MoE 路由水印的信息论形式化定义
Information-Theoretic Formalization of MoE Routing Watermarking

Yunhao Yilong Qingxiao

Abstract

本文提出了混合专家（MoE）架构中路由水印的信息论形式化框架，聚焦于模型输入空间的语义层面释义攻击。我们建立了基于参数扰动的信道模型：编码器将水印消息映射为模型参数的结构化扰动 $\Delta\theta$，信道是数据分布 $p(x)$ 对这些扰动的响应，输出是验证者在测试集上观测到的激活模式统计量。我们量化了激活模式（组合码）的编码容量，并揭示了其在语义层面释义攻击下的脆弱性：水印可能对触发器输入过拟合，只学习表层 token 关联而非语义关联。在约束模型性能和隐蔽性的条件下，我们建立了二维率失真（Rate-Distortion）框架，同时考虑性能失真 D_{perf} 和统计失真 D_{detect}。我们提出了基于约束优化的编码器算法，引入语义一致性约束来对抗输入空间的释义攻击，通过带约束的微调过程实现水印嵌入。我们定义了语义鲁棒性 $\mathcal{R}_{\text{input}}$ 来量化水印对输入空间规避攻击的抵抗能力，并建立了四维帕累托前沿。在验证过程中，我们将问题形式化为复合假设检验，采用广义似然比检验（GLRT）或序贯概率比检验（SPRT）构建验证器，并基于 Chernoff/Bhattacharyya 上界间接推导了检测质量的 AUC 下界。最后，我们定义了综合性能指标和扩展的帕累托前沿，为 MoE 路由水印的设计和评估提供了严格的理论基础。

1 引言

混合专家（Mixture-of-Experts, MoE）架构通过稀疏激活机制实现了大型神经网络的高效设计。在 MoE 架构中，路由器（Router）负责为每个输入令牌选择性地激活专家子集，这种路由机制为水印嵌入提供了独特的信息载体。然而，目前缺乏对 MoE 路由水印的理论分析框架，特别是从信息论角度对其容量和检测能力的严格量化。

核心挑战：在实际应用中，水印面临的主要威胁来自模型输入空间的攻击，特别是语义层面的释义攻击。攻击者通过保持语义不变但改变输入表达的方式来规避水印检测，这要求水印系统具备对输入空间变化的鲁棒性。路由器是一个习得的函数 $\mathbf{r} = \text{Router}(x)$，我们通过修改模型参数（如路由器的权重 $\mathbf{w}, b$）或训练过程来间接影响其行为。

本文提出了 MoE 路由水印的信息论形式化定义，聚焦于模型输入空间的语义层面释义攻击。我们建立了基于参数扰动的信道模型，将编码器定义为从水印消息到模型参数扰动的映射，信道是数据分布对这些扰动的响应。我们量化了激活模式（组合码）的编码容量，并深入分析了其在输入空间攻击下的脆弱性。在此基础上，我们建立了二维率失真框架，同时考虑性能失真和统计失真（隐蔽性），并提出了基于约束优化的编码器算法，引入语义一致性约束来对抗输入空间的释义攻击。本文的贡献包括：

- 建立了基于参数扰动的信道模型，明确了编码器、信道和输出的定义，聚焦于模型输入空间的攻击场景
- 量化了激活模式（组合码）的编码容量，这是最稳定的信息维度
- 建立了二维率失真（Rate-Distortion）框架，同时考虑性能失真 D_{perf} 和统计失真 D_{detect}（隐蔽性）
- 提出了基于约束优化的编码器算法，引入语义一致性约束来对抗输入空间的释义攻击，通过带约束的微调过程实现水印嵌入，使用基于间隔的水印损失函数
- 揭示了 I_{pattern} 在输入空间攻击下的脆弱性：水印可能对触发器输入过拟合，只学习表层 token 关联而非语义关联，导致在释义攻击下实际可实现鲁棒容量可能骤降为 0
- 定义了语义鲁棒性 $\mathcal{R}_{\text{input}}$ 来量化水印对输入空间规避攻击的抵抗能力，并建立了四维帕累托前沿
- 构建了基于复合假设检验的验证框架（GLRT/SPRT），强调只使用激活模式特征进行验证
- 深入分析了语义层面释义攻击对水印检测的影响，揭示了过拟合问题并提出了语义一致性约束的应对策略

2 基础符号与系统定义

2.1 MoE 路由机制基础

设 MoE 模型的第 l 层为：

$$\mathrm{MoE}_l(x) = \sum_{i=1}^{n} g_i(x) \cdot E_i(x)$$

其中：

- $x \in \mathbb{R}^d$：输入向量
- $g_i(x)$：第 i 个专家的路由权重（由路由器 Router 产生）
- $E_i(x)$：第 i 个专家的输出
- n：专家总数

路由器输出：

$$\mathbf{r} = \mathrm{Router}(x) = \mathrm{Softmax}(\mathbf{w}^T x + b) \in \Delta^{n-1}$$

其中 Softmax$(\cdot)$ 为 softmax 函数，得到路由分布 $\mathbf{r} = [r_1, r_2, \ldots, r_n]$，位于概率单纯形 $\Delta^{n-1} = \{\mathbf{r} \in \mathbb{R}^n : r_i \geq 0, \sum_{i=1}^n r_i = 1\}$。温度参数 T 用于控制分布的锐度：$\mathrm{Softmax}_T(\mathbf{z})_i = \exp(z_i/T)/\sum_j \exp(z_j/T)$。

2.2 水印系统定义

水印系统 $\mathcal{W} = (\mathcal{E}, \mathcal{V})$ 包括：

- $\mathcal{E}$：嵌入过程（Embedding），将水印消息 m 映射为模型参数的结构化扰动 $\Delta\theta$
- $\mathcal{V}$：验证过程（Verification），从测试集上的观测特征推断模型是否包含水印

信道模型：

- 路由器是一个习得的函数：$\mathbf{r} = \mathrm{Router}(x) = \mathrm{Softmax}(\mathbf{w}^T x + b)$，其输出依赖于输入 x
- 编码器的任务：通过参数扰动 $\Delta\theta$ 在数据分布 $p(x)$ 上使 $\mathcal{S}(x)$ 的统计分布 $p(\mathcal{S}|\theta_{\mathrm{wm}})$ 发生可检测的、偏向消息 m 的偏移
- 信道定义：
  - 输入：参数扰动 $\Delta\theta$
  - 信道：数据分布 $p(x)$ 对 $\Delta\theta$ 的" 探测" 和" 响应"，同一个 $\Delta\theta$ 在不同输入 x_i 上产生不同的 $\mathcal{S}(x_i)$
  - 输出：验证者在测试集 $\mathcal{T} = \{x_1, \ldots, x_N\}$ 上观测到的特征统计量 $\mathbf{f} = [\mathcal{F}(x_1), \ldots, \mathcal{F}(x_N)]$

3 水印信息的编码空间

3.1 路由状态空间

对于输入 x，在第 l 层的路由状态表示为：

$$\mathcal{S}(x) = (\mathbf{r}, \Sigma, \pi) \in \Delta^{n-1} \times 2^{[n]} \times S_k$$

其中：

- $\mathbf{r} \in \Delta^{n-1}$：路由分布向量（位于概率单纯形）
- $\Sigma \subset [n]$：激活的专家集合，满足 $|\Sigma| = k$（top-k 稀疏门控）
- $\pi \in S_k$：激活顺序（排列），表示 Σ 中专家按路由权重降序排列

为避免记号歧义，我们明确区分：Softmax$(\cdot)$ 表示函数，Σ 表示激活集合，π 表示排列。

3.2 可用信息维度

路由状态空间提供了三个主要的信息维度用于编码水印：

维度 **1**：激活模式（组合码）

从 n 个专家中选择 k 个激活，信息容量为：

$$I_{\mathrm{pattern}} = \log_2 \binom{n}{k} = \log_2 \frac{n!}{k!(n-k)!}$$

对于 $n = 128, k = 2$ 的典型配置，使用斯特林近似（Stirling's approximation）：

$$I_{\mathrm{pattern}} \approx 13 \text{ bits}$$

维度 **2**：排列顺序（排列码）

对前 k 个专家的排列，信息容量为：

$$I_{\mathrm{order}} = \log_2(k!)$$

对于 $k = 4$：

$$I_{\mathrm{order}} = \log_2(24) \approx 4.6 \text{ bits}$$

注意：该容量仅在" 前 k 专家可完全排序且可控" 的假设下成立。若路由器含温度缩放或分数权重相近导致排序不稳定，应引入有限分辨率/随机噪声的" 有效容量" 修正。

维度 **3**：权重量化（连续码）

由于 $\mathbf{r} \in \Delta^{n-1}$ 且满足 $\sum_i r_i = 1$，自由度为 $(n-1)$。在 top-k 稀疏门控下，仅对激活的 k 个专家权重进行量化，且满足 $\sum_{i \in \Sigma} r_i = 1$，因此自由度为 $(k-1)$。假设每个路由权重 r_i（$i \in \Sigma$）量化为 b-bit 精度：

$$I_{\mathrm{weight}} \leq (k-1) \cdot b \text{ bits}$$

对于 $k = 4, b = 4$（16 级量化）：

$$I_{\mathrm{weight}} \leq 12 \text{ bits}$$

考虑温度 T 与归一化噪声 σ 的影响，有效容量需乘以系数 $\eta(T,\sigma) \in (0,1)$：

$$I_{\text{weight}}^{\text{eff}} = \eta(T,\sigma) \cdot (k-1) \cdot b$$

总容量上界：

$$\begin{aligned} C_{\max} &= I_{\text{pattern}} + I_{\text{order}} + I_{\text{weight}}^{\text{eff}} \\ &= \log_2 \binom{n}{k} + \log_2(k!) + \eta(T,\sigma) \cdot (k-1) \cdot b \end{aligned} \tag{1}$$

4 信息容量的量化

4.1 可实现容量（Achievable Capacity）与编码本视角

从编码本（Codebook）视角，编码器 $\mathcal{E}$ 将水印消息 m 映射为模型参数的结构化扰动 $\Delta\theta$。信道模型为"参数扰动 $\Delta\theta \to$ 数据分布 $p(x)$ 的响应 $\to$ 路由状态的后验分布 $p(\mathcal{S}(x)|\theta_{\text{wm}}) \to$ 验证统计量 $\mathbf{f}$"。

在约束模型性能和隐蔽性的条件下，可实现的容量定义为：

$$C_{\text{achievable}} = \max_{p(m),p(\Delta\theta|m)} I(m;\mathbf{f})$$

其中 $\mathbf{f} = [\mathcal{F}(x_1),\ldots,\mathcal{F}(x_N)]$ 为测试集 $\mathcal{T}$ 上的特征统计量，$I(\cdot;\cdot)$ 为互信息。

特征选择：排列码 I_{order} 和权重量化 I_{weight} 在语义变化下不稳定。因此，验证者 $\mathcal{V}$ 应该只依赖于激活模式 Σ。信道输出 $\mathbf{f}$ 可以被精简为：在 N 个样本上观测到的激活模式 Σ 的经验分布（或计数向量）。

4.2 率失真（Rate-Distortion）问题

在数据分布 $p(x)$ 上定义失真。率失真函数定义为：

$$\begin{aligned} R(D_{\text{perf}}, D_{\text{detect}}) = &\max_{p(m),p(\Delta\theta|m)} I(m;\mathbf{f}) \\ &\text{s.t.} \quad D_{\text{perf}} \le \epsilon_1, \quad D_{\text{detect}} \le \epsilon_2 \end{aligned} \tag{2}$$

其中失真是一个二维向量 $D = (D_{\text{perf}}, D_{\text{detect}})$：

1. 性能失真 D_{perf}：嵌入水印对模型主要任务造成的损害

$$D_{\text{perf}} = \Delta_{\text{perf}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{watermarked}}$$

2. 统计失真 D_{detect}（可隐蔽性）：水印在路由统计上留下的痕迹，可被量化为干净模型和水印模型在路由输出分布上的 KL 散度

$$D_{\text{detect}} = \mathbb{E}_{x\sim p(x)}\left[D_{\text{KL}}\left(p(\mathcal{S}(x)|\theta_{\text{clean}})||p(\mathcal{S}(x)|\theta_{\text{wm}})\right)\right]$$

失真的双重作用：

- D_{detect} 越小，水印越隐蔽（越难被攻击者发现）
- D_{detect} 越大，水印越可检测（越容易被验证者验证）

信息-性能-隐蔽性权衡：通过扫描超参数（如 λ_{wm}），可以凭经验绘制出 $R(D_{\text{perf}}, D_{\text{detect}})$ 的帕累托前沿，其中：

- X 轴：D_{perf}（模型准确率下降）
- Y 轴：R（水印检测器的统计显著性或 AUC）
- Z 轴：D_{detect}（隐蔽性，越小越好）

5 基于约束优化的编码器算法

5.1 问题的形式化定义

编码器 $\mathcal{E}$ 的任务是将水印消息 m 映射为模型参数扰动 $\Delta\theta$。这可以形式化为一个约束优化问题：

$$\begin{aligned} \max_{\Delta\theta} \quad & \underbrace{\mathbb{E}_{x\sim p(x)}\left[\log p(\Sigma_m|x,\theta_{\text{clean}}+\Delta\theta)\right]}_{\text{水印强度 (R)}} \\ & - \lambda_1 \underbrace{\Delta\mathcal{L}_{\text{task}}(\Delta\theta)}_{\text{性能失真 } (D_{\text{perf}})} - \lambda_2 \underbrace{D_{\text{detect}}(\Delta\theta)}_{\text{隐蔽性失真 } (D_{\text{detect}})} \end{aligned} \tag{3}$$

其中：

- $m \to \Sigma_m$：通过带密钥 K 的哈希函数 $H_K(m)$ 将消息 m 映射到目标激活模式 Σ_m（例如，对于 $n=8, k=2$，共有 $\binom{8}{2}=28$ 种可能的激活模式）
- λ_1, λ_2：拉格朗日乘子，用于在 R、D_{perf} 和 D_{detect} 之间进行权衡，构成帕累托前沿

5.2 实用的优化算法

我们不需要从头开始求解上述问题，可以将其构建为一个带约束的微调（Fine-tuning）过程：

算法：基于约束优化的水印嵌入

1. 目标定义：选择消息 m，通过哈希函数 $H_K(m)$ 生成目标激活模式 Σ_m。
2. 损失函数设计：定义复合损失函数 $\mathcal{L}_{\text{total}}$ 来微调 θ_{clean}：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}}(x) + \lambda_{\text{wm}} \cdot \mathcal{L}_{\text{wm}}(x) + \lambda_{\text{con}} \cdot \mathcal{L}_{\text{con}}(x,x')$$

其中：

- $\mathcal{L}_{\text{task}}$：原始模型任务损失（如交叉熵），确保保持在 $D_{\text{perf}} \leq \epsilon_1$ 的约束内
- λ_{wm}：水印强度超参数，直接控制 R 和 D_{perf} 之间的权衡
- $\mathcal{L}_{\text{wm}}$：水印损失，用于实现 $\max p(\Sigma_m)$ 的目标
- λ_{con}：语义一致性强度超参数，控制 $\mathcal{R}_{\text{input}}$ 和 Δ_{perf} 之间的权衡
- $\mathcal{L}_{\text{con}}$：语义一致性约束，定义为 $\mathcal{L}_{\text{con}}(x, x') = \text{Dist}(\mathcal{S}(x), \mathcal{S}(x'))$，其中 x' 是 x 的语义等价释义，Dist 是路由状态之间的距离度量（如 KL 散度或总变分距离）

3. 水印损失 $\mathcal{L}_{\textbf{wm}}$ 的设计：对于一个输入 x，路由器产生 n 个 logits：$\mathbf{l} = [l_1, \ldots, l_n]$。我们的目标是使目标模式 Σ_m 中的专家成为 Top-k。采用基于间隔（**Margin-based**）的损失：

   - 找到 Σ_m 中的最低分：$l_{\text{min\_target}} = \min_{i \in \Sigma_m} l_i$
   - 找到非 Σ_m 中的最高分：$l_{\text{max\_other}} = \max_{i \notin \Sigma_m} l_i$
   - 损失函数：$\mathcal{L}_{\text{wm}} = \text{ReLU}(l_{\text{max\_other}} - l_{\text{min\_target}} + \text{margin})$

   该损失的含义是：" 要求目标专家（Σ_m 中）的最低分，必须比所有其他专家的最高分还要高出一个 margin。如果这个条件满足，损失为 0；否则，就施加一个惩罚。"

4. 执行编码：使用 $\mathcal{L}_{\text{total}}$ 对 θ_{clean}（或仅仅是路由器参数）进行微调几个（甚至几十个）epoch。

   这个过程自动找到了一个" 折衷" 的 $\Delta\theta$：

   - 它只在那些不会严重损害 $\mathcal{L}_{\text{task}}$ 的输入 x 上" 悄悄地" 推高 Σ_m 的概率
   - 它在" 最容易" 被操纵的输入上嵌入水印，而不是在所有输入上强行嵌入
   - λ_{wm} 的大小直接控制了 R 和 D_{perf} 之间的权衡
   - λ_{con} 的大小直接控制了 $\mathcal{R}_{\text{input}}$ 和 Δ_{perf} 以及 $C_{\text{achievable}}$ 之间的权衡
   - 通过 $\mathcal{L}_{\text{con}}$，模型被迫学习语义层面的特征，而非仅仅依赖表层 token 关联，从而提高了对释义攻击的鲁棒性

5.3 与理论框架的连接

该算法完美地连接回了形式化框架：

- 编码器 $\mathcal{E}$：不再是简单的加法 $\tilde{\mathbf{w}} = \mathbf{w} + \delta\mathbf{b}_{\text{target}}$，而是一个优化过程（即上述的微调算法）
- 可实现容量 $C_{\textbf{achievable}}$：容量 R 不再是理论上的 $\log_2 \binom{n}{k}$，而是由 λ_{wm} 控制的、在 D_{perf} 和 D_{detect} 约束下的实际互信息 $I(m; \mathbf{f})$
- 率失真 $R(D)$：可以通过扫描 $\lambda_{\textbf{wm}}$ 来凭经验绘制出 $R(D_{\text{perf}}, D_{\text{detect}})$ 曲线，完美体现了" 信息-性能-隐蔽性权衡" 的思想

6 语义层面释义攻击的影响

6.1 释义攻击的定义与机制

模型输入空间的攻击：在实际应用中，水印面临的主要威胁来自模型输入空间的攻击，特别是语义层面的释义攻击（Paraphrasing Attack）。这是一个更高级、更隐蔽的威胁：攻击者甚至不需要修改模型参数，只是在使用模型时，通过改变输入的措辞来规避水印的检测。

攻击者目标：规避（Evasion）。

攻击者假设：攻击者拥有一个黑盒或灰盒的水印模型 f_{wm}。他怀疑模型在处理某些特定输入时会表现出异常（例如，激活特定的专家组合 Σ_m）。

攻击者行为：攻击者不直接使用输入 x（例如一个提示：" 请总结一下莎士比亚的《哈姆雷特》"），而是将其释义为 x'（例如：" 用几句话概括《哈姆雷特》剧情"）。

攻击成功的条件：输入 x 能够触发水印统计偏差（$p(\Sigma_m|x) \gg p(\Sigma_m|x, \theta_{\text{clean}})$），而释义后的输入 x' 却不能触发（$p(\Sigma_m|x') \approx p(\Sigma_m|x', \theta_{\text{clean}})$）。

设输入为 x，其语义等价的释义为 x'，满足：

$$\text{Semantic}(x) \approx \text{Semantic}(x')$$

其中 Semantic$(\cdot)$ 表示输入的语义内容。

6.2 释义对路由状态的影响

由于路由器 Router(x) 是一个习得的函数，语义等价的输入 x 和 x' 可能产生不同的路由状态：

$$\mathcal{S}(x) = (\mathbf{r}(x), \Sigma(x), \pi(x)) \neq \mathcal{S}(x') = (\mathbf{r}(x'), \Sigma(x'), \pi(x'))$$

这种差异可能导致：

- 激活模式变化：$\Sigma(x) \neq \Sigma(x')$，即使语义相同，激活的专家集合可能不同
- 路由分布偏移：$\mathbf{r}(x) \neq \mathbf{r}(x')$，路由权重分布发生变化

6.3 水印检测的挑战

在语义层面释义攻击下，水印检测面临以下挑战：

1. 语义不变性要求：水印应该对语义等价的输入保持稳定，即对于语义等价的 x 和 x'，水印检测结果应该一致。

2. 激活模式的语义关联：激活模式 $\Sigma(x)$ 应该与输入的语义内容相关，而非仅仅依赖于表面的词汇或表达方式。这要求水印嵌入过程能够学习到语义层面的特征。

3. 统计分布的稳定性：在测试集上，即使输入经过释义，激活模式的统计分布 $p(\Sigma|\theta_{\text{wm}})$ 应该保持相对稳定，使得验证者能够检测到水印信号。

6.4 应对策略：语义一致性约束

为了应对输入空间的释义攻击，编码器 $\mathcal{E}$ 需要引入语义一致性约束（**Semantic Consistency Constraint**）：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}}(x)+\lambda_{\text{wm}}\cdot\mathcal{L}_{\text{wm}}(x)+\lambda_{\text{con}}\cdot\text{Dist}(\mathcal{S}(x),\mathcal{S}(x'))$$

其中 x' 是 x 的一个释义（或通过数据增强、对抗性输入产生），Dist 是一个距离度量，用于惩罚 x 和 x' 之间路由行为的不一致性。

通过这种方式，我们强迫 $\Delta\theta$ 去学习一个语义鲁棒的路由偏好。这使得水印嵌入的成本更高（D_{perf} 可能更大），但 $\mathcal{R}_{\text{input}}$ 也相应增强了。

权衡关系：

1. $\mathcal{R}_{\textbf{input}}$ **vs** $\Delta_{\textbf{perf}}$（语义鲁棒性 **vs** 性能）：
   - 为了提高 $\mathcal{R}_{\text{input}}$，我们必须提高 λ_{con}。
   - 但这迫使模型在语义相似的输入上（x 和 x'）给出一致的路由行为。
   - 这是一个非常强的约束，它会干扰模型为 $\mathcal{L}_{\text{task}}$ 所做的正常优化，从而不可避免地导致更大的性能下降 $\Delta_{\textbf{perf}}$。

2. $\mathcal{R}_{\textbf{input}}$ **vs** $C_{\textbf{achievable}}$（语义鲁棒性 **vs** 容量）：
   - $\mathcal{L}_{\text{con}}$ 约束越多，模型能够" 藏匿" 水印的" 自由空间" 就越少。
   - 必须在整个语义簇上嵌入水印，而非仅针对特定触发词。
   - 这使得嵌入单个比特（例如 Σ_m）的难度急剧增加，因此可实现容量 $C_{\textbf{achievable}}$ 会显著下降。

帕累托前沿：编码器必须在四维空间（$C_{\text{achievable}}$, D_{perf}, D_{detect}, $\mathcal{R}_{\text{param}}$, $\mathcal{R}_{\text{input}}$）中寻找权衡。

7 验证过程与检测能力

7.1 假设检验框架（复合假设）

验证问题可形式化为复合假设检验：

- H_0：模型未被水印化（原始模型），参数 $\theta_0 \in \Theta_0$ 未知
- H_1：模型被正确的水印化，参数 $\theta_1 \in \Theta_1$ 未知

由于参数未知，H_0 和 H_1 均为复合假设。给定测试集 $\mathcal{T} = \{x_1, \ldots, x_N\}$，提取特征向量：

$$\mathbf{f} = [\mathcal{F}(x_1), \ldots, \mathcal{F}(x_N)]$$

特征选择：排列码 I_{order} 和权重量化 I_{weight} 在语义变化下不稳定。因此，验证者 $\mathcal{V}$ 应该只依赖于激活模式 Σ。特征提取函数 $\mathcal{F}$ 定义为：

$$\mathcal{F}(x) = \Sigma(x)$$

即只提取激活的专家集合。特征向量为：

$$\mathbf{f} = (\Sigma_1, \ldots, \Sigma_N)$$

信道输出：验证者观测到的信道输出 Y 是在 N 个样本上观测到的激活模式 Σ 的经验分布（或计数向量）。对于 n 个专家、k 个激活的配置，共有 $\binom{n}{k}$ 种可能的激活模式，因此 $\mathbf{f}$ 可以表示为长度为 $\binom{n}{k}$ 的计数向量，其中每个元素表示对应激活模式在测试集上的出现次数。

7.2 广义似然比检验（**GLRT**）与序贯概率比检验（**SPRT**）

由于 H_0 和 H_1 为复合假设，采用广义似然比检验（**GLRT**）以消去未知参数：

$$\Lambda_{\text{GLRT}}(\mathbf{f}) = \frac{\max_{\theta_1\in\Theta_1} p(\mathbf{f}|\theta_1)}{\max_{\theta_0\in\Theta_0} p(\mathbf{f}|\theta_0)} \gtrless \tau$$

或采用序贯概率比检验（**SPRT**）以降低测试样本开销：

$$\Lambda_{\text{SPRT}}(\mathbf{f}_t) = \prod_{i=1}^{t} \frac{p(\mathbf{f}_i|H_1)}{p(\mathbf{f}_i|H_0)} \gtrless \tau$$

其中 t 为当前样本数，SPRT 可在达到决策阈值时提前终止。

检测功率（功效）定义为：

$$\beta = \inf_{\theta_1\in\Theta_1} P(\Lambda > \tau|H_1, \theta_1)$$

假阳性率（False Positive Rate）：

$$\alpha = \sup_{\theta_0\in\Theta_0} P(\Lambda > \tau|H_0, \theta_0)$$

7.3 ROC 曲线与 AUC 下界（基于错判率上界）

检测质量由接收者操作特征（ROC）曲线量化：

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$$

对于信息论上界，我们采用 **Chernoff/Bhattacharyya** 上界先给出错判率或 **Bayes** 风险的指数型上界，再间接推导对 AUC 的保守下界。

Bhattacharyya 距离：

$$D_B = -\ln \int \sqrt{p(\mathbf{f}|H_0)p(\mathbf{f}|H_1)}\, d\mathbf{f}$$

对于高斯分布族，错判率上界为：

$$P_{\text{error}} \leq \exp(-D_B)$$

进而可推导 AUC 的保守下界（需注明适用分布族，如高斯族）：

$$\text{AUC} \geq 1 - \exp(-D_B) - \alpha$$

其中 α 为假阳性率上界。

8 综合性能指标

8.1 水印质量函数与帕累托前沿

质量定义：聚焦于模型输入空间的攻击，我们将鲁棒性拆分为两个独立且相互竞争的指标：

1. $\mathcal{R}_{\textbf{param}}$（参数鲁棒性）：对抗微调、蒸馏等参数空间攻击。

2. $\mathcal{R}_{\textbf{input}}$（语义鲁棒性）：对抗释义、同义词替换等输入空间规避攻击。

综合定义水印系统的质量为四维指标：

$$\mathcal{Q} = (C_{\text{achievable}}, \mathcal{R}_{\text{param}}, \mathcal{R}_{\text{input}}, \text{AUC})$$

其中：

- $C_{\text{achievable}}$：可实现容量
- $\mathcal{R}_{\text{param}}$：参数鲁棒性（对抗参数修改）
- $\mathcal{R}_{\text{input}}$：语义鲁棒性（对抗输入空间规避）
- AUC：检测 AUC

多目标优化：编码器 $\mathcal{E}$（水印嵌入算法）在四维空间中寻找权衡点：

$$\max\{C_{\text{achievable}}, \mathcal{R}_{\text{param}}, \mathcal{R}_{\text{input}}, \text{AUC}\}$$

$$\text{s.t.} \quad \Delta_{\text{perf}} \leq \epsilon, \quad \alpha \leq \alpha_{\max}$$

帕累托前沿定义为非被支配的解集合，即不存在其他解在所有目标上都不劣于当前解且至少在一个目标上更优。

关键权衡关系：

1. $\mathcal{R}_{\textbf{input}}$ **vs** $\Delta_{\textbf{perf}}$（语义鲁棒性 **vs** 性能）：为了提高 $\mathcal{R}_{\text{input}}$，必须提高 λ_{con}，但这会干扰模型为 $\mathcal{L}_{\text{task}}$ 所做的正常优化，导致更大的性能下降 Δ_{perf}。

2. $\mathcal{R}_{\textbf{input}}$ **vs** $C_{\textbf{achievable}}$（语义鲁棒性 **vs** 容量）：$\mathcal{L}_{\text{con}}$ 约束越多，模型能够" 藏匿" 水印的" 自由空间" 就越少，不能再简单地将水印" 过拟合" 到特定触发词上，必须在整个语义簇上嵌入水印，使得嵌入单个比特的难度急剧增加，因此可实现容量 $C_{\text{achievable}}$ 会显著下降。

权重设定：质量函数的加权形式 $Q_{\text{normalized}} = \alpha C_n + \beta R_{\text{param},n} + \gamma R_{\text{input},n} + \delta A_n$（其中 $\alpha + \beta + \gamma + \delta = 1$）仅用于政策选择（根据具体应用需求选择折中点），而非理论结论。权重来源与任务依赖需明确：不同数据集/任务（如生成式 vs 分类式 MoE）可能需要不同的权重配置。建议以 **Pareto** 前沿为主图呈现不加权的解集，再由应用需求选择折中点。

与开放问题的连接：该框架回应了如何构造" 稳健验证器" 以处理" 域漂移" 的开放问题。" 文本释义" 正是" 对抗性域漂移" 的一种形式。通过引入 $\mathcal{L}_{\text{con}}$ 作为实现 $\mathcal{R}_{\text{input}}$ 的机制，我们提供了一个具体的工程路径来探索四维帕累托前沿。

9 数学推导示例

9.1 示例：基于激活模式的水印嵌入

假设设计仅基于激活模式 Σ 进行水印，使用 $n = 8$ 个专家、$k = 2$ 个激活的配置：

信息来源：从 $n = 8$ 个专家中选择 $k = 2$ 个激活，共有 $\binom{8}{2} = 28$ 种可能的激活模式

$$I_{\text{pattern}} = \log_2 \binom{8}{2} = \log_2 28 \approx 4.8 \text{ bits}$$

消息映射：通过带密钥 K 的哈希函数 $H_K(m)$ 将消息 m（例如"U-C-Berkeley"）映射到目标激活模式 Σ_m（例如，$\Sigma_m = \{$专家 3, 专家 7$\}$）。

编码方案：使用基于约束优化的微调过程，而非简单的偏置向量加法。定义复合损失函数：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{wm}} \cdot \mathcal{L}_{\text{wm}}$$

其中水印损失 $\mathcal{L}_{\text{wm}}$ 采用基于间隔的损失：

$$\mathcal{L}_{\text{wm}} = \text{ReLU}\left(\max_{i \notin \Sigma_m} l_i - \min_{i \in \Sigma_m} l_i + \text{margin}\right)$$

该损失要求目标专家（Σ_m 中）的最低 logit 必须比所有其他专家的最高 logit 还要高出一个 margin。

性能约束：在数据分布 $p(x)$ 上的性能失真和统计失真

$$D_{\text{perf}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{watermarked}} \leq \epsilon_1$$

$D_{\text{detect}} = \mathbb{E}_{x\sim p(x)}\left[D_{\text{KL}}\left(p(\mathcal{S}(x)|\theta_{\text{clean}})||p(\mathcal{S}(x)|\theta_{\text{wm}})\right)\right] \le \epsilon_2$

可实现容量：通过扫描 λ_{wm}，可以凭经验测量实际互信息 $I(m;\mathbf{f})$，其中 $\mathbf{f}$ 是测试集上激活模式的经验分布。实际容量 $C_{\text{achievable}}$ 通常小于理论容量 I_{pattern}，因为需要在性能失真和统计失真之间进行权衡。

检测能力：基于 Pinsker 不等式，若水印模型和干净模型的路由分布满足 $D_{\text{KL}}(p_{\text{wm}}||p_{\text{clean}}) \ge \delta$，则验证成功率下界为：

$$P_{\text{detect}} \ge 1 - \exp\left(-\frac{\delta}{2}\right) - \alpha$$

其中 α 为假阳性率上界。在语义层面释义攻击下，需要考虑语义等价输入对分布的影响。

10 开放的理论问题

尽管本文建立了 MoE 路由水印的信息论框架，但仍存在一些开放的理论问题：

1. 稀疏门控与单纯形容量的精确化：在 top-k 稀疏门控与概率单纯形约束下，路由水印的有效自由度与容量是多少？如何用类型法（method of types）为离散部分给出可靠编码区与错误指数？

2. 嵌入-性能的率失真闭环：在给定性能降级门限下，能否得到显式的容量上/下界？如何将性能损失上界映射为对路由分布的散度球约束（KL 或 Rényi 散度）？

3. 复合假设检验的稳健验证器：如何构造稳健最优的验证器以处理语义层面的释义攻击？如何比较以 Σ、π、$\mathbf{r}_\Sigma$ 为特征的不同充分性与功效？

4. 隐蔽性（**Steganographic Security**）与可检性下界：路由分布的统计隐蔽性如何量化？如何定义对手的检测器族并给出最小可辨性下界？

5. 语义层面的稳定性与泛化能力：如何设计水印嵌入算法，使得激活模式对语义等价的输入保持稳定？如何量化语义变化对路由状态分布的影响？如何避免水印对触发器输入的过拟合，确保学习到语义概念关联而非表层 token 关联？

6. 四维帕累托前沿的探索：如何在更高维度的空间（$C_{\text{achievable}}$, $\mathcal{R}_{\text{param}}$, $\mathcal{R}_{\text{input}}$, AUC）中寻找最优权衡点？如何量化 $\mathcal{R}_{\text{input}}$ 与 Δ_{perf} 以及 $\mathcal{R}_{\text{input}}$ 与 $C_{\text{achievable}}$ 之间的权衡关系？

11 结论

本文提出了 MoE 路由水印的信息论形式化框架，聚焦于模型输入空间的语义层面释义攻击，为路由水印的设计和评估提供了严格的理论基础。我们建立了基于参数扰动的信道模型，明确了编码器、信道和输出的定义。我们量化了激活模式（组合码）的编码容量，并揭示了其在输入空间攻击下的脆弱性：水印可能对触发器输入过拟合，只学习表层 token 关联而非语义关联，导致实际可实现鲁棒容量可能骤降为 0。我们建立了二维率失真框架，同时考虑性能失真 D_{perf} 和统计失真 D_{detect}（隐蔽性），并提出了基于约束优化的编码器算法，引入语义一致性约束来对抗输入空间的释义攻击，通过带约束的微调过程实现水印嵌入。我们定义了语义鲁棒性 $\mathcal{R}_{\text{input}}$ 来量化水印对输入空间规避攻击的抵抗能力，并建立了四维帕累托前沿，揭示了 $\mathcal{R}_{\text{input}}$ 与 Δ_{perf} 以及 $\mathcal{R}_{\text{input}}$ 与 $C_{\text{achievable}}$ 之间的关键权衡关系。在验证过程中，我们构建了基于复合假设检验的框架（GLRT/SPRT），强调只使用激活模式特征进行验证，并基于错判率上界间接推导了检测质量的 AUC 下界。这些理论结果为 MoE 路由水印的实践应用提供了指导，并为未来的研究指明了方向，包括稀疏门控与单纯形容量的精确化、嵌入-性能-隐蔽性的率失真闭环、复合假设检验的稳健验证器、隐蔽性分析、语义层面的稳定性与泛化能力以及四维帕累托前沿的探索等开放问题。