摘要

OneRec-Think，一个将对话、推理与个性化推荐无缝融合的统一框架。
OneRec 严重缺乏显式且可控的推理能力

> 为什么生成式模型需要具有推理能力?
> 如果模型不具备推理能力，它会：
>
> - 生成与用户复杂目标不一致的结果
> - 难以解释为何推荐某个物品
> - 无法处理用户的复杂、多跳、多因素偏好
>
> 推理是让模型在开放生成空间中“自我约束”的关键。

OneRec-Think 包含三个关键组件：
(1) Itemic Alignment（项目对齐）：将语义token作为LLM额外的词表去做预训练, 让Item语义token对齐LLM的文本token embedding空间
(2) Reasoning Activation（推理激活）：在推荐上下文中激活模型的推理能力
(3) Reasoning Enhancement（推理增强）：设计了一个面向推荐任务的奖励函数，利用特定于推荐的奖励函数来捕获用户偏好的多样性

下图为OneRec-Think将对话、推理与个性化推荐融合的例子：

Figure 1: Examples of OneRec-Think's Unified Dialogue, Reasoning and Recommendation Framework.

相关工作

近年来，**基于推理的推荐系统**开始通过多步推断（multi-step deduction）实现更准确且可解释的推荐。

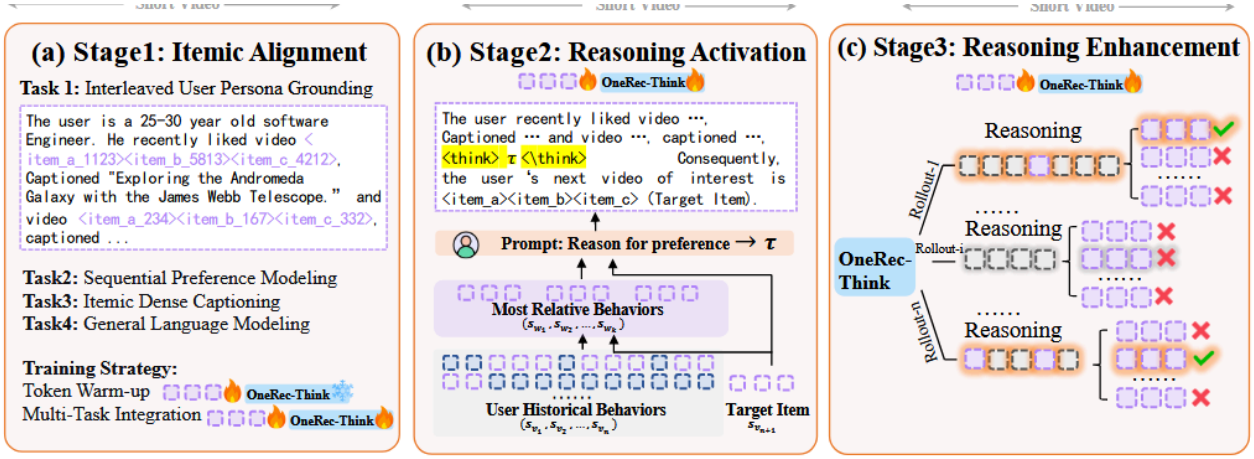
现有方法大致分为两类：

- **显式推理方法（explicit reasoning）**
  输出推荐+可读的解释性推理路径，比如：
  你最近观看了多条军事技术分析，并对战斗机主题表现出持续兴趣。 该视频详细介绍了 J-35 的最新飞行测试因此我推荐 `<item_xyz>`。
- **隐式推理方法（implicit reasoning）**
  进行潜在推理，但不提供可解释的文本理由，比如：
  我推荐视频：<item_xyz>
  OneRec-Think 将显示推理引入生成式推荐系统

框架图

OneRec-Think的整体框架如下图所示:

主要包含3个模块：



预备知识 Preliminary

Itemic Token

把物品 Item 当成“单词 token 序列”一样表示，然后把用户的交互行为看作为一个由这些 token 组成的序列，最终让生成式模型像预测下一句话一样预测下一件要推荐的物品。

传统生成式与OneRec-Think的对比

传统模型：
输入用户历史 → 直接输出推荐 item
OneRec-Think：
输入用户历史 → 先生成推理文本（reasoning）→ 再输出推荐 item

方法论 Methodology

Itemic Alignment

Itemic 对齐
LLM 本来只懂“语言 token”，而不懂“itemic token”（物品 token），所以必须通过对齐让它能同时理解语言和物品，从而把推荐知识和语言推理能力融合起来。

使用多任务预训练实现 item 语义对齐

构建训练预料数据有4个部分：

1. 双模态训练样本 Interleaved User Persona Grounding：交织使用“itemic token”与“用户画像文本 token”（包含了用户画像描述、近期行为如搜索、点赞描述、用户兴趣摘要）
2. 序列偏好建模 Sequential Preference Modeling：基于用户的时间序列行为构建训练数据，引导模型学习“根据历史交互预测下一次物品消费”的能力
3. Itemic 密集描述生成 Itemic Dense Captioning：LLM能学习、解码到Item的语义Token，所表示具体文本描述性内容。eg: `<item_a_2387>` = 牛排 , `<item_b_998>` = 烤虾等
4. 通用语义建模 General Language Modeling：使用通用语料库作为增加的训练数据，保证LLM不会丧失原有的世界知识。

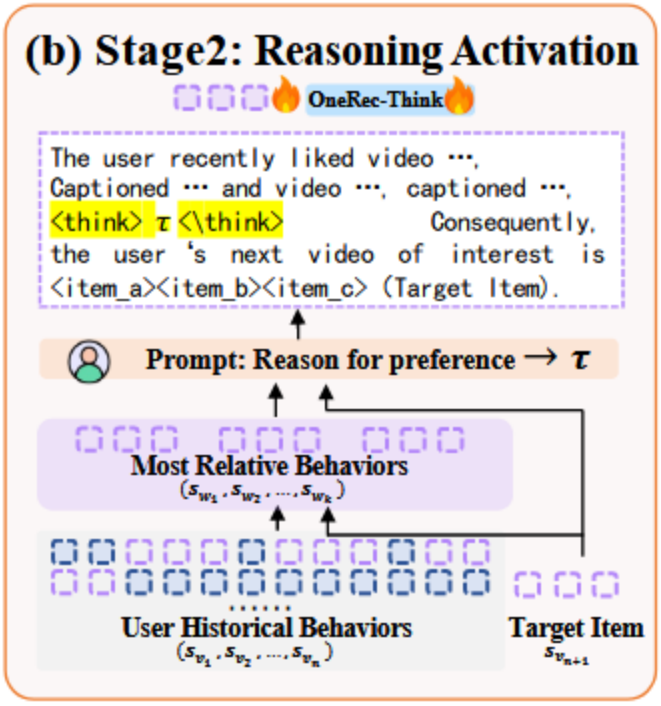
两阶段训练策略

- Token Warm-up阶段（token热身）：训练 itemic token 的嵌入向量， 并且冻结基础 LLM， 训练任务仅为 1.Interleaved User Persona Grounding。
- Multi-Task Integration 阶段（多任务整合）：在该阶段，解冻模型的所有参数，对上述4个任务进行联合优化

Reasoning Activation

推理激活
现实场景常常无法产生有效的链式思维（CoT）推理，其根本原因是现实用户行为序列往往又噪声又冗长。为了解决这个问题，我们通过监督学习的框架，引导模型生成高质量的推理路径，从

而提升推荐的准确性和可解释性。该过程包含两个步骤：



基于裁剪上下文的推理能力引导

（Bootstrapping with Pruned Contexts）
该过程最终会生成逻辑严谨、且与目标交互高度相关的高质量推理理由，从而为推理能力的归纳学习提供理想的训练信号。

从噪声序列中学习推理

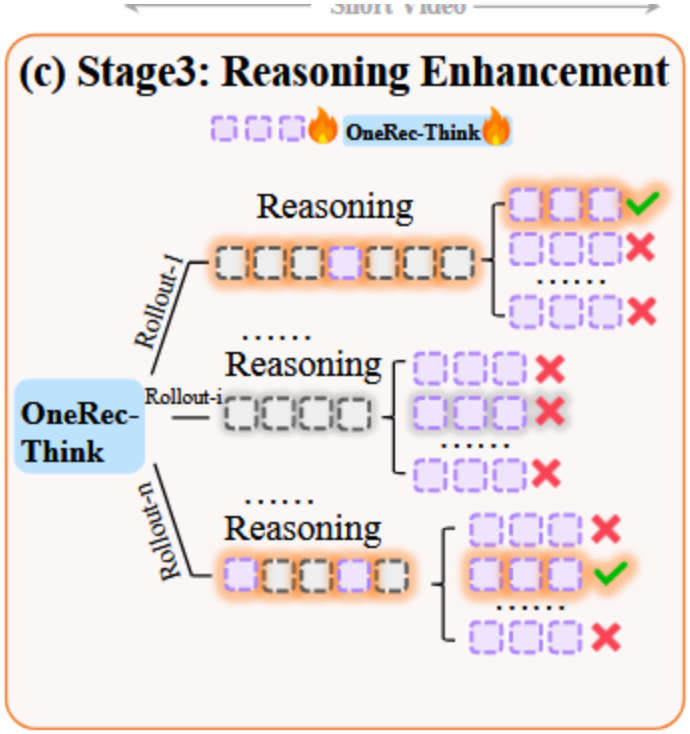
（Learning to Reason from Noisy Sequences）
上述蒸馏出的推理理由会被用作监督信号，帮助模型从原始（带噪声）的序列中学习推理能力。

上面两个步骤使用LoRA微调推理过程

Reasoning Enhancement

推理增强
在前一阶段通过推理激活获得初步的链式推理能力后（CoT），本阶段推理增强是通过设计特定

的奖励机制, 使用强化学习来进一步优化模型的推理能力和推荐准确性。



Beam Candidate Reward Maximization

Beam 候选奖励最大化
在推荐场景中，标准的可验证奖励（如精确匹配）面临稀疏性挑战，因为大多数推理路径无法精确命中Target Item，导致奖励信号过于稀疏，得到的奖励都是 0。为此, 定义了一种新的奖励机制，称为 Rollout-Beam奖励，即"**取Beam Search结果中的最大最好的路径Reward**"

部署：预思考（Think-Ahead）架构

在工业级推荐系统中部署 OneRec-Think 面临一个关键挑战：
如何在保证实时响应（极低时延）的前提下，处理多步推理带来的高计算开销。
解决方案：推理拆分为两个阶段（离线 + 在线）

第一阶段：离线处理（计算量大）

在这一阶段，我们使用完整的 OneRec-Think 模型 离线生成：

- 推理路径（这些路径代表了用户行为与潜在兴趣之间的逻辑联系。）
- 初始 item token（例如前两个 itemic tokens）

第二阶段：在线处理（低延迟）

在线阶段使用 实时更新的 OneRec 模型（Zhou et al., 2025a），以确保工业级的实时性能。
它利用离线生成的初始 item token 作为约束前缀（constrained prefix），快速生成：

- 最终的 itemic token（即最终推荐结果）

实验

消融实验（Ablation Study）

目的：验证模型中某个模块或组件对整体性能的贡献
方法：移除（或关闭）某个模块，对比模型性能下降幅度，如果移除后性能明显下降，就说明这个模块对模型效果很重要。在 Beauty 数据集上，通过消融实验验证 IA 模块和 R 模块的效果。

具体怎么做：
先用完整模型（包含 IA + R）做推荐，记录性能指标（如准确率、NDCG、Recall 等）
再分别移除 IA 或 R 模块，重新训练或推理
对比性能差异，评估 IA 和 R 模块对最终推荐效果的贡献

IA（Itemic Alignment）模块
功能：将物品 token（itemic token）和语言 token 对齐，让 LLM 理解物品语义
消融方法：去掉 IA 模块，让模型直接处理 item token，不做对齐
R（Enhanced Reasoning Mechanism）模块
功能：通过 CoT + 强化学习对推理路径进行优化，提升推荐的可解释性和准确性
消融方法：去掉 R 模块，模型只做普通推荐，不生成强化的推理路径

线上A/B实验

用了1.29%的线上流量做实验, App停留时间显著提升0.159%

局限性

当前公开数据集仍存在质量限制，例如用户行为序列较短、物品集合规模受限等。这些限制阻碍了我们的方法在 Reasoning Activation 与 Reasoning Enhancement 模块中习得与工业级数据相当的高质量推理能力。
未来将构建一个更大规模的 benchmark（标准化评测基准），其中包含更长的用户行为轨迹和更丰富的物品类型