。总之，基于概率偏置的文本水印方法具有实现简单、无需修改模型参数等优点，但其在鲁棒性与文本质量之间仍需进行合理权衡。

2.2.3 文本水印检测统计原理

文本水印检测的核心在于判断生成序列中是否存在由嵌入机制引入的系统性概率偏移。该问题可形式化为统计假设检验，即检验观测序列是否显著偏离无水印条件下的随机分布。为刻画水印信号，引入指示函数 $\mathbf{1}_{\{x_t \in \mathcal{G}_t\}}$ 表示第 t 个词元是否属于绿色词表，则绿色词元的总出现次数为

$$k = \sum_{t=1}^{N} \mathbf{1}_{\{x_t \in \mathcal{G}_t\}} \tag{2.18}$$

其中 N 表示待检测文本长度。在对应于无水印的原假设 H_0 下，绿色词表对模型而言等价于随机划分。若绿色词表比例为 p，并假设各时间步近似独立，则随机变量 k 可近似建模为二项分布：

$$k \sim \text{Binomial}(N, p) \tag{2.19}$$

其期望与方差为

$$\mathbb{E}[k] = Np, \quad \text{Var}(k) = Np(1-p) \tag{2.20}$$

该独立性假设在实际语言生成过程中虽然不严格成立，但在序列较长且依赖关系相对局部时，该近似通常能够较好地刻画统计行为。

在存在水印这一判定对应的备择假设 H_1 下，由于绿色词表词元被施加正向偏置，其采样概率相对提高，使统计量 k 相较于无水印情形产生正向偏移。为刻画该偏移程度，引入标准化统计量（z-score）：

$$z\text{-score} = \frac{k - Np}{\sqrt{Np(1-p)}} \tag{2.21}$$

根据中心极限定理，当 N 足够大时，二项分布可由正态分布近似，在原假设 H_0 下有 z-score $\approx \mathcal{N}(0,1)$。故将检测问题转化为单侧假设检验：给定显著性水平 α，选取对应阈值 $\tau = \Phi^{-1}(1-\alpha)$，当 z-score $> \tau$ 时拒绝原假设 H_0，判定文本中存在水印信号。从检测性能角度来看，显著性水平 α 控制误检率（False Positive Rate），而统计

量的偏移程度则影响检测功效（Detection Power）。当水印强度较弱或文本长度较短时，统计偏移可能不足以跨越判决阈值，导致漏检等后果。

在多子集划分场景下，需要刻画多个子集的分布偏移，构造卡方统计量：

$$\chi^2 = \sum_{i=1}^{m} \frac{(k_i - Np_i)^2}{Np_i} \tag{2.22}$$

其中 k_i 是第 i 个子集的观测次数，理论期望为 Np_i。在原假设成立条件下，χ^2 统计量近似服从自由度为 $m-1$ 的卡方分布。卡方统计量用于衡量观测分布与理论分布之间的整体偏差，实现对多比特水印的联合检测。基于统计量的水印检验方法依赖生成文本的统计特性完成水印判别，无需访问模型内部参数，具有良好的可扩展性与实际应用价值。

2.2.4 文本水印系统的形式化建模

为统一刻画大语言模型生成过程中的文本水印机制，可将其抽象为由嵌入函数与检测函数构成的形式化框架。嵌入函数在文本生成过程中引入水印信号，检测函数则对生成文本进行统计判别，判断其是否包含预设水印。嵌入函数可表示为：

$$W(M, C, K, \gamma) \to X \tag{2.23}$$

其中，M 代表生成式语言模型，C 是输入上下文，K 为密钥，γ 是水印嵌入相关参数，本文中多指概率偏置强度，X 是生成文本序列。在该过程中，嵌入函数通常通过密钥控制的随机机制，对模型输出的概率分布进行调制，使生成结果在统计意义上产生可检测的偏移。与之对应，水印检测过程可表示为判别函数：

$$D(X, K, \tau) \to \{0, 1\} \tag{2.24}$$

其中，X 为待检测文本，K 是和嵌入阶段相同的检测密钥，τ 为判定阈值。当输出结果为 1 时，表示检测到水印信号；当输出结果为 0 时，则表示当前文本未通过水印检测。检测函数通常基于密钥重建对应的概率划分或映射关系，并结合文本统计特征构造检测统计量，通过假设检验方法判断文本是否存在显著偏移。阈值 τ 的选取将直接影响误检率与漏检率之间的权衡。

现有基于概率偏置的文本水印方法通常采用固定偏置策略，实现简单，能够在一定程度上形成稳定统计特征，但调制强度在生成过程中保持不变限制了水印信号

在序列中呈现相对集中的分布结构。当文本受到语义保持改写、局部替换或跨模型再生成等扰动时，这种结构特征易被破坏，直接引发检测统计量下降，进而影响水印检测性能。此外，固定偏置策略缺乏对上下文变化的自适应能力，使得水印嵌入在隐蔽性与鲁棒性之间难以取得平衡。因此，设计更加灵活的概率调制机制，使水印信号能够随上下文动态变化，并在生成序列中以更加分散且稳定的方式存在，对于提升水印系统在复杂扰动环境下的检测可靠性具有重要意义。

总体而言，生成式文本水印系统可视为一个由概率调制与统计判别共同构成的闭环过程。嵌入阶段的策略是对生成概率分布进行结构化调制，在文本中逐步累积水印信号；检测阶段通过统计分析识别该分布偏移，实现对文本来源的判定。从系统设计角度出发，文本水印方法通常需要在隐蔽性、鲁棒性与安全性之间进行综合权衡，隐蔽性关注生成质量影响，鲁棒性关注扰动下的可检测性，而安全性要求在未知密钥条件下难以被伪造或移除。

2.3 文本水印性能评价指标体系

与传统数字水印类似，文本水印不仅要求能够对生成内容实现稳定、准确的来源识别，还需要在尽量不影响文本语义表达与语言自然性的前提下保持较好的隐蔽性与可用性。在对词元生成概率分布施加受控调制实现水印的方法下，水印嵌入强度会直接影响检测性能与文本质量之间的平衡关系。较强的概率扰动虽然有助于增强统计检测信号，但可能降低文本流畅性与自然度，而较弱的扰动虽然能够更好地保持生成质量，却可能导致检测显著性下降。当文本经历改写、删减或再生成等后处理操作时，水印信号可能出现衰减而进一步影响检测结果。本文从检测性能、文本质量与鲁棒性三个维度构建统一的评价指标体系，全面评估文本水印方法在实际应用场景中的综合性能。

2.3.1 检测性能指标

检测性能指标用于衡量水印检测算法区分带水印文本与无水印文本的能力。该问题可形式化为二分类任务，根据检测结果与真实标签的关系，可将样本划分为真阳性（TP）、假阳性（FP）、真阴性（TN）和假阴性（FN）。真阳性率（TPR）与假阳

性率（FPR）是最基本的评价指标，分别表示对正样本的识别能力与误检水平：

$$\text{TPR} = \frac{TP}{TP + FN} \tag{2.25}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{2.26}$$

两者之间通常存在权衡关系，在实际应用中往往需要对 FPR 进行约束以控制误报风险。F1-score 通过综合考虑精确率与召回率，对不平衡数据具有更稳健的评价能力：

$$\text{F1} = \frac{2TP}{2TP + FP + FN} \tag{2.27}$$

对于基于统计检验的检测方法，常利用标准化统计量 z-score 刻画观测分布相对于无水印假设的偏离程度，数值越大，表明水印信号越显著。该类方法的检测性能通常与文本长度及嵌入强度密切相关。为分析不同阈值下的检测表现，常采用 ROC 曲线进行评估。该曲线以 FPR 为横轴、TPR 为纵轴，曲线越接近左上角说明判别能力越强。进一步地，通过曲线下面积（AUC）对整体性能进行量化，其取值范围为 $[0, 1]$。另外，还可在固定误报率约束下评估检测能力，如在给定 FPR=α 条件下考察对应的 TPR（TPR@FPR=α），以反映严格约束下的有效检测性能。

2.3.2 文本质量指标

在数字水印系统中，文本质量指标主要用于评估水印嵌入过程对生成文本自然性、语义一致性及表达多样性的影响。考虑到文本水印对语言模型生成概率分布施加偏置来嵌入隐式标识信息，会对原有生成分布造成一定的干扰，不可避免地影响文本质量。评价文本水印方法时，需要从多个维度对水印嵌入前后的文本进行系统分析。现有研究从流畅性、表层相似性、语义一致性以及多样性等方面构建综合评价指标体系，为后续评估水印机制在隐蔽性与生成质量之间的平衡能力打下基础。

1）流畅性指标衡量生成文本是否符合自然语言的统计规律，即句子通顺且符合语法，困惑度（Perplexity, PPL）是其中最常用的评价指标之一，定义为：

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{t=1}^{N}\log P(x_t \mid x_{<t})\right) \tag{2.28}$$

困惑度越低，说明文本越符合语言模型学习到的概率分布，生成结果越自然流畅。在文本水印场景中，若水印嵌入对生成分布的干预过强，可能导致困惑度升高，该指标直接反映出文本流畅性的下降。需要注意的是，困惑度依赖于具体语言模型，不同模型之间的评估结果可能存在差异，实验中应该保持模型的一致。

2）表层相似性指标评估水印嵌入前后文本在词汇与句法层面的接近程度，常用指标包括 BLEU 与 METEOR 等。在数学层面具体表现为基于 n-gram 匹配程度的计算，BLEU 主要关注精确匹配比例，而 METEOR 还会加入词形变化与同义词匹配等考量，提供更为灵活的相似性评估。表层相似性指标能够有效反映文本改写过程中是否保持了原有表达结构，所以对语序变化较为敏感，当遇到语义等价但表达不同的文本时可能出现评估不足的情况，因而往往结合语义层面的指标进行综合分析。

3）语义一致性指标衡量水印嵌入前后文本在语义表达上的保持程度。常见方法是使用 Sentence-BERT 等预训练语义表示模型将文本映射为向量表示，再通过余弦相似度计算其语义接近程度：

$$\text{Sim}(X, Y) = \frac{x \cdot y}{|x||y|} \tag{2.29}$$

其中，x 与 y 分别表示原始文本与水印文本的语义向量。相似度越高，说明语义保持程度越好。部分研究还引入蕴含分数（Entailment Score）来判断两个文本之间是否存在语义蕴含或冲突关系，从逻辑一致性的角度补充语义相似度指标的不足。语义一致性能够反映水印机制是否对原有语义内容产生明显干扰。如果水印嵌入导致文本出现语义偏移或上下文逻辑变化，相似度也就相应地变现出下降趋势。

4）多样性指标评估生成文本在表达形式上的丰富程度，反映水印机制对生成分布的约束强度。采用 Log Diversity 指标综合刻画不同 n-gram 唯一性比例，定义为：

$$\text{LogDiversity} = -\log\left(1 - \sum_{n=1}^{N}(1 - u_n)\right) \tag{2.30}$$

其中，u_n 表示 n-gram 的唯一性比例，统计生成文本中不同 n-gram 在全部 n-gram 中的占比，N 表示最大 n 阶数。该指标通过对不同阶 n-gram 多样性信息进行聚合，从整体层面刻画生成文本的表达分布特性，数值越大表示文本在词汇与结构层面越丰富、重复程度越低。u_n 从局部统计角度反映文本重复情况，是 Log Diversity 进行跨阶聚合的重要基础。在文本水印方法中，该指标能够反映生成分布的集中程度。当水印机制对特定词元产生较强偏置时，可能引入局部重复，降低整体多样性。

综上所述，文本质量评估通常从流畅性、表层相似性、语义一致性以及多样性等多个维度对水印嵌入后的生成结果进行综合分析。其中，困惑度反映文本对语言模型分布的符合程度，表层相似性指标刻画文本表达形式的接近程度，语义一致性指标关注语义内容的保持情况，多样性指标衡量生成分布的丰富性。由于不同指标侧重的评价维度存在差异，单一指标不能全面反映文本质量的变化，实际研究做法是对多种指标进行联合评估，以更加客观、全面地分析文本水印方法对生成质量的影响。

2.3.3 鲁棒性指标

鲁棒性指标用于评估文本水印在受到外部扰动或攻击条件下的可检测能力，是衡量水印系统实际应用价值的重要指标之一。自然语言具有较强的语义灵活性与表达多样性，攻击者即使在不改变文本核心语义的情况下，也能够通过改写、删减或重新生成等方式对文本进行调整，破坏水印在统计层面的分布特征。这种语言层面的高自由度使得文本水印相比传统数字水印更容易受到扰动影响，鲁棒性问题成为当前文本水印研究中的核心挑战之一。在文本水印研究中，常见攻击方式主要包括改写攻击、删除攻击、插入攻击以及再生成攻击等。

1）改写攻击（Paraphrase Attack）主要通过同义词替换、句式变换或语序调整等方式改变文本表达形式，同时尽量保持原有语义不变。这类攻击能够在较大程度上改变词元分布结构，从而削弱基于概率偏置的水印信号。攻击者可以利用自动改写模型对原文本进行重写，使原本受到偏置控制的词元被新的表达方式替代，进而降低检测统计量。由于改写后的文本在语义层面仍保持较高一致性，该类攻击通常被认为是评估文本水印鲁棒性的典型场景之一。

2）删除攻击（Deletion Attack）通过删除文本中的部分词元、句子或段落来破坏水印结构。由于许多文本水印方法依赖于序列中的统计累积特征，当有效文本长度减少时，水印检测所依赖的统计样本数量也会随之下降，降低检测显著性。在极端情况下，如果包含较强水印信号的关键部分被删除，可能直接导致检测失败。总而言之，删除攻击主要通过减少有效统计信息的方式影响水印检测能力。

3）插入攻击（Insertion Attack）则通过向原文本中加入额外内容来稀释水印信号。攻击者可能插入与主题相关但不包含水印特征的文本，使原有水印统计特征在整体序列中的占比下降。对于依赖词元频率统计的检测方法而言，插入大量无关内

容可能导致检测统计量趋近于无水印状态，从而影响检测结果。该类攻击本质上是通过扩展文本规模来削弱水印信号的相对强度。

4）再生成攻击（Regeneration Attack）通常被认为是对文本水印威胁较大的攻击方式之一。该方法利用生成式语言模型对原文本进行重新生成或摘要重写，在保留原始语义的基础上重新构建文本表达结构。由于再生成过程会重新采样新的词元序列，原有水印分布特征可能被大幅改变甚至完全消除。尤其是在跨模型再生成场景下，不同模型之间的生成分布差异可能进一步削弱原始水印信号，使检测性能显著下降。因此，再生成攻击通常被用于评估文本水印在复杂生成环境中的鲁棒性。

不同攻击方式对水印的影响机制存在明显差异。改写攻击与再生成攻击改变文本的概率分布结构来破坏水印特征，而删除攻击与插入攻击则分别通过减少有效样本数量或稀释统计特征来影响检测结果。仅依赖单一攻击场景往往难以全面反映水印系统的稳定性，现有研究在多种攻击条件下对检测性能进行综合测试，分析水印在复杂环境中的保持能力，更加全面地评估文本水印算法的稳定性与抗攻击能力。这不仅有助于揭示不同水印机制在复杂场景下的性能差异，也能够为后续方法的优化提供理论依据，提升生成式文本水印系统在实际应用中的可靠性与安全性。

2.4 本章小结

本章围绕生成式大语言模型文本水印的理论基础进行了系统分析。首先，从文本生成机制出发，介绍了大语言模型的自回归生成框架、Transformer 注意力机制以及常用概率采样策略，明确了水印嵌入的实现基础与作用位置。然后，对文本水印方法进行了归纳，重点分析了基于概率调制的水印嵌入机制，为后续方法设计提供了理论支撑。最后，构建了涵盖检测性能、文本质量与鲁棒性的评价指标体系。综上，本章从生成机制、方法建模与性能评价三个方面对文本水印问题进行了系统梳理，构建了统一的分析框架。后续将在此基础上开展具体水印方法的设计与实现。

第三章　基于多偏置调制的文本水印

3.1　引言

生成文本水印是如今对模型生成内容进行有效标识和来源追溯的技术热点，通过在大语言模型生成过程中隐入不可觉察的水印信息来实现。知识产权保护作为人工智能时代下极具挑战的话题之一，激励了人工智能生成内容检测、图像水印、音频水印、视频水印及文本水印等技术手段的快速发展。因自然语言是高密度、低冗余的精确载体，文本水印嵌入技术相较于其他载体的发展稍显滞后，自 2017 年 Transformer 的兴起带动了生成文本质量的提高，文本水印技术迎来突破性进展。现有文本水印方法可实现有效性，但在鲁棒性方面表现较差。

KGW 方法的提出开创了大语言模型水印范式，为传统后处理水印方法到生成水印方法的转变奠定了实用且有效的技术基础。KGW 方法在生成过程中为特定词元的 logits 添加偏置，引导这部分词元被更大概率地选择，与原模型生成文本有所区别即完成水印嵌入。该类方法不仅几乎无需多余算力，还实现了较高精度的检测，文本质量和水印有效性均保持良好。给人工智能文本打水印的理论构想变成落地标准，不过人工智能文本不可避免地会被人为修改，其在面临文本攻击时鲁棒性不足的问题是后续研究的关键。基于上述分析，本章提出一种基于多偏置调制的大语言模型生成文本水印嵌入与检测框架，保持文本生成质量的同时具有更好的鲁棒性。

3.2　总体框架

图 3.1 展示了基线 KGW 方法与本文基于多偏置随机选择机制的文本水印方法在水印嵌入阶段的整体结构对比，两种方法均在大语言模型生成过程中完成水印嵌入。基线方法主要采用固定偏置调制机制，本章方法在保留上下文哈希与动态词表划分机制的基础上，水印嵌入的实现主要由上下文哈希模块、伪随机控制模块、多偏置生成模块、词表划分模块以及概率调制模块构成，各模块在统一控制信号下协同作用，完成水印信号的稳定嵌入。将传统固定偏置调制扩展为上下文驱动的动态随机调制过程，使水印信号在时间维度上呈现更加分散的统计分布特征，有效降低固定模式被学习或预测的风险。

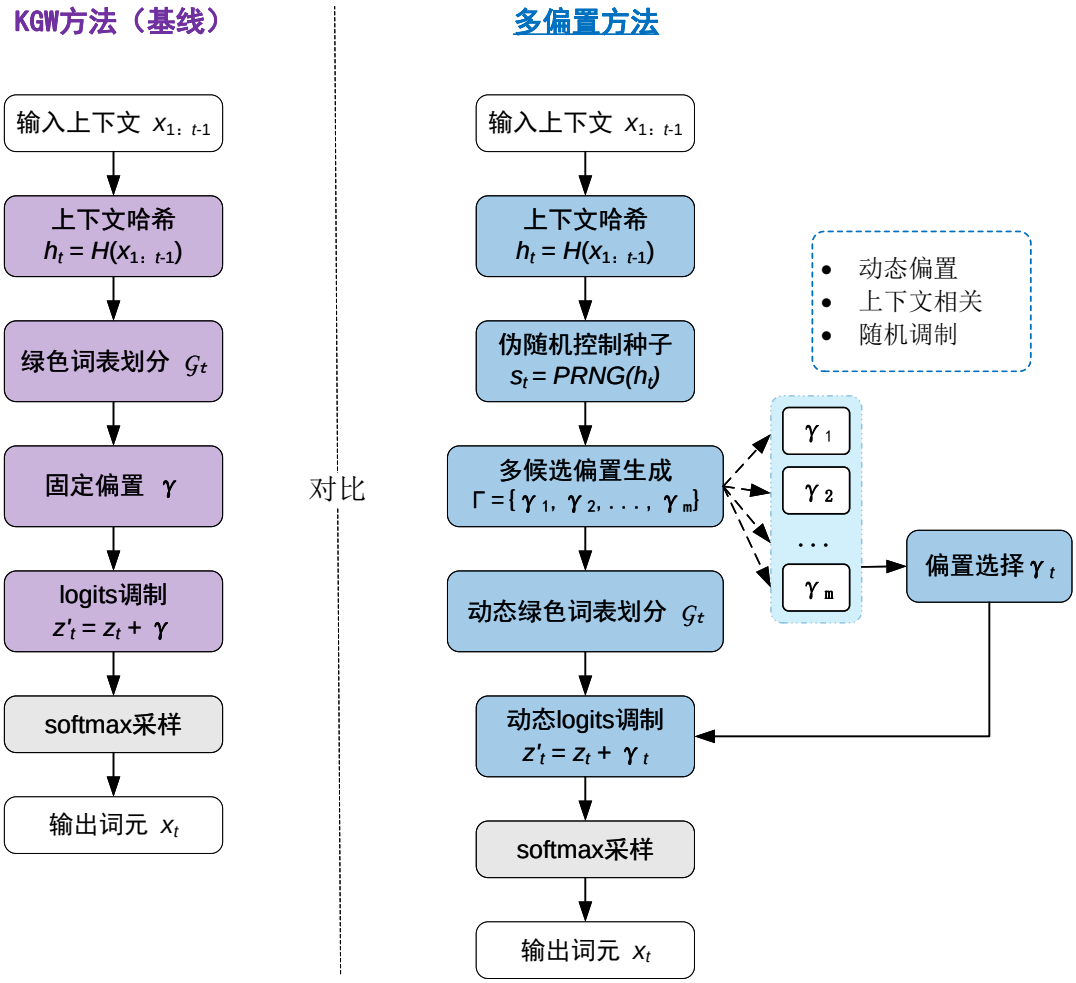


图 3.1 KGW 方法与本文多偏置文本水印方法总体框架对比

在文本生成阶段，模型在每一个时间步基于当前上下文预测下一个词元的概率分布。首先，上下文哈希模块对已有生成序列进行编码，得到与上下文相关的随机种子，该种子作为后续随机过程的输入。随后，伪随机控制模块基于该种子生成控制信号，用于统一驱动偏置选择与词表划分过程。由于该控制信号由上下文决定，水印嵌入过程能够与文本内容保持一致，保证了水印的隐蔽性与安全性。

在概率调制过程中，多偏置生成模块在每个时间步生成多个候选偏置值，并依据伪随机控制信号选择其中一个用于当前分布的调制。相较于单一固定偏置，该策略在时间维度上引入了更多随机性，使不同位置的水印信号呈现差异化特征。与此同时，词表划分模块在相同随机机制下对词表进行划分，得到对应的绿色词表与红色词表，保证偏置选择与词表划分的一致性。经上述操作后，概率调制模块对绿色词表中词元的 logits 施加选定偏置，提升其被采样的概率，顺理成章地完成水印信息

的嵌入过程。

在水印检测阶段，采用与嵌入过程一致的上下文哈希机制与伪随机生成策略，对给定文本进行重建分析。具体而言，通过对文本序列进行逐步解析，恢复每一位置对应的词表划分结果，才能识别哪些词元属于绿色集合。随后，统计生成序列中绿色词元的出现比例，结合统计检验方法构造标准化检测量。将该检测量与预设阈值进行比较，可以判断文本中是否包含水印信号，系统层面上实现对生成内容来源的有效识别。

总之，该框架在不修改模型参数的条件下，通过引入基于上下文的随机调制机制，实现了水印嵌入过程与文本生成过程的紧密结合，而且使水印信号在序列中呈现分散分布，很好地提升其隐蔽性与抗攻击能力。另外，该方法在保证可检测性的前提下，尽可能减小对文本质量的影响，在鲁棒性与自然性之间取得平衡。后续章节将对各模块的具体设计与实现细节进行进一步说明。

3.3 水印嵌入

3.3.1 上下文哈希机制

在文本水印嵌入过程中，如果仅依赖固定规则或静态随机策略对生成概率进行调制，容易导致水印模式呈现出稳定且可预测的结构特征，增加被攻击者识别、分析甚至逆向去除的风险。为了提升水印系统的安全性与隐蔽性，结合上下文哈希机制，本文将水印嵌入过程与生成文本的上下文动态绑定，使得不同语境下的水印控制信号呈现出显著差异，从源头上避免攻击者拿到可被利用的固定规则。上下文哈希机制通过对当前时间步 t 的历史生成序列进行编码，构造一个与上下文强相关的确定性随机种子，并结合预设密钥共同参与计算，形成具有访问控制能力的随机驱动信号，伪随机生成机制的使用使得相同输入条件下系统输出保持一致且在统计意义上又具有随机分布特性。该过程可以形式化表示为：

$$s_t = H_{key}(C_t), \quad r_t = G(s_t, t) \tag{3.1}$$

其中，C_t 表示当前时间步的上下文信息，H_{key} 表示受密钥控制的哈希函数，s_t 为生成的随机种子，G 为伪随机生成函数，r_t 为最终用于控制水印嵌入的随机信号。

从数学形式上，上下文哈希函数可定义为：

$$H : \mathcal{C} \to \{0,1\}^d \tag{3.2}$$

其中，$\mathcal{C}$ 表示上下文空间，d 为哈希输出长度。为增强安全性，引入密钥 key，构造如下形式的哈希函数：

$$s_t = H(C_t \parallel key) \tag{3.3}$$

其中，$\parallel$ 表示拼接操作。该设计使得即使攻击者能够获取生成文本内容，也无法在未知密钥的情况下恢复对应的随机控制序列。实际实现中也可以采用基于 HMAC 的哈希方式，以进一步增强抗碰撞性与安全性。

上下文哈希机制具有两个关键性质。首先是上下文敏感性，即不同上下文输入会产生不同的哈希输出：

$$C_t \neq C_{t'} \Rightarrow H_{key}(C_t) \neq H_{key}(C_{t'}) \tag{3.4}$$

保证了水印嵌入策略在不同生成路径上具有差异性，避免全局一致的水印模式。其次是确定性，即在相同上下文与相同密钥条件下，哈希输出保持一致：

$$C_t = C_{t'} \Rightarrow H_{key}(C_t) = H_{key}(C_{t'}) \tag{3.5}$$

最重要的是，检测过程需要在没有嵌入策略的情况下重建嵌入过程中的随机控制序列，这一性质对于水印检测阶段至关重要。

在得到随机种子 s_t 后，进一步通过伪随机生成函数 G 构造控制信号 r_t。保证在给定种子条件下生成序列具有完全可复现性，同时在统计分布上近似随机。伪随机机制的引入，使得水印嵌入不仅依赖上下文，还具备较强的不可预测性，大大提升了系统的抗分析能力。在具体应用中，伪随机控制信号主要用于驱动词表划分与偏置选择过程。首先基于随机信号对词表进行排列或采样，得到当前时间步对应的绿色词表 $\mathcal{G}_t$，用于后续概率调制。因为该过程由上下文与密钥共同决定，所以不同时间步的词表划分结果彼此独立且不可预测。

综上所述，上下文哈希机制通过结合密钥控制与伪随机生成过程，将水印嵌入策略与文本生成上下文紧密耦合。这一整体设计保证了水印嵌入过程的可复现性，使检测阶段能够准确重建嵌入过程。另外，上下文相关性与随机性的引入，有效提升

了水印系统的隐蔽性、安全性以及对抗分析攻击的能力，为后续多偏置调制机制的实现提供了关键支撑。

3.3.2 多候选偏置生成机制

在传统基于概率偏置的文本水印方法中，以 KGW 方法为代表，通常采用固定偏置强度对伪随机生成的绿色词表进行概率调制。该类方法通过密钥驱动的上下文哈希机制实现词表的动态划分，水印检测过程依赖于相同密钥，具备一定的安全性与可验证性。尽管绿色词表在不同时间步发生变化，对应的概率提升幅度却是保持一致的，生成序列整体表现出较为统一的分布偏移特征，即水印信号在统计意义上呈现相对稳定的调制特征。这种单一调制模式在长文本或大规模样本条件下可能被学习型检测方法捕捉，用于区分带水印文本与自然生成文本。而且，在面对文本改写或扰动时，由于缺乏调制强度上的动态调整机制，水印信号容易整体衰减，从而影响检测的稳定性。

针对上述问题，本文在保持基于密钥的伪随机词表划分机制的基础上，引入多候选偏置生成策略，在每一时间步对调制强度进行动态选择。在偏置层面引入随机性使水印信号在序列中呈现多尺度变化特征，可以实现在不破坏原有检测机制的前提下，进一步提升水印的隐蔽性与鲁棒性。具体而言，在时间步 t，系统首先基于伪随机机制生成一组候选偏置集合：

$$\Gamma_t = \{\gamma_t^1, \gamma_t^2, \ldots, \gamma_t^m\} \tag{3.6}$$

其中，m 表示候选偏置的数量。每个候选偏置由基础偏置项与随机扰动共同构成，其生成过程定义为：

$$\gamma_t^j = \gamma_0 + \delta_t^j, \quad \delta_t^j \sim \mathcal{N}(0, \sigma^2) \tag{3.7}$$

其中，γ_0 表示全局基础偏置强度，δ_t^j 表示服从高斯分布的随机扰动项，σ 控制偏置波动幅度。该设计使得不同时间步以及不同候选偏置之间均存在显著差异，避免了单一调制强度导致的模式固化问题。

在生成候选偏置之后，系统需要确定当前时间步实际使用的偏置参数。为保证该选择过程既具备随机性又保持可复现性，本文引入由上下文哈希生成的控制索引

r_t，并基于该索引在候选集合中进行确定性选择：

$$j_t = r_t \bmod m, \quad \gamma_t = \gamma_t^{j_t} \tag{3.8}$$

其中，j_t 表示当前时间步选中的偏置索引，γ_t 为最终用于概率调制的偏置参数。该选择机制保证在相同上下文与密钥条件下，候选偏置选择结果是确定的，同时由于哈希输出的高敏感性，使得不同上下文下的偏置路径呈现高度不确定性。

多候选偏置生成策略的核心优势在于引入离散调制空间。相较于传统单一标量偏置，本方法在每个时间步构造一个局部偏置分布集合，使水印嵌入不再依赖固定强度，而是表现为随上下文变化的动态调制过程。这种机制显著增加了攻击者对水印规律的建模难度，尤其是在面对重写、截断或采样扰动时，能够有效降低水印结构的可预测性。在候选偏置的生成过程引入随机扰动项不仅确保了统计分布在全局范围内是稳定状态，还在局部时间步呈现出不可见的波动性。这种全局稳定兼顾局部随机的特性，使得模型在保证生成质量不受明显影响的前提下，实现水印强度的细粒度调节，在隐蔽性与鲁棒性之间取得更优平衡。

综上所述，多候选偏置生成机制通过构建偏置集合、引入随机扰动以及基于上下文索引的选择策略，将传统单一偏置调制扩展为动态离散偏置空间。这一机制不仅增强了水印嵌入过程的随机性与表达能力，同时也为后续词表划分与概率调制模块提供了更灵活的控制基础，是本文多偏置文本水印方法的重要组成部分。

3.3.3 词表构建与概率调制

在完成上下文哈希计算与多候选偏置选择之后，水印系统需要进一步确定具体作用于语言模型输出分布的词元集合。基于概率偏置的基本嵌入机制已在第二章中给出，本节在此基础上，结合多候选偏置选择策略，对词表构建与概率调制过程进行针对性设计。在每一个时间步 t，系统基于伪随机控制信号对词表 $\mathcal{V}$ 进行动态划分，生成绿色词表（Greenlist）与红色词表（Redlist）。其中绿色词表的规模由比例系数 α 控制，用于刻画水印嵌入的覆盖范围与强度。该划分过程依赖于上下文哈希与随机控制信号，使得词表划分结果随生成上下文动态变化，避免形成可被识别的固定模式。

在得到绿色词表之后，模型输出分布将在 logits 层进行选择性调制。利用上一节确定的动态偏置参数 γ_t，仅对绿色词表中的词元进行概率增强，而对其余词元保持

不变。该操作本质上是在原始分布上引入局部偏置，使目标子集在采样过程中具有更高的被选中概率。不再采用固定偏置参数，从多候选偏置集合中随机选取 γ_t 用于当前时间步的调制，使不同位置的调制强度呈现离散变化。这种动态调制方式避免了单一强度在全序列中的重复作用，降低水印信号在统计层面的集中性。完成调制后，模型基于调整后的概率分布进行采样生成词元。由于偏置仅作用于部分词元集合，整体分布结构不会发生显著改变，在保证生成质量的同时实现水印嵌入。概率调制过程需与采样策略协同作用，以避免目标词元在候选空间裁剪过程中被过滤，影响嵌入有效性。

该过程通过动态词表划分融合多偏置随机调制的组合方式，在原始语言模型分布上引入受控扰动。由于词表划分与偏置选择均依赖于上下文哈希，嵌入路径随生成过程不断变化，使水印信号呈现出非固定的分布特征。与传统 KGW 方法相比，其采用固定偏置参数对绿色词集合进行统一调制，而本文通过引入时间相关的动态偏置 γ_t，将嵌入过程由静态调制扩展为条件驱动的随机调制机制。这一改进使水印信号在序列中更加分散，提升其在文本改写、截断及局部扰动等场景下的鲁棒性。由于绿色词表划分与偏置选择均由密钥与上下文共同决定，该过程在相同条件下具有确定性，在不同上下文条件下呈现出显著差异性。在保证可检测性的同时，有效降低了水印模式被逆向分析的风险。综上，通过动态词表构建与多偏置调制策略，本文方法实现了对生成分布的细粒度控制，在维持文本自然性的前提下提升了水印嵌入的稳定性与安全性。

3.3.4 嵌入算法流程

多偏置调制文本水印的嵌入过程在 Transformer 架构下逐步执行，不改变模型参数的前提下引入了由上下文驱动的动态概率调制机制，实现水印信息的隐式嵌入。在时间步 t，系统首先基于历史生成上下文 C_t 计算上下文哈希值 h_t，并由此生成随机控制种子。随后，在该控制信号作用下构造候选偏置集合。与传统固定偏置方法不同，本文在每个时间步动态生成多个候选偏置参数，并利用上下文哈希索引确定当前实际采用的偏置强度。为保证生成阶段与检测阶段的一致性，候选偏置中的随机扰动由上下文哈希控制的伪随机生成器产生，使偏置选择过程具备可复现性。

在完成偏置生成后，系统基于相同随机控制信号构建绿色词集合 $\mathcal{G}_t$，用于引导概率分布的局部增强。随后，对语言模型输出 logits 进行选择性调制，并通过 softmax

函数归一化为概率分布，最终基于该分布完成词元采样生成。上述过程不断迭代直至生成结束，以此得到完整的水印文本序列。该嵌入过程本质上使水印嵌入从静态规则映射转化为动态随机调制过程，生成文本中的水印信号呈现出更加分散的统计特征，有效降低固定偏置带来的模式固化问题。为便于描述，将整个嵌入过程形式化为算法 3.1。

算法 3.1: 多偏置调制文本水印嵌入算法

输入: 大语言模型 LLM，词表 $\mathcal{V}$，密钥 key，基础偏置 γ_0，绿色词比例 α，候选偏置数量 m，扰动方差 σ^2，生成长度 T

输出: 水印文本序列 X

初始化：$X \leftarrow \emptyset$;

for $t = 1$ **to** T **do**

  计算上下文哈希：$h_t = H_{key}(X_{<t})$;

  生成候选偏置：$\gamma_t^j = \gamma_0 + \delta_t^j$，$\delta_t^j \sim \mathcal{N}(0, \sigma^2)$;

  构建候选偏置集合：$\Gamma_t = \{\gamma_t^1, \ldots, \gamma_t^m\}$;

  选择当前偏置：$\gamma_t = \Gamma_t[h_t \bmod m]$;

  构建绿色词集合：$\mathcal{G}_t = \text{TokenSelect}(\mathcal{V}, h_t, \alpha)$;

  计算 logits：$z_t = LLM(X_{<t})$;

  偏置调制：$z_t'(i) = z_t(i) + \gamma_t \cdot \mathbb{I}(v_i \in \mathcal{G}_t)$;

  概率归一化与采样：$P_t = \text{softmax}(z_t')$，$x_t \sim P_t$;

  更新序列：$X \leftarrow X \oplus x_t$;

end

return X;

该算法在标准自回归生成框架中引入一个上下文驱动的动态调制层，不改变模型参数，仅在推理阶段对 logits 分布进行结构化扰动来实现水印信息的隐式嵌入。与传统 KGW 方法相比，本文算法在嵌入阶段引入了两个关键改进，多候选偏置机制和上下文哈希驱动的词表划分机制，这两个机制共同作用，使水印嵌入过程从静态规则映射转变为一个随上下文演化的随机过程。该嵌入算法在保证生成文本质量不受显著影响的同时，能够显著增强水印的隐蔽性与抗攻击能力，并为后续基于统计检验的水印检测提供一致性基础。

3.4 水印提取

3.4.1 统计检验

文本水印的提取过程可以建模为一个统计假设检验问题，其基本统计原理已在第二章中给出。本节在此基础上，重点从实现角度出发，对检测阶段的具体计算流程进行描述。在仅给定待检测文本的条件下，结合共享密钥与上下文哈希机制，可以在黑盒设置下重建嵌入阶段的判定规则，实现水印存在性的判别。首先对待检测文本序列 $x = (x_1, x_2, \ldots, x_N)$ 进行逐词元处理。对于每一个位置 i 计算对应的上下文哈希值，并基于该哈希结果生成当前时间步的绿色词表 $\mathcal{G}_i$。该过程需与嵌入阶段保持一致，包括相同的哈希函数、密钥以及随机控制逻辑，保证绿色词表划分在检测端的可复现性。

在得到各位置对应的绿色词表后，对文本序列进行一次线性扫描。对于每个词元 x_i，判断其是否属于当前时间步的绿色词表 $\mathcal{G}_i$，并通过指示函数进行记录。在遍历完整个序列后，得到绿色词命中总次数：

$$X = \sum_{i=1}^{N} \mathbb{I}(x_i \in \mathcal{G}_i) \tag{3.9}$$

其中 $\mathbb{I}(\cdot)$ 为指示函数，X 表示绿色词命中总次数。该统计量反映了文本中偏置词元的整体出现频率，是后续判别的核心依据。

在无水印假设 H_0 下，绿色词表对模型而言等价于随机划分，其命中概率由预设比例参数 α 决定。基于第二章中的建模结果，可直接采用对应的期望与方差形式对统计量进行刻画，并据此构造标准化检验统计量：

$$z\text{-score} = \frac{X - \alpha N}{\sqrt{N\alpha(1-\alpha) + \epsilon}} \tag{3.10}$$

其中 ϵ 为数值稳定项。在实际实现中，该计算仅依赖于统计量 X、文本长度 N 以及参数 α，因此计算复杂度较低，适用于长文本检测场景。在备择假设 H_1 下，由于嵌入阶段对绿色词集合施加了正向概率偏置，词元落入绿色词表的概率整体提高，使得统计量 X 相较于无水印情形产生正向偏移。尽管本文方法中偏置参数 γ_t 随时间步动态变化，但其始终作用于绿色词集合，因此不会改变检测统计量的构造方式，而

仅影响偏移程度。基于上述统计量，可以将检测问题形式化为如下假设检验：

$$\begin{cases} H_0 : \text{文本由未嵌入水印的语言模型生成} \\ H_1 : \text{文本由嵌入水印的语言模型生成} \end{cases} \tag{3.11}$$

最终，根据给定显著性水平确定判决阈值 τ，并构造决策函数：

$$\delta(X) = \begin{cases} 1, & z\text{-score} \geq \tau \\ 0, & z\text{-score} < \tau \end{cases} \tag{3.12}$$

在实际应用中，当统计量超过阈值时判定文本包含水印，否则认为未检测到水印信号。整个检测流程主要包括上下文重建，词表生成，命中统计，统计判决四个关键步骤，各步骤之间依赖关系明确，且无需访问模型内部参数。本章方法在检测阶段不需要额外引入复杂计算，仅需复现词表划分过程即可完成统计判别，同时由于水印信号在序列中分布更加分散，其在面对局部扰动或文本改写时仍能够保持稳定的统计偏移。

3.4.2 提取算法流程

多偏置调制文本水印的提取过程是在 Transformer 架构下逐步执行的，其核心思想是在不访问模型参数的前提下，通过重建由上下文驱动的动态判定机制，实现水印信号的显式识别。该过程在每一个时间步均依赖历史文本序列，保证水印检测策略具有良好的时序一致性与上下文对齐能力。在时间步 t，系统首先基于历史上下文 $x_{<t}$ 计算上下文哈希值 h_t，并由此恢复随机控制信号。而后，根据该信号重建绿色词候选集合，通过一致性映射机制确定当前时间步的词表划分。同时，系统基于相同哈希路径生成绿色词表 $\mathcal{G}_t$，用于刻画潜在的概率偏置方向。

在获得绿色词表后，对待检测文本序列进行逐位置扫描，并通过指示函数统计词元是否落入绿色词表，累积得到绿色词命中计数。随后，在全局范围内对统计结果进行归一化处理，并构造标准化检验统计量，最终基于该统计量完成假设判决。该过程不断累积直至序列结束，以此得到整体的水印检测结果。提取过程的重要量值就是统计判别项，使水印检测从局部规则匹配转化为全局显著性检验过程。该设计不仅提升了检测结果的稳定性，还增强了其在文本扰动场景下的鲁棒性。为便于描述，将整个提取过程形式化为算法 3.2。

算法 3.2: 多偏置调制文本水印提取算法

输入: 待检测文本序列 $X = \{x_1, \ldots, x_N\}$，密钥 key，词表 $\mathcal{V}$，比例参数 α，阈值 τ
输出: 检测结果 $\delta(X)$
初始化：$X_{\text{hit}} \leftarrow 0$;
for $t = 1$ **to** N **do**
    构建上下文 $C_t = x_{<t}$;
    计算哈希值 $h_t = H_{key}(C_t)$;
    构建绿色词集合 $\mathcal{G}_t = \text{TokenSelect}(\mathcal{V}, h_t, \alpha)$;
    if $x_t \in \mathcal{G}_t$ **then**
        $X_{\text{hit}} \leftarrow X_{\text{hit}} + 1$;
    end
end
计算统计量 z-score（见式 (3.10)）;
得到显著性检测结果 $\delta(X)$（见式 (3.12)）; // 基于 z-score 统计检验
return $\delta(X)$;

从整体流程来看，该算法本质上是在标准序列分析框架中引入一个上下文驱动的动态重建层。该重建层并不依赖生成模型内部信息，而是在检测阶段对词元分布进行结构化统计分析来实现水印信号的显式判别。与传统方法不同的是引入了两个关键改进，上下文哈希驱动的词表重建机制和全局统计显著性检验策略，这两个机制共同作用，使水印检测过程不再是静态分布检验，而是一个随上下文演化的序列统计过程。该提取算法在保证计算复杂度线性可控的同时，能够显著提高检测的可靠性与抗攻击能力，并为后续基于假设检验的理论分析提供统一基础。

3.5 实验结果分析

3.5.1 数据集与实验设置

本实验基于 MARKLLM 框架[86]的评测流程展开，将本章提出的多偏置调制文本水印方法作为新增算法模块接入统一评估体系中进行对比分析。数据集部分采用 C4 英文语料库[87]作为通用生成场景的基础文本来源，从中随机抽取长度适中的语句构

建验证样本集合，覆盖不同语言风格与语义结构的自然分布特征。对于每条输入文本，截取其前 30 个 token 作为上下文提示，在此基础上由大语言模型生成后续 200 个 token，模拟标准自回归文本生成过程。大语言模型选用 Meta 发布的 LLaMA3-8B 作为主要生成模型，并在部分对照实验中引入 LLaMA3-8B-Instruct 版本[88]，用于验证方法在不同模型指令对齐程度下的适用性与稳定性。该模型采用基于 Transformer 的自回归生成架构，词表规模约为 128k，在开放式文本生成任务中具有较好的语言建模能力与上下文表达能力。在生成阶段，不对模型参数进行任何修改，而是在 logits 层引入多偏置调制机制，对候选词分布进行动态扰动，实现水印信息的隐式嵌入。

所有实验在 MARKLLM 统一评测脚本下执行，随机种子固定为 42，以保证不同方法之间结果具有可复现性与可比性。水印相关超参数在所有实验中保持一致，其中绿色词表比例参数 α 用于控制词表划分中绿色集合的覆盖比例，多候选偏置数量 m 用于限定偏置候选空间的离散规模，基础偏置强度 γ_0 用于刻画整体概率提升幅度，随机扰动标准差 σ 用于引入时间步层面的偏置波动，检测阈值 τ 用于控制统计检验中的显著性判定标准。

评测指标遵循 MARKLLM 的标准评估协议，主要包括文本质量与水印检测两类指标。文本质量方面采用困惑度（Perplexity, PPL）衡量生成文本对大语言模型分布的贴合程度，数值越低代表语言自然性越强。水印检测方面采用 z-score 统计量衡量绿色词元出现频率相对于理论期望的偏移程度，数值越高表示水印信号越明显，检测置信度更强。为了测试方法在非理想环境下的稳定性，在 MARKLLM 的 robustness evaluation pipeline 中引入多种文本扰动操作，包括词元替换、随机删除、局部插入以及再生成扰动，用于模拟真实传播过程中可能出现的文本修改行为，评估水印在攻击条件下的保持能力。

3.5.2 实验结果与分析

本节基于 MARKLLM 统一评测框架，对本章提出的多偏置调制文本水印方法进行实验验证，并与 KGW、Unigram、SIR、EXP 等具有代表性的文本水印算法进行对比分析。所有方法均采用相同的数据划分方式、生成长度以及检测流程，在统一实验环境下完成评测，以减少模型设置与采样策略差异带来的影响。

在可检测性评估方面，本节主要从 z-score、TPR、F1 以及 AUC 四个角度对不同方法进行分析。其中，z-score 用于衡量绿色词命中数量相对于无水印随机分布理论

期望的标准化偏离程度，能够直接反映水印统计信号强弱。TPR 与 F1 用于评估不同误报率约束条件下检测器的识别能力与综合检测性能，AUC 则用于衡量检测器在不同判别阈值范围内的整体区分能力。实验结果如表 3.1 所示，展示了在 FPR $= 10\%$ 与 FPR $= 1\%$ 条件下不同水印方法的检测性能。

表 3.1　不同误报率约束条件下多种水印方法的可检测性对比

方法	z-**score**	**10%FPR-TPR**	**10%FPR-F1**	**1%FPR-TPR**	**1%FPR-F1**	**AUC**
KGW	6.82	1.000	0.952	1.000	0.995	0.999
Unigram	6.95	1.000	0.957	1.000	0.995	1.000
SWEET	6.74	1.000	0.952	1.000	0.995	0.999
UPV	–	–	–	–	–	0.996
EWD	6.88	1.000	0.952	1.000	0.995	0.999
SIR	–	0.995	0.950	0.990	0.990	0.995
X-SIR	–	0.995	0.950	0.940	0.964	0.988
EXP	–	1.000	0.952	1.000	0.995	1.000
EXP-Edit	–	1.000	0.952	0.995	0.990	0.994
Ours	6.84	1.000	0.954	1.000	0.996	0.999

可以看出，KGW、Unigram 与 EXP 等方法在标准检测场景下均能够取得较高的检测性能，其 TPR、F1 以及 AUC 指标均接近理想状态。这类方法大多采用基于概率偏置或采样扰动的水印嵌入机制，在无攻击条件下能够与自然文本形成较为明显的统计差异，较高的 z-score 结果同样说明其绿色词命中比例相对于理论随机分布存在显著偏移。而 SIR 与 X-SIR 在低误报率条件下出现一定程度的性能波动，在严格检测阈值条件下，基于语义约束或分布扰动的水印策略，其检测统计量在正负样本之间的可分性有所下降。从 AUC 指标来看，X-SIR 的整体检测能力同样略低于多数概率偏置类方法。EWD 在多个指标上均取得较优结果，说明基于熵加权的检测机制能够有效提升统计检测稳定性。

本章方法在不同误报率约束条件下均保持稳定检测性能，整体结果与主流概率偏置类方法处于相近水平。本章方法的 z-score 指标达到 6.84，能够维持较高的绿色词统计偏移强度，动态偏置策略并未削弱整体水印信号。本章方法的 AUC 指标达到 0.999，表明在不同检测阈值条件下仍能够保持较强的全局判别能力，进一步验证了所提出方法在可检测性方面的有效性与稳定性。

在鲁棒性评估方面，进一步分析不同文本扰动条件下各类水印方法的检测稳定

性。实验分别在无攻击、词元替换、随机删除、局部插入以及再生成扰动五种场景下进行。实验结果如表 3.2 所示，采用 z-score 与 AUC 作为鲁棒性评价指标。相比仅在固定阈值条件下统计 TPR 或 F1 指标，z-score 更适合描述水印统计量在扰动后的衰减趋势，AUC 则能够避免单一阈值带来的偶然性影响，更加适用于鲁棒性场景下的综合性能分析。

表 3.2　不同文本扰动场景下多种水印方法的鲁棒性对比

方法	**无攻击**		**词元替换**		**随机删除**		**局部插入**		**再生成扰动**	
	z-**score**	**AUC**	z-**score**	**AUC**	z-**score**	**AUC**	z-**score**	**AUC**	z-**score**	**AUC**
KGW	6.82	0.999	5.94	0.978	5.11	0.947	5.36	0.956	4.72	0.901
Unigram	6.95	1.000	6.37	0.989	5.92	0.972	5.86	0.968	5.21	0.934
SWEET	6.74	0.999	5.88	0.973	5.20	0.941	5.31	0.948	4.63	0.893
UPV	–	0.996	–	0.962	–	0.921	–	0.930	–	0.902
EWD	6.88	0.999	6.02	0.980	5.24	0.950	5.41	0.959	4.84	0.908
SIR	–	0.995	–	0.972	–	0.961	–	0.964	–	0.938
X-SIR	–	0.988	–	0.951	–	0.932	–	0.939	–	0.911
EXP	–	1.000	–	0.975	–	0.949	–	0.944	–	0.861
EXP-Edit	–	0.994	–	0.981	–	0.969	–	0.965	–	0.921
Ours	6.84	0.999	6.21	0.984	5.78	0.963	5.91	0.971	5.12	0.928

从整体结果可以看出，不同文本扰动都会对水印检测性能产生一定程度影响，其中再生成扰动对多数方法造成的性能下降最为明显。KGW、SWEET 以及 EXP 等基于概率偏置或采样扰动的方法在再生成扰动条件下均出现较明显的 AUC 衰减，说明攻击后原有水印统计结构被部分破坏，导致水印文本与自然文本之间的整体可分性明显减弱。Unigram、SIR 与 EXP-Edit 等方法在局部扰动场景下的 AUC 下降幅度相对较小，在再生成扰动条件下，其整体检测性能仍然出现不同程度下降，特别是 AUC 指标开始呈现较明显衰减趋势。

本章方法在不同扰动场景下均保持了较稳定的检测结果，尤其在随机删除、局部插入以及再生成扰动条件下，其 AUC 指标均明显高于多数基线方法。从整体趋势来看，多偏置调制机制使水印信号在序列内部呈现更加分散的统计结构，即使部分位置受到修改，整体绿色词命中趋势仍能够维持较高稳定性。由于不同时间步采用动态偏置选择策略，攻击过程难以针对单一固定模式进行集中破坏，因此在复杂扰

动条件下仍能够保持较高的整体可分性。动态偏置机制能够增强水印统计结构的抗扰动能力，使检测信号在生成路径发生变化后仍然保留较好的检测性能。

在文本质量评估方面，分析不同水印方法对生成文本自然性与下游任务性能的影响。实验结果如表 3.3 所示，评价指标主要包括困惑度（PPL）、语言多样性（Log Diversity）、下游任务性能以及基于大语言模型判别的主观质量评价结果。BLEU 与 Pass@1 分别用于评估机器翻译与代码生成任务中的语义保持能力，GPT-4 胜率刻画生成文本在人类偏好层面的整体可读性与自然程度。

表 3.3　不同水印方法对生成文本质量与下游任务性能的影响

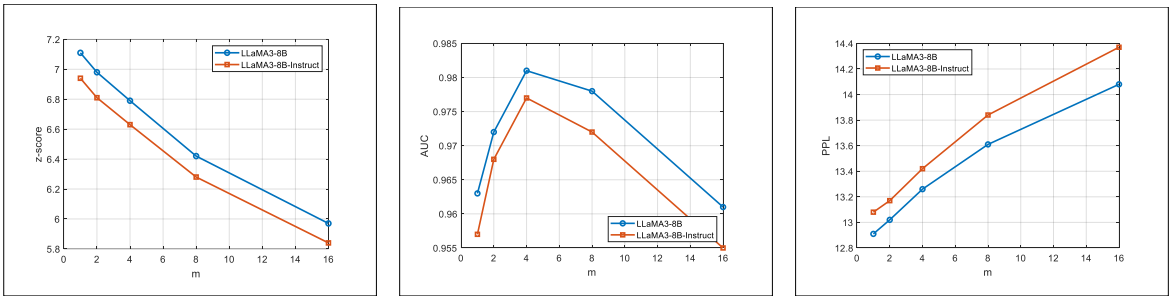
方法	**PPL**	**Log Diversity**	**下游任务性能**		**GPT-4 胜率**
			BLEU	**Pass@1**	
无水印	8.314	8.486	31.642	43	–
KGW	13.428	7.954	28.103	34	0.30
Unigram	13.615	7.846	27.214	32	0.32
SWEET	13.582	8.041	28.103	37	0.31
UPV	10.462	7.756	28.415	37	0.32
EWD	13.276	8.164	28.103	34	0.29
SIR	13.084	7.946	28.694	36	0.31
X-SIR	12.731	7.884	28.052	35	0.32
EXP	19.284	8.142	–	20	–
EXP-Edit	21.317	8.487	–	15	–
Ours	13.502	8.201	28.914	35	0.31

水印嵌入过程会对原始语言模型分布造成一定扰动，PPL 指标普遍高于无水印生成结果。EXP 与 EXP-Edit 方法的 PPL 指标明显高于其他基线方法，较强的采样调制与概率约束会对语言模型生成分布产生更明显干扰，这两类方法在代码生成任务中的 Pass@1 指标下降较为明显，强采样约束会影响模型对任务目标与语义结构的保持能力。KGW、SWEET、EWD 以及 SIR 等方法在文本质量方面保持了相对稳定的表现，PPL 变化幅度整体较为接近，基于概率偏置或语义约束的水印机制能够更好地兼顾文本自然性与水印可检测性。Unigram 方法的 Log Diversity 与 BLEU 指标下降较多，固定绿色词表划分机制限制了生成表达的丰富性，影响模型对原始语义结构的保持能力。

本章方法在 PPL 指标上与 KGW 类方法整体处于相近水平，可以看出，多偏置调制机制未显著增加语言模型的生成负担。在语言多样性指标上，本章方法相较部分基线方法表现出更高稳定性，说明动态偏置选择机制在一定程度上缓解了固定偏置造成的表达模式集中问题。同时，本章方法在 BLEU 与 Pass@1 指标上均保持了较高水平，整体性能变化相对平缓，在 GPT-4 主观评价结果中同样维持了与主流方法相近的胜率。这表明所提出方法在保证水印可检测性的同时，仍能够较好维持生成文本的可读性、流畅性以及整体自然程度。

为进一步分析多偏置调制机制中关键超参数对水印性能的影响，本节分别从候选偏置数量 m、随机扰动标准差 σ 以及基础偏置强度 γ_0 三个方面开展参数敏感性实验。所有实验均在与前文一致的实验设置下进行，模型采用 Meta 发布的 LLaMA3-8B 与 LLaMA3-8B-Instruct 两种模型，分析水印机制的泛化能力。检测阶段统一采用相同统计检验流程，除当前分析参数外，其余参数均保持默认设置不变。参数敏感性实验主要从水印检测能力和文本质量维度进行分析。

候选偏置数量 m 决定了系统在每个时间步可选择的偏置空间规模。当候选偏置数量 $m = 1$ 且随机扰动标准差 $\sigma = 0$ 时，本章方法退化为本质上与 KGW 类概率偏置水印方法保持一致的传统固定偏置调制形式。随着 m 增大，水印嵌入过程中的随机性与动态性逐渐增强。实验中分别设置 $m \in \{1, 2, 4, 8, 16\}$，结果如图 3.2。实验结果同时在 LLaMA3-8B 与 LLaMA3-8B-Instruct 两种模型上进行统计，以分析动态偏置机制在不同生成分布条件下的稳定性表现。



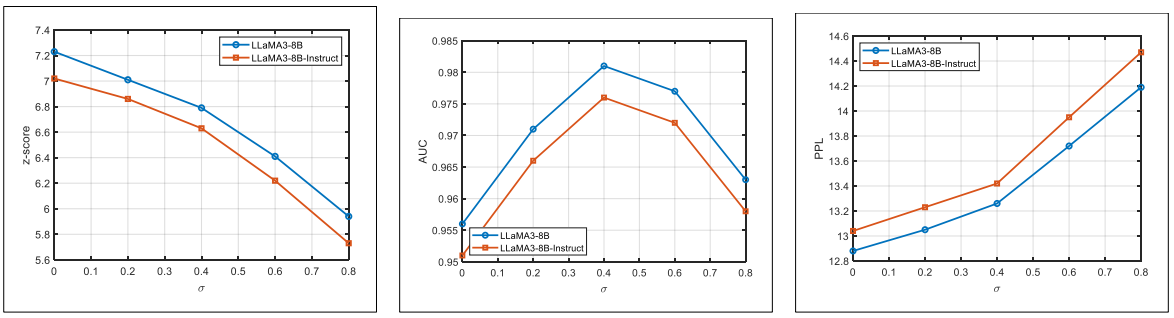
(a) z-score 影响　　(b) AUC 影响　　(c) PPL 影响

图 3.2　候选偏置数量 m 对多偏置水印方法性能的影响

两种模型整体均呈现相似变化趋势，随着候选偏置数量不断增加，z-score 整体呈现缓慢下降趋势。更大的候选偏置空间能够提升序列内部的随机性，却也会削弱绿色词统计偏移的一致性，导致检测信号出现一定程度衰减。相比基础模型，LLaMA3-

8B-Instruct 在各参数设置下的 z-score 略低，指令微调过程在一定程度上增强了生成分布的稳定性。当 $m = 4$ 时，两种模型均在检测性能与文本质量之间取得较为均衡的结果。此时 PPL 仅出现有限幅度增长，同时 AUC 保持较高水平。当 m 继续增大时，由于不同时间步之间的偏置波动进一步增强，整体统计结构开始趋于分散，导致检测性能逐渐下降。

随机扰动标准差 σ 用于控制候选偏置中的随机波动幅度，决定不同时间步之间局部调制强度的离散程度。实验中设置 $\sigma \in \{0, 0.2, 0.4, 0.6, 0.8\}$，对应结果如图 3.3 所示。当 $\sigma = 0$ 时，系统退化为无随机扰动的固定偏置结构，此时水印调制呈现高度一致的统计偏移特征，z-score 相对较高，但 AUC 偏低，说明检测边界较为集中但泛化区分能力不足。同时，PPL 处于最低水平，表明该设置对原始语言模型分布干扰最小。



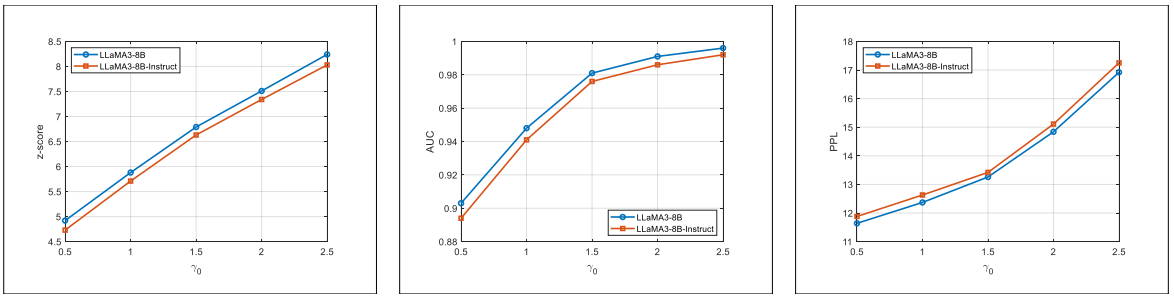
(a) z-score 影响　　(b) AUC 影响　　(c) PPL 影响

图 3.3　随机扰动标准差 σ 对多偏置水印方法性能的影响

随着 σ 从 0 增大至 0.4，候选偏置中的随机波动逐渐增强，水印信号在时间维度上呈现更分散的统计形态。在该区间内，AUC 出现明显提升并达到峰值，说明适度随机性能够改善水印与自然文本之间的整体可分性。z-score 略有下降，而 PPL 仅呈现轻微上升，表明生成质量仍保持在可接受范围内。该阶段体现出检测性能与生成质量之间的较优平衡。当 σ 进一步增大至 0.6 及以上时，随机扰动开始主导局部调制过程，偏置结构稳定性下降，绿色词统计偏移被部分噪声化削弱，导致 z-score 与 AUC 同步下降,PPL 持续上升，说明过强的随机扰动会逐步破坏原始语言模型的概率分布结构。$\sigma = 0.4$ 在检测可分性与文本质量之间取得较优折中，其既能够引入足够的随机性以增强水印鲁棒性，又不会显著损害语言模型生成分布的稳定性。

基础偏置强度 γ_0 用于控制绿色词在 logits 调制阶段的整体概率增益，其大小决定水印信号强度与文本生成质量之间的权衡关系。实验中设置 $\gamma_0 \in$

$\{0.5, 1.0, 1.5, 2.0, 2.5\}$，结果如图 3.4 所示。当 $\gamma_0 = 0.5$ 时，偏置强度较弱，水印信号偏移不明显，z-score 与 AUC 均处于较低水平，但 PPL 最小，说明对生成分布扰动较小，文本质量最高。随着 γ_0 增大至 1.5，水印信号逐渐增强，z-score 与 AUC 持续上升并达到较优水平，同时 PPL 仅小幅增长，检测性能与文本质量在该区间取得较好平衡。当 $\gamma_0 \geq 2.0$ 时，偏置作用进一步增强，检测指标继续提升但趋于饱和，而 PPL 明显上升，过强的偏置开始影响语言模型生成分布，降低文本自然性。综合来看，γ_0 控制了水印强度与文本质量之间的核心权衡。考虑检测性能与生成质量的折中，选择 $\gamma_0 = 1.5$ 作为默认设置。整体趋势表明，基础偏置增强使检测能力呈单调提升但边际收益递减，而文本质量损耗持续增加，体现出典型的性能权衡关系。



(a) z-score 影响　　(b) AUC 影响　　(c) PPL 影响

图 3.4　基础偏置强度 γ_0 对多偏置水印方法性能的影响

综合上述实验结果可以看出，多偏置调制文本水印方法在可检测性、鲁棒性以及文本质量之间取得了较为均衡的性能表现。在标准检测场景下，本章方法能够维持较高的 z-score 与 AUC 指标，整体检测能力与主流概率偏置方法保持相近水平。在多种文本扰动条件下，动态偏置机制能够有效缓解固定模式水印在局部修改后的统计退化问题，使水印信号在随机删除、局部插入以及再生成攻击下仍保持较好的稳定性与可分性。在文本自然性与下游任务性能方面未出现明显下降。参数敏感性实验表明，候选偏置数量、随机扰动强度以及基础偏置增益之间存在较明显的性能权衡关系，合理参数设置能够在检测性能与文本质量之间取得较优平衡。整体实验结果验证了基于动态偏置选择的文本水印机制在提升鲁棒性与保持生成质量方面的有效性与可行性。

3.6 本章小结

本章围绕大语言模型生成文本中的概率偏置类水印方法展开研究，针对传统固定偏置水印在鲁棒性、隐蔽性以及抗扰动能力方面存在的不足，提出了一种基于多偏置调制机制的文本水印方法。该方法在 KGW 类概率偏置框架基础上，引入上下文哈希驱动的动态偏置选择策略，构建多候选偏置集合后结合伪随机控制机制，在不同时间步动态选择调制强度，使水印嵌入过程由固定偏置调制扩展为随上下文变化的随机调制过程。与此同时，结合上下文相关的动态词表划分机制，使绿色词表与偏置选择在统一随机控制信号下协同变化，增强水印信号在序列中的分散性与不可预测性。

实验结果表明，本章方法在可检测性方面能够保持与主流概率偏置类方法相近的检测性能，在复杂攻击场景下表现出更稳定的检测结果，文本质量评估指标均保持了较为平稳的表现，生成文本仍能够保持较好的自然性、流畅性与语义一致性。此外，通过参数敏感性实验分析了候选偏置数量、随机扰动强度以及基础偏置增益对系统性能的影响。合理设置候选偏置规模与扰动强度后，系统能够在检测性能、鲁棒性以及文本质量之间取得较好的平衡。本章提出的多偏置调制文本水印方法在保持较高检测性能的同时，增强了水印结构的动态性与鲁棒性，为后续研究更加自适应、更强抗攻击能力的文本水印机制提供了重要基础。

第四章　基于结构信息的自适应模型文本水印

4.1　引言

生成文本水印通过在语言模型生成过程中引入隐式标记，实现对生成内容的识别与溯源，是当前大语言模型安全与可控性研究中的重要方向。第三章提出的多偏置随机选择机制，引入随机偏置嵌入策略，使水印信号在序列中呈现离散分布，在一定程度上缓解了固定偏置方法中规律性明显与鲁棒性不足的问题。该方法在不修改模型参数的前提下对 logits 分布进行轻量调制，在保证生成文本质量基本稳定的同时，实现了较好的检测性能，并在面对局部文本扰动时表现出一定的稳定性。该方法在不同生成位置上仍采用统一的偏置调控策略，未对上下文结构特征与预测分布的不确定性进行显式建模。在语法结构关键位置或高置信度区域施加相同强度的扰动，可能对文本流畅性产生负面影响，同时也未能充分利用生成过程中不同位置的不确定性差异，限制了水印性能的进一步提升空间。

针对上述问题，本章在已有方法基础上引入结构信息与生成不确定性建模，提出一种自适应文本水印方法。该方法通过分析生成过程中不同位置的上下文特征，构建结构敏感性度量与预测不确定性度量，并据此对偏置强度进行动态调控。在语法关键或高置信度区域降低干预，以减少对文本自然性的影响，在不确定性较高的区域适当增强嵌入，使水印信号更多分布于对生成结果影响较小的位置。同时，通过在时间维度上引入平滑约束，对相邻时间步的偏置变化进行限制，可以避免概率分布的剧烈波动，保证生成过程的稳定性。将原有统一的概率调控机制扩展为位置感知的自适应调控过程，使水印嵌入不仅依赖随机性，还能够结合上下文信息进行优化。该策略在保持水印可检测性的同时，有助于降低对生成文本质量的影响，并提升在文本编辑、重写及其他扰动场景下的鲁棒性与检测稳定性，为后续方法设计与实验分析提供基础。

4.2　总体框架

与第三章侧重于通过随机机制增强水印分布差异性不同，本章方法在大语言模型生成阶段进一步引入位置相关的调控信息，在不改变模型结构与参数的前提下，对

生成概率分布进行差异化控制，实现水印信息的自适应嵌入。如图 4.1，本章方法整体框架仍围绕自回归生成过程展开，但水印嵌入不再仅依赖统一的随机调制信号，而是结合上下文结构特征与生成状态信息进行动态调整。该方法主要增加了结构分析模块、不确定性估计模块、自适应偏置调控模块和平滑约束模块，实现水印信号的稳定注入。

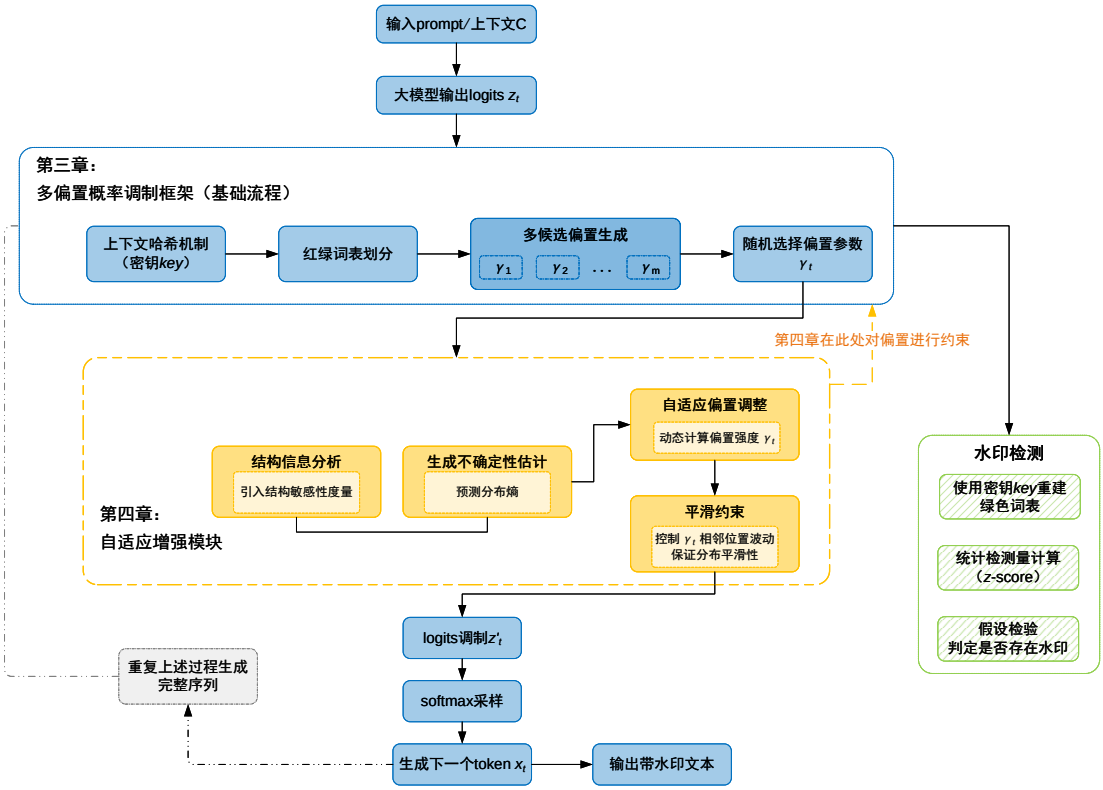


图 4.1　自适应增强模块与多偏置文本水印框架的关联

该框架在文本生成的每一个时间步引入位置感知的调控机制，并在随机控制的基础上叠加结构信息与不确定性信息，实现对生成分布的精细化调制。与第三章中基于随机偏置选择的动态调制方式相比，本章方法不再仅关注水印信号在序列中的分散性，而是进一步考虑不同生成位置在语法结构与语义表达中的作用差异，使水印嵌入过程具有针对性。通过这种方式，水印信号在序列中呈现出与上下文特征相关的非均匀分布，在降低对关键位置干扰的同时，提高在非关键区域的嵌入强度，在自然性与鲁棒性之间取得更优平衡。

本章方法的核心模块均作用于文本生成阶段，结构分析模块对当前位置进行结构敏感性评估，用于识别语法约束较强或语义承载较高的位置。不确定性估计模块基于预测概率分布计算熵值，对当前时间步的生成不确定性进行量化。在概率调制

过程中，自适应偏置调控模块根据结构敏感性与不确定性度量结果动态生成偏置强度，并通过平滑约束模块对相邻时间步的偏置变化进行限制，以避免分布突变带来的生成不稳定问题。最后，概率调制模块对绿色词表中的 logits 施加自适应偏置，引导词元采样过程完成水印信息的嵌入。

与第三章中通过伪随机机制选择离散偏置不同，本章方法在偏置层面引入连续调控策略，使水印强度能够随位置特征平滑变化。词表划分过程仍由上下文驱动的随机机制控制，以保持水印嵌入的安全性与不可预测性。水印检测整体流程与第三章保持一致，系统基于共享密钥重建上下文相关的词表划分规则，对输入文本进行逐位置分析。通过统计绿色词命中情况并构造标准化检验统计量，实现对水印存在性的判别。由于嵌入阶段引入了位置感知与平滑调控机制，水印信号在序列中的分布更加均衡，使检测统计量在面对文本扰动时具有更好的稳定性，提升整体检测可靠性。后续章节将对各模块的具体设计与实现方法进行详细展开。

4.3 水印嵌入

4.3.1 结构感知与上下文建模机制

在文本水印嵌入过程中，若对所有时间步采用一致的概率调制方式，虽然可以保证水印信号在整体上的稳定性，但由于缺乏对生成位置差异的刻画，容易在语法约束较强或语义承载较高的区域引入不必要的分布偏移，从而影响局部生成质量。这类位置通常对应更为集中的预测分布，过强干预会放大偏移效应，降低文本的自然性与连贯性。在保持原有随机调制机制与可检测性框架不变的前提下，本文引入结构感知与上下文建模方法，对当前时间步的上下文表示进行特征建模，并构建反映语法约束与语义重要性的敏感性度量，再对调制强度进行位置相关的自适应调整，使概率干预由单一随机驱动转变为融合上下文信息的细粒度控制，在不破坏检测一致性的前提下减少对文本质量的影响。

在原有上下文驱动机制基础上，本文不再依赖显式句法解析过程，而是直接基于语言模型在时间步 t 的上下文隐状态对结构信息进行隐式建模。具体而言，令 $C_t = x_{<t}$ 表示历史生成序列，语言模型在该上下文下输出隐表示 h_t^{LM}，该表示已编码局部语义依赖与句法结构信息。在此基础上，引入结构映射函数对隐状态进行投影，构

建结构特征表示，并定义结构敏感性度量为：

$$h_t = H_{key}(C_t), \quad s_t = F(C_t) = \sigma\big(W_s \cdot h_t^{LM}\big) \tag{4.1}$$

其中，H_{key} 为密钥控制的哈希函数，h_t 为随机控制种子；W_s 为映射参数；$\sigma(\cdot)$ 为归一化函数，用于将 s_t 约束在 $[0,1]$ 区间。该定义通过对隐状态进行低维投影，将复杂的结构信息压缩为单一标量，实现对当前位置结构约束强度的连续刻画。

上述结构敏感性度量由上下文隐表示唯一确定，因此在相同生成路径下具有确定性，同时能够反映不同位置在语义表达与句法组织中的差异。当语言模型对当前位置的预测具有较强结构约束时，隐状态通常呈现更稳定的表示模式，对应较高的 s_t；反之，在语义不确定或上下文约束较弱的区域，s_t 值相对较低。通过该方式，无需显式引入词性标注或依存分析过程，即可实现对结构信息的有效建模，保证方法在生成阶段的可实现性与计算效率。

基于上述定义，结构敏感性度量直接参与偏置调制过程，通过对原始随机偏置进行重参数化实现自适应控制。设第三章中生成的随机偏置为 γ_t，则结构约束后的有效偏置定义为：

$$\tilde{\gamma}_t = \gamma_t \cdot (1 - s_t) \tag{4.2}$$

该形式保证在结构敏感位置（s_t 较大）自动降低调制强度，在结构约束较弱位置（s_t 较小）保持原有偏置水平，从而实现基于上下文特征的差异化调控。该过程为逐时间步的确定性计算，仅依赖当前上下文隐表示，不引入额外训练目标或复杂推理过程。

在系统集成层面，结构建模模块以轻量映射形式嵌入生成流程，在每一时间步对语言模型隐状态进行一次线性变换与归一化处理，并将得到的结构敏感性度量作为控制变量传递至偏置调制模块。该机制不改变语言模型参数结构，仅对 logits 调制项进行重参数化处理，因此在计算复杂度上与原方法保持同一量级，同时避免引入额外的外部解析模块。

综上，本文通过对语言模型隐状态进行结构化映射，将隐含的语法与语义信息转化为连续敏感性度量，将其直接作用于偏置调制过程，实现对生成分布的结构约束调节。在保留随机调制机制与检测一致性的基础上，引入位置相关的控制变量，使水印嵌入由均匀干预转变为条件调制过程，不仅理论上降低了关键位置的分布扰动

风险，还为后续不确定性调控提供统一接口。

4.3.2 不确定性驱动的自适应偏置机制

在基于概率调制的文本水印方法中，偏置参数通常由随机机制或离散候选集合确定，其调制强度在不同时间步之间缺乏对生成状态的显式刻画。尽管第三章方法通过多候选偏置引入了时间维度上的随机性，但该调制过程仍未利用语言模型输出分布中所蕴含的置信信息，使得不同生成位置在调控策略上保持相似形式。当模型对当前词元预测具有较高确定性时，过强干预可能放大分布偏移；而在预测分布较为平坦的区域，若未充分利用其可调空间，则会降低水印嵌入效率。因此，有必要在原有随机调制框架上引入与生成分布相关的状态量，实现更具针对性的偏置控制。

在此基础上，本文引入基于输出概率分布的不确定性度量，对每一时间步的生成状态进行量化建模。设语言模型在时间步 t 输出归一化概率分布 $p_t(i)$，则定义不确定性度量为分布熵：

$$u_t = -\sum_i p_t(i) \log p_t(i) \tag{4.3}$$

该度量能够刻画当前分布的离散程度，当概率质量集中于少数候选词时，u_t 较小，表示模型预测具有较高置信度；当多个候选词概率接近时，u_t 增大，表明当前生成位置存在较高不确定性。为保证不同时间步之间的可比性，可对 u_t 进行归一化处理：

$$\hat{u}_t = \frac{u_t}{\log |\mathcal{V}|} \tag{4.4}$$

使其取值范围稳定在 $[0, 1]$ 区间，作为统一尺度的调控输入。

在获得不确定性度量后，将其与上一节定义的结构敏感性 s_t 共同作用于偏置调制过程。具体而言，不再直接使用第三章生成的候选偏置 γ_t^j，而是对其进行重参数化，构造自适应调制函数：

$$\tilde{\gamma}_t^j = \gamma_t^j \cdot g(\hat{u}_t, s_t) \tag{4.5}$$

其中 $g(\cdot)$ 为确定性调制函数，用于融合不确定性与结构信息。在实现上采用线性组合形式：

$$g(\hat{u}_t, s_t) = \alpha_u \hat{u}_t + \alpha_s (1 - s_t) \tag{4.6}$$

其中 $\alpha_u, \alpha_s \in [0,1]$ 为权重系数。该形式保证当不确定性较高时调制增强，而在结构敏感位置通过 $(1-s_t)$ 抑制偏置，形成双因素约束。

在完成偏置重参数化后，仍沿用第三章基于上下文哈希的索引选择机制确定最终调制参数：

$$j_t = h_t \bmod m, \quad \gamma_t = \tilde{\gamma}_t^{j_t} \tag{4.7}$$

该过程保证在相同上下文与密钥条件下偏置选择具有确定性，同时由于不确定性与结构特征均由当前上下文唯一确定，整个自适应调制过程在检测阶段具备可重建性且不破坏原有统计检验框架。

考虑到自适应调制可能在相邻时间步产生较大波动，进一步在时间维度引入一阶平滑约束，对偏置序列进行递推更新：

$$\gamma_t \leftarrow \lambda\gamma_t + (1-\lambda)\gamma_{t-1} \tag{4.8}$$

其中 $\lambda \in (0,1)$ 为平滑系数。该操作相当于对偏置序列施加低通滤波，限制调制强度的瞬时变化，降低对 logits 分布的突变扰动，保证生成过程的稳定性与连续性。

从实现角度看，该机制在 logits 调制前引入确定性标量计算，输入由当前时间步的概率分布与上下文解析结果直接给出，不涉及模型参数更新或额外训练过程，整体计算复杂度与原方法保持同一数量级，可无缝嵌入现有生成流程。在此基础上，通过分布熵度量与结构敏感性联合建模，对原有随机偏置进行重参数化，使调制强度随生成状态连续变化，在不改变第三章检测一致性的前提下，将水印嵌入由随机选择形式转化为状态感知调制过程，在文本自然性与鲁棒性之间获得更稳定的平衡。

4.3.3 词表构建与自适应概率调制

在完成结构敏感性建模与自适应偏置计算之后，水印嵌入过程需要将上述调控结果作用于语言模型输出分布，形成完整的生成闭环。该过程沿用第三章中的词表划分与概率调制框架，基于结构感知与不确定性引入的自适应偏置本质依然是一个偏置量。时间步 t 对应词表 $\mathcal{V}$ 仍由上下文哈希驱动划分为绿色词表 $\mathcal{G}_t$，该划分过程保持与嵌入阶段一致的确定性与随机性特征。对此绿色词表，模型输出 logits 在调制阶段引入前述计算得到的自适应偏置 γ_t，实现对目标词元的选择性增强。相较于第三章中固定或随机选取的偏置形式，此处的 γ_t 已包含结构约束与不确定性信息，其

数值由当前位置的生成状态唯一确定。

因此，概率调制过程本质上保持不变，但调制强度由静态或随机变量转化为状态相关变量，使水印信号在序列中呈现位置依赖特征。该改动不影响词表划分机制及检测阶段的统计一致性，仅在 logits 层引入额外的控制自由度。该过程可视为在原有概率调制框架中引入一层状态感知映射，将结构信息与分布特征通过偏置参数传递至生成分布，实现对水印嵌入位置与强度的协同控制。在不改变原有系统结构的前提下，该方法完成了从统一调制到自适应调制的过渡，使嵌入过程更加符合自然语言生成的局部特性。

4.3.4 嵌入算法流程

自适应调制文本水印的嵌入过程核心是在第三章随机调制机制基础上，引入结构敏感性与不确定性度量，对偏置参数进行重参数化，实现位置相关的动态概率调制。该过程不改变模型参数，仅在推理阶段对 logits 分布进行条件化调整，使水印嵌入策略在不同生成位置具备差异化特征。在时间步 t，系统首先基于历史生成序列 $x_{<t}$ 计算上下文哈希值 h_t，用于驱动词表划分与偏置索引选择。对当前上下文执行结构解析，得到结构敏感性度量 s_t，同时基于模型输出分布计算不确定性度量 u_t。对第三章生成的候选偏置集合进行自适应重参数化，结合哈希索引确定当前时间步的有效偏置参数 γ_t，再引入递推形式的平滑约束对偏置进行更新。词表划分与概率调制过程基本保持不变。在获得调制后的 logits 后，经 softmax 归一化得到采样分布，并据此生成当前词元。上述过程逐步迭代直至序列结束，形成完整的水印文本。该流程在结构上与第三章保持一致，仅在偏置计算阶段引入额外控制变量，实现对嵌入强度的细粒度调节。为便于描述，将嵌入过程形式化为算法 4.1。

算法 4.1: 自适应调制文本水印嵌入算法

输入: 大语言模型 LLM，词表 $\mathcal{V}$，密钥 key，嵌入比例 α，生成长度 T

输出: 水印文本序列 X

初始化：$X \leftarrow \emptyset$;

for $t = 1$ **to** T **do**

  计算上下文哈希：$h_t = H_{key}(X_{<t})$;

  结构信息建模：$s_t = F(X_{<t})$;

  计算 logits：$z_t = LLM(X_{<t}), \quad P_t = \text{softmax}(z_t)$;

  计算不确定性：$u_t = -\sum_i P_t(i) \log P_t(i)$;

  生成候选偏置集合：$\Gamma_t = \{\gamma_t^1, \ldots, \gamma_t^m\}$; // 沿用第三章

  自适应偏置调制：$\tilde{\gamma}_t^j = \gamma_t^j \cdot g(u_t, s_t)$;

  选择偏置参数：$\gamma_t = \tilde{\gamma}_t^{\,h_t \bmod m}$;

  平滑约束：$\gamma_t \leftarrow \lambda\gamma_t + (1-\lambda)\gamma_{t-1}$;

  构建绿色词集合：$\mathcal{G}_t = \text{TokenSelect}(\mathcal{V}, h_t, \alpha)$;

  偏置调制：$z_t'(i) = z_t(i) + \gamma_t \cdot \mathbb{I}(v_i \in \mathcal{G}_t)$;

  概率归一化与采样：$P_t' = \text{softmax}(z_t'),\ x_t \sim P_t'$;

  更新序列：$X \leftarrow X \oplus x_t$;

end

return X;

该算法在第三章多偏置随机调制框架上增加状态感知调控层，将结构敏感性与分布不确定性引入偏置计算过程，使调制强度由随机变量扩展为上下文相关变量。与仅依赖随机机制的调制方式相比，该过程在保持检测一致性的前提下，对不同生成位置施加差异化干预，减少结构敏感区域的分布扰动，同时在高不确定性区域增强水印嵌入，从而提升水印信号的分布合理性与抗扰动能力，并为后续统计检测提供更稳定的统计基础。

4.4 水印提取

4.4.1 统计检验与提取实现

在嵌入阶段完成结构敏感性与不确定性驱动的概率调制之后，水印提取的核心任务是在不访问语言模型内部状态与嵌入阶段中间变量的条件下，仅依赖待检测文本恢复隐藏的水印信号。该过程仍建立在第三章提出的统计检验框架之上，检测端不引入新的推理结构或额外特征建模模块，整体算法形式保持一致。从实现角度来看，提取过程可以统一描述为基于上下文重建的词表恢复与逐位置统计累积，其目标是在离散文本序列中还原由概率调制引入的微弱分布偏移。系统通过密钥驱动的确定性哈希机制生成控制索引，该索引不参与概率计算，仅用于恢复当前时间步的词表划分状态，在检测端重建与嵌入阶段一致的随机结构。

在具体实现过程中，系统首先以待检测文本为输入，按照自回归顺序逐步构造历史上下文表示，并在每一个时间步基于密钥生成对应的哈希索引。该索引用于确定当前时间步的词表划分方式，恢复绿色词集合的生成状态，但该过程不依赖任何结构敏感性或不确定性变量，因此与嵌入阶段的复杂调制机制完全解耦。检测端能够在不访问生成路径或模型输出分布的情况下，仅依靠上下文信息复现词表划分逻辑，使绿色词集合在不同实现环境下保持严格一致性与可复现性。

在完成词表重建后，系统对生成序列中的词元进行逐位置匹配判断，并根据其是否属于绿色词集合构建二值指示序列，将自然语言文本转化为可统计分析的离散表示。随后，对全序列中的绿色词命中结果进行累积统计，并通过标准化处理将其映射至统一分布空间，最终结合预设阈值完成水印判定。该过程本质上仍属于经典假设检验框架，其中原假设对应无水印文本分布，备择假设对应经过概率调制后的水印文本分布。由于检测阶段未引入额外的结构恢复或概率修正机制，因此整体检测流程与第三章保持一致，仅嵌入阶段动态偏置调制所带来的分布偏移特性会影响最终统计结果。

尽管嵌入阶段引入了结构敏感性与动态不确定性调制机制，但检测阶段并不需要对相关控制变量进行显式恢复或额外建模。水印提取过程整体仍与第三章保持一致，其核心依然基于上下文驱动的词表重建机制与逐位置统计判别流程。由于词表划分完全由密钥与上下文哈希共同决定，因此检测端无需依赖连续控制变量或外部结构信息，仅通过对词表划分函数的重复调用以及序列遍历与集合匹配即可完成命

中统计构建。该解耦设计保证了提取算法在实现层面的稳定性，使其在不改变整体计算结构与复杂度的前提下，仅通过输入文本统计分布特性的变化体现嵌入策略带来的差异。

系统首先对输入文本进行逐时间步扫描，并基于密钥生成对应的上下文哈希索引。随后利用该索引恢复当前时间步的词表划分状态，并构建绿色词表，实现对嵌入阶段随机结构的确定性复现。该过程完全由上下文与密钥驱动，不依赖任何连续控制变量或嵌入阶段中间表示，使检测端在不同运行环境下均可保持一致的划分结果。在完成词表恢复之后，系统对当前词元执行集合归属判断，并将结果转换为二值指示变量，将自然语言序列映射为统计分析序列。随后对全序列中的命中结果进行累积求和，并构造标准化统计量用于最终判别。该统计量仍采用第三章一致的归一化形式，依赖命中计数与序列长度，不引入额外参数估计或模型修正过程，检测流程在计算复杂度上保持稳定不变。本章方法检测端未发生结构性变化，完整提取算法在形式上与第三章一致，不再单独重复给出伪代码表达。

4.4.2 统计行为分析

在第三章检测框架中，绿色词命中过程通常可近似建模为独立伯努利随机变量序列，其统计性质由词表比例参数控制。在该假设下，命中统计量具有明确的期望与方差结构，因此可以直接构造标准化检验统计量用于判别。然而在本章方法中，虽然检测端算法结构保持不变，但由于嵌入阶段引入结构感知与不确定性驱动的调制机制，使得观测序列在统计分布层面呈现非均匀扰动特征，以此改变统计量的局部分布形态。

具体而言，在高不确定性区域，模型预测分布较为分散，使嵌入阶段的偏置增强作用更为明显，从而提升该区域的绿色词命中概率，形成局部正向偏移。而在结构敏感区域，由于调制强度受到抑制，命中概率相对降低，使统计贡献在这些位置被弱化。两种机制共同作用，使整体统计偏移不再集中于特定区域，而是沿序列方向呈现分散式分布结构，降低局部异常对整体统计量的影响。与此同时，嵌入阶段引入的时间平滑约束在检测端表现为弱相关结构，使得相邻时间步之间的命中事件不再严格独立，而是呈现随时间间隔衰减的相关性关系。这种相关性不会改变统计量的渐近增长阶，但会显著降低有限长度序列中的方差波动，使统计量的收敛过程更加平滑稳定。这一特性在短文本检测或局部扰动场景中表现尤为明显，提升整体

判别稳定性。

从鲁棒性角度来看，自适应嵌入机制倾向于将水印信号分布于语义冗余或预测不确定性较高的位置，这些位置在文本编辑或重写过程中被修改的概率较低，因此水印信息具有更高的保留率。相比之下，语法结构约束较强的位置由于嵌入强度被抑制，即使发生局部扰动，其对整体统计量的影响也较为有限。这种空间分布特性使检测过程在面对文本变换时仍具有较强稳定性。

4.5 实验结果分析

4.5.1 数据集与实验设置

第四章实验整体沿用第三章的实验环境与评测流程，数据集仍采用 C4 英文语料库，生成模型选用 LLaMA3-8B，在相同随机种子与采样参数条件下完成实验，以保证不同方法之间结果具有可比性。评测指标继续采用文本质量、检测性能与鲁棒性三类指标。文本质量采用困惑度衡量生成文本自然性，检测性能采用 z-score 衡量水印检测显著性与检测准确率，鲁棒性实验测试不同方法在攻击条件下的水印保持能力。此外，为进一步验证各组成模块对整体性能的贡献，增加消融实验与参数敏感性实验，对结构感知机制、不确定性调制机制以及平滑约束机制进行独立分析，以评估不同模块对文本质量与检测性能的影响。

4.5.2 实验结果与分析

表 4.1 展示了不同方法在文本质量与检测性能上的实验结果，采用困惑度（PPL）与检测统计量（z-score）作为评估指标。KGW 方法通过固定偏置对绿色词概率进行增强，能够形成较明显的统计偏移，其 z-score 达到 6.82，但由于固定强度调制会持续对生成分布产生干扰，因此其 PPL 达到 13.43，说明该方法在文本自然性方面存在一定影响。第三章提出的多偏置随机调制方法通过动态切换不同偏置参数，使水印信号在时间维度上呈现更加离散的分布特征，PPL 稍稍增至 13.50，取得 6.84 的 z-score，随机偏置机制能够在保证检测能力的同时增强统计显著性。

第四章提出的自适应调制方法取得最低的 PPL，为 12.79，对语言模型原始生成分布的干扰最小。同时，其 z-score 达到 6.88，高于 KGW 与第三章方法，不仅改善了文本质量，还增强了水印统计偏移的稳定性与检测显著性。整体来看，第四章方

法在文本自然性与检测性能之间取得了更优平衡。相比 KGW 方法，自适应调制机制能够有效降低固定概率偏置带来的生成扰动。相比第三章随机调制方法，结构感知与不确定性建模提升了水印嵌入与生成状态之间的匹配程度，使水印信号分布更加稳定合理。

表 4.1　不同方法文本质量与检测性能对比

方法	PPL	z-score
KGW	13.43	6.82
第三章方法	13.50	6.84
第四章方法	12.79	6.88

为测试本章方法在文本扰动条件下的稳定性，依旧在词元替换、随机删除、局部插入以及语义保持改写等攻击条件下进行鲁棒性实验，结果如表 4.2 所示。由于鲁棒性实验主要关注文本扰动条件下水印信号的保持能力，而攻击过程本身会对文本语言分布产生额外影响，所以本节主要采用 z-score 对不同方法的检测稳定性进行分析。z-score 越高表示水印统计偏移越明显，对应更强的检测置信度与鲁棒性。

表 4.2　不同攻击条件下的各方法检测性能对比

方法	无攻击	词元替换	随机删除	局部插入	改写攻击
KGW	6.82	5.94	5.11	5.36	4.72
第三章方法	6.84	6.21	5.78	5.91	5.12
第四章方法	6.88	6.53	6.29	6.45	5.96

在词元替换攻击条件下，KGW 的 z-score 下降至 5.94，第三章方法与第四章方法分别达到 6.21 与 6.53。相比第三章方法，第四章方法提升约 5.2%，说明结构感知与不确定性调制机制能够降低词汇替换对局部统计分布的破坏，使水印信号在词元级修改条件下仍具有较好的保持能力。在随机删除攻击条件下，KGW 的 z-score 下降至 5.11，第三章方法达到 5.78，而第四章方法达到 6.29。相比第三章方法提升约 8.8%，相比 KGW 提升约 23.1%。这表明自适应调制机制在文本删减条件下仍能够维持较稳定的统计偏移。在局部插入攻击条件下，KGW、第三章方法与第四章方法的 z-score 分别为 5.36、5.91 与 6.45。第四章方法相比第三章提升约 9.1%，相比 KGW 提升约 20.3%。该策略能够有效缓解局部插入带来的上下文扰动，使检测统计量保持更好的稳定性。在改写攻击条件下，不同方法之间差异最为明显。KGW 的 z-score 下

降至 4.72，而第三章方法与第四章方法分别达到 5.12 与 5.96。相比第三章方法，第四章方法提升约 16.4%，相比 KGW 提升约 26.3%。说明第四章方法能够在语义保持改写场景下维持更加稳定的统计特征，体现出更强的抗重写能力与鲁棒性。

整体来看，第三章方法通过随机化偏置增强了水印信号在序列中的分散性，相比 KGW 在各类攻击条件下均表现出更好的鲁棒性。而第四章方法进一步结合结构感知、不确定性建模与时间平滑约束，使水印信号能够更加稳定地嵌入文本生成过程，在不同文本扰动场景下均取得最高检测统计量，表明该方法在复杂攻击环境下具有更强的鲁棒性与检测稳定性。

为验证各组成模块对整体性能的贡献，开展了消融实验，结果如表 4.3 所示。移除结构感知调制模块后，PPL 由 12.79 上升至 13.18，z-score 由 6.88 下降至 6.42。可以看出结构感知机制能够有效降低生成扰动，并提升水印统计稳定性。去除不确定性感知模块后，PPL 上升至 13.05，z-score 下降至 6.17。相比完整方法，检测统计量下降约 10.3%，下降幅度最大，表明不确定性调制机制对增强统计偏移与提升检测性能具有重要作用。去除平滑调制策略后，PPL 略有上升，z-score 下降至 6.28，分别变化约 1.3% 与 8.7%。时间平滑约束能够降低相邻时间步之间的概率波动，提升生成稳定性与检测一致性。三种模块移除后均会导致 PPL 上升与 z-score 下降。其中，不确定性感知模块对检测性能影响最明显，结构感知模块对文本质量改善最明显，而平滑调制策略能够进一步增强生成稳定性。完整方法在取得最低 PPL 的同时获得最高 z-score，各模块协同能够在文本自然性与水印检测性能之间取得更优平衡。

表 4.3　不同模块对文本质量与水印检测性能影响的消融实验结果

方法设置	困惑度（PPL）	检测统计量（z-score）
去除结构感知调制模块	13.18	6.42
去除不确定性感知模块	13.05	6.17
去除平滑调制策略	12.96	6.28
第四章完整方法	12.79	6.88

本节分别从文本质量、鲁棒性及模块贡献三个方面对方法进行了实验验证。第四章方法在 PPL 指标上优于 KGW 及第三章方法，在保证文本自然性的同时降低了生成分布扰动，在多种文本攻击条件下取得更高的 z-score，具有更好的稳定性与鲁棒性。消融实验进表明不确定性调制模块对检测性能提升最为关键，结构感知模块主要改善文本质量，平滑约束模块提升生成稳定性，三者协同实现更优整体性能。

4.6 本章小结

本章针对第三章多偏置随机调制方法在不同生成位置采用统一偏置策略、缺乏上下文结构感知的问题，提出了一种基于结构信息的自适应模型文本水印方法。在不改变语言模型结构与参数的前提下，引入结构敏感性度量与生成不确定性度量，对水印偏置强度进行动态调节，并结合时间平滑约束机制，实现对生成概率分布的自适应调制。该方法能够根据不同位置的上下文特征差异调整嵌入强度，在降低关键位置生成扰动的同时增强水印嵌入稳定性。实验结果表明，所提方法在文本质量、检测性能与鲁棒性方面均取得较优表现。相比 KGW 方法与第三章方法，本章方法在保证检测能力的同时降低了文本生成扰动，在改写攻击条件下表现出更高的检测稳定性。消融实验进一步验证了结构感知机制、不确定性调制机制与平滑约束策略对整体性能提升的有效性，说明各模块协同能够在文本自然性与水印鲁棒性之间取得更优平衡。

第五章　总结与展望

5.1　工作总结

随着生成式大语言模型在文本生成任务中的广泛应用，模型输出内容的来源标识与可追溯性问题逐渐成为研究重点。一种在生成过程中嵌入隐式标记的文本水印技术手段，能够在不影响文本可读性的前提下实现内容识别，因此在模型安全与内容监管中具有重要意义。自然语言本身具有高度的表达冗余性与重写灵活性，同一语义往往可以通过多种不同的词句形式进行表达，使得嵌入在文本中的水印信号容易在后续编辑、改写或重采样过程中发生削弱甚至丢失，影响检测结果的稳定性。保证文本生成质量的同时提升水印在复杂扰动条件下的鲁棒性，成为当前文本水印研究中的核心挑战之一。本文从大语言模型的概率分布调控机制出发，对文本水印嵌入过程进行了系统化分析，并重点研究了不同调制策略在文本质量与检测性能之间的影响关系。构建了从随机调制到自适应调制的改进路径，使水印嵌入方式从单一控制逐步演化为与生成状态相关的动态调节机制。

在基础方法层面，针对传统单一偏置或固定调制策略在面对文本局部扰动时易出现水印不稳定的问题，本文引入多偏置随机选择机制。该方法设置多个候选偏置参数，在每一生成时间步结合上下文信息对其进行动态选择，使水印信号在序列中呈现出更为分散的分布特征。相较于集中式嵌入方式，该策略降低了水印对局部位置的依赖程度，使得当文本发生局部替换或删改时，整体统计结构仍能够保持稳定性。实验结果表明，该方法在不显著影响生成文本流畅性的前提下，提高了水印检测的整体鲁棒性，但在结构敏感区域仍存在一定的干扰风险。

在进一步研究中，本文考虑到文本生成过程中不同位置在语义重要性与预测不确定性方面存在差异，因此在嵌入机制中引入结构信息与不确定性建模思想。通过对生成上下文进行分析，对不同位置的敏感程度进行区分，并据此对偏置强度进行动态调整，使水印嵌入不再采用统一扰动方式，而是形成与生成状态相关的差异化控制。同时，在时间维度上引入平滑约束，对相邻时间步的调制变化进行限制，避免概率分布出现突变现象，提升生成过程的连续性与稳定性。实验结果表明，该方法在多种文本扰动场景下均表现出更优的稳定性，在保持文本自然性的同时增强了水

印信号的可检测性。

总体而言，本文从概率调控的角度对文本水印嵌入机制进行了系统改进，使水印嵌入过程由静态随机调制逐步过渡为基于上下文状态的自适应调制。在保证生成质量基本不受影响的前提下，提升了水印在复杂扰动环境中的稳定性与鲁棒性，在一定程度上缓解了文本可读性与水印可检测性之间的矛盾关系，为后续相关研究提供了新的思路与方法基础。

5.2 工作展望

尽管本文围绕基于概率调控的文本水印方法进行了系统性研究，并在结构自适应与不确定性建模方面取得了改进，但在更复杂的应用环境与攻击条件下，该方法仍存在优化空间，未来研究可以从多个方向继续拓展。

首先，嵌入精度与文本自然性可以得到更精准地优化。当前方法主要通过结构信息与概率分布特征控制调制强度，但这种控制仍然停留在较为宏观的层面，在更细粒度的语义表达中，仍可能存在轻微但累积性的分布偏移。未来可以考虑引入更细粒度的语义约束机制，使水印嵌入不仅依赖句法或概率特征，还能够更精确地对语义一致性进行建模，降低对文本表达自然性的潜在影响。

其次，针对复杂文本变换场景的鲁棒性仍有提升空间。在实际应用中，文本可能经历多轮改写、压缩、扩展甚至跨模型重生成，这类操作会显著改变词序结构与局部统计特征，对水印信号的稳定性提出更高要求。因此，未来可以从跨变换一致性建模角度出发，研究水印在多种编辑操作下的保持机制，使检测过程具备更强的跨分布泛化能力。

再次，特定需求下要考量水印容量与表达效率之间的权衡。当前方法主要关注水印是否可检测以及检测稳定性，但在信息表达效率方面具有提升空间。在受限词表与自然语言约束条件下，如何在有限扰动范围内承载更多可验证信息，同时避免对生成分布造成明显影响，是一个值得深入研究的问题。

最后，随着文本生成模型逐步进入真实应用环境，水印系统的安全性与对抗能力也需要增强。未来可以考虑从攻击建模角度出发，对潜在的规避策略、重写攻击以及逆向检测攻击进行系统分析，并在此基础上构建更加稳健的嵌入与检测框架，使水印机制在开放环境中仍然具备可靠性与长期有效性。

参考文献

[1] 吴飞, 阳春华, 兰旭光, 等. 人工智能的回顾与展望[J]. 中国科学基金, 2018, 32(3): 243-250.

[2] JIANG Y, LI X, LUO H, et al. Quo vadis artificial intelligence?[J]. Discover Artificial Intelligence, 2022, 2(1): 1-19.

[3] ERTEL W. Introduction to artificial intelligence[M]. Springer Nature, 2024.

[4] BANH L, STROBEL G. Generative Artificial Intelligence[J]. Electronic Markets, 2023, 33(1): 63.

[5] BHARADIYA J. A comprehensive survey of deep learning techniques natural language processing[J]. European Journal of Technology, 2023, 7(1): 58-66.

[6] LAURIOLA I, LAVELLI A, AIOLLI F. An introduction to deep learning in natural language processing: Models, techniques, and tools[J]. Neurocomputing, 2022, 470: 443-456.

[7] NADKARNI P M, OHNO-MACHADO L, CHAPMAN W W. Natural Language Processing: An Introduction[J]. Journal of the American Medical Informatics Association, 2011, 18(5): 544-551.

[8] 张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, 26(1): 32-42.

[9] PAULS A, KLEIN D. Faster and smaller n-gram language models[C]//Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies. 2011: 258-267.

[10] AL-SELWI S M, HASSAN M F, ABDULKADIR S J, et al. RNN-LSTM: From applications to modeling techniques and beyond—Systematic review[J]. Journal of King Saud University - Computer and Information Sciences, 2024, 36(5): 102068.

[11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA,

USA. 2017: 5998-6008.

[12] LIU L, XU X. Self-attention mechanism at the token level: gradient analysis and algorithm optimization[J]. Knowledge-Based Systems, 2023, 277: 110784.

[13] LI D, JIANG B, HUANG L, et al. From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge[C]//Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. Suzhou, China: Association for Computational Linguistics, 2025: 2757-2791.

[14] 车万翔, 窦志成, 冯岩松, 等. 大模型时代的自然语言处理: 挑战, 机遇与发展[J]. 中国科学: 信息科学, 2023, 53(9): 1645-1687.

[15] NAVEED H, KHAN A U, QIU S, et al. A comprehensive overview of large language models[J]. ACM Transactions on Intelligent Systems and Technology, 2025, 16(5): 1-72.

[16] ZHANG S, DONG L, LI X, et al. Instruction tuning for large language models: A survey[J]. ACM Computing Surveys, 2026, 58(7): 1-36.

[17] CHAUDHARI S, AGGARWAL P, MURAHARI V, et al. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms[J]. ACM Computing Surveys, 2025, 58(2): 1-37.

[18] WU H, WU D. 生成式人工智能教育应用: 发展历史, 国际态势与未来展望[J]. International & Comparative Education, 2024, 46(6): 13.

[19] SCHLAGWEIN D, WILLCOCKS L. ‘ChatGPT et al.’ : The Ethics of Using (Generative) Artificial Intelligence in Research and Science[J]. Journal of Information Technology, 2023, 38(3): 232-238.

[20] PAVLIK J V. Collaborating with ChatGPT: Considering the Implications of Generative Artificial Intelligence for Journalism and Media Education[J]. Journalism & Mass Communication Educator, 2023, 78(1): 84-93.

[21] NGUYEN-DUC A, CABRERO-DANIEL B, PRZYBYLEK A, et al. Generative Artificial Intelligence for Software Engineering—A Research Agenda[J]. Software: Practice and Experience, 2025, 55(11): 1806-1843.

[22] XIANG L, LI N, LIU Y, et al. AI-Generated Text Detection: A Comprehensive Review of Active and Passive Approaches[J]. Computers, Materials & Continua, 2026, 86(3).

[23] 吴汉舟, 张杰, 李越, 等. 人工智能模型水印研究进展[J]. 中国图象图形学报, 2023, 28(6): 1792-1810.

[24] 张新鹏, 吴汉舟. 深度模型水印[J]. 自然杂志, 2022, 44(4): 267-273.

[25] 冯乐, 朱仁杰, 吴汉舟, 等. 神经网络水印综述[J]. 应用科学学报, 2021, 39(6): 881-892.

[26] MEGÍAS D, KURIBAYASHI M, ROSALES A, et al. DISSIMILAR: Towards Fake News Detection Using Information Hiding, Signal Processing and Machine Learning [C]//Proceedings of the 16th International Conference on Availability, Reliability and Security. 2021: 1-9.

[27] ANTOUN W, SAGOT B, SEDDAH D. From Text to Source: Detecting Large Language Model Generated Content[C]//Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Turin, Italy: ELRA, 2024: 7531-7543.

[28] CROTHERS E N, JAPKOWICZ N, VIKTOR H L. Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods[J]. IEEE Access, 2023, 11: 70977-71002.

[29] CHEN Y, KANG H, ZHAI V, et al. Token prediction as implicit classification to identify LLM-generated text[C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023: 13112-13120.

[30] BORILE C, ABRATE C. How to Generalize the Detection of AI-Generated Text: Confounding Neurons[C]//Findings of the Association for Computational Linguistics: EMNLP 2025. 2025: 25461-25476.

[31] WANG C, GAO M, WANG Z, et al. Prompt-Induced Linguistic Fingerprints for LLM-Generated Fake News Detection[C]//Proceedings of the ACM Web Conference 2026. 2026: 7633-7644.

[32] KUMARAGE T, LIU H. Neural Authorship Attribution: Stylometric Analysis on Large Language Models[C]//2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2023: 51-54.

[33] VENKATRAMAN S, UCHENDU A, LEE D, et al. GPT-who: An information density-based machine-generated text detector[C]//Findings of the Association for Computational Linguistics: NAACL 2024. 2024: 103-115.

[34] SHI Y, SHENG Q, CAO J, et al. Ten words only still help: Improving black-box AI-generated text detection via proxy-guided efficient re-sampling[C]//Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. 2024: 494-502.

[35] FRÖHLING L, ZUBIAGA A. Feature-based Detection of Automated Language Models: Tackling GPT-2, GPT-3 and Grover[J]. PeerJ Computer Science, 2021, 7: e443.

[36] GALLÉ M, ROZEN J, KRUSZEWSKI G, et al. Unsupervised and distributional detection of machine-generated text[J]. arXiv preprint arXiv:2111.02878, 2021.

[37] KIM Z M, LEE K H, ZHU P, et al. Threads of subtlety: detecting machine-generated texts through discourse motifs[C]//Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024: 5449-5474.

[38] ETHAYARAJH K, XU W, MUENNIGHOFF N, et al. Model Alignment as Prospect Theoretic Optimization[C]//Proceedings of the 41st International Conference on Machine Learning: vol. 235. JMLR.org, 2024: 12634-12651.

[39] ALHIJAWI B, JARRAR R, ABUALRUB A, et al. Deep learning detection method for large language models-generated scientific content[J]. Neural Computing and Applications, 2025, 37(1): 91-104.

[40] GEHRMANN S, STROBELT H, RUSH A M. GLTR: statistical detection and visualization of generated text[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2019: 111-116.

[41] MITCHELL E, LEE Y, KHAZATSKY A, et al. DetectGPT: zero-shot machine-generated text detection using probability curvature[C]//Proceedings of the 40th International Conference on Machine Learning. 2023: 24950-24962.

[42] HANS A, SCHWARZSCHILD A, CHEREPANOVA V, et al. Spotting LLMs with binoculars: zero-shot detection of machine-generated text[C]//Proceedings of the 41st International Conference on Machine Learning. 2024: 17519-17537.

[43] POPESCU-APREUTESEI L E, IOSUPESCU M S, NECULA S C, et al. Upholding academic integrity amidst advanced language models: evaluating BiLSTM networks with GloVe embeddings for detecting AI-generated scientific abstracts[J]. Computers, Materials & Continua, 2025, 84(2): 2605-2644.

[44] OGHAZ M M, SAHEER L B, DHAME K, et al. Detection and classification of ChatGPT-generated content using deep transformer models[J]. Frontiers in Artificial Intelligence, 2025, 8: 1458707.

[45] GUO B, ZHANG X, WANG Z, et al. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection[J]. arXiv preprint arXiv:2301.07597, 2023.

[46] WANG Y, GUO X, LIU Z, et al. Detection and comparative study of differences between AI-generated and scholar-written Chinese abstracts[J]. Journal of Intelligence, 2023, 42(9): 127-134.

[47] LIU X, ZHANG Z, WANG Y, et al. Coco: coherence-enhanced machine-generated text detection under data limitation with contrastive learning[C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023: 16167-16188.

[48] MIAO Y, GAO H, ZHANG H, et al. Efficient detection of LLM-generated texts with a Bayesian surrogate model[C]//Findings of the Association for Computational Linguistics. 2024: 6118-6130.

[49] BAO G, ZHAO Y, TENG Z, et al. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature[C]//International Conference on Learning Representations. 2024: 24814-24836.

[50] LIU S, LIU X, WANG Y, et al. Does DetectGPT fully utilize perturbation? Bridging selective perturbation to fine-tuned contrastive learning detector would be better [C]//Proceedings of the 62nd Annual Meeting of the Association for Computational

Linguistics. 2024: 1874-1889.

[51] SU J, ZHUO T Y, WANG D, et al. DetectLLM: leveraging log-rank information for zero-shot detection of machine-generated text[C]//Findings of the Association for Computational Linguistics: EMNLP. 2023: 12395-12412.

[52] YANG X, CHENG W, WU Y, et al. DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text[C]//International Conference on Learning Representations. 2024: 48572-48597.

[53] GUO Z, YU S. AuthentiGPT: detecting machine-generated text via black-box language models denoising[J]. arXiv preprint arXiv:2311.07700, 2023.

[54] WEI D, MAO M, FANG X, et al. Short-PHD: detecting short LLM-generated text with topological data analysis after off-topic content insertion[J]. arXiv preprint arXiv:2504.02873, 2025.

[55] ZHU X, REN Y, CAO Y, et al. Reliably bounding false positives: a zero-shot machine-generated text detection framework via multiscaled conformal prediction[C] //Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025: 12298-12319.

[56] ZENG Z, LIU S, SHA L, et al. Detecting AI-generated sentences in human-AI collaborative hybrid texts: challenges, strategies, and insights[C]//Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. 2024: 7545-7553.

[57] LIU Y, ZHANG Z, ZHANG W, et al. ArguGPT: evaluating, understanding, and identifying argumentative essays generated by GPT models[J]. arXiv preprint arXiv:2304.07666, 2023.

[58] CHEN Z, FENG Y, DANG J, et al. IPAD: Inverse Prompt for AI Detection-A Robust and Interpretable LLM-Generated Text Detector[J]. Advances in Neural Information Processing Systems, 2026, 38: 167441-167467.

[59] KIRCHENBAUER J, GEIPING J, WEN Y, et al. A watermark for large language models[J]. Proceedings of Machine Learning Research, 2023, 202: 17061-17084.

[60] KRISHNA K, SONG Y, KARPINSKA M, et al. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense[C]//Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023: 27469-27500.

[61] CHRIST M, GUNN S, ZAMIR O. Undetectable watermarks for language models[C]//Proceedings of the Thirty-Seventh Annual Conference on Learning Theory. 2023: 1125-1139.

[62] 郭钊均, 李美玲, 周杨铭, 等. 人工智能生成内容模型的数字水印技术研究进展[J]. 网络空间安全科学学报, 2024, 2(1): 13-39.

[63] LIU A, PAN L, LU Y, et al. A survey of text watermarking in the era of large language models[J]. ACM Computing Surveys, 2024, 57(2): 1-36.

[64] LIU A, PAN L, HU X, et al. A Semantic Invariant Robust Watermark for Large Language Models[C]//International Conference on Learning Representations. 2024: 6499-6519.

[65] HOU A, ZHANG J, HE T, et al. SemStamp: a semantic watermark with paraphrastic robustness for text generation[C]//Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2024: 4067-4082.

[66] CHEN R, WU Y, GUO J, et al. Improved unbiased watermark for large language models[C]//Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025: 20587-20601.

[67] WANG Y, REN Y, CAO Y, et al. From trade-OFF to synergy: a versatile symbiotic watermarking framework for large language models[C]//Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025: 10306-10322.

[68] YANG X, ZHANG J, CHEN K, et al. Tracing text provenance via context-aware lexical substitution[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 36: 10. 2022: 11613-11621.

[69] YANG X, CHEN K, ZHANG W, et al. Watermarking text generated by black-box

language models[J]. arXiv preprint arXiv:2305.08883, 2023.

[70] HAO J, QIANG J, ZHU Y, et al. Post-hoc watermarking for robust detection in text generated by large language models[C]//Proceedings of the 31st International Conference on Computational Linguistics. 2025: 5430-5442.

[71] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving Language Understanding by Generative Pre-Training[J]. OpenAI Technical Report, 2018.

[72] MAO R, CHEN G, ZHANG X, et al. GPTEval: A survey on assessments of ChatGPT and GPT-4[C]//Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). 2024: 7844-7866.

[73] LUND B D, WANG T. Chatting about ChatGPT: how may AI and GPT impact academia and libraries?[J]. Library Hi Tech News, 2023, 40(3): 26-29.

[74] LI Y, LI Z, ZHANG K, et al. ChatDoctor: A medical chat model fine-tuned on a large language model Meta-AI (LLaMA) using medical domain knowledge[J]. Cureus, 2023, 15(6).

[75] JAWAHAR G, SAGOT B, SEDDAH D. What does BERT learn about the structure of language?[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 3651-3657.

[76] NIE S, ZHU F, YOU Z, et al. Large Language Diffusion Models[J]. Advances in Neural Information Processing Systems, 2026, 38: 50608-50646.

[77] KAZEMNEJAD A, PADHI I, NATESAN RAMAMURTHY K, et al. The impact of positional encoding on length generalization in transformers[J]. Advances in Neural Information Processing Systems, 2023, 36: 24892-24928.

[78] 刘建伟, 刘俊文, 罗雄麟. 深度学习中注意力机制研究进展[J]. 工程科学学报, 2021, 43(11): 1499-1511.

[79] BEBIS G, GEORGIOPOULOS M. Feed-Forward Neural Networks[J]. IEEE Potentials, 2002, 13(4): 27-31.

[80] 张焕, 张庆, 于纪言. 激活函数的发展综述及其性质分析[J]. 西华大学学报 (自然科学版), 2021, 40(4): 1-10.

[81] WILT C, THAYER J, RUML W. A comparison of greedy search algorithms[C]// Proceedings of the International Symposium on Combinatorial Search: vol. 1: 1. 2010: 129-136.

[82] RENZE M. The Effect of Sampling Temperature on Problem Solving in Large Language Models[C]//Findings of the Association for Computational Linguistics: EMNLP 2024. Miami, FL, USA: Association for Computational Linguistics, 2024: 7346-7356.

[83] KOOL W, VAN HOOF H, WELLING M. Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement[J]. Journal of Machine Learning Research, 2020, 21(47): 1-36.

[84] RAVFOGEL S, GOLDBERG Y, GOLDBERGER J. Conformal Nucleus Sampling [C]//Findings of the Association for Computational Linguistics: ACL 2023. 2023: 27-34.

[85] FREITAG M, AL-ONAIZAN Y. Beam search strategies for neural machine translation[C]//Proceedings of the First Workshop on Neural Machine Translation. 2017: 56-60.

[86] PAN L, LIU A, HE Z, et al. MarkLLM: An Open-Source Toolkit for LLM Watermarking[C]//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2024: 61-71.

[87] DODGE J, SAP M, MARASOVIC A, et al. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021: 1286-1305.

[88] MICHEL G, EPURE E V, HENNEQUIN R, et al. Evaluating LLMs for Quotation Attribution in Literary Texts: A Case Study of LLaMa3[C]//Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computa-

tional Linguistics: Human Language Technologies (Volume 2: Short Papers). 2025: 742-755.

攻读硕士学位期间取得的研究成果

[1] BAO Siyuan, SHI Ying, YANG Zhiguang, et al. Yet Another Watermark for Large Language Models[C]//2025 13th International Conference on Communications and Broadband Networking (ICCBN). Chengdu, China, 2025: 51-55. （一作）

[2] 张新鹏, 鲍思源, 吴汉舟. 一种基于偏置选择的模型文本水印嵌入和检测方法: 2026104090531 [P]. 2026-03-31. （二作，导师一作）

致　谢

书写到此，我猛然惊觉，这一刻，是真的结束了吧……几年的硕士时光，也像实验室深夜仍亮着的灯、校园路上铺满的樟树果实一样，被悄悄装订进了这一册论文之中。在上海大学，我不仅学会了如何完成科研工作，也学会了如何面对压力、如何在失败后重新出发。学校开放包容的氛围，让我能够不断接触新的知识与思想，也给我机会遇到了形形色色的同路人。

我衷心感谢张新鹏老师将我带进这个团队，您每一次会议上的发言，同您的每一次交流，都在诠释学术研究是有意义有温度的。感谢吴汉舟老师在论文选题、研究思路以及实验设计过程中给予我的耐心指导，您将“严谨、勤奋、求实、创新”的学风融入自身，成为学生最好的榜样，这让我深刻体会到研究生导师教书育人的真正含义。感谢课题组的各位同学，大家在科研讨论中总会迸发出新的创意，那些一起赶 deadline、一起调参数、一起吐槽服务器崩掉的时刻，最终都变成了值得珍藏的记忆。实验室里键盘敲击声与讨论声交织的日常，也构成了我研究生阶段最真实的底色。

此外，还要感谢参与论文评审与答辩的各位专家老师。感谢各位老师在百忙之中审阅本文，并提出宝贵的意见与建议。您专业的知识也给本篇研究给本篇研究输送了给养，铸就了更完整的作品，向各位专家老师致以诚挚的敬意与感谢。

感谢我的朋友们。感谢你们在我焦虑、低落和自我怀疑的时候给予陪伴，会托起我发在社交软件上不起眼的消极情绪，微光吸引微光，微光照亮微光。研究生阶段像一场漫长的马拉松，我是那逆风飞翔的鸟，无惧风暴的咆哮，羽翼上闪耀着不屈的光芒，心中有信念，未来便可期。最后的最后，我最想感谢的是我的家人始终如一的理解与支持。你们对研究生知之甚少，却愿意相信我、鼓励我，成为我继续坚持下去的动力。我知道总有一天我们的努力会重归尘土，太阳会吞噬我们唯一拥有的地球，但我还是爱你们，这是我的生命从开始到结束的唯一本能。

至此，学生时代即将翻过一页。回望这段时光，有焦虑，有遗憾，也有收获与成长。那些曾经觉得难熬的日子，如今回头看时，反而像深夜实验室窗外的一盏灯，微弱却明亮。谨以此文，献给所有陪伴我走过这段旅程的人。