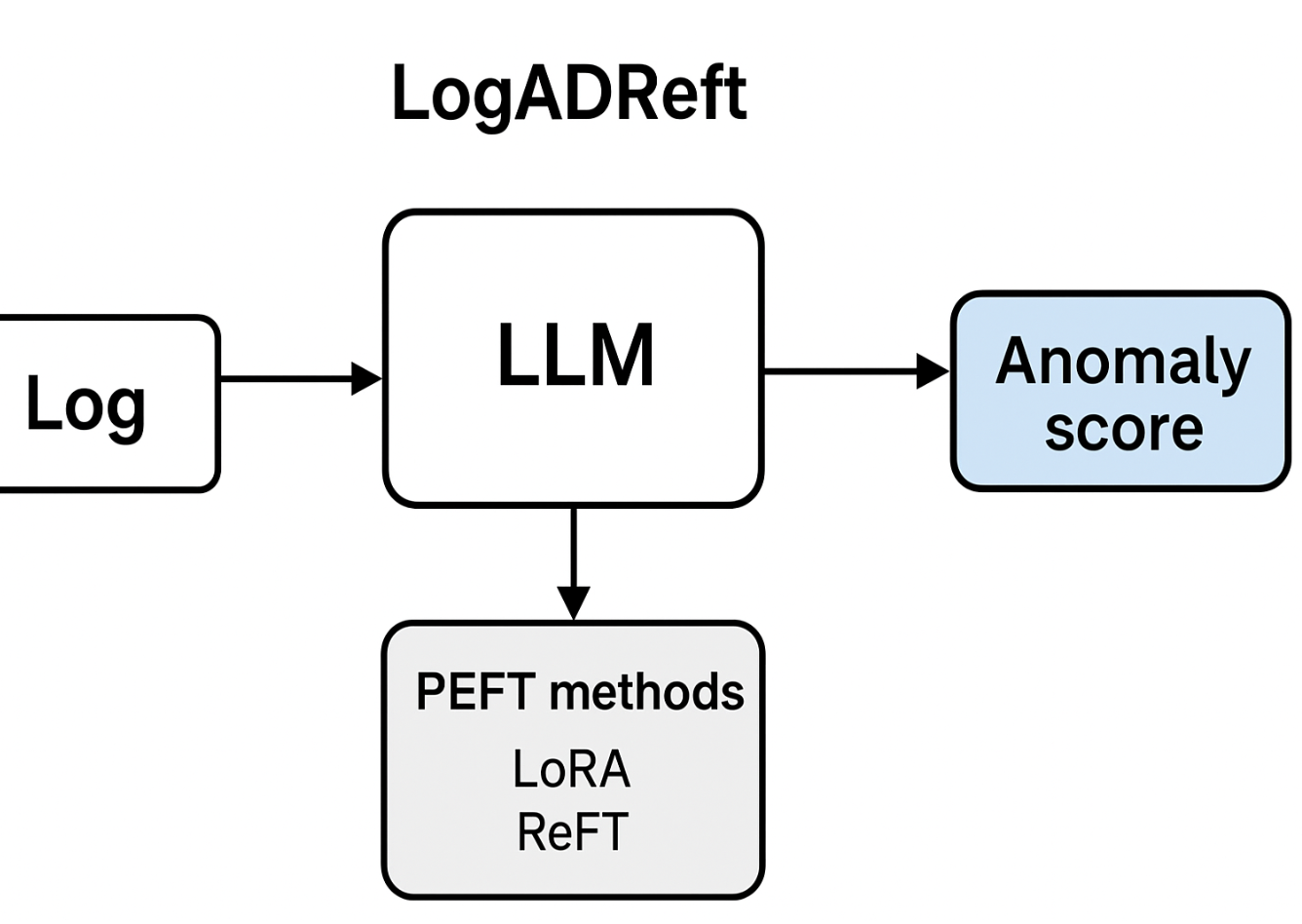
**LogADReft** 是 PAKDD’25 论文 *“Adapting Large Language Models for Parameter-Efficient Log Anomaly Detection”* 的官方实现。  
它的目标是：  
**利用大语言模型（LLMs）+ 参数高效微调（PEFT）方法，来做日志异常检测（Log Anomaly Detection, LAD）**

**项目核心点**

1. **为什么要用 LLM？**
   * 日志数据复杂、格式多样、含有大量自然语言信息。
   * 传统深度学习模型需要人工特征工程，迁移性差。
   * LLM 可以直接理解上下文，迁移到新系统日志时更方便。
2. **为什么要参数高效微调（PEFT）？**
   * 直接微调 LLM 太耗资源。
   * 他们重点比较了两种方法：
     + **LoRA**（低秩适配）
     + **ReFT**（Reference-tuning，新方法）
   * 实验发现 ReFT 在多数场景下更 **稳定、高效**。
3. **主要功能**
   * 支持 **日志预处理**（分词、滑窗切片）
   * 支持 **多种 PEFT 方法**（LoRA、ReFT）
   * 提供 **公共数据集实验**（HDFS, BGL, Spirit, Thunderbird）

简单来说：  
**LogADReft = 把 LLM 变成“日志异常检测专家”，还能在低算力设备上高效部署。**



**📥 输入（Input）**

1. **原始日志数据**（系统运行产生的 log 文件，比如 HDFS、BGL、Spirit、Thunderbird）
   * 每一行是一条日志
   * 包含时间戳、日志级别（INFO、WARN、ERROR）、日志内容
2. **预处理后的日志序列**
   * 经过 **解析 (parsing)** → 抽取模板、替换参数
   * **切片 (sliding window)** → 按固定窗口（如 50 行日志）切分成片段
   * 最终得到的就是 **日志片段文本**，可以直接送入 LLM

**📤 输出（Output）**

1. **异常分数 (Anomaly Score)**
   * 模型会给每个日志片段打分，表示它“正常”还是“异常”的概率
2. **预测标签 (Anomaly / Normal)**
   * 通过设定阈值，把日志片段标记为“异常”或“正常”
3. **评估指标**（在实验里用）
   * Precision（查准率）
   * Recall（查全率）
   * F1-score

