使用兴趣解耦重训练基于图的推荐器

Yitong Ji 南洋理工大学 新加坡 s190004@e.ntu.edu.sg

Aixin Sun 南洋理工大学 新加坡 axsun@ntu.edu.sg

Jie Zhang 南洋理工大学 新加坡 zhangj@ntu.edu.sg

arXiv:2305.03624v1 [cs.IR] 5 May 2023

摘要

在实际的推荐系统中，新的交互会不断出现。其中一些交互是可预期的，因为它们在很大程度上符合用户的长期偏好。另一些交互则表明用户偏好变化或新物品营销定位中的近期趋势。因此，推荐系统需要定期重新训练或更新，以捕捉新趋势，同时又不遗忘长期偏好。本文提出了一种新颖且通用的重训练框架，称为解耦增量学习（DIL），适用于基于图的推荐器。我们假设，长期偏好已通过根据过往交互学习得到的模型参数，充分体现在现有模型中。新的偏好则可从利用新观察到的交互构建的用户-物品二部图中学习。在 DIL 中，我们设计了一个信息提取模块，从现有模型中提取历史偏好。随后，我们通过解耦模块，以新图中的节点嵌入形式融合历史偏好与新偏好。解耦模块的核心是通过精心设计的损失函数，使历史偏好与新偏好去相关，从而充分捕捉两者。通过在三个基准数据集上的实验，我们展示了 DIL 在捕捉用户-物品交互动态方面的有效性。我们还通过将 DIL 应用于两个基础模型——LightGCN 和 NGCF，证明了其鲁棒性。

CCS 概念

• **信息系统** →**推荐系统**。

关键词

推荐系统，增量学习，流数据

ACM 参考格式：

Yitong Ji、Aixin Sun 和 Jie Zhang。2018。《使用兴趣解耦重新训练基于图的推荐系统》。载于*Make sure to enter the correct conference title from your rights confirmation emai(Confer- ence acronym 'XX)* 会议论文集。ACM，美国纽约州纽约，11 页。https://doi.org/ XXXXXXX.XXXXXXX

允许免费制作本作品全部或部分内容的数字或纸质副本，供个人或课堂使用，但不得出于营利或商业目的制作或分发副本，且副本首页须附有本声明和完整引文。对于本作品中由 ACM 以外的其他方拥有版权的组件，必须尊重其版权。注明出处后允许摘录。若要以其他方式复制、再版、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。请向 permissions@acm.org 申请许可。会议缩写 *'XX，2018*年*6*月*3*日至*5*日，纽约州伍德斯托克 © 2018 计算机协会。ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

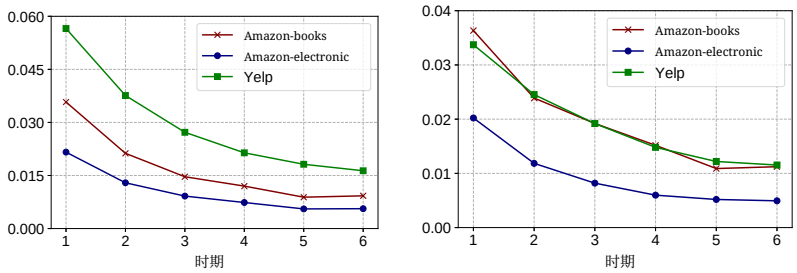
1 引言

在现实场景中，随着用户和物品数量不断增加，面向成长型业务的推荐系统也在持续扩展。此外，用户与物品之间的交互也在不断产生。为了更好地服务用户，推荐模型应不时更新，以跟进最新变化，例如近期交互所反映的新用户、新物品和新趋势。图 1 展示了使用最新交互刷新推荐模型的必要性。我们使用来自 Amazon-books、Amazon-electronic 和 Yelp 的特定数量的数据训练 LightGCN 和 NGCF。[1] 训练完成后，LightGCN 和 NGCF 将保持固定，不进行任何形式的重新训练。随后，使用这些固定模型，在接下来的 6 个时间段内进行推荐；时间段因数据集而异：Amazon-books 为一个月，Amazon-electronic 和 Yelp 则为半年。图 1 显示，推荐性能会随着时间推移而下降。这种下降主要是因为未经重新训练的过时模型无法有效捕捉推荐系统中的最新趋势。因此，这凸显了在观察到新的交互时更新推荐模型的重要性。在配备推荐系统的现实平台中，推荐模型也会定期重新训练 [4, 17]。

不失一般性，我们考虑模型重新训练在某些预定时间点进行的情形。这些时间点可以按固定频率定期安排例如，每隔几个月，也可以根据上次模型更新后新观察到的交互数量来安排。图2展示了一个典型的重新训练过程，其中模型重新训练发生在某些时间窗口 t_0, t_1结束时，直至 t_n。首先，有一个根据 t_0之前观察到的交互训练得到的初始模型 M_{init}。然后M_{init} 会在时间窗口 t_0内投入服务。模型会在 t_0结束时重新训练，更新后的模型 M_0 将于 t_1投入服务。在任一时间窗口结束时，例如， t_n，模型重新训练可以利用三类信息： R_n、 $R_{<n}$和M_{n-1}。其中，R_n 是 t_n期间观察到的新交互集合。$R_{<n}$ 指的是 t_n之前观察到的所有历史交互。M_{n-1} 是最新模型，它在前一个时间窗口 t_{n-1}内更新。其参数代表推荐系统在 t_n 之前捕获的最新知识，这些知识是在 M_{init}以来经过多次更新后形成的。

最简单的重新训练策略是**微调**。以最新模型 M_{n-1} 作为基础模型，微调使用新观测到的交互 R_n对 M_{n-1} 进行重新训练，从而得到 M_n。然而，这种策略

[1]我们分别使用2017年1月至2017年6月、2013年至2014年以及2012年至2014年的数据，在 Amazon-books、Amazon-electronic 和 Yelp 上进行了实验。



(a) LightGCN (b) NGCF

图1：在 Amazon-books、Amazon-electronic 和 Yelp 数据集上，若不对 LightGCN 和 NGCF 进行重新训练，其 6 个时期内的 Recall@20。

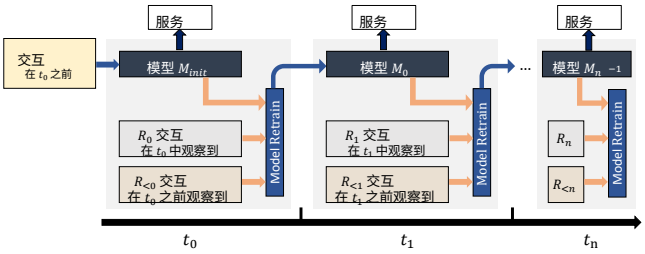


图2：通过模型重新训练进行增量学习。重新训练可使用以下输入：（i）当前模型；（ii）当前时间窗口内新观察到的交互；以及（iii）该时间窗口之前的历史交互。

会带来一个潜在问题——灾难性遗忘 [1, 42]。也就是说，用户的长期偏好可能会被遗忘，因为模型在每次更新时都被迫始终关注当前趋势。在图 1 中可以观察到，虽然不重新训练推荐模型时，Recall@20 会随着时间推移而下降，但即使连续 6 个时期不进行重新训练，也不会降至 0。Recall@20 的值不为零这一事实表明，用户偏好和物品特征可能会随时间保持相对稳定，使得基于历史交互训练的旧模型能够准确预测新的交互。因此，为了更准确地进行推荐，应保留历史交互所表达的历史信息。为了充分利用历史信息和新信息，“极端”策略是**完全重新训练**。完全重新训练会弃用当前模型，并利用所有已发生的交互从头构建推荐系统即 $R_{<n} \cup R_n$。这种解决方案存在可扩展性问题，因为随着交互数量增加，训练成本也会随时间上升。更重要的是，并非所有历史交互都包含与近期情境相关的信息 [12, 26, 35, 39]。

鉴于上述讨论，现有研究 [5, 20, 27, 35, 40]转而采用一种可扩展且有效的再训练策略——**增量学习**。所有增量学习策略都会充分利用当前时间窗口内新观测到的交互。同时，它们采用不同技术，借助采样得到的旧交互、当前模型参数，或两者兼用，来迫使模型“记住”旧信息。例如， [9, 21, 27] 会对一部分历史交互进行采样，将其作为长期信息的代表性样本。SML [40] 和 ASMG [20] 将过去学习到的模型参数（例如，用户嵌入）视为历史信息。

本文提出了一种名为**解耦增量学习（DIL）**的增量学习策略，适用于基于图卷积网络（GCN）的推荐系统。我们关注基于 GCN 的推荐系统，因为它能够建模用户与物品之间的复杂关系，并且已被证明效果显著 [10, 16, 29, 37]。

与其他增量学习策略类似，设计 DIL 面临两项主要挑战：（i）如何提取历史信息；（ii）如何在重新训练后的模型中融合历史信息与新信息 。对于第一个挑战，一些现有方法将先前模型学习到的全部节点表示视为历史信息 [5, 34]。在设计 DIL 时，我们采用了不同的方法。没有必要完整地利用先前的节点表示，因为这些表示所编码的信息并非全部都对应于长期存在的属性。此外，由于图卷积网络会汇集来自不同跳数邻居的信息来构建节点表示，我们将不同跳数邻居的信息分开处理。我们认为，先前图中不同跳数上的邻居节点对历史信息的贡献各不相同。因此，我们在 DIL 中设计了一个信息提取模块（*IEM*），以动态调整从先前模型中提取哪些历史信息以及提取多少。IEM 的核心思想是学习用于调整提取内容及提取量的权重，而不是直接从先前模型中提取某些预先定义的信息。为解决融合新旧信息这一第二项挑战，我们提出了一个解耦模块（*DM*），以降低两类信息之间的相关性。DM 确保模型重新训练时以不同方式处理新旧信息，从而体现它们各自的语义。不过，这两类信息会通过图卷积网络的学习协同发挥作用，以实现准确的推荐。

本研究的贡献主要有三点。首先，所提出的 IEM 提供了一种从先前模型中提取历史信息的新方法，并提出了两种实现方案（第 4.1 节）。其次，我们引入了历史信息与新信息的解耦，以确保两类信息都能得到充分捕获（第 4.2 节）。据我们所知，

我们是首个在推荐模型重新训练中引入长期偏好与短期偏好解耦的研究。第三，为了结合这两个模块，我们也是首个重新设计基于 *GCN* 的推荐器中嵌入层以用于模型重新训练的研究。嵌入表示节点的特征。现在，嵌入同时包含 IEM 提取的历史信息，以及从当前交互图中学习的新信息。通过 GCN 中的消息传播，信息会在相邻节点和间接邻居之间交换，从而带来更准确的推荐（第 4.3 节）。我们在两个基础模型 NGCF [29]和 LightGCN [10],上实现了 DIL，并在三个基准数据集上开展实验：亚马逊图书、亚马逊电子产品和 Yelp。实验结果表明，DIL 在重新训练 GCN 模型时效果显著。此外，DIL 适用于基于 GCN 的模型重新训练，并且在扩展到基于 GCN 的推荐模型时表现良好。

2 相关工作

本节首先简要回顾基于GCN的模型，然后将我们的 DIL 与现有的重新训练策略进行区分。

2.1 基于GCN的推荐

非基于图的推荐模型通常利用观测到的用户-物品交互来学习表示。用户-用户、物品-物品等其他关系无法得到良好建模。NGCF [29] 是较早利用图结构捕获这些额外关系的模型之一。它遵循图卷积网络模型，并设计了邻居聚合层，以递归方式从多跳邻居中提取信息。随后，移除对推荐任务贡献不大的操作后，该模型被改进为更简单的 Light-GCN [10] 。从邻居中提取信息可能会带来较高的计算成本，因此 PinSage [36]采用随机游走和图卷积，在大规模推荐系统中实现高效的表示学习。尽管考虑额外的用户-用户、物品-物品关系具有成效，但不同邻居对推荐所贡献的信息量可能不同。KGAT [28] 在邻居聚合层中采用注意力机制，为不同邻居分配不同权重。不同于 KGAT 在同一跳内区分邻居，NIA-GCN [24] 提出区分不同层中的邻居。除了使用单一的用户-物品交互图外，multi-GCCF [25]还通过使用成对余弦相似度构建用户-用户图和物品-物品图。这些图被用作表示学习的额外信息。 [6] 中的作者提出利用用户-用户社交图作为推荐的额外信息。为进一步增强基于 GCN 的推荐模型的鲁棒性，SGL [31] 在图学习中采用自监督学习。该方法提出了三种图增强技术，用于生成图的多个视图，以进行自监督学习。然而， [37] 中的作者证明，图增强并非那么必要。他们提出通过向节点嵌入添加均匀噪声来生成对比视图。

2.2 推荐中的增量学习

在业界，重新训练推荐系统以捕捉最新变化 [2, 23] 是一种常见做法。除了第 1 节提到的简单微调 和完全保留 之外，研究人员还提出了一些增量学习策略。

2.2.1 经验回放。 这是一种被广泛采用的重新训练策略 [9, 18, 21, 27]。其核心思想是维护一个历史交互蓄水池，以便与新的交互一起进行重新训练。SPMF [27] 将难以准确预测的样本，即困难样本，保存在蓄水池中。其关键动机在于，困难样本需要更多的训练。SSRM [9], 、GAG [21] 和 Ader [18]在流式会话推荐中采用了类似的方法。需要注意的是，尽管 GAG 在训练过程中使用了图卷积操作，但该操作仅用于编码会话，而不作用于用户-物品交互图。因此，这与我们的研究属于不同的主题。Ader [18] 为当前时期处于活跃状态的每个物品存储样例。一个物品的样例数量取决于该物品在前一时期的流行度。除了基于深度学习的方法外，简单的 k近邻方法在处理流式会话时也能取得良好的效果 [15]。

2.2.2 基于模型的重新训练。 这是另一种广泛采用的重新训练策略 [8, 20, 40]。SML [40]和 ASMG [20]假设，利用历史交互学习到的模型参数能够充分捕捉用户偏好和物品特征。作者认为，为避免灾难性遗忘，无需维护回放记忆。相反，他们设计了迁移模块，将历史模型与根据新到达交互学习得到的新模型相结合，以预测未来数据。SML [40]中的迁移模块是卷积网络。在 ASGM [20], 中，迁移模块是在一系列历史模型之上构建的 GRU网络。

*2.2.3 GCN*增量学习。 上述重新训练策略并非为基于GCN的模型设计。具体而言，重新训练过程中没有考虑图结构。GraphSAIL [35]包含三个组件，分别用于保留三类信息：图的局部结构、图的全局结构，以及在前一时期学到的自信息。然而，GraphSAIL将历史最终表示作为一个整体加以利用，没有区分当前时期不同邻居的贡献。CI [5] 和 IGCN [34] 等方法则区别对待不同层的节点表示。CI [5] 通过聚合过去学到的各层旧表示以及从新交互图中学到的新表示，来更新用户和节点表示。与 CI 类似，IGCN [34] 融合旧表示和新表示，以捕捉长期和短期信息。CI 在融合时使用卷积神经网络（CNN），而 IGCN 使用时序CNN网络。与上述基于深度学习的重新训练框架不同，FIRE [33] 是一种用于基于图的推荐的非参数增量学习框架。它使用时序图滤波器修改用户-物品交互矩阵，以用于图卷积。借助该滤波器，在学习过程中历史交互会被赋予较低权重。

我们提出的 DIL 与 CI 和 IGCN 的关系更为密切。CI 和 IGCN 都会逐层融合在前一时期和当前时期学习到的节点表示。也就是说，它们会融合从同一层学习到的新旧表示。在某些方面，不同跳数的邻居会有所区别。然而，我们

请注意，由于每个时间段都会构建一个用户-物品交互图，因此从不同图中学习到的相同层表示可能反映出完全不同的邻居。对于同一个用户，其在不同时间段中的邻居会存在很大差异。简单地融合从相同层学习到的表示，会忽略不同层之间的信息差异。我们的 DIL 在设计上与 CI 和 IGCN 存在本质区别。我们不会直接融合节点表示。相反，我们重新设计了 GCN 的嵌入层，以确保初始特征同时包含旧信息和可学习的新信息。在此基础上，我们进一步引入了一个解耦模块，用于区分旧信息和新信息，而 CI 和 IGCN 并未采用这种做法。

3 预备知识：基于图卷积网络的模型

在介绍所提出的 DIL 之前，我们先简要解释基于 GCN 的推荐模型，该模型将作为 DIL 的基础模型。我们关注利用用户-物品二部图 - G = (V,E) 的推荐。也就是说，图中的节点 V 要么是用户节点，要么是物品节点。边集 E 表示用户与物品之间相应的交互。如果用户与某个物品发生交互，则图中二者之间存在一条边。请注意，用户之间或物品之间不存在直接相连的边。

基于图卷积网络（GCN）的推荐模型通常由以下部分组成：(i) 嵌入层，(ii) L 邻居聚合与信息更新层，以及 (iii) 最终表示层 [7, 32]。

嵌入层。 与其他协同过滤推荐模型类似 [11, 41], , 基于图的推荐系统始于一个嵌入层。通过用户ID和物品ID，可以获得相应的用户嵌入 $e_u \in \mathbb{R}^d$ 和物品嵌入 $e_v \in \mathbb{R}^d$，作为用户和物品的初始特征。这里， d 表示嵌入向量的维度。用户嵌入和物品嵌入均可训练，并在模型训练过程中不断更新。

邻居聚合与信息更新。 给定一个图以及用户/物品节点的特征，基于 GCN 的推荐系统允许节点从其邻居接收信息。例如，用户节点 u 的邻居 $N(u)$ 发送的消息可以如下所示。

$$h_{N(u)} = AGGREGATION\left(h_v, \forall v \in N(u)\right) \tag{1}$$

为了进一步利用图结构，可以扩展邻居聚合，以实现高阶消息传播。也就是说，可以堆叠 L 层邻居聚合，从而递归地从 L 跳邻居中提取信息。在第 ℓ^{th} 步，用户 u 的邻居所接收到的消息为：

$$h_v^0 = e_v \tag{2}$$

$$h_{N(u)}^{(\ell-1)} = AGGREGATION\left(h_v^{(\ell-1)}, \forall v \in N(u)\right) \tag{3}$$

基于从邻居聚合得到的消息，随后对用户节点在每个传播层的表示进行优化：

$$h_u^\ell = UPDATE\left(\left\{h_u^{\ell-1}, h_{N(u)}^{\ell-1}\right\}\right) \tag{4}$$

在上述描述中，我们以用户节点为例。相同的聚合和更新过程也适用于物品节点。通过用户-物品二部图，基于 GCN 的推荐系统不仅能够对用户-物品关系进行建模，还能够通过多跳聚合对间接的用户-用户关系和物品-物品关系进行建模。

最终表示层。 通过堆叠 L 消息传播层，我们可以获得 L 节点表示，每一层对应一个。推荐系统可以决定如何为用户/物品生成最终表示。例如，PinSage [36] 和 STAR-GCN [38]选择最后一层表示作为最终表示h_u^{final}。其他模型 [10, 29, 30] 则通过拼接和均值池化等技术，将所有 L 节点表示与初始嵌入进行聚合。

$$h_u^{final} = AGGREGATION\left(h_u^\ell, \forall \ell \in [0, \ldots, L]\right) \tag{5}$$

用户 u 和物品 v 之间的预测偏好是通过两个最终表示 h_u^{final}和 h_v^{final}计算得出的。

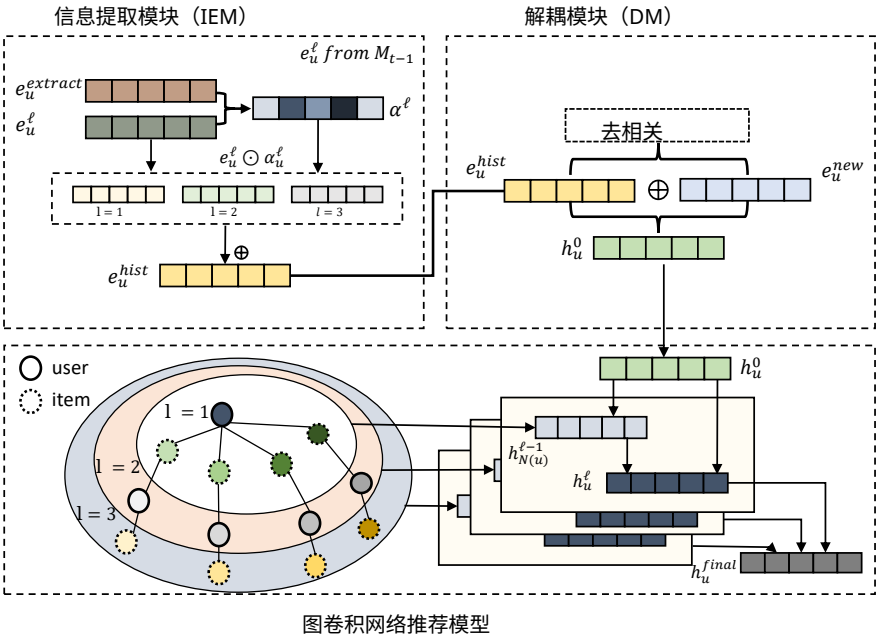
4 提出的模型

我们遵循图2所示的模型重新训练过程。回想一下，在时间 t_n 结束时，模型重新训练包含三部分信息：(i) R_n 时间窗口 t_n内新观察到的用户-物品交互，(ii) $R_{<n}$ 在 t_n之前发生的历史交互，以及 (iii) M_{n-1} 已部署用于服务的最新模型。在下文讨论中，相对于重新训练后的模型 M_n， M_{n-1} 也称为“先前模型”。

由于新近观测到的交互最能反映当前趋势，并且与近期的推荐更加相关，因此我们使用 R_n构建一个新的用户-物品二部图。为了融入长期偏好，我们从 M_{n-1}中提取历史信息，然后将这些信息融合到新模型 M_n中。为避免混淆，我们在本节介绍所提出的 DIL 时，假设所有用户和物品在 t_{n-1}期间均发生过交互，这一假设适用于大多数用户和物品。在本节末尾，我们将详细说明模型如何将其他类型的用户/物品作为特殊情况进行处理。此外，我们还将讨论模型在部署到服务 t_{n+1}一段时间后，如何处理未见过的用户/物品。

架构概述。 在用户-物品二部图中，同一重新训练过程同时适用于用户节点和物品节点。因此，为避免重复，我们只介绍应用于用户节点的操作。由于本文讨论的只有用户节点，为便于表述，我们直接用 e 表示用户嵌入。需要区分二者时，我们分别用 e_u 和 e_i 表示用户嵌入和物品嵌入。

所提出的解耦增量学习（DIL）框架的整体模型架构如图3所示。首先，信息提取模块（IEM）从在 M_{t-1} 中学习到的表示中提取历史信息 e^{hist}。它表示长期用户偏好。随后，解耦模块（DM）将 e^{hist} 与 e^{new} 融合，形成新构建的二部图中用户节点的初始嵌入 h^0。可学习的 e^{new} 表示需要通过模型训练从 R_n 中新的用户-物品交互中学习的动态用户偏好。它可以通过随机方式初始化，也可以参考先前模型进行初始化。在我们的实现中，我们使用在 M_{t-1} 中学习到的最终表示 h^{final} 来初始化 e^{new}。在此，我们认为 h^{final} 能够很好地反映近期的推荐上下文，而这与当前时间段密切相关。然而，它并不代表长期信息，因此 e^{hist} 在我们的模型设计中仍然至关重要。为了确保 DIL 能够区分历史信息和新信息。



图卷积网络推荐模型

图3：以用户节点为例的 DIL 模型架构。DIL 由两个组件组成：信息提取模块和解耦模块。从 DIL 中学到的表示可用作任何基于图卷积网络的推荐模型的输入。

信息，DM 对 e^{hist} 和 e^{new}进行去相关处理，如图3所示。随后，将 h^0 用作用户 u 的初始特征，并输入基础图卷积网络推荐模型进行训练。通过这种方式，历史信息得以保留，同时与新信息区分开来。作为我们的贡献之一，我们首次重新设计了基于图卷积网络的推荐系统中的嵌入层，以用于模型重新训练。

4.1 信息提取模块(IEM)

如前所述，我们认为，在 M_{t-1} 中学习到的最终表示能够体现近期的推荐上下文，但无法很好地表示用户的长期偏好以及物品的持久特征。举例来说，假设某位用户从羽毛球转向了网球，但总体而言，他或她仍然是一位体育爱好者。尽管近期的一些交互更多与羽毛球有关，但如果我们考察该用户在用户-物品图中的所有交互，以及多跳邻居的相关信息，就能很好地体现其体育爱好者的特征。同样的情况也适用于将手机升级为同一品牌最新款的用户。过去与各种手机配件的交互，可能充分反映出用户对该品牌的忠诚度。因此，我们认为，用户的长期偏好和物品的持久特征并不是先前模型中简单的表示。相反，表示中的部分信息才反映了随时间推移而形成的长期偏好和持久特征。此外，长期信息分布在不同的层中，以从多跳邻居聚合而来的信息形式呈现。

基于上述理解，我们从 M_{n-1} 多个层的表示中提取历史信息。请注意， M_{n-1} 不限于交互集合 R_{n-1}，它还保留了从先前模型中提取的历史信息，而先前模型又通过回溯从 M_{init} 中提取了信息。我们的设计旨在捕捉随时间推移在各个模型间缓慢更新的历史信息。

为了确定要提取哪些信息，以及从每一层提取多少信息，我们设计了两个参数。我们将第一个参数命名为提取器嵌入，它决定从每个节点提取哪些信息。第二个参数权重向量则决定从每一层提取的信息量。在本研究中，我们提出了提取器嵌入和权重向量的两种实现设计。

设计 1。 我们为每个节点初始化一个可学习的提取器嵌入 $e^{extract}$ ，以确定要提取哪些历史特征。然后，对于每一层，我们计算一个权重参数，以确定该层中特征的权重。令 e^{ℓ} 表示 M_{n-1}中第 ℓ 层的输出（或聚合信息）。回顾一下， e^{ℓ}反映了在时间 $n-1$、 $\ell \in [0,1,\ldots,L]$从 ℓ 跳邻居收集的信息。第 ℓ 层的权重计算如下：

$$\alpha^{\ell} = \sigma\left(e^{extract} \odot e^{\ell} + pos^{\ell}\right); \quad \ell \in [0,1,\ldots,L] \tag{6}$$

这里， σ是一个 Sigmoid 函数。它控制从每个维度提取的特征，使其取值范围为 0 到 1。 $\odot$表示逐元素乘积。此外， pos^{ℓ}是一个位置嵌入，用于进一步

控制从上一周期的 ℓ 跳邻居中提取的信息量。利用为每一层计算的权重 α^ℓ，我们通过一个 $AGGREGATION(\cdot)$ 函数聚合提取的信息，该函数可以是求和、均值池化或拼接。

$$e^{hist} = AGGREGATION\left(\alpha^\ell \odot e^\ell, \forall \ell \in [0, 1, \dots L]\right) \tag{7}$$

设计 2。与设计 1 类似，我们使用 $e^{extract}$ 来确定每个节点 $\ell \in [0, \dots, L]$ 在 e^ℓ 中的长期属性。我们不使用权重向量来确定要提取的信息量，而是为每一层初始化一个可学习矩阵 $W^\ell \in \mathbb{R}^{d \times d}$。$W^\ell$ 直接从 ℓ 跳邻居中提取有用信息。

$$e^\ell_{hist} = W^\ell(e^{extract} \odot e^\ell) \tag{8}$$

然后，我们通过聚合函数聚合从每一层提取的信息。

$$e^{hist} = AGGREGATION(e^\ell_{hist}, \forall \in [0, \dots, L]) \tag{9}$$

我们指出，信息提取还有其他可行的设计方案。可以用一种适用于特定领域的新提取操作替换所提出的设计，例如，时尚推荐和视频推荐。由于 IEM 仅负责提取 e^{hist}，其设计不会影响 DIL 用于增量学习的整体架构。

4.2 解耦模块（DM）

现在，我们已经通过 IEM 为每个节点提取了历史信息 e^{hist}。e^{hist} 是从历史模型中提取的信息。为了利用交互集合 R_n 捕捉新的推荐上下文，我们为每个节点初始化一个嵌入 e^{new}。为融合新旧信息，我们将两个嵌入相加，得到：

$$h^0 = e^{hist} + e^{new} \tag{10}$$

然后，h^0 将作为基于 GCN 的推荐模型的初始特征，例如 LightGCN 和 NGCF。

为确保模型重新训练能够区分 e^{hist} 和 e^{new}，我们引入了解耦模块。解耦的核心思想是确保基础推荐模型在历史知识 e^{hist} 的基础上，尽可能多地学习额外信息。额外信息是仅通过新交互体现的新信息。

为实现解耦，我们对表示进行去相关。受 [30], 启发，我们采用距离相关 $CORR$ 作为损失项，计算方式如下：

$$L_{CORR}\left(e^{hist}, e^{new}\right) = \frac{dCov\left(e^{hist}, e^{new}\right)}{\sqrt{dVar\left(e^{hist}\right) \cdot dVar\left(e^{new}\right)}} \tag{11}$$

在上述公式中，$dCov(\cdot)$ 表示两个向量之间的距离协方差，而 $dVar(\cdot)$ 表示一个向量的距离方差。请注意，L_{CORR} 值越低，输入向量之间的依赖性越弱，并且 $L_{CORR} \in [0, 1]$。通过去相关，我们希望在 e^{hist} 和 e^{new} 中学习到的信息彼此独立，但二者都能在消息传播过程中发挥作用，最终实现更准确的推荐。

e^{hist} 和 e^{new} 共同构成节点的初始嵌入。在训练过程中，新的交互被用作真实数据实例，以监督 e^{hist} 和 e^{new} 的学习。然而，由于 e^{hist} 表示存在于所有时间段中的长期信息，因此需要历史交互为 e^{hist} 提供额外的监督信号。因此，我们提出利用历史交互来监督 e^{hist} 的学习，具体如下。

4.3 模型优化

我们将 e^{hist} 和 e^{new} 融合，用于初始化节点表示 h^0。因此，用户表示为 h^0_u，物品表示为 h^0_i。h^0_u 和 h^0_i 均作为基础图卷积网络推荐模型的输入。按照第 3 节所述，经过 L 层消息传播后，我们得到最终的节点表示 h^{final}_u 和 h^{final}_i，分别对应用户和物品。二者的点积用于预测用户对某个物品的偏好。

$$\hat{y}_{ui} = h^{final}_u \cdot h^{final}_i \tag{12}$$

我们采用成对 BPR 排序损失 [22] 作为目标函数。其核心思想是：我们希望优化模型参数，使用户在其点击/评分/购买过的物品上的预测偏好，高于在与该用户没有交互的物品上的预测偏好。我们认为，点击/评分/购买行为体现了用户对某件物品的兴趣。在增量学习场景中，我们会随着时间 t_n 利用新观测到的交互更新模型参数。具体而言，我们优化 BPR 损失 L_{new}：

$$L_{new} = \sum_{(u,i,j) \in O} -\ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right) \tag{13}$$

物品 i 是当前时期 t_n 内与用户 u 发生过交互的物品，而物品 j 则是从截至 t_n 尚未与用户 u 发生交互的物品中采样得到的负物品。

如前文所述，我们进一步监督 e^{hist}_u，以确保其能够充分反映历史信息。为此，我们使用从 $R_{<n}$ 中采样的历史交互对其进行监督。因此，我们将公式 13 中的损失项修订为：

$$L = \sum_{(u,i,k,j) \in O} -\ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right) - \ln \sigma\left(e^{hist}_u \cdot e^{hist}_k - e^{hist}_u \cdot e^{hist}_j\right) \tag{14}$$

在公式 14 中，物品 k 是从前一时期与用户 u 发生过交互的物品中采样得到的。物品 j 是采样得到的负样本。需要注意的是，由于不存在与用户 u 的交互，因此 j 可以同时作为新交互和历史交互训练中的负样本。额外的监督进一步确保解耦表示能够分离新信息和历史信息的语义。

加上解耦项后，我们的最终目标函数如下，其中 λ 是一个超参数，用于控制解耦项的权重。

$$L_{final} = L + \lambda L_{CORR} \tag{15}$$

4.4 非活跃用户和推理

在增量学习场景中，用户可分为三类。“活跃用户”在前一时期 t_{n-1} 和当前时期 t_n 均有交互。此外，还有 “ 非活跃用户 ”，他们在 t_{n-1} 没有交互，但在 t_{n-1}之前曾经活跃。最后，还有“新用户”，他们在 t_n首次发生交互。

在重新训练过程中，我们可以从 M_{n-1} 获取活跃用户的信息。对于新用户，他们的特征会随机初始化，并在当前时期进行更新。对于非活跃用户，他们在 M_{n-1}中没有历史信息。为他们初始化 e^{hist} 有三种方式。最简单的方法是将其视为新用户。第二种方法是在用户上次活跃时从模型中提取 e^{hist} 。这需要保留所有历史模型参数。最后一种方法是从一开始就将非活跃用户纳入模型重新训练 即，t_0。尽管他们在非活跃时期构建的图中没有边，但其参数会随着模型重新训练而更新。在我们的实现中，我们采用了最后一种方法。

我们注意到，活跃用户是一个时间段 t_n内最大的用户群体，因为用户往往会定期访问平台。以我们在实验中使用的数据集为例，一个时间段内活跃用户的平均占比分别为：亚马逊图书 81.1% 、亚马逊电子产品 64.3% 和 Yelp 69.7%。

重新训练后，新模型 M_n 将在时间 t_{n+1}部署并提供服务。在时间 t_{n+1}可能会出现“未见用户”，他们首次产生交互。未见用户的特征将被随机初始化，推荐模型将使用随机初始化的向量进行推荐。实际上，如果平台能够通过注册等方式获取用户的某些属性 例如，则可以根据已知属性初始化其特征。物品同样如此。

5 实验设置

我们在三个常用数据集——亚马逊图书 [19],、亚马逊电子产品 [19] 和 Yelp[2]——上开展实验。由于 DIL 专为基于图卷积网络的推荐模型设计，我们选择了两个基于图的推荐模型作为基础模型：LightGCN [10] 和 NGCF [29]。通过实验，我们旨在验证所提出的增量学习框架的有效性和稳健性。

5.1 增量学习场景

我们遵循推荐系统中增量学习的常见设置 [20, 35, 40]。如图2所示，对于每个数据集，我们设置一个预热期，以初步了解用户和物品。基准推荐模型使用预热期内发生的所有交互进行批量训练。随后，对于这三个数据集，我们将预热期之后的交互划分为 6 个时段。第一组 R_0 用于在 t_0 进行再训练，然后更新后的模型预测 R_1。在 R_1 上的预测准确率用作超参数调优的验证指标。剩余的 4 组交互 R_2 至 R_5 作为测试集。对于每个测试集，前 10% 的交互用于提前停止，我们计算在剩余 90% 交互上的推荐准确率。4 个测试集上的平均推荐准确率即为该再训练策略的准确率。

数据集。 数据集的统计数据如表1所示。所有三个数据集均经过10核过滤预处理，即， 所有用户和物品至少有10次交互。

对于亚马逊图书，我们使用了2017年的交互数据。将2017年前6个月设为预热期。对于剩余的6个月，每个月的交互数据构成一组用于重新训练的数据。对于亚马逊电子产品，我们选取了2013年至2017年的交互数据进行实验。在这5年期间，将前两年设为预热期。随后，将剩余3年的交互数据划分为6组用于重新训练，每半年的数据实例构成一组。我们选取了2012年至2017年的*Yelp* 交互数据进行实验。预热期设定为2012年1月1日至2014年12月31日。随后，将2015年至2017年的交互数据划分为6组用于重新训练。每组包含半年内发生的交互数据。如表1所示，所有数据集在预热期内的交互数量大致相当。

评估指标。 我们将任务定义为一个基于隐式反馈的 Top-N 推荐任务。随后，我们通过 Top-20 推荐的准确率来评估所提出的框架和基线方法。也就是说，在时间 t_n，我们从截至 t_n已浏览过的所有物品 中，为每位用户推荐 20 件物品。[3] 全物品排序的目的在于避免推荐结果中可能存在的偏差 [3] [14]。

对于评估指标，我们报告 Recall@20 和 NDCG@20。[4] Recall 表示测试实例中被正确推荐的实例所占的百分比。NDCG（归一化折损累计增益）进一步考虑了被正确推荐物品的排名位置。需要注意的是，Recall@20 和 NDCG@20 的绝对值看起来较低，原因有二。其一，我们对所有物品进行排序，在大量候选物品中推荐出正确物品十分困难。其二，在我们的设置中，训练集/测试集的数据划分严格遵循时间线。新时间窗口中的推荐准确率同时受到未见用户和未见物品的影响。

5.2 基线

我们将 DIL 与五种基线方法进行比较。其中，微调和完全重新训练是两种极端的重新训练策略。SPMF 是一种经验回放策略；GraphSAIL 和 CI 则专为基于图卷积网络的推荐系统设计。GraphSAIL 是一种基于蒸馏的增量学习框架，而 CI 是一种基于模型的增量学习框架。为此，我们在实验中涵盖了不同类型的增量学习技术。

微调 仅使用当前时期观察到的交互重新训练基础推荐模型。使用最新的交互微调基础模型时，我们用前一时期学到的模型参数初始化基础模型，以此记忆旧信息。

完全重新训练 会使用截至重新训练时观察到的所有交互，重新训练基础推荐模型。

[2]https://www.yelp.com/dataset

[3]对于时间 t_n新观察到的物品，没有它们的历史信息。我们随机初始化其嵌入。[4] 对于前 5 和前 10 项推荐的 Recall 和 NDCG，也有类似的观察结果。因此，为清晰起见，我们仅报告 Recall@20 和 NDCG@20。

表1：实验中使用的数据集的统计数据。

数据集	时间段	#用户	#物品	#预热交互	#重新训练交互
亚马逊图书	2017年1月1日至2017年12月31日	34，195	25，798	458，145	483，293
亚马逊电子产品	2013年1月1日—2017年12月31日	91，592	34,401	448，394	1,028,395
Yelp	2012年1月1日—2017年12月31日	42，124	29，099	453，470	649，627

SPMF 是一种非基于图的增量学习策略，会在重新训练过程中重放记忆。过去未获得良好推荐的交互被视为需要额外训练的样本，因此会保存在记忆中以供重放。我们将记忆大小调整为 {20000,40000, 60000}。

GraphSAIL 是一种基于图的重新训练策略。它设计了三种蒸馏损失，以保留图的局部结构和全局结构，并保留过去的自信息。我们将三个蒸馏系数设置为 {0.000001，0.001，1}。此外，GraphSAIL 通过对基于 GCN 的推荐模型学习到的节点表示进行 k均值聚类，来构建全局图结构。我们将聚类数量 k 设置为 {5，10，20}。

CI 也是一种基于图的重新训练策略。它将过去学到的旧节点表示与根据新观察到的交互学到的新表示融合起来。此外，即使没有观察到这些节点的交互，它也会更新距离 k最近的非活跃节点的表示。这是通过将它们与活跃节点相连来实现的。我们将 k近邻中的邻居数量设置为 {10，30，50}。我们将非活跃节点更新的系数设置为 {0.25，0.5，0.75，1}。

对于所有模型，我们进一步在搜索空间 $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ 中调整学习率，并从 {32，64，128} 中选择嵌入维度。对于正则化系数，我们在 $\{10^{-6}, 10^{-4}, 10^{-2}\}$ 中进行搜索。

5.3 DIL 实现细节

通过对候选超参数进行网格搜索，我们发现，在 IEM模块中，**设计 1** 在亚马逊图书和 Yelp 数据集上，以及在亚马逊电子产品数据集上搭配 LightGCN 时，推荐性能更佳。对于亚马逊电子产品数据集搭配 NGCF 的情况，**设计 2** 是更好的选择。在研究的其余部分中，我们报告各数据集采用最优设计时的实验结果。我们使用 Adam优化器训练 DIL，在亚马逊图书和 Yelp 数据集上的初始学习率为 0.0001，在亚马逊电子产品数据集上为 0.00001。亚马逊图书和 Yelp 数据集使用的正则化系数为 0.0001，而亚马逊电子产品数据集为 0.000001。我们将所有三个数据集的维度大小设为 128。解耦项的权重，即即 λ 公式 15 中的权重，在不同数据集和模型之间有所不同。其取值从 $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ 中调优。

对于基础模型，我们将最终表示设定为对所有层进行聚合。所使用的聚合函数为均值池化。尽管 NGCF 中原始的聚合函数是将所有节点表示进行拼接，但作者 [29] 提到，聚合函数是可以替换的，而我们发现，均值池化能够带来更好的推荐性能。

6 实验结果

我们使用两个基础模型，在三个数据集上比较 DIL 与五种基线的推荐准确率。然后，我们测试 DIL 中两个主要组成部分的有效性。

6.1 有效性与鲁棒性

如表 2 所示，在所有数据集上，DIL 在 Recall 和 NDCG 两项指标上均优于所有基线。以 LightGCN 为基础模型的版本在亚马逊图书和 Yelp 上大幅优于其 NGCF 版本。不过，在亚马逊电子产品上，采用 LightGCN 的 DIL 表现略逊于其 NGCF 版本。GraphSAIL 和 SPMF 是表现最佳的基线方法之一。尤其是，采用 LightGCN 的 GraphSAIL 在亚马逊图书和亚马逊电子产品上均取得了第二好的结果。

我们进一步以 Amazon 图书上的 LightGCN 为例，绘制图 4，展示 4 个测试时间段的 Recall@20 分数。一个有趣的观察是，各基线方法的性能排名顺序会随时间段而变化。在 [13], 的其他数据集上也观察到了类似现象。这说明需要持续监控评估结果，以避免特定时间段可能带来的偏差。在本研究中，我们对各时间段的评估分数取平均，仍然发现 DIL 的性能更优。这清楚地表明，DIL 是一种稳健且有效的重新训练框架。

在所有基线中，CI 被认为与我们的方法最相似，因为 CI 也融合了来自 M_{n-1} 的历史知识与当前时期学到的新信息。关键区别在于提取和融合新旧信息的方式。也就是说，CIF 采用“后学习融合”，聚合 GCN 在两个不同时期输出的表示。它提取历史信息的方式很直接。与 CI 不同，DIL 采用“前学习融合”，将新旧信息融合后作为 GCN 的输入（节点的初始特征）。通过解耦，DIL 能够很好地区分历史信息和新信息。请注意，在重新训练期间，DIL 会通过改进信息提取模块来调整所提取的历史信息。实验表明，DIL 的表现远胜 CI。

在大多数设置下，微调的表现优于完全重新训练，只有在 Yelp 上使用 NGCF 的版本例外。这清楚地表明，较新的交互与当前的推荐场景更相关。尽管如此，在大多数情况下，微调的结果并不十分有竞争力（在亚马逊图书上使用 NGCF 进行微调的情况除外）。这说明，完全遗忘历史信息也不是理想的选择。SPMF 和 DIL 都会在训练过程中从 $R_{<n}$ 中采样历史交互。在我们的实验中，我们发现，SPMF 所需的最佳交互数量并非超参数候选项中的最大值。这也意味着，更多的历史交互不一定能带来更好的推荐。

表 2：在 LightGCN 和 NGCF 上采用不同重训练策略后，亚马逊图书、亚马逊电子产品和 Yelp 的 Recall@20 和 NDCG@20 结果。最佳结果以粗体显示，次佳结果加下划线。

数据集		亚马逊图书		亚马逊电子产品		Yelp	
基础模型	基线	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	微调	0.0368	0.0140	0.0256	0.0099	0.0591	0.0237
	完全重新训练	0.0332	0.0116	0.0210	0.0080	0.0560	0.0217
	SPMF	0.0459	0.0169	0.0262	0.0101	0.0627	0.0249
	GraphSAIL	0.0490	0.0180	0.0265	0.0101	0.0598	0.0240
	CI	0.0390	0.0159	0.0192	0.0071	0.0521	0.0205
	DIL	**0.0690**	**0.0263**	**0.0271**	**0.0102**	**0.0649**	**0.0259**
NGCF	微调	0.0618	0.0221	0.0206	0.0078	0.0355	0.0136
	完全重新训练	0.0285	0.0106	0.0187	0.0071	0.0362	0.0137
	SPMF	0.0582	0.0224	0.0265	0.0102	0.0445	0.0170
	GraphSAIL	0.0590	0.0204	0.0240	0.0094	0.0425	0.0165
	CI	0.0571	0.0200	0.0236	0.0091	0.0442	0.0176
	DIL	**0.0635**	**0.0233**	**0.0274**	**0.0105**	**0.0478**	**0.0186**

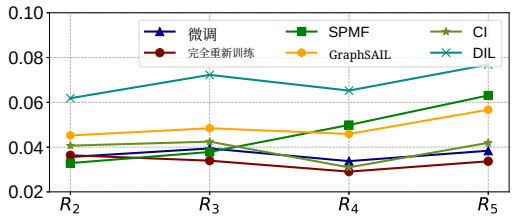


图4：在亚马逊图书数据集上，通过重新训练 LightGCN，展示其在 4 个测试时期内的 Recall@20。

最后，亚马逊电子产品上的绝对值远低于其他数据集。原因之一是该数据集较为稀疏。其用户群规模是其他数据集的两到三倍，但交互数量仅多出约 50%。另一个原因是，与其他类别的物品相比，电子产品中的物品可能种类更多、生命周期更短，即书籍和餐厅，因此这是一个颇具挑战性的数据集。

6.2 消融实验

我们进一步开展实验，以验证 DIL 中各个组件的有效性。具体而言，我们分别探索了信息提取模块和解耦模块的不同潜在设计方案。

信息提取模块 (IEM) 旨在区分不同的邻居，并区别对待 L 层的输出。我们进行实验以验证区分邻居的必要性。具体而言，我们将 DIL 修改为 DIL (mean)、DIL (first) 和 DIL (last)。在 DIL (mean) 中，我们从所有 L 节点表示的平均值以及节点的初始嵌入中提取信息。对于 DIL (first) 和 DIL (last)，我们分别从初始嵌入和最后一层表示中提取历史信息。

参见表 3 的结果，不同跳数的邻居提供不同的信息，因此导致 Recall@20 不一致。以及 NDCG@20。我们还观察到，DIL 变体的排名顺序会因数据集和模型而异。例如，在 LightGCN 上实例化时，相较于使用初始嵌入，GCN 的节点平均表示对亚马逊图书上的推荐性能贡献更大。而对于 Yelp，最重要的因素是初始嵌入。这表明，应根据研究领域为邻居赋予不同的权重。

我们提出的 DIL 在 IEM 中区分来自不同跳数的邻居。根据表 3，我们的设计在大多数情况下都有效，因为 DIL 展现出最佳的推荐准确率。由于 IEM 是 DIL 中的独立模块，因此不必严格遵循所提出的设计。可以重新审视并重新设计 IEM，以处理不同场景下的推荐。替换 IEM 不会影响 DIL 用于重新训练基于 GCN 的推荐模型的整体架构。

解耦模块 (DM) 对长期信息和新信息进行去相关。去相关过程由采样的历史交互监督。我们通过移除去相关项和 DIL 中的监督，开展实验以探究推荐准确率的变化。结果见表4。

与 DIL 相比，DIL-decorr 和 DIL-supervision 的 Recall@20 和 NDCG@20 值有所下降，这表明去相关项和监督有助于提升推荐效果。在亚马逊图书数据集上，我们观察到 DIL-decorr-supervision 的表现优于 DIL-supervison。这说明缺少监督时，去相关可能会失效。一种可能的原因是，模型最终得到了一种平凡但无意义的解。

6.3 讨论

我们通过在实验中将 DIL 与其他增量学习框架进行比较，展示了 DIL 的有效性。需要指出的是，我们的实验旨在公平评估所有重新训练策略，并采用在固定时间间隔进行的简单重新训练计划。在实际应用中，重新训练计划也是一个会显著影响推荐准确率的超参数。我们还注意到，DIL 忽略了

表3：用于验证信息提取模块中各项设计有效性的消融实验。

数据集		亚马逊图书		亚马逊电子产品		Yelp	
基础模型	基线	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	DIL（均值）	0.0677	0.0258	0.0285	**0.0111**	0.0640	0.0256
	DIL（首次）	0.0613	0.0234	0.0269	0.0099	**0.0658**	**0.0262**
	DIL（末次）	0.0663	0.0197	**0.0287**	0.0085	0.0630	0.0187
	DIL	**0.0690**	**0.0263**	0.0271	0.0102	0.0649	0.0259
NGCF	DIL（平均）	0.0609	0.0229	0.0261	0.0101	0.0435	0.0165
	DIL（首次）	**0.0635**	0.0230	0.0183	0.0071	0.0458	0.0176
	DIL（最后一次）	0.0582	0.0221	**0.0283**	0.0084	0.0413	0.0160
	DIL	**0.0635**	**0.0233**	0.0274	**0.0105**	**0.0478**	**0.0186**

表4：用于验证解耦模块中各项设计有效性的消融实验。

数据集		亚马逊图书		亚马逊电子产品		Yelp	
基础模型	基线	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	DIL-去相关	0.0617	0.0231	0.0255	0.0095	0.0643	0.0258
	DIL-监督	0.0514	0.0192	0.0249	0.0092	0.0640	0.0257
	DIL-去相关监督	0.0575	0.0208	0.0251	0.0093	0.0640	0.0251
	DIL	**0.0690**	**0.0263**	**0.0271**	**0.0102**	**0.0649**	**0.0259**
NGCF	DIL-去相关	0.0620	0.0229	0.0255	0.0098	0.0468	0.0181
	DIL-监督	0.0502	0.0182	0.0245	0.0095	0.0461	0.0181
	DIL-去相关监督	0.0516	0.0188	0.0262	0.0101	0.0446	0.0172
	DIL	**0.0635**	**0.0233**	**0.0274**	**0.0105**	**0.0478**	**0.0186**

e^{hist} 在时间 t_{n-1}之前其他时间段中学习到的内容，在时间窗口 t_n内进行更新时，例如，忽略在时间t_{n-2}中学习到的 e^{hist}_{n-2}。根据定义， e^{hist}_{n-2} 既包含来自 R_{n-3} 的信息，也包含来自 $R_{<n-3}$的信息，而 e^{hist}_{n-1} 则包含来自$R_{n-2} \cup R_{<n-2}$的信息。可以推断， e^{hist}_{n-1} 和 e^{hist}_{n-2} 之间存在交集，尤其是在存在影响用户跨时间段行为的长期属性时。DIL并未直接对 e^{hist} 的连续性进行建模，以记忆长期偏好，而是通过解耦模块中的监督操作来维持连续性。由于不同时期的 e^{hist} 似乎处于不同的潜在空间中，因此使用 e^{hist} 直接对连续性进行建模并非易事。此外，连续性仅存在于 e^{hist}的部分特征中。我们将把这一探索留作未来工作。

7 结论

本文提出了一种新颖且通用的增量学习策略 DIL，适用于基于 GCN 的推荐系统。DIL 的核心思想是预学习融合。我们从前一时期训练得到的模型中提取长期历史信息。然后，将历史信息与可学习的新信息融合。融合后得到的嵌入被视为基于 GCN 的推荐模型中节点的初始特征。为确保可学习的新信息能够通过训练反映推荐系统中的新趋势，我们采用解耦来确保新信息独立于历史信息。通过实例化 DIL 在两种图卷积网络模型，即 LightGCN 和 NGCF 上，我们展示了 DIL 在重新训练中是有效的，并且能够为推荐学习短期和长期信息。“预学习融合”的理念为如何提取和融合历史信息以进行模型重新训练提供了另一种思路。

参考文献

[1] Kian Ahrabian、Yishi Xu、Yingxue Zhang、Jiapeng Wu、Yuening Wang 和 Mark Coates。2021年。《面向基于图的推荐系统增量学习的结构感知经验回放》。载于*CIKM '21*：第*30*届*ACM*国际信息与知识管理会议。ACM，2832–2836。[2] Marie Al-Ghossein、Talel Abdessalem 和 Anthony Barré。2022年。《基于流的推荐系统综述》。*ACM* 计算调查54卷，5期（2022年），104:1–104:36。[3] Alejandro Bellogín、Pablo Castells 和 Iván Cantador。2011年。《面向精确率的推荐系统评估：算法比较》。载于*RecSys*。ACM，333–336。[4] Heng-Tze Cheng、Levent Koc、Jeremiah Harmsen、Tal Shaked、Tushar Chandra、Hrishi Aradhye、Glen Anderson、Greg Corrado、Wei Chai、Mustafa Ispir、Rohan Anil、Zakaria Haque、Lichan Hong、Vihan Jain、Xiaobing Liu 和 Hemal Shah。2016年。《面向推荐系统的宽深度学习》。载于第一届推荐系统深度学习研讨会论文集（*DLRS 2016*）。计算机协会，波士顿，马萨诸塞州，美国，7–10。https://doi.org/10.1145/ 2988450.2988454[5] Sihao Ding、Fuli Feng、Xiangnan He、Yong Liao、Jun Shi 和 Yongdong Zhang。2022年。用于推荐系统重新训练的因果增量图卷积。*IEEE*神经网络与学习系统汇刊（2022年）。[6] Wenqi Fan、Yao Ma、Qing Li、Yuan He、Yihong Eric Zhao、Jiliang Tang 和 Dawei Yin。2019年。《用于社交推荐的图神经网络》。载于万维网大会，*WWW*。ACM，417–426。[7] Chen Gao、Yu Zheng、Nian Li、Yinfeng Li、Yingrong Qin、Jinghua Piao、Yuhan Quan、Jianxin Chang、Depeng Jin、Xiangnan He 和 Yong Li。2023年。《推荐系统图神经网络综述：挑战、方法与方向》。1卷，1期（2023年）。https://doi.org/10.1145/3568022

[8] Renchu Guan、Haoyu Pang、Fausto Giunchiglia、Ximing Li、Xuefeng Yang 和 Xiaoyue Feng。2022年。《基于元学习、可部署且可持续的推荐系统及快速用户增量更新》。载于*SIGIR '22*：第*45*届国际*ACM SIGIR*信息检索研究与发展会议。ACM，1423–1433。[9] Lei Guo、Hongzhi Yin、Qinyong Wang、Tong Chen、Alexander Zhou 和 Nguyen Quoc Viet Hung。2019年。《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（*KDD*）。ACM，1569–1577。[10] Xiangnan He、Kuan Deng、Xiang Wang、Yan Li、Yong-Dong Zhang 和 Meng Wang。2020年。《LightGCN：简化并增强用于推荐的图卷积网络》。载于第*43*届国际*ACM SIGIR*信息检索研究与发展会议论文集。ACM，639–648。[11] Xiangnan He、Lizi Liao、Hanwang Zhang、Liqiang Nie、Xia Hu 和 Tat-Seng Chua。2017年。《神经协同过滤》。载于第*26*届万维网国际会议论文集（*WWW*）*2017*。ACM，173–182。[12] Yitong Ji、Aixin Sun、Jie Zhang 和 Chenliang Li。2022年。《忠诚用户会获得更好的推荐吗？从时间视角理解推荐准确性》。载于*ICTIR '22*：*2022*年*ACM SIGIR*信息检索理论国际会议。ACM，92–97。[13] Sergey Kolesnikov 和 Mikhail Andronov。2022年。《CVTT：跨时间交叉验证》。*CoRR* abs/2205.05393（2022年）。[14] Walid Krichene 和 Steffen Rendle。2022年。《关于物品推荐的采样指标》。*ACM* 通讯65卷，7期（2022年），75–83。[15] Sara Latifi 和 Dietmar Jannach。2022年。《流式会话推荐：图神经网络与邻域相遇》。载于*RecSys '22*：第十六届*ACM*推荐系统会议，美国华盛顿州西雅图，*2022*年*9*月*18*日至*23*日。ACM，420–426。[16] Zihan Lin、Changxin Tian、Yupeng Hou 和 Wayne Xin Zhao。2022年。《通过邻域增强的对比学习改进图协同过滤》。载于*WWW '22*：*2022*年*ACM*万维网大会。ACM，2320–2329。[17] Zhuoran Liu、Leqi Zou、Xuan Zou、Caihua Wang、Biao Zhang、Da Tang、Bolin Zhu、Yijie Zhu、Peng Wu、Ke Wang 和 Youlong Cheng。2022年。《Monolith：采用无冲突嵌入表的实时推荐系统》。载于与第*16*届*ACM*推荐系统会议同期举办的第*5*届在线推荐系统与用户建模研讨会论文集，João Vinagre、Marie Al-Ghossein、Alípio Mário Jorge、Albert Bifet 和 Ladislav Peska（编），第3303卷。CEUR-WS.org，美国华盛顿州西雅图。[18] Fei Mi、Xiaoyu Lin 和 Boi Faltings。2020年。《ADER：通过自适应蒸馏样例回放实现会话推荐的持续学习》。载于*RecSys 2020*：第十四届*ACM*推荐系统会议。ACM，408–413。[19] Jianmo Ni、Jiacheng Li 和 Julian J. McAuley。2019年。《利用远程标注的评论和细粒度方面为推荐提供依据》。载于*2019*年自然语言处理经验方法会议暨第*9*届自然语言处理国际联合会议论文集（*EMNLP-IJCNLP*）。计算语言学协会，188–197。[20] Danni Peng、Sinno Jialin Pan、Jie Zhang 和 Anxiang Zeng。2021年。《学习用于增量更新推荐系统的自适应元模型生成器》。载于*RecSys '21*：第十五届*ACM*推荐系统会议，阿姆斯特丹。ACM，411–421。[21] Ruihong Qiu、Hongzhi Yin、Zi Huang 和 Tong Chen。2020年。《GAG：面向流式会话推荐的全局属性图神经网络》。载于第*43*届国际*ACM SIGIR*信息检索研究与发展会议论文集。ACM，669–678。[22] Steffen Rendle、Christoph Freudenthaler、Zeno Gantner 和 Lars Schmidt-Thieme。2009年。《BPR：基于隐式反馈的贝叶斯个性化排序》。载于*UAI 2009*：第二十五届人工智能不确定性会议论文集。AUAI Press，452–461。[23] Tobias Schnabel、Mengting Wan 和 Longqi Yang。2022年。《将推荐系统置于实践情境中：迈向归纳学习与增量更新》。*CoRR* abs/2211.06365（2022年）。[24] Jianing Sun、Yingxue Zhang、Wei Guo、Huifeng Guo、Ruiming Tang、Xiuqiang He、Chen Ma 和 Mark Coates。2020年。《面向推荐的邻居交互感知图卷积网络》。载于第*43*届国际*ACM SIGIR*信息检索研究与发展会议论文集。ACM，1289–1298。[25] Jianing Sun、Yingxue Zhang、Chen Ma、Mark Coates、Huifeng Guo、Ruiming Tang 和 Xiuqiang He。2019年。《多图卷积协同过滤》。载于*2019*年*IEEE*国际数据挖掘会议（*ICDM*）。IEEE，1306–1311。[26] Robin Verachtert、Lien Michiels 和 Bart Goethals。2022年。《我们是否忽略了什么？使用最优训练窗口正确评估推荐系统》。载于与第*16*届*ACM*推荐系统会议（*RecSys 2022*）同期举办的*2022*年推荐系统评估观点研讨会论文集（*CEUR*研讨会论文集，第*3228*卷）。CEUR-WS.org。[27] Weiqing Wang、Hongzhi Yin、Zi Huang、Qinyong Wang、Xingzhong Du 和 Quoc Viet Hung Nguyen。2018年。《基于流的排序推荐系统》。载于第*41*届国际*ACM SIGIR*研究与开发会议信息检索. ACM，525–534。[28] Xiang Wang、Xiangnan He、Yixin Cao、Meng Liu 和 Tat-Seng Chua。2019年。KGAT：用于推荐的知识图谱注意力网络。载于第*25*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议（*KDD*）论文集。ACM，950–958。[29] Xiang Wang、Xiangnan He、Meng Wang、Fuli Feng 和 Tat-Seng Chua。2019年。神经图协同过滤。载于第*42*届 *ACM SIGIR* 信息检索研究与开发国际会议论文集。ACM，165–174。[30] Xiang Wang、Hongye Jin、An Zhang、Xiangnan He、Tong Xu 和 Tat-Seng Chua。2020年。解耦图协同过滤。载于第*43*届 *ACM SIGIR* 信息检索研究与开发国际会议论文集。ACM，1001–1010。[31] Jiancan Wu、Xiang Wang、Fuli Feng、Xiangnan He、Liang Chen、Jianxun Lian 和 Xing Xie。2021年。用于推荐的自监督图学习。载于*SIGIR '21*：第*44*届 *ACM SIGIR* 信息检索研究与开发国际会议。ACM，726–735。[32] Shiwen Wu、Fei Sun、Wentao Zhang、Xu Xie 和 Bin Cui。2023年。推荐系统中的图神经网络：综述。*ACM* 计算调查55，5（2023年），97:1–97:37。[33] Jiafeng Xia、Dongsheng Li、Hansu Gu、Jiahao Liu、Tun Lu 和 Ning Gu。2022年。FIRE：基于图信号处理的快速增量推荐。载于*WWW '22*：*ACM Web* 会议*2022*。ACM，2360–2369。[34] Jiafeng Xia、Dongsheng Li、Hansu Gu、Tun Lu、Peng Zhang 和 Ning Gu。2021年。用于协同过滤的增量图卷积网络。载于*CIKM '21*：第*30*届 *ACM* 信息与知识管理国际会议。ACM，2170–2179。[35] Yishi Xu、Yingxue Zhang、Wei Guo、Huifeng Guo、Ruiming Tang 和 Mark Coates。2020年。GraphSAIL：面向推荐系统、具备图结构感知能力的增量学习。载于*CIKM '20*：第*29*届 *ACM* 信息与知识管理国际会议。ACM，2861–2868。[36] Rex Ying、Ruining He、Kaifeng Chen、Pong Eksombatchai、William L. Hamilton 和 Jure Leskovec。2018年。面向 Web 规模推荐系统的图卷积神经网络。载于第*24*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议（*KDD*）论文集。ACM，974–983。[37] Junliang Yu、Hongzhi Yin、Xin Xia、Tong Chen、Lizhen Cui 和 Quoc Viet Hung Nguyen。2022年。图增强是否必要？用于推荐的简单图对比学习。载于*SIGIR '22*：第*45*届 *ACM SIGIR* 信息检索研究与开发国际会议。ACM，1294–1303。[38] Jiani Zhang、Xingjian Shi、Shenglin Zhao 和 Irwin King。2019年。STAR-GCN：面向推荐系统的堆叠式重构图卷积网络。载于第二十八届国际人工智能联合会议论文集，*IJCAI*。ijcai.org，4264–4270。[39] YangZhang、Fuli Feng、Xiangnan He、Tianxin Wei、Chonggang Song、Guohui Ling 和 Yongdong Zhang。2021年。通过因果干预利用推荐中的流行度偏差。载于*SIGIR '21*：第*44*届 *ACM SIGIR* 信息检索研究与开发国际会议。ACM，11–20。[40] Yang Zhang、Fuli Feng、Chenxu Wang、Xiangnan He、Meng Wang、Yan Li 和 Yongdong Zhang。2020年。如何重新训练推荐系统？一种序列元学习方法。载于第*43*届 *ACM SIGIR* 信息检索研究与开发国际会议论文集。ACM，1479–1488。[41] Yu Zheng、Chen Gao、Jianxin Chang、Yanan Niu、Yang Song、Depeng Jin 和 Yong Li。2022年。解耦长期与短期兴趣以进行推荐。载于*WWW '22*：*ACM Web* 会议 *2022*。ACM，2256–2267。[42] Fan Zhou 和 Chengtai Cao。2021年。利用经验回放克服图神经网络中的灾难性遗忘。载于第三十五届 *AAAI* 人工智能会议（*AAAI 2021*）、第三十三届人工智能创新应用会议（*IAAI 2021*）、第十一届人工智能教育进展研讨会（*EAAI*）。AAAI 出版社，4714–4722。