混合专家 (MoE) 架构中基于专家权重与路由信息的隐式水印机制分析

作者姓名
所属机构

Abstract

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。本报告分析了 MoE 架构中两种隐式水印方法：基于参数的水印（通过专家权重嵌入）和基于行为的水印（通过路由信息嵌入）。我们探讨了这两种方法的核心机制、鲁棒性特征以及面临的攻击向量，包括路由干扰、专家重训练、剪枝和蒸馏攻击。通过分析，我们提出了一种双重编码混合水印策略，该策略能够同时利用 MoE 的控制平面和数据平面，提供分层防御机制。实验表明，精心设计的混合水印在模型转换后仍能实现 60-80% 的信号保留率，显著优于单一方法。本研究为 MoE 模型的知识产权保护和溯源提供了新的技术路径。

1 引言：MoE 范式的二元性作为模型溯源的新向量

1.1 MoE 的架构分叉

混合专家 (Mixture-of-Experts, MoE) 架构代表了大型神经网络设计中的一次范式转变，其通过稀疏激活来强调效率 [1]。与所有参数均被激活的密集模型不同，MoE 采用一个路由器 (router) 来为每个输入令牌 (token) 选择性地激活一个” 专家”（即专业化的子模型）子集 [1]。

这种架构（例如在 Switch Transformers 和 Mixtral 模型中普及）的核心创新在于它将模型清晰地分叉 (bifurcation) 为两个截然不同的平面：

1. ” 控制平面” **(Control Plane):** 由门控网络（Gating Network）或路由器构成，负责决策，即决定哪些专家处理哪些令牌 [1]。

2. ” 数据平面” **(Data Plane):** 由众多专家子模型构成，负责实际的计算和信号处理 [1]。

与密集模型不同，MoE 的稀疏激活（通常通过 Top-K 门控实现）为信息嵌入创造了两个独特的轨迹 (loci)。水印可以存在于路由器的行为中（一种” 基于行为” 的水印），也可以存在于专家的参数中（一种” 基于参数” 的水印）。这种架构的二元性是本报告所探讨的隐式水印方法的中心论点 [1]。

1.2 企业需求与 MoE 特定水印的驱动力

企业在部署基于 MoE 的大型语言模型时，面临着日益增长的问责和治理压力。对水印技术的核心需求包括：

- 模型可追溯性（出处）：验证内容是否由特定模型生成，这对于打击深度伪造或追踪未授权的衍生模型至关重要 [1]。

- 合规性与许可证明：验证付费 API 的使用或确保模型未被用于违反服务条款的用途（例如，在专有数据上进行微调） [1]。

- 知识产权 **(IP)** 保护：保护高昂训练成本所代表的智力资产 [2]。

理想的水印解决方案必须满足严格的约束条件：(1) 不可感知性，避免在输出中产生人为痕迹；(2) 鲁棒性，能够抵抗对抗性攻击，如输出转述 (paraphrasing)、模型微调或剪枝；(3) 高性能，对模型的推理速度或准确性（如困惑度）的影响可忽略不计（例如，低于 1% 的性能下降） [1]。

研究表明，MoE 的模块化特性可能允许比密集模型更轻量级的水印嵌入。例如，MoE 中的路由修改或专家子集扰动可能仅导致低于 0.5% 的准确率下降，而密集模型中的全局参数更改可能导致 1-2% 的性能损失 [1]。因此，MoE 独特的架构需要定制化的水印解决方案，因为适用于密集模型的方法可能并非最佳或不兼容。

2 基于参数的隐式水印：通过专家权重信息嵌入

本节探讨利用 MoE 的” 数据平面”——即专家子模型的权重——来嵌入隐式水印。

2.1 核心机制：” 修改专家子集，但不改变路由器”

基于参数的水印方法在 MoE 中的核心机制被定义为” 修改专家子集，但不改变路由器” (Modify expert subsets without router changes) [1]。

这意味着水印信号是通过对部分或全部专家子模型的权重进行微小扰动 (perturbation) 来编码的 [1]。

该方法明确避免了对路由器逻辑的任何更改 [1]。路由器的行为（即 Top-K 选择）保持正常，但当它确实选择了一个被植入水印的专家时，该专家的计算（及其对最终 logits 的贡献）会受到其权重中嵌入的秘密信号的微妙偏置影响。

2.2 方法论背景：白盒水印

此方法是" 白盒水印" (white-box watermarking) 的一种形式。信号被嵌入到模型的内部参数（权重）中，因此验证水印的存在需要（至少在某种程度上）访问这些内部参数 [3]。

仅仅进行简单的" 权重扰动" [1] 可能不足以抵抗攻击。更先进的实现方式借鉴了相关研究，以提高鲁棒性：

1. 纠错码 **(Error Correction Codes, ECCs):** 在嵌入之前，可以使用 ECCs 对水印信号进行编码。这确保了即使在模型受到攻击（如微调）导致部分水印信息损坏后，验证者仍能可靠地提取和恢复原始信号 [2]。
2. 权重置换 **(Weight Permutations):** 使用秘密密钥来定义特定专家层内权重矩阵的特定置换（排列顺序）作为水印载体 [2]。这种结构性而非数值性的改变可能更难通过常规的权重正则化或微调来消除。
3. 基于后门的鲁棒公式**:** 虽然与水印不同，但鲁棒后门嵌入的原则（例如 [4] 中提出的" 最小-最大化公式"）可以被借鉴。这些技术旨在寻找对参数变化（如微调）具有弹性的参数空间，从而可以用于保护嵌入在权重中的水印信号 [4]。

2.3 鲁棒性概况与局限性

基于参数的水印方法具有独特的优势和劣势：

- 优势**:** " 对输出编辑具有高鲁棒性" (High robustness to output editing) [1]。由于水印存在于模型的参数中，而不是其输出文本中，因此它完全不受所有输出后处理攻击（如转述、同义词替换或文本编辑）的影响 [1]。
- 关键缺陷**:** " 易受微调或剪枝攻击" (Vulnerable to fine-tuning or pruning) [1]。
  - 微调 **(Fine-tuning):** 即使是对模型进行短暂的再训练，也可能通过梯度下降更新权重，从而覆盖或" 冲刷" 掉嵌入在权重中的微妙扰动，导致信号丢失 [4]。
  - 剪枝 **(Pruning):** 如果承载水印的专家被剪枝算法（旨在移除不活跃或冗余的专家）视为" 不活跃"，那么这些专家（及其携带的水印）可能会被完全删除，导致水印永久丢失 [1]。

3 基于行为的隐式水印：利用专家路由信息

本节探讨利用 MoE 的" 控制平面"——即路由器的决策过程——来嵌入隐式水印。

3.1 核心机制：" 路由序列作为水印载体"

这种方法的核心思想是将专家激活的序列 (sequence) 本身作为水印信号 [1]。水印不再是静态的权重扰动，而是一种动态的、基于输入的行为。[1] 中描述了两种概念性方法：

1. 隐式（统计性）**:** 通过微调路由器，在" 专家使用上创造统计偏差" (statistical deviations in expert usage) [1]。例如，一个带水印的模型在处理中性提示时，可能表现出对专家 3 的异常高使用率。验证者可以通过运行大量查询并进行统计分析，来检测这种在非水印模型中统计上极不可能发生的异常模式。
2. 显式（密钥驱动）**:** " 注入与秘密密钥绑定的路由器偏置" (Injecting router biases tied to a secret key) [1]。在这种模式下，一个特定的、预定义的输入（" 密钥" 或" 触发器"）会激活一个隐藏规则，迫使路由器做出一个预定义的、非自然的路由选择（例如，忽略 Top-K 逻辑，强制选择专家 7）。

3.2 信息论形式化框架

为了从理论上理解路由水印的容量和鲁棒性，我们引入信息论形式化框架 [**?**]。该框架为路由水印提供了严格的理论基础。

3.2.1 路由状态空间与编码容量

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 $x \in \mathbb{R}^d$ 为输入向量，$g_i(x)$ 为第 i 个专家的路由权重（由路由器产生），$E_i(x)$ 为第 i 个专家的输出，n 为专家总数。路由器输出为：

$$\mathbf{r} = \mathrm{Router}(x) = \sigma(\mathbf{w}^T x + b) \in \mathbb{R}^n$$

其中 σ 为 softmax 函数，得到路由分布 $\mathbf{r} = [r_1, r_2, \ldots, r_n]$。

对于输入 x，在第 l 层的路由状态表示为：

$$\mathcal{S}(x) = (\mathbf{r}, \sigma, \tau) \in \mathbb{R}^n \times \mathbb{N} \times \{0,1\}^*$$

其中 $\mathbf{r}$ 为路由分布向量，$\sigma = \{i : r_i > \epsilon_{\text{th}}\}$ 为激活的专家集合，$\tau = \mathrm{argsort}(\mathbf{r})$ 为激活序列（按权重排序）。

路由状态空间提供了三个主要的信息维度用于编码水印：

1. 激活模式维度：从 n 个专家中选择 k 个激活，信息容量为：

$$I_{\text{pattern}} = \log_2 \binom{n}{k}$$

对于 $n = 128, k = 2$ 的典型配置，$I_{\text{pattern}} \approx 13$ bits。

2. 排列顺序维度：对前 k 个专家的排列，信息容量为：

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

对于 $k = 4$，$I_{\text{order}} = \log_2(24) \approx 4.6$ bits。

3. 权重量化维度：假设每个路由权重 r_i 量化为 b-bit 精度：

$$I_{\text{weight}} = n \cdot b \text{ bits}$$

对于 $n = 128, b = 4$（16 级量化），$I_{\text{weight}} = 512$ bits。

总容量上界为：

$$C_{\max} = I_{\text{pattern}} + I_{\text{order}} + I_{\text{weight}}$$

3.2.2 可实现容量与性能权衡

在约束模型性能的条件下，可实现的容量定义为：

$$C_{\text{achievable}} = \max_{\delta \in (0,\epsilon)} I(\mathbf{m}; \mathcal{S}(x)|x)$$

其中 $\mathbf{m}$ 为待嵌入的水印信息，δ 为嵌入强度参数，满足模型性能下降 $\leq \epsilon$，$I(\cdot;\cdot|\cdot)$ 为条件互信息。具体形式为：

$$C_{\text{achievable}} = \int_{\mathcal{X}} p(x) \max_{\delta} I(\mathbf{m}; \mathbf{r}_\delta(x))\, dx$$

其中 $\mathbf{r}_\delta(x)$ 为被修改的路由分布。
信息-性能权衡函数 $\mathcal{T}$ 定义为：

$$\mathcal{T}(\epsilon) = \{C : \exists \delta \text{ s.t. } \Delta_{\text{perf}} \leq \epsilon \text{ and } I(\mathbf{m}; \mathcal{S}(x)) = C\}$$

其中 $\Delta_{\text{perf}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{watermarked}}$ 表示性能下降。

3.2.3 鲁棒性的信息论下界

设攻击为随机变换 $\mathcal{A} : \mathcal{S} \to \mathcal{S}'$（如路由器微调或输入扰动），水印的鲁棒性定义为：

$$\text{Robustness} = \frac{\sum_{\mathcal{A} \in \mathcal{A}_{\text{adv}}} \mathbb{1}[\mathcal{V}(f_{\text{watermarked}}, \mathcal{A})]}{|\mathcal{A}_{\text{adv}}|}$$

其中 $\mathcal{V}(f, \mathcal{A}) = 1$ 表示在攻击 $\mathcal{A}$ 后仍能成功验证。
对于独立同分布的攻击，水印的鲁棒性下界由 Fano 不等式给出：

$$P_{\text{error}} \geq 1 - \frac{I(\mathbf{m}; \mathcal{S}'(x)) + 1}{\log_2 |\mathcal{M}|}$$

其中 $\mathcal{S}'(x) = \mathcal{A}(\mathcal{S}(x))$ 为攻击后的路由状态，$|\mathcal{M}|$ 为水印消息空间大小。该下界表明，在攻击后保留的互信息 $I(\mathbf{m}; \mathcal{S}'(x))$ 越大，检测错误率越低，鲁棒性越强。
对于特定的攻击类型，可以进一步量化鲁棒性。例如，对于路由器重训练攻击，考虑 Kullback-Leibler 散度：

$$D_{\text{KL}}(\mathbf{r} || \mathbf{r}') = \sum_i r_i \log \frac{r_i}{r'_i} \leq \beta$$

鲁棒性与 β 的关系为：

$$\text{Rob}_{\text{retrain}} = 1 - \exp(-\lambda D_{\text{KL}})$$

其中 λ 为编码的纠错码强度参数。

3.2.4 验证过程与检测能力

验证问题可形式化为二元假设检验：

- H_0：模型未被水印化（原始模型）
- H_1：模型被正确的水印化

给定测试集 $\mathcal{T} = \{x_1, \ldots, x_N\}$，提取特征向量 $\mathbf{f} = [\mathcal{F}(x_1), \ldots, \mathcal{F}(x_N)]$，其中 $\mathcal{F}$ 为特征提取函数（如路由激活模式）。
采用 Neyman-Pearson 引理，最优检测器为：

$$\Lambda(\mathbf{f}) = \frac{p(\mathbf{f}|H_1)}{p(\mathbf{f}|H_0)} \gtrless \tau$$

检测功率定义为：

$$\beta = P(\text{detect}|H_1 \text{ true}) = P(\Lambda > \tau | H_1)$$

假阳性率为：

$$\alpha = P(\text{detect}|H_0 \text{ true})$$

检测质量由接收者操作特征（ROC）曲线量化，其 AUC 下界由 Bhattacharyya 距离给出：

$$\text{AUC} \geq \Phi\left(\frac{D_B}{2}\right)$$

其中 Φ 为标准正态 CDF，D_B 为 Bhattacharyya 距离：

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

3.3 高级案例研究：将"BadSwitch"（后门）改编为水印

[1] 指出，显式（密钥驱动）的路由水印是一个" 新兴的想法"，缺乏广泛的实证验证 [1]。然而，最近在 AI 安全领域的研究为实现这一理论提供了具体的蓝图。

一篇题为"Who Speaks for the Trigger? Dynamic Expert Routing in Backdoored Mixture-of-Experts Transformers" (arXiv:2510.13462) 的论文，详细介绍了一种名为"BadSwitch" 的后门攻击机制，该机制精确地利用了 MoE 路由的漏洞 [5]。

通过分析可以发现，后门攻击的机制和密钥驱动的水印机制在功能上是等同的：

1. 一个（密钥驱动的）水印的目标是" 注入与秘密密钥绑定的路由器偏置" [1]。
2. BadSwitch（后门）的目标是" 集成任务耦合的动态触发器优化"，并" 将 Top-K 门控机制约束到... 目标专家" [5]。

在这两种情况下，" 后门触发器" 等同于" 秘密密钥"，" 约束门控" 等同于" 注入路由器偏置"。因此，BadSwitch 论文 [5] 为 [1] 中理论化的行为水印提供了一个具体的、经过实证验证的实施蓝图。

水印实施蓝图（改编自 **BadSwitch**）：

1. 水印密钥（触发器）：优化一个特定的、非显而易见的令牌序列（" 密钥"），这类似于 BadSwitch 的" 动态触发器" [5]。
2. 信号路径（敏感专家）：在训练期间，使用" 敏感性引导的 Top-S 专家追踪机制" [5] 来识别哪些专家对该密钥最为敏感和具影响力。然后，选择一个独特的、在正常情况下极不可能被激活的专家路径（例如，专家 7 $\rightarrow$ 专家 2 $\rightarrow$ 专家 14）作为水印信号。
3. 嵌入（约束门控）：对路由器的门控机制 [6] 进行微调，使其仅在检测到水印密钥时，强制选择这个特定的、非自然的专家路径（即约束 Top-K 逻辑）。
4. 验证（溯源）：验证者（例如，模型所有者）使用秘密密钥发起一个黑盒查询。通过（可能的）白盒访问激活日志，或通过观察由该特定路径产生的可预测的（尽管微妙的）输出偏差，验证者可以确认模型是否采取了预定的异常路径，从而证明其出处 [1]。

3.4 鲁棒性概况与局限性

- 优势：具有" 低性能影响"，并且在不知道触发器（密钥）的情况下" 难以检测或移除" [1]。信号是行为性的，而非参数性的，使其对标准的权重分析（如寻找扰动）隐形。
- 关键缺陷：在许多情况下，" 需要访问激活日志进行验证" [1]，这对于黑盒 API 验证是一个重大障碍。更重要的是，它极易受到" 路由干扰" (Router Disruption) 攻击 [1]。

4 MoE 水印的关键漏洞和攻击面

MoE 架构在提供新水印载体的同时，也引入了独特的攻击向量。

4.1 控制平面攻击：路由干扰 (Router Disruption)

深入研究" 路由干扰"（Router Disruption）这一控制平面攻击，可以发现这是一种专门针对 MoE 架构中门控网络（Gating Network）或路由器的攻击向量 [1]。这种攻击利用了 MoE 架构的独特结构，对基于行为的水印构成了严重威胁。

4.1.1 攻击机制与漏洞

" 路由干扰" 攻击的核心机制是利用了路由器（Router）相对于庞大的专家（Experts）网络而言，通常具有" 轻量级" (lightweight) 的特性 [1]。攻击者可以利用这一特性，通过以下方式发起攻击：

1. 重新训练 **(Retraining):** 攻击者可以对路由器的门控网络进行微调或完整的重新训练。由于路由器参数量相对较少，这种攻击的成本远低于重训练整个模型 [1]。
2. 注入噪声 **(Noising):** 攻击者可以在路由决策过程中注入噪声，干扰其正常的令牌分配逻辑 [1]。
3. 改变路由权重 **(Altering Routing Weights):** 相关的对抗性攻击研究也证实，存在专门针对路由器的攻击，其目的就是" 改变路由权重"，同时不影响专家的输出 [1]。

4.1.2 对水印的影响

这种攻击的直接目标是基于" 行为"（Behavior-Based）或" 路由聚焦"（Router-Focused）的隐式水印 [1]。

- 消除控制平面信号 **(Erases Control-Plane Signals):** 路由干扰攻击的最终影响是" 消除控制平面信号" [1]。
- 破坏水印载体：如果水印依赖于特定的路由序列（例如，由秘密密钥触发的特定专家路径）或依赖于" 专家使用上的统计偏差" 来编码信号，那么对路由器的重训练或干扰会彻底破坏这个载体，导致水印失效 [1]。

4.1.3 缓解策略与防御机制

针对路由干扰攻击，研究中提出了几种防御策略，其复杂度和鲁棒性各不相同：

A. 策略一：加密哈希与冗余 **(Cryptographic Hashing & Redundancy)**

这是文献中提到的主要缓解策略，旨在加固路由逻辑，使其难以被轻易修改：

- 策略：" 在路由中使用加密哈希" (Use cryptographic hashing in routing) [1]。
- 潜在机制：尽管具体的实现机制未被详细阐述 [1]，但相关概念（如" 显式水印"）提到可以" 将一个密钥哈希嵌入到路由决策中" (embedding a keyed hash into routing decisions) [1]。这表明，路由决策可能不仅仅依赖于可训练的权重，还可能依赖于一个基于秘密密钥和输入令牌计算出的、不可更改的哈希值，从而抵抗微调。
- 有效性：这种方法并非绝对可靠，但其有效性依赖于冗余设计。相关研究证据表明，在门控网络中结合使用这种策略和" 冗余" (redundancy) 设计（例如，可能将信号分散到多个路由层或决策点），可以实现"50-70% 的信号恢复" (50-70% signal recovery) [1]。

B. 策略二：鲁棒性感知路由 **(Robustness-Aware Routing)**

这是一种更先进的主动防御机制，它使路由器本身具备对抗攻击的能力：

- 策略：构建一个" 鲁棒性感知路由器" (robustness-aware router)，例如在 RGMoE（Robust Graph MoE）框架中提出的那样 [1]。
- 机制：这种路由器经过专门训练，能够" 识别受干扰的模式" (identify the perturbed patterns) [1]。
- 防御行为：当路由器识别出输入可能受到了对抗性攻击（例如，旨在干扰其决策的噪声或扰动）时，它会" 自适应地将每个受干扰的样本路由到相应的鲁棒专家" (adaptively routes each perturbed sample to the corresponding expert that exhibits robustness against that perturbation) [1]。简而言之，路由器学会了在检测到攻击时" 激活仅有鲁棒的专家" (activate only the robust experts)，从而绕过攻击的影响 [1]。

4.2 数据平面攻击：专家级别攻击 (Expert-Level Attacks)

- 机制 **1** (专家重训练)：攻击者可以微调单独的专家，特别是那些他们怀疑携带水印信号的专家 [1]。
- 影响：这会" 改变子模型的行为" [1]，并可能擦除嵌入在专家权重中的基于参数的水印 [1]。
- 机制 **2** (专家剪枝)：攻击者可以移除不活跃或冗余的专家 [1]。
- 影响：如果基于参数的水印被嵌入到少数几个被视为" 不活跃" 的专家中，此攻击将物理上删除携带水印的组件 [1]。

4.3 结构性攻击：蒸馏为密集模型 (Distillation to Dense)

- 机制：这是对 MoE 特定水印最严重的威胁。攻击者为了提高部署效率（例如，消除路由开销），将 MoE 模型蒸馏 (distill) 为一个等效的密集模型 [1]。
- 影响：这种攻击" 消除了 MoE 结构" [1]。它" 抹去" (obliterate) 了所有的路由依赖性 [1]，从而彻底摧毁任何基于路由（行为）的水印。

4.4 后处理攻击 (Post-Processing Attack)

- 机制：文本转述、编辑或通过另一个 LLM 进行增强 [1]。
- 影响：这种攻击只对基于输出 (Output-Based) 的水印有效（例如，操纵词频） [1]。它对于本报告讨论的两种隐式方法（基于参数和基于路由）是无效的 [1]。

4.5 MoE 水印攻击向量与缓解策略汇总

为了系统地综合 MoE 水印面临的威胁，表 1总结了攻击向量、其目标、影响以及文献中提出的缓解策略和有效性。

5 高级缓解策略与鲁棒性基准

基于上述漏洞，必须设计能够抵御多重威胁的先进缓解策略。

5.1 保护控制平面（路由器）

- 策略：" 在路由中使用加密哈希" (Use cryptographic hashing in routing) [1]。
- 机制分析：尽管 [1] 指出具体机制未详述，但可以推断其工作原理。该策略可能涉及使用一个秘密密钥 K 和令牌的表示 x 来动态生成一个偏置项（例如，$bias = Hash(K, x)$），该偏置项被添加到路由器的 logits 中。这会强制执行" 正确" 的水印路径，而不会在模型权重中存储

Table 1: MoE 水印攻击向量与缓解策略汇总

攻击向量	目标水印类型	对水印的影响	缓解策略	文献引用的有效性
路由干扰 (Router Disruption)	基于行为 (路由)	擦除控制平面信号 [1]	在路由中使用加密哈希 [1]	使用冗余设计可实现 50-70% 的信号恢复 [1]
专家重训练 (Expert Retraining)	基于参数 (权重)	改变子模型行为 [1]	跨多个专家嵌入水印 [1]	若无防护，鲁棒性损失 20-30% [1]
专家剪枝 (Pruning)	基于参数 (权重)	移除不活跃的专家 [1]	为稀疏弹性而设计 [1]	水印降解高达 40% [1]
蒸馏为密集模型 (Distillation to Dense)	所有 MoE 特定的 (路由 & 专家)	消除 MoE 结构 [1]	采用能在转换中存活的混合水印 [1]	精心设计可实现 60-80% 的信号保留 [1]
文本后处理 (Text Post-Processing)	基于输出 (文本)	模糊输出模式 [1]	专注于行为（非输出）水印 [1]	文本方法的检测率降低 30-50% [1]

一个易于通过微调去除的明显偏置。[1] 中提到的"50-70% 信号恢复" [1] 表明，这种哈希机制可能与跨多个路由层或专家的冗余编码相结合。

5.2 保护数据平面（专家）

- 策略 **1 (**冗余**):** " 跨多个专家嵌入水印" (Embed watermarks across multiple experts) [1]。这是对抗剪枝和重训练的有效防御。攻击者必须成功攻击（重训练或剪枝）所有携带水印信号的专家，而不仅仅是一个，这大大增加了攻击成本。
- 策略 **2 (**弹性**):** " 为稀疏弹性而设计" (Design for sparse resilience) [1]。这暗示了一种更智能的嵌入策略：将基于参数的水印嵌入到那些活跃的、高价值的专家中，这些专家在任何合理的剪枝算法下都不太可能被移除。

5.3 蒸馏悖论与混合水印

对 MoE 水印最严峻的挑战是蒸馏攻击，它引发了一个明显的悖论：

1. 蒸馏攻击会移除路由器 [1]。
2. 基于路由的水印依赖于路由器 [1]。
3. 因此，一个纯粹基于路由的水印无法在蒸馏后存活。

然而，一个关键的数据点（ [1]）引用了" 混合水印" 在模型转换（蒸馏）后仍能实现"60-80% 的信号保留" [1]。

这种保留不可能来自于已被消除的路由信号。它必须来自于混合水印的另一部分——即基于参数（权重）的组件 [1]。

机制解析：当一个 MoE 模型被蒸馏时，其所有专家的权重通常被" 合并" 或" 平均" 以创建新的密集层。嵌入在专家权重中的基于参数的水印（即权重扰动） [1] 会在这个过程中被" 涂抹" (smeared) 到新的密集权重中。虽然信号被稀释了，但它并不会完全消失。通过精心的设计（例如使用 ECCs 或鲁棒嵌入技术） [2]，这个" 涂抹" 后的信号仍然可以被白盒检测器识别出来。

这一分析揭示了最鲁棒的策略：双重编码混合水印 **(Dual-Encoding Hybrid Watermark)**。

- 编码器 **1 (**路由器**):** 一个基于路由的偏置（如 BadSwitch 改编版） [5]，提供隐蔽性，并可能用于黑盒验证。
- 编码器 **2 (**专家**):** 一个基于参数的信号（如 ECC 编码的权重扰动） [2]，嵌入到路由器被偏置指向的相同专家中。

最终的鲁棒性**:** 这种双重模型提供了分层防御。它能抵抗路由攻击（参数水印存活）和专家微调攻击（路由偏置可能存活）。最关键的是，它也能抵抗蒸馏攻击——路由信号丢失，但参数信号以 60-80% 的保留率" 涂抹" 并存活在最终的密集模型中 [1]。

6 综合与未来研究方向

6.1 路由信息 vs. 权重信息：对比分析

本报告的分析表明，单独依赖路由信息或权重信息都不足以提供全面的保护。

- 基于路由 (行为) [1]: 优势在于高隐蔽性、低性能影响，但其" 控制平面" 信号非常脆弱，易受路由干扰和结构性（蒸馏）攻击。
- 基于权重 (参数) [1]: 优势在于对输出编辑的完全免疫，但其" 数据平面" 信号易受模型编辑（微调、剪枝）攻击。

结论是，未来的研究方向必然是双重编码的混合水印。这种方法利用 MoE 的两个平面（控制平面和数据平面）进行分层防御，提供了对抗多种、相互冲突的攻击向量的最强鲁棒性。

6.2 开放性问题与未来方向

尽管 MoE 水印前景广阔，但该领域仍处于起步阶段，面临诸多挑战 [1]：

1. 可扩展的检测工具**:** 最大的开放性问题是，如何在无法访问激活日志的情况下（例如，通过公共 API）可靠地验证基于路由的水印 [1]。这对于黑盒溯源至关重要。
2. 标准化基准**:** 目前缺乏专门针对 MoE 漏洞（如路由干扰、剪枝、蒸馏）的水印评测基准。开发类似于 WMD 或 AIGCDetect 但专注于 MoE 的标准将是必要的 [1]。
3. 将检测作为 **MoE:** 一个有趣的反向概念在 [8] 中被提出，即使用 MoE 架构（一个门控模型 + 多个专家检测器）来进行水印检测。这暗示了一个未来，即验证工具本身可能会采用与它们所检查的模型相同的先进架构。

总之，MoE 的路由和专家分叉结构为隐式水印提供了一个极具吸引力但尚未充分探索的向量。成功利用这一向量将取决于通过创新的混合设计来克服其固有的复杂性和脆弱性。

致谢

感谢所有为本研究提供支持和反馈的同事和审稿人。

References

[1] LLM MoE watermark.pdf

[2] Robust and Efficient Watermarking of Large Language Models Using Error Correction Codes, accessed November 7, 2025, https://petsymposium.org/popets/2025/popets-2025-0126.pdf

[3] Robust Watermarking of Tiny Neural Networks by Fine-Tuning and Post-Training Approaches - MDPI, accessed November 7, 2025, https://www.mdpi.com/2073-8994/17/7/1094

[4] Towards Robust Model Watermark via Reducing Parametric Vulnerability - CVF Open Access, accessed November 7, 2025, https://openaccess.thecvf.com/content/ICCV2023/papers/Gan_Towards_Robust_Model_Watermark_via_Reducing_Parametric_Vulnerability_ICCV_2023_paper.pdf

[5] Who Speaks for the Trigger? Dynamic Expert Routing in Backdoored Mixture-of-Experts Transformers - arXiv, accessed November 7, 2025, https://arxiv.org/html/2510.13462v1

[6] (PDF) Who Speaks for the Trigger? Dynamic Expert Routing in Backdoored Mixture-of-Experts Transformers - ResearchGate, accessed November 7, 2025, https://www.researchgate.net/publication/396518065_Who_Speaks_for_the_Trigger_Dynamic_Expert_Routing_in_Backdoored_Mixture-of-Experts_Transformers

[7] [2510.13462] Who Speaks for the Trigger? Dynamic Expert Routing in Backdoored Mixture-of-Experts Transformers - arXiv, accessed November 7, 2025, https://arxiv.org/abs/2510.13462

[8] FT-Shield: A Watermark Against Unauthorized Fine-tuning in Text-to-Image Diffusion Models, accessed November 7, 2025, https://arxiv.org/html/2310.02401v2