Slot + Gated Cross 条件熵代理框架易懂手册

自动生成

November 6, 2025

Abstract

这份文档面向第一次接触项目的读者。我们会用生活化的比喻解释Slot Attention + Gated Cross Attention 版CEM 的整套流程：为什么要这么做、代码里每一步在干什么、门控和阈值的作用是什么、训练时怎么与主任务拼在一起。目标是让你读完后，即便不熟悉注意力或条件熵，也能明白模型的运作逻辑，并知道哪些开关可以调、遇到问题该看哪里。

Contents

1 先说结论：我们想解决什么问题?

- **我们希望类内特征更“抱团”**：同一类别的样本在特征空间里越集中，攻击者就越难根据激活重建原图。
- **CEM（条件熵最小化）就是衡量“抱团”程度**：如果类内很分散，条件熵高；如果集中，条件熵低。
- **老办法不好用**：KMeans / GMM 需要选簇数、初始化，特征维高的时候不稳。
- **我们的方法**：先找“代表队长”（Slot Attention），再让每个样本向队长取经（Gated Cross Attention），最后做一套门控统计判断类内是否紧凑。

2 整体流程像是“班干部+分工”

1. **阶段一：挑队长（Slot Attention）**
同类样本站成一排，竞争成为几个“队长”。队长会多轮征召、更新，直到每个子群体都有代表。
2. **阶段二：跟队长学习（Gated Cross Attention）**
每个样本向所有队长请教，按注意力权重把队长的经验加到自己身上，但有门控，避免一下子改变太多。
3. **阶段三：统计班级是否稳定（门控方差+ CEM）**
用余弦相似度算每个样本跟哪个队长最亲，再按权重统计方差。多重门控过滤噪声、突出重点，最后得到“这类是否松散”的分数。

3 步骤1：Slot Attention（挑队长）

1.1 为什么需要它?

如果直接对全部样本求方差，会把所有细节都记下来，既费算力又不稳定。Slot Attention 用“几个代表”概括整个类，既能捕捉多模态，又能保持计算量可控。

1.2 它具体做什么?

1. **预处理**：每个样本先做LayerNorm，避免谁数字特别大。

2. **生成Key/Value**：把样本投影成Key（用于和队长比对）和Value（真正要聚合的信息）。
3. **初始化队长**：从一个可学习的高斯分布里随机抽几个初始槽（队长）。初始就有差异，防止卡在一个点。
4. **多轮竞争更新**：
   每一轮包括：
   - 用上一轮的队长向样本发出Query（“我想找和我最像的同学”），点积后用softmax 分配权重。
   - 按权重把样本的Value 加权和，得到新的输入。
   - 用GRU + MLP 更新队长，既吸收新信息，又保留记忆。
5. **多轮迭代后输出**：最终每个队长代表一组相似样本，像“子簇”的中心。

1.3 小技巧

- 点积里有一个温度系数$\tau = d^{1/4}$：把分数压缩一些，避免softmax 太尖锐。
- LayerNorm 和GRU + MLP 残差结构确保训练稳定。

4 步骤2：Gated Cross Attention（跟队长学习）

2.1 目的

有了队长，还要让每个样本从队长那里“听取意见”，得到更稳的表示。这里用的是跨注意力（样本是Query，队长是Key/Value）。

2.2 流程

1. **对齐尺度**：样本、队长分别做LayerNorm 和线性映射。
2. **算注意力权重**：样本Query 去和每个队长Key 点积，softmax 后得到“我最该听哪个队长”的权重。
3. **聚合信息**：按权重加权平均队长的Value，就得到一个增强特征。
4. **门控残差融合**：
   先把增强特征通过线性层，再乘上$\tanh(\alpha_{\text{xattn}})$ 加回原特征。α 是可学的，初始很小，就像一开始只允许少量建议进入。
   再走一遍前馈网络（FFN），同样用$\tanh(\alpha_{\text{ffn}})$ 控制力度。

2.3 为什么要门控?

- 防止模型一开始就“听命于注意力”，导致主任务学不动。
- 训练过程中会慢慢加大门值，等网络学会了，就能充分利用注意力。

5 步骤3：混合槽统计& 条件熵判定

3.1 三个问题

1. 哪个样本应该归哪个队长？（责任分配）
2. 每个队长的“队员”分布集中吗？（加权均值/方差）
3. 哪些方差是真问题，哪些只是噪声？（门控和阈值）

3.2 逐步拆开

1. **再做一次LayerNorm**：确保增强后的样本可比。
2. **余弦相似度+ softmax**：让每个样本得到一组对队长的责任权重r_{mk}。温度β 可学习，决定“分配多平均还是更偏向某一个队长”。
3. **算加权均值/方差**：像计算带权重的统计量一样，得到每个队长的μ_s 和σ_s^2。
4. **多重门控过滤噪声**：
   - **维度软门**：对$\log \sigma^2$ 做LayerNorm + MLP + Sigmoid，判定“这个维度可靠不可靠”。
   - **SNR 硬门**：看$\sigma^2/(\mu^2+\varepsilon)$，信噪比低的维度被压低。
   - **Softplus 阈值**：不用硬ReLU，而是用平滑的Softplus 函数，让阈值附近也有梯度。
5. **槽权重**：看每个队长有多少队员（slot mass），对权重做幂次放大，保证“人多的队长”更有话语权。
6. **类级聚合**：把上述门控乘起来，先按队长权重求和，再对所有维度取平均，得到类级指标L_c。
7. **类级门**：如果某类在当前batch 里样本很少，就用Sigmoid 门把它的贡献压小，防止统计误判。

3.3 最终输出

- **rob_loss**：所有类的L_c 按样本占比加权平均，就得到正则损失。
- **intra_mse**：同时记录类内MSE，方便看类内是否在收缩（只是日志，不参与梯度）。

6 训练里怎么用?

4.1 触发条件

- 不随机初始化聚类中心、$\lambda > 0$、当前epoch 超过warmup（默认3）。
- 首次使用时实例化模块，并把参数加入优化器。

4.2 梯度处理

1. 先对rob_loss 反向，保存编码器和注意力模块的梯度。
2. 清梯度，再对主任务交叉熵反向。
3. 把rob_loss 的梯度按学习率等比例缩放后加回编码器参数里。
4. 注意力模块的梯度直接相加。

这样做是为了“显式注入”正则的梯度，而不是把它和主损失简单相加导致训练早期不稳定。

4.3 数值保护

- rob_loss 或intra_mse 出现NaN/Inf 会被自动置零。
- 早期关断逻辑：如果门值过大或还在warmup，就返回0，但是梯度图仍然连着。

7 调参和排障清单

5.1 必调/常用参数

- `num_slots`：队长数量，默认8。类内子模式越多可以适当增大。
- `num_iterations`：Slot Attention 迭代次数，默认3，更多会更精细但更慢。
- `assign_temp`、`slot_power` 等可学习参数无需手调，训练中会自行调节。
- `attention_loss_scale`：rob_loss 的全局缩放，默认0.25。若正则力度不够，可以逐步调大。
- `var_threshold`：阈值越大，越宽容；如果希望更严格，可减小该值。

5.2 出问题时看哪里?

- **日志关键字**：`[CEM-GATE]` 打印会显示门平均值、rob_loss。门值太高意味着大部分维度都被判为“不可靠”，可能需要调整阈值或warmup。
- **NaN**：检查输入特征是否有Inf/NaN，或者Softplus 参数是否过大。
- **正则太弱/太强**：调`attention_loss_scale`，观察分类准确率与rob_loss 的平衡。

8 常见疑问速查

- **Q：为什么要先Slot 再Cross?**
  A：Slot 负责找“代表”。Cross 负责让每个样本在代表的帮助下更稳定。缺一不可。
- **Q：和Gated Attention（gated-att）那套相比?**
  A：gated-att 是“单层加权平均”，计算轻；Slot+Cross 支持多子簇，结构更丰富，但也更复杂。
- **Q：如果batch 很小呢?**
  A：类内样本少时，类级门会把权重压小，避免噪声统计影响训练。

9 你可以做的下一步

- 对每个类画出责任分布（r_{mk}）和slot mass，看队长是否覆盖均匀。
- 记录训练过程中的rob_loss、intra_mse 曲线，确认正则在起作用。
- 结合新的流程图（见文档同目录下的PDF/PNG），在组会上逐阶段解释，帮助团队理解这套机制。