这个项目旨在构建一个融合知识追踪（Knowledge Tracing, KT）模型与强化学习机制的个性化教育推荐系统。项目的核心任务是模拟学生学习过程，根据其答题历史动态预测知识掌握水平，并在此基础上设计具有奖励机制的学习路径推荐策略，从而提升学习效果。

项目中首先通过 train.py 脚本完成 KT 模型的训练。该文件使用 MXNet 框架实现了 EmbedDKT 模型结构，并封装在 EmbedDKTModule 中。训练过程通过 train\_EDKT(train\_data\_path, test\_data\_path, \*\*kwargs) 函数进行调用，支持灵活的参数配置，包括数据集路径、模型保存位置、训练轮数、设备（GPU/CPU）选择等。在调用过程中，系统会加载指定训练数据和测试数据，构建网络结构，定义损失函数 DKTLoss() 并自动进行多轮训练，同时支持 PRF 和 Accuracy 等多项指标的评估与日志输出。训练完成后，模型可用于学生知识状态预测任务。

在 KT 模型训练完成后，可以使用 HYZ\_Env.py 文件中定义的 Env 类创建一个仿真学习环境。该环境的核心作用是为推荐系统提供强化学习的状态、动作、奖励机制，从而便于策略学习。环境初始化时通过 Env(kt\_model, dataset\_path, difficulty\_path, reward\_type, kc\_num, type="train") 构造函数加载学生数据、题目难度信息和训练模式（train/test）。其中 kt\_model 为训练好的知识追踪模型，kc\_num 表示知识点数量。

环境支持两种模式：训练模式随机抽取学生用于模拟学习过程，而测试模式则按指定顺序评估个体表现。每轮推荐开始前调用 env.begin() 方法初始化学生状态，随后通过 env.state\_transform(action) 实现学生对推荐题目的响应、知识状态更新和即时奖励计算。奖励函数设计支持多种策略，如：

ep：目标知识点掌握度的进步比例；

ep/t：考虑耗时的掌握进步；

New：整体知识提升的效率；

unknow：基于难度变化与掌握度进展的综合指标。

推荐过程可追踪当前状态、累积奖励与推荐路径，最终通过 env.get\_result() 输出完整学习轨迹与总收益评估。该设计不仅可用于策略训练，也便于在实验中对不同奖励机制进行效果对比。

以下是一个完整的使用示例，展示如何加载训练好的模型并在环境中模拟一个学生的推荐过程：



