

DLCRec: A Novel Approach for Managing Diversity in LLM-Based Recommender Systems

Chongming Gao^{*1}, Jiaju Chen^{†1}, Shuai Yuan^{‡2}, Shuchang Liu^{§3}, Qingpeng Cai^{¶3}, and Peng Jiang^{||3}

¹University of Science and Technology of China

²Hong Kong University of Science and Technology

³Kuaishou Technology

摘要

随着大语言模型 (LLMs) 的迅速发展, 将大语言模型融入推荐系统显著提升了性能。然而, 这通常以牺牲推荐多样性为代价, 从而可能影响用户满意度。为了解决这一问题, 可控推荐作为一种有前景的解决方案应运而生, 允许用户指定其偏好, 并获得满足其多样性需求的推荐。尽管这一方法具有潜力, 但现有的可控推荐系统往往依赖于简化的机制, 如单一提示 (prompt), 来调节多样性——这种方法未能充分捕捉用户偏好的复杂性。针对这些局限性, 作者提出了 DLCRec, 一种新颖的框架, 旨在实现 LLM-based 推荐中对多样性的细粒度控制。与传统方法不同, DLCRec 采用细粒度任务分解策略, 将推荐过程分解为三个顺序子任务: 类别预测、类别填充和物品预测。这些子任务独立训练, 并根据用户定义的控制数字按顺序推断, 从而确保对多样性进行更精确的控制。此外, 用户行为数据中关于多样性的稀缺性和不均匀分布为微调带来了重大挑战。为了解决这些问题, 作者引入了两种数据增强技术, 以增强模型对噪声和分布外数据的鲁棒性。这些技术使模型能够接触到更广泛的模式, 提高了其在生成不同多样性水平的推荐时的适应性。他们的广泛实证评估表明, DLCRec 不仅能够提供精确的多样性控制, 还在多个推荐场景中超越了现有的最先进基准模型。

关键词: 推荐; 大语言模型; 多样性

*Corresponding Author: chongming.gao@gmail.com

†cjj01@mail.ustc.edu.cn

‡syuanaf@connect.ust.hk

§liushuchang@kuaishou.com

¶cqpcurry@gmail.com

||jiangpeng@kuaishou.com

1 引言

随着大语言模型 (LLMs) 的出现, 推荐系统的能力得到了显著增强, 能够通过语义理解提供少样本推荐 [2, 4]。然而, 基于 LLM 的推荐系统存在一个主要的限制, 即它们倾向于生成与用户历史交互高度相似的同质化推荐结果 [3, 18], 因此, 研究领域已经从仅关注准确性, 转向考虑更广泛的推荐质量指标, 包括多样性 [18, 19]。此外, 越来越多的研究认识到, 赋予用户控制推荐结果的能力对于促进更具包容性和互动性的用户体验至关重要。赋予用户对推荐多样性的控制, 使得推荐系统能够更好地与个人偏好对接, 促进兴趣发现 [5], 并减少算法单一化带来的风险 [7]。

鉴于 LLM 中嵌入了广泛的语义信息, [13] 强调了这些模型在理解并可能管理根据用户需求生成多样化推荐的内在能力。然而, 直接将这些模型应用于具有不同多样性需求的可控推荐任务时, 面临着显著的挑战, 主要是因为缺乏专门针对这些任务进行明确训练和调整。尽管现有的基于 LLM 的可控推荐系统通常依赖单一提示来调节多样性 [8], 我们的研究发现, 这种方法可能过于简化。此外, 传统的微调方法 [18] 和思维链 (CoT) 提示技术 [22] 可能无法提供有效管理推荐多样性的精细控制。这一挑战进一步由于用户行为数据中与多样性相关的需求的稀缺性和不均匀分布而加剧 [18, 19]。因此, LLMs 很难从有限的示例中建立对多样性控制的深入理解。

为了应对这些挑战, 我们提出了一种新方法, 称为多样性导向的 LLM 可控推荐系统 (DLCRec), 该方法能够精确控制 LLM 推荐中的多样性。DLCRec 生成具有不同多样性水平的推荐——多样性定义为推荐列表中多类项的覆盖——并优化推荐系统以实现这一目标。我们的方法建立在细粒度任务分解策略上, 该策略引导 LLM 充分利用其能力, 生成多样化的推荐。具体而言, 我们将复杂的推荐任务分解为三个连续的子任务: (1) 预测未来推荐的类别; (2) 填充类别; (3) 预测未来物品。我们利用少样本示例为每个子任务构建特定的数据, 并进行微调训练。在训练阶段, 我们分别训练这三个子任务; 而在推理阶段, DLCRec 接收关于推荐多样性的具体用户控制数量, 并顺序执行三个子任务, 以提供满足用户需求的推荐。为了促进后两个任务之间的顺利衔接, 我们使用简洁且固定的输出格式来限制 LLM 的输出。

在推理阶段, 控制目标可能是未见过的, 或者落在训练数据的分布之外, 这可能会影响模型对新控制推荐结果的泛化能力。为了解决这个问题, 并减轻多样性信号在数据中稀缺和偏斜的问题, 我们提出了两种数据增强方法, 专门针对类别填充和物品预测任务。首先, 我们向训练样本中注入噪声, 以模拟前置任务中的错误, 从而提高模型对可能被前期任务错误预测的控制目标的鲁棒性。其次, 我们修改控制目标的分布, 包括那些在训练中很少遇到的或极端的情况, 增强训练数据, 使 DLCRec 能够更好地应对训练过程中罕见的场景。通过这些增强的数据集与原始训练数据结合, 我们显著提升了模型在生成基于类别控制的物品时的鲁棒性和适应能力。

我们通过在两个真实世界数据集上的广泛实验, 全面评估了我们提出方法的有效性, 并与一系列竞争基准方法进行了比较。实验结果表明, 我们的方法能够精确控制多样性, 仅在准确性上做出微小牺牲, 展示了其在多样性至关重要的实际应用中具有广阔的潜力。

我们的工作主要贡献总结如下:

我们提出了一种面向多样性的可控推荐的新范式, 使得对推荐物品的多样性实现细粒度控制。我们引入了 DLCRec, 一种新的多样性可控推荐框架, 通过将复杂的可控推荐任务分

解为可管理的子任务，并利用数据增强技术提高模型的鲁棒性和对多样性控制数量的适应能力。我们对 DLCRec 进行了全面的实证评估，展示了其在提供精确的多样性控制方面的有效性，并在多种推荐场景下超越了现有最先进的基准方法。

2 相关工作

在本节中，我们简要概述了基于大语言模型（LLM）的推荐系统，并探讨了现有的利用 LLM 进行可控推荐的研究。

2.1 用于推荐的 LLM

强大的大语言模型（LLM），如 GPT-4 [1] 和 Llama-3 [6] 的出现，引发了将 LLM 集成到推荐系统中的日益关注 [15, 23, 27]。一种常见的方法是对 LLM 进行特定推荐任务的微调 [2, 16, 21, 24]。借助其卓越的语义理解能力，基于 LLM 的推荐系统在冷启动或少样本学习场景中，特别是在推荐准确性上，已表现出相较于传统基于 ID 的模型的优势 [2, 13]。然而，这些微调方法通常依赖于相对容易学习的显式信号，而当面临更复杂的信号（如多样性控制）时，其局限性便显现出来。与此不同，我们的研究提出了一种任务分解方法，将复杂的多样性控制推荐问题拆解为多个易于管理的子任务，更加适合微调框架。其他研究则集中在通过协同过滤信息 [14, 26]（如来自传统模型的物品嵌入）来增强基于 LLM 的推荐系统的性能，特别是在数据密集型领域。虽然这些方法显示出一定的潜力，但我们的工作则采用不同的方向，探索 LLM 本身在控制推荐多样性方面的固有能力。通过这种方式，我们旨在充分挖掘 LLM 在推荐系统中的潜力，并提供更细致的理解。

2.2 用于可控推荐的 LLM

目前，基于 LLM 的可控推荐方法大致可以分为两类：指令微调（instruction tuning）[8, 9, 21, 25] 和强化学习微调（RLFT）[18–20]。指令微调通过对 LLM 进行多任务微调，使其能够泛化到不同的用户需求。例如，P5 [9] 和 InstructRec [25] 采用多种推荐任务，以满足不同的用户需求。[18] 提出了基于提示的控制方法，用来调节单一类别的比例。然而，在确保多样性的同时保持高准确性是一个更为复杂的挑战，单一任务的指令往往难以有效解决这一问题，因为它通常需要平衡多个类别。而强化学习微调（RLFT）方法 [18, 19] 则通过精心设计奖励信号，在监督学习后进一步优化 LLM 的指令执行能力。然而，RLFT 方法通常需要大量数据，并且面临收敛性问题。与此不同，我们的方法通过数据增强来提升模型对稀有和噪声控制的鲁棒性，为数据稀疏的场景提供了一种高效且有效的解决方案。

3 本文方法

3.1 DLCRec 框架

为了充分发挥大语言模型（LLMs）在推荐系统中控制多样性的潜力，论文作者提出了一个三阶段框架，称为 DLCRec。该框架将列表推荐任务分解为三个独立的子任务：类别预测（Genre Prediction, GP）、类别填充（Genre Filling, GF）和物品预测（Item Prediction, IP）。

如图 1所示，DLCRec 包括两个主要组成部分，训练框架和控制框架。

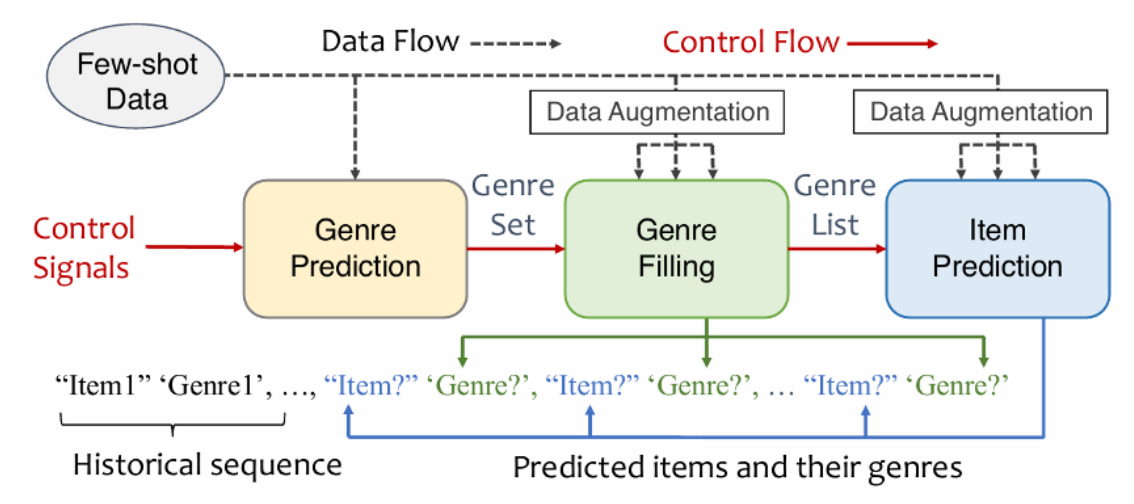


图 1. DLCRec 框架概述。DLCRec 将推荐任务分解为三个子任务：类别预测（Genre Predicting）、类别填充（Genre Filling）和物品预测（Item Predicting）。该框架由两个互补的组成部分构成：训练框架（黑线）和控制框架（红线）。在训练框架中，作者利用少样本数据和数据增强独立训练每个子任务。而控制框架通过三个子任务之间传递目标类别覆盖信号（目标数量），实现对最终推荐列表多样性的显式控制。该架构通过调整输入控制数量，能够有效操控推荐列表的多样性。

3.1.1 训练框架

对于三个子任务中的每一个，将 few-shot 数据调整为专门的训练数据集。每个阶段的具体任务格式详见表 1。每个数据集样本包含以下三个关键组成部分：

- **Instruction:** 定义任务的目标，提供上下文和指导，帮助 LLM 理解预期结果。
- **Input:** 代表用户的交互行为，例如历史偏好和兴趣。
- **Output:** 作为标签或目标响应，LLM 在接收到任务说明和输入后应该生成的内容。

为了训练 LLM，将任务说明和输入组合成一个提示（prompt），然后将其输入模型以生成输出。以下是对每个子任务的描述：

- **类别预测（GP）：**通过微调大语言模型，根据历史交互轨迹（即物品及其对应的内容类别）预测可能出现在推荐列表中的未来内容类别。输出是一组可能的内容类别。
- **类别填充（GF）：**通过指导大语言模型使用预测的内容类别填充未来内容类别占位符，同时结合历史交互轨迹进行操作。
- **物品预测（IP）：**通过指导大语言模型根据历史交互轨迹和预测的未来内容类别来预测未来的物品（即填充物品占位符）。

(a) Genre predicting (GP)	(b) Genre filling (GF)	(c) Item predicting (IP)
Instruction: Given a list of movies and their corresponding genres the user has watched before, please provide the 3 most likely genres in the future recommendation list. Output the genres only, without movie names, explanations, or numbers. The output format is "Genre1, Genre2, ..., Genre3".	Instruction: Below is a user's interaction trail of movies he likes. Each movie is in double quotes "", followed by its genre in single quotes '. Your task is to fill in the genre placeholders represented by "?" with the following genres: [Action, Comedy, Drama]. The "_" represents placeholder tokens that you should not consider. The output should maintain the same format as the input.	Instruction: Below is a user's interaction trail of movies he likes. Each movie is in double quotes "", followed by its genre in single quotes '. Your task is to fill in the movie placeholders represented by "?". The "_" represents placeholder tokens that you should not consider. The output should maintain the same format as the input.
Input: The user has watched the following movies with their corresponding genres in "()" before: "Star Wars: Episode V - The Empire Strikes Back (1980)" (Action), "Mission: Impossible (1996)" (Action), ..., "Stargate (1994)" (Action).	Input: "Star Wars: Episode V - The Empire Strikes Back (1980)" 'Action', "Mission: Impossible (1996)" 'Action', ..., "Stargate (1994)" 'Action', "_" '?', "_" '?', ..., "_" '?'	Input: "Star Wars: Episode V - The Empire Strikes Back (1980)" 'Action', "Mission: Impossible (1996)" 'Action', ..., "Stargate (1994)" 'Action', "?" 'Action', "?" 'Comedy', ..., "?" 'Comedy'
Output: Action, Comedy, Drama	Output: "Star Wars: Episode V - The Empire Strikes Back (1980)" 'Action', "Mission: Impossible (1996)" 'Action', ..., "Stargate (1994)" 'Action', "_" 'Action', "_" 'Comedy', ..., "_" 'Comedy'	Output: "Star Wars: Episode V - The Empire Strikes Back (1980)" 'Action', "Mission: Impossible (1996)" 'Action', ..., "Stargate (1994)" 'Action', "Saving Private Ryan (1998)" 'Action', "Pretty Woman (1990)" 'Comedy', ..., "Ghost (1990)" 'Comedy'

表 1. DLCRec 框架中的三个子任务示意图：Genre Predicting (GP)、Genre Filling (GF) 和 Item Predicting (IP)。(a) 任务 GP：基于历史交互轨迹预测未来的内容类型 (Genre)。在控制框架中，我们通过调整目标类型的数量来实现控制。(b) 任务 GF：根据目标内容类型和历史交互轨迹填充未来的内容类型。在控制框架中，通过修改目标内容类型来进行调整。(c) 任务 IP：基于预测的未来内容类型和历史交互轨迹预测未来的推荐项目。在控制框架中，通过调整预测的未来内容类型来控制推荐结果。

3.1.2 控制框架

在训练完这三个子任务后，它们可以通过利用各自的专业能力来整合，从而解决面向多样性的推荐任务。DLCRec 的主要目标是生成符合不同多样性需求的推荐列表，用户可以指定列表应该包含多少或哪些特定的内容类别。为了实现这一目标，控制输入信号以针对推荐中的特定内容类型覆盖至关重要。每个子任务在控制框架内调整其提示的方法：

- **类别预测 (GP)**：任务 GP 的输出通过限制指令提示中目标内容类别的数量来进行控制。通过修改这个提示并使用控制数来指定目标内容类别的数量，可以有效地引导 GP 的输出。
- **类别填充 (GF)**：任务 GF 的输出通过限制指令提示中的目标内容类别来进行管理，这决定了用于填充未来物品的内容类别。通过控制这些目标内容类别，特别是利用来自任务 GP 的输出，可以根据控制数来塑造 GF 的输出。
- **物品预测 (IP)**：任务 IP 的输出通过修改输入提示中的未来内容类别来进行控制。通过控制这些未来内容类别，可以影响推荐列表的内容类型组成。具体而言，来自任务 GF 的输出作为未来内容类别，使得控制数能够有效地引导 IP 的输出。

3.2 数据增强

为了解决训练数据中多样性信号的稀缺的问题，作者提出了两种新的数据增强方法，分别用于 GF 和 IP 任务，增强模型对噪声和分布外控制目标的鲁棒性。通过将原始数据与增强

数据结合，创建一个更加多样化和具有代表性的训练集，使模型能够学习到更具普适性的模式，从而提高在生成具有多样性水平的推荐时的性能。

3.2.1 GF 中的数据增强

如图 2 所示，作者在 GF 采用两种数据增强方法 GF-N 和 GF-D 来增强模型对噪声和控制目标类别数量的鲁棒性。

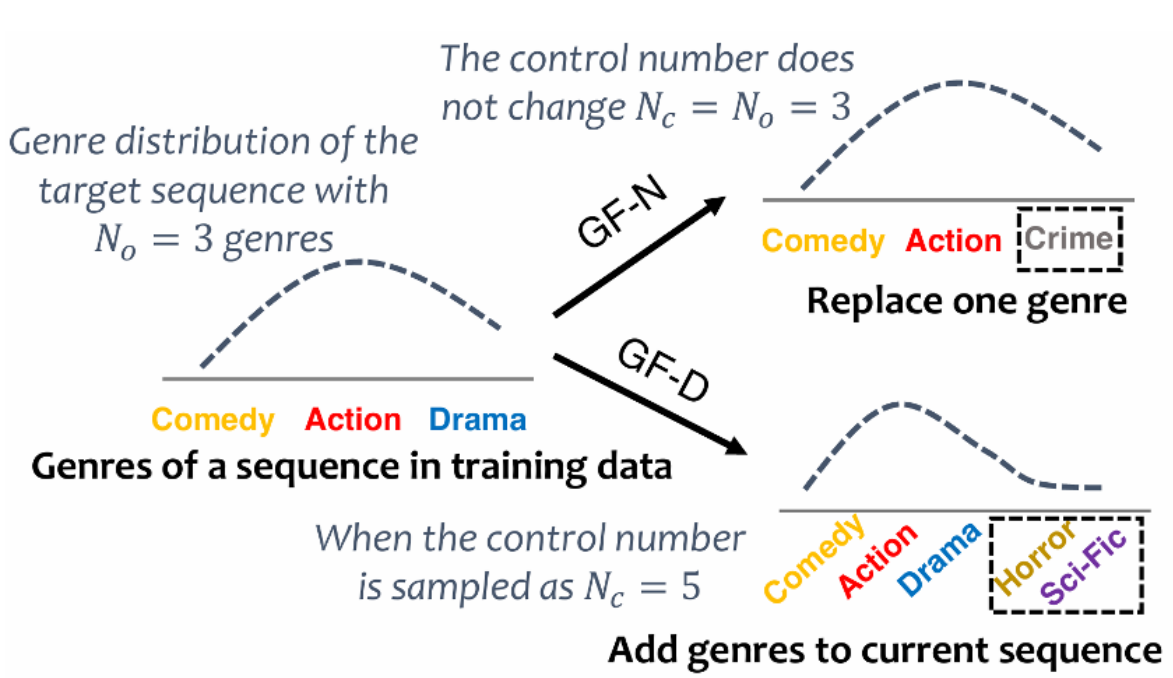


图 2. 任务 GF 中使用的数据增强方法说明。采用了两种策略：(a) “GF-N”，通过用噪声类别替换原始类别来引入噪声；(b) “GF-D”，通过随机增加或删除类别，直到达到抽样控制阈值，来调整类别分布，从而促进均匀的目标分布。

- **GF-N:** 为了增强模型对噪声目标类别的鲁棒性，通过手动引入噪声来修改训练序列，将每个序列中的一个类别替换为一个噪声类别。这样可以确保控制的类别数 N_C 保持与原始类别数 N_0 相等。
- **GF-D:** 为了进一步扰动序列中的类别分布，从 (1, 10) 范围内采样控制数量 N_C 。如果原始类别数 N_0 大于控制数量 N_C ，则随机替换最不常见的类别。相反，如果 N_0 小于 N_C ，则根据训练集中类别的分布，额外采样类别以达到 N_C 。

3.2.2 IP 中的数据增强

根据上述 GF-N 和 GF-D 任务，进一步把这两种数据增强的方法应用到训练数据中的原始物品序列中。

- **IP-N:** 为了考虑任务 GF 预测中可能出现的错误，这些错误可能来源于任务 GP 和 GF 的不准确性，在 IP 任务的输入中引入噪声。具体来说，定义一个误差率 r ，表示 GF 任务中预测错误的概率。然后，根据训练集中物品的分布，将推荐列表中的 r 比例的物品替换为其他类别的物品。

- **IP-D:** 继 GF-D 策略之后, 进一步将每个类别中的物品替换为相应的物品, 方法是根据训练数据中每个类别的物品分布进行采样。这样可以确保模型接触到具有不同类别覆盖范围的各种物品, 从而增强其在生成跨类别物品时的能力。

4 复现细节

4.1 与已有开源代码对比

本复现实验在基于原文作者开源的代码的基础上进行, 源代码网址: <https://github.com/Jiaju-Chen/DLCRec>

在此基础上, 对于源代码中及原文中没有给出的细节, 我做了一些自己的尝试。

1. 在研究问题 RQ2 中, 作者探究了将任务 GP 和 GF 分开或结合起来的性能分别是怎样的, 在原文和源代码中都没有给出细节或注释, 我做的一个尝试是: 在训练了三个子任务后得到的 GP, GF 和 IP 模型, 对于 Seperate 部分直接用到 data 文件夹里两个不同数据集的 evaluate_GF 方法来直接评估得到的 GF 模型推理的结果。而对于 Combined 部分, 先让 GP 模型经过控制框架控制后 (控制数量为 5), 再让 GF 模型来经过控制框架得到 GF 模型推理的结果, 然后评估这个得到的结果。

2. 在研究问题 RQ3 中, 作者探究了用不同数据增强策略的 DLCRec 的性能, 这部分也在原文和源代码中都没有给出细节或注释, 所以我做的一个尝试是: 对于分别经过两种数据增强策略后的两个 GF-N (IP-N) 及 GF-D (IP-D) 模型, 因它们是微调模型, 即 Lora Adapter, 我用 python 的 peft 库中的 add_weighted_adapter 方法将两个模型融合在一起, 两者的权重都设置为 0.5, 重要的参数 combination_type 设置为 linear, 经过融合后的模型作为原文中提及的 GF-ND (IP-ND) 模型来进行实验。

4.2 实验环境搭建

实验在一台 RTX4090 服务器上进行, 服务器上有 4 张 RTX4090 24G 的计算卡。程序环境为 python 环境, 用到的 python 库主要为 PyTorch、Transformer、Peft 等库。

5 实验结果分析

在原论文中, 作者提出了三个研究问题:

- **RQ1:** 他们的方法与其他 baseline 方法相比, 对控制目标类别数量的服从度有多高?
- **RQ2:** 他们提出的任务分解框架在实验中的性能如何?
- **RQ3:** 他们提出的数据增强方法对整体的框架的性能提升多少?

5.1 实验设置

5.1.1 数据集

实验采用的数据集为:

- **MovieLens10M 数据集 [10]**: 这是一个常用于电影推荐的数据集。
- **Steam 数据集 [12]**: 这个数据集包含了 Steam 用户和 Steam 商店里的游戏的交互行为。

对于这两个数据集，作者他们集中注意在积极的交互行为，对 MovieLens10M 数据集来说，积极的交互行为定义为高于 3 的评价（满分为 5）；对 Steam 数据集来说，积极的交互行为定义为超过 3 小时的游戏时长。为了便于模型训练，将序列长度设为 20，前 10 个交互行为作为历史序列，后 10 个交互行为作为未来序列。对于交互少于 20 的序列予以丢弃，对剩余序列按时间顺序排序，并按 8:1:1 的比例划分为训练集、验证集和测试集。随后，从每个数据集中随机抽取 1,000 条序列，组成最终的训练集、验证集和测试集。

5.1.2 评价指标

为了全面评估 DLCRec 和其他 baseline 模型在生成具有不同多样性水平的推荐列表方面的性能，作者设计了一个多方面的评估框架，其中包含三个不同的控制设置，每个设置对应不同的控制数量：低（2 个类别）、中（5 个类别）和高（8 个类别）。这些控制数量既能代表数据集中稀有的控制需求，也能涵盖常见的控制需求，从而评估模型适应不同用户偏好的能力。实验采用两个准确性指标 NDCG@K [11] 和 Recall@K 来评估推荐项目的相关性。此外，实验还使用了两个控制指标 Cov@K 和 MAE_Cov@K 来评估模型满足多样性需求的能力，其中 Cov@K 用于测量推荐列表中类别的覆盖度，而 MAE_Cov@K 用于计算实际类别覆盖度与目标类别覆盖度之间的平均绝对误差。目的是在保持较高准确性的同时，实现较低的 MAE_Cov@K，表明模型能够在相关性和多样性之间实现有效平衡。

5.1.3 实验细节

实验用 Llama3 [6] 作为 backbone 大语言模型。对所有基于 LLM 的方法，采用一个统一的训练流程：每个模型在 batch size 为 250 的条件下被训练最多 25 个 epoch，学习率在 [0.0001, 0.0005, 0.001] 之间调整以优化模型的性能，实验中用 AdamW 优化器 [17] 来进行模型训练，在推理阶段，采用贪心搜索策略来生成输出序列。

5.2 总体性能 (RQ1)

5.2.1 Baselines

- **BIGRec_div [2]**: 在原论文中作者用了 BIGRec 来适应多样性推荐的任务，用到的是通过含有不同控制数量的 prompt 来微调模型。BIGRec 原始是被设计成用来推荐单个物品的序列推荐模型 (sequential recommender)，作者通过修改任务的目标来让模型可以生成一个推荐的列表。
- **BIGRec_CoT [22]**: 作者用了思维链的 prompt 策略在 BIGRec 上以促进多样性推荐任务。主要是用特定的 prompt 让模型生成与控制数量对应的类别，然后让模型根据这些类别来进行推荐。

方法	Movie				Steam			
	NDCG@10	HR@10	Cov@10	MAE_Cov@10	NDCG@10	HR@10	Cov@10	MAE_Cov@10
控制数量 =2								
BIGRec_div	0.0490	0.0428	2.526	1.112	0.0211	0.0173	2.689	0.827
BIGRec_CoT	0.0466	0.0396	2.517	0.987	0.0268	0.0218	2.242	0.486
DLCRec	0.0357	0.0306	2.478	0.798	0.0236	0.0201	1.655	0.445
控制数量 =5								
BIGRec_div	0.0481	0.0426	2.515	2.529	0.0218	0.0182	2.713	2.291
BIGRec_CoT	0.0469	0.040	2.519	2.541	0.0277	0.0225	2.257	2.745
DLCRec	0.0450	0.0405	4.468	0.662	0.0289	0.0272	4.582	0.434
控制数量 =8								
BIGRec_div	0.0480	0.0426	2.508	5.492	0.0216	0.0179	2.743	5.257
BIGRec_CoT	0.0470	0.0403	2.525	5.475	0.0274	0.0223	2.253	5.747
DLCRec	0.0436	0.0383	7.495	0.511	0.0273	0.0248	7.873	0.131

表 2. DLCRec 和其他 baseline 方法在 Movie 和 Steam 数据集上的性能表现对比。在控制指标中的最好结果用粗体表示。(原文)

5.2.2 复现结果

在原文中，如表 2 所示，DLCRec 在所有情况下都能取得最小的 MAE_Cov@10，表明它能最成功地在推荐列表中展现出符合控制数量的多样性推荐，DLCRec 的推荐结果与控制数字紧密契合，展示了其在满足多样性要求方面的有效性。相比之下，尽管 BIGRec_div 和 BIGRec_CoT 通过指令调优修改提示，指示 LLM 根据控制数量进行推荐，但它们在学习这一复杂任务时遇到了困难。DLCRec 将微调过程分解为三个子任务并顺序执行，使其在这一面向多样性的可控推荐任务中表现卓越。本次复现的结果如表 3 所示，从结果中可以看到复现结果与原文结果相差不大，表明复现这一部分是比较成功的。

5.3 消融实验 (RQ2 & RQ3)

5.3.1 任务分解框架

原文中作者进行了实验以验证分解框架的必要性。正如表 2 所示，使用单一任务对 LLM 进行微调无法使其适应不同的多样性需求。此外，实验结果表明，将任务 GP 和 GF 结合起来并不是可行的解决方案。如图 3 所示，将任务 GP 和 GF 结合会导致后面类别填充的性能显著下降，从覆盖控制指标 COV_MAE@10 和类别准确性指标 RECALL@10 可以看出这一点。图 4 是复现的结果，由于原文中没有给出该部分的实验细节，复现的尝试是直接评估 GF 任务 (Separate 部分) 和评估经过控制数量为 5 的 GP 任务后的 GF 任务 (Combined 部分)，得到的 COV_MAE@10 结果与原文相差较大，表明复现细节与原文的实现细节相差较大。

方法	Movie				Steam			
	NDCG@10	HR@10	Cov@10	MAE_Cov@10	NDCG@10	HR@10	Cov@10	MAE_Cov@10
控制数量 = 2								
BIGRec_div	0.0478	0.0404	2.481	1.115	0.0199	0.0196	2.837	0.919
BIGRec_CoT	0.0468	0.0401	2.485	1.029	0.0239	0.0212	2.580	0.698
DLCRec	0.0392	0.0338	2.518	0.786	0.0277	0.0236	1.522	0.664
控制数量 = 5								
BIGRec_div	0.0468	0.0395	2.489	2.569	0.0234	0.0210	2.819	2.185
BIGRec_CoT	0.0465	0.0397	2.478	2.560	0.0249	0.0223	2.570	2.430
DLCRec	0.0444	0.0394	4.293	0.779	0.0317	0.0288	4.529	0.513
控制数量 = 8								
BIGRec_div	0.0466	0.0395	2.480	5.520	0.0232	0.0208	2.828	5.172
BIGRec_CoT	0.0466	0.0398	2.470	5.530	0.0248	0.0222	2.583	5.417
DLCRec	0.0450	0.0400	5.396	2.604	0.0248	0.0217	6.548	1.452

表 3. DLCRec 和其他 baseline 方法在 Movie 和 Steam 数据集上的性能表现对比。在控制指标中的最好结果用粗体表示。(复现)

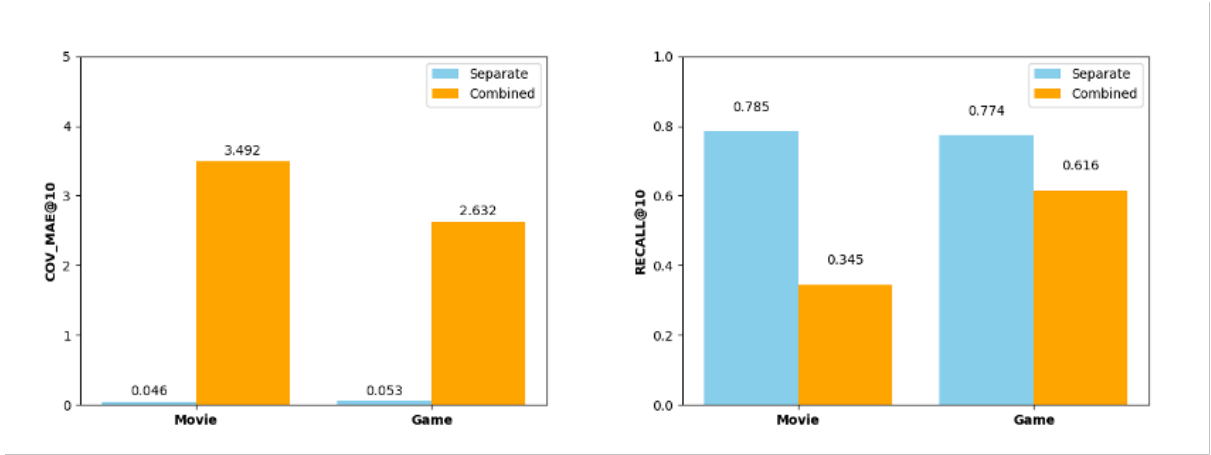


图 3. 将任务 GP 和 GF 分开或结合起来的性能 (原文)

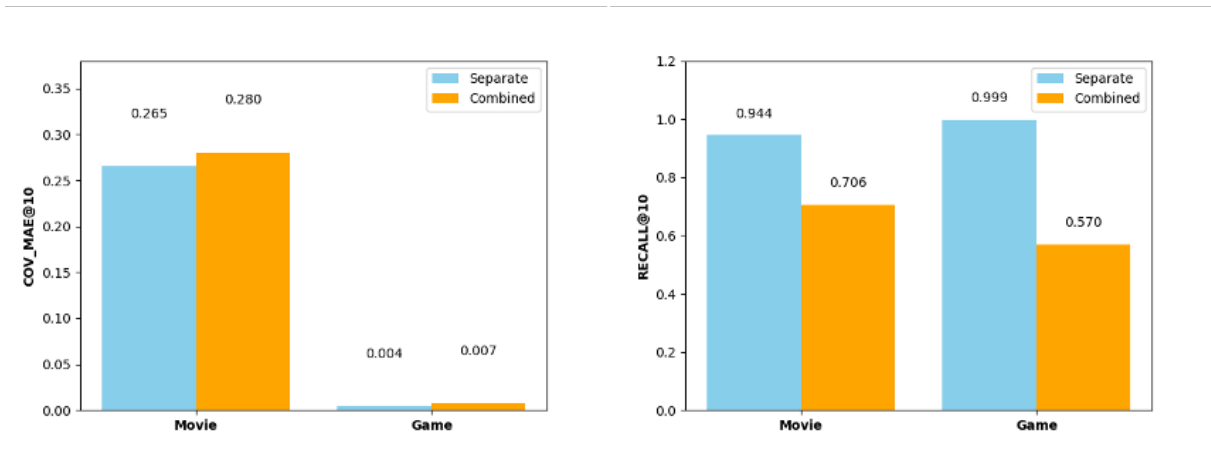


图 4. 将任务 GP 和 GF 分开或结合起来的性能 (复现)

IP DA	GF DA	Movie				Steam			
		NDCG@10	RECALL@10	Cov@10	COV_MAE@10	NDCG@10	RECALL@10	Cov@10	COV_MAE@10
IP	GF	0.0390	0.0354	3.925	1.133	0.0274	0.0256	4.300	0.704
IP-ND	GF	0.0430	0.0396	4.426	0.710	0.0265	0.0249	4.529	0.485
IP-ND	GF-N	0.0437	0.0399	4.509	0.603	0.0280	0.0262	4.524	0.494
IP-ND	GF-D	0.0425	0.0381	4.487	0.611	0.0260	0.0242	4.822	0.192
IP-ND	GF-ND	0.0450	0.0405	4.468	0.662	0.0289	0.0272	4.582	0.434
IP	GF-ND	0.0449	0.0416	3.927	1.109	0.0281	0.0260	4.338	0.666
IP-N	GF-ND	0.0431	0.0390	4.296	0.740	0.0240	0.0214	4.514	0.492
IP-D	GF-ND	0.0454	0.0395	3.845	1.251	0.0282	0.0254	4.799	0.219
IP-ND	GF-ND	0.0450	0.0405	4.468	0.662	0.0289	0.0272	4.582	0.434

表 4. 用不同数据增强策略的 DLCRec 的性能表现对比。“DA”表示“Data Augmentation”,“GF-ND”和“IP-ND”表示同时用了两种数据增强策略的模型,“GF”和“IP”表示没有用数据增强策略的模型。每组的最好结果用粗体表示。(原文)

IP DA	GF DA	Movie				Steam			
		NDCG@10	RECALL@10	Cov@10	COV_MAE@10	NDCG@10	RECALL@10	Cov@10	COV_MAE@10
IP	GF	0.0444	0.0394	4.293	0.779	0.0317	0.0288	4.529	0.513
IP-ND	GF	0.0311	0.0274	3.457	1.627	0.0211	0.0194	3.838	1.232
IP-ND	GF-N	0.0291	0.0256	3.148	1.912	0.0204	0.0186	3.649	1.389
IP-ND	GF-D	0.0316	0.0277	3.717	1.333	0.0222	0.0197	4.237	0.875
IP-ND	GF-ND	0.0320	0.0280	3.545	1.519	0.0195	0.0172	3.797	1.263
IP	GF-ND	0.0451	0.0403	4.319	0.741	0.0301	0.0269	4.399	0.653
IP-N	GF-ND	0.0396	0.0354	4.144	0.880	0.0254	0.0246	4.204	0.798
IP-D	GF-ND	0.0421	0.0368	3.595	1.473	0.0246	0.0224	4.039	1.011
IP-ND	GF-ND	0.0320	0.0280	3.545	1.519	0.0195	0.0172	3.797	1.263

表 5. 用不同数据增强策略的 DLCRec 的性能表现对比。“DA”表示“Data Augmentation”,“GF-ND”和“IP-ND”表示同时用了两种数据增强策略的模型,“GF”和“IP”表示没有用数据增强策略的模型。每组的最好结果用粗体表示。(复现)

5.3.2 数据增强

为了研究数据增强策略对 DLCRec 在任务 GF 和 GP 中性能的影响,作者通过不同的数据增强策略组合进行了消融实验。实验结果如表 4 所示,展示了 DLCRec 在不同数据增强策略下的表现。为了隔离每个策略的效果,先保持一个任务稳定,同时分析另一个任务的性能趋势。表 4 展示的是控制数量为 5 时的结果,这些发现突出了数据增强策略在提升 DLCRec 性能中的重要性,并展示了精心选择最有效策略组合以实现最佳结果的价值。表 5 是复现的结果,由于原文中没有给出“GF-ND”和“IP-ND”的实现细节,复现的尝试是将用“N”策略和“D”策略训练出来的两个各自的 lora adapter 融合在一起,用到的是 python 的 peft 库中的 add_weighted_adapters 方法。从结果上看,复现的结果与原文的结果相差也比较大,说明在同时使用两种数据增强策略的复现实现方法与原文实现方法还是有些不一致的。

5.4 敏感度分析

为了进一步评估 DLCRec 的控制能力,作者进行了实验,测试其生成具有不同多样性水平的推荐列表的能力。具体来说,将控制数量设置为 1 到 10,分别对应最低和最高多样性。这使我们能够评估 DLCRec 是否能够生成准确反映用户所需多样性水平的推荐列表。如图 5 所

示，实验结果表明，DLCRec 的推荐列表与控制数量之间存在强相关性。当控制数量从 1 增加到 10 时，推荐物品中的类别覆盖度也以几乎线性的方式增加，斜率略小于 1。与完美线性关系的轻微偏差可以归因于大语言模型的类别映射能力的局限性。图 6 是复现的结果，这个实验是基于上述数据增强的方法（结合两种“N”策略和“D”策略），然后根据不同的控制数量（1 10）来进行的，复现的结果也与原文的结果比较有差异，显然也是因为如何融合数据增强策略的细节原文没有给出而导致的。

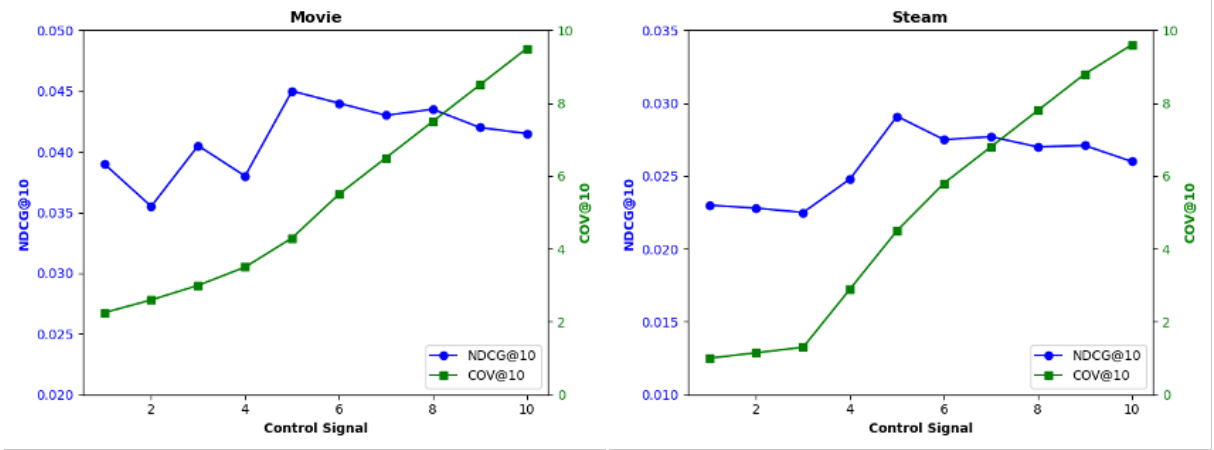


图 5. DLCRec 对多种控制数量的多样性推荐的敏感度（原文）

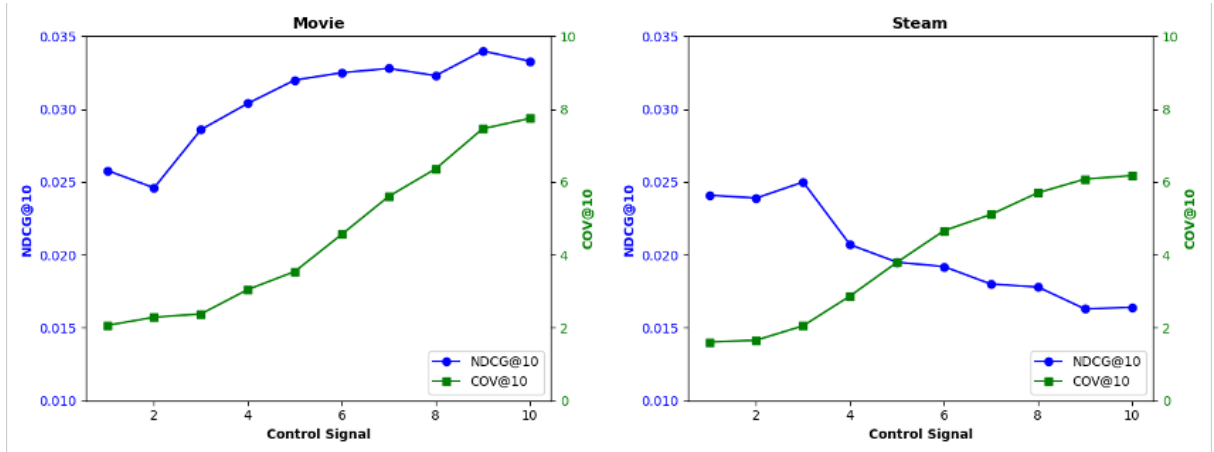


图 6. DLCRec 对多种控制数量的多样性推荐的敏感度（复现）

6 总结与展望

在本研究中，作者提出了 DLCRec，一个新颖的面向多样性的可控推荐框架，它将任务分解为三个子任务：类别预测、类别填充和物品预测。这些子任务独立训练，并根据用户定义的控制数按顺序推断，从而确保对多样性有更精确的控制。为了应对训练数据稀缺和分布偏斜的问题，作者采用了两种数据增强策略，以提高模型对不同多样性需求的鲁棒性和适应性。通过广泛的实验，验证了 DLCRec 在应对不同用户请求时调整推荐多样性的有效性。此外，作者还进行了两项数据消融研究，以验证任务分解框架和数据增强策略的有效性。

本研究提出了一种有前景的方法，用于对 LLM 进行微调，以实现可控的推荐。尽管论文的主要关注点是控制推荐列表的多样性，但所提出的框架也可以适应其他任务，如新颖性

(novelty) 和偶然性 (serendipity)。未来的研究方向可能包括优化 LLM 在列表推荐中的生成能力，进一步提高可控推荐系统的灵活性和有效性。

通过这次实验，我了解到了如何完整地去复现一篇论文的实验，并在这个过程中学习到论文原作者的方法和对应的思想，不仅锻炼了实际的应用能力，还拓展了我的理论知识和思考能力，为日后的科研生活打下更牢固的基础。

参考文献

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. A bi-step grounding paradigm for large language models in recommendation systems. *arXiv preprint arXiv:2308.08434*, 2023.
- [3] Keqin Bao, Jizhi Zhang, Yang Zhang, Xinyue Huo, Chong Chen, and Fuli Feng. Decoding matters: Addressing amplification bias and homogeneity issue for llm-based recommendation. *arXiv preprint arXiv:2406.14900*, 2024.
- [4] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014, 2023.
- [5] Jiaju Chen, Wang Wenjie, Chongming Gao, Peng Wu, Jianxiong Wei, and Qingsong Hua. Treatment effect estimation for user interest exploration on recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1861–1871, 2024.
- [6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [7] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 238–248, 2023.
- [8] Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Zijian Zhang, Wanyu Wang, Yuyang Ye, Shanru Lin, et al. Llm-enhanced reranking in recommender systems. *arXiv preprint arXiv:2406.12433*, 2024.

- [9] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022.
- [10] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- [11] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [12] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [13] Yueqing Liang, Liangwei Yang, Chen Wang, Xiong Xiao Xu, Philip S Yu, and Kai Shu. Taxonomy-guided zero-shot recommendations with llms. *arXiv preprint arXiv:2406.14043*, 2024.
- [14] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Llara: Aligning large language models with sequential recommenders. *arXiv preprint arXiv:2312.02445*, 2023.
- [15] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*, 2023.
- [16] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. A multi-facet paradigm to bridge large language model and recommendation. *arXiv preprint arXiv:2310.06491*, 2023.
- [17] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [18] Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. Aligning large language models for controllable recommendations. *arXiv preprint arXiv:2403.05063*, 2024.
- [19] Aleksandr Petrov and Craig Macdonald. Aligning gptrec with beyond-accuracy goals with reinforcement learning. *arXiv preprint arXiv:2403.04875*, 2024.
- [20] Amit Sharma, Hua Li, Xue Li, and Jian Jiao. Optimizing novelty of top-k recommendations using large language models and reinforcement learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5669–5679, 2024.
- [21] Yancheng Wang, Ziyang Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojian Huang, Yanbin Lu, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*, 2023.

- [22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [23] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, 2024.
- [24] Fan Yang, Zheng Chen, Ziyang Jiang, Eunah Cho, Xiaojian Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*, 2023.
- [25] Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*, 2023.
- [26] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*, 2023.
- [27] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023.