用于序列推荐的并行潜在推理

Jiakai Tang*
tangjiakai5704@ruc.edu.cn
GSAI，中国人民大学，北京，中国

Xu Chen†
xu.chen@ruc.edu.cn GSAI，
中国人民大学，北京，中国

Wen Chen†
chenyu.cw@alibaba-inc.com
阿里巴巴集团，北京，中国

Jian Wu
joshuawu.wujian@alibaba-inc.com 阿里巴巴集团，北京，中国

Yuning Jiang
mengzhu.jyn@alibaba-inc.com
阿里巴巴集团
北京，中国

Bo Zheng
bozheng@alibaba-inc.com
阿里巴巴集团
中国北京

arXiv:2601.03153v1 [cs.IR] 6 Jan 2026

摘要

从稀疏的行为序列中捕捉复杂的用户偏好，仍然是序列推荐领域的一项根本性挑战。近期的潜在推理方法通过多步推理扩展测试时计算，展现出良好前景；然而，这些方法完全依赖沿单一轨迹进行深度级扩展，随着推理深度增加，边际收益递减。为解决这一局限，我们提出并行潜在

推理（PLR），这是一种新颖框架，开创了通过同时探索多条不同推理轨迹实现宽度级计算扩展的途径。PLR 借助连续潜在空间中的可学习触发标记构建并行推理流，通过全局推理正则化保持各流之间的多样性，并通过推理流混合聚合自适应地整合多流输出。在三个真实世界数据集上的大量实验表明，PLR 的表现显著优于最先进的基线方法，同时保持实时推理效率。理论分析进一步验证了并行推理在提升泛化能力方面的有效性。我们的工作为突破现有深度扩展的局限、提升序列推荐中的推理能力开辟了新方向。

CCS 概念

• **信息系统 → 推荐系统**；**协同过滤**。

关键词

序列推荐、潜在推理、个性化推荐

ACM 引用格式：

Jiakai Tang、Xu Chen、Wen Chen、Jian Wu、Yuning Jiang 和 Bo Zheng。2018年。用于序列推荐的并行潜在推理。载于论文集

*此项工作是在阿里巴巴集团实习期间完成的。†两位作者均为通讯作者。

授权您免费为个人或课堂使用而制作本作品全部或部分内容的数字副本或纸质副本，前提是这些副本不得用于或分发以牟利或获取商业利益，并且副本须附有本声明及首页上的完整引文。对于本作品中由作者以外的其他方拥有版权的部分，必须尊重其版权。允许在注明出处的情况下摘录。若要以其他方式复制、重新发布、发布到服务器或分发至列表，则须事先获得具体许可和／或支付费用。请通过 permissions@acm.org 申请许可。会议缩写“*XX*”，纽约州伍德斯托克© 2018年版权归所有者／作者所有。出版权授权给 ACM。ACM ISBN 978-1-4503-XXXX-X/2018/06 https://doi.org/XXXXXXX.XXXXXXX

的 请务必根据版权确认邮件中的内容填写正确的会议标题（会议缩写“*XX*”）。ACM，纽约，纽约州，美国，12 页。https: // doi.org/XXXXXXX.XXXXXXX

1 引言

推荐系统已成为现代数字生态系统中不可或缺的基础设施，推动电子商务 [37, 42], 、流媒体平台 [12, 44], 和社交网络 [21, 56] 等不同领域的个性化内容发现与用户参与度。然而，推荐领域长期面临的一项挑战是用户行为数据的稀疏性，这从根本上限制了传统模型通过复杂逻辑推断来发掘用户潜在兴趣模式和偏好变化 [36, 38, 60]的能力。受大语言模型（LLMs） [5, 16, 30], 近期突破的启发——在这些模型中，融入推理机制已显著提升其在数学问题求解 [2, 43]和代码生成 [9, 22], 等高难度任务上的表现——序列推荐领域的一种新兴范式开始采用测试时计算策略 [63]。这种范式转变引入了显式和隐式的推理能力，将传统的浅层模式匹配方法（输入—输出）转变为复杂的推理驱动框架（输入—思考—输出），显著增强了模型的计算能力。

尽管前景可期，现有的推理增强型推荐方法仍面临若干关键局限。对于显式推理方法，仍存在两项根本性挑战：**（1）推理延迟开销**：生成冗长的推理链会带来高昂的计算成本，导致实时在线服务场景中的延迟难以接受。**（2）推理链定义歧义**：与数学推理或代码合成任务不同，在这些任务中，大语言模型（如 DeepSeek-R1 [11], Qwen3 [55], 和 Kimi1.5 [40]）会在后训练期间利用大量思维链（CoT）数据，并且这些数据可以通过客观真实标签进行严格验证。然而，推荐场景缺乏定义明确的推理轨迹。由于缺少专家标注的高质量“正确”推理链，在推荐场景中引导显式推理本质上颇具挑战。

相比之下，潜在推理增强推荐方法采用数据驱动的无标注方法，释放模型的推理潜力。然而，这些方法面临一个独特的瓶颈：当前技术主要依赖深度计算扩展——通过反复的自回归步骤加深推理过程。实证证据表明，进一步增加

推理深度带来的性能提升会逐渐减弱，甚至转为负面影响 [24, 35, 65]。我们认为，这一局限可能源于两个相互交织的因素：**(i)** 模型最初的推理方向可能并非最优；**(ii)** 随着推理链不断延长，错误累积会逐步降低思考质量，最终限制模型的推理能力上限。

在本研究中，为满足实时在线推理的需求，我们在重新思考其计算方法的同时，采用了潜在推理范式。我们不再追求沿单一轨迹进行更深层次的推理，而是从认知科学中汲取灵感 [6, 17], 。认知科学表明，人类通常会在得出全面结论之前，同时探索多条合理的解决路径。基于这一洞见，我们提出了一种新的计算扩展维度：宽度级推理增强。多流潜在推理使模型能够通过多样化的推理轨迹，捕捉用户动态且多方面的兴趣偏好，从而避免过早锁定于单一且可能欠佳的思维过程。然而，应用这一理念并非易事，原因在于以下挑战：**挑战 1：如何构建多流推理?** 不同于大语言模型中的离散解码（其中束搜索 [49, 51], 、自洽性 [1, 46], 和多数投票 [4, 45]等技术作用于语言标记序列），如何在连续潜在表示空间中有效构建多个推理流，仍不明确。**挑战 2：如何避免推理同质化?** 尽管宽度扩展提升了计算能力，但至关重要的是，要防止多个流趋同于相似的推理模式，否则会造成资源冗余消耗，却无法带来积极收益。**挑战 3：如何聚合不同的推理流?** 经过并行探索后，模型需要综合多条推理路径中的见解，形成统一的解决方案。朴素的聚合策略（例如平均池化）可能会稀释优质推理流的贡献，同时放大劣质推理流带来的干扰。

为应对这些挑战，我们提出了一种简单而有效的序列推荐并行潜在推理框架，称为 **并行潜在推理（PLR）**。具体而言，为了将序列编码器的潜在表示扩展为多条流，我们引入了可学习的触发词元 ，主动引导模型开展并行多流推理。这一简单设计有效解决了连续潜在空间中的流构建难题。此外，为了增强不同流之间以及每条流内不同推理步骤之间的差异性，我们引入了全局推理正则化 机制，以缓解同质化问题。为了汇总多个并行推理流的结果，我们设计了推理流混合 聚合模块，自适应地为不同流的输出赋权并加以组合，从而生成最终的序列表示。此外，为了提升模型的推理鲁棒性，我们提出了推理对比学习目标，以增强模型处理稀疏用户行为的能力。我们还提供了深入的理论分析，阐释所提方法的有效性，并为深化对推理增强推荐的理解提供见解。

我们的主要贡献总结如下：

- 我们率先探索了序列推荐中的宽度级计算扩展，以实现潜在推理，从而构建一种兼顾广度与深度推理的新型架构，同时保持实时推理效率。
- 我们提出了 PLR，这是一种模型无关的并行潜在推理框架。该框架利用可学习触发标记，在连续潜在空间中构建多样化的推理流，通过正则化全局推理模式来保留多样性，并通过推理流混合聚合多流输出。
- 在三个真实世界数据集上的大量实验表明，PLR 取得了显著提升，并为序列推荐树立了新的性能标杆，为潜在推理增强推荐研究开辟了新方向。

2 预备知识

2.1 问题定义

设 $\mathrm{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ 表示用户集，$\mathrm{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$ 表示物品集。对于每位用户 $u \in \mathrm{U}$，我们将其按时间顺序排列的交互序列记为 $\mathrm{S}_u = [v_1^u, v_2^u, \ldots, v_n^u]$，其中 $v_i^u \in \mathrm{V}$ 表示用户 u交互的第 i 个物品， n 表示序列长度。序列推荐任务旨在根据用户 u 的历史序列 S_u，预测用户 u 最有可能交互的下一个物品 v_{n+1}^u。形式上，该问题可以表述为学习一个函数 $f_\theta : \mathrm{S}_u \to \mathrm{V}$，使真实目标物品的条件概率最大化：

$$\theta^* = \arg\max_\theta \sum_{u \in \mathcal{U}} \log p(v_{n+1}^u | \mathcal{S}_u; \theta), \tag{1}$$

其中 θ 表示模型参数。在推理阶段，模型根据预测概率对所有候选物品进行排序，并推荐得分最高的前K 个物品。

2.2 潜在推理增强推荐

传统推荐方法依赖单次前向传播来编码序列表示，而固定的计算预算限制了其建模能力，尤其是在稀疏交互场景中。为了实现更细粒度的推理和更深入的表示学习，近期研究探索了潜在推理增强推荐系统 [8, 24, 35, 64], ，这些系统从传统的浅层输入-输出 范式转向多步输入-思考-输出 推理框架。

形式化地，给定输入序列 S_u，潜在推理增强模型首先通过编码器 f_{enc}将序列编码为潜在表示$\mathbf{h}_0$:

$$\mathbf{h}_0 = f_{\text{enc}}(\mathcal{S}_u; \theta_{\text{enc}}),$$

其中 θ_{enc} 表示编码器参数。随后，模型执行 T 个迭代推理步骤，逐步优化表示：

$$\mathbf{h}_t = f_{\text{rea}}(\mathbf{h}_{t-1}; \theta_{\text{rea}}), \quad t = 1, 2, \ldots, T,$$

其中， f_{rea} 是参数为 θ_{rea} 的推理模块（根据具体方法，参数可以与 θ_{enc} 共享，也可以不同）。最后，模型基于精炼表示$\mathbf{h}_T$生成预测。这一范式的核心原则是通过迭代推理增强模型的计算表达能力，从而提高模型在复杂任务上的准确率。

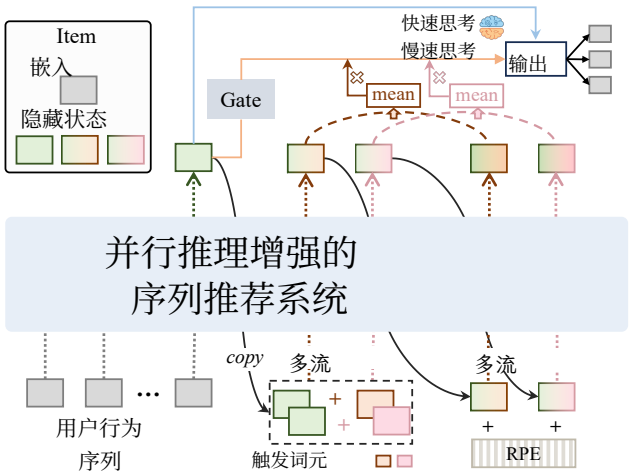


图1：并行潜在推理框架的整体架构。RPE 表示推理位置嵌入。

然而，对现有潜在推理增强方法的实证研究揭示了一个关键局限：在叠加多个推理步骤后（例如，ReaRec 中的 2 步 [35], 3、LARES 中的 4 步 [24],以及 LatentR 中的 1 步[3] [65]），模型的性能提升往往微乎其微，甚至出现负向提升，这种现象称为过度思考 [29, 53]。这一问题表明，仅通过增加深度进行计算扩展，可能不足以释放模型的全部潜力。要进一步释放推荐模型的潜在推理能力，还需要更先进的推理机制。

3 方法

本节介绍用于序列推荐的**并行潜在推理（PLR）**框架。整体架构如图1所示。我们首先在第3.1节介绍并行潜在推理骨干网络，该网络通过可学习触发标记，将当前仅沿深度进行的推理扩展为宽度级并行流。随后，第3.2节介绍三种关键优化机制：全局推理正则化，用于保持不同推理流之间的多样性；推理对比学习，用于增强推理鲁棒性；以及推理流混合聚合，用于自适应地整合多流输出。最后，第3.3节讨论完整的训练目标和双过程推理机制。

3.1 并行潜在推理骨干网络

本节介绍并行潜在推理（PLR）框架的核心架构。我们首先描述其基础组件：基于注意力的序列编码器、深度级潜在推理，以及我们提出的宽度级并行推理方法。

3.1.1 **基于注意力的序列编码。** 遵循主流序列推荐方法 [50, 62, 69],，我们采用 Transformer 架构 [41] 作为骨干网络。给定一个用户交互序列 $S_u = [v_1^u, v_2^u, \ldots, v_n^u]$，我们首先将每个项目嵌入为稠密向量表示。形式上，令 $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}|\times d}$ 表示物品嵌入矩阵，其中 d 为嵌入维度。

初始序列表示通过以下方式获得：

$$\mathbf{X}_0 = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n] + \mathbf{P},$$

其中，$\mathbf{e}_i = \mathbf{E}[v_i^u] \in \mathbb{R}^d$ 是物品 v_i^u 的嵌入，而 $\mathbf{P} \in \mathbb{R}^{n\times d}$ 表示位置编码，用于向模型注入序列顺序信息。随后，将 $\mathbf{X}_0$ 输入多层多头自注意力模块，以捕获序列中物品之间的复杂依赖关系。具体而言，对于第 l层，其计算公式为：

$$\mathbf{X}_l = \mathrm{MultiHead}(\mathbf{X}_{l-1}) + \mathbf{X}_{l-1},$$

其中，多头自注意力运算定义如下：

$$\mathrm{MultiHead}(\mathbf{X}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_H)\mathbf{W}^O,$$

每个注意力头的计算方式如下：

$$\mathrm{head}_h = \mathrm{Attention}(\mathbf{X}\mathbf{W}_h^Q, \mathbf{X}\mathbf{W}_h^K, \mathbf{X}\mathbf{W}_h^V),$$

其中，$\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d\times d_h}$ 是第 h个注意力头的可学习投影矩阵，$\mathbf{W}^O \in \mathbb{R}^{Hd_h\times d}$ 是输出投影矩阵，H是注意力头的数量。经过 L 个编码器层后，我们得到最终的编码表示 $\mathbf{H}_{\mathrm{enc}} = \mathbf{X}_L \in \mathbb{R}^{n\times d}$。

3.1.2 **深度级潜在推理。** 遵循近期的潜在推理方法（例如 ReaRec [35]），我们将传统的单次编码范式扩展为包含深度级多步推理计算的范式。我们不再直接使用编码器输出 $\mathbf{H}_{\mathrm{enc}}$ 作为最终的序列表示，因为受限的计算能力难以捕捉细粒度的用户兴趣；相反，我们引入额外的自回归推理步骤，以逐步细化潜在表示。

具体而言，我们提取序列中最后一个项目的表示作为初始推理状态：$\mathbf{h}_0 = \mathbf{H}_{\mathrm{enc}}[-1] \in \mathbb{R}^d$。随后，模型执行 T迭代推理步骤：

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{r}_t; \theta), \quad t = 1, 2, \ldots, T, \tag{2}$$

其中，f表示由 Transformer 层实现的推理模块，该模块与编码器共享参数 θ；$\mathbf{r}_t \in \mathbb{R}^d$表示步骤 t 对应的推理位置嵌入（RPE），用于区分推理阶段与编码阶段。

3.1.3 **宽度级并行推理。** 然而，仅进行深度方向的计算扩展仍容易受到若干关键限制的影响：初始推理方向欠佳，以及连续推理链中的误差累积。为缓解这些挑战，我们探索了一种新的计算扩展维度：宽度级并行潜在推理。其中的关键挑战在于，将连续的潜在状态向量扩展为潜在空间中的多个并行流。

为实现这一扩展，我们引入**可学习的触发词元**$\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_M\} \in \mathbb{R}^d$，以明确引导不同的推理流，其中 M表示并行推理流的数量。具体而言，我们采用一种简单的方法：将每个触发词元添加到初始推理状态$\mathbf{h}_0$ 中，从而获得 M 个不同的起始推理状态，形式化表示如下：

$$\mathbf{h}_{0,m} = \mathbf{h}_0 + \boldsymbol{\tau}_m, \quad m = 1, 2, \ldots, M. \tag{3}$$

这种设计使每条流都能从扰动后的初始状态出发，从而鼓励探索多样化的推理轨迹。

对于多个推理流之间的注意力机制，我们采用一种结合共享输入注意力的**流隔离因果推理**方法。形式上，对于推理步骤 t上的每条流 m，其计算过程可表示为：

$$\mathbf{h}_{t,m} = \text{Attention}(\mathbf{Q}_{t,m}, \mathbf{K}_{t,m}, \mathbf{V}_{t,m}) + \mathbf{h}_{t-1,m}, \tag{4}$$

其中：

$$\begin{aligned} \mathbf{Q}_{t,m} &= (\mathbf{h}_{t-1,m} + \mathbf{r}_t)\mathbf{W}^Q, \\ \mathbf{K}_{t,m} &= [\mathbf{X}_0; \mathbf{h}_{1,m}; \ldots; \mathbf{h}_{t-1,m}]\mathbf{W}^K, \\ \mathbf{V}_{t,m} &= [\mathbf{X}_0; \mathbf{h}_{1,m}; \ldots; \mathbf{h}_{t-1,m}]\mathbf{W}^V, \end{aligned} \tag{5}$$

其中，$[\cdot;\cdot]$ 表示沿序列维度进行拼接。该设计确保：**(i)** 每个流都关注共享的交互上下文，从而保留对原始输入信息的访问；**(ii)** 每个流中的推理遵循因果模式，即步骤 t 的状态仅关注同一流中的先前步骤；**(iii)** 不同流在推理过程中彼此隔离，从而能够独立探索多样的推理路径。

经过 T个推理步骤后，我们得到 M个精炼表示组$\{\mathbf{H}_1^{\text{rea}}, \mathbf{H}_2^{\text{rea}}, \ldots, \mathbf{H}_M^{\text{rea}}\}$，其中$\mathbf{H}_m^{\text{rea}} = \{\mathbf{h}_{1,m}, \mathbf{h}_{2,m}, \ldots, \mathbf{h}_{T,m}\}$对应第 m个流的推理轨迹。遵循ReaRec提出的集成推理学习（*ERL*）策略，我们对每个推理流中的多步输出进行平均池化，以获得流特定表示：

$$\mathbf{z}_m = \frac{1}{T}\sum_{t=1}^{T} \mathbf{h}_{t,m}, \quad m = 1, 2, \ldots, M. \tag{6}$$

随后，通过聚合所有推理流的输出 $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$，得到最终的序列表示。在第 3.2.3 节中，我们将介绍一种自适应推理流混合方法，用于组合多流推理结果。

3.2 多流推理优化

本节介绍三种用于优化并行推理过程的关键机制：通过全局推理正则化保持多样性，通过推理对比学习增强鲁棒性，以及通过推理流混合实现自适应聚合。

3.2.1 全局推理正则化。 为防止多个推理流坍缩为同质模式，即不同流呈现出高度相似的推理状态，我们引入了一种全局推理正则化 机制。具体而言，我们通过双向 Kullback-Leibler（KL）散度正则化，对所有推理状态（包括流内和流间）施加多样性约束。

形式化地，令$\mathbf{h}_{t,m}$ 表示流 m 在步骤 t 的隐藏状态。我们计算每个推理状态在物品词表上的分布，如下所示：

$$p_{t,m}(v) = \text{softmax}(\mathbf{h}_{t,m}^\top \mathbf{E}[v]),$$

其中，$\mathbf{E}[v]$ 是项目 v的嵌入。全局推理正则化损失定义如下：

$$\mathcal{L}_{\text{KL}} = \frac{1}{TM(TM-1)} \sum_{(t,m)} \sum_{(t',m')\neq(t,m)} \text{KL}(p_{t,m} \| p_{t',m'}), \tag{7}$$

其中，求和遍历所有 $T \times M$ 位置上彼此不同的推理状态对。该约束鼓励：**(i)**在每个流内，不同的推理步骤逐渐探索不同的推理方向，以捕捉高阶特征交互；**(ii)** 在不同流之间，并行推理路径彼此分化，以发现多样的序列模式。通过在全局范围内惩罚所有推理状态之间的相似性，该机制可同时缓解流内和流间同质性。

3.2.2 推理对比学习。 为增强模型的推理鲁棒性，使其能够更好地从稀疏交互中捕捉用户偏好，我们设计了推理对比学习（*RCL*）目标。具体而言，我们沿表示和交互这两个正交维度引入丢弃策略，以构造多样的对比视图。

在 Transformer 的前向传播过程中，我们同时对隐藏表示（丢弃率为 p_{rep}）和注意力分数矩阵（丢弃率为 p_{attn}）应用随机丢弃。通过使用不同的丢弃掩码进行两次独立的前向传播，我们得到两个增强版本的推理输出：$\{\mathbf{h}_{1\,1}^{(1)}, \ldots, \mathbf{h}_{T}^{(1),M}\}$ 和 $\{\mathbf{h}_{1\,1}^{(2)}, \ldots, \mathbf{h}_{T}^{(2),M}\}$，其中上标表示不同的丢弃掩码。

为确保两个增强视图之间的一致性，我们采用批内对比学习目标 [13, 23, 32]。具体而言，对于流 m 在推理步骤 t上的表示，我们将同一用户在两个丢弃视图中的对应推理状态视为正样本对，同时将同一批次中其他用户的状态视为负样本。给定一个包含 B 个用户的批次，流 m 在步骤 t 上的对比损失定义如下：

$$\mathcal{L}_{\text{RCL}}^{t,m} = -\frac{1}{B}\sum_{i=1}^{B}\left(\log\frac{\exp(s_{ii}^{(1,2)}/\tau)}{\sum_{j=1}^{B}\exp(s_{ij}^{(1,2)}/\tau)} + \log\frac{\exp(s_{ii}^{(2,1)}/\tau)}{\sum_{j=1}^{B}\exp(s_{ij}^{(2,1)}/\tau)}\right),$$

其中，$s_{ij}^{(1,2)} = \frac{\mathbf{h}_{t,m,i}^{(1)\top}\mathbf{h}_{t,m,j}^{(2)}}{\|\mathbf{h}_{t,m,i}^{(1)}\|\,\|\mathbf{h}_{t,m,j}^{(2)}\|}$表示不同视图之间的余弦相似度（$s_{ij}^{(2,1)}$同理），$\tau$是温度超参数。整体推理对比损失聚合了所有推理步骤和推理流：

$$\mathcal{L}_{\text{RCL}} = \frac{1}{TM}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathcal{L}_{\text{RCL}}^{t,m}. \tag{8}$$

该目标鼓励模型学习与任务相关的表示，使其在不同增强视图之间保持不变，从而提升泛化能力和鲁棒性。

3.2.3 推理流混合聚合. 为整合多个推理流的输出，直接采用均匀平均可能会使较优的推理输出受到较差输出的干扰，从而导致最终表示不理想。为此，我们提出了一种推理流混合（MoRS）聚合方法，可对不同推理流进行自适应加权。

受门控机制的启发， [26, 31, 52], 我们引入一个轻量级门控网络来计算各流的特定重要性权重。具体而言，我们将编码表示$\mathbf{h}_0$ （从$\mathbf{H}_{\text{enc}}$中提取）输入门控网络：

$$\mathbf{g} = \text{softmax}(\mathbf{W}_g\mathbf{h}_0 + \mathbf{b}_g),$$

其中$\mathbf{W}_g \in \mathbb{R}^{M\times d}$和$\mathbf{b}_g \in \mathbb{R}^M$是可学习参数，$\mathbf{g} = [g_1, \ldots, g_M]^\top \in \mathbb{R}^M$表示归一化的门控权重

对于 M 个流。最终输出计算如下：

$$\mathbf{z}_{\text{rea}} = \sum_{m=1}^{M} g_m \mathbf{z}_m, \tag{9}$$

其中$\mathbf{z}_m$ 是来自流 m 的池化表示（见公式（6））。

双过程推理。受人类认知科学中的双过程理论启发，该理论 [18, 33],认为人类认知同时涉及快速直觉思维（系统1）和慢速审慎推理（系统2），我们在推理过程中结合编码器输出（快速思考）与推理输出（慢速思考）。具体而言，最终的序列表示为：

$$\mathbf{z}_{\text{final}} = \mathbf{h}_0 + \mathbf{z}_{\text{rea}}, \tag{10}$$

其中$\mathbf{h}_0$ 直接表示来自编码器的快速思维组件，而$\mathbf{z}_{\text{rea}}$ 则汇集了多流推理审慎提炼出的见解。这种设计结合快速模式识别与深入逻辑推断，以进行下一项预测。

3.3 训练目标

整体训练目标将下一项预测（*NIP*）损失（式（1））与上述全局正则化项（式（7））和对比学习目标（式（8））结合起来：

$$\mathcal{L} = \mathcal{L}_{\text{NIP}} + \mathcal{L}_{\text{RCL}} + \lambda \mathcal{L}_{\text{KL}}, \tag{11}$$

其中 λ 是用于平衡正则化强度的超参数。值得注意的是，为了充分利用双过程推理框架，我们在训练和推理时采用了不同的策略：

🔥 **使用推理输出学习**。训练期间，我们仅使用慢速思考输出$\mathbf{z}_{\text{rea}}$（式（9））执行下一项预测任务。这一设计促使模型培养真正的推理能力，而不是走捷径、仅依赖快速思维编码器输出$\mathbf{h}_0$。通过在训练期间阻止模型直接访问编码器表示，我们确保模型学会进行审慎的多步推理。

❄ **双过程集成推理**。推理期间，我们结合快速思考和慢速思考的输出，使用$\mathbf{z}_{\text{final}}$（式（10））对候选物品进行排序。这种双过程集成利用了两种推理模式互补的优势：快速思维组件$\mathbf{h}_0$ 基于直接编码实现快速模式识别，而慢速思维组件$\mathbf{z}_{\text{rea}}$ 则通过审慎的多流推理提供更精细的洞见。这两种模式相辅相成，从而带来更稳健、更准确的推荐。

4 理论分析

本节从理论基础出发，解释为何并行多流推理优于仅深度基线。我们从三个角度进行分析：**(1)** 集成理论阐明了多样化推理流如何降低预测误差（§4.1）；**(2)** 动力系统分析揭示了深度导向细化与宽度导向多样性之间的根本张力（§4.2）；**(3)**门控理论解释了自适应聚合如何利用流专门化（§4.3）。这些结果共同为设计有效的并行推理架构提供了切实可行的见解。完整的理论证明见附录A。

4.1 为什么多样性能够降低误差

我们首先确立组合多个推理流的理论依据。令 $\hat{p}_m(v|S_u)$ 表示来自流 m 的预测分布，令 $\bar{p}(v|S_u) = \frac{1}{M}\sum_{m=1}^{M} \hat{p}_m(v|S_u)$ 表示均匀加权的集成预测。

定理4.1（集成误差分解）。定义集成损失$\mathcal{L}_{ens} = \mathbb{E}_{(S_u,v)\sim\mathcal{D}}[-\log \bar{p}(v|S_u)]$ 以及平均个体损失 $\bar{\mathcal{L}}_{ind} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}[-\log \hat{p}_m(v|S_u)]$。则：

$$\mathcal{L}_{ens} \le \bar{\mathcal{L}}_{ind} - \mathbb{E}_{S_u\sim\mathcal{D}}[\mathcal{I}(S_u)],$$

其中，专门化收益$\mathcal{I}(S_u) \ge 0$ 量化了多样性带来的收益，当且仅当所有$\hat{p}_m$ 都相同时，等号成立。

证明（附录A.1）利用了凹对数函数的詹森不等式。直观而言，当各流产生多样化的预测时，集成会平均掉个体误差，从而使损失低于平均个体损失。

为了将表示空间中的多样性与预测空间中的多样性联系起来，我们建立如下结果：

命题4.2（多样性-专门化联系）。将表示多样性定义为 $D(S_u) = \frac{1}{M(M-1)}\sum_{m\neq m'} \|\mathbf{z}_m - \mathbf{z}_{m'}\|^2$。在线性评分且项目嵌入有界的条件下，专门化带来的收益满足：

$$\mathcal{I}(S_u) \ge c \cdot D(S_u),$$

其中，$c > 0$ 取决于嵌入几何。

这表明，表示多样性（由我们的触发 token 和正则化直接控制）会转化为预测多样性，而预测多样性反过来会降低集成误差。

4.2 精炼-多样性权衡

尽管定理4.1表明了多样性的价值，但迭代推理带来了一个根本矛盾：随着步骤增加，精炼质量会提高，但多样性可能会下降。我们利用压缩映射理论将这一权衡形式化。

假设 4.3（利普希茨连续性）。推理函数 f 是 L-利普希茨连续的：

$$\|f(\mathbf{h},\mathbf{r};\theta) - f(\mathbf{h}',\mathbf{r};\theta)\|_2 \le L\|\mathbf{h}-\mathbf{h}'\|_2.$$

Transformer 中的层归一化会自然地约束输出范数，并带来收缩行为。对于较浅的架构（例如我们实现中的 2 层 Transformer），利普希茨常数 L 通常有一个较小的上界。

定理4.4（迭代下的多样性衰减）。在假设 *4.3* 下，令 $D(t) = \frac{1}{M(M-1)}\sum_{m\neq m'} \|\mathbf{h}_{t,m} - \mathbf{h}_{t,m'}\|^2$ 表示推理步骤 t处的多样性。则：

$$D^{(T)} = L^{2T} D^{(0)} + o(L^{2T}).$$

当 L的上界为小于 *1* 的常数时，多样性会呈指数衰减：$D(T) = \exp(-2\gamma T)D^{(0)}$其中 $\gamma := -\log L > 0$。

证明（附录 A.3）可通过迭代应用利普希茨性质得出。这一结果揭示了仅深度缩放最终为何会失效：过多的推理步骤会使所有流都收敛到同一个不动点，从而消除多样性带来的益处。

推论 4.5（精炼-多样性权衡）。结合定理 *4.1* 和 *4.4*，第 T 步的集成损失满足：

$$\mathcal{L}_{ens}^{(T)} \leq \bar{\mathcal{L}}_{ind}^{(T)} - c \cdot e^{-2\gamma T} D^{(0)},$$

其中 $\bar{\mathcal{L}}^{(T)} ind$ 会随着 T（质量提升）而降低，但专门化收益会呈指数衰减（多样性损失）。

这一形式化解释了经验观察到的过度思考现象：起初，增加 T 会使 $\bar{\mathrm{L}}^{(T)}$ ind 降低得比多样性衰减更快；最终，精炼带来的边际收益递减将被多样性损失抵消。

4.3 门控收益

前述分析假设采用均匀加权。现在，我们来刻画学习式门控的收益及其对泛化的影响。

定理4.6（基于互信息的门控收益）。令 $\tilde{p}(v|S_u) = \sum_{m=1}^{M} w_m(S_u) \hat{} p_m(v|S_u)$ 表示门控集成。则：

$$\mathcal{L}_{gated} \leq \mathcal{L}_{uniform} - \mathbb{E}_{S_u}[I(Z;V|S_u)],$$

其中，$Z \sim \mathbf{w}(S_u)$ 表示流选择，$V \sim p^*(v|S_u)$ 表示目标，而 $I(Z;V|S_u)$ 表示条件互信息。

这表明，当不同流各有所长时，门控能够带来收益：如果不同流在不同的序列视角下表现出色，互信息 $I(Z;V|S_u) > 0$，自适应加权便优于均匀平均。

5 相关工作

5.1 序列推荐

序列推荐旨在根据按时间顺序排列的用户行为序列预测用户的下一次交互，是推荐系统中的主流范式。早期研究探索了多种用于序列建模的神经网络架构，包括循环神经网络 [7, 10], 卷积网络 [39, 54], 以及基于 Transformer 的模型 [19, 34]。SASRec [19] 率先将自注意力机制应用于捕捉序列依赖关系，而 BERT4Rec [34] 则通过掩码物品预测将其扩展为双向建模。然而，这些基于 ID 的方法高度依赖高质量的交互数据，且难以应对稀疏性和冷启动场景。

为解决这些局限，近期研究引入了多模态辅助信息，使物品表示不再局限于简单的 ID [15, 48, 61]。UniSRec [15] 通过文本描述学习通用物品表示，并借助统一的文本建模实现跨域迁移。MoRec [61] 全面研究了多模态物品表示与传统基于ID的序列之间的性能差距，展现了多模态融合的潜力。尽管取得了这些进展，传统序列方法仍采用浅层的输入-输出范式，并受固定计算预算的限制，因而难以对用户偏好进行复杂推理。

5.2 推理增强推荐

大语言模型（LLMs）的非凡成功 [27, 28, 66],，尤其是通过思维链（CoT）提示 [47],，彻底改变了人工智能领域的格局，使其从浅层模式匹配转向由深度推理驱动的生成。这一突破不再单纯依赖编码能力，而是在测试时扩展计算，由此开启了推理增强型推荐研究的新时代。

显式推理方法。第一类研究利用大语言模型的逻辑能力，为推荐任务生成显式推理链。RecGPT系列 [57, 58] 和 OneRec系列 [25, 68]利用大语言模型将对用户兴趣的洞察转化为语言，从而通过可解释的推理路径提高预测准确率。DeepRec [67] 引入多轮交互，以迭代精炼推荐结果；而 R^{2}ec [59] 则在自回归大语言模型框架内统一推理与推荐。然而，显式推理面临两项关键挑战：**(i)** 推荐场景缺乏定义明确且可验证的推理链（不同于具有客观真实标签的数学或编程任务）；以及**(ii)** 冗长的令牌生成导致推理延迟过高，使这些方法难以应用于实时工业部署。

潜在推理方法。为克服这些局限，一种新兴范式采用数据驱动的潜在推理，在连续表示空间中进行，从而无需显式的思维链标注，同时保持较低的延迟。ReaRec [35] 率先探索这一方向，通过引入多步自回归推理，并采用集成（ERL）和渐进式（PRL）学习策略。LARES [24]提出将编码（预处理块）模块与推理（核心块）模块进行架构解耦，并采用自监督预训练和强化学习来提升推理质量。OnePiece [8] 升级为逐块潜在推理，并引入多任务渐进式监督，以提升推理质量。LatentR^3 [65] 则将这一范式扩展至大语言模型骨干网络，并采用基于强化学习的优化。尽管现有潜在推理方法前景可期，但它们都仅通过序列推理步骤进行深度层面的计算扩展，因而面临边际收益递减和过度思考问题。与此不同，我们的工作率先探索宽度层面的 并行推理，同时探究多条不同的推理轨迹，以进一步提升性能，同时保持实时效率。

6 实验

在本节中，我们开展了广泛的实验与分析，以证明所提出的 PLR 框架的优越性。我们旨在回答以下研究问题：（**研究问题1**）PLR 相较于最先进基线方法的表现如何？（**RQ2**）PLR 中各组件的贡献是什么？（**RQ3**）关键超参数如何影响 PLR 的性能？（**RQ4**）通过对 PLR 推理行为的深入分析，我们能获得哪些洞见?

6.1 实验设置

6.1.1 数据集。我们在 Amazon Review 2023 数据集 [14] 的三个领域开展实验：**CD与黑胶唱片**、**电影与电视**以及**视频与游戏**。按照推荐系统 [35, 67], 中的常见做法，我们将评分高于 3 的交互视为正向用户交互。我们依据官方时间戳按时间顺序划分每位用户的交互序列，这更能反映真实的工业场景，也有助于公平比较各基线方法的性能。对于 CD与黑胶唱片和视频与游戏

表1：数据集统计信息。

数据集	#用户	#项目	#交互	稀疏度
CD与黑胶唱片	35,238	87,969	943,399	99.97%
电影与电视	51,566	101,114	1,314,578	99.97%
视频与游戏	90,678	22,933	728,661	99.96%

数据集，我们剔除了交互次数少于 10 次的用户。数据集的详细统计信息见表1。

特征构建： 对于基于*ID*的序列模型（例如，SAS-Rec、BERT4Rec），我们使用数据集提供的多层级类别体系、店铺ID和物品ID构建物品表示。对于基于文本的模型（例如，UniSRec），我们将物品标题、层级类别和店铺名称拼接起来，生成文本描述。随后使用BGE嵌入模型 [3] 对这些文本序列进行编码，以获得稠密句子嵌入。

6.1.2 评估指标。 我们采用推荐系统领域广泛使用的两种排序指标来评估模型的个性化建模能力：**K位归一化折损累积增益**（NDCG@K，简称N@K）和**K召回率**（R@K）。我们报告 $K \in \{10, 20\}$对应的结果。NDCG@K通过对数折损考虑相关物品在排序列表中的位置，而Recall@K则衡量成功检索到的真实物品占前K项推荐的比例。

6.1.3 基线方法。 为了全面评估 PLR 的有效性，我们选取了三种具有代表性的骨干架构和三种最先进的潜在推理增强方法。

基础模型。 我们选择了三种广泛采用的序列推荐模型作为骨干架构：

• **SASRec** [19]：一种开创性的基于 Transformer 的序列推荐模型，采用单向多头自注意力来捕捉用户行为中的序列依赖关系。

• **BERT4Rec** [34]：SASRec 的扩展模型，利用结合 Cloze 任务（掩码项目预测）的双向自注意力进行序列建模，从而获得更丰富的上下文表示。

• **UniSRec** [15]：一种基于文本的序列推荐模型，通过文本特征和混合专家（MoE）适配器学习通用物品表示，以实现跨域迁移。

潜在推理增强方法。 为确保在推理延迟和模型容量相近的条件下进行公平比较，我们仅关注潜在推理基线方法。由于显式推理方法和基于大语言模型的方法在实时在线服务中的计算开销过高，且参数规模差异巨大，因此不予纳入。

• **ReaRec-ERL** [35]：首个用于序列推荐的潜在推理框架，首次在潜在空间中引入多步自回归计算，以探索测试时推理。ERL（集成推理学习）通过平均池化聚合所有推理步骤的输出。

• **ReaRec-PRL** [35]：ReaRec 的一个变体，采用 PRL（渐进式推理学习），逐步优化表示，并仅使用最后一步的输出进行预测。

• **LARES** [24]：一种两阶段推理架构，由用于编码的预处理块和用于推理的核心块组成。LARES 结合了自监督预训练和基于强化学习的后训练，以增强推理能力。

我们在三种骨干架构上实现了不同的潜在推理方法，以进行全面评估。

6.1.4 实现细节. 所有实验均使用 PyTorch 框架，在 NVIDIA A100 GPU 上进行。为确保所有模型之间的公平比较，我们将嵌入维度统一设为 $d = 256$，批次大小统一设为 $B = 2048$。我们使用 Adam优化器 [20]，学习率为 $1e^{-3}$，并采用 GeLU激活函数。用户交互序列截断至最多 50 个物品。对于基于文本的模型，我们使用主成分分析（PCA），将 BGE 编码表示从原始维度降至 768 维，同时保留信息量最高的特征。

对于 PLR，我们通过网格搜索确定最优超参数：推理流的数量 M和推理步骤数 T在 {1, 2, 3, 4,5}中搜索；KL正则化权重 λ在 $\{1e^{-3}, 1e^{-2}, 1e^{-1}, 1\}$中调整；丢弃率 p_{rep}和 p_{attn}设为相同值，并在 {0.1, 0.2, 0.3, 0.4, 0.5,0.6}中搜索；温度 τ固定为 1。我们根据验证集NDCG@10采用早停策略，耐心值设为 10 个训练轮次。为提高推理效率，我们采用 ***KV*缓存** 技术，在所有推理流和推理步骤中存储并复用输入序列编码得到的键值对，从而显著减少多流并行推理过程中的冗余计算开销。

6.2 总体性能（研究问题1）

表2展示了三个数据集上的总体性能比较。我们总结了以下几项关键观察：

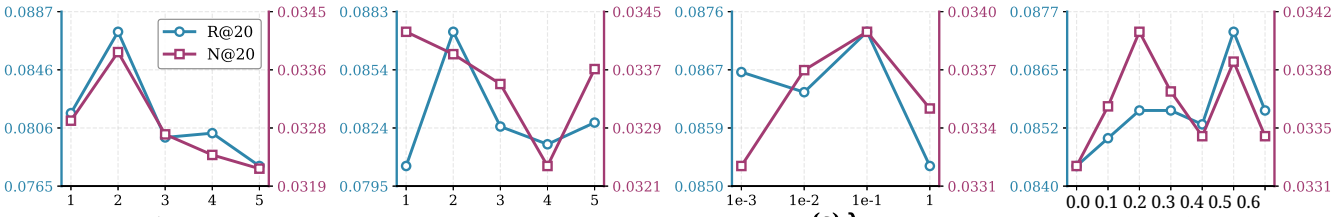
(1)推理增强方法在大多数设置下都优于基础模型，验证了测试时计算扩展的有效性。在CD与黑胶唱片数据集上使用SASRec时，ReaRec-ERL将Recall@10提高了5.15%，将Recall@20提高了2.23%。在电影与电视数据集上使用BERT4Rec时，ReaRec-ERL的Recall@10和NDCG@10分别大幅提升了16.67%和13.49%。这些提升验证了在推理过程中分配额外计算资源能够实现更深入的推理，这对稀疏数据集尤其有益。

(2) 在仅深度推理方法中，ReaRec-ERL通常在大多数场景下表现最佳。例如，在CD与黑胶唱片数据集上使用SASRec时，ReaRec-ERL的Recall@10达到0.0572，NDCG@10达到0.0269，优于ReaRec-PRL和LARES。不过，这些方法之间的性能差异仍相对有限，表明仅深度推理存在固有局限：单纯增加推理深度会带来边际收益递减。

(3) PLR 在大多数指标上均显著优于所有基线方法。在采用 SASRec 的 CDs & Vinyl 数据集上，PLR 的 Recall@20 较最佳基线方法 ReaRec-ERL 提升了 12.07%；在 UniSRec 上，Recall@10 的提升更为显著，达到 14.91%。尽管不同数据集和指标上的提升幅度有所差异，PLR 在大多数场景中均表现出色，尤其是在 CDs & Vinyl 等较稀疏的数据集上，提升幅度超过 10%。这些结果表明，宽度级并行推理能够捕捉多样化的用户兴趣，弥补单轨迹深度推理的不足。至关重要的是，这些提升仅带来极小的推理开销（表 4），使 PLR 能够应用于实时系统。

表2：不同方法在三个数据集上的性能比较。最佳结果以粗体标出，次佳结果带下划线。“提升”表示PLR相对于最佳基线方法的相对提升。*表示使用 $p < 0.05$ 进行配对t检验后结果具有统计显著性。

方法	CD与黑胶唱片				电影与电视				视频与游戏			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
SASRec												
Base	0.0544	0.0762	0.0250	0.0305	0.0502	0.0664	0.0230	0.0270	0.0624	0.0937	0.0280	0.0359
ReaRec-ERL	0.0572	0.0779	0.0269	0.0321	0.0502	0.0680	0.0237	0.0282	0.0660	0.0953	0.0315	0.0388
ReaRec-PRL	0.0553	0.0748	**0.0273**	0.0323	0.0504	0.0650	0.0244	0.0280	0.0630	0.1003	0.0311	0.0405
LARES	0.0555	0.0757	0.0260	0.0310	0.0504	0.0645	0.0232	0.0267	0.0648	0.0968	0.0314	0.0395
PLR（我们的方法）	**0.0604**	**0.0873**	0.0272	**0.0339**	**0.0532**	**0.0701**	**0.0250**	**0.0293**	**0.0676**	**0.1033**	**0.0326**	**0.0416**
提升	+5.59%*	+12.07%*	-0.37%	+4.95%*	+5.56%*	+3.09%*	+2.46%*	+3.90%*	+2.42%*	+2.99%*	+3.49%*	+2.72%*
BERT4Rec												
Base	0.0555	0.0745	0.0260	0.0308	0.0444	0.0592	0.0215	0.0253	0.0622	0.0912	0.0302	0.0376
ReaRec-ERL	0.0589	0.0793	0.0285	0.0337	0.0518	0.0664	0.0244	0.0281	**0.0680**	0.1017	0.0334	0.0418
ReaRec-PRL	0.0541	0.0782	0.0272	0.0333	0.0502	0.0648	0.0235	0.0271	0.0629	0.0989	0.0301	0.0392
LARES	0.0513	0.0759	0.0256	0.0318	0.0502	0.0652	0.0232	0.0270	0.0595	0.0960	0.0319	0.0411
PLR（我们的方法）	**0.0626**	**0.0827**	**0.0293**	**0.0344**	**0.0523**	**0.0685**	**0.0247**	**0.0288**	0.0656	**0.1018**	**0.0343**	**0.0435**
提升	+6.28%*	+4.29%*	+2.81%*	+2.08%*	+0.97%	+3.16%*	+1.23%	+2.49%*	-3.53%	+0.10%	+2.69%*	+4.07%*
UniSRec												
Base	0.0419	0.0638	0.0195	0.0250	0.0557	0.0724	0.0249	0.0291	0.0656	0.1024	0.0339	0.0432
ReaRec-ERL	0.0456	0.0689	0.0208	0.0266	0.0567	0.0782	0.0252	0.0307	0.0673	0.1006	0.0351	0.0435
ReaRec-PRL	0.0422	0.0680	0.0195	0.0260	0.0557	0.0754	0.0252	0.0303	0.0688	0.1032	0.0340	0.0427
LARES	0.0434	0.0680	0.0194	0.0255	0.0548	0.0765	0.0254	0.0309	0.0687	0.1029	**0.0353**	**0.0439**
PLR（我们的方法）	**0.0524**	**0.0748**	**0.0223**	**0.0279**	**0.0620**	**0.0816**	**0.0271**	**0.0321**	**0.0695**	**0.1054**	0.0341	0.0431
提升	+14.91%*	+8.56%*	+7.21%*	+4.89%*	+9.35%*	+4.35%*	+6.69%*	+3.88%*	+1.02%	+2.13%*	-3.40%	-1.82%



(a) 流数量 (b) 步数 **(c) λ** (d) 丢弃法

图 2：基于 SASRec 骨干网络的 CDs & Vinyl 数据集参数敏感性分析

e.

表 3：消融实验。实验基于 SASRec，在 CD与黑胶唱片和 Video & Games 数据集上进行。

方法	CD与黑胶唱片		视频与游戏	
	R@20	N@20	R@20	N@20
不使用 MoRS	0.0785	0.0321	0.0997	0.0415
不使用 RCL	0.0782	0.0322	0.0970	0.0387
不使用 KL	0.0853	0.0334	0.0979	0.0380
PLR（完整模型）	**0.0873**	**0.0339**	**0.1033**	**0.0416**

6.3 消融实验（RQ2）

为验证 PLR 各个组件的有效性，我们在两个具有代表性的数据集（CDs & Vinyl 和 Video & Games）上，使用 SASRec 骨干网络进行了消融实验。表 3 展示了四种消融变体的结果：**(i) 不使用 MoRS** 将自适应推理流混合聚合替换为均匀平均池化；**(ii) 不使用 RCL** 移除了推理对比学习目标；而**(iii) 不使用 KL** 则去除了全局正则化。

结果表明，各组件都对 PLR 的性能作出了有意义的贡献。移除 MoRS 会导致性能显著下降（在 CD与黑胶唱片上，Recall@20 从 0.0873 降至 0.0785），验证了自适应门控能够有效识别更优的推理流，这与定理4.6一致。移除 RCL 会导致性能下降，在视频与游戏上尤为明显（Recall@20 从 0.1033 降至 0.0970），证实了它在增强鲁棒性方面的作用。移除 KL 也会导致两个数据集上的性能下降，不过降幅相对较小，这表明多样性正则化能够带来稳定但适度的收益。总体而言，所有组件均有积极贡献，其中 MoRS 和 RCL 的效果更为显著。

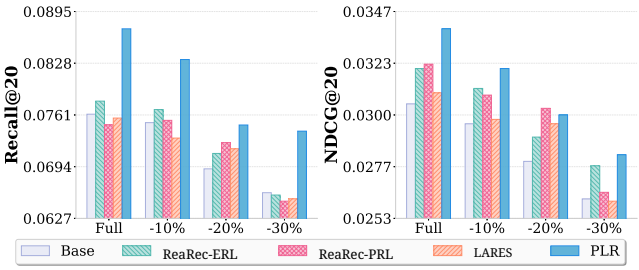


图3：CD与黑胶唱片 数据集上的鲁棒性分析。

6.4 敏感性分析（RQ3）

我们对关键超参数进行了敏感性分析，以了解它们对 PLR 性能的影响。图 2 展示了在采用 SASRec 骨干网络的 CD与黑胶唱片数据集上的结果，评估指标为 NDCG@20 和 Recall@20。

推理流数量（M）。如图2(a)所示，性能在 $M = 2$时达到峰值，NDCG@20和Recall@20均为最优。将 M 增加到2以上会导致性能下降，这可能是因为过多的流会引入冗余推理路径，削弱表现更优的流所作出的贡献；此外，有限的训练数据不足以有效优化众多专门化的流。我们将 $M = 2$ 设为默认配置。

推理步骤数（T）。图2(b)显示，性能在 $T = 2$时达到峰值，随后随着推理加深而下降。这与我们的理论分析（定理4.4）一致：适度的推理步骤有助于充分精炼，而过深则会因利普希茨收缩导致多样性衰减。最优的 $T = 2$能够兼顾精炼质量与多样性保持，这证实了仅深度方法中的过度思考现象。

正则化权重（λ）。图 2(c) 显示，在 $\lambda = 0.1$ 时性能最佳。正则化不足（$\lambda < 0.1$）无法确保多样性，导致各流坍缩为同质模式。正则化过强（$\lambda > 0.1$）则会产生负面影响，使不同推理流彼此相距过远，从而阻碍模型收敛到高质量解。

丢弃率（p）。图 2(d) 表明，适中的丢弃率 $p \in [0.2, 0.5]$ 可带来最佳性能。丢弃率较低（$p <$ 0.2）无法为对比学习提供足够的数据增强，而丢弃率过高（$p > 0.5$）则会因移除过多信息而破坏推理过程。最佳范围有效平衡了增强多样性与信息保留。

6.5 深入分析（RQ4）

6.5.1 鲁棒性分析。为评估模型在极端数据稀疏性下的鲁棒性，我们从 CD与黑胶唱片测试集中的用户序列里随机移除 10%、20% 和 30% 的交互。图 3 显示，尽管所有模型的性能都会随着稀疏度增加而下降，但推理增强方法表现出

与基础模型相比，鲁棒性更强。当缺失率为 20% 时，基线 SASRec 的性能显著下降，而推理方法则保持相对稳定的性能。在所有方法中，PLR 的鲁棒性最强：缺失率为 30% 时，PLR 的 NDCG@20 仍保持在 0.028 左右，显著优于其他方法。这种更强的鲁棒性源于 PLR 的并行推理机制，其中多个推理流

表4：不同方法的效率比较

指标	SASRec	ReaRec	PLR
浮点运算次数（$\times 10^8$）	1.3448	1.3810	1.4150
提升	-	+2.69%	+5.22%
延迟（秒）	0.7844	0.8279	0.8299
成本增加	-	+5.55%	+5.80%

使模型能够探索多样化的路径，并在某些交互模式不可用时弥补丢失的行为信号。这些结果表明，PLR 在真实世界推荐场景中具有实际价值，因为这类场景中的用户行为数据天然稀疏且不完整。

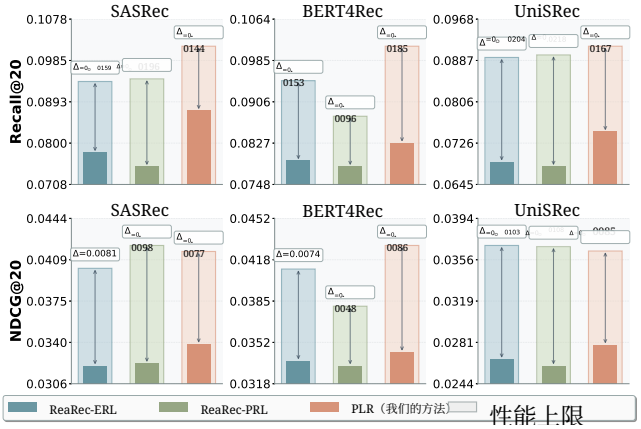


图4：不同骨干网络在 CDs & Vinyl 数据集上的性能上限分析。

6.5.2 性能上限分析。 为了解不同推理方法的理论潜力，我们通过独立评估每个推理步骤的输出，并为每个用户样本选择表现最佳的步骤（预言机选择），来分析性能上限。图4对比了 CDs & Vinyl 数据集上三种骨干网络的当前性能（采用固定聚合策略）与预言机性能上限。结果表明，PLR 和 ReaRec 系列的预言机性能上限相当（例如，NDCG@20 $\in [0.0402, 0.0416]$），说明二者的理论上限相近。然而，PLR 的当前性能与预言机性能之间的差距明显更小，这表明其并行多流设计能够更有效地利用推理能力。相比之下，仅深度方法的性能差距更大，说明适当的步骤选择更加关键，但若缺乏宽度级探索，步骤选择也更具挑战性。这表明，PLR 的深度与宽度协同设计提供了一条更易实现近最优性能的路径，在保持具有竞争力的性能上限潜力的同时，减少了对预言机级步骤选择的依赖。

6.5.3 效率分析。 为评估计算开销，我们比较了采用 SASRec 骨干网络的 CDs & Vinyl 数据集上每个样本的浮点运算次数和推理延迟。如表4所示，与基线 SASRec 相比，PLR 的浮点运算次数增加了 5.22%，延迟增加了 5.80%。考虑到性能的大幅提升，这是一个非常高效的权衡。值得注意的是，PLR 的计算开销与 ReaRec 相当

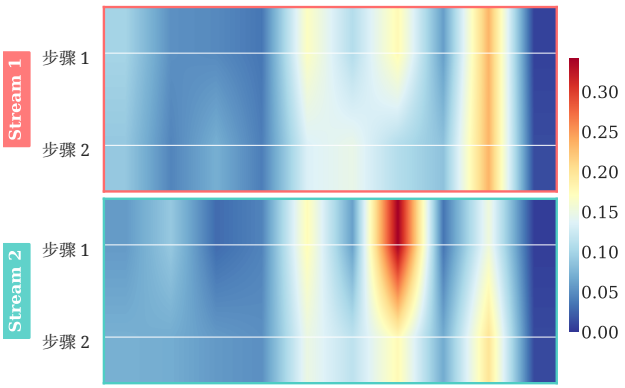


图5：注意力可视化示意图。

（浮点运算次数增加 2.69%，延迟增加 5.55%），尽管它同时探索了多个推理流。这种效率源于两项关键设计选择：**(i)** KV缓存机制，在所有流和步骤中复用来自输入和先前推理状态的共享编码键值对，避免冗余计算；**(ii)**向量化并行多流推理，利用现代 GPU 并行性同时处理 M 个流，避免延迟随流数成比例增长。因此，PLR 实现了宽度级计算扩展，同时保持实时推理的可行性，使其适用于对准确率和响应时间均有严格要求的工业部署场景。

6.5.4 注意力可视化分析。 为了深入了解多流推理机制，我们在图5中可视化了不同流和推理步骤的注意力分布。可视化结果表明，不同流呈现出各异的注意力模式，分别关注输入序列中互为补充的方面：一个流专注于近期交互，另一个流则探索长期依赖。注意力分配的这种多样性证实，我们的多流架构能够鼓励探索异质推理路径，从而更全面地覆盖用户兴趣，并助力PLR取得优异性能。

7 结论

在这项工作中，我们提出了并行潜在推理（PLR），这是一种开创性地将宽度级计算扩展应用于序列推荐的新型框架。不同于面临边际收益递减问题的现有仅深度推理方法，PLR通过可学习的触发词元、全局推理正则化和自适应推理流混合聚合，同时探索多条多样化的推理轨迹。在三个真实世界数据集上进行的大量实验表明，PLR的表现显著优于最先进基线方法，同时保持了实时推理效率。理论分析进一步验证了并行推理通过集成多样性降低预测误差以及缓解过度思考问题的有效性。我们的工作表明，宽度级扩展为深度扩展提供了互补维度，有望释放模型的推理潜力，从而为推理增强推荐开辟了新途径。

展望未来，我们计划进一步探索协同整合策略，同时优化并行与深度推理扩展，从而有望实现更复杂的多维推理架构。

参考文献

[1] Pranjal Aggarwal、Aman Madaan、Yiming Yang 等。2023。《让我们逐步采样：借助自适应一致性实现大语言模型的高效推理与编码》。arXiv 预印本 arXiv:2305.11860（2023）。[2] Janice Ahn、Rishu Verma、Renze Lou、Di Liu、Rui Zhang 和 Wenpeng Yin。2024。大语言模型的数学推理：进展与挑战。arXiv 预印本 arXiv:2402.00157（2024）。[3] Jianlv Chen、Shitao Xiao、Peitian Zhang、Kun Luo、Defu Lian 和 Zheng Liu。2024。BGE M3-Embedding：通过自知识蒸馏实现多语言、多功能、多粒度文本嵌入。arXiv 预印本 arXiv:2402.03216（2024）。[4] Lingjiao Chen、Jared Davis、Boris Hanin、Peter Bailis、Ion Stoica、Matei Zaharia 和 James Zou。2024。更多大语言模型调用就是你所需要的一切吗？探究复合人工智能系统的扩展特性。《神经信息处理系统进展》37（2024），45767–45790。[5] Qiguang Chen、Libo Qin、Jinhao Liu、Dengyun Peng、Jiannan Guan、Peng Wang、Mengkang Hu、Yuhang Zhou、Te Gao 和 Wanxiang Che。2025。迈向推理时代：长链式思维推理大语言模型综述。arXiv 预印本 arXiv:2503.09567（2025）。[6] Andy Clark。1989。《微观认知：哲学、认知科学与并行分布式处理》。第6卷。麻省理工学院出版社。[7] Qiang Cui、Shu Wu、Qiang Liu、Wen Zhong 和 Liang Wang。2018。MV-RNN：用于序列推荐的多视图循环神经网络。《IEEE知识与数据工程汇刊》32，2（2018），317–331。[8] Sunhao Dai、Jiakai Tang、Jiahua Wu、Kun Wang、Yuxuan Zhu、Bingjun Chen、Bangyang Hong、Yu Zhao、Cong Fu、Kangle Wu 等。2025。OnePiece：将上下文工程和推理引入工业级级联排序系统。arXiv 预印本 arXiv:2509.18091（2025）。[9] Yangruibo Ding、Jinjun Peng、Marcus Min、Gail Kaiser、Junfeng Yang 和 Baishakhi Ray。2024。SemCoder：通过全面的语义推理训练代码语言模型。《神经信息处理系统进展》37（2024），60275–60308。[10] Tim Donkers、Benedikt Loepp 和 Jürgen Ziegler。2017。基于用户的序列循环神经网络推荐。载于第十一届 ACM推荐系统大会论文集。152–160。[11] Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Ruoyu Zhang、Runxin Xu、Qihao Zhu、Shirong Ma、Peiyi Wang、Xiao Bi 等。2025。DeepSeek-R1：通过强化学习激励大语言模型的推理能力。arXiv 预印本 arXiv:2501.12948（2025）。[12] Lei Guo、Hongzhi Yin、Qinyong Wang、Tong Chen、Alexander Zhou 和 Nguyen Quoc Viet Hung。2019。流式会话推荐。载于第25届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。1569–1577。[13] Yupeng Hou、Zhankui He、Julian McAuley 和 Wayne Xin Zhao。2023。学习向量量化物品表示以构建可迁移序列推荐系统。载于 ACM Web会议2023论文集。1162–1171。[14] Yupeng Hou、Jiacheng Li、Zhankui He、An Yan、Xiusi Chen 和 Julian McAuley。2024。连接语言与物品以实现检索和推荐。arXiv 预印本 arXiv:2403.03952（2024）。[15] Yupeng Hou、Shanlei Mu、Wayne Xin Zhao、Yaliang Li、Bolin Ding 和 Ji-Rong Wen。2022。迈向推荐系统的通用序列表示学习。载于第28届 ACM SIGKDD知识发现与数据挖掘大会论文集。585–593。[16] Jie Huang 和 Kevin Chen-Chuan Chang。2023。迈向大语言模型推理：综述。载于计算语言学协会 ACL 2023 论文集。1049–1065。[17] Ray Jackendoff。2011。并行架构及其在认知科学中的地位。《多维句法与形态学》。编者：A. Nolda、O. Teuber。柏林、纽约：Mouton De Gruyter（2011），17–44。[18] Daniel Kahneman。2011。《思考，快与慢》。麦克米伦。[19] Wang-Cheng Kang 和 Julian McAuley。2018。自注意力序列推荐。载于2018年 IEEE 国际数据挖掘会议（ICDM）。IEEE，197–206。[20] Diederik P Kingma。2014。Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980（2014）。[21] Ioannis Konstas、Vassilios Stathopoulos 和 Joemon M Jose。2009。关于社交网络与协同推荐。载于第32届 ACM SIGIR信息检索研究与发展国际会议论文集。195–202。[22] Chengshu Li、Jacky Liang、Andy Zeng、Xinyun Chen、Karol Hausman、Dorsa Sadigh、Sergey Levine、Li Fei-Fei、Fei Xia 和 Brian Ichter。2023。代码链：借助语言模型增强型代码模拟器进行推理。arXiv 预印本 arXiv:2312.04474（2023）。

[23] Zihan Lin、Changxin Tian、Yupeng Hou 和 Wayne Xin Zhao。2022。通过邻域增强对比学习改进图协同过滤。载于 ACM Web会议2022论文集。2320–2329。[24] Enze Liu、Bowen Zheng、Xiaolei Wang、Wayne Xin Zhao、Jinpeng Wang、Sheng Chen 和 Ji-Rong Wen。2025。LARES：面向序列推荐的潜在推理。arXiv 预印本 arXiv:2505.16865（2025）。[25] Zhanyu Liu、Shiyao Wang、Xingmei Wang、Rongzhou Zhang、Jiaxin Deng、Honghui Bao、Jinghao Zhang、Wuchao Li、Pengfei Zheng、Xiangyu Wu 等。2025。Onerec-think：生成式推荐中的文本内推理。arXiv 预印本 arXiv:2510.11639（2025）。[26] Jiaqi Ma、Zhe Zhao、Xinyang Yi、Jilin Chen、Lichan Hong 和 Ed H Chi。2018。利用多门混合专家模型对多任务学习中的任务关系进行建模。载于第24届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。1930–1939。[27] Shervin Minaee、Tomas Mikolov、Narjes Nikzad、Meysam Chenaghlu、Richard Socher、Xavier Amatriain 和 Jianfeng Gao。2024。大语言模型：综述。arXiv 预印本 arXiv:2402.06196（2024）。[28] Humza Naveed、Asad Ullah Khan、Shi Qiu、Muhammad Saqib、Saeed Anwar、Muhammad Usman、Naveed Akhtar、Nick Barnes 和 Ajmal Mian。2025。大语言模型全面概述。《ACM 智能系统与技术汇刊》16，5（2025），1–72。[29] Keqin Peng、Liang Ding、Yuanxin Ouyang、Meng Fang 和 Dacheng Tao。2025。从自我怀疑的视角重新审视长链式思维中的过度思考。arXiv 预印本 arXiv:2505.23480（2025）。[30] Aske Plaat、Annie Wong、Suzan Verberne、Joost Broekens、Niki van Stein 和 Thomas Back。2024。大语言模型推理综述。arXiv 预印本 arXiv:2407.11511（2024）。[31] Zihan Qiu、Zekun Wang、Bo Zheng、Zeyu Huang、Kaiyue Wen、Songlin Yang、Rui Men、Le Yu、Fei Huang、Suozhi Huang 等。2025。大语言模型的门控注意力：非线性、稀疏度与无注意力汇聚。arXiv 预印本 arXiv:2505.06708（2025）。[32] Evgenia Rusak、Patrik Reizinger、Attila Juhos、Oliver Bringmann、Roland S Zimmermann 和 Wieland Brendel。2024。InfoNCE：识别理论与实践之间的差距。arXiv 预印本 arXiv:2407.00143（2024）。[33] Steven A Sloman。1996。支持双系统推理的经验依据。《心理学通报》119，1（1996），3。[34] Fei Sun、Jun Liu、Jian Wu、Changhua Pei、Xiao Lin、Wenwu Ou 和 Peng Jiang。2019。BERT4Rec：采用 Transformer 双向编码器表示的序列推荐。载于第28届 ACM 信息与知识管理国际会议论文集。1441–1450。[35] Jiakai Tang、Sunhao Dai、Teng Shi、Jun Xu、Xu Chen、Wen Chen、Jian Wu 和 Yuning Jiang。2025。推荐前先思考：释放序列推荐的潜在推理能力。arXiv 预印本 arXiv:2503.22675（2025）。[36] Jiakai Tang、Sunhao Dai、Zexu Sun、Xu Chen、Jun Xu、Wenhui Yu、Lantao Hu、Peng Jiang 和 Han Li。2024。通过决策边界感知图对比学习实现稳健推荐。载于第30届 ACM SIGKDD知识发现与数据挖掘大会论文集。2854–2865。[37] Jiakai Tang、Yujie Luo、Xunke Xi、Fei Sun、Xueyang Feng、Sunhao Dai、Chao Yi、Dian Chen、Zhujin Gao、Yang Li 等。2025。具有主动用户指令的交互式推荐智能体。arXiv 预印本 arXiv:2509.21317（2025）。[38] Jiakai Tang、Zuxu Sun、Wei Yao 和 Xu Chen。2024。通过噪声感知对抗扰动实现稳健图推荐。载于高级应用数据库系统国际会议。Springer，134–150。[39] Jiaxi Tang 和 Ke Wang。2018。通过卷积序列嵌入实现个性化 Top-N 序列推荐。载于第十一届 ACM Web搜索与数据挖掘国际会议论文集。565–573。[40] Kimi 团队、Angang Du、Bofei Gao、Bowei Xing、Changjiu Jiang、Cheng Chen、Cheng Li、Chenjun Xiao、Chenzhuang Du、Chonghua Liao 等。2025。Kimi k1.5：利用大语言模型扩展强化学习。arXiv 预印本 arXiv:2501.12599（2025）。[41] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017。Attention Is All You Need。《神经信息处理系统进展》30（2017）。[42] Chuan Wang、Gaoming Yang、Han Wu、Jiakai Tang、Jiahao Yu、Jian Wu、Jianwu Hu、Junjun Zheng、Shuwen Xiao、Yeqiu Yang 等。2025。ReaSeq：通过推理释放世界知识以实现序列建模。arXiv 预印本 arXiv:2512.21257（2025）。[43] Peng-Yuan Wang、Tian-Shuo Liu、Chenyang Wang、Ziniu Li、Yidi Wang、Shu Yan、Chengxing Jia、Xu-Hui Liu、Xinwei Chen、Jiacheng Xu 等。2025。大语言模型数学推理综述。《计算机调查》（2025）。[44] Weiqing Wang、Hongzhi Yin、Zi Huang、Qinyong Wang、Xingzhong Du 和 Quoc Viet Hung Nguyen。2018。基于流式排序的推荐系统。载于第41届 ACM SIGIR信息检索研究与发展大会。525–534。[45] Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc Le、Ed Chi、Sharan Narang、Aakanksha Chowdhery 和 Denny Zhou。2022。自洽性提升了语言模型的思维链推理能力。arXiv 预印本 arXiv:2203.11171（2022）。[46] Ziqi Wang、Boye Niu、Zhongli Li、Linghui Meng、Jing Liu、Zhi Zheng、Tong Xu、Hua Wu、Haifeng Wang 和 Enhong Chen。2025。A2R：一种非对称两阶段并行推理框架。arXiv预印本 arXiv:2509.22044（2025）。[47] Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Fei Xia、Ed Chi、Quoc V Le、Denny Zhou等。2022。思维链提示能够在大语言模型中 elicitation 推理。神经信息处理系统进展35（2022），24824–24837。[48] Wei Wei、Chao Huang、Lianghao Xia和Chuxu Zhang。2023。用于推荐的多模态自监督学习。载于2023年ACM Web会议论文集。790–800。[49] Sam Wiseman和Alexander M Rush。2016。序列到序列学习作为束搜索优化。arXiv预印本 arXiv:1606.02960（2016）。[50] Liwei Wu、Shuqing Li、Cho-Jui Hsieh和James Sharpnack。2020。SSE-PT：通过个性化Transformer进行序列推荐。载于第14届ACM推荐系统大会论文集。328–337。[51] Yuxi Xie、Kenji Kawaguchi、Yiran Zhao、James Xu Zhao、Min-Yen Kan、Junxian He和Michael Xie。2023。用于推理的自评估引导的束搜索。神经信息处理系统进展36（2023），41618–41650。[52] Lanqing Xue、Xiaopeng Li和Nevin L Zhang。2020。并非所有注意力都是必需的：用于序列数据的门控注意力网络。载于AAAI人工智能大会论文集，第34卷。6550–6557。[53] Anushka Yadav、Isha Nalawade、Srujana Pillarichety、Yashwanth Babu、Reshmi Ghosh、Samyadeep Basu、Wenlong Zhao、Ali Nasaeh、Sriram Balasubramanian和Soundararajan Srinivasan。2025。跳跃、略过与过度思考：诊断推理模型在多跳分析中失误的原因。arXiv预印本 arXiv:2508.04699（2025）。[54] An Yan、Shuo Cheng、Wang-Cheng Kang、Mengting Wan和Julian McAuley。2019。CosRec：用于序列推荐的二维卷积神经网络。载于第28届ACM信息与知识管理国际会议论文集。2173–2176。[55] An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv等。2025。Qwen3技术报告。arXiv预印本 arXiv:2505.09388（2025）。[56] Xiwang Yang、Harald Steck和Yong Liu。2012。在线社交网络中的基于圈子的推荐。载于第18届ACM SIGKDD知识发现与数据挖掘国际会议论文集。1267–1275。[57] Chao Yi、Dian Chen、Gaoyang Guo、Jiakai Tang、Jian Wu、Jing Yu、Mao Zhang、Wen Chen、Wenjun Yang、Yujie Luo等。2025。RecGPT-V2技术报告。arXiv预印本 arXiv:2512.14503（2025）。[58] Chao Yi、Dian Chen、Gaoyang Guo、Jiakai Tang、Jian Wu、Jing Yu、Mao Zhang、Sunhao Dai、Wen Chen、Wenjun Yang等。2025。RecGPT技术报告。arXiv预印本 arXiv:2507.22879（2025）。[59] Runyang You、Yongqi Li、Xinyu Lin、Xin Zhang、Wenjie Wang、Wenjie Li和Liqiang Nie。2025。R2ec：迈向具备推理能力的大型推荐模型。载于第三十九届神经信息处理系统年会。[60] Fuqiang Yu、Lizhen Cui、Wei Guo、Xudong Lu、Qingzhong Li和Hua Lu。2020。基于类别感知深度模型的稀疏签到数据连续兴趣点推荐。载于2020年Web会议论文集。1264–1274。[61] Zheng Yuan、Fajie Yuan、Yu Song、Youhua Li、Junchen Fu、Fei Yang、Yunzhu Pan和Yongxin Ni。2023。推荐系统下一步何去何从？重新审视基于ID与基于模态的推荐模型。载于第46届ACM SIGIR信息检索研究与发展大会论文集。2639–2649。[62] Gaowei Zhang、Yupeng Hou、Hongyu Lu、Yu Chen、Wayne Xin Zhao和Ji-Rong Wen。2024。大型序列推荐模型的缩放定律。载于第18届ACM推荐系统大会论文集。444–453。[63] Jiaqi Zhang、Junliang Yu、Zongwei Wang、Wei Yuan、Tong Chen、Quoc Viet Hung Nguyen、Bin Cui和Hongzhi Yin。2025。迈向推理感知的推荐系统：大语言模型时代的综述。Authorea预印本（2025）。[64] Junjie Zhang、Beichen Zhang、Wenqi Sun、Hongyu Lu、Wayne Xin Zhao、Yu Chen和Ji-Rong Wen。2025。序列推荐中的慢思考。arXiv预印本 arXiv:2504.09627（2025）。[65] Yang Zhang、Wenxin Xu、Xiaoyan Zhao、Wenjie Wang、Fuli Feng、Xiangnan He和Tat-Seng Chua。2025。用于基于大语言模型的推荐的强化潜在推理。arXiv预印本 arXiv:2505.19092（2025）。[66] Wayne Xin Zhao、Kun Zhou、Junyi Li、Tianyi Tang、Xiaolei Wang、Yupeng Hou、Yingqian Min、Beichen Zhang、Junjie Zhang、Zican Dong等。2023。大语言模型综述。arXiv预印本 arXiv:2303.18223 1，2（2023）。[67] Bowen Zheng、Xiaolei Wang、Enze Liu、Xi Wang、Lu Hongyu、Yu Chen、Wayne Xin Zhao和Ji-Rong Wen。2025。DeepRec：通过基于大语言模型的推荐深入探索物品空间。arXiv预印本 arXiv:2505.16810（2025）。[68] Guorui Zhou、Honghui Bao、Jiaming Huang、Jiaxin Deng、Jinghao Zhang、Junda She、Kuo Cai、Lejian Ren、Lu Ren、Qiang Luo等。2025。OpenOneRec技术报告。arXiv预印本 arXiv:2512.24762（2025）。[69] Pablo Zivic、Hernan Vazquez和Jorge Sánchez。2024。使用Transformer扩展序列推荐模型。载于第47届ACM SIGIR信息检索研究与发展大会论文集。1567–1577。

完整理论证明

A.1 定理4.1的证明（集成误差分解）

证明。对于任意序列 S_u，在 $p^*(v|S_u)$ 下的期望负对数似然为：

$$\mathbb{E}_{v\sim p^*}[-\log \bar{p}(v|\mathcal{S}_u)] = -\sum_v p^*(v|\mathcal{S}_u)\log\left(\frac{1}{M}\sum_{m=1}^{M}\hat{p}_m(v|\mathcal{S}_u)\right).$$

根据詹森不等式（对数函数的凹性）：

$$-\log\left(\frac{1}{M}\sum_{m=1}^{M}\hat{p}_m(v|\mathcal{S}_u)\right) \le -\frac{1}{M}\sum_{m=1}^{M}\log\hat{p}_m(v|\mathcal{S}_u),$$

当预测不同时，不等式严格成立。因此：

$$\begin{aligned}\mathbb{E}_{v\sim p^*}[-\log \bar{p}(v|\mathcal{S}_u)] &\le \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{v\sim p^*}[-\log\hat{p}_m(v|\mathcal{S}_u)]\\ &= \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_m(\mathcal{S}_u),\end{aligned}$$

其中，$L_m(S_u)$ 是流 m 在序列 S_u上的损失。对 $S_u \sim D$ 取期望，得到：

$$\mathcal{L}_{\text{ens}} \le \bar{\mathcal{L}}_{\text{ind}}.$$

差值 $I(S_u) = \bar{L}_{\text{ind}} - L_{\text{ens}}(S_u)$ 量化了多样化预测的益处。根据算术几何平均不等式，$I(S_u) \ge 0$，当且仅当所有 $\hat{p}_m(v|S_u)$ 都相同时，等号成立。□

A.2 命题4.2的证明（多样性-专门化联系）

证明。在线性评分下，$\hat{p}_m(v|S_u) = \text{softmax}(z_m, e_v)$。对于两个不同的流 m, m'：

$$\|\hat{p}_m(\cdot|\mathcal{S}_u) - \hat{p}_{m'}(\cdot|\mathcal{S}_u)\|_{\text{TV}} \ge \frac{\delta_e}{2R}\|\mathbf{z}_m - \mathbf{z}_{m'}\|,$$

其中，$\|\cdot\|_{\text{TV}}$ 是总变差距离，R是项目嵌入的界，δ_e 是最小嵌入间隔。根据 Pinsker不等式，总变差距离可界定 KL散度：

$$\text{KL}(\hat{p}_m\|\hat{p}_{m'}) \ge \frac{1}{2}\|\hat{p}_m - \hat{p}_{m'}\|_{\text{TV}}^2 \ge \frac{\delta_e^2}{8R^2}\|\mathbf{z}_m - \mathbf{z}_{m'}\|^2.$$

专门化收益（来自詹森不等式的间隙）以下界为平均成对KL散度：

$$\mathcal{I}(\mathcal{S}_u) \ge \frac{1}{M(M-1)}\sum_{m\ne m'}\text{KL}(\hat{p}_m\|\hat{p}_{m'}) \ge \frac{\delta_e^2}{8R^2}D(\mathcal{S}_u).$$

令 $c = \delta_e^2/(8R^2) > 0$，证明完成。□

A.3 定理4.4（多样性衰减）的证明

证明。根据利普希茨性质（假设 4.3）：

$$\|\mathbf{h}_{t,m} - \mathbf{h}_{t,m'}\| = \|f(\mathbf{h}_{t-1,m}, \mathbf{r}_t) - f(\mathbf{h}_{t-1,m'}, \mathbf{r}_t)\| \le L\|\mathbf{h}_{t-1,m} - \mathbf{h}_{t-1,m'}\|.$$

从 $t = 0$ 迭代到 $t = T$：

$$\|\mathbf{h}_{T,m} - \mathbf{h}_{T,m'}\| \le L^T\|\mathbf{h}_{0,m} - \mathbf{h}_{0,m'}\| = L^T\|\boldsymbol{\tau}_m - \boldsymbol{\tau}_{m'}\|.$$

两边平方：

$$\|\mathbf{h}_{T,m} - \mathbf{h}_{T,m'}\|^2 \le L^{2T}\|\boldsymbol{\tau}_m - \boldsymbol{\tau}_{m'}\|^2.$$

对所有对 (m, m') 求平均：

$$D^{(T)} = \frac{1}{M(M-1)}\sum_{m\ne m'}\|\mathbf{h}_{T,m} - \mathbf{h}_{T,m'}\|^2 \le L^{2T}D^{(0)}.$$

对于 $L < 1$，用 $\gamma = -$ 对 $L = e^{-\gamma}$ 取对数 $L > 0$，可得指数形式：

$$D^{(T)} \le e^{-2\gamma T}D^{(0)}.$$

□

A.4 定理4.6（门控收益）的证明

证明 . 对于任意 S_u，根据詹森不等式：

$$\begin{aligned}\mathbb{E}_{v\sim p^*}[-\log\tilde{p}(v|\mathcal{S}_u)] &= -\sum_v p^*(v|\mathcal{S}_u)\log\left(\sum_{m=1}^{M}w_m(\mathcal{S}_u)\hat{p}_m(v|\mathcal{S}_u)\right)\\ &\le -\sum_v p^*(v|\mathcal{S}_u)\sum_{m=1}^{M}w_m(\mathcal{S}_u)\log\hat{p}_m(v|\mathcal{S}_u)\\ &= \sum_{m=1}^{M}w_m(\mathcal{S}_u)\mathbb{E}_{v\sim p^*}[-\log\hat{p}_m(v|\mathcal{S}_u)].\end{aligned}$$

门控集成与均匀集成之间的差距为：

$$\mathcal{L}_{\text{uniform}}(\mathcal{S}_u) - \mathcal{L}_{\text{gated}}(\mathcal{S}_u) = \mathbb{E}_{v\sim p^*}\left[\log\tilde{p}(v|\mathcal{S}_u) - \log\bar{p}(v|\mathcal{S}_u)\right].$$

当各流呈现专业化（每个流的损失随 m 和 S_u而变化）时，最优加权会为损失较低的流分配更高的 $w_m(S_u)$。这种增益可由条件互信息量化：

$$I(Z;V|\mathcal{S}_u) = \sum_{m=1}^{M}w_m(\mathcal{S}_u)\mathbb{E}_{v\sim p^*}\left[\log\frac{\hat{p}_m(v|\mathcal{S}_u)}{\tilde{p}(v|\mathcal{S}_u)}\right],$$

该量度了对选定的流 Z的了解揭示了多少关于目标 V的信息。在专业化条件下，$I(Z;V|S_u) > 0$。

□