

Attention-CEM 防御框架总览

Slot+Gated Cross Attention vs. Gated Attention Pooling

项目阶段性汇报

组会交流

Attention Privacy Project

为什么

- 协同推理的隐私风险
- CEM 框架基本思路

Gated Attention Pooling

- 方案动机与流程
- 条件熵代理公式

Slot+Gated Cross Attn

- 模块结构图
- 数学细节与门控
- 训练集成

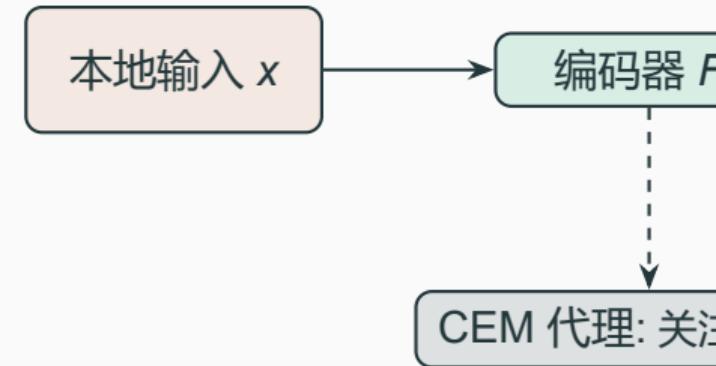
总结与计划

- 两者并列对比
- 实践建议与风险点
- 下一步重点

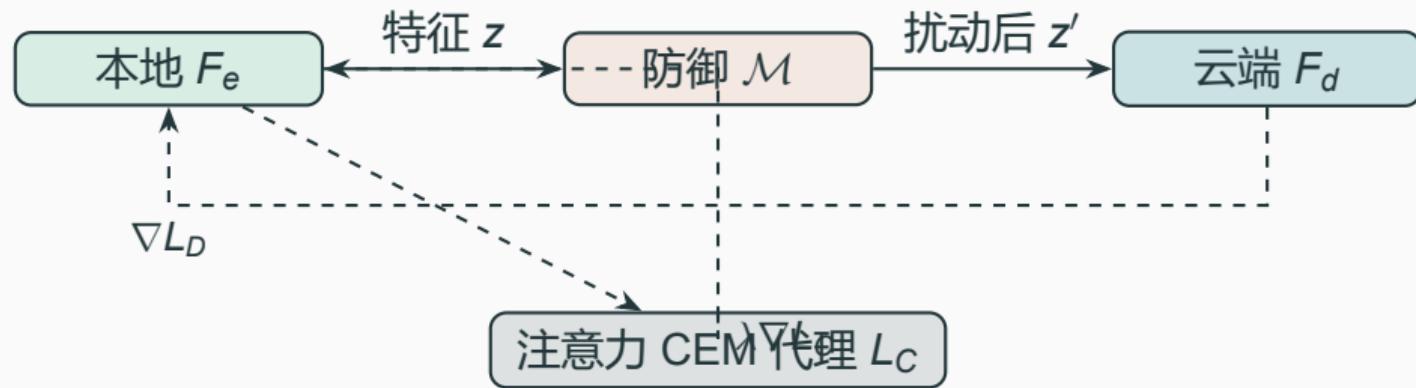
项目背景

协同推理与 CEM 目标

- **协同推理**: 本地编码器 F_e 产生中间特征 z ,
云端解码器 F_d 完成预测。
- **隐私问题**: 中间特征易被模型反演攻击恢复输入。
- **CEM 思路**: 最大化条件熵 $H(x|z)$, 提高攻击者的最优重建误差 ξ 。
- 原始代码使用 **KMeans/GMM** 近似条件熵,
下文两套注意力框架即替换该近似器。



训练循环中的注意力 CEM



Slot + Gated Cross Attention

- 优势: 显式建模多模态模式; 门控丰富, 能够“削”走冗余信息。
- 难点: 结构庞大, 超参数多, 需要细致日志监控。

SlotAttention 工作流

1. LayerNorm 后得到 K, V 。
2. 以学习到的 μ, σ 初始化 S 个 slot。
3. 每轮更新:
 - 归一化 slot → 投影为查询 Q 。
 - $r = \text{softmax}(KQ^\top)$, 加入 ϵ 再归一化。
 - 使用 GRU+MLP 残差更新 slot 状态。

关键超参

- slot 数量 $S = 8$
- 迭代次数 $T = 3$
- 温度缩放: $\text{slot_dim}^{-1/4}$

Gated Cross Attention 细节

- 结构:

$$y = q + \tanh(\alpha_{\text{attn}}) \cdot \text{CrossAttn}(q, s), \quad y = y + \tanh(\alpha_{\text{ffn}}) \cdot \text{FFN}(\text{LN}(y)).$$

- 多头交叉注意力: slots 作为 KV, 类内样本为查询。
- 门控参数初值小正 (0.1), 保证注意力/FFN 渐进式生效。
- 每个子层前置 LayerNorm, 提升训练稳定性。

多级门控 + CEM 代理

1. **Slot 方差**: $r_{ms} = \text{softmax}(\beta \cdot \text{sim}(x_m, s_s))$, 求得 μ_s, σ_s^2 。
2. **Per-dim Gate**: LayerNorm($\log \sigma_s^2$) → MLP → Sigmoid。
3. **SNR Gate**: $g_{\text{snr}} = \sigma(\kappa(\sigma^2 / (\mu^2 + \epsilon) - \tau_{\text{snr}}))$ 。
4. **Softplus Margin**: $L_{\text{base}} = \frac{1}{\beta'} \log(1 + e^{\beta'(\log \sigma^2 - \log \tau - m)})$ 。
5. **Slot Mass Gate**: $(\text{mass}/M)^\gamma$ 强调主导 slot。
6. **Class Gate**: $g_{\text{class}} = \sigma(a(M/B - b))$ 调节不同样本量类别。
7. **Early Shutoff**: 前 100 步或门控统计过高立即输出 0, 防止梯度抖动。

训练集成

1. Warmup: 'self.attention_warmup_epochs = 3'。
2. 首次调用时把 'SlotCrossAttentionCEM' 参数加入主优化器。
3. 'rob_loss' 反向，缓存编码器/注意力梯度，再执行 L_D 回传。
4. 将 CEM 梯度乘以 λ 或 'attention_loss_scale' 重叠至编码器参数。
5. 保持与噪声、Dropout、ARL 等防御模块相同的调用顺序。

Gated Attention Pooling

方案动机

- Slot 方案在高分辨率或大 batch 下计算开销大、调参繁琐。
- Gated Attention Pooling 借鉴多实例学习：用单注意力权重汇聚类内特征，结构极简，可快速部署。

数学表达

$$a_m = \frac{\exp(w^\top [\tanh(Vx_m) \odot \sigma(Ux_m)])}{\sum_j \exp(\cdot)},$$

$$\mu = \sum_m a_m x_m, \quad \sigma^2 = \sum_m a_m (x_m - \mu)^2,$$

$$L_C = \max\{0, \log(\sigma^2 + \gamma) - \log(\tau)\},$$

$$\tau = \text{var_threshold} \cdot \text{reg_strength}^2 + \gamma.$$

特点

- 仅需两层线性映射 + softmax，速度快。
- 配合 LayerNorm 避免注意力塌缩。
- 不再依赖 slot 或额外门控，调参集中在阈值与缩放。

训练集成要点

- Warmup 更长 (默认 5 个 epoch) 以稳定 softmax 权重。
- 首次调用时同样将模块参数加入主优化器。
- ‘attention_loss_scale’ 控制 CEM 梯度强度， 默认 0.25。
- 与 Slot 框架共享完全相同的噪声 / Dropout / ARL 链路。
- 算力友好，适合快速验证或放大 batch 以估计鲁棒性。

对比总结

两种注意力方案对比

	Slot + Gated Cross Attn	Gated Attention Pooling
表达能力	多 slot + 多头 cross-attn, 精细建模类内多模态	单注意力分布, 更偏向“聚合摘要”
数值稳定	依赖多级门控 + early shutdown 保护	LayerNorm + softmax 足够稳定
参数/算力	GRU/slot/门控较多, 算力大	轻量 MLP, 适合高分辨率或大 batch
调参成本	需同时调 slot extunder-score power、class extunderscore gate、阈值等	主要关注 var extunder-score threshold 与 loss 缩放
适用场景	复杂多模态、隐私泄露风险高的数据集	快速迭代、资源受限或原型验证

优先检查

- Warmup 是否完成? ‘current_epoch’
- ‘rob_{loss}’ MSENaN/Inf
- Slot 框架的门控均值是否过高触发 early shutoff?

调参指南

- Slot 框架: 先固定噪声强度, 再调整 ‘slot_{power}’ ‘class_{gate_a/b}’ ‘attention_{loss}scale’
- Gated 框架: 锁定 ‘reg_strength’ ‘var_threshold’ ‘loss’

下一步计划

- 在 CIFAR-10/100、FaceScrub、TinyImageNet 等数据集上完成系统实验。
- 与原 GMM 方案对比 MIA 指标 (MSE/SSIM) 与训练开销。
- 尝试混合策略:Gated pooling 预筛查 → Slot 框架精细优化高风险类别。
- 深入分析 early shutoff、class gate 等超参对隐私-准确率折中的影响。

感谢指导，欢迎讨论！