中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

Multi-bit Watermark in LLMs

Eden Wang 2025.5.9

https://ascii-iie.github.io/

Roadmap

- **Paradigm A: Enumeration-based multi-bit watermark**
  - Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs (ICLR 2024)
- **Paradigm B: Bit assignment-based multi-bit watermark**
  - Advancing Beyond Identification: Multi-bit Watermark for Large Language Models via Position Allocation (NAACL 2024)
  - Provably Robust Multi-bit Watermarking for AI-generated Text (USENIX 2025)
- **Paradigm C: Post Processing-based multi-bit watermark**
  - Waterfall: Framework for Robust and Scalable Text Watermarking and Provenance for LLMs (EMNLP 2024)
- **Paradigm D: Training-based multi-bit watermark**
  - REMARK-LLM: A Robust and Efficient Watermarking Framework for Generative Large Language Models (USENIX 2024)
  - Robust Multi-bit Text Watermark with LLM-based Paraphrasers (ICLR 2025 Workshop)

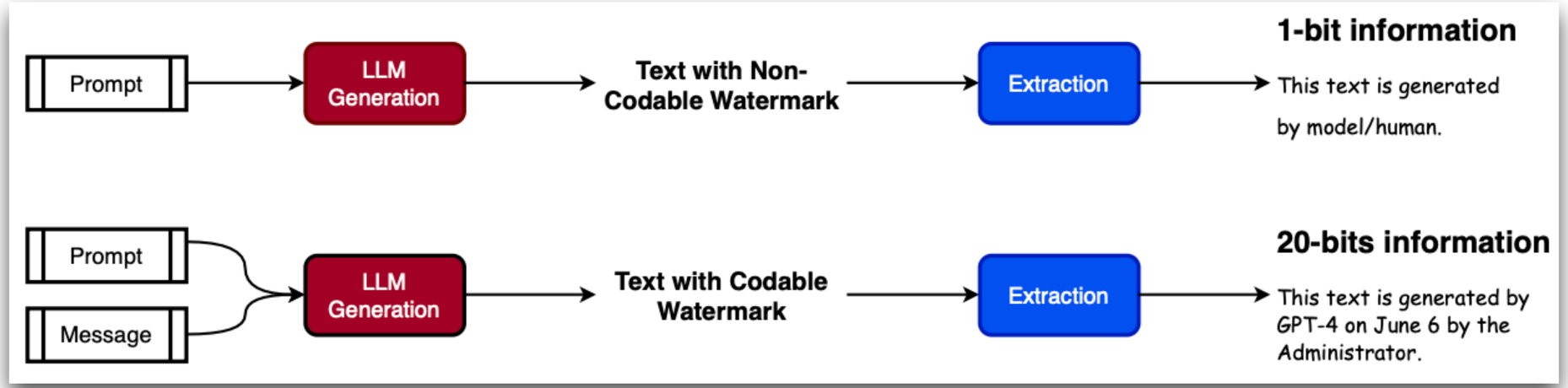
A Survey of Text Watermarking in the Era of Large Language Models（Arxiv 2024.8）

Paradigm A: Enumeration-based

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Background

随着 LLM 应用场景越来越多样化，灵活编码各种定制化信息（如服务商，模型版本，生成时间，UserID 等）的需求也变得越来越强，因此，本文首次设计了 LLM 生成阶段的多比特水印技术。



* 多比特水印评估指标：可检测性 ↔ 鲁棒性 ↔ 编码率 ↔ 计算开销 ↔ 文本质量

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Formulation

prompt, message, text

Encoding: $P \times M \to T, \mathrm{Enc}(\mathbf{x}^{\mathrm{prompt}}, m) = \mathbf{t}$

Decoding: $T \to M, \mathrm{Dec}(\mathbf{t}) = m$

message function

Decoding Target: $m = \arg\max_{m' \in M} P_w(m' \mid \mathbf{t})$

Bayes Formula

✓ 枚举所有消息，选择使得文本概率最高的消息

$$m = \arg\max_{m' \in M} P_w(\mathbf{t} \mid m') = \arg\max_{m' \in M} \{ \sum_{l=1}^{L} \log P_w(t_l \mid m', \mathbf{t}_{:(l-1)}) \}$$

Encoding Target:

$$\max_{\mathbf{t}} \left\{ P_w(\mathbf{t} \mid m) / \max_{m' \neq m} P_w\left(\mathbf{t} \mid m'\right) \right\} \Longleftrightarrow \max_{\mathbf{t}} \left\{ \sum_{l=1}^{L} \log P_w\left(t_l \mid m, \mathbf{t}_{:(l-1)}\right) - \max_{m' \neq m} \sum_{l=1}^{L} \log P_w\left(t_l \mid m', \mathbf{t}_{:(l-1)}\right) \right\}$$

$$s.t.\ \mathrm{PPL}(\mathbf{t} \mid \mathbf{x}^{\mathrm{prompt}}) \leq \mathrm{PPL}(\mathbf{t}^{ori} \mid \mathbf{x}^{\mathrm{prompt}}) + \epsilon$$

✓ 编码阶段目标，即最大化文本 $\mathbf{t}$ 在嵌入消息 m 下生成概率与其他消息 m' 下生成概率的 gap，同时保证文本质量

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Message Encoding Framework

拉格朗日乘子法推导得，$$\max_{\mathbf{t}} \sum_{l=1}^{L} \left\{ \log P_{LLM}\left(t_l \mid \mathbf{x}^{\text{prompt}}, \mathbf{t}_{:(l-1)}\right) + \delta \left(\log P_w\left(t_l \mid m, \mathbf{t}_{:(l-1)}\right) - \log P_w\left(t_l \mid \hat{m}, \mathbf{t}_{:(l-1)}\right) \right) \right\}$$

Algorithm 1: A General Message Encoding Framework for A Settled P_w

Input: Language model LLM, prompt $\mathbf{x}^{prompt}$, message m, watermarking weight δ

for $l = 1, \cdots, L$ **do**

1. Calculate $\log P_{LLM}(v|\mathbf{x}^{prompt}, \mathbf{t}_{:(l-1)})\}$ for each v in the vocabulary using LLM;
2. Calculate $\log P_w(v|m, \mathbf{t}_{:(l-1)})$ based on the settled P_w;
3. Select $t_l = \arg\max_v \{\log P_{LLM}(v|\mathbf{x}^{prompt}, \mathbf{t}_{:(l-1)}) + \delta(\log P_w(v|m, \mathbf{t}_{:(l-1)}) - \frac{1}{|\mathcal{M}|} \sum_{m' \in \mathcal{M}} \log P_w(v|m', \mathbf{t}_{:(l-1)})\}$

end

Output: watermarked text $\mathbf{t} = \{t_1, t_2, \cdots, t_L\}$

$$\hat{m} = \arg\max_{m' \neq m} \sum_{l=1}^{L} \log P(t_l \mid m', t_{:(l-1)})$$

需要已知完整的生成文本 $\mathbf{t}$，因此编码时无法计算

近似计算

$$L\left(m, \mathbf{x}^{\text{prompt}}, \mathbf{t}_{:(l-1)}\right) = \max_v \{ \underbrace{\log P_{LLM}\left(v \mid \mathbf{x}^{prompt}, \mathbf{t}_{:(l-1)}\right)}_{\text{model logit}} + \underbrace{\delta \log P_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right) - \frac{1}{|M|} \sum_{m' \in M} \log P_w\left(v \mid m', \mathbf{t}_{:(l-1)}\right)}_{\text{message logit}} \}$$

如何设计 P_w？

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Message Function: Vanilla-Marking

$$L\left(m, \mathbf{x}^{\text{prompt}}, \mathbf{t}_{:(l-1)}\right) = \max_{v}\{\underbrace{\log P_{LLM}\left(v \mid \mathbf{x}^{prompt}, \mathbf{t}_{:(l-1)}\right)}_{\text{model logit}} + \underbrace{\delta \log P_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right) - \frac{1}{|M|}\sum_{m' \in M} \log P_w\left(v \mid m', \mathbf{t}_{:(l-1)}\right)}_{\text{message logit}}\}$$

model logit 是 LLM 对于当前 token 的预测概率，而 message logit 依赖于 P_w，因此我们的目标是设计一个消息函数 P_w，使得在消息 m 下生成 v 的概率能够尽可能超过其他消息生成 v 的均值。

$$\log \hat{P}_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right) = \begin{cases} 1, & h\left(v, m, \mathbf{t}_{:(l-1)}\right) = 1 \\ 0, & h\left(v, m, \mathbf{t}_{:(l-1)}\right) = 0 \end{cases}$$

$$\log P_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right) = \log \frac{\hat{P}_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right)}{\sum_v \hat{P}_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right)}$$

上述 Naive 的想法可类比红绿词表，根据消息 m 哈希决定不同 token 的 message logit

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Message Function: Balance-Marking

Vaniall-Marking 缺点：只考虑最大化 message logit，而忽略了使 message logit 最大的 token v 并不一定具有较高的 model logit，因此模型可能会被强迫生成语义不符的 token。

Algorithm 2: Algorithm of Choosing Subset $V_{m,\mathbf{t}_{:(l-1)}}$ (Practical Version in Red)

Input: Message m, text prefix $\mathbf{t}_{:(l-1)}$, language model LLM, proxy-LM LM_{proxy}, $\mathcal{M}_A = \{1, \cdots, A\}$, hash functions h and $\hat{h}$.

1. Calculate a seed $s = h(m, \mathbf{t}_{:(l-1)})$, or $s = h(\hat{h}(m), \mathbf{t}_{:(l-1)})$ where $\hat{h}$ maps m to $\hat{h}(m) \in \mathcal{M}_A$;
2. Shuffle the vocab list $(v_1, \cdots, v_{|\mathcal{V}|})$ to $(v'_1, \cdots, v'_{|\mathcal{V}|})$ with the seed s;
3. Select the first k tokens in the shuffled list so that k is the minimal value to make $\{v'_1, \cdots, v'_k\}$ satisfy Eq. (9) or Eq. (11).

Output: $V_{m,\mathbf{t}_{:(l-1)}} = \{v'_1, \cdots, v'_k\}$

$$\sum_{v \in V_{m,\mathbf{t}_{:(l-1)}}} P_{\text{LLM}}(v \mid x^{\text{prompt}}, \mathbf{t}_{:(l-1)}) \geq \sigma$$

0.5

$$\log \hat{P}_w\left(v \mid m, \mathbf{t}_{:(l-1)}\right) = \begin{cases} 1, & v \in V_{m,\mathbf{t}_{:(l-1)}} \\ 0, & v \notin V_{m,\mathbf{t}_{:(l-1)}} \end{cases}$$

✓ 为了保证编码解码的一致性，删除 $\mathbf{x}^{\text{prompt}}$，将 $\mathbf{t}_{:(l-1)}$ 截断为固定长度窗口 $\mathbf{t}_{l-1-L:(l-1)}$

✓ 为了适用不同场景，用代理模型 LM_{proxy} 选择 $V_{m,\mathbf{t}_{:(l-1)}}$

✓ 为了高效解码，用哈希 $\hat{h}$ 将消息空间映射到更小的空间中：$m \rightarrow \hat{h}(m, \text{prefix\_token}) \in M_A = \{1, \cdots, A\}$

王乐安 星期一
收件人： 王一丹 >

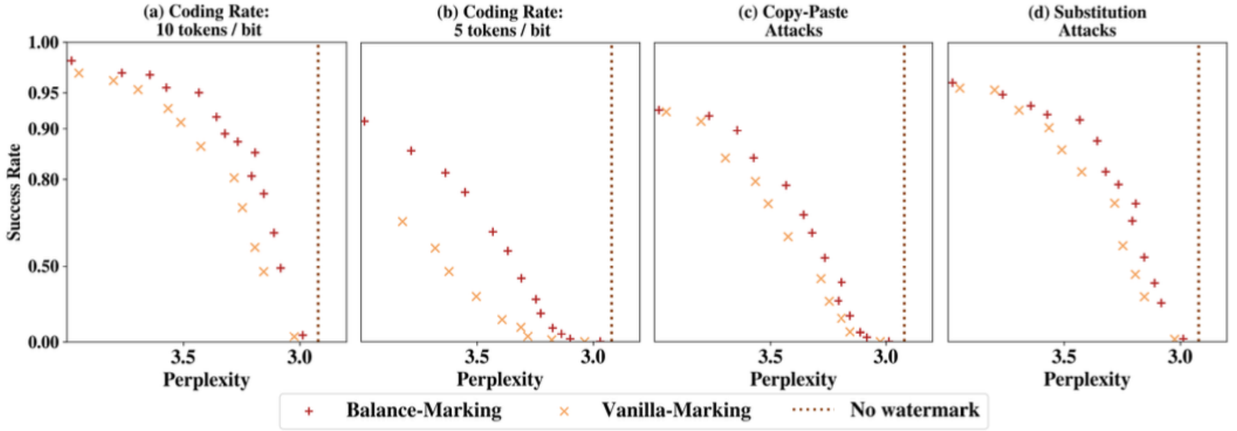
回复： Re: Re: Re: Re: Question Regarding "Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs"

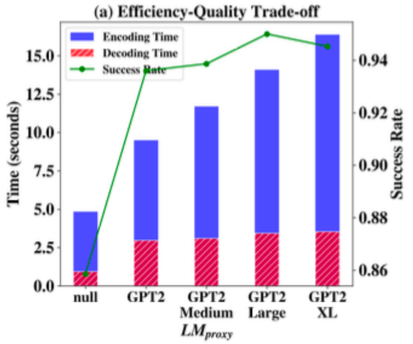
哦对哦，不好意思，你说的对，这儿写错了，\hat{h}也要接受prefix token作为参数的，我的锅

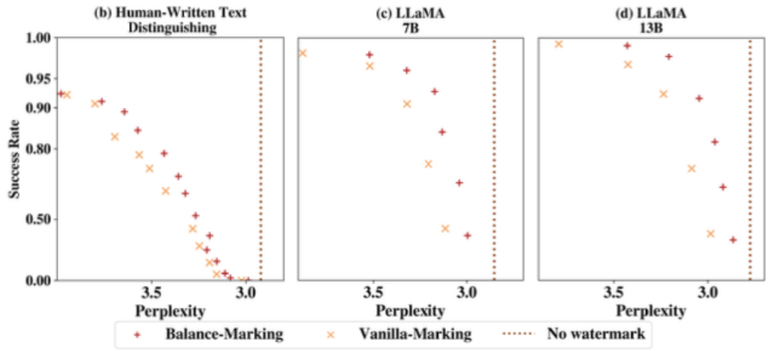
Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs

Experiment





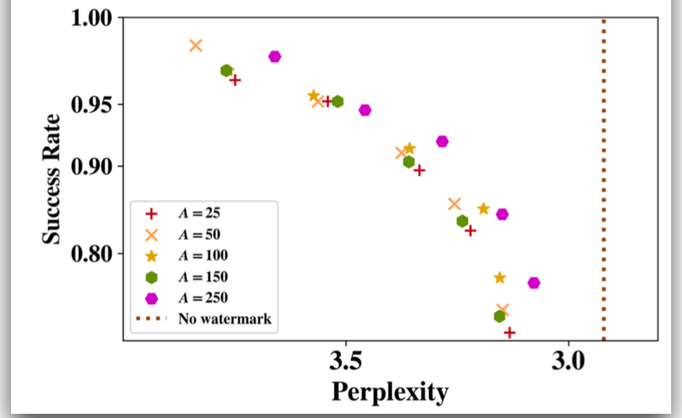


Coding Rate 即平均嵌入 1 bit 信息需要多少 token

Experiment Setup

- Models: OPT-1.3B, LLaMA-7/13B, Proxy LM: GPT-2
- Datasets: C4 dataset (500 prompt + 200 token)
- Message space: 20 bit, $M = \{0,1,\cdots,2^{20}-1\}$
- Hyper-parameters: $A = 100$, $L = 10$, $\sigma = 0.5$

A	25	50	100	150	250
Encoding Time (s)	9.39	9.48	9.50	9.22	9.37
Decoding Time (s)	1.89	2.27	2.97	4.49	6.24



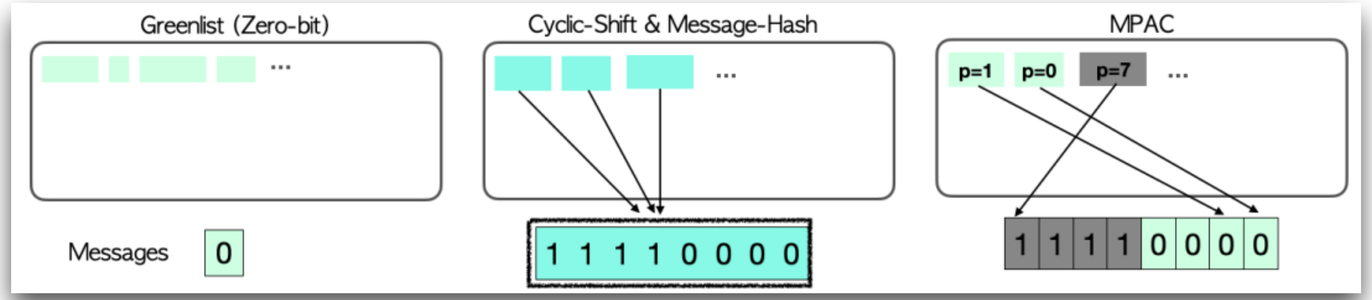
Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs（Peking University. ICLR 2024）

Paradigm B: Bit assignment-based

Advancing Beyond Identification: Multi-bit Watermark for Large Language Models via Position Allocation

Outline

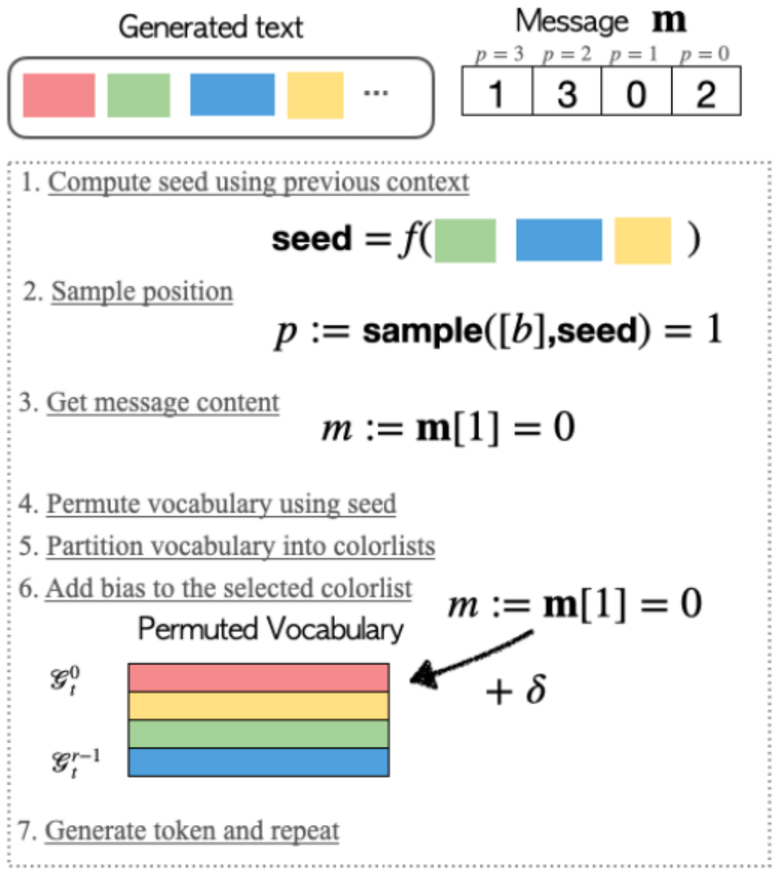
将词表切分为 2^r 份扩大每个比特的容量，并将每个 token 通过哈希分配给不同的消息比特位。



Algorithm 1: Message Decoding

Input: Text $X_{1:T}$, context width h, effective message length $\tilde{b}$, counter $\mathbf{W} \in \mathbb{R}^{\tilde{b} \times r}$
Output: Predicted message $\hat{\mathbf{m}}$, number of colorlisted tokens w

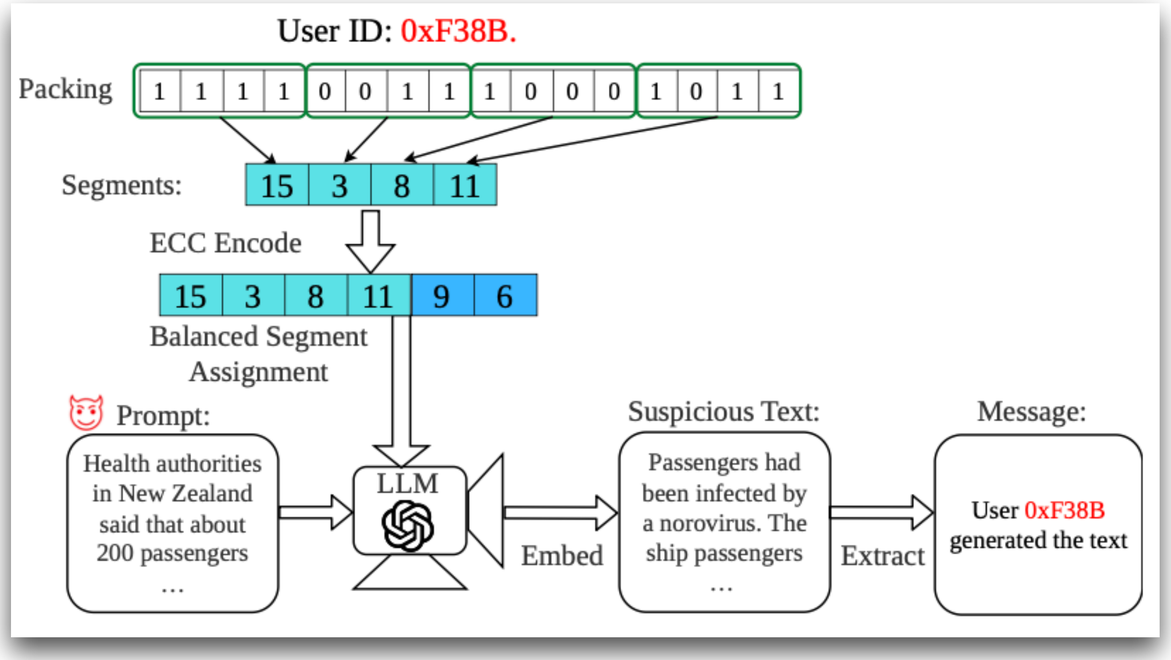
```
/* Initialize counter */
W[p][m] = 0 ∀p, m
/* Count tokens in colorlists */
for t in [h+1, T] do
    s = f(X_{t-h:t-1})
    p = sample([b̃]) using s as seed
    V_t = permute(V_t) using s as seed
    for m in [r] do
        if X_t ∈ G_t^m then
            W[p][m] += 1
            continue
        end
    end
end
/* Predict message */
m̂_r = " "
w = 0
for p in [b̃] do
    w += max(W[p][:])
    m̂ = argmax(W[p][:])
    m̂_r += str(m̂)
end
Get bit message m̂ by converting m̂_r
return m̂, w
```

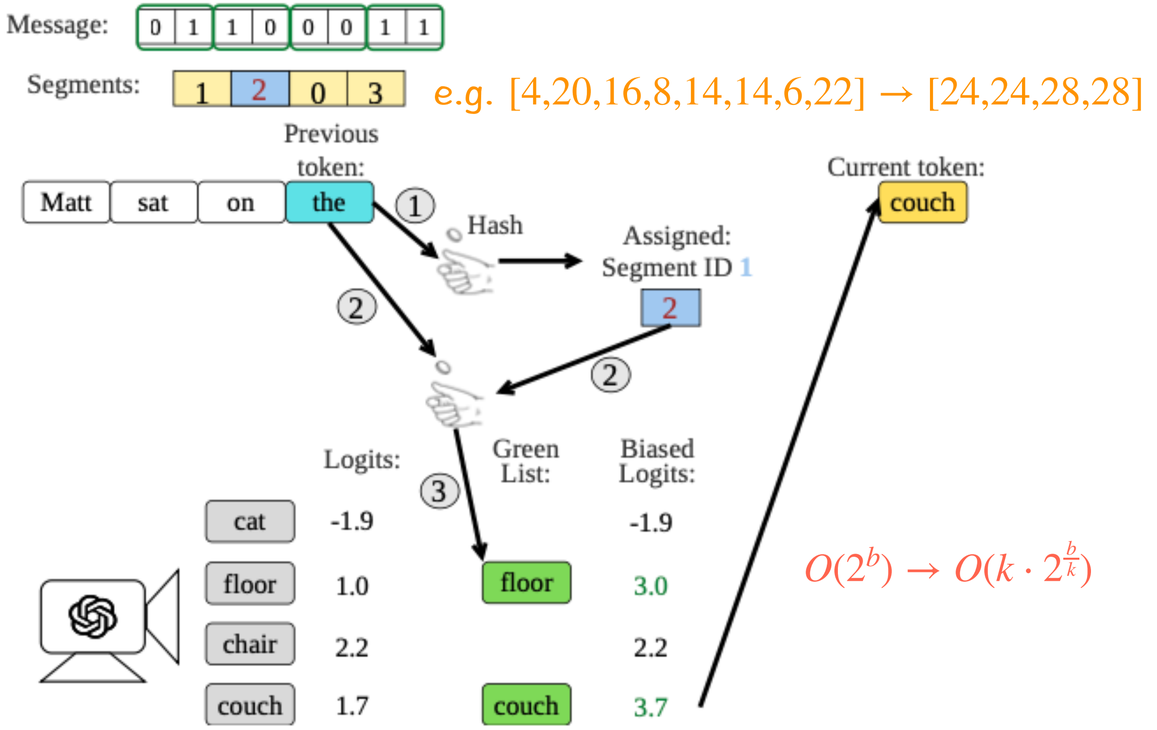


Provably Robust Multi-bit Watermarking for AI-generated Text

Outline

MPAC 缺点在于 token 词频不同会使比特位分配不均匀，并且词表切分越多会对文本质量影响越大，因此本文将消息多个比特聚合成更少的 segments，并引入 RS 纠错码提高解码的准确率。





Provably Robust Multi-bit Watermarking for AI-generated Text（National University of Singapore. USENIX Security 2025）

Provably Robust Multi-bit Watermarking for AI-generated Text

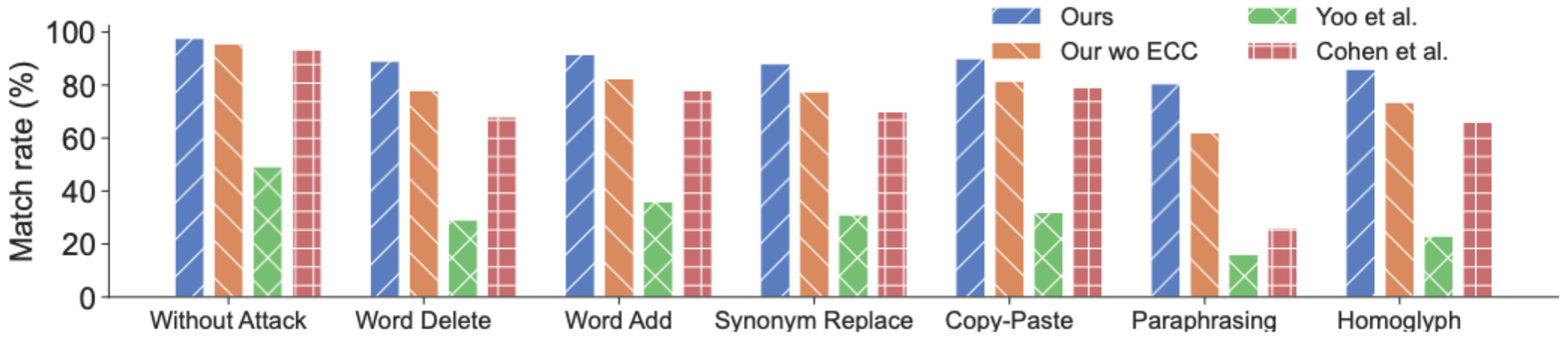
Experiment

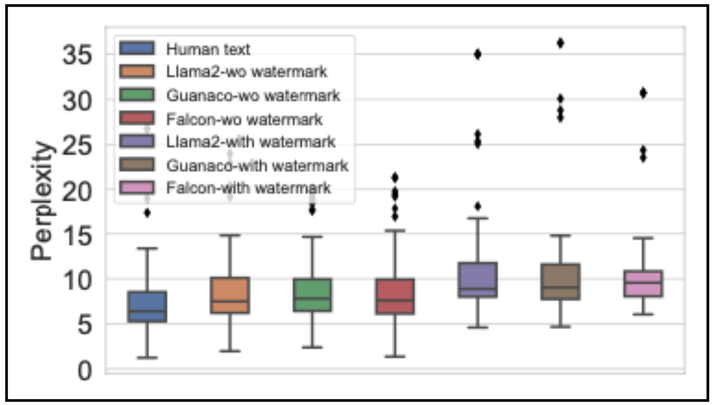
本文从多比特检测准确率、解码时间、攻击鲁棒性、文本质量多个方面评估了方法的有效性。

Bit length b	12			16			20			24			32		
	Match rate	Bit Acc	Time (s)	Match rate	Bit Acc	Time (s)	Match rate	Bit Acc	Time (s)	Match rate	Bit Acc	Time (s)	Match rate	Bit Acc	Time (s)
Fernandez et al. [19]	99.6	100.0	0.12	99.6	100.0	0.34	99.2	99.9	5.04	98.0	99.8	110	NA	NA	29000 (Estimated)
Wang et al. [65]	99.6	100.0	0.16	98.8	99.9	0.58	98.4	99.8	3.14	97.2	99.6	35.5	NA	NA	8300 (Estimated)
Yoo et al. [71]	86.4	96.5	0.01	73.6	94.6	0.01	49.2	90.8	0.01	30.4	89.4	0.01	8.4	81.2	0.01
Cohen et al. [13]	93.2	98.7	0.01	88.8	97.0	0.02	78.4	95.3	0.02	65.6	94.7	0.03	27.2	89.7	0.04
Ours	98.8	99.9	0.04	98.0	99.7	0.06	97.6	99.6	0.10	96.0	99.5	0.18	94.0	99.1	0.6

Experiment Setup

- Models: LLaMA-2-7B
- Datasets: OpenGen, C4, Essays
- Metric: Match Rate, Bit Acc, PPL
- Hyper-parameters: $\delta = 6$, $b = 20$



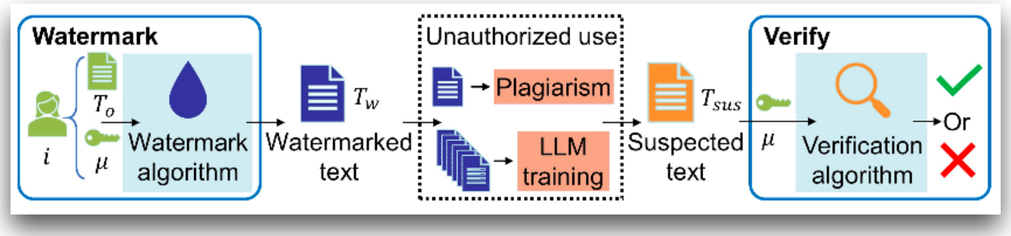


Paradigm C: Post Processing-based

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs

Outline

本文希望设计一种通用（支持不同文本格式如自然语言或者代码）、鲁棒且可扩展（计算成本低可支持数百万用户）的文本水印框架，以保护知识产权免遭抄袭以及未授权的大模型训练。



$\mathbb{T}^W_{c,s} = \{T \in \mathbb{T} : S(T_o, T) \geq s\}$，其中 S 为文本保真度指标

✓ 用 LLM 作为重述器，增加集合大小

1. 水印：$W(\mu_i, T_o) \rightarrow T_w^{(i)}$，其中 μ_i 为水印 ID，T_o 为原始文本
2. 攻击：$A(T_w^{(i)} \rightarrow T_{\text{sus}}^{(i)})$，其中 $T_{\text{sus}}^{(i)}$ 为攻击者处理后的文本
3. 验证：$q = V(\mu_i, T_{\text{sus}})$，若 $q \geq \bar{q}$ 则认为该文本有水印

✓ 每 n-gram 嵌入水印信息，增强鲁棒性

$\mathbb{A}_1$:增删改文本 $T_w^{(i)}$ 中的单词

$\mathbb{A}_2$:用 LLM 翻译、重述文本 $T_w^{(i)}$

$\mathbb{A}_3$:向文本 $T_w^{(i)}$ 中嵌入不同的水印

$\mathbb{A}_4$:将文本 $T_w^{(i)}$ 作为上下文提示

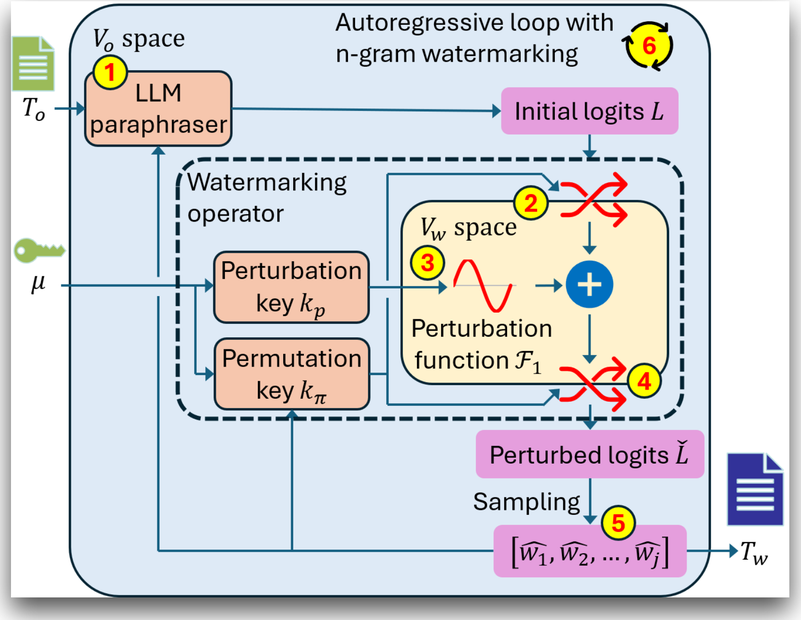
$\mathbb{A}_4$:将文本 $T_w^{(i)}$ 用于微调 LLM

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs（National University of Singapore. EMNLP 2024）

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs

Watermarking schematic

1. 利用 LLM 作为重述器对原始文本 T_0 进行重述，对第 j 个 token 产生初始的 logits L_j
2. 利用 ID μ 和元数据 z 构建排序密钥 $k_\pi = h_\pi(\mu, \hat{w}_{j-n+1:j-1}) \in \mathbb{K}_\pi$ 和扰动密钥 $k_p = h_p(\mu, z) \in \mathbb{K}_p$
3. 对 logits 进行扰动：$\check{L}_j = W(k_\pi, k_p, L_j) = P^{-1}(k_\pi, F(k_p, \kappa, P(k_\pi, L_j)))$，$\kappa$ 为控制扰动幅度的标量
4. 从扰动的 logits 中采样，生成下一个 token



P为排列词表运算

$|F_1| \cdot |\mathbb{V}|!$

$$F_1 : \mathbb{K}_p \hookrightarrow \{\phi : V_w \to \mathbb{R}^{|V_w|} \mid \langle \phi_i, \phi_l \rangle = \delta_{il}\}$$

$$F(k_p, \kappa, L_j) = L_j + \kappa F_1(k_p)$$

$$\phi_{k_p}(j) = \begin{cases} \cos\left(2\pi k_p \frac{j}{|\mathbb{V}|}\right) & \text{if } k_p \le \frac{|\mathbb{V}|}{2} \\ \sin\left(2\pi\left(k_p - \frac{|\mathbb{V}|}{2}\right)\frac{j}{|\mathbb{V}|}\right) & \text{otherwise} \end{cases}$$

傅立叶基函数

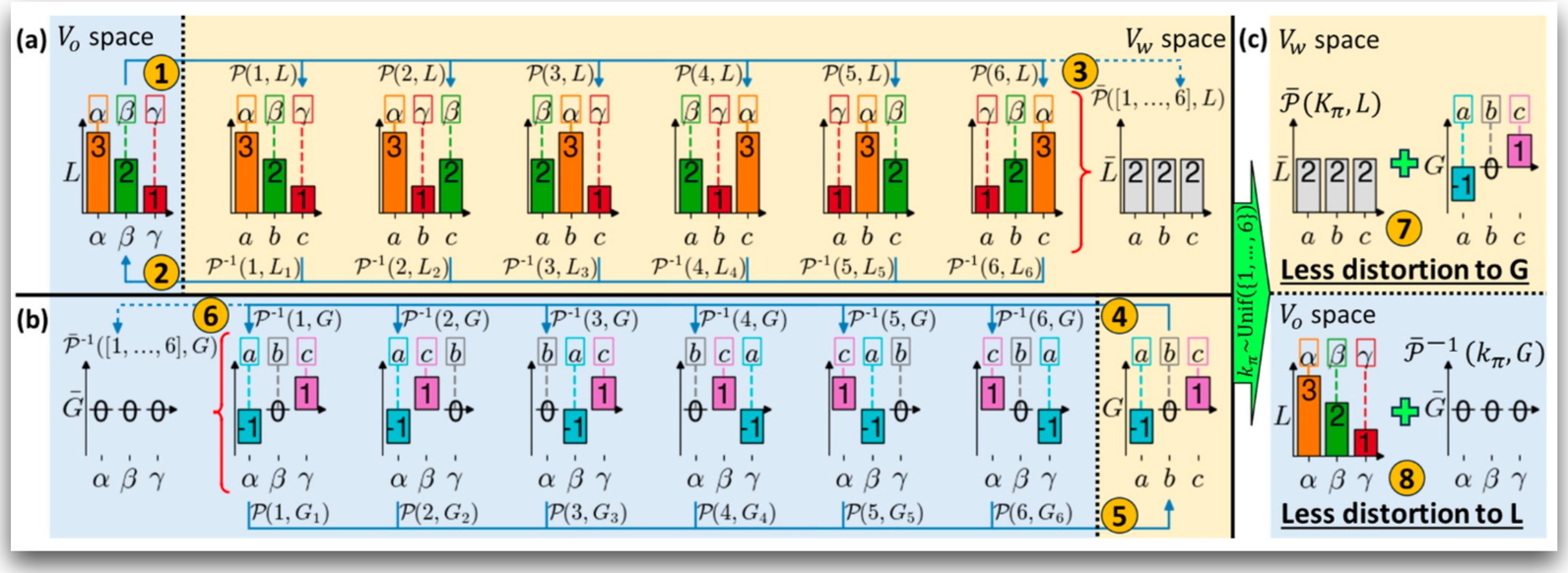
每个密钥 k_p 对应预先定义的正交函数族的唯一函数

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs（National University of Singapore. EMNLP 2024）

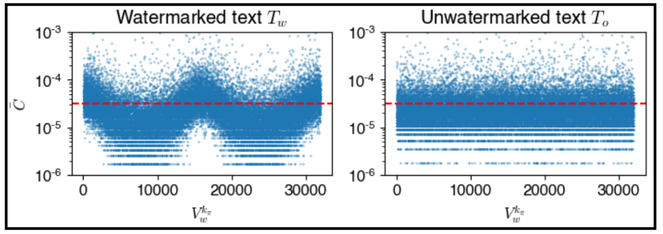
Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs

Toy Example

- 如图 a,c 所示，若在所有步骤对 L 应用 P 转换至 V_w 空间并添加扰动 G，则期望上将获得均匀分布 $\overline{L}$，因此对扰动后的信号 $\overline{L}+G$ 应用 $\overline{P}^{-1}$，将使 G 不产生失真，从而提高可验证性。
- 如图 b,c 所示，若在所有步骤对 G 应用 P^{-1} 转换回 V_o 空间生成 token，则期望上将获得均匀分布的噪声 $\overline{G}$，因此对还原后的信号 $L+\overline{G}$ 应用 $\overline{P}$，将使 L 不产生失真，从而提高保真度。



$$\overline{P}\left(K_\pi, L\right) \triangleq \frac{1}{\left|K_\pi\right|} \sum_{k_\pi \in K_\pi} P\left(k_\pi, L\right)$$

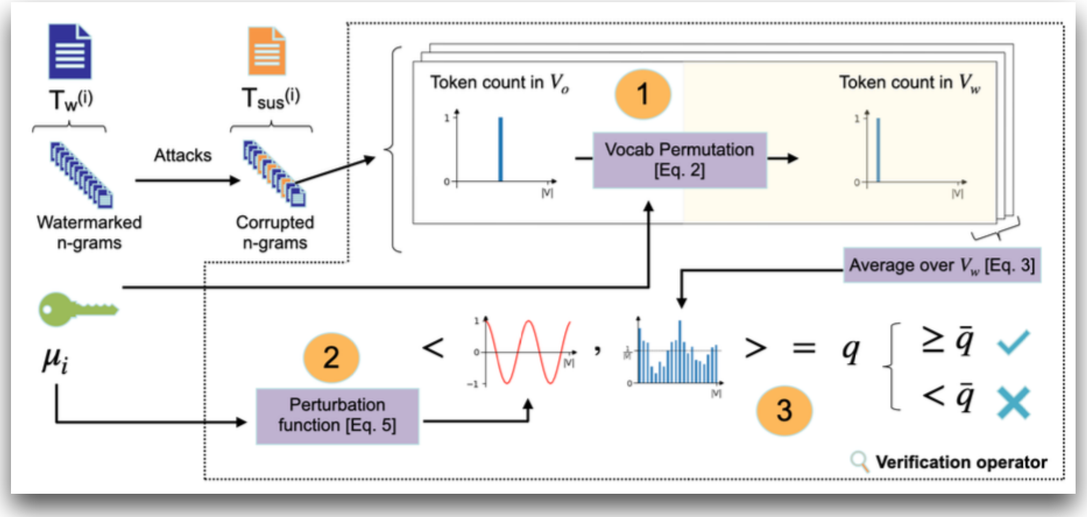


Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs（National University of Singapore. EMNLP 2024）

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs

Verification schematic

1. 利用 μ 和先前 $n-1$ 个 token 将 T_{sus} 中的 token 进行排列，从 V_o 映射至 V_w
2. 在水印空间 V_w 中统计 token 数目获得 token 累积分布 C
3. 计算与 μ 相关的扰动函数 $F_1(k_p)$ 的水印分布和累积分布 C 的内积得到水印分数 q
4. 加水印文本将具有与 $F_1(k_p)$ 相似的分布 C，因此可比较分数 q 和阈值 $\bar{q}$，判断有无水印



Algorithm 2 WATERFALL Verification algorithm

Input: Suspected text $T_{\text{sus}} = [\hat{w}_1, \ldots, \hat{w}_N]$, ID μ, n-gram length n, keys function h_π, perturbation key k_p, test threshold $\bar{q}$.
Initialize a vector C of length $|V_o|$, which keeps track of token counts, to 0.
for $j = 1, \ldots, |T_{\text{sus}}|$ **do**
  Compute $k_\pi = h_\pi(\mu, \hat{w}_{j-n+1:j-1})$ and permutation operator $\mathcal{P}(k_\pi)$, given Equation (2).
  Set $C(\mathcal{P}(k_\pi, \hat{w}_i)) + +$.
end for
Compute avg cumulative token distribution $\bar{C} = C/N$.
Compute verification score $q = \langle \bar{C}, \frac{\mathcal{F}_1(k_p)}{\|\mathcal{F}_1(k_p)\|_2} \rangle$ based on Equation (5).
Output: Returns true if $q \geq \bar{q}$.

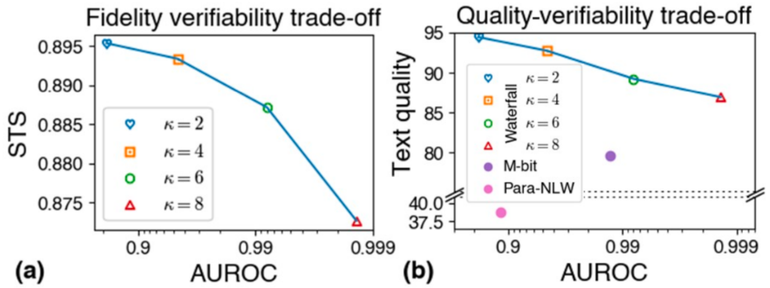
无水印文本或使用不同 μ 的文本将得到平坦的 C，使用不同 k_p 将得到正交的 C，因此分数较低

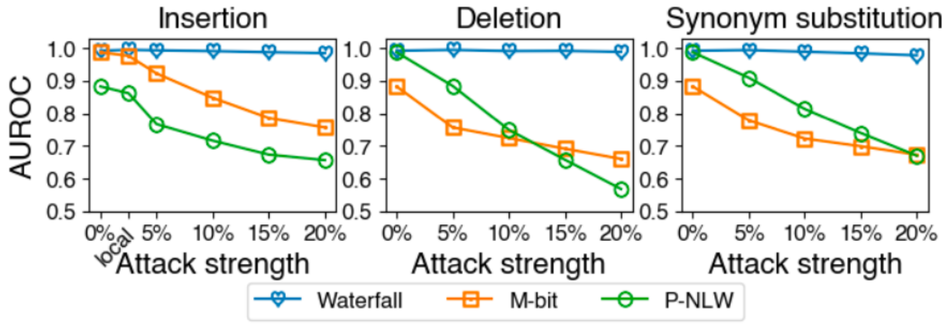
Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs（National University of Singapore. EMNLP 2024）

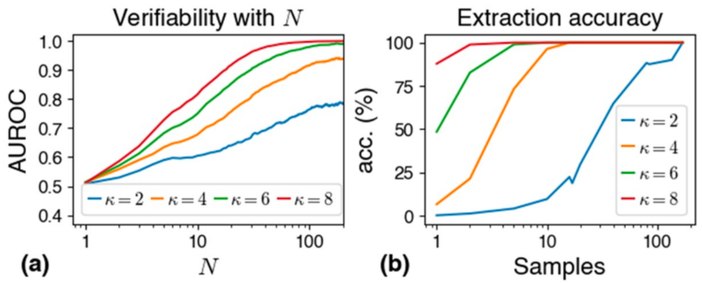
Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs

Experiment

本文从可检测性、文本质量、鲁棒性、可扩展性角度分别进行了实验，结果表明，即使对于长度仅为 100 token 的较短文本，waterfall 也可以实现更优的可检测性和更高的文本质量。







Token length N

	Fidelity (Pass@10)	Verifiability (AUROC) Pre-attack	Verifiability (AUROC) Post-attack	Scalability (# of users)
SRCMARKER	0.984	0.726	0.662	10^5
WATERFALL	0.969	0.904	0.718	10^{130274}

	WATERFALL	M-BIT	P-NLW
Watermark	24.8s	**2.97s**	147s
Verification	**0.035s***	2.61s	148s

Experiment Setup

- Models: LLaMA-2-13B（重述器）
- Baselines: M-BIT, P-NLM（文本水印）
- Datasets: OpenGen, C4, Essays
- Metric: AUROC, GPTScore, STS (语义相似度)
- Hyperparameter: k (水印强度)

Waterfall: Scalable Framework for Robust Text Watermarking and Provenance for LLMs（National University of Singapore. EMNLP 2024）

Paradigm D: Training-based

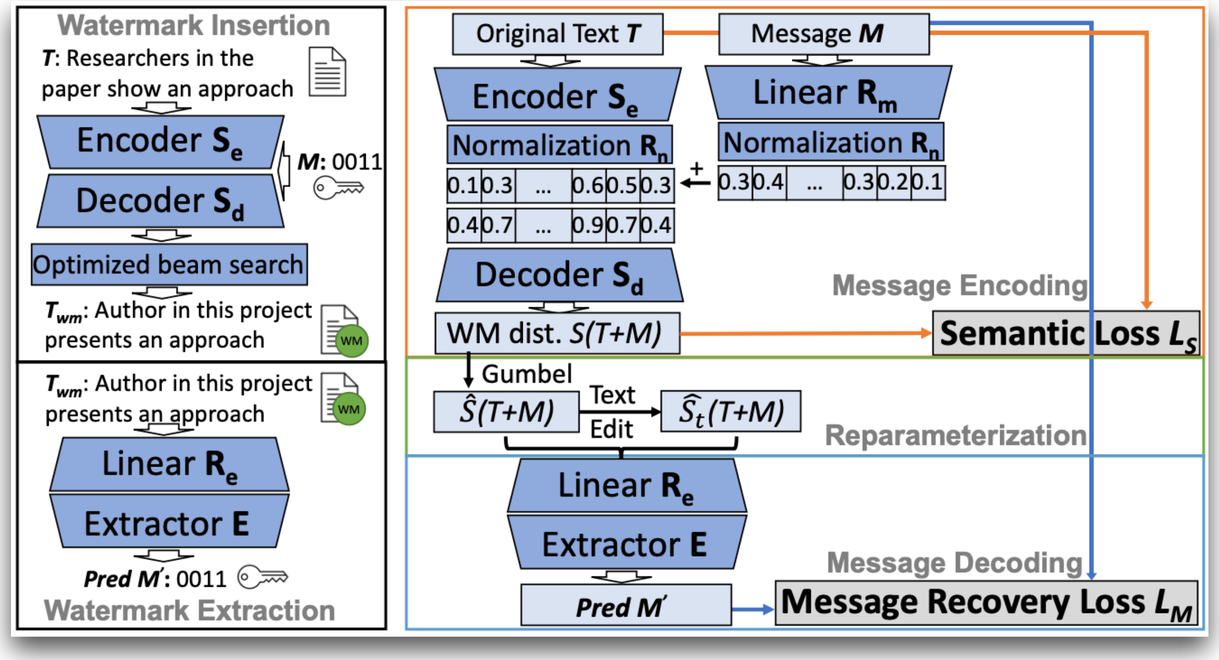
REMARK-LLM: A Robust and Efficient Watermarking Framework for Generative Large Language Models

Outline

✓ 消息编码模块：对文本 T 和消息 M 分别编码，在隐空间将消息嵌入每个 token 得到 $S(T+M)$

✓ 重参数化模块：利用 Gumbel-Softmax 将水印分布 $S(T+M)$ 近似为更加稀疏的编码 $\hat{S}(T+M)$

✓ 消息提取模块：从重参数化的稀疏编码分布中提取得到最终的消息，可增加扰动提高鲁棒性



$$\hat{S}_i = \frac{\exp(\log(S_i) + g_i)/\tau}{\sum_{j=1}^{|V|} \exp((\log(S_j) + g_j)/\tau)}, \text{for } i = 1, \cdots, |T|$$

$$\begin{cases} L_S(T, S(T \cdot T_M + M)) = -\dfrac{1}{|T|} \displaystyle\sum_{i=1}^{|T|} \sum_{j=1}^{|V|} T_{ij} \log(S_{ij}(T \cdot T_M + M)) \\ L_M(M, M', M_t') = w_w \displaystyle\sum_{i=1}^{|M|} |M_i - M'| + w_t \sum_{i=1}^{|M|} |M_i - M_t'| \\ L = w_1 L_s + w_2 L_M \end{cases}$$

mask

对抗扰动

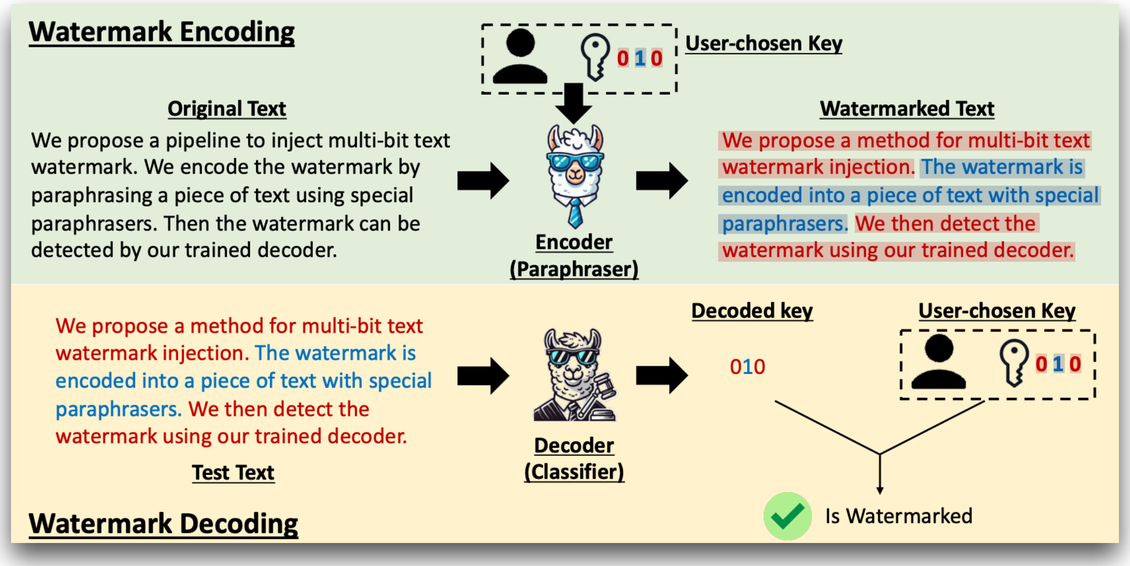
REMARK-LLM: A *R*obust and *E*fficient Water*mark*ing Framework for Generative Large Language Models（University of California, San Diego. USENIX Security 2024）

Robust Multi-bit Text Watermark with LLM-based Paraphrasers

Outline

✓ 编码阶段，训练两个重述器，根据多比特密钥，对原始文本进行分段交替重述生成水印文本

✓ 解码阶段，训练分类器，对水印文本进行句子级分类获得水印密钥的每个比特位



Algorithm 1 Watermark Encoding Algorithm $x^w = E(x^o, M; \mathcal{S}, \theta_0, \theta_1)$.

Require: Input text x^o; Watermark code M; Text segmentor $\mathcal{S}$; Parameters for two paraphrasers θ_0 and θ_1.
Ensure: Watermarked text x^w

$x^w \leftarrow [\,]$
$i \leftarrow 0$ `/* index of current watermark bit */`
while $x^w[-1] \neq \langle \text{EOS} \rangle$ **do**
  $bit \leftarrow M[i]$
  x^w.append$(f(x^o, x^w; \theta_{bit}))$
  `/* Switch to the next bit if the current segmentation ends. */`
  if $\mathcal{S}(x^w; \text{mode}{=}E) = 1$ **then**
    $i \leftarrow i + 1$
  end if
end while
return x^w

Algorithm 2 Watermark Decoding Algorithm $M' = D(x^w; \mathcal{S}, \theta_d)$.

Require: Input text x^w; Text segmentor $\mathcal{S}$; Parameters for the text classifier θ_d.
Ensure: Decoded watermark M'

$M' \leftarrow [\,]$
for $\tilde{x}_i \in \mathcal{S}(x^w; \text{mode}{=}D)$ **do**
  M'.append$(g(\tilde{x}_i; \theta_d))$
end for
return M'

Robust Multi-bit Text Watermark with LLM-based Paraphrasers（Hang Li et al. ICLR 2025 Workshop）

Robust Multi-bit Text Watermark with LLM-based Paraphrasers

Co-Training Framework

本文利用 PPO 强化学习策略，将分类的结果作为奖励，同时交替训练优化编码器和解码器。

$$\ell_{\text{init } it}(\theta_0, \theta_1; x_o^{SFT}, x_{\text{para}}^{SFT}) = \ell_{SFT}(\theta_0; x_o^{SFT}, x_{\text{para}}^{SFT}) + \ell_{SFT}(\theta_1; x_o^{SFT}, x_{\text{para}}^{SFT})$$

$$-\lambda_{JS} \cdot \text{JS}(\pi_{\theta_0}(x_{\text{para}}^{SFT} \mid x_o^{SFT}), \pi_{\theta_1}(x_{\text{para}}^{SFT} \mid x_o^{SFT}))$$

$$\ell_D(\theta_d; x^w, M) = \sum_{i=1}^{|D(x^w)|} \left(M[i] \cdot g_s(\tilde{x}_i^w; \theta_d) + (1 - M[i]) \cdot \left(1 - g_s(\tilde{x}_i^w; \theta_d)\right) \right)$$

$\in (0,1)$

Algorithm 3 Training Algorithm of the Encoder and the Decoder.

Require: Dataset $\mathcal{D}$; Initialized parameters $\theta_0, \theta_1, \theta_d$; Text Segmentor $\mathcal{S}$

Ensure: Trained parameters $\theta_0, \theta_1, \theta_d$

1: **for all** $x_o \in \mathcal{D}$ **do**
2: $M \sim \{0,1\}^\infty$
3: $x_w \leftarrow E(x^o, M; \mathcal{S}, \theta_0, \theta_1)$
4: Calculate the advantage function $A_t(x_w, x_o, M)$ with the reward function in Equation 4.
5: Update θ_d with decoder loss $\ell_D(\theta_d; x^w, M)$ in Equation 2.
6: Update θ_0, θ_1 with the encoder loss $\ell_E(\theta_0, \theta_1; A_t)$ in Equation 5.
7: **end for**
8: **return** M'

$$r_w(x^w, M) = \sum_{i=1}^{\text{len}(D(x^w))} \mathbb{I}D(x^w)[i] = M[i]\}$$

$$r(x^w, x^o, M) = \lambda_w \cdot r_w(x^w, M) + \lambda_s \cdot r_s(x^w, x^o)$$

$$\ell_E(\theta_0, \theta_1) = \sum_t \mathbb{I}\left\{x_t \sim \pi_{\theta_0}(\cdot \mid x_{<t})\right\} \cdot \left(-\mathbb{E}_t \left[\frac{\pi_{\theta_0}(x_t \mid x_{<t})}{\pi_{ref}(x_t \mid x_{<t})} A_t \right] + \lambda_k \text{KL}\left(\pi_{\theta_0}(\cdot \mid x_{<t}), \pi_{ref}(\cdot \mid x_{<t})\right) \right)$$

$$+ \sum_t \mathbb{I}\left\{x_t \sim \pi_{\theta_1}(\cdot \mid x_{<t})\right\} \cdot \left(-\mathbb{E}_t \left[\frac{\pi_{\theta_1}(x_t \mid x_{<t})}{\pi_{ref}(x_t \mid x_{<t})} A_t \right] + \lambda_k \text{KL}\left(\pi_{\theta_1}(\cdot \mid x_{<t}), \pi_{ref}(\cdot \mid x_{<t})\right) \right)$$

Robust Multi-bit Text Watermark with LLM-based Paraphrasers

Experiment

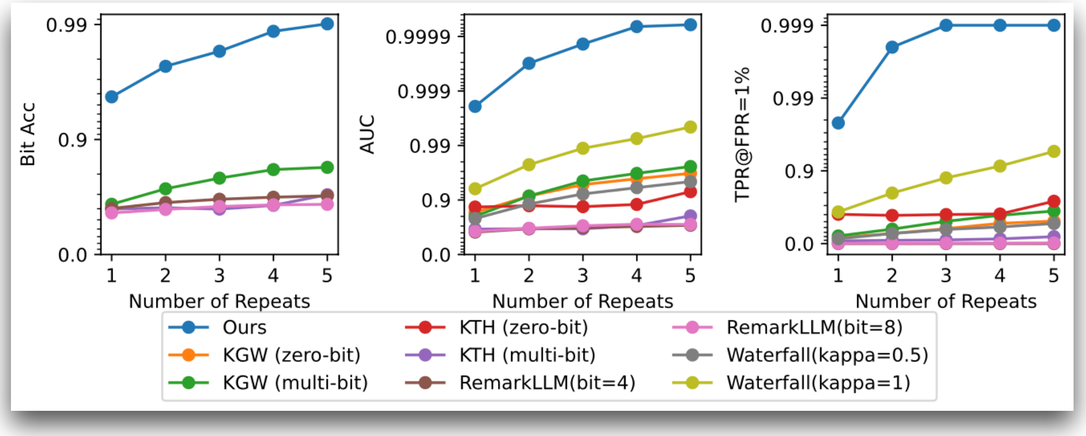
Baseline 中的多比特版本是指将不同水印方案直接当作 LLM-based 的两个重述器。

Method	Bit-wise Accuracy		Text-wise Accuracy			Fidelity
	Bit Acc	Bit Num	AUC	TPR@FPR=1%	TPR@FPR=0.01%	Similarity
RemarkLLM (4bit)	0.7663	4.0	0.7861	0.0%	0.0%	0.8096
RemarkLLM (8bit)	0.6953	**8.0**	0.8023	3.7%	0.0%	0.7793
KGW (zero-bit)	-	-	0.8652	25.9%	18.1%	0.7745
KGW (multi-bit)	0.6381	4.46	0.8327	22.9%	6.3%	0.8123
KTH (zero-bit)	-	-	0.8919	61.4%	46.6%	0.8200
KTH (multi-bit)	0.6129	4.26	0.6775	10.9%	2.3%	0.8176
Waterfall($\kappa = 0.5$)	-	-	0.7787	14.0%	3.8%	0.8499
Waterfall($\kappa = 1$)	-	-	0.9392	62.4%	35.5%	0.8423
Ours	**0.9563**	5.57	**0.9981**	**98.0%**	**78.0%**	**0.8739**

Method	Translate			LlamaPara			PegasusPara		
	bitacc	AUC	TPR@1%	bitacc	AUC	TPR@1%	bitacc	AUC	TPR@1%
RemarkLLM (4bit)	0.6885	0.7142	0.0%	0.7063	0.7311	0.0%	0.7033	0.7248	0.0%
RemarkLLM (8bit)	0.6124	0.6904	1.4%	0.6023	0.6751	1.5%	0.6018	0.6687	1.2%
KGW (zero-bit)	-	0.4872	0.2%	-	0.4872	0.2%	-	0.4900	0.0%
KGW (multi-bit)	0.4997	0.5829	1.6%	0.4765	0.5383	1.5%	0.4817	0.5654	1.5%
KTH (zero-bit)	-	0.8600	30.6%	-	0.8559	32.0%	-	0.8618	43.7%
KTH (multi-bit)	0.4923	0.4990	0.8%	0.4952	0.4957	1.7%	0.4949	0.5025	1.3%
Waterfall($\kappa = 0.5$)	-	0.6041	4.0%	-	0.5833	1.9%	-	0.5981	5.0%
Waterfall($\kappa = 1$)	-	0.7432	11.8%	-	0.6519	3.1%	-	0.7283	13.2%
Ours	0.8206	0.9310	67.4%	0.7137	0.8649	**43.9%**	0.7388	0.8616	**53.7%**
Ours(advt)	**0.9003**	**0.9709**	**78.1%**	**0.8487**	**0.9239**	36.8%	**0.8648**	**0.9546**	45.7%

Experiment Setup

- Models: TinyLlama-1.1b
- Training Datasets: C4
- Metrics: Bit Acc, Bit Num, AUC, TPR@FPR, Similarity
- Hyper-parameters: $\lambda_w = 0.1$, $\lambda_s = 1.0$, $\lambda_k = 0.02$, $\lambda_{JS} = 1.0$



Robust Multi-bit Text Watermark with LLM-based Paraphrasers（Hang Li et al. ICLR 2025 Workshop）

Thanks !