推荐系统中关于数据扩散增强的研究

Group2:

唐宁，吴慧玲，陈一如，刘代宇，孙翰澈

1. 背景

近年来，**推荐系统**的价值日益凸显，已广泛应用于社交媒体、电子商务及视频流媒体平台[4, 5]等多个领域。为提升推荐性能，大量基于用户与物品交互行为的模型相继涌现，例如社交媒体中的分享/点赞、电商平台的评分或视频流媒体的观看时长等[6]。这些系统虽致力于更好地拟合用户行为数据，但学界业界逐渐认识到：基于观测数据的隐式反馈存在固有偏差，可能对推荐系统产生多重影响[9, 10]。

理想情况下，推荐系统应当基于观察到的用户行为捕捉其兴趣的全面维度，从而提供多样、准确的推荐。然而由于偏差的存在，用户行为可能无法准确反映其真实兴趣。 一些经典的场景如：由于系统推荐范围有限，符合用户某些兴趣特征的内容可能尚未获得交互机会，导致反馈缺失，这种现象被称为曝光偏差。此外，由于个体差异，用户会采取不同的交互策略，从而引发选择偏差[10]。比如某些用户可能仅对购买的鞋类商品进行评分，却不对日常用品留下评价，但这并不代表他们对后者缺乏兴趣。

在工业场景中，推荐系统还面临更加复杂的序列推荐的挑战。例如，电商平台通过分析用户点击、购买的时序数据，可精准推荐下一个可能感兴趣的商品；流媒体平台依据用户观看历史预测后续内容偏好。这类模型的核心目标是捕捉用户交互序列中的时序依赖关系，同样的，该场景下也会受到长尾用户的稀疏数据的困扰。真实场景中，大量用户仅与少量物品交互，导致模型难以学习到稳健的用户偏好，传统模型对交互较少的用户泛化能力不足，对其兴趣捕捉存在显著缺陷。

数据增强是缓解上述挑战的重要途径。**扩散模型（Diffusion Models）**近年来已成为数据增强领域强有力的工具，并在学术界获得了广泛应用。大量实证研究一致表明，这类模型能够有效生成高质量、多样化的样本，显著提升数据集的整体丰富度和覆盖范围[11-13]。更重要的是，基于此类扩散模型增强构建的数据集进行训练的目标模型，其最终性能在多样性甚至是准确率的指标下往往也能获得可观的提升，验证了生成数据的有效性和泛化能力。

基于以上观察，我们考虑设计、训练任务相关的扩散模型，在非序列和序列推荐的场景下对训练数据生成伪记录来挖掘用户的隐式偏好，并在ml1m[8]和twitter[7]两个数据集对协同过滤(Collaborative Filtering, CF)、Lightgcn[1]、Bert4Rec [14]三类推荐基模型上进行比对试验，结果表明，在大部分的对比试验中，扩散增强的数据能够提升推荐模型的召回与排序质量。代码可见 https://github.com/downing777/Diffu_aug.git。

2. 任务定义

令$\mathcal{U}$ 和 $\mathcal{I}$ 分别表示用户集和物品集。 $\mathcal{R} \subseteq \mathcal{U} \times \mathcal{I}$ 表示用户和物品之间的交互集合，为了防止数据泄漏，首先划分为 $\mathcal{R}_{\text{train}}$ 与 $\mathcal{R}_{\text{test}}$。一般意义上，我们考虑在 $\mathcal{R}_{\text{train}}$ 上训练扩散模型 $\mathcal{D}$，并通过 $\mathcal{D}$ 生成为交互序列 $\mathcal{R}_{\text{aug}} \subseteq (\mathcal{U} \times \mathcal{I}) \setminus \mathcal{R}_{\text{train}}$。对于推荐模型 $\mathcal{F} : \mathcal{R} \to \mathcal{R}$ 以及评估准则 $\mathcal{E}$，我们的任务是对比在增强前后模型预测的差异 : $||\mathcal{E}[\mathcal{F}(\mathcal{R}_{\text{train}} \cup \mathcal{R}_{\text{aug}}), \mathcal{R}_{\text{test}}] - \mathcal{E}[\mathcal{F}(\mathcal{R}_{\text{train}}), \mathcal{R}_{\text{test}}]||$。

2.1 非序列推荐

在非序列推荐的场景中，我们按照一定比例的比例划分 $\mathcal{R}_{\text{train}}$ 与 $\mathcal{R}_{\text{test}}$， 给定用户的交互记录 $S_u = \{i_1^u, ...i_k^u\}$，其中 k 表示训练集中该用户的交互物品储量。推荐模型的目的是需要给出TopK个最有可能出现的交互记录，即 $\text{Topk}_{i \in \mathcal{I}} P(i|S_u)$。扩散模型的任务即根据训练数据 $\mathcal{R}_{\text{train}}$ 生成伪交互 $\mathcal{R}_{\text{aug}}$，扩散模型的训练可以继续在训练集上进行划分， 增强后的的推荐任务转化为 $\text{Topk}_{i \in \mathcal{I}} P(i|S_u \cup S_{aug}^u)$。

2.2 序列推荐

在序列推荐的场景中，我们考虑最简单的leave last one情形。给定用户 u 的交互序列 $S_u = \{i_1^u, ...i_{n_u-1}^u\}$, 其中 n_u 是用户 u 的交互序列长度，推荐模型需要预测出最有可能下一次交互的物品 $i_{n_u}^u$, 即： $\arg\max_{i \in \mathcal{I}} P(i_{n_u} = i|S_u)$。 为了不引入过多时序上的复杂关系，扩散模型的任务被限定在交互序列的前序插入，即生成 $\mathcal{S}_{aug}^u = \mathcal{D}(S_u) = \{i_{-M}^u, ...i_{-1}^u\}$，对应的，增强后的推荐任务则转化为 $\arg\max_{i \in \mathcal{I}} P(i_{n_u} = i|S_u \cup S_{aug}^u)$。

3. 数据集与评估指标

3.1 数据集

数据集我们考虑了两种常用的推荐数据集，分别是**ml1m**[8]和**Twitter**[7]，对应物品和文本两种推荐场景。在ml1m的原始数据中，用户数为6,040， 物品数为3,416，总交互数量约1M。我们只考虑正向交互，即保留了用户评分高于3的点评记录，同时剔除了交互记录小于5的用户。我们采用二元隐式反馈，即用户物品之间的交互为1/0数值代表有/无交互。Twitter数据集最初被用于机器人检测，我们去除掉了机器人数据，用户推文之间的交互包括点赞、转发。同样的，我们剔除了交互记录少于10条的用户，最终统计数据用户数量为2118，物品数量为7199，整体交互数量为40223。在非序列场景中，数据集按照9:1的比例被划分为训练、测试集，在序列场景中，用户的最后一次交互记录被划分为了测试集。

3.2 评估指标

我们考虑两种对于推荐序列的评估方法， 同时考虑召回物品的相关性以及推荐序列的排序质量。给定一个长度为k的推荐序列与真实的测试样本(relevant items)：

Recall@K

$$\text{Recall@K} = \frac{\text{relevant items in top K}}{\text{all relevant items}}$$

NDCG@K

对于推荐序列，我们同时又希望相关性得分高的物品能够排在推荐序列的靠前位置，相关性得分可以通过数据集中的side-information来获得，例如用户对于物品的打分。记第 i 个物品的相关性得分为 rel_i ，其计算方式涉及两个部分:

- 折损累计增益: $DCG@K = \sum_{i=1}^{K}(rel_i/log_2(i+1))$,
- 理想累计增益: $IDCG@K = \sum_{i=1}^{min(K,|rel|)}(rel_{sorted_i}/log_2(i+1))$,

最终，指标NDCG通过二者比值计算得到: $NDCG@K = \frac{DCG@K}{IDCG@K}$ 。

4. 模型架构

4.1 非序列场景

我们利用一个简易的DDPM的模型架构，对于非序列场景下的交互记录进行建模。模型包含嵌入层、条件引导机制、噪声预测网络，扩散与逆向生成五部分，通过前向扩散和逆向生成过程实现高质量的物品序列生成。

嵌入层

用户和物品的嵌入 $\mathbf{e}_u, \mathbf{e}_i$ 由两个独立的 `nn.embedding` 层生成。

条件引导

$\mathbf{c}_u = \text{AvgPool}(\mathcal{H}_u) + \mathbf{e}_u$, 其中 $\mathcal{H}_u$ 为用户历史交互。

噪声预测网络

由一个两层神经网络构成，通过SiLU函数激活。

扩散过程

前向扩散：给定物品序列嵌入 $\mathbf{x}_0$ ， $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim N(0, I)$ 。

去噪过程

在去噪过程中，引入之前的条件信息 $\mathbf{c}_u$ 用于捕捉当前用户的特征，

$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}_u)\right) + \sigma_t \mathbf{z}$ 。

4.2 序列场景：Diffuasr

在序列推荐的场景中，我们参考了Diffuasr模型[3]的架构，设计了扩散模型的架构与训练、数据增强的工作流。

扩散过程

扩散过程分为**前向扩散**和**逆向生成**两个阶段，通过扩散模型生成高质量的交互序列以增强推荐数据。

1. 前向扩散（Forward Process）

- **输入**：用户的原始交互序列 $S_{\text{raw}} = \{i_1, i_2, \ldots, i_{n_u-1}\}$。
- **嵌入转换**：将待生成的伪序 $S_{\text{aug}} = \{i_{-M}, \ldots, i_{-1}\}$ 中的每个物品 i_j 转换为嵌入向量 $\mathbf{e}_j \in \mathbb{R}^d$，得到初始嵌入序列 $\mathbf{x}_0 = [\mathbf{e}_{-M}, \ldots, \mathbf{e}_{-1}]$（维度 $M \times d$）。
- **噪声添加**：通过马尔可夫链逐步添加高斯噪声，第 t 步的噪声化序列为：

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

其中 $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$，$\beta_s$ 为预设的噪声方差表（采用线性调度策略）。最终 $\mathbf{x}_T$ 近似标准高斯分布 $\mathcal{N}(0, \mathbf{I})$。

2. 逆向生成（Reverse Process）

- **目标**：从噪声 $\mathbf{x}_T$ 重建原始嵌入序列 $\mathbf{x}_0$。
- **噪声预测**：使用 **SU-Net** 预测每一步的噪声 $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$：

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\right) + \sigma_t \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

- **条件注入**：$\mathbf{c} = \text{AvgPool}(S_{\text{raw}})$ 是原始序列的嵌入均值，用于引导生成与用户偏好一致的伪序列。
- **物品解码**：生成嵌入序列 $\hat{\mathbf{x}}_0$ 后，通过余弦相似度匹配物品库 $\mathcal{I}$：

$$i_j = \arg\max_{v_i \in \mathcal{V}} \text{sim}(\hat{\mathbf{e}}_j, \mathbf{e}_i)$$

最终输出伪序列 S_{aug} 并拼接至原始序列前，形成增强序列 $S' = [S_{\text{aug}}, S_{\text{raw}}]$。

3. 引导策略（Guidance Strategies）

为提升生成质量，有多种user信息嵌入的引导策略可以参考，这里我们考虑最一般的不依赖分类模型的场景(**Classifier-Free**），通过条件插值避免预训练依赖：

$$\epsilon = (1+\gamma)\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - \gamma\epsilon_\theta(\mathbf{x}_t, t, \emptyset)$$

$\emptyset$ 表示空条件（如填充向量）。

SU-Net（Sequential U-Net）

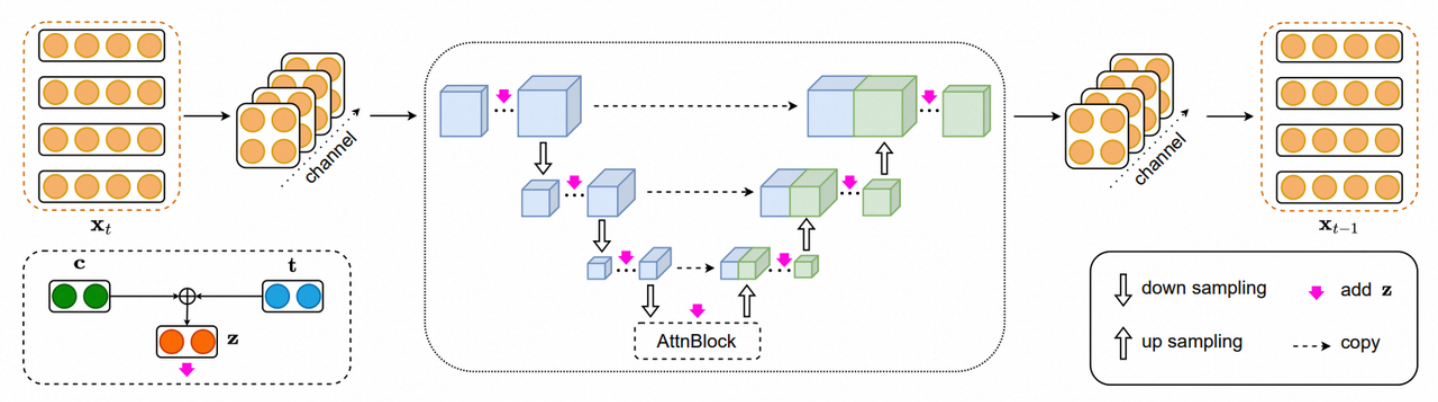
U-Net在扩散模型中应用广泛，但其最初是为图像数据设计的。若直接用于处理嵌入序列，可能丢失序列中的关键信息。因此，我们提出对U-Net的输入结构进行适应性修改，使其兼容嵌入序列特征。SU-

Net结构如下：

1. 输入适配

- **序列重塑**：将嵌入序列 $\mathbf{x}_t$ （$M \times d$）的每个向量 $\mathbf{e}_j$ 重塑为 $\sqrt{d} \times \sqrt{d}$ 矩阵，得到 M 通道的特征图（维度 $M \times \sqrt{d} \times \sqrt{d}$），以适配卷积操作。

2. 网络结构



- **残差块（ResNet Blocks）**：每个块包含卷积层、归一化层和SiLU激活，注入扩散步数 t 与条件 $\mathbf{c}$ 的融合向量 $\mathbf{z} = \mathrm{Embed}(t) + \mathbf{c}$。
- **注意力块（AttnBlock）**：位于网络中部，通过自注意力机制捕获序列依赖：

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

- **下采样/上采样**：采用步幅卷积（下采样）和转置卷积（上采样），跳跃连接保留细节特征。

3. 输出还原

- 将输出特征图（$M \times \sqrt{d} \times \sqrt{d}$）还原为嵌入序列 $\hat{\mathbf{x}}_0$ （$M \times d$），用于逆向生成。

4.3 backbone模型

在非序列场景中，我们选择协同过滤与lightgcn分别代表统计和深度学习模型作为推荐的基模型，而在序列推荐中，我们选择bert4rec作为基模型。

协同过滤（Collaborative Filtering, CF）

协同过滤建立在"群体智慧"的基础上，其核心假设是：相似用户具有相似的偏好，以及用户会喜欢与自己过去喜欢的物品相似的物品。以此来利用群体行为模式预测用户偏好，这种方法不依赖物品本身的特征属性（如电影类型、商品属性），而是完全基于用户与物品的交互历史数据，通过挖掘用户群体中的行为模式来预测个体偏好。

实现原理

- **用户-物品交互矩阵**：构建矩阵 $R_{m \times n}$ （m 表示用户数，n 表示物品数），元素 r_{ui} 表示用户 u 对物品 i 的两类行为特征，其中显示反馈主要即用户主动表达的偏好强度，比如用户对于商品的直接打分；而隐式反馈主要是用户行为隐含的偏好信号，比如用户的购买记录，点击记录等。

- **相似度计算**：相似度度量是协同过滤的算法引擎，其精度直接影响推荐质量，主要有以下两部分相似度计算。
  - 用户相似度：$\text{sim}(u,v)=\cos(\mathbf{r}_u,\mathbf{r}_v)=\dfrac{\mathbf{r}_u\cdot\mathbf{r}_v}{||\mathbf{r}_u||\cdot||\mathbf{r}_v||}$
  - 物品相似度：$\text{sim}(i,j)=\dfrac{|N_i\cap N_j|}{\sqrt{|N_i|\cdot|N_j|}}$（$N_i$ 为喜欢物品 i 的用户集合）
- **预测评分**：得到相似度矩阵后，可以通过加权领域方法对用户-物品对进行预测。主要有两种主流的预测方式
  - 用户协同过滤：$\hat{r}_{ui}=\bar{r}_u+\dfrac{\sum_{v\in N_u}\text{sim}(u,v)\cdot(r_{vi}-\bar{r}_v)}{\sum_{v\in N_u}|\text{sim}(u,v)|}$
  - 物品协同过滤：$\hat{r}_{ui}=\dfrac{\sum_{j\in S_i^k}\text{sin}(i,j)\cdot r_{uj}}{\sum_{j\in S_i^k}|\text{sin}(i,j)|}\quad\forall j\in I_u$

其中 $\bar{r}_u$ 与 $\bar{r}_v$ 分别为用户 u 与其邻域用户 v 的平均评分，N_u 表示与 u 相似并对物品 i 有评分的用户集合。

LightGCN（Light Graph Convolution Network）

LightGCN 是专门为协同过滤推荐任务设计的一种轻量级图卷积网络（Graph Convolutional Network，GCN）变体，其核心目标在于去除传统 GCN 中的冗余计算（如特征变换和非线性激活），仅保留邻域聚合机制，从而既能有效捕获用户与物品之间复杂的高阶关系，又能大幅提升模型训练和推理效率。

实现原理

- **图构建**：在LightGCN中，首先将推荐系统中的用户集 $\mathcal{U}$ 与物品集 $\mathcal{I}$ 构建为一个无向二分图 $\mathcal{G}=(\mathcal{U}\cup\mathcal{I},\mathcal{E})$，其中边集 $\mathcal{E}$ 表示用户与其交互过的物品之间的链接关系。每个节点(用户或物品)在第0层初始化一个低维嵌入向量 $\mathbf{e}_u^{(0)}$ 或 $\mathbf{e}_i^{(0)}$，通常通过随机初始化或预训练得到。
- **传播层**：随后，LightGCN 采用多层邻域聚合 (propagation) 来逐步融合图结构信息。对于第 I 层的节点嵌入，新的更新规则仅包含邻居节点投票，而不进行任何线性变换或激活函数，其中 $\mathcal{N}_u$ 表示与用户 u 相连的物品集合，$\mathcal{N}_i$ 表示与物品 i 相连的用户集合；$\dfrac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$ 为对称归一化系数，保证不同度数节点的嵌入在不同层次上具有可比性。

$$\mathbf{e}_u^{(l+1)}=\sum_{i\in\mathcal{N}_u}\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\mathbf{e}_i^{(l)},\quad\mathbf{e}_i^{(l+1)}=\sum_{u\in\mathcal{N}_i}\frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_u|}}\mathbf{e}_u^{(l)}$$

- **最终嵌入**：在完成 L 层聚合后，LightGCN 会将各层嵌入按照等权重或可学习权重进行线性组合，以综合利用各阶邻域信息：

$$\mathbf{e}_u=\sum_{l=0}^{L}\alpha_l\mathbf{e}_u^{(l)}\quad(\alpha_l=1/(L+1))$$

- **预测**：最终，模型通过计算用户与物品组合嵌入的内积来给出预测分数：

$$\hat{y}_{ui}=\mathbf{e}_u^T\mathbf{e}_i$$

lightgcn省略了传统 GCN 中的特征变换矩阵和非线性激活，使参数量和计算量大幅减少，训练更高效；多层聚合不仅能捕获一阶邻域，也能有效整合二阶、三阶乃至更高阶的用户–物品协同信号。

Bert4Rec

Bert4Rec是一种将Transformer架构引入推荐系统行为序列建模的先进方法，其核心目标在于通过双向自注意力机制充分挖掘用户历史交互序列中的前后文信息，以捕捉用户偏好的动态演变。

实现原理

- **输入编码**：具体而言，给定用户交互序列 $\mathbf{s} = [v_1, v_2, \ldots, v_T]$，Bert4Rec 首先为序列中的每个位置生成两类嵌入：物品嵌入 $\mathbf{E}_{\text{item}}[v_t] \in \mathbb{R}^d$ 和位置嵌入 $\mathbf{E}_{\text{pos}}[t] \in \mathbb{R}^d$，并将它们逐元素相加得到初始输入矩阵。
  - 位置嵌入：$\mathbf{E}_{pos} \in \mathbb{R}^{T \times d}$
  - 物品嵌入：$\mathbf{E}_{item} \in \mathbb{R}^{|\mathcal{V}| \times d}$
  - 最终输入：$\mathbf{H}^{(0)} = \mathbf{E}_{item} + \mathbf{E}_{pos}$
- **Transformer层**：随后，模型通过 L 层堆叠的多头自注意力 (Multi-Head Self-Attention) 和前馈网络(FFN)对该输入进行编码。在第 L 层中，首先计算

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

  其中 $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ 均来源于 $\mathbf{H}^{(l-1)}$ 的线性映射；然后将注意力输出通过前馈网络，并在残差连接后进行层归一化：

$$\mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l-1)} + \text{FFN}(\text{MultiHeadAttn}(\mathbf{H}^{(l-1)})))$$

- **预训练任务**：在预训练阶段，Bert4Rec借鉴BERT的掩码语言模型 (Masked Language Modeling,MLM) 思想：随机掩码序列中若干位置的物品标识，模型需基于剩余未掩码上下文预测被遮蔽的物品类型，即最大化

$$P(v_t | \mathbf{s}_{\setminus t}) = \text{softmax}(\mathbf{W}\mathbf{h}_t^{(L)})$$

  其中 $\mathbf{h}_t^{(L)}$ 是第 L 层对应位置的隐藏表示， $\mathbf{W}$ 为输出映射矩阵。通过这一双向上下文学习，Bert4Rec 能在编码时同时利用物品之前和之后的信息，从而有效捕获用户在行为序列中潜在的偏好变化。
- **微调**：完成预训练后，需要在具体推荐任务中微调，例如给定用户最近 T 次交互，预测其下一次可能交互的物品分布预测下一物品 。

$$P(v_{T+1} \mid \mathbf{s}_{1:T}) = \text{softmax}(\mathbf{W}\mathbf{h}_{T+1}^{(L)}).$$

5. 实验

参数设定

协同过滤基于统计方法，利用0/1交互矩阵生成item embedding， 在基于物品相似度计算生成推荐序列。对于LightGCN我们采用两层卷积层，隐空间维度为32，利用Adam优化器学习了150个epoch，batch size为2048，学习率5e-3。对于Bert4rec，参数设定遵照原论文[3]。在本次工作中，所有用户的增强交互数量固定为3。召回物品数量K取10。

实验数据统计如下:

Backbone		ml1m		Twitter	
		Recall@10	NDCG@10	Recall@10	NDCG@10
CF(non-seq)	w/o	0.0450	0.0185	0.0310	0.0140
	w/	**0.0482**	**0.0199**	**0.0349**	**0.0161**
lightgcn(non-seq)	w/o	0.0994	0.0437	0.0567	0.0272
	w/	**0.1084**	**0.0486**	**0.0572**	**0.0293**
Bert4rec(se q)	w/o	0.0159	0.0267	0.0092	0.0166
	w/	**0.0166**	0.0252	**0.0115**	**0.0173**

注：这里的指标计算与最初的报告中有所不同，为了计算的统一，在序列和非序列的场景下，召回的时候我们强制剔除了训练集中的交互数据，在剩余的物品中进行排序召回。

结果显示，在大部分场景下，推荐模型在经过增强的交互数据中能够更好地捕捉用户偏好，在物品召回的覆盖(recall)以及排序的质量(NDCG)上均能得到提升，显示出扩散模型能够通过引入伪交互的方式给训练数据引入合理的扰动，在嵌入空间上拉近用户与隐式偏好的距离。而在稠密数据集ml1m上，这一方法对于排序质量产生的负面效果，可能的原因是对于用户的全量增强引入了过多噪声，尤其是在序列推荐的场景下造成了较为严重的分布偏移，可以考虑设计更加保守的增强与采样策略。

6. 不足与未来工作

在我们的实验中，基于扩散增强的数据能够有效提升推荐模型的召回与排序能力。然而，本次工作依旧存在多方面的局限性与不足,体现在：

- 缺少和其他数据增强的方法。
- 缺少必要的超参数分析与消融实验。
- 忽略多数推荐数据集上物品的类别信息，在推荐召回阶段事实上可以基于此给出更好的采样策略以提升推荐的多样性指标。

7. 参考文献

1. Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, Meng Wang . Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval,* pp. 639-648. 2020.
2. Sun, Fei, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441-1450. 2019.
3. Liu, Qidong, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. "Diffusion augmentation for sequential recommendation." In *Proceedings of the 32nd ACM International conference on information and knowledge management*, pp. 1576-1586. 2023.
4. Yue Xu et al. “Multi-factor sequential re-ranking with perception-aware diversification”. In:Proceedings of the 29th ACM SIGKDD Conference on knowledge discovery and data mining.2023, pp. 5327–5337.
5. Kartik Sharma et al. “A survey of graph neural networks for social recommender systems”. In:ACM Computing Surveys 56.10 (2024), pp. 1–34.
6. Jianmo Ni, Jiacheng Li, and Julian McAuley. “Justifying recommendations using distantlylabeled reviews and fine-grained aspects”. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019, pp. 188–197.
7. Shangbin Feng et al. “Twibot-22: Towards graph-based twitter bot detection”. In: Advances in Neural Information Processing Systems 35 (2022), pp. 35254–35269.
8. http://files.grouplens.org/datasets/movielens/ml-10m-README.html
9. Shilong Bao et al. “The minority matters: A diversity-promoting collaborative metric learning algorithm”. In: Advances in Neural Information Processing Systems 35 (2022), pp. 2451–2464.
10. Jiawei Chen et al. “Bias and debias in recommender system: A survey and future directions”. In: ACM Transactions on Information Systems 41.3 (2023), pp. 1–39.
11. X. Yang, T. Ye, X. Yuan, W. Zhu, X. Mei and F. Zhou, "A Novel Data Augmentation Method Based on Denoising Diffusion Probabilistic Model for Fault Diagnosis Under Imbalanced Data," in IEEE Transactions on Industrial Informatics, vol. 20, no. 5, pp. 7820-7831, May 2024, doi: 10.1109/TII.2024.3366991.

12. J. Xie, K. Zhang, W. Li, Y. Wu, and L. Zhang, "MosaicFusion: Diffusion Models as Data Augmenters for Large Vocabulary Instance Segmentation," *arXiv:2309.13042 [cs.CV]*, Sep. 2023.
13. S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic Data from Diffusion Models Improves ImageNet Classification," arXiv preprint arXiv:2304.08410, 2023.