

论文《Causal Incremental Graph Convolution for Recommender System Retraining》的复现报告

摘要

现实世界中的推荐系统需要定期重新训练以补充新数据来保持模型的良好性能。在这项工作中，CILightGCN 的作者考虑如何有效地重新训练基于图卷积网络 (GCN) 的推荐模型。为了追求高效率，他们将目标设置为仅使用新数据进行模型更新，同时与模型完全再训练相比不会牺牲太多推荐准确性，因此他提出了增量图卷积模块 (IGC) 和蒸馏碰撞效应模块 (CED) 来对图卷积模型进行增量学习。在本文中，我在复现了 CILightGCN 实验结果的同时，将其增量图卷积模块应用到了先进的多行为推荐模型——级联残差图卷积网络 (简称 CRGCN) 中，并成功在不损失太多性能的情况下高效率地完成了模型重训练。CRGCN 是一种轻量级的多行为推荐模型，该模型可以显式地利用不同行为之间的交互顺序进行学习。

关键词：推荐系统；增量学习；图卷积网络

1 引言

近年来，基于 GCN 的推荐模型（如 PinSAGE [25] 和 LightGCN [4]）引起了很大的关注，这些模型在交互图上执行节点表示学习并表现出良好的性能。它们的核心是邻域聚合，即使用来自其邻居节点的信息来增强节点自身的表示。通过领域聚合操作，图结构信息可以显式地集成到隐向量空间中，从而增强提高用户和物品的表示。在实际应用中，推荐系统需要定期（例如每天）重新训练以补充新的交互数据来保持模型的性能。在这项工作中，作者研究了用于推荐系统中的 GCN 模型再训练问题，该问题目前受到审查相对较少。

使用新的交互数据来更新旧的 GCN 模型存在以下三种简单的策略：

- 完全地重新训练模型：合并旧的交互数据和新的交互数据来执行完整的模型训练。由于使用了所有数据，因此该解决方案保留了最高的保真度。然而，它的内存占用和时间耗费都非常大，因为交互数据通常规模很大并且随着时间的推移而不断增加。
- 使用旧的交互图进行微调。交互数据参与 GCN 的两个部分：构成图结构以执行图卷积操作以及构成模型训练时损失函数的正负对。这种微调策略仅使用新的交互构建训练样例，

同时仍使用完整的图结构。因此，尽管该解决方案的资源成本低于完全地重新训练模型，但由于使用旧的交互图，其成本仍然很高。

- 仅使用新的交互图进行微调：这个策略仅使用新的交互数据来进行模型训练和图卷积，并且进行图卷积中操作时不再使用旧的交互数据构成的图，仅使用新的交互数据构成的图，即增量图。由于增量图的稀疏性较高，从而节省了大量的计算资源和存储资源（见图1）。然而，新的交互数据仅包含用户的短期偏好，这可能与长期表现有很大差异，并且数据更加稀疏。因此，它很容易遭受遗忘和过度拟合问题。

考虑到上述简单策略的优缺点，作者提炼出有效且高效的 GCN 推荐模型再训练的三个重要考虑因素：(1) 分离旧的交互图；(2) 保留旧的（长期）偏好信号；(3) 融合新旧偏好信号。简而言之，我们的目标是仅使用新的交互，实现与完全再训练相当甚至更好的推荐准确性。关键在于如何根据旧节点表示和增量图（即从新交互构建的图，参见图1）来模拟全图卷积的输出（即与在整个图上进行完全再训练相当）。

根据以上三个重要因素，作者首先提出增量图卷积（Incremental Graph Convolution, IGC），它根据目标节点的旧表示和增量图结构来近似目标节点的完全图卷积。具体来说，作者设计了一个节点度同步器来学习基于节点度的节点归一化权重，从而接近全图卷积中的归一化权重。同时，作者还设计了一个表示聚合器来将目标节点的旧表示与新邻居的表示组合起来以有效地编码长期和短期信号 [2]。除此之外，作者还设计了碰撞效应蒸馏（Colliding Effect Distillation, CED）来更新非活跃节点的表示，从而模拟新的交互数据对非活跃节点的表示的影响。简单来说，CED 根据非活跃节点的旧表示和活跃节点表示之间的相似性来提取碰撞效果。

然而，作者提出的 CILightGCN 只利用一种类型的交互数据进行推荐（例如购买记录）。随着电子商务平台上可供选择的产品数量众多，用户最终购买的产品数量非常少，导致购买行为的数据极其稀疏。因此，依赖单一类型行为的模型不能很好地捕捉交互数据有限的用户偏好（即数据稀疏性或冷启动问题），导致性能急剧下降。幸运的是，在与信息系统交互时，还有其他行为，例如点击和收藏，这也提供了用户和物品之间的交互信息。在实际场景中，用户通常采取一系列行为与物品交互以获得更多关于物品的信息，从而准确评估物品是否符合他们的个人偏好。这些交互行为往往服从一定的顺序（例如点击->收藏->购买），更重要的是，不同的行为揭示了用户对目标物品的不同偏好。出于以上动机并且在复现 CILightGCN 的实验结果中发现 CED 这个模块对性能的影响不是很大，所以我们将 CILightGCN 中的 IGC 模块应用到了近年来比较先进的多行为推荐模型——级联残差图卷积网络（Cascading Residual Graph Convolutional Network, CRGCN）[24] 中进行多行为推荐模型的增量学习。

总的来说，本文的主要工作如下：

- 我们首先复现了 CILightGCN 在两个数据集上的效果，并且与原文报告的结果进行了对比，实验结果与原文报告的结果基本一致。除此之外，还在两个数据集上做了对 CED 模块的消融实验。

- 为了能够利用丰富的异构行为信息，我们将 CILightGCN 中的 IGC 模块应用到了近年来先进的多行为推荐模型 CRGCN 中，最终实验结果表明，与完全重新训练模型相比，增量版 CRGCN 能够有效的提高模型训练效率。与简单地微调模型或无视新交互数据的旧模型相比，增量版 CRGCN 能够表现出更好的推荐性能。

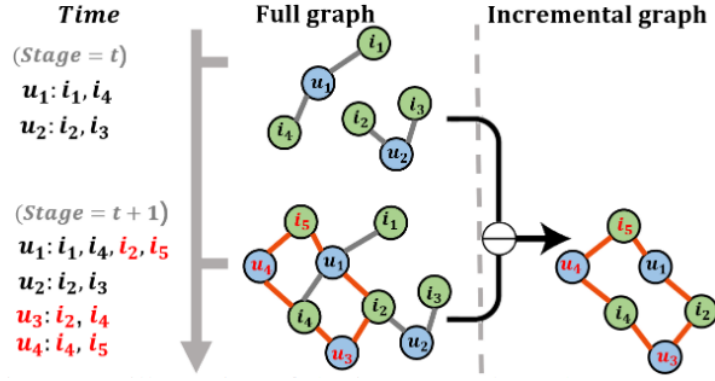


图 1. 增量图与完全图

2 相关工作

2.1 基于图卷积的推荐模型

近年来，图卷积已成为推荐系统的前沿技术 [6, 26]。人们一直致力于设计图卷积模型，从交互图中学习全面的用户和物品表示以进行协同过滤 [4]。除了用户-物品交互之外，有的研究还探索了图卷积模型来编码更多类型的关系，例如物品关系、社交网络 [3, 13, 20] 和知识图谱等 [14, 19]。然而，现有的研究大多侧重于模型的性能，很大程度上忽略了模型训练的效率，这在实际使用中非常重要。

2.2 动态的图卷积模型

在现实世界中存在许多不断变化的图结构，例如社交网络和学术论文引用网络。针对这个问题，之前的大量工作探索动态图卷积来捕获图数据的时间特征 [9, 10]。动态图主要分为以下两类：(1) 根据某一时刻观察到的图；(2) 根据某一个时间间隔观察到的图。例如，DySAT [12] 将不同时刻观察到的图按照时间顺序排列成序列，并使用两个解耦模块对序列内的结构特征和时间特征进行编码。DyRep [16] 将边的出现建模为点过程，并使用神经网络对强度函数进行参数化。有些工作也尝试将动态图卷积模型应用于推荐问题以捕捉用户物品交互的时间模式。例如，JODIE [7] 将 RNN 与 GCN 结合起来，从一系列交互图中学习用户/物品的动态表示。GraphSAIL [23] 可以通过全局和局部结构蒸馏来重新训练基于图的推荐模型。尽管动态图卷积在捕捉短期和长期偏好方面取得了成功，但它需要长期的历史互动来进行训练，这会导致内存和时间成本增加。

2.3 基于因果推理的推荐模型

人们越来越关注将因果推理技术融入各种机器学习应用 [15, 17]。这些方法也成功地应用于推荐系统以解决偏差问题并提高模型可靠性。例如 DecRS [18]、MACR [22] 和 KDCRec [8] 从因果关系的角度分析了推荐的偏差，并使用反事实学习技术来消除偏差。同样，干预技术也可用于消除偏差 [27]。现有的工作重点是利用因果推理技术来消除偏差，而 CILightGCN 作者专注于模型再训练和碰撞效应蒸馏，其中碰撞效应蒸馏是一种新的因果推理技术。

3 本文方法

我们将随着时间到达的交互数据表示为 $\mathcal{I} = \{I_0, I_1, \dots, I_{t-1}, I_t\}$, 其中 I_t 表示在第 t 阶段收集到的交互数据。交互数据所构成的图表示为 $\mathcal{G} = \{G_0, G_1, \dots, G_{t-1}, G_t\}$, 其中 G_t 表示根据第 t 阶段的交互数据 I_t 构成的用户-物品二部图, 也即增量图。在单行为推荐问题中, I_t 是单个行为的交互数据, 同时 G_t 是由单个行为的交互数据构成的同构图。在多行为推荐问题中, I_t 是多个行为 (例如点击, 购买) 的交互数据, 同时 G_t 可以是分为不同行为的交互数据构成的同构图, 也可以是不同行为共同构成的异构图。一个阶段可以是任何时间段 (例如, 一个/多个小时/天), 也可以是收集的交互数据到达一定数量之时, 这取决于预期的模型新鲜度和我们可以承受的训练成本。

那么为了达到高效率重训练图卷积模型的目标, 我们将任务形式化如下:

$$\theta_{t-1} \xrightarrow[\text{retrain}]{(I_t, G_t)} \theta_t \xrightarrow[\text{serve}]{} I_{t+1}, \quad (3.1)$$

其中 θ_t 为在第 t 阶段模型训练后得到的模型参数。

在这个公式中, 我们仅使用最新阶段的交互 I_t 和增量图 G_t 进行重新训练。具有参数 θ_t 的模型用于下一阶段 $t+1$ 的重训练, 因此我们使用 I_{t+1} 的数据来评估模型重训练后的性能。

3.1 本文方法概述

在本节中我们首先回顾了应用在推荐问题中的一般图卷积操作, 随后我们分别介绍了 CI-LightGCN 中的两个模块——IGC 和 CED, IGC 模块完成近似全图卷积的图卷积操作, CED 模块则是通过使用碰撞效应蒸馏来更新非活跃节点的表示。最后我们介绍了我们改进后的增量版 CRGCN, 我们将 IGC 模块应用在了多行为推荐模型 CRGCN 上, 使得 CRGCN 能够实现高效率的模型重训练。

3.2 一般图卷积推荐模型

我们在本小节回顾图卷积在协同过滤中的工作原理, 假设我们在直到阶段 t 的全图上执行图卷积操作, 此时的图记为 $G_{0 \sim t}$, 那么常用的图卷积公式为:

$$e_{i,t}^{(l+1)} = \sigma \left(\frac{1}{\sqrt{d_{i,0 \sim t}}} \overbrace{\sum_{j \in \mathcal{N}_{i,0 \sim t}} \frac{1}{\sqrt{d_{j,0 \sim t}}} e_{j,t}^{(l)}}^{\text{neighborhood aggregation}} \right), \quad (3.2)$$

其中 $e_{j,t}^{(l)}$ 表示节点 j 在第 l 层的表示, $\mathcal{N}_{i,0 \sim t}$ 表示节点 i 在图 $G_{0 \sim t}$ 中的邻居, $d_{i,0 \sim t}$ (节点累积度) 等于 $\mathcal{N}_{i,0 \sim t}$ 中的节点数量。图卷积的核心是邻居聚合, 它为目标节点 i 聚合邻居节点的表示。节点度起到归一化的作用, 对 GCN 的性能有很大影响 [4]。特征变换函数 $\sigma(\cdot)$ 有多种形式, 如线性和双线性映射。在这项工作中, 我们重点关注邻居聚合, 省略特征变换函数 $\sigma(\cdot)$ 。

第 0 层的隐向量是需要学习的模型参数, 这些参数通过最小化损失函数来训练:

$$\sum_{(u,i) \in I/I^-} L(y_{u,i}, \hat{y}_{u,i}) + \lambda \|\theta\|^2, \quad (3.3)$$

其中 I^- 是负样本, $y_{u,i}$ 是交互的标签, $\hat{y}_{u,i}$ 是相应的模型预测值, 一般通过用户和物品隐向量的内积得到。 $L(\cdot)$ 是推荐损失函数, 例如成对的贝叶斯个性化排序 (BPR) [11] 和逐点交叉熵 [5], λ 是用于 L2 正则化的超参数。

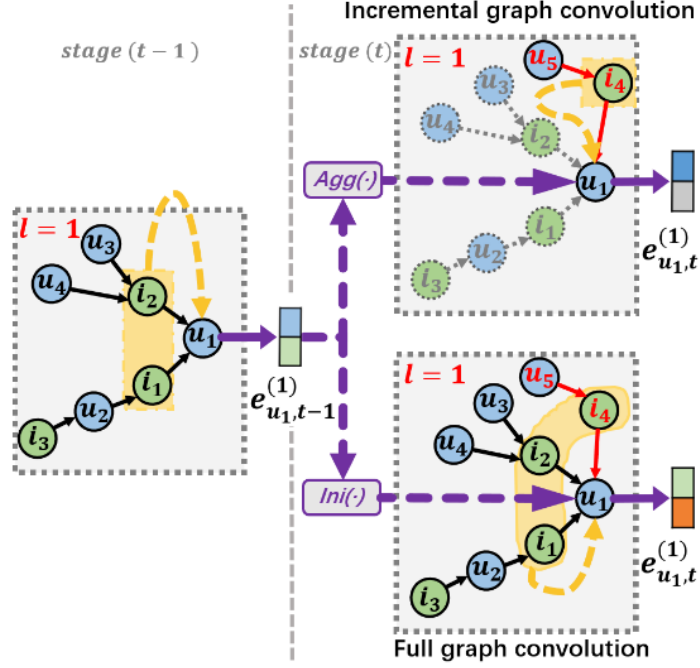


图 2. 增量图卷积

3.3 增量图卷积

随着时间增加, 数据越来越多, 完全重新训练的成本会越来越高。由于 $N_{i,0 \sim t}$ 等于 $N_{i,0 \sim t-1} \cup N_{i,t}$, 如果模型重新训练可以绕过 $N_{i,0 \sim t-1}$, 也即仅使用 $N_{i,t}$ 来近似公式 (3.2) 中的全图卷积, 那么将可以提高训练效率。假设旧的表示 $e_{i,t-1}^{(l+1)}$ 是在 $G_{0 \sim t-1}$ 上训练的, 那么它能够保存在 $N_{i,0 \sim t-1}$ 上的图结构信息。因此, 将它与仅新的邻居节点表示进行巧妙地结合可以很好地近似全图卷积的结果。全图卷积和增量图卷积的区别如图2所示。故而对于目标节点 i 的增量图卷积公式可以表示如下:

$$e_{i,t}^{(l+1)} = \frac{1}{\sqrt{d'_{i,t}}} \cdot \phi \left(\sqrt{d_{i,0 \sim t-1}} \cdot e_{i,t-1}^{(l+1)}, \sum_{j \in N_{i,t}} \frac{1}{\sqrt{d'_{j,t}}} \cdot e_{j,t}^{(l)} \right), \quad (3.4)$$

其中 $d'_{j,t} = f(d_{j,0 \sim t-1}, d_{j,t})$ 表示由节点度同步器得到的归一化权重, ϕ 为邻居信息聚合器, 用于聚合历史邻居信息和当前邻居信息的聚合器。

具体来说, 节点度同步器的计算如下:

$$f(d_{j,0 \sim t-1}, d_{j,t}) = \beta \cdot d_{j,0 \sim t-1} + d_{j,t}, \quad (3.5)$$

其中 β 通过一个 1×1 的线性层训练来控制旧节点度的影响。

而邻居信息聚合器则是通过堆叠新邻居信息和旧邻居信息并将它们传入到一个卷积层:

$$p \left(\left\langle w_f^{(l)}, \left[\sqrt{d_{i,0 \sim t-1}} \cdot e_{i,t-1}^{(l+1)}, \sum_{j \in N_{i,t}} \frac{1}{\sqrt{d'_{j,t}}} \cdot e_{j,t}^{(l)} \right] \right\rangle \right), \quad (3.6)$$

其中 $p(\cdot)$ 是一个池化操作, $w_f \in \mathbb{R}^{2 \times 1}$ 表示 CNN 层的第 f 个滤波器, 并且一层中可以有多
个滤波器。通过这种方式, 聚合器可以在训练过程中调整长期和短期偏好的重要性。

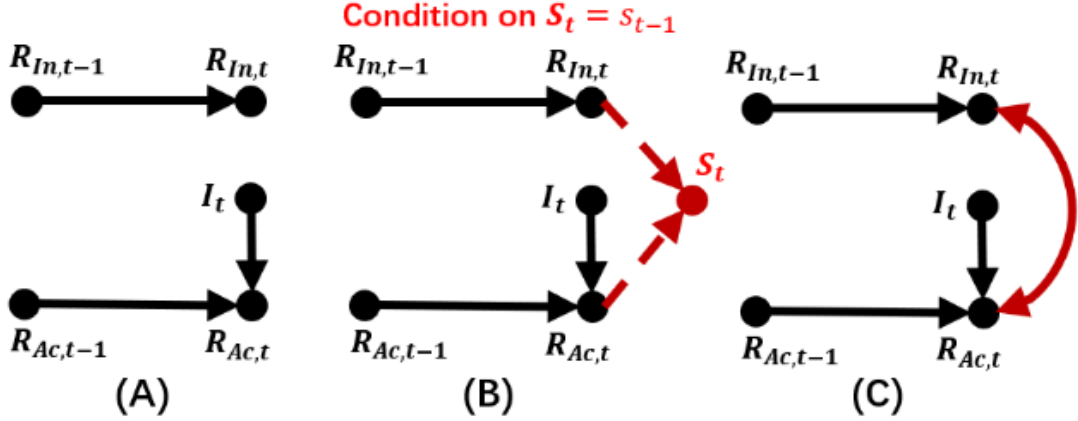


图 3. 碰撞效应蒸馏

3.4 碰撞效应蒸馏

作者将 IGC 应用在先进的协同过滤图卷积模型 LightGCN 中 [4], 得到 I-LightGCN。在 I-LightGCN 中, 通过堆叠 L 层 IGC 层, 我们则可以将所有节点的最终表示通过对所有图卷积层的输出求均值得到, 如下所示,

$$r_{i,t} = \frac{1}{L} \sum_{l=0}^L e_{i,t}^{(l)}, \quad (3.7)$$

我们将出现在 I_t 中的节点称为活跃节点, 其余节点称为非活跃节点。相应地, 我们将活跃节点和非活跃节点的表示分别记为 $R_{Ac,t}$ 和 $R_{In,t}$ 。同样地, 它们的旧表示分别表示为 $R_{Ac,t-1}$ 和 $R_{In,t-1}$ 。

由于我们使用新数据 I_t 来训练 I-LightGCN, IGC 主要更新活跃节点的表示, 这就会导致非活跃节点表示面临过时的问题。原因在于与非活跃节点对应的参数 (即节点隐向量) 并不参与训练过程 (例如反向传播)。鉴于此, 作者考虑从两个方向来适当地更新非活跃节点的表示: 1) 直接将来自活跃节点的新偏好信号注入到非活跃节点的表示中; 2) 将非活跃节点参数纳入训练目标, 间接推动其表示更新。

直接更新非活跃节点的表示。 为了实现这一目的, 作者设计了碰撞效应蒸馏 (CED), 公式如下:

$$\tilde{r}_{m,t} = \gamma_1 \cdot r_{m,t} + \frac{1 - \gamma_1}{K} \cdot \sum_{n \in KNN(R_{Ac,t-1,m}, K, \delta)} r_{n,t}, \quad (3.8)$$

其中 m 是阶段 t 中的一个非活跃节点, $\tilde{r}_{m,t}$ 是连接 K 个活跃节点后的最终表示。 $KNN(\cdot)$ 表示最近邻搜索操作, 它根据旧的节点表示 $r_{m,t-1}$ 与 $R_{Ac,t-1}$ 之间的距离度量 δ (例如欧几里得距离), 计算出距离节点 m 最近的 K 个活跃节点。 $\gamma_1 \in [0, 1]$ 是一个超参数, 用于控制活跃节点的影响程度。简单地说, CED 利用前一阶段表现出相似属性的活跃节点的最新状态来更新非活跃节点的表示。

间接更新非活跃节点的表示。虽然公式 (3.8) 更新了非活跃节点的表示，但由于我们仅从 I_t 构建训练数据，非活跃节点参数仍然未被触及。因此，CED 也对活跃节点进行操作，其公式表示为：

$$\tilde{r}_{n,t} = \gamma_2 \cdot r_{n,t} + \frac{1 - \gamma_2}{K} \cdot \sum_{m \in KNN(R_{In,t-1}, n, K, \delta)} r_{m,t}, \quad (3.9)$$

通过这种方式，非活跃节点参数被纳入目标函数，可以在重新训练过程中得到更新。

在这项工作中，作者将 IGC 和 CED 都应用于 LightGCN，得到的模型称为 CI-LightGCN。在得到 CED 的输出后，使用 $\tilde{r}_{u,t}$ 和 $\tilde{r}_{i,t}$ 的内积来生成一对用户 u 和物品 i 的预测结果：

$$\hat{y}_{u,i} = \langle \tilde{r}_{u,t}, \tilde{r}_{i,t} \rangle. \quad (3.10)$$

4 CRGCN 模型介绍

CRGCN 是一个多行为推荐模型，它由一组具有残差设计的 LightGCN（根据行为交互顺序排序）组成，采用级联结构根据行为交互顺序连接多个残差块，将前一个行为图中学到的用户/物品表示传递到下一个行为图中 [24]。通过这种方式，CRGCN 在学习过程中明确利用了不同行为之间的偏好关系，用户偏好可以通过行为交互顺序不断学习和细化。CRGCN 的模型架构如图4所示。

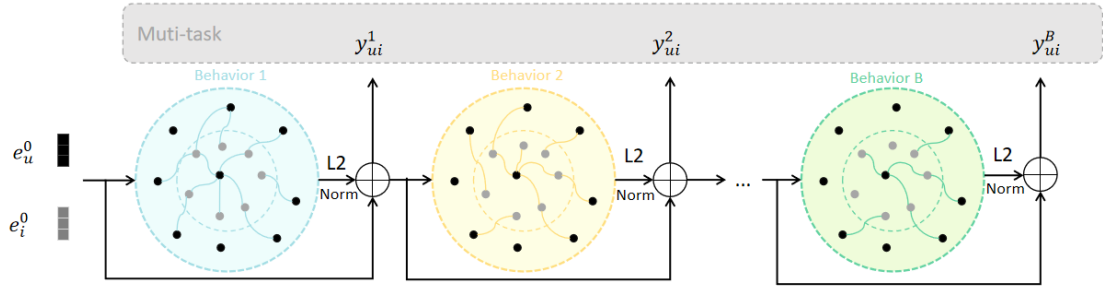


图 4. CRGCN 模型示意图

在单个行为图中，它采用了 LightGCN 中的聚合函数，即：

$$e_u^{(l)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(l-1)}, e_i^{(l)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} e_u^{(l-1)}, \quad (4.1)$$

其中 $\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$ 表示归一化系数， $e_u^{(l-1)}$ ($e_i^{(l-1)}$) 表示来自第 $(l-1)$ 层的用户（物品）输出， $e_u^{(l)}$ 和 $e_i^{(l)}$ 是 GCN 的最终输出（即行为特征）。

在每个级联块中，将在该块中学习到的用户/物品表示与输入（即前一个块的输出）融合，以保留前一个块的特征。使用求和方法将这两个表示合并作为输出。由于 GCN 学习后的表示数值可能与块的输入处于不同范围，直接求和可能导致一个特征表示在生成的结果中占主导地位，使另一个可忽略不计。为避免此问题，CRGCN 在求和前采用 L_2 归一化对用户/物品表示进行归一化操作：

$$\tilde{e}_u = \frac{e'_u}{\|e'_u\|_2}, \tilde{e}_i = \frac{e'_i}{\|e'_i\|_2}, \quad (4.2)$$

归一化之后，通过残差连接将当前块中学习到的用户/物品表示与前一个块的用户/物品表示融合，即：

$$e_{u_{out}} = e_{u_{in}} + \tilde{e}_u, e_{i_{out}} = e_{i_{in}} + \tilde{e}_i, \quad (4.3)$$

其中 $e_{u_{in}}$ 和 $e_{i_{in}}$ 是当前块的输入用户和物品表示，也是前一个块的输出。 $e_{u_{out}}$ 和 $e_{i_{out}}$ 分别表示当前块的输出用户和物品表示。

有了设计好的残差块，随后按照行为交互顺序连接残差块，如图4所示，在这种结构中，前一个残差块的输出被用作下一个块的输入，以便将从一种行为中提取的行为信息传递到下一个行为。具体来说，用户/物品表示在级联残差块中是这样学习和细化的：

$$\begin{cases} e^1 = e^0 + \tilde{e}^1 \\ e^2 = e^1 + \tilde{e}^2 \\ \dots \\ e^B = e^{B-1} + \tilde{e}^B \end{cases}, \quad (4.4)$$

其中 B 是行为的数量， e^{B-1} 表示第 B 个残差块的输入（即第 $(B-1)$ 个残差块的输出）， \tilde{e}^B 表示从第 B 种行为中学习到的归一化表示， e^0 表示初始化的用户和物品表示。

CRGCN 为每种行为设置一个损失函数以监督行为特征的学习过程。如图4所示，CRGCN 模型在对每种行为进行学习后，可以获得每个用户的表示集合 $\{e_u^1, e_u^2, \dots, e_u^B\}$ 和每个物品的表示集合 $\{e_i^1, e_i^2, \dots, e_i^B\}$ ，其中 B 表示行为的数量。然后，通过计算两者的内积来获得用户-物品交互的得分 y_{ui} ，如下所示：

$$y_{ui} = e_u^\top e_i. \quad (4.5)$$

与 CILightGCN 相同，CRGCN 采用 BPR 损失函数确保观察到的用户-物品对的得分高于未观察到的。以第一种行为为例，损失函数公式如下：

$$\mathcal{L}_1 = \sum_{(u,i,j) \in O} -\ln \sigma(y_{ui} - y_{uj}), \quad (4.6)$$

其中 $O = \{(u, i, j) | (u, i) \in R^+, (u, j) \in R^-\}$ 被定义为正样本和负样本对， $R^+(R^-)$ 表示在当前行为中已观察到（未观察到）的样本。这里 $\sigma(\cdot)$ 表示 sigmoid 函数。其他行为的损失函数类似。然后我们可以得到损失函数集 $\mathcal{B} = \{\mathcal{L}_1, \dots, \mathcal{L}_b, \dots, \mathcal{L}_B\}$ ，其中 b 是第 b 种行为。根据多任务学习（Muti-Task Learning, MTL），将每种行为下的学习视为一个任务。最终损失是不同行为的所有损失的求和。因此 CRGCN 最终的损失函数公式如下：

$$\mathcal{L} = \sum_{i \in \mathcal{B}} \mathcal{L}_i + \beta \cdot \|\Theta\|_2. \quad (4.7)$$

5 复现细节

5.1 与已有开源代码对比

本文主要复现的模型是 CILightGCN，该论文的作者公开了其代码，除此之外，多行为推荐模型 CRGCN 的作者也有公开的源码。因此我在参考了两个模型的源码的情况下，在

CRGCN 的源码的基础上添加了 CILightGCN 的 IGC 模块，使得 CRGCN 能够实现增量训练。我撰写的核心代码如下：

```
def gcn_propagate(self):
    """
    gcn propagate in each behavior
    """
    all_embeddings = {}
    total_embeddings = torch.cat([self.user_embedding.weight, self.item_embedding.weight], dim=0)
    for behavior in self.behaviors:
        layer_embeddings = total_embeddings
        mean_embeddings, _0, _1, allLayerEmbs, degree_molecular, degree_Denominator, old_user_degree, old_item_degree = self.get_layers_weight(
            behavior, layer_embeddings)
        del _0, _1
        old_degree = torch.cat([old_user_degree, old_item_degree], dim=0)
        old_scale = self.oldscale_forward(old_degree)
        old_scale = old_scale.pow(0.5)
        old_scale = torch.mul(degree_molecular, old_scale)
        new_scale = degree_Denominator
        rscale_vec = torch.div(old_scale, new_scale + 1e-9)
        old_layer_embeddings = self.LastStage_embeddings[behavior]
        new_layer_embeddings = [total_embeddings]
        for layer in range(self.behaviors_layers[behavior]):
            old_layer_embeddings[layer] = old_layer_embeddings[layer].to(self.device)
            old_layer_embeddings[layer] = old_layer_embeddings[layer] * rscale_vec
            new_layer_embeddings.append(
                self.transfer_forward(old_layer_embeddings[layer], allLayerEmbs[layer], behavior, layer))

        new_layer_embeddings = torch.stack(new_layer_embeddings, dim=1)
        layer_embeddings = torch.mean(new_layer_embeddings, dim=1)
        layer_embeddings = F.normalize(layer_embeddings, dim=-1)
        total_embeddings = layer_embeddings + total_embeddings
        all_embeddings[behavior] = total_embeddings
    return all_embeddings
```

图 5. 增量版 CRGCN 核心代码

该核心代码中实现了 CRGCN 中每一个行为图进行增量图卷积操作时同时聚合历史领域信息和前一个级联块的输出信息的核心功能。

5.2 实验数据集和评估方式

因为 CILightGCN 是单行为推荐模型，所以它采用的数据集是由单个行为交互数据构成的，原文采用的数据集是 Gowalla 和 Yelp。而 CRGCN 是多行为推荐模型，因此我采用的数据集是有多个行为交互数据的 JD 数据集。Yelp 数据集来自于 2019 年的 Yelp Challenge，它有 10 多年的时间里了客户和本地业务之间的交互记录。Gowalla 数据集保存了在 go.gowalla.com 网站上一年中的签到记录。JD 数据集是一个从中国最大的电子商务平台之一——京东上收集的开放数据集。该数据集中有四种类型的行为，包括点击、标签为喜欢、加入购物车和购买四种行为。这三个数据集的统计信息如表 1 所示。

表 1. 数据集统计信息

Dataset	Users	Items	Interactions
Yelp	122,816	59,082	3,014,421
Gowalla	29,858	40,981	1,027,370
JD	11,367	13,520	359,564

其中单行为数据集的增量划分与原文保持一致，而多行为的增量划分与之前的工作的划分方法保持一致 [1, 21, 23]，我将交互数据按时间进行排序后，按照 0.6、0.1、0.1、0.1、0.1 的比例分为五个部分，随后对后面四个 0.1 比例的数据进行处理，删除后面四个 0.1 比例的数据中不存在于第一个 0.6 比例数据中的 user 和 item，此时就得到了最终的五个部分的数据。对

于后面四个 0.1 比例的数据，根据时间顺序，再分别划分前半部分时间的购买数据作验证集，后半部分时间的购买数据做测试集。那么增量训练模型的分为两个阶段：(1) 预训练：取 0.6 之前的数据作为训练集进行预训练，取 0.6-0.65 的购买数据作为验证集，取 0.65-0.7 的购买数据作为测试集；(2) 增量训练：取 0.6-0.7 的数据作为增量块进行训练，取 0.7-0.75 的购买数据作为验证集，取 0.75-0.8 的购买数据作为测试集，后面几个增量块的增量训练、验证集测试集划分以此类推。

为评估模型性能，在单行为推荐和多行为推荐问题上均采用与 CILightGCN 论文中相同的评估指标，即 top-k 的召回率 (Recall@k) 和归一化折损累计增益 (NDCG@k)，其中在单行为推荐上设置 k=5,20, 多行为推荐上设置 k=5,10,20,50。

5.3 创新点

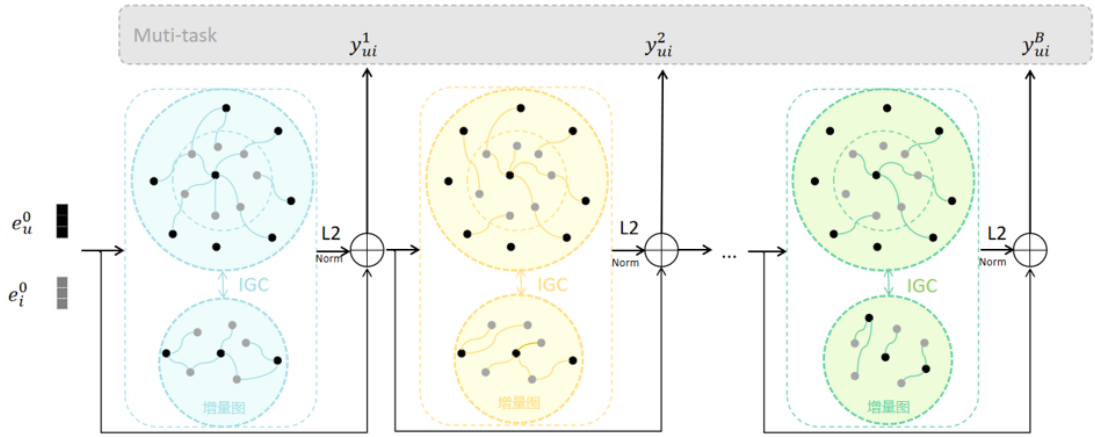


图 6. 增量版 CRGCN 模型架构图

在本文中我将单行为推荐模型 CILightGCN 中的 IGC 模块应用到了多行为推荐模型 CRGCN 中得到增量版 CRGCN，其模型架构如图6所示。为了使得每一个行为图内能够实现增量学习以及保留行为图之间的级联关系，我们将每个行为图下的增量图卷积操作设计为以下公式：

$$e_{u,t}^{b,(l+1)} = \begin{cases} \frac{1}{\sqrt{d'_{u,t}}} * \varphi \left(\sqrt{d_{u,0 \sim t-1}} * e_{u,t-1}^{b,(l+1)}, \sum_{i \in \mathcal{N}_{u,t}} \frac{1}{\sqrt{d'_{i,t}}} * e_{i,t}^{b,(l)} \right), & \text{当 } l > 0 \\ \frac{1}{\sqrt{d'_{u,t}}} * \varphi \left(\sqrt{d_{u,0 \sim t-1}} * e_{u,t-1}^{b,(1)}, \sum_{i \in \mathcal{N}_{u,t}} \frac{1}{\sqrt{d'_{i,t}}} * e_{i,t}^{b-1} \right), & \text{当 } l = 0 \end{cases}, \quad (5.1)$$

这里列出了第 t 阶段第 b 个行为图下用户表示的更新方式，物品表示的更新方式同理。可以看到，当级联块第一次进行图卷积时需要聚合上一阶段的邻居信息和上一个级联块的行为特征，通过这种方式我们能够保留用户/物品的历史邻居信息和上一个行为学习到用户偏好和物品特征。在后续的实验中也看到与完全重新训练模型相比，增量版 CRGCN 能够有效的提高模型训练效率。与简单地微调模型或无视新交互数据的旧模型相比，增量版 CRGCN 能够表现出更好的推荐性能。

6 实验结果分析

这一章主要展示 CILightGCN 的复现结果以及增量版 CRGCN 与三个朴素策略的性能对比实验结果以及效率对比实验结果。

6.1 CILightGCN 复现结果

表 2. CILightGCN 复现结果

	Yelp				Gowalla			
	Recall@5	Recall@20	NDCG@5	NDCG@20	Recall@5	Recall@20	NDCG@5	NDCG@20
原文	0.3284	0.5695	0.2294	0.2956	0.3209	0.5421	0.2272	0.2908
复现	0.3179	0.5614	0.2197	0.2896	0.3181	0.5382	0.2251	0.2883

如表2所示，在两个数据集上复现的结果与原文报告的结果非常接近。同时我也做了对 CILightGCN 的 CED 模块的消融实验，消融实验结果如下所示：

表 3. CED 模块的消融实验

	Yelp				Gowalla			
	Recall@5	Recall@20	NDCG@5	NDCG@20	Recall@5	Recall@20	NDCG@5	NDCG@20
with CED	0.3179	0.5614	0.2197	0.2896	0.3181	0.5382	0.2251	0.2883
w/o CED	0.3146	0.5507	0.2196	0.2873	0.3172	0.5367	0.2245	0.2875
下降百分比%	-1.04%	-1.91%	-0.05%	-0.79%	-0.28%	-0.28%	-0.27%	-0.28%

如表3所示，从 CED 模块在两个数据集上的消融实验结果可以看到，在删去了 CED 模块后，在 Yelp 数据集和 Gowalla 数据集上分别最多下降了 1.04% 和 0.28% 的性能，这说明 CED 模块虽然能够带来性能提升，但是提升的幅度并不大。所以将 CRGCN 改为增量版本时就不考虑 CED 模块了，我仅将 IGC 模块扩展到 CRGCN 中以得到增量版 CRGCN。

6.2 增量版 CRGCN 实验结果

在本小节中，我主要将展示在三个增量块上增量版 CRGCN 与三个朴素策略的性能对比以及效率对比。这三个朴素策略分别是：(1) ignore: 无视新到达的数据，继续沿用历史参数进行推荐；(2) finetune: 使用新到达的数据，利用历史参数初始化对新到达的数据进行模型微调；(3) full retrain: 合并新到达的数据和完全历史数据重新训练模型。

三个增量块上的实验结果如下所示，表4、表5和表6分别记录了在三个增量块上增量训练 CRGCN 和三个朴素策略的性能，我们使用粗体标记最佳性能，下划线标记次优性能。从表4、表5和表6中分别可以看到无论是第几个增量块到达时，采用完全重训练策略一直能够保持最佳性能，这是因为完全重训练合并了新旧数据进行训练，能够充分学习用户物品表示。ignore 策略在大部分情况下性能是最差的，这是因为它沿用历史参数进行推荐而不进行重训练，这样会捕捉不到用户近期的偏好而造成推荐性能下降。finetune 策略虽说能比 ignore 策略在

大部分情况下要好，但它没有有意的保留历史领域信息，因此它在大部分情况下性能都比我们的增量版 CRGCN 差。从表6中可以看出，在第三个增量块到达时，我们的增量版 CRGCN 已经完全表现出了次优性能，仅次于完全重训练。

总的来说，增量版 CRGCN 与 ignore、finetune、full retrain 三个朴素策略对比，性能上优于 ignore 和 finetune，虽不如 full retrain 但差距逐渐缩小；效率上与 finetune 相当，比 full retrain 加速近一倍，这表明增量版 CRGCN 能够实现高效重训练。

表 4. 第一个增量块上的性能对比

	ndcg@5	recall@5	ndcg@10	recall@10	ndcg@20	recall@20	ndcg@50	recall@50
ignore	0.078	0.1049	0.1070	<u>0.1873</u>	0.1245	0.2454	0.1390	0.3056
finetune	<u>0.1087</u>	0.1250	0.1251	0.1664	0.1429	0.2217	0.1579	0.2865
incremental	0.1047	<u>0.1268</u>	<u>0.1266</u>	0.1830	<u>0.1483</u>	<u>0.2521</u>	<u>0.1678</u>	<u>0.3345</u>
full retrain	0.1108	0.1595	0.1322	0.2164	0.1506	0.2771	0.1704	0.3621

表 5. 第二个增量块上的性能对比

	ndcg@5	recall@5	ndcg@10	recall@10	ndcg@20	recall@20	ndcg@50	recall@50
ignore	<u>0.0481</u>	<u>0.0674</u>	<u>0.0639</u>	0.1098	0.0800	0.1648	0.0973	0.2371
finetune	0.0443	0.0664	0.0567	0.0981	0.0704	0.1409	0.0886	0.2185
incremental	0.0424	0.0652	0.0626	<u>0.1204</u>	<u>0.0829</u>	<u>0.1884</u>	<u>0.1014</u>	<u>0.2653</u>
full retrain	0.0698	0.0978	0.0930	0.1604	0.1117	0.2219	0.1345	0.3201

表 6. 第三个增量块上的性能对比

	ndcg@5	recall@5	ndcg@10	recall@10	ndcg@20	recall@20	ndcg@50	recall@50
ignore	0.0544	0.0755	0.0675	0.1090	0.0848	0.1653	0.1011	0.2364
finetune	0.0515	0.0703	0.0717	0.1227	0.0917	0.1877	0.1121	0.2743
incremental	<u>0.0637</u>	<u>0.0922</u>	<u>0.0788</u>	<u>0.1316</u>	<u>0.1005</u>	<u>0.2042</u>	<u>0.1205</u>	<u>0.2888</u>
full retrain	0.0916	0.1193	0.1113	0.1707	0.1324	0.2409	0.1517	0.3244

回顾我们做增量学习的初衷，是为了在短时间内快速更新模型参数并且与完全重训练相比不损失太大的性能。我们做了与微调和完全重训练的效率对比实验，实验结果如下所示：

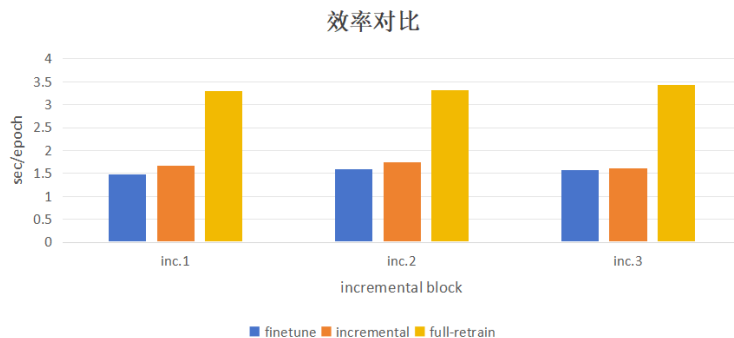


图 7. 效率对比实验结果

在图7中可以看到，增量版 CRGCN 的效率与微调相当，与完全重训练相比加速了几乎一倍。总的来说，增量版 CRGCN 能够实现高效的重训练，并且在性能方面优于不进行重训练和微调后的 CRGCN 模型，在训练效率方面每一个 epoch 比重新训练模型加速了几乎一倍，这说明我们成功将 IGC 模块迁移到多行为模型 CRGCN 实现了高效的重训练。

7 总结与展望

在这项工作中，CILightGCN 作者提出增量图卷积 (IGC)，通过节点度同步器和表示聚合器近似目标节点的完全图卷积，还设计了碰撞效应蒸馏 (CED) 更新非活跃节点表示。本文将 IGC 模块应用于多行为推荐模型 CRGCN，使 CRGCN 能利用丰富异构行为信息实现高效重训练。通过实验结果表明，增量版 CRGCN 的效率与微调相当，与完全重训练相比加速了几乎一倍。这表明增量版 CRGCN 能够实现高效的重训练，在性能和效率方面达到了较好的平衡，成功将 IGC 模块迁移到多行为模型 CRGCN 并实现了预期目标。

尽管增量版 CRGCN 在性能和效率上取得一定成果，但仍有提升空间。可进一步探索如何更好地融合不同行为信息，优化 IGC 模块中节点度同步器和表示聚合器的设计，使模型能更精准捕捉用户偏好，提高推荐准确性。除了文中提到的重训练策略，可探索其他更高效且性能良好的策略，如结合强化学习、迁移学习等技术，开发新的增量学习方法，更好适应数据动态变化和模型快速更新需求。

参考文献

- [1] Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2832–2836, 2021.
- [2] Sihao Ding, Fuli Feng, Xiangnan He, Yong Liao, Jun Shi, and Yongdong Zhang. Causal incremental graph convolution for recommender system retraining. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4718–4728, 2022.
- [3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.
- [4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and 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*, pages 639–648, 2020.
- [5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

- [6] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 75–84, 2020.
- [7] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1269–1278, 2019.
- [8] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 831–840, 2020.
- [9] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunye Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Companion proceedings of the the web conference 2018*, pages 969–976, 2018.
- [10] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegc: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5363–5370, 2020.
- [11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.
- [12] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 519–527, 2020.
- [13] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining*, pages 555–563, 2019.
- [14] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. Multi-modal knowledge graphs for recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1405–1414, 2020.
- [15] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in neural information processing systems*, 33:1513–1524, 2020.

- [16] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.
- [17] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10760–10770, 2020.
- [18] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1717–1725, 2021.
- [19] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.
- [20] Xiaodong Wang, Zhen Liu, Nana Wang, and Wentao Fan. Relational metric learning with dual graph attention networks for social recommendation. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pages 104–117. Springer, 2020.
- [21] Yuening Wang, Yingxue Zhang, and Mark Coates. Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3518–3522, 2021.
- [22] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1791–1800, 2021.
- [23] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2861–2868, 2020.
- [24] Mingshi Yan, Zhiyong Cheng, Chen Gao, Jing Sun, Fan Liu, Fuming Sun, and Haojie Li. Cascading residual graph convolutional network for multi-behavior recommendation. *ACM Transactions on Information Systems*, 42(1):1–26, 2023.
- [25] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

- [26] Wenhui Yu and Zheng Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *International Conference on Machine Learning*, pages 10936–10945. PMLR, 2020.
- [27] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 11–20, 2021.