计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440204
61(8): 2067−2096, 2024

图终身学习:综述

刘 壮[1] 董子宸[1] 董宜琳[1] 尚家名[1] 张 帆[1] 陈雨然[1] 楼佩妍[1] 孙欣然[1] 王 昱[1] 赵 军[2]
Wayne Lin[3]
[1](东北财经大学金融科技学院 辽宁大连 116025)
[2](大连理工大学计算机学院 辽宁大连 116024)
[3](南加州大学计算机学院 美国加利福尼亚州洛杉矶 90007)

Lifelong Graph Learning: A Comprehensive Review

Liu Zhuang[1], Dong Zichen[1], Dong Yilin[1], Shang Jiaming[1], Zhang Fan[1], Chen Yuran[1], Lou Peiyan[1], Sun Xinran[1], Wang Yu[1], Zhao Jun[2], and Wayne Lin[3]
[1](*School of Fintech*, *Dongbei University of Finance and Economics*, *Dalian*, *Liaoning* 116025)
[2](*School of Computer Science*, *Dalian University of Technology*, *Dalian*, *Liaoning* 116024)
[3](*School of Computer Science*, *University of Southern California*, *Los Angeles*, *CA*, *USA* 90007)

Abstract Lifelong Graph Learning (LGL) is an emerging field aimed at achieving continuous learning on graph-structured data, addressing the catastrophic forgetting issue in existing tasks and enabling models to adapt to newly emerging graph tasks in a sequential manner. Despite demonstrating strong learning capabilities, improving LGL's sustained performance remains a crucial challenge. To address the gaps in existing research, we provide a comprehensive survey and summary of recent developments in the field of LGL. Firstly, we reclassify existing methods of LGL, focusing particularly on approaches to overcome catastrophic forgetting. Secondly, we systematically analyze the strengths and weaknesses of these methods, and discuss potential solutions for achieving sustained performance improvements. Our study emphasizes on how to avoid forgetting old tasks during the continual learning process while swiftly adapting to the challenges of new tasks. Finally, we also discuss the future directions of LGL, covering potential impacts in the application domains, open issues, and specifically analyze their potential effects on sustained performance improvements. These discussions will help guide future research directions in LGL, promoting further development and application in this field.

Key words lifelong graph learning; graph neural networks; replay; regularization; continual learning; incremental learning; catastrophic forgetting

摘 要 图终身学习(lifelong graph learning, LGL)是一个新兴领域,旨在实现对图结构数据的持续学习,以解决现有任务上的灾难性遗忘问题,并使得顺序更新的模型能够适应新出现的图任务.尽管LGL展现出良好的学习能力,但如何持续提高其性能仍然是一个至关重要的问题.为填补现有研究对这一方面的空白,对最近在LGL领域的研究进行了全面调查和总结.首先,重新分类了LGL的现有方法,重点关注克服灾难性遗忘的方法.随后,系统地分析了这些方法的优缺点,并探讨了实现持续性能提升的潜在解决方案.该研究着重于如何在持续学习的过程中避免对旧任务的遗忘,同时快速适应新任务的挑战.最后,还就LGL的未来发展方向进行了讨论,涵盖了其在应用领域、开放性问题等方面的潜在影响,并具体分析

收稿日期:2024-03-15;**修回日期**:2024-04-23
基金项目:国家自然科学基金(72272028)
This work was supported by the National Natural Science Foundation of China (72272028).
通信作者:刘壮(liuzhuang@dufe.edu.cn)

了这些方向对持续性能改进的潜在影响.这些讨论将有助于指导未来LGL研究的方向,推动这一领域的进一步发展与应用.

关键词 图终身学习（LGL）;图神经网络（GNN）;重放;正则化;持续学习;增量学习;灾难性遗忘

中图法分类号 TP391

图终身学习(lifelong graph learning, LGL)是一个旨在实现图结构数据终身学习的新兴领域,可以连续学习的同时保留以前学到的知识.尽管LGL在研究界受到广泛关注,大多研究仅仅局限于解决灾难性遗忘问题,而忽视持续性能改进的问题.为此,本文将全面概述LGL的最新进展,包括现有最先进的分类法、潜在应用以及开放研究问题的讨论等,以应对LGL面临的独特挑战[1].此外,分析了如何应用现有各种方法实现LGL的性能提升.最后,讨论了目前仍然亟待解决的问题,并提出了未来的发展方向.

随着深度学习在生活各领域的成功应用,人们开始渴望更强大的通用人工智能.尽管具有很好的发展前景,基于神经网络的终身学习(lifelong learning, LL)还面临着一个严重的遗忘问题:在新任务上的学习通常会导致旧任务上性能的急剧下降,这被称为灾难性遗忘(catastrophic forgetting, CF)[2].将终身学习(lifelong learning, LL)与图学习(graph learning, GL)结合起来显然是减轻灾难性遗忘的普遍解决方案,二者的结合巩固也被称为LGL.尽管LGL具有潜力,但由于欧几里得数据和图之间的结构差异,一般终身学习和LGL之间存在简单或复杂的差异,包括模型、任务设置和方法[3].此外,终身学习和LGL都主要关注于克服灾难性遗忘,而忽略了终身的性能改进.尽管关于LGL的研究越来越多,但关于LGL的调查却很少.为了弥补这一差距,本文旨在对LGL的研究工作进行一个全面的调查.

1 准备工作

1.1 终身学习

1.1.1 终身学习概念

终身学习是一种前沿机器学习技术,也被称为终身机器学习(lifelong machine learning, LML).终身机器学习的架构如图1所示.其核心在于在连续的、非平稳的任务流中持续地学习新知识并保持旧知识的有效性.这种方法下,学习的目标和数据都可能随着时间的推移而发生变化,不受每次训练时的任务目标是否独立同分布(independent and identically distributed, IID)的限制[4-5].这一特性使得终身学习在面对现实世界中快速变化和复杂任务的情境下具有显著的优势,因为它能够适应环境的动态性,并在不断变化的情况下保持学习的持续性.

传统机器学习的经典模式是在给定数据集上运行机器学习算法和构建模型,然后将模型应用于实际任务[6].这种学习方式被称为孤立学习,因为它仅仅关注当前任务,而不考虑其他相关信息或过去学习到的知识.然而,孤立学习的不足在于缺乏记忆,即它无法保留学到的知识以供未来学习使用,因此需要大量的训练样本[7-8].终身学习则模仿了人类学习的方式和能力,不断积累过去学到的知识,并利用这些知识来学习更多.在这个过程中,系统会不断地学习新任务,并将需要保留的信息或知识发送到存储器中.终身学习是一种前沿深度学习技术,它在连续非平稳任务流中持续学习新知识并保持旧知识,学习的目标和数据会随时间而变化,不要求每次训练时的任务目标是独立同分布的.在学术界,终身学习有多个近义词,例如持续学习、增量学习或不间断学习.本文将这种在连续非平稳任务流中不断学习新知识并不忘记旧知识的前沿机器学习技术称之为终身学习.

目前,主流的机器学习范式是孤立学习的.在这种模式下,算法直接从给定的训练数据集中生成模型,而不会试图保留已学知识以提高未来学习的效率.尽管孤立学习已经取得了成功,但它需要大量的

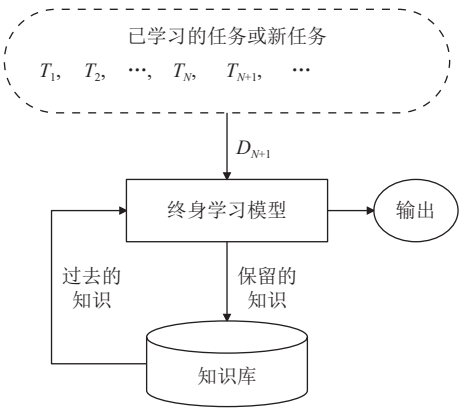


Fig. 1 Lifelong machine learning architecture

图1 终身机器学习架构

训练实例，并且只适用于定义明确且范围狭窄的任务.

随着机器学习领域的迅速发展，本文呼吁摒弃传统的孤立学习模式，转向更具普适性和灵活性的终身学习研究，以推动机器学习领域迈向新的高度. 在当前的技术背景下，一个系统若缺乏积累与应用先前学得知识的能力，其真正智能的实现将面临障碍[7]. 诸如智能助手、聊天机器人和物理机器人等应用迫切需要具备终身学习能力的支持，以适应不断变化的环境和任务需求.

1.1.2 终身学习的动机

在机器学习领域，终身学习致力于解决模型训练中普遍存在的一个缺陷：灾难性遗忘. 简言之，一般的机器学习模型，尤其是基于反向传播的深度学习方法，当在新任务上训练时，往往会导致在旧任务上的表现显著下降[9].

针对灾难性遗忘问题，研究者们提出了 2 种主要策略：一是通过增加模型容量来存储新知识以缓解遗忘；二是研究不同的灾难性遗忘机制，以使模型保持在“稳定性-可塑性”环境中. 尽管第一种策略能在一定程度上有效地缓解遗忘问题，但在实际应用中，模型容量是有限的，无法无限增长. 因此，本文重点分析了各种不同的灾难性遗忘求解策略，包括正则化、重放、架构调整、特征表示和预演等策略[10]，以及结合不同单种策略优势的联合策略.

1.1.3 终身学习的特点

终身学习包括 3 个特点：

1）模型在学习新知识的同时，能够保留先前学习到的大部分知识，使得在旧任务和新任务上均能展现出良好的表现.

2）随着类别数量的增加，计算能力和内存需求应该保持稳定或缓慢增长. 理想情况下，一旦完成某一任务的学习，该任务的观测样本就可以被全部丢弃.

3）模型应该具备终身学习新知识的能力，无论是从新任务还是新数据中学习. 即使新任务在不同的时间出现，模型也应该是可训练的.

1.1.4 终身学习的应用

随着人工智能技术的不断发展，传统的深度神经网络单一的训练模式已经难以满足更贴近现实应用的非平稳数据流的学习需求. 当前阶段，终身学习在工业领域的应用主要体现在将终身学习算法与功能模型相结合，以适应多任务和多场景的变化[11]. 图 2 展示了终身学习在应用场景中的一般框架. 作为一种辅助手段，终身学习方法结合了诸如目标检测等功能性模型，使得系统具备一定的灾难性遗忘抵抗能力，能够有效执行非平稳数据流任务，并且在学习过程中不断地积累和转移知识.

1）计算机视觉. 视觉计算是终身学习应用最为广泛的领域，各种终身学习策略研究基本都是以连续视觉识别任务作为评测任务. 在语义分割领域，卷积神经网络（CNN）的引入提高了语义分割的效果，然而传统模型中，所有类别事先被定义且只能进行一次性学习，这种设置在实际应用中受到了一定的限制. 在实际场景中，一种能够持续学习新类别并保持旧任务性能的终身语义分割方法逐渐成为一个新兴的趋势.

2）自主机器人. 随着全球机器人市场规模不断扩大，机器人技术正从机器人 3.0 时代迈向机器人 4.0 时代. 这一发展要求机器人不仅具备感知和智能协作能力，还需要在面对不断变化的现实环境时具备理解和决策能力，最终实现自主服务. 终身学习的发展使得自主机器人的实现成为可能[12-13].

3）自然语言处理. 目前，终身学习的研究主要集中在计算机视觉领域，而在自然语言处理领域却并未得到足够的关注. 这其中一个主要原因是自然语言处理领域目前主要关注于以 BERT 为代表的自监督表示学习. 随着 ChatGPT 大规模预训练模型的广

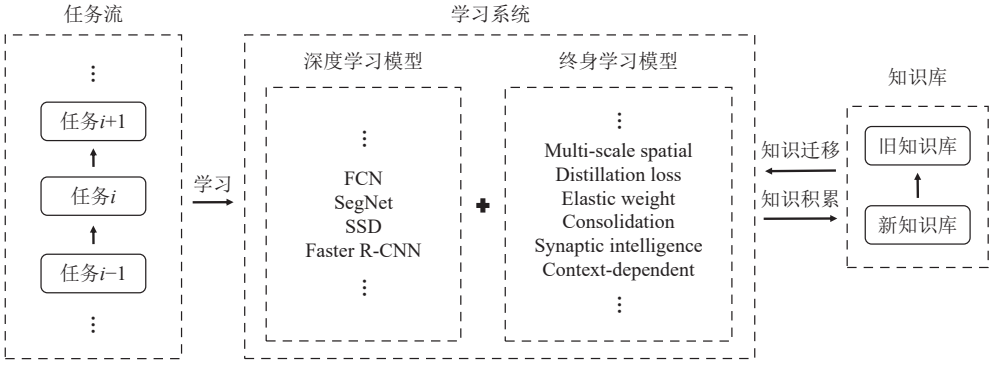


Fig. 2 The general framework of lifelong learning application scenarios

图 2 终身学习应用场景的一般框架

泛应用,终身学习在自然语言处理中的应用价值并不十分明显.

1.1.5 终身学习面临的困难和方法

终身学习本身是一个非常开放的概念,目前基于深度学习的增量学习研究大多限制在有监督分类、任务式增量和多头网络结构的框架下.然而,这种特定领域的训练方案通常难以直接应用于高度动态化和非结构化的真实环境中.

目前的终身学习方法通常隐含地要求任务的性质差异不大.当任务的性质和难度差异较大时,大部分终身学习方法的性能都会严重下降,甚至低于简单的基线模型.另外,有许多研究表明目前尚无任何一种终身学习方法在所有条件下都表现良好,大部分终身学习方法对于模型结构、数据性质和超参数设定都十分敏感.因此,探索在所有任务设置中表现更为稳健的终身学习方法具有重要意义.

1.2 图学习

图,又称为图网络,具备从各种丰富实体之间的关系中提取信息的能力.社交网络、生物网络、专利网络、流量网络、引文网络和通信网络等多种常见图形式被广泛应用于不同领域中.图学习作为一种先进的方法,能够有效解决节点预测、边缘预测、图预测等任务.近年来,随着人工智能技术的持续发展,图学习在捕获复杂关系和对应网络结构方面备受瞩目,受到研究者和从业者的广泛关注[14].

然而,图在真实世界的应用程序中可能会不断发展,这意味着可能会出现新类型的节点及其关联的边,或者在分类任务中包含新的类标签.此外,与许多流行的机器学习算法一样,图学习算法受到孤立学习范式概念的限制,即模型不能执行连续的学习来适应终身学习新任务和保留新的信息.在这种情况下,图数据可能会表现出分布偏移,终身学习模型会不断地适应新的任务,从而不可避免地遭受图中灾难性遗忘问题.

鉴于此,将终身学习与图学习结合起来开始受到各个领域越来越多的关注.终身学习与图学习的结合,即 LGL,正逐渐引起各个领域的广泛关注.LGL 的发展潜力巨大,然而由于图数据的复杂性,现有的 LGL 方法在目标图类型、学习设置、基本技术和评估指标方面具有高度异构的特点,这给试图进入该领域的研究人员带来了挑战.此外,由于该领域仍处于起步阶段,尽管 LGL 的研究数量不断增加,但对于 LGL 的综述却相对较少.因此,本文旨在提供关于 LGL 研究进展的全面综述,并聚焦于持续性能的改进.

除了旨在减轻遗忘问题的 LGL 努力外,我们将阐述 LGL 与一些容易混淆的主题之间的关键区别,例如动态图.动态图学习(dynamic graph learning, DGL),也称为时间图嵌入[15],可以分为 2 种图类型:一种是离散时间演化图,它将图表示为在不同时间步骤下的动态图快照;另一种是连续时间演化图,这种图携带更多的信息和更具复杂性,被定义为图流而不是快照[16].一些工作涉及离散时间演化图,例如 Goyal 等人[17]提出的 DynGEM 方法,该方法基于深度自动编码器,旨在通过处理离散时间内的图快照来执行动态嵌入.DynGEM 利用从前一个时间步骤学习到的嵌入来初始化当前时间步骤的嵌入,以传递知识并确保新的嵌入与之前的保持接近.另一方面,zhou 等人[18]提出了一种新的动态网络嵌入方法,称为动态三联体,其目的是通过施加一个三联体(3 个顶点组)来学习动态图的表示,同时保留时间信息.这些动态图学习方法与 LGL 共享相同的目标,即在不同时间步骤中转移过去的知识,产生合并的知识表示以帮助模型有效地学习.

然而,动态图学习与 LGL 之间存在明显的差异.动态图学习旨在积累历史知识并捕获新的模式,从而避免从头开始的训练[19].大多数动态图学习(dynamic graph learning, DGL)研究的主要目标是捕获不断变化的图结构,并保持最新的图表示,以方便访问所有先验信息,而不是解决遗忘问题.相比之下,LGL 专注于解决遗忘问题,通常在以前任务的数据不可用的情况下操作.此外,动态图学习仅关注图的嵌入过程,而不考虑下游任务的添加,例如在任务中添加新的类别或数据实例,这与终身学习的概念密切相关.此外,在学习新知识时,模型没有考虑执行向后转移或者细化图表示的预先嵌入.特别地,目前尚没有明确的机制来解决灾难性遗忘问题,而这正是终身学习的主要挑战.

LGL 是一种旨在逐步学习新模式的同时避免灾难性遗忘发生的方法.在 LGL 中,图结构数据带来了 2 个新的挑战:样本(节点)级的依赖关系和任务(图)级的依赖关系.前者是由节点之间的连接生成的,而后者则源自前一个图和新的子图之间的连接.处理这 2 种依赖关系是 LGL 的核心任务.在适应不断变化的图结构数据时,LGL 需要有效地管理和利用这些依赖关系,以确保模型在学习新任务时能够保持对旧任务的记忆,并且能够在新任务中适应性地整合先前学到的知识[20].这种能力使得 LGL 能够在复杂的图数据中持续学习,并不断改进模型的性能.

2 图终身学习（LGL）

LGL是一个新兴的跨学科领域，旨在整合终身学习和图学习，在不遗忘已有知识的图基础上逐步学习新知识的模式. 目前，大多数图学习方法假设图是静态的，并且在训练过程中是已知的. 然而，这种假设限制了这些方法的适用性，因为这些方法无法应用于随着时间推移而逐步出现的底层图任务. 因此，对于这种类型的应用，我们需要一种终身学习的方法，该方法能够连续地学习图数据以适应新的信息，同时保留先前学到的知识. 为了实现在图像和文本等领域的连续学习，LGL方法关注多个层面的样本数据，包括样本级别、特征级别和任务级别的连接数据[21]；同时考虑静态图和动态图，使得模型能够在适应新数据的同时缓解灾难性遗忘所带来的问题. 通过增量学习，LGL方法实现了模型持续性能的改进，这是因为在不断学习的过程中保持对过去知识的积累，同时灵活地适应新的数据和任务，从而使得模型能够在不断演化的图结构中保持高效性.

2.1 图持续学习的发展前景

终身学习和图学习各自在处理数据时面临一些限制，如终身学习仅考虑单个样本数据而忽略了样本间的联系，而图学习假设图是静态的，无法应对动态变化的图结构. LGL作为图学习和终身学习的整合，正是为了弥补这些缺陷而诞生的. LGL可以处理动态图数据，使模型能够在图结构不断变化的环境中持续学习，并且保留先前学习到的知识. 表1详细展示了在图结构数据上进行学习任务时的性能表现.

1）动态图数据处理. 现实世界中的图数据通常是动态变化的，新的节点和边不断加入，旧的节点和边不断消失或更新[19, 22-23]. LGL具有处理这种动态变化的能力，使得模型能够实时更新并适应新的数据. 这种机制使得模型可以不断适应图的演化，从而保持其在动态环境中的有效性和性能.

2）知识保留与转移. LGL克服了灾难性遗忘问题，即在学习新数据时不会忘记已经学到的知识，这对于终身学习在复杂图结构数据上的应用至关重要，有助于保持模型的稳定性和性能. LGL不仅关注知识的前向迁移，而且还关注知识的后向迁移. 知识的前向迁移是指从旧任务到新任务序列进行知识转移的过程，将旧任务中学习到的知识帮助后续任务学习；而知识后向迁移的过程是从新任务中转移知识来帮助提高先前任务[24]的性能.

3）跨领域应用. LGL不仅适用于特定领域，如社交网络[25]，还可以扩展到其他领域，如金融学习[26-30]、交通流量预测[31]等. 在这些领域中，图数据同样具有丰富的信息和复杂的结构，LGL可以帮助我们更好地理解和利用这些数据. 这种跨领域的应用潜力将进一步推动LGL的发展.

4）与深度学习的结合. 深度学习在动态图[16]处理中已经取得了显著的成功，但其主要考虑模型是否适应新数据，而忽略了灾难性遗忘问题，而LGL与深度学习的结合将进一步提高模型的性能. 通过结合图嵌入、图神经网络（GNN）等技术，LGL可以更好地捕捉图数据的复杂结构和潜在信息. 这种结合将为LGL带来更广阔的应用前景，使得模型在不同领域的图数据处理中发挥更大的作用.

Table 1 Performance of Learning in Graph Structure

表1 在图结构中学习的性能表现

图结构学习	描述	性能表现
在线学习	该模型暴露于一个连续的信息流中	–
向前传输	从以前的任务中学到的知识可以促进未来的学习	+
向后传输	从后期任务中学到的知识可以支持早期任务	–
知识整合	该模型容易发生灾难性遗忘	+
固定模型大小	关于任务编号的模型大小保持不变	=
固定存储大小	关于任务编码的存储大小保持不变	=
任务和数据自由	没有任务空间和数据空间的限制	–
高效率	时间复杂度比再训练低得多	+

注：“+”表示性能严格符合预期，“=”表示一般符合预期，“–”表示低于预期.

2.2 LGL的动机

LGL的动机在于使模型能够灵活地适应进化的图结构变化，并不断改进先前的知识以获取新的信息，从而能够有效地应对基于图的领域中的未知任务. 为了实现这一目标，LGL研究了有效地重新训练模型的策略，并确定了正确的重新训练时机，同时也考虑了需要保留多少过去数据的问题. 目前的图学习方法是有限的，这些方法只能适应预定义的任务[32-33]. 而在图结构数据中，随着时间的推移，图的拓扑结构和节点属性可能会发生变化，新的节点和边可能会不断引入，旧的节点和边可能会变得不再重要. 模型为了保持对这些变化的适应性，LGL需要解决如何在学习新任务时有效地利用先前学到的知识来避免灾难性遗忘的问题，这就需要探索何时以及如何更新模型，何时应该保留过去的数据. 模型通过LGL能够不断地从新的图结构中学习，并根据先前的经验

进行调整和更新，从而使得模型能够更好地适应未知的新任务. 这种灵活性和持续性的学习方式，使得LGL在应对图结构数据不断变化的环境中具有重要的意义和价值. 因此，研究LGL的动机在于提高模型的泛化能力和适应性，使其能够在不断演变的图结构数据中保持有效性和性能.

2.3 与其他学习方法的比较

2.3.1 动态图表示学习

动态图表示学习旨在捕获不断变化的图结构，并持续更新图的表示，以确保对所有先验信息的访问，而不仅仅解决遗忘问题[4,32,34-35]. 该方法侧重于掌握动态图在每个节点的嵌入过程，但并未考虑到下游任务的动态添加. 虽然动态图表示学习提供了一种更新图结构的方式，但并未解决模型如何在学习新任务时利用旧任务的经验问题，这导致了灾难性遗忘的挑战[36]. 尽管如此，动态图表示学习在捕获图的变化方面仍具有一定优势，尤其适用于对动态性强、信息频繁变化的图结构数据进行建模.

2.3.2 图神经网络（GNN）

GNN作为一种新兴的深度学习框架，直接针对图结构数据进行学习，其出色的性能引起了学术界的广泛关注和深入探索. 通过制定适当的节点和边的策略，GNN能够将图结构数据转换为规范且标准的表示形式，并输入到多种不同的神经网络中进行训练. 在节点分类、边信息传播和图聚类等任务上，GNN展现出优异的表现[27,29,36-43]. 它在处理不随时间变化的图结构数据方面表现出色，通过节点之间的信息传递和聚合，能够有效地学习到节点的表示. 然而，现实世界中的许多图网络是动态的，如社交网络、金融交易和推荐系统，这些图结构在时间上不断演变和变化，而GNN作为静态图结构处理的算法，在处理这些动态图时存在一定的挑战. 一个主要的挑战是GNN难以有效地适应动态图结构的变化. 在动态图中，新的节点可能会加入，现有节点之间的连接关系可能会改变，这些变化会导致静态GNN模型无法被直接应用. 例如，当新用户加入社交网络或新产品推出时，GNN需要重新训练以适应这些变化，而这可能会消耗大量的计算资源和时间. 因此，对于动态图的实时变化，GNN在处理新节点或新边引入新类的情况时存在一定的限制[36,41-42]. 尽管GNN在静态图的特征学习和表示方面具有显著优势，但对于动态图的适应能力有限. 为了解决这一问题，一些研究者提出了对GNN进行扩展，使其能够处理动态图的情况. 例如，一种常见的做法是引入时间维度[40,42]，将图结构的变化建模为时间序列，从而使GNN能够在时间上进行迭代更新. 另一种方法是通过记忆机制[27,29,36-39,43]，允许模型记忆和更新过去的信息，以适应新的动态变化. 这种方法在一定程度上提高了GNN在动态图上的表现，但仍然存在一些挑战和限制. 未来的研究方向可能包括进一步改进GNN的动态适应性，使其能够更好地处理动态图的变化. 例如，可以探索更有效的时间建模方法、设计新的记忆机制，或者结合其他学习框架来增强GNN在动态图上的性能. 这些努力将有助于推动GNN在动态图领域的应用和发展.

2.3.3 离散时间演化图

离散时间演化图作为一种动态图的表示方式，利用深度自动编码器来执行动态嵌入[4,32,35]. 它通过在离散时间内的图快照上执行嵌入，利用从前一个时间步学习到的嵌入来初始化当前时间步的嵌入，以传递知识并确保当前嵌入与前一个时间步的接近程度. 这种方法不仅能够有效地处理动态图的变化，还能在学习新任务时利用旧任务的经验，有助于解决灾难性遗忘问题. 因此，离散时间演化图在终身学习的情境下展现出了良好的潜力和适用性.

由此可见，动态图表示学习、GNN和离散时间演化图在处理动态图时各自具有优势和局限性. 动态图表示学习注重图的动态性和最新性，但忽略了终身学习中的关键问题；GNN在静态图上表现出色，但对动态图的处理有限；而离散时间演化图在动态嵌入和灾难性遗忘方面表现出色. 根据具体任务需求和应用场景，选择合适的方法进行动态图分析和终身学习是至关重要的.

2.4 灾难性遗忘问题

2.4.1 根本原因

灾难性遗忘现象是指在连续学习多个任务的过程中，学习新知识的过程会迅速破坏之前获得的信息，导致模型在旧任务中的性能急剧下降[7,15,17,19,21]. 其根本原因在于新知识对现有知识的覆盖. 传统模型假设数据分布是固定或平稳的，且训练样本是独立同分布的，因此模型可以一遍又一遍地看到所有任务相同的数据. 然而，当数据变为连续的数据流时，训练数据的分布就变得非平稳. 在模型从非平稳的数据分布中持续不断地获取知识时，新知识会干扰旧知识，从而导致模型性能的快速下降，甚至完全覆盖或遗忘之前学习到的旧知识.

2.4.2 LGL的克服方法

1)经验重放与迁移学习. 经验重放与迁移学习

是一种有效的方法，旨在克服 LGL 中的挑战. 该方法的核心思想是存储一些先前学习过的样本，并在训练新数据时将这些样本与新数据混合. 这样做有助于模型回顾并巩固已学习的知识，防止模型在学习新任务时忘记旧任务. 同时，该方法还利用在旧任务上学到的知识来加速新任务的学习过程，从而提高了整体学习效率.

在 LGL 中，经验重放与迁移学习起到了重要的作用. 通过迁移学习，模型能够更好地保留先前的知识，并将这些知识应用到新的学习任务中. 这样一来，模型可以有效利用已有的知识库，提高泛化能力和适应性. 在实践中，经验重放和迁移学习的结合不仅能够帮助模型克服灾难性遗忘的问题，还可以使模型更具备持续学习的能力，适应不断变化的任务和环境[13,32,44].

因此，经验重放与迁移学习作为 LGL 的克服方法，为模型的持续学习和适应性提供了有效的手段. 它不仅有助于防止遗忘问题，还能加速新任务的学习过程，提高模型的泛化能力. 在未来的研究和应用中，可以进一步探索和优化这些方法，以应对图结构数据的持续学习需求.

2)正则化与约束. 为了提高图持续学习的泛化能力并防止过拟合，可以引入正则化项或约束条件，如 MSLGL[22] 中所提到的. 正则化项和约束条件在 LGL 中起到了关键作用[32,45-53]，有助于优化模型并保持其稳定性. 首先，L1 正则化是一种常见的正则化方法，通过向损失函数添加模型权重的绝对值之和来惩罚大的权重. 这种方法有助于保持模型的稀疏性，从而防止对旧任务的遗忘. 在 LGL 中，L1 正则化可以帮助模型保持对旧任务的记忆，减少不必要的参数，提高模型的稳定性和泛化能力. 其次，L2 正则化，也称为权重衰减，是另一种常见的正则化方法. 该方法通过在损失函数中增加权重的平方和来限制权重的大小. 这种方法有助于稳定模型的参数，减少在学习新任务时对旧任务知识的干扰. 在 LGL 中，L2 正则化可以有效地控制模型的复杂度，避免过拟合，同时提高模型在新任务上的表现. 除了正则化项，引入约束条件也是一种有效的方法. 约束条件可以确保模型在学习过程中的稳定性和一致性. 例如，可以通过约束模型的权重范围或限制模型的输出，从而进一步提高模型的泛化能力和适应性.

因此，正则化与约束是在 LGL 中用于提高模型泛化能力和稳定性的重要手段. 通过引入 L1 或 L2 正则化以及适当的约束条件，可以有效控制模型的复杂度、防止过拟合，并保持模型在学习新任务时对旧任务的记忆. 在未来的研究中，可以进一步探索和优化这些方法，以更好地应对图结构数据的持续学习需求.

3)弹性权重共享. 弹性权重共享是 LGL 中的一项重要技术，在某些情况下起到了关键作用[31,54-63]. 这种技术允许模型在多个任务之间共享部分权重，从而在学习新任务时减少过拟合的风险，同时保持对旧任务的记忆. 一个典型的例子是 TrafficStream，具体来说 TrafficStream 是一个基于 GNN 和持续学习的交通流量预测框架，它采用了弹性权重共享技术来改善 LGL 的效果[31]. 在弹性权重共享中，TrafficStream 可以根据不同任务的特性动态地调整权重共享的程度. 这样一来，该模型可以在保留旧任务知识的同时，更好地适应新任务的数据特征，提高了模型的泛化能力和适应性. 通过这种方式，弹性权重共享不仅有助于减少对新任务的过拟合，还能够在学习过程中更好地平衡新旧任务之间的权衡关系. 在 TrafficStream 中，弹性权重共享技术的应用使得模型能够在处理交通流量预测等任务时取得良好的效果. 通过允许部分权重的共享，模型可以更有效地利用已有的知识，并在学习新任务时保持稳定性和一致性. 这种技术的引入为 LGL 提供了一种灵活而高效的方法，能够平衡新旧任务的需求，提高整体的学习性能和效果.

总的来说，弹性权重共享技术在 LGL 中具有重要意义. 它通过允许模型在多个任务之间共享权重，减少了对新任务的过拟合风险，同时保留了对旧任务的记忆. 在未来的研究和应用中，可以进一步探索和优化弹性权重共享技术，以提高 LGL 的性能和适应性.

4)知识蒸馏和模型压缩. 知识蒸馏和模型压缩是 LGL 中常用的技术，旨在将大型、复杂模型的知识有效地转移到小型、高效的模型中，以降低计算成本并提高模型的部署效率[26,37-38,54,64-66]. 这些方法的应用使得学生模型在保持与教师模型相当性能的同时，具备了更小的模型规模和更低的计算成本. 首先，知识蒸馏技术通过让小型模型学习大型模型的知识，实现了知识的传递和转移. 大型模型拥有更多的参数和复杂的结构，在训练过程中可以产生丰富的知识表示. 而小型模型通过蒸馏大型模型的知识，可以在保持相当性能的同时，大幅度减小模型规模和计算成本. 这使得模型更加轻量化，更适合于资源受限的环境下的部署和应用. 其次，模型压缩技术进一步

减小了模型的大小和计算成本. 通过剪枝、量化、低秩近似等方法，模型压缩可以在不明显损失性能的情况下，显著减少模型的参数量和计算量. 例如，DyGRAIN[38]，ERDIL[26]，Geometer[54] 等模型在 LGL 中应用了模型压缩技术，取得了良好的效果. 这些方法的成功应用进一步证明了模型压缩在 LGL 中的重要性和有效性.

因此，知识蒸馏和模型压缩是 LGL 中的关键技术，为模型的轻量化和高效部署提供了重要手段. 这些技术的应用使得学生模型能够在保持性能的同时，具备更小的模型规模和更低的计算成本. 在未来的研究和应用中，可以进一步探索和优化这些技术，以满足图结构数据持续学习的需求，并促进模型的实际应用和部署.

2.5 持续性能改进

目前对 LGL 的研究主要集中在减轻灾难性遗忘的问题，而忽略了持续的性能改进. 持续性能改进作为终身学习的最终目标，指的是持续获取新知识并对现有知识进行补充[4,21,26,31,34,31-38,54].

2.5.1 知识增强

知识增强指的是后续任务的知识可以增强先前任务的知识，从而在处理不断变化的图数据时不断优化和改进模型的性能.

1)动态图嵌入与表示学习. 动态图嵌入与表示学习[67] 专注于学习能够随时间变化的图数据的低维表示. 通过捕捉图的拓扑结构、节点属性和时间演化信息，动态图嵌入能够生成有效的节点表示，从而提升图持续学习的性能. 这种方法使得模型能够更好地理解图数据的演化过程，从而更准确地适应新的数据和任务.

2)增量学习与自适应更新. 增量学习允许模型在处理新数据时仅更新部分参数或结构，而不是重新训练整个模型. 关键在于设计能够自适应调整模型参数和结构的算法，以适应图数据的动态变化，例如 MSLGL[37]，RLC[58] 等算法. 通过增量学习，模型可以在不断学习新知识的同时保持对旧知识的积累，进而提高整体的学习效率和性能.

3)图结构与关系的强化. 强化图的结构和关系表达能够增强模型的感知能力. 例如，引入 GNN 等模型能够执行归纳学习[50]，可以更好地捕捉图数据的复杂结构和关系，进而提升模型的性能. 通过对图结构进行建模和学习，模型能够更准确地理解节点之间的关联和影响，从而在处理图数据时取得更好的表现.

4)对抗性训练与鲁棒性增强. 通过对抗性训练，模型能够抵御噪声、异常值和攻击等干扰因素，从而提高其鲁棒性. 这种方法可以帮助模型更好地应对实际应用中可能出现的各种挑战，保证模型在面对不确定性和噪声时依然能够稳健地工作. 对抗性训练是一种有效的方法，使得模型在面对复杂环境和攻击时仍能保持高效性和可靠性.

由此可见，知识增强在 LGL 中扮演着重要的角色，通过动态图嵌入、增量学习、图结构强化和对抗性训练等方法，使得模型能够持续学习、适应新的数据和任务，并不断提高其性能和鲁棒性. 这些方法的结合应用，为 LGL 提供了强大的工具和策略，使得模型能够在不断变化的环境中保持有效性和可靠性.

2.5.2 优化控制

优化控制方法旨在控制学习过程，涵盖了多种技术和策略，共同确保学习过程的效率和性能的提升.

1)自适应学习率调整. 在 LGL 的过程中，学习率的设置对模型的收敛速度和性能至关重要. 自适应学习率调整方法，如 Adam[55]，RMSprop[56] 等，能够根据模型的训练情况动态调整学习率，加快收敛速度并避免陷入局部最优. 这些方法使得模型能够更灵活地适应不同的数据和任务，从而提高学习的效率和性能.

2)优化器的选择与设计. 选择合适的优化器对图持续学习的性能至关重要. 除了传统的梯度下降优化器，还可以考虑使用更先进的优化器，如动量优化器[68]、AdaMax[7,55] 等，能够更好地处理梯度消失或爆炸等问题，提高学习过程的稳定性. 优化器的选择和设计直接影响模型的收敛速度和最终性能，因此在 LGL 中，需要综合考虑不同优化器的特点和适用场景，选择最适合的优化器.

3)模型剪枝与压缩. 随着 LGL 过程中数据的不断增加和模型的更新，模型可能会变得过于复杂，导致计算和存储成本增加. 通过模型剪枝和压缩技术[69]，可以去除冗余的参数和结构、降低模型的复杂度、提高计算效率. 这些技术可以帮助模型在不损失性能的情况下减少参数量、加速推理过程，并且使得模型更适合部署在资源受限的环境中.

4)集成学习与模型融合. 集成学习结合了不只一种终身学习方法，以利用每种方法最大化模型的性能[24,61-62]. 在 LGL 中，可以利用集成学习的方法将多个不同时间步或不同结构的模型进行融合，以获得更稳定、更准确的预测结果. 这种方法可以降低模型过拟合的风险，提高模型在处理新任务时的泛化能

力，并且能够更好地捕捉图数据的复杂性和关联性.

2.6 LGL 方法分类

LGL 本质上是终身学习的一个子领域，因此其目标与常规终身学习相同，即通过增量学习不断提高模型的性能. 然而，由于图中节点之间的相互依赖性，LGL 的方法比常规图学习更加复杂和多样化. 为此，我们提出了当前 LGL 方法的新分类法，共分为 6 类：基于重放的方法、基于正则化的方法、基于架构的方法、基于表示的方法、基于预演的方法和混合方法，如图 3 所示.

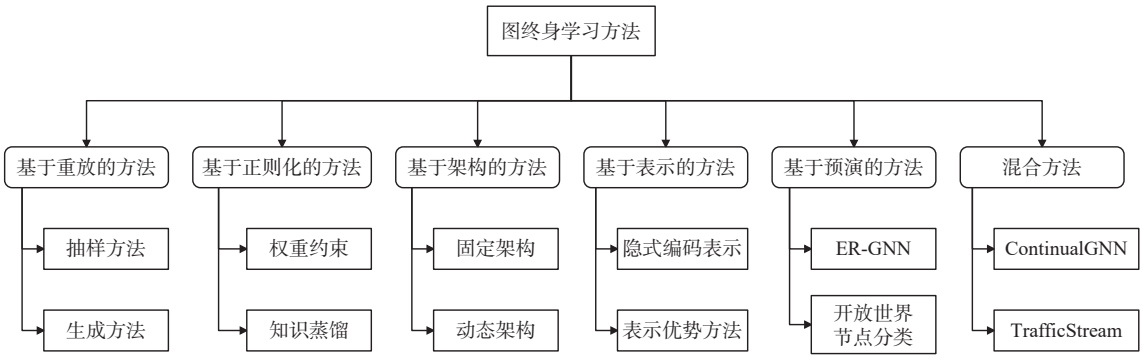


Fig. 3 The classification of lifelong graph learning methods

图 3 LGL 方法的分类

由于目前的研究较少涉及图持续性能的改进，我们拟定的分类法主要集中在那些明确声称可以克服灾难性遗忘的方法上. 接下来，将从知识的角度探讨灾难性遗忘的根本原因和持续性能改进的关键，并进一步讨论这些方法是否以及如何实现持续性能的改进.

3 基于重放的方法

基于重放的方法通过维护一个记忆缓冲区，保存先前任务的信息，并在学习新任务时重放信息来避免灾难性遗忘. 在记忆检索方面，一些方法提取了涉及特定经验的情境记忆（例如：原始样本的子集），而其他方法则根据一些结构知识（例如：生成的伪样本、旧任务的梯度）提取语义记忆. 基于回放的方法的关键在于获取从以前的任务中学到的知识，这些知识通常是通过抽样或生成模型获得的. 图 4 总结了基于重放的方法. 基于重放的方法是 LGL 中一种重要策略，它允许模型在学习新任务时保持对旧任务的记忆，从而防止灾难性遗忘的发生. 这种方法利用记忆缓冲区存储先前任务的关键信息，当需要学习新任务时，模型可以从这个记忆中重放以前的经验. 这样一来，模型可以继续积累知识，同时适应新的任务要求，保持在不断变化的图数据中的有效性. 在基于重放的方法中，记忆的提取和重放是关键步骤. 一些方法会提取涉及特定经验的情境记忆，例如提取存储原始样本的子集，这些样本可能是关键的代表性样本. 另一些方法则会根据一些结构知识，如生成的伪样本或利用旧任务的梯度信息来提取语义记忆. 从而可以帮助模型更好地理解和记忆先前任务的关键特征和信息.

基于重放的方法在 LGL 中扮演着重要角色，该方法通过建立和维护记忆缓冲区，以及重复使用先前任务的经验，为模型提供了持续学习和适应新任务要求的机制. 通过不断地回顾和重新学习旧任务的经验，模型能够在应对不断变化的图数据时保持稳健性和高性能水平. 这种方法的应用不仅有助于模型应对新的挑战和任务，还有效地避免了灾难性遗忘现象的发生.

3.1 抽样方法

抽样方法具指从以前的任务中选择一些历史数据并将其存储在缓冲区中，通过保存所学的知识以克服灾难性遗忘. 由于缓冲区空间有限，如何开发适当的采样策略以有效地利用缓冲区是一个挑战. 具体来说，抽样方法涉及到缓冲区的构建和更新，即对于缓冲区的构建，采样策略应该指定缓冲区的大小和采样的范围；对于缓冲区更新，采样策略需要指定节点选择的标准是什么，以及如何定量地评估这些标准.

3.1.1 随机抽样

最简单的策略是随机抽样，在该方法中，每个节点以等概率被选择. StreamingGNN[57] 提到了一种均匀随机抽样方法，该方法从目前看到的图中选择一个固定大小的样本来用新数据进行训练. 然而，这种

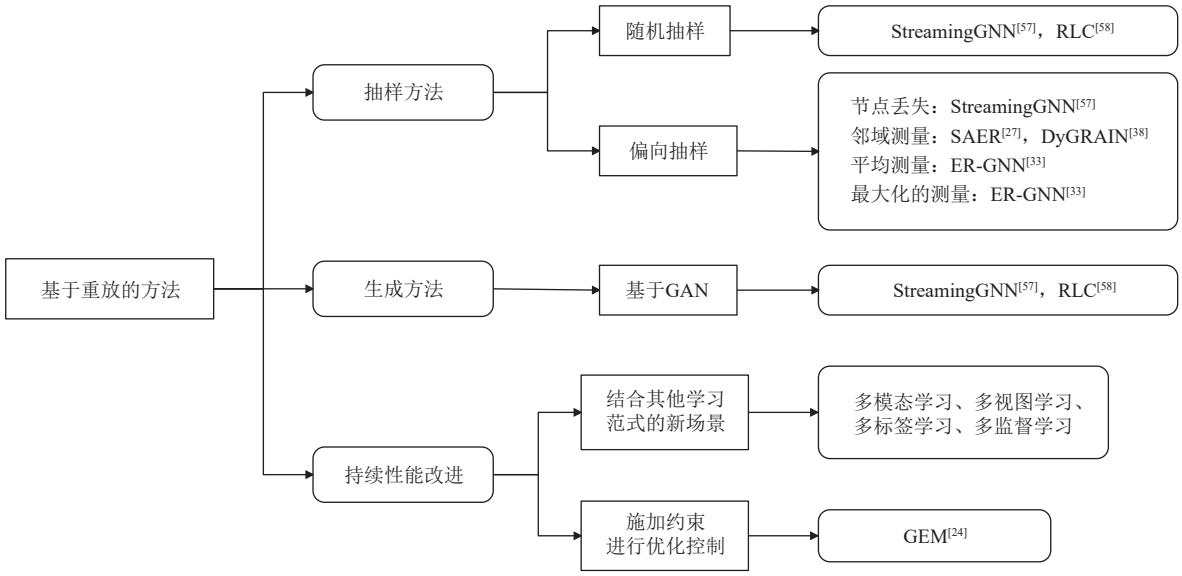


Fig. 4 Replay-based lifelong graph learning methods

图 4 基于重放的 LGL 方法

抽样只能很好地处理非平衡数据，而不能处理非平衡数据. 处理复杂结构依赖关系的另一种策略是有偏差采样，该采样方法中的节点以不同的概率被选择，节点的重要性越高，被选中的概率就越大. 因此，关键问题是节点重要性的度量. RLC[58] 采用储层采样策略，以等概率随机选择节点.

然而，随机抽样忽视了节点的重要性，因此不能有效地更新缓冲区. 随着任务数量的增加，随机采样要么通过多次重复采样历史节点来构建缓冲区，要么通过随机替换缓冲区中的节点进行更新.

3.1.2 偏向抽样

偏向抽样考虑节点之间的差异，并根据某些规则赋予它们重要性，例如特征平均值、节点损失、节点度、影响力、覆盖范围、结构相似性和属性相似性. 偏向抽样也遵循固定大小的缓冲区原则，通常基于蓄水池采样进行开发. 由于涉及更多复杂的策略，偏向抽样通常优于随机采样. 对于不同的偏向采样方法，一个重要的问题是它们是否能对拓扑结构的变化保持敏感，并根据这些变化高效地更新节点的重要性. 另一个问题是基于更新策略重要性的采样是否能够有效地保留先前任务的知识，从而有效地克服灾难性遗忘进而实现连续的性能改进.

邻居测量遵循一个直观的观点，即节点的重要性由其邻居决定. SAER[27] 认为与大型节点相比，交互次数较少的节点信息含量较低. 因此，提出了每次交互的概率与其用户节点的度数的倒数成正比，以防止信息丢失的观点. DyGRAIN[38] 强调了时间变化的接受域，并从结构和特征的角度定义了影响传播. DyGRAIN 引入了传播和影响矩阵以及相应的分数来衡量新数据对每个节点的影响. 基于节点的特征向量提出了特征影响分数，以捕获节点接受域中的变化量，得分高的节点将被选为重放的重要节点.

StreamingGNN[57] 提出通过计算节点损失来确定节点的优先级，被采样的概率与节点的预测误差成正比. 这种策略基于稀有节点的误差比普通节点的误差下降得慢，并且稀有节点上的更多选择会导致更少的遗忘.

ER-GNN[33] 提出均值和最大化测量方法. 识别重要节点的一种简单方法是采用被嵌入最接近每个类别特征的均值的固定大小的节点，也被称为特征均值. 然而，该方法受到每个类别中节点数量的限制. 针对节点数量较少的限制，ER-GNN[33] 提出了覆盖率最大化，旨在最大化嵌入空间的覆盖率，使其与同一任务中其他类别节点数量成正比且距离固定. 影响最大化将参数变化作为衡量节点重要性的指标，其中参数变化通过比较去除节点后得到的参数和未去除节点时得到的最优参数来评估，变化越大，说明该节点越重要. 由于每个被移除的训练节点的影响计算成本很高，所以将去除训练节点转化为在最优参数上加权.

相似度度量是确定节点重要性的常用方法，可以通过多种方式进行定义，包括结构相似度、属性或特征相似度以及分布相似度. SEA-ER[27] 认为，重放节点应能够很好地推广到其他节点. 因此，提议选择与其他节点具有最接近结构相似度的节点. 具体而言，结构相似度是通过节点到其他节点的最短路径

距离来评估的. ContinualGNN[32] 认为, 如果节点的属性与其邻居节点明显不同, 则该节点很重要; 邻居中不同标签的节点数越多, 节点的重要性越高. TrafficStream[31] 提出了一种基于输出特征在不同时间间隔中的分布差异来检测发生了很大变化的节点的方法. 分布差异通过 Jensen-Shannon 散度来衡量, 该数值越大意味着变化越大, 受影响的节点越多. 而较低的分数则表明数据流更稳定, 可以被选择进行重放.

3.2 生成方法

生成方法通过生成模型产生以前任务的伪数据以保留旧知识, 并与新数据进行融合. 该方法的核心在于生成模型本身, 具体而言, 包括 2 个关键问题: 1)如何有效地利用生成模型生成的伪数据来表示先前的知识; 2)生成模型如何在面对增量任务时进行更新. SGNN-GR[70] 提出了一种基于生成对抗网络(generative adversarial network, GAN)框架的生成模型用于生成节点邻域. 该方法将节点邻域的学习问题转化为图上的随机游走学习. 与简单的随机游走相比, 该方法允许在少数步骤内对节点邻域进行全面探索, 从而保证了生成模型的良好性能.

在 LGL 的背景下, 理想的生成模型应能够进行增量更新, 并随着任务的增加生成更为优质的伪数据, 从而更好地保留先前的知识. 目前已经涌现出许多深度图生成模型, 包括基于 GAN 的(例如 MolGAN[71])、基于变分自编码器(Variational AutoEncoder, VAE)的(例如 GraphVAE[72])、基于流的(例如 GraphAF[73])、基于强化学习的(例如 GCPN[74])以及基于自回归的(例如 GraphRNN[75] 和 BiGG[76])等多个类别. 现有的研究中, SGNN-GR[70] 专注于基于 GAN 的模型, 为该方法的进一步探索留下了广泛的空间. 在这些生成模型中, 基于 GAN 的方法通常通过对抗训练的方式学习生成数据的分布, 从而能够生成与真实图数据类似的样本. 例如, MolGAN 利用 GAN 的框架生成分子图, 并通过判别器和生成器之间的博弈, 逐步提高生成样本的质量. 相比之下, 基于 VAE 的方法则通过 VAE 的学习方式生成数据, 例如 GraphVAE 可以通过编码器和解码器的结构实现对图结构的压缩和解码, 从而生成图数据. 其次基于流的方法采用流模型来学习数据的分布, 并且能够进行快速的采样和生成. 例如, GraphAF 利用自动流模型对图数据的生成过程进行建模, 从而实现了高效的图生成. 基于强化学习的方法, 如图卷积策略网络(graph convolutional policy network, GCPN)[74] 利用强化学习框架, 通过与环境的交互学习生成数据. 这种方法能够在生成过程中考虑到不同节点之间的关系和特性, 从而生成更加丰富和真实的图数据. 最后, 基于自回归的方法(如 GraphRNN 和 BiGG)则通过递归地生成图的结构和节点特征来实现图的生成. 基于自回归的方法可以根据已生成的部分来预测下一个节点或边, 从而逐步生成整个图. 这种自回归的方式能够有效地保持图的结构和连续性, 生成的图数据更加符合真实数据的特征.

不同类型的深度图生成模型各具特点, 适用于不同场景下的图数据生成需求. 在 LGL 中, 这些模型的应用可以帮助模型在学习新任务时生成更多样化、更真实的数据, 从而更好地保留先前学习到的知识. 对于基于 GAN 的模型, 如 MolGAN 等, 其生成的图数据质量可能受到噪声和模式限制的影响, 因此在未来的研究中, 可以探索如何进一步提高生成图的质量和多样性.

3.3 持续性能改进

基于重放的方法仅存储部分历史数据, 这些数据虽然可以将部分现有知识转移到下一个任务, 但不能将下一个任务的新知识转移到先前任务. 在这个过程中并没有明确的知识增强, 因此理论上无法实现持续性能改进. 然而, 模型的性能由许多因素共同决定, 如数据质量、数据量、任务难度和算法能力. 在确定任务并选择算法后, 数据的质量和数量是性能的决定性因素. 如果后续任务中所包含的知识, 如数据分布, 与先前任务的相似或一致, 也可能存在持续性能改进的机会. 实现这种可能性的其他方法还包括一些结合其它学习范式的新场景, 例如多模态学习、多视图学习、多标签学习和多监督学习. 在这些范式中, 前一个任务的数据通过新的模态、视图、标签和监督(从弱到强, 从监督到无监督等)进行增强.

此外, 可以通过施加约束来进行优化控制, 从而实现持续性能改进, 如 GEM[24] 中提出的约束先前任务损失不增加的方法. 其他方法可能包括各种正则化和优化方法, 如提前停止、信息漏失等.

4 基于正则化的方法

基于正则化的方法是一种有效的策略, 旨在通过明确考虑拓扑结构, 并向损失函数添加相应的正则化项来平衡新旧任务的学习, 以限制参数的剧烈变化, 从而克服灾难性遗忘. 其着重解决 2 个核心问题: 1)哪些拓扑信息对任务至关重要? 2)如何有效地对结构信息进行编码? 通过明确考虑拓扑结构,

并向损失函数添加相应的正则化项能够平衡新旧任务的学习，并限制参数的剧烈变化，以保持先前任务中参数的重要性. 通常，正则化项可分为约束和蒸馏2种方式. 图5总结了基于正则化的方法.

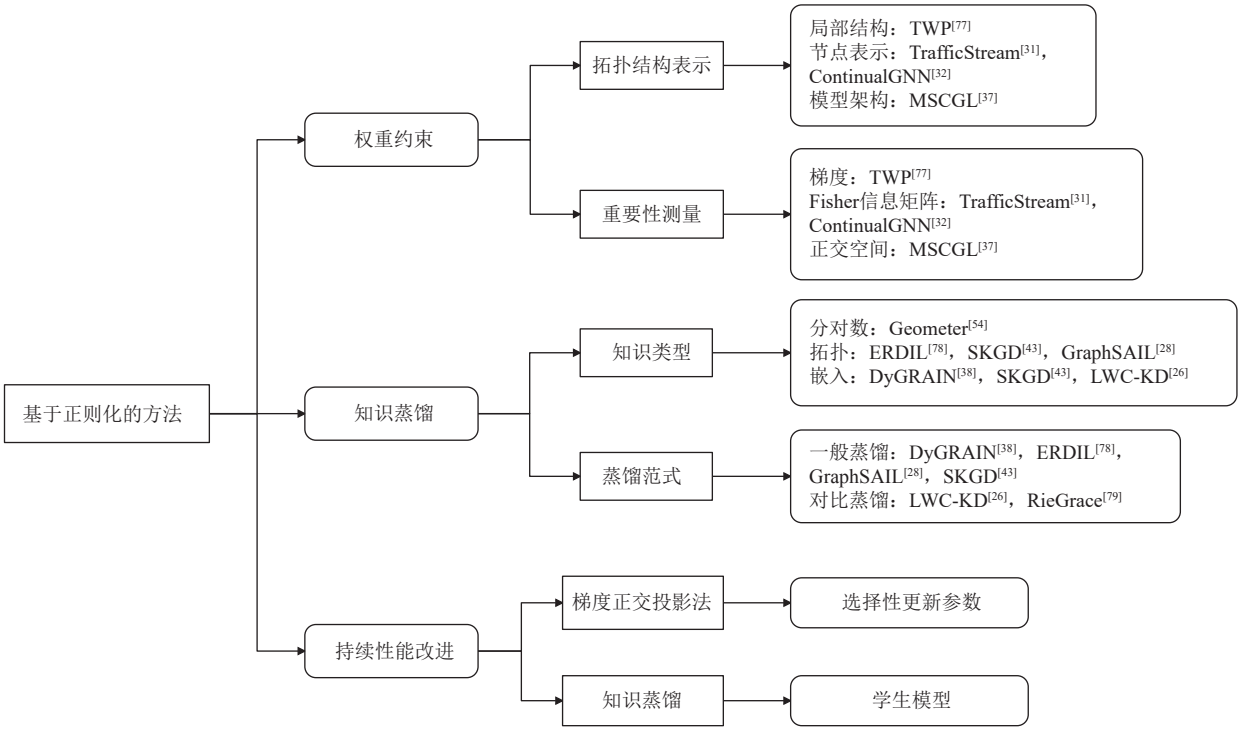


Fig. 5 Regularization-based lifelong graph learning methods

图5 基于正则化的LGL方法

1)约束(constraint). 正则化的一种方式是在损失函数中引入参数的约束条件，以控制参数的范围和变化. 例如，L1正则化通过向损失函数添加模型权重的绝对值之和来惩罚大的权重，有助于保持模型的稀疏性，从而避免对旧任务的遗忘. 相比之下，L2正则化(也称为权重衰减)则通过增加权重的平方和到损失函数中以限制权重的大小，提高模型的泛化能力，并降低对旧任务知识的干扰.

2)蒸馏(distillation). 另一种常见的正则化方式是知识蒸馏，即将大型、复杂模型的知识转移到小型、高效的模型中. 这种技术有助于在保持与教师模型相当性能的同时减小模型的规模和计算成本. 通过蒸馏技术，模型可以更有效地利用先前任务的知识提高新任务的学习速度和性能.

4.1 权重约束

从本质上讲，LGL中的权值约束方法与一般终身学习中的权值约束方法相同，都是将重要参数的更新限制在之前的任务中. 不同的是，在LGL中拓扑信息对任务的学习也很重要. 目前的工作通过各种方式将拓扑信息显式或隐式地纳入终身学习过程，如关注[77]、节点表示[32]和架构[31,37].

权重约束方法通过惩罚旧任务重要参数的更新来保留旧知识，同时限制了新任务的学习. 考虑拓扑结构时要着重关注2个问题，即哪些知识可以表示拓扑结构以及如何对它们进行编码，哪些参数是重要的以及如何测量其重要性，这2个问题涉及拓扑表示和重要性测量.

4.1.1 拓扑表示

在图持续学习任务中，考虑拓扑信息是必要的. 在大多数情况下，关注图的局部结构对于有效的GNN至关重要，这是因为，当GNN使用2层或3层图卷积时，它们通常能够更好地处理过度平滑问题，从而表现出最佳的性能. 对拓扑信息进行编码的一般方法是通过节点间的注意力机制. TWP[77]是“注意即相关性”原则的典型实践，通过引入非参数化的注意力机制，将注意力系数推广到一般的GNN. TWP模型明确地将局部结构作为拓扑信息，并将其集成到深度图模型中. 然而，在许多情况下，拓扑信息没有显式编码. ContinualGNN[32]和TrafficStream[31]将新增节点的2跳邻居识别为受影响节点，其结构信息包含在受影响节点表示的变化中.

除了节点表示的变化，模型体系结构也是知识

的载体. MSLGL[37] 根据神经结构搜索的结果，移除旧体系结构中过时的块并添加新的块.

4.1.2 重要性度量

在评估参数的重要性时，一种常用的方法是利用前一任务中的 Fisher 信息矩阵来估计这些参数的重要性. Fisher 信息矩阵是一个正定矩阵，其逆矩阵即为协方差矩阵，这使得它能够用于计算参数估计量的标准误差. 此外，Fisher 信息矩阵在解决灾难性遗忘问题中扮演了关键作用，因为它帮助我们识别出在之前任务中扮演重要角色的参数，从而在学习新任务时对这些参数进行特别关注. ContinualGNN[32] 和 TrafficStream[31] 等模型均采用了这种方法，即通过使用 Fisher 信息矩阵的对角元素进行近似计算以提高计算效率.

由于参数的更新是由梯度方向决定的，TWP[76] 将注意力系数相对于特定参数的梯度定义为其重要性，并将特定任务目标视为重要的知识，同时将特定任务损失相对于特定参数的梯度定义为其重要性，最终得到的分数是注意力系数和特定任务损失这两者重要性的加权和. 在寻找重要参数时，一种简单的方法是将前一个任务的所有参数都视为重要参数，并按照前一个参数空间的正交方向更新来学习新任务的参数. MSLGL[37] 则是通过搜索最佳的任务特定架构和参数来鼓励更新与前一个任务正交的参数.

总的来说，参数重要性的衡量既考虑了学习任务的重要性，也考虑了拓扑信息的保持. 这种衡量方法包括梯度、Fisher 信息矩阵和正交投影等. 然而，目前大部分工作都隐含地包含了拓扑信息，这给拓扑表示和参数重要性估计带来了效率问题.

4.2 知识蒸馏

在 LGL 中，知识提炼的主要目的是为了提高模型的性能，而非模型压缩. 约束方法通过在参数空间对模型进行正则化，而蒸馏方法则对模型的泛函空间进行正则化. 因此，蒸馏方法可以防止模型偏离之前学到的知识从而保留旧知识. 除了一般的逻辑和基于嵌入的知识外，基于拓扑的知识也是 LGL 中提炼的重要考虑因素，例如局部结构、全局结构和关系结构等，明确哪种基于拓扑的知识对任务学习至关重要是一个关键问题.

与传统的两步蒸馏方案相比，LGL 中的知识蒸馏可以看作是一种自蒸馏模式. 这里将之前学习的模型比作老师，当前学习的模型则是学生，因此可以将蒸馏方法简化为 2 个关键问题：老师希望学生学到什么知识(提炼什么)？知识如何通过蒸馏从老师传递给学生(如何蒸馏)？这 2 个问题被概括为知识类型和蒸馏范式.

4.2.1 知识类型

在早期的知识蒸馏中，基于逻辑的知识被广泛用于蒸馏，在图深度模型中也存在这种情况. Geometer[54] 计算软化对数的 KL 散度，使学生模型能够学习教师模型对旧班级进行分类.

图中的拓扑信息也是一个重要的知识来源，一些工作[28,77-78] 明确地保留拓扑信息以供终身学习. ERDIL[78] 将范例关系图中的角度关系视为重要的知识，并利用精馏来传递结构信息以进行终身学习. GraphSAIL[28] 考虑了推荐系统中重要的局部和全局结构信息，并通过知识蒸馏保存它们，SKGD[43] 明确保留边缘权重作为记忆知识图的结构信息进行蒸馏. 除了基于逻辑的知识和基于拓扑的知识外，基于特征的知识是用于提炼的另一种知识，其中学生模型从教师模型中学习中间层特征，代表性作品有 DyGRAIN[38]、LWC-KD[26]、GraphSAIL[28]、SKGD[43]、RieGrace[79]. 这些模型都遵循师生架构，将基于嵌入的信息作为知识进行提炼，包括中间层嵌入、师生表示之间的互信息、中心节点及其相邻节点嵌入、低级特征和高级特征、语义特征等.

4.2.2 蒸馏范式

蒸馏范式规定了如何蒸馏知识，其与构建的蒸馏体系结构和蒸馏目标密切相关. 基本的蒸馏范式是一般蒸馏，它遵循通用的“教师-学生”架构，利用教师模型的预测结果作为监督信号来指导学生模型的学习过程. 大多数研究都采用了一般蒸馏方法，例如 DyGRAIN[38]、ERDIL[78]、Geometer[54]、SKGD[43] 和 GraphSAIL[28]. 也有少数研究进行对比蒸馏分析，通过利用特定于数据或任务的增强来最大化不同增强视图下的特征一致性，从而学习更稳健地表征[80]，这使得终身学习更有可能减少遗忘，甚至持续提高性能.

另一方面，在对比蒸馏中，学生模型不直接模仿教师模型的预测，而是通过生成对比示例来训练以区分正确的类和其他不正确的类. 学生模型学习区分具有不同标签的相似输入对，这使模型能够捕获更细粒度的细节并提高其泛化能力，RieGrace[79] 和 LWC-KD[26] 则是使用对比蒸馏的典型例子.

在 LGL 中，基于正则化的方法可以帮助模型保持对先前任务的记忆，并在学习新任务时有效地利用已有的知识. 这些方法的应用使得模型在处理动态图数据时更加稳健，同时避免了灾难性遗忘的问题. 对于约束方法，我们可以进一步探索如何设计适

合图数据的约束条件，以及如何调整正则化项的权重来平衡新旧任务的学习. 而在蒸馏方法方面，可以研究如何更好地传递知识，并探索不同的蒸馏策略以提高模型的性能. 这些方法的不断改进和优化将有助于推动 LGL 的发展，并为应对复杂图结构数据的持续学习提供更有效的解决方案.

4.3 持续性能改进

基于正则化的方法通常将后续终身学习视为贝叶斯框架下的后验分布近似问题，而将之前的数据分布视为先验知识并传递给下一个任务. 这种方法的本质在于建立在模型过度参数化的基础上，其中大多数参数对任务几乎没有贡献，因此可以近似设置为零或修剪. 然而，由于参数之间存在不可避免的更新干扰，通常需要在旧任务和新任务之间进行性能权衡. 一些研究使用梯度正交投影法(gradient orthogonal projection)有选择地更新参数，以最小化新任务学习对先前任务参数的影响，从而将遗忘程度降低到极低的水平. 然而，类似于基于重放的方法，这种方法在这个过程中缺乏知识的增强，理论上难以实现持续的性能提升. 在这一框架下，知识蒸馏作为一种可能对性能持续提升的方法，一些研究表明，学生模型可以优于教师模型，特别是当教师模型过度参数化或所面对的任务过于复杂时. 在这些情况下，学生模型通常通过相互学习、自我训练和集成学习等方式进行改进，实现超越. 这些方法增强了知识的多样性，因此在持续的图学习设置中，可能会实现持续的性能提升. 此外，这些方法也为 LGL 提供了更多探索的可能性，为解决图结构数据动态变化的挑战提供了新的思路.

因此，基于正则化的方法和知识蒸馏等技术为 LGL 的持续性能改进提供了有益的思路和方法. 在未来的研究中，我们可以进一步探索这些方法的深度和有效性以解决 LGL 中的挑战，并不断提高模型的学习能力和性能.

5 基于架构的方法

大部分终身学习策略基本上侧重于学习所有具有共享参数集(由单个模型和一个参数空间组成)的增量任务，容易造成任务间干扰，而构造特定于任务的参数可以明确地解决此问题. 因此，基于架构的方法就是通过特定架构为任务分配特定任务的参数或网络(部分共享或不共享)，以避免任务之间的干扰. 在一般的终身学习环境中，这种方法侧重于修改网络、激活函数或算法层的特定架构，以解决新任务并防止忘记以前的任务[81-83]，而在 LGL 场景中，这种方法依赖于改变图结构、扩展更多单元和执行压缩等技术. 根据架构的可变性，下面我们将从固定架构和动态架构分别展开讨论，并进一步讨论这 2 种架构如何实现持续的性能改进.

5.1 固定架构

具有固定架构的神经网络为每个任务分配特定的参数或神经元，并在学习新任务时冻结属于旧任务的参数或神经元以避免灾难性遗忘. 固定架构由于限制了模型容量，要求模型识别当前任务重要参数，并释放不重要的参数来学习新任务，因此随着新任务增量的引入，“空闲”参数趋于饱和，参数或神经元需要在任务之间部分共享，也就是对参数的使用进行稀疏性约束，并选择性地重用冻结的旧参数. 参数重用有 2 个主要问题：参数分配(如何获取最佳参数/架构)和参数效率(如何确保参数重用的效率，即架构和参数的最大共享).

5.1.1 参数分配

参数分配具有一个独立的参数子空间，专用于整个网络中的每个任务，指定如何将特定的模型参数分配给不同任务. 在参数分配的过程中可以通过迭代剪枝[69]、激活值[84]、神经网络搜索和不确定性估计[85]等方式明确识别当前任务的重要神经元或参数，并将不重要的部分释放到后面的任务中. MSLGL[37]从 GraphNAS[86]中汲取灵感，采用神经架构搜索，通过强化学习训练控制器为每个任务搜索最佳架构. 搜索过程表述为优化问题，并训练控制器联合搜索模型架构和权重；搜索空间包含 GNNs 的组件列表，包括聚合、激活和相关评估等操作.

5.1.2 参数效率

由于模型容量的限制，引入更多新任务时可用参数会逐渐减少. 为保证模型的可扩展性，固定架构方法需要考虑参数效率，即参数如何在任务之间重用或共享. 目前的工作通常是对模型参数施加稀疏约束的正则化，选择性地重用冻结的旧参数，确保参数的可重用性. MSLGL[37]对所有参数施加群稀疏约束，并对共享参数施加正交约束，前一操作保留先前任务的关键知识，后一操作可防止参数更新干扰先前的性能. 搜索空间使得参数在任务相同聚合之间共享，以减少不必要的模型扩展. 此外，还探索了几种参数共享策略，这些策略在相同的 GNN 单元、相同的聚集体和所有聚集体之间共享参数，这也提高了参数效率.

5.2 动态架构

当具有固定架构的模型容量达到学习极限(不足以很好地学习新任务)时，动态架构模型是固定模型的替代方案，可动态扩展模型架构用于持续学习. 由于架构随着任务增量的引入而扩展，因此通常需要寻找最佳架构，即架构学习. 此外，由于模型架构扩展的速度比任务增加的速度慢得多，为确保模型的可扩展性，动态架构的稀疏性和重用性约束的参数效率仍然十分重要.

5.2.1 架构学习

与固定架构方法相比，动态架构具有更宽广的参数空间和更灵活的架构学习策略，可以进行显式优化，提高参数效率和知识转移，如强化学习、神经架构搜索和变分贝叶斯等. RLC[58]由一个基于强化学习的控制器和一个可训练的基本网络子网络组成. 由长短期记忆(long short-term memory, LSTM)网络组成的基于强化学习的控制器旨在学习最佳状态策略，以从 CNN 的每个可训练层上用户定义的搜索空间中选择适当的行动过程.

5.2.2 参数效率

虽然动态架构没有模型容量的限制，但参数效率的提高也很重要. 参数效率过低，则模型架构扩展的速度赶不上任务增加的速度，导致模型快速膨胀，从而降低学习效率. 除了稀疏性约束之外，迭代修剪或冻结参数等方法也可以确保参数效率. RLC[58]采用剪枝的方式删除隐藏的 GNN 节点，以保持架构的可扩展性.

5.3 讨 论

基于架构的一系列方法通过改变模型架构构造特定于任务的参数以实现 LGL. 其中，MSLGL[37]，DiCGRL[35]等一些方法为特定任务或场景隔离网络组件以减轻灾难性遗忘. 虽然组件隔离可能会扩大网络大小，影响任务的学习，但在某些特定任务中组件隔离是必要的. 因此，不同的任务可能会有 2 个完全不同的解决方案，使用同一网络执行所有任务很难在每个任务上都获得令人满意的性能. 但隔离每个任务的特定组件也不可行，因为为每个任务训练一个独立的网络，具有很高的计算和内存开销，可能会将终身学习退化为通用的机器学习. 因此，组件共享对基于架构的方法实现知识转移也至关重要. 基于上述讨论，在设计基于架构的方法时，应考虑 2 个问题：1)在什么条件下我们需要分离或共享网络组件？2)重用共享组件时如何缓解灾难性遗忘？

针对问题 1，启发式和非启发式的策略可以为我们确定分离或共享网络组件时的条件. 首先，启发式策略利用先验知识来做出决策. 渐进式神经网络[82]、Piggyback[87]等方法假设每个新任务都与旧任务无关，并将其分配给特定的网络组件. 虽然能有效减轻灾难性遗忘，但在最坏的情况下，网络大小(特定于任务的组件数量)会随着增量任务的传入而线性增加，从而导致高计算和内存开销. 因此，组件不能完全隔离，应在相似的任务或场景之间共享，实现知识转移和控制空间复杂性. 此外，如果现有参数不能很好地处理新任务，网络扩展是必要的，否则就会造成资源浪费. 其实，在 LGL 中几乎没有启发式策略的研究，但我们可以从终身学习方法中汲取灵感. DEN[88]将任务相似性分解为单元级别以测量每个隐藏单元的语义漂移，组件在有轻微偏差的任务之间共享，而那些有大偏差的任务则进行组件隔离. 在 LGL 中，考虑到终身学习模型利用了图拓扑，图相似性是决定是否隔离参数的潜在关键因素. 例如，我们可以像 TWP[76]对图拓扑进行编码，并测量每个隐藏单元的拓扑漂移. 那些具有高拓扑漂移的单元可能会因新的传入任务而破坏所学的拓扑知识，因此应在任务中将其隔离. 非启发式策略鼓励网络组件隔离. DiCGRL[35]和 MSLGL[37]是 2 个采用非启发式策略的 LGL 项目. DiCGRL[35]导入注意力机制来选择相关组件，MSLGL[37]采用强化学习来搜索最佳网络架构.

针对问题 2，大多数终身学习的方法通常适用于共享组件，以确保先前学到的知识得以保留，并且有助于减轻灾难性遗忘的影响. 例如，MSLGL[37]通过对共享参数施加正交约束，而 DEN[88]则采用弹性惩罚来达到这个目的. 然而，目前针对 LGL 的研究主要集中在解决灾难性遗忘的问题上，对于持续性能的改进则相对不足. 这一局限性意味着我们还需要更多关于如何在终身学习过程中不断提升模型性能的研究，以实现更加全面和持久的学习效果，如图 6 所示.

5.4 持续性能改进

基于上述分析可知，基于架构的方法尚未能实现知识增强，也未能有效实现持续的性能改进. 由于不同基本架构的设计和选择会对持续学习的性能产生显著影响，例如，批量归一化[9]倾向于为当前任务引入偏差，从而导致先前任务的性能下降. 因此，我们需要对方法进行改进，以找到一套能够实现持续性能改进的终身学习方法. 同时，我们可以考虑利用其他组件、模块或网络来传递知识，从而实现持续的性能改进.

与此同时，我们观察到与其他方法相比，大多数

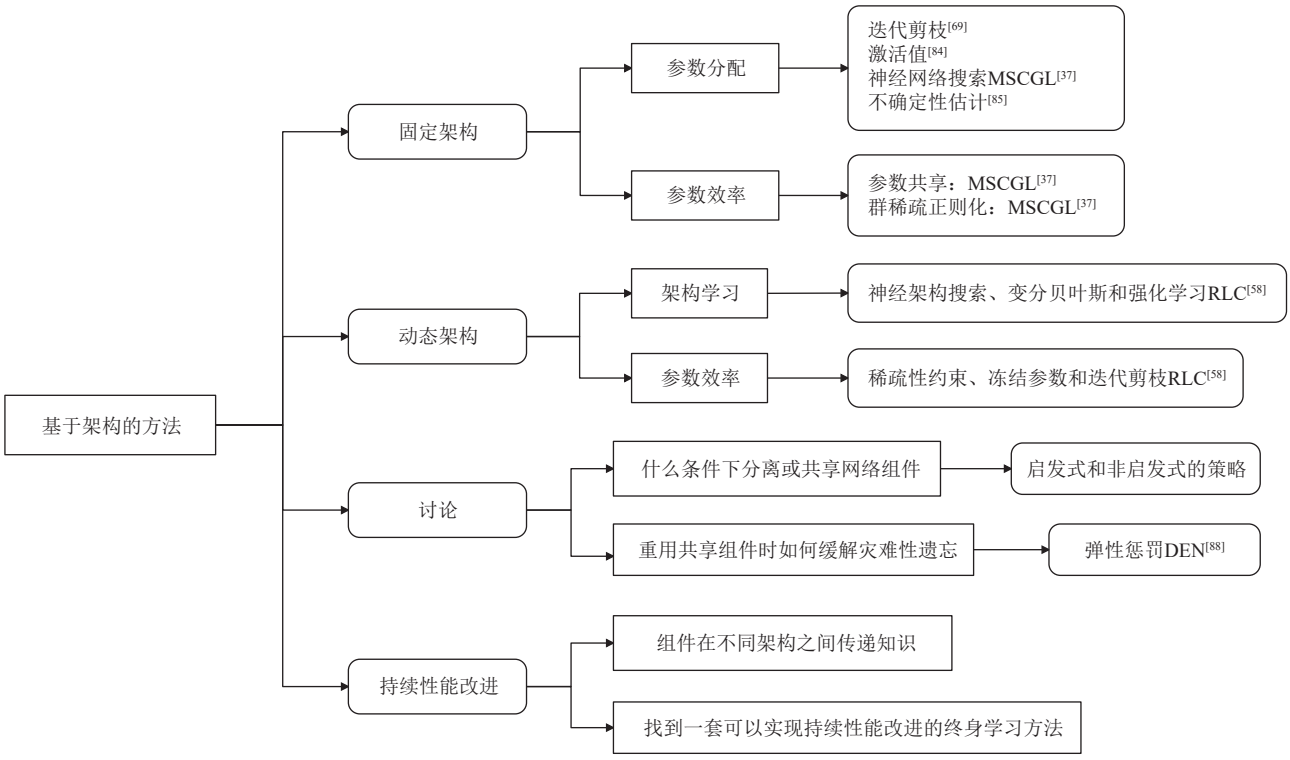


Fig. 6 Architecture-based lifelong graph learning methods

图 6 基于架构的 LGL 方法

基于架构的方法实际上是在网络参数中取消了增量任务之间的相关性，从而几乎完全避免了灾难性遗忘的问题. 然而，这种做法可能会影响模型的可扩展性和任务间的泛化性. 特别是针对通常需要使用任务标识来确定要应用的参数组，这在很大程度上限制了这些方法的适用范围. 为了克服这一局限性，我们可以通过从每个任务特定模型的响应中推断任务特征(预测不确定性)或者从增量任务中明确地学习任务特征预测功能来实现改进.

6 基于表示的方法

目前，对于基于表示的 LGL 方法存在 2 种不同的理解方式. 一种理解来自 Tian 等人[21]的观点，他们将基于表示的 LGL 方法解释为以隐式编码方式表达的形式. 另一种理解则来自 Wang 等人[26,89]的观点，他们将基于表示的 LGL 方法理解为通过创建和利用表达优势来持续学习. 尽管这 2 种观点被归类为基于表示的 LGL 方法.

6.1 隐式编码表示

随着新的增量图的引入，先前的图以及任务间的边存在相互依赖关系，这意味着先前任务中的知识可以被迁移到后续任务中. 然而，将这些知识显式整合到 LGL 中是一项挑战，因为这些知识通常是隐式编码在节点嵌入中的. Tian 等人[21]将这种方式归类为基于表示的方法. 其核心原理是，现有的节点嵌入已经包含了下游任务所需的所有必要信息，因此通过该方法得到的嵌入可以看作是弱化联合训练嵌入的一种形式. 基于隐式编码表示的方法通常可以概括为分离和传输 2 个主要方面.

6.1.1 嵌入分离

嵌入分离方法通常首先识别出图嵌入中的受影响组件和未受影响组件，然后对这些嵌入进行差异化处理，以便更新表示. 在嵌入分离的过程中，存在 2 个关键问题：1)分离准则，即确定要分离的最小结构组件；2)更新策略，即如何更新这些嵌入.

目前，分离准则主要包括以语义为准则和以原型为准则 2 种. 以语义为准则是指通过对图嵌入进行语义划分来实现受影响组件和未受影响组件的分离. 例如，DiCGRL[35]方法根据语义将图中的关系三元组解耦为多个独立的组件，并利用图嵌入方法学习分离开的图嵌入. 而以原型为准则是指以原型为基础结构来提取图中的知识. 例如，分层原型网络[90](hierarchical prototype networks, HPNs)方法提出通过原型的形式来提取不同层次的抽象知识，其中每个节点嵌入由 3 个层次的原型表示.

基于不同的分离准则，存在不同的更新策略. 更新策略指定了更新嵌入相关组件的方式. 在以语义为准则的方法中，DiCGRL[35] 采用了语义共享邻元素激活的更新策略. 即首先识别来自先前任务中需要更新的关系三元组，同时它们的邻元素也受到影响. 同时为了减少计算成本，每个三元组只激活具有相关语义组件的邻元素，然后使用现有的图嵌入方法来更新它们的特征. 之后在每个时期中，不断迭代训练新的关系三元组和激活邻元素的关系三元组的相关语义，由此具有共同语义的组件也得以更新. 通过该训练过程，DiCGRL[35] 不仅可以有效地防止灾难性遗忘，而且可以学习新数据的嵌入.

在基于原型的分离准则方法中，HPNS 采用了 3 种不同类型的原型来表示图嵌入，分别是原子级别的原型（A-prototypes）、节点级别的原型（N-prototypes）和类级别的原型（C-prototypes）. 原子级别的原型直接由原子嵌入集合进行细化或创建，用于表示节点的基本特征[90]. 节点级别的原型则是通过将原子级别的原型与每个节点关联来描述整个节点. 而类级别的原型是由节点级别的原型生成的，用于描述一组相似节点所共享的常见特征. 这 3 种级别的原型从原子级别到类级别，代表了不同尺度下子图的抽象知识，类似于 CNN 在不同层次上的特征图. 随着图数据的增加会提取原子嵌入，这些原子嵌入要么与现有的原子原型匹配，要么用于构建新的原子原型. 随后，将节点嵌入和类嵌入与现有的节点原型和类原型匹配，或者用于建立新的节点原型和类原型. 因此，当新的知识无法与现有原型匹配时，就会建立新的原型. 这种方法保证了模型能够随着新知识的增加而不断地演化和扩展，从而更好地适应不断变化的数据环境.

6.1.2 知识传递

知识传递方法是将先前积累的经验和知识传递给模型以帮助其学习新任务. 这些方法的关键在于所传递知识的类型以及传播方式.

1）知识类型. 除了来自当前任务数据集的知识外，其他来源的知识也可能对任务学习至关重要. 例如，任务相关性、时间信息和长期偏好等. 以 CML-BGNN[39] 为例，它将元学习视为连续的图学习任务，其中每个任务被建模为一个图，而来自查询集和支持集的每个样本则被建模为一个节点. 该方法保留了节点之间的任务内交互以及任务间的关联，以便将它们传递至下一个任务中. 对于 IGCN[91] 来说，时间信息被认为对协同过滤至关重要. 该方法保存了上一时期的时间信息，并将其与当前时期的时间信息进行增量融合，从而实现了高效准确的时间感知特征学习. 而对于 CIGC[29] 来说，长期偏好信号在推荐系统中具有重要意义，并通过聚合旧特征和新特征来保留这些信号.

2）传递方式. 除了知识类型，传递方式也是一个关键问题. 目前的研究主要通过循环神经网络（RNN）和 CNN 来传递知识. 例如，CML-BGNN[39] 采用门控循环单元将历史信息和任务信息传递给下一个任务，以捕获并传递长期任务相关性. 相反，ICGN[91] 使用 CNN 来进行知识传递，因为基于 RNN 的方法在更新用户-项目嵌入时存在灾难性遗忘问题. 另外，CIGC[39] 采用了简化的 CNN 设计来聚合和融合新旧表征，以应对灾难性遗忘.

其他研究还探索了不同的知识传递方法，包括元学习、动态记忆和遗忘抑制等. Fortunato 等人[92] 采用了元学习技术，通过利用先前任务中学习到的元知识（如优化策略和模型架构），来加速新任务的学习过程. 另外，Lopez-Pa 等人[24] 使用了动态记忆方法，通过构建动态记忆单元来存储和更新先前任务的信息以便在新任务中利用. Rebuffisa 等人[93] 使用了遗忘抑制（forgetting suppression）方法，通过采用特定的记忆管理策略来减少先前任务的遗忘，以防止新知识学习过程中的信息丢失.

6.2 表示优势利用

在 LGL 中，节点和边的表示方法的能力对于有效捕捉图结构中的信息至关重要. 这些表示方法的优势主要体现在它们如何有效地表达和捕捉图中的节点和边. 这种优势被归类为基于表示的方法，例如，Wang 等人[89] 将基于表示的方法划分为自我监督学习和预训练. 这些方法包括上游预训练和下游微调，它们需要终身学习以有效地利用这些表示优势，而预训练的表示则可以选择性地固定为学习特定的下游任务.

这些表示方法的优势不仅仅在于能够捕获图中节点和边的结构关系、节点属性的特征表达，还在于它们具备适应图中动态变化的能力. 举例来说，表示方法能够全面地捕捉图中节点和边的结构、特征和关系，从而准确地反映节点之间的相似性、连接性和潜在的语义信息，更好地描述图的拓扑结构和属性. 此外，这些表示方法还具有一定的动态性和灵活性，能够在学习过程中动态地更新和调整表示，以适应新的数据和任务的变化.

在早期的研究中，Javed 等人[94] 提出了从元训练

中获得稀疏表示的方法. 然而，在最近的研究中，Yang 等人[15]尝试结合自监督学习(self-supervised learning, SSL)和大规模预训练[46]的优势改进初始化和终身学习中的表示. 这么做的原因在于，自监督学习和大规模预训练在数据丰富性、提高模型的迁移、泛化能力以及增强鲁棒性等方面表现出良好的性能. 通过利用大规模无标签数据学习图数据的表示，自监督学习和大规模预训练使得表示更加丰富和全面，能够更好地泛化到新的任务上，并具有更强的抵抗力以对抗数据中的噪声和扰动. 需要注意的是，大规模预训练的数据中没有明确的标签，而 SSL 的性能主要通过对一系列下游任务进行微调来评估.

6.2.1 自监督学习

自监督学习，又称无监督学习(unsupervised learning, UL)，是一种重要的学习范式，其核心思想是利用数据自身的结构或属性来生成标签，从而实现模型的学习. 在 LGL 中，自监督学习通常伴随着对比损失的使用，它是基于表示方法中的一个关键子方法. 自监督学习的主要目标是通过数据本身的特征来训练模型，而无需依赖外部标签. 这种方法在终身学习中具有重要意义，因为它可以利用大规模无标签数据进行模型的预训练，从而提高模型的泛化能力和鲁棒性.

Madaan 等人[47]对无监督终身学习(unsupervised continual learning, UCL)进行了深入研究，他们通过分析所学习到的特征表示，发现相比有监督视觉表示，无监督视觉表示对灾难性遗忘具有惊人的鲁棒性，并始终表现出更好的性能. 此外，他们还观察到无监督终身学习比有监督的终身学习(supervised continual learning, SCL)更能够泛化到分布外任务. 鉴此，Madaan 等人[47]提出了终身无监督混合(lifelong unsupervised mixup, LUMP)技术，该技术能够在新旧任务的实例之间进行插值，从而减轻无监督表示的灾难性遗忘现象.

Purushwalkam 等人[48]提出了使用重放缓冲来缓解低效率和时间相关性的问题，并进一步提出了一种新方法，即最小冗余度(MinRed)缓冲. 这种方法能够在最具挑战性的流场景中学习到有效的表示，这些场景由从单个实体代理获得的顺序视觉数据组成. MinRed 缓冲通过保持最少冗余样本，同时去除存储的旧训练样本的相关性，进一步促进了经验回放的多样性，有效缓解了从具有非平稳语义分布的数据中学习的灾难性遗忘问题.

Cha 等人[95]提出了 Co2L(contrastive continual learning)终身学习算法，该算法专注于终身学习和维护可转换的表示，采用对比学习目标来学习这些表示，并通过自监督蒸馏步骤来保留学习到的表示. 经过实验证明，提出的不对称形式的对比学习损失以及实例关系蒸馏有助于模型学习并保留新的和旧的表示.

6.2.2 预训练

预训练作为基于表示的方法中的重要子方法，在下游终身学习中发挥着关键作用. 通过预训练，不仅可以实现强大的知识迁移，还能够使得终身学习在面对灾难性遗忘时更加鲁棒，特别是当使用更大规模的数据、更庞大的模型以及对比损失进行预训练时，其好处更为显著. 然而，关键问题在于如何自适应地利用预训练的知识于当前任务，并同时保持对未来任务的泛化能力. 针对这一问题，已经提出了多种策略，策略的采用具体取决于预训练主干是否固定.

Zhang 等人[96]提出了“side-tuning”网络自适应方法，该方法通过训练一个轻量级的“Side”网络并行于主干，利用加法融合将其与预训练的网络相结合，实现了对预训练网络的自适应. 这种方法不易过拟合，能够很好地解决增量学习中的灾难性遗忘问题. Shon 等人[97]提出了一种基于线性化和均方误差损失的连续微调框架和深度线性连续微调(deep linear continual fine-turning, DLCFT)方法，该方法与主干并行，不断地从预训练的表示中微调模型，并利用预训练神经网络的线性化技术进行简单有效的终身学习. Boschini 等人[98]提出的无遗忘迁移(TwF)混合连续迁移学习方法则是通过一个固定的预训练兄弟网络来实现知识传递，并通过基于层次的损失项持续传播主干中的知识.

另一方面，持续的预训练(CPT)也是一种重要策略. 由于预训练所需的大量数据通常是以增量方式收集的，因此需要进行上游终身学习以提高下游性能. Cossua 等人[99]的研究发现，在视觉领域中，使用监督图像分类进行预训练的模型(如 ResNet 和 ViT)在面对新任务时会出现较大的准确率下降. 相比之下，使用自监督图像建模进行预训练的模型(如 BEiT)在面对新任务时几乎不会出现遗忘，但需要进行一次微调以恢复到原始性能. 同时，持续的预训练还需要解决一个问题：基类的预训练知识是逐步丰富和适应的. Liu 等人[64]在元学习的元训练阶段进行增量，并提出了一种可以与多种现有的基于度量的元学习算法一起使用的方法，使得新旧模型进行区分、相对于旧中心进行对齐，并让嵌入自由地适应新任务.

6.2.3 持续的性能改进

这些方法的核心概念在于通过不断更新表示来增强模型的知识，从而在理论上保证性能的持续提升. 其中，基于原型理论的方法是一个典型的例子[13,100]. 这些方法的学习过程基于原型，通过对现有原型的组合表示来适应新任务的学习，如图 7 所示.

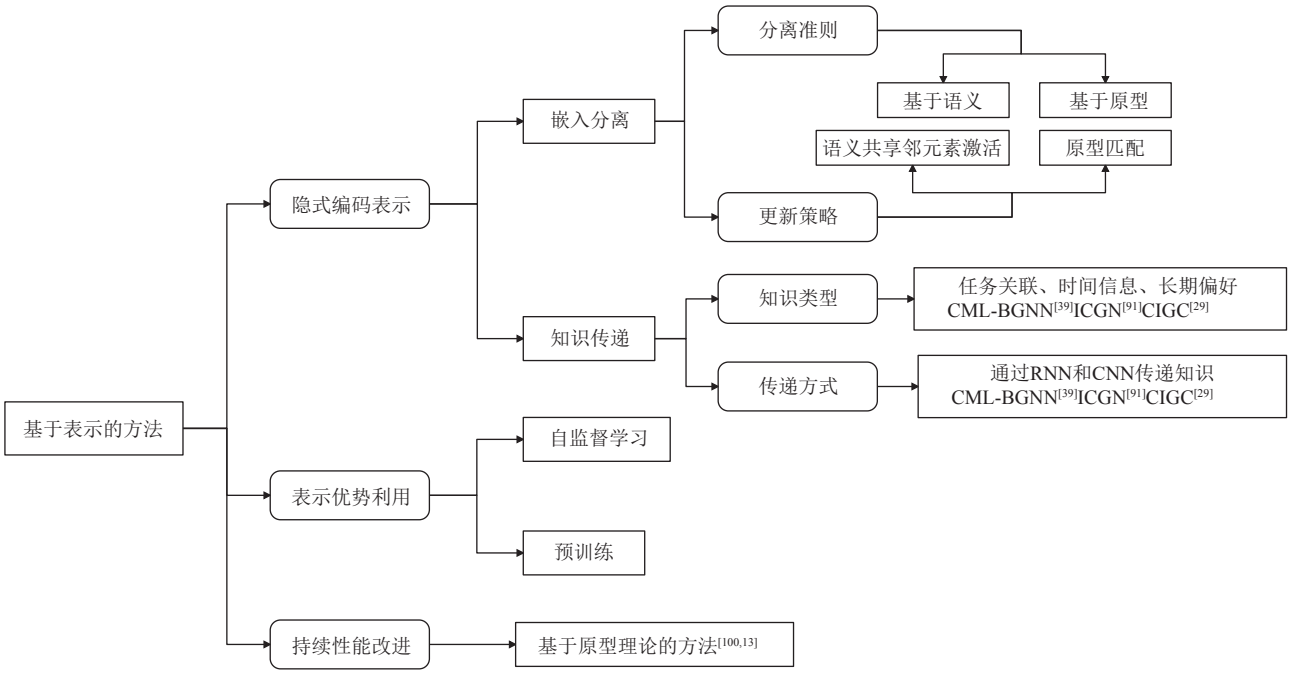


Fig. 7 Representation-based lifelong graph learning methods

图 7 基于表示的 LGL 方法

7 基于预演的方法

在再训练过程中，引入预演方法可以有效地激活与先前任务相关的神经网络和记忆路径. 通过重复暴露于这些模拟场景，个体或系统能够逐渐巩固对任务的记忆，并提高在未来遇到类似任务时的应对能力. 基于预演的 LGL 方法可以选择适当的节点和边表示样本，仔细考虑了样本的数量，以最小化计算的复杂度. 最近，人们提出了新的基于预演的 LGL 方法，如经验节点回放的 GNN 模型（ER-GNN）[32] 和开放世界节点分类（open-world node classification）[44]. 图 8 详细展示了基于预演的重要方法.

7.1 经验节点回放的 GNN 模型

Zhou 等人[33] 基于经验既然点回放的 LGL 框架，提出了 ER-GNN. 该模型将节点分类任务分为多个子任务，利用终身学习的特点针对具体任务提高其性能. 模型设立了经验缓冲池，用来存放前期子任务训

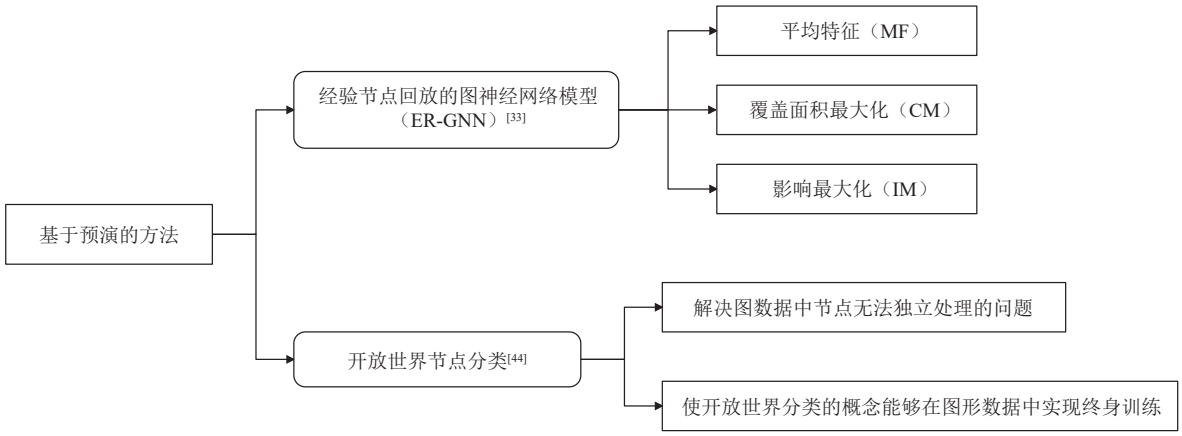


Fig. 8 Rehearsal-based lifelong graph learning methods

图 8 基于预演的 LGL 方法

练得到的经验节点. 在处理新的子任务时, ER-GNN同时利用新的训练集和从缓冲池中提取的经验节点进行训练, 以吸收先前任务的经验. 为了避免终身学习中常见的灾难性遗忘问题, 该研究提出了3种经验节点选择策略: 平均特征(MF)[93]、覆盖面积最大化(CM)[101]和影响最大化(IM).

ER-GNN的提出主要受到生物领域"互补学习系统"(complementary learning systems, CLS)的启发, 这个系统在人类大脑中扮演着重要角色. CLS展示了海马体与新皮质之间在虚拟体验机制上的协同作用, 是人类学习和记忆的一个典型示例. 在学习过程中, 当知识需要转移到长期记忆之前, 新皮质系统通过回放海马体中的记忆来巩固知识, 这种过程有助于加深对学习内容的理解和记忆. 这种生物学上的启示激发了ER-GNN模型的设计, 试图模拟人类大脑中这种记忆巩固机制, 以实现在图数据学习中的类似效果. 以节点分类任务为例, ER-GNN将该任务分为多个子任务 $T=\{T_1,T_2,\cdots,T_i,\cdots,T_M\}$, 同时对应地将数据集也进行划分. 每次子任务$T_i$有对应的子训练集$D_i^{\mathrm{tr}}$和测试集$D_i^{\mathrm{te}}$. T_i使用训练集D_i^{tr}的资源来使测试集D_i^{te}的性能最大化. 该模型还设立了经验节点缓冲池B, 对一个子任务D_i而言, 需要训练优化的节点除其对应的训练集D_i^{tr}外, 还有从缓冲池B中挑选出的经验节点. 之后将训练集D_i^{tr}和经验节点B一起提供给分类器f_θ. 节点分类任务的自然损失函数选择是交叉熵损失函数:

$$\mathcal{L}_{T_i}(f_\theta,D)=-\left(\sum_{(x_i,y_i)\in D}(y_i\log f_\theta(x_i)+(1-y_i)\log(1-f_\theta(x_i)))\right).$$

在引入经验节点时, 需要设置一个权重因子b来平衡训练集和缓冲池, 以避免模型偏向于某些具有特点的节点集, 即$\beta=|B|/(|D_i^{\mathrm{tr}}|+|B|)$, 引入经验节点后损失函数变为:

$$\mathcal{L}'_{T_i}\left(f_\theta,D_i^{\mathrm{tr}},B\right)=\beta\mathcal{L}_{T_i}\left(f_\theta,D_i^{\mathrm{tr}}\right)+(1-\beta)\mathcal{L}_{T_i}\left(f_\theta,D_i^{\mathrm{tr}}\right).$$

ER-GNN使用改进后的交叉熵损失函数进行节点分类, 通过使用Adam优化器[55]等优化方法来最小化损失, 以此实现参数更新并且获得最优参数. 在更新参数后, 模型在当前的训练数据集D_i^{tr}执行节点选择, 将经验存储在缓冲区中. 文献[32]提出了3种策略来选择节点, 分别是: 平均特征(mean of feature, MF)、覆盖面积最大化(coverage maximization, CM)和新方法影响最大化(influence maximization, IM), 以此提高图连续学习的性能.

为了评估ER-GNN的性能, 文献[32]采用了一个节点分类任务, 其中模型基于额外的新节点进行了建模. 该研究使用了一系列数据集, 包括Cora[55], Citeseer[49]和Reddit[50]等, 以及深度行走[51]、Node2Vec[52]和GCN[53]等图学习模型. 这些模型与微调策略进行了比较, 这意味着虽然模型经过了持续的训练, 但未使用终身学习方法. 实验结果显示, ER-GNN在Cora数据集上表现最佳, 其平均精度达到了95.66%. 与采用微调策略的图学习模型相比, ER-GNN的平均遗忘率最低, 为17.08%. 这些实验结果表明, ER-GNN相较于一系列先进的节点分类任务的GNN, 在减少遗忘率方面性能有所提升.

7.2 开放世界节点分类

Kirkpatrick等人[60]提出了开放世界节点分类的概念, 旨在解决图数据中节点处理的复杂性. 该模型利用先前学习到的知识来辅助当前任务的学习, 这些知识可以显式地存储为历史数据, 也可以隐式地存储在模型参数中. 随后, 分析存储的知识对网络模型优化的影响. 该模型的首要目标是处理图数据中节点间复杂关系的挑战, 因为节点之间的交互需要适当的嵌入机制来表达和传递信息. 其次, 该模型旨在将开放世界分类的理念应用于图数据, 并实现终身学习, 使其能够适应和以前完全不同的新任务. 为了实现这一目标, 引入了一种创新的度量方法, 旨在调节具有不同特征的图数据的演化过程. 该方法具备整合GNN中各向同性和各向异性特性的能力, 从而使模型能够更好地适应不同类型的图数据.

该模型的问题定义如下: 假设有限数量的图数据处理任务为$T_1,T_2,\cdots,T_t$, 一个参数为θ的模型$\mathcal{F}$. 在该模型中进行增量训练, 图的节点和边的结构是随时间变化而变化的, 在准备下一个任务T_{t+1}之前, 模型会基于当前任务T_t的标签进行训练. 通过这个过程, 模型能够学习到与任务T_t相关的知识, 并将这些知识编码为参数θ_t. 当l个新类出现时, 模型将相应数量的参数添加到f_t的输出层中. 因此, 模型将有一个新的参数化层:

$$\begin{cases}\left|\theta^t_{\mathrm{outputweight}}\right|=\left|\theta^{t-1}_{\mathrm{outputweight}}\right|+l,\text{and}\left|\theta^t_{\mathrm{outputbias}}\right|=\\ \qquad\left|\theta^{t-1}_{\mathrm{outputbias}}\right|+l,\\ \left|\theta^t_{\mathrm{outputweight}}\right|=\left|\theta^{t-1}_{\mathrm{outputweight}}\right|+l,\text{and}\left|\theta^t_{\mathrm{outputbias}}\right|=\\ \qquad\left|\theta^{t-1}_{\mathrm{outputbias}}\right|+l.\end{cases}$$

在模型进入下一步骤时, 为了实现预演模式的有效启用, 提供了2种训练选择: 一是从头开始重新训练模型(冷重启), 二是利用之前任务中的最终模型参数(热重启). 为了避免全历史重新训练的复杂性, 该模型引入了人工历史大小的概念, 用于确定模

型在训练迭代中使用的历史数据量.

考虑到 LGL 中预先编制数据集的稀缺性，该模型进行了在科学出版物数据集 PharmaBio 和 DBLP 上的实验验证[102]. 这个实验的核心是通过引入新类来模拟和适应新任务. 在实验中，使用了平均准确度和前向变换 2 种指标来评估使用先前参数的效果. 同时，还运用了多种 GNN 技术进行表示学习，例如 GAT, GCN 和 GraphSage 等. 实验结果显示，使用有限历史数据进行的终身学习与使用完整历史数据进行的训练相比表现几乎相当. 尤其是，在采用了 GNN 技术的模型中，至少达到了 95% 的准确率，这表明在再训练过程中，只需使用数据集的一小部分即可覆盖全范围.

8 混合方法

在本节中，我们深入探讨一种融合多种终身学习技术的混合方法[34,31-32,61-62]. 该方法旨在集合各种技术的优势，以此激发模型的最大潜能，进而显著提升其性能表现. 具体而言，我们将重点讨论 2 种 LGL 的混合方法：ContinualGNN[32] 和 TrafficStream[31]. 图 9 为基于混合的图终身学习的分类.

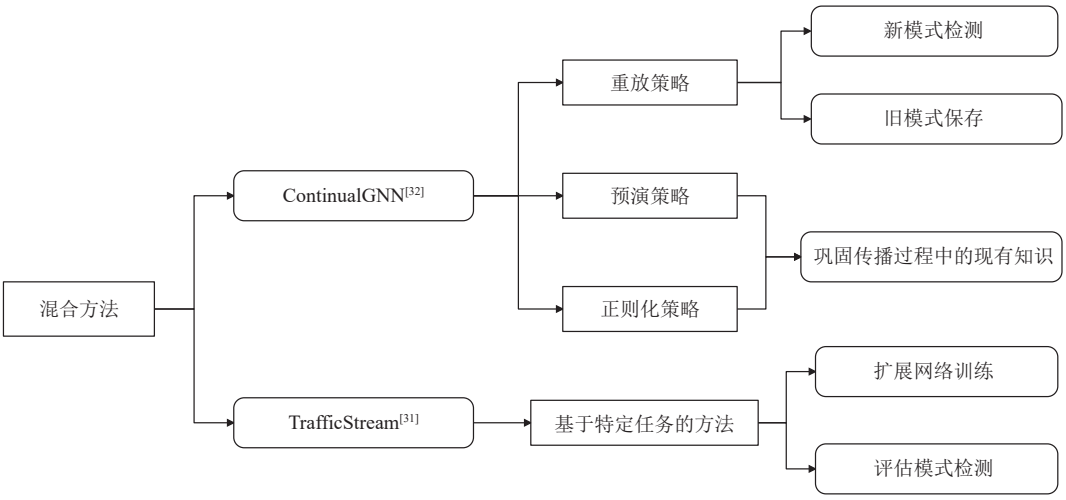


Fig. 9 Hybrid lifelong graph learning method

图 9 混合 LGL 方法

在 ContinualGNN 方法中，研究人员采用了一种整合了经验重放和记忆增强的策略，以有效地处理终身学习过程中的挑战. 该方法通过综合利用经验回放和记忆增强，实现了模型性能的持续提升和知识的稳固积累. 而 TrafficStream 方法则着眼于交通网络领域，通过融合多种终身学习技术，包括重放缓冲、元学习和动态记忆等，以应对交通流量数据中的时空动态性和复杂性. 通过将这些技术相互结合，TrafficStream 方法能够在不同任务和场景中实现更好的性能表现，同时提高模型的鲁棒性和适应性. 这 2 种混合方法的提出，为解决 LGL 中的挑战提供了新的思路和方法. 它们的成功应用表明，通过充分利用不同终身学习技术的优势，并将它们融合在一起，可以实现更强大、更灵活的模型，为复杂任务和领域带来更有效的解决方案.

8.1 ContinualGNN

在 Wang 等人[32] 在提出的 ContinualGNN 方法中构建了一个实际场景模型，该模型包含一个完善的网络重放策略作为正则化方法的补充. ContinualGNN 将 2 种方法结合起来，以减轻灾难性遗忘并保持现有的学习模式. 该模型的主要目标有 2 个：首先是检测图结构中的新模式，这些模式可能会对网络中的所有节点产生显著影响；其次是将这些知识整合到整个网络中. 由于 ContinualGNN 模型能够捕获流图数据中的新模式，因此它提出了一种基于传播过程的方法，即在学习新模式时可以有效地挖掘受影响节点上的信息，然后利用预演和正则化这 2 种方法的组合来巩固传播过程中已有的知识. 接下来，我们将详细讨论 ContinualGNN 如何将重播和正则化方法相结合，以缓解灾难性遗忘问题.

ContinualGNN 从 2 个方面设计了重放策略：新模式检测和旧模式保存. 在新模式检测方面，模型对容易被新出现的节点扰动的旧节点的子集进行采样（称为“扰动传播”）. 节点 i 的扰动被量化为新数据收入后的变化. 接收到的扰动超过阈值的节点应该会产生新的模式. 在旧模式保留方面，该模型采用分层抽

样策略和基于重要性的抽样策略来生成记忆. 前者只选择一些图数据的节点和相邻节点的重要信息进行保存和重放，保证了内存和整个数据集之间类别分布的一致性. 后者选择的节点包含来自其他类别的大量邻居，这些节点位于类边界处，应该对梯度做出更多贡献. 这 2 种采样方法都嵌入到储层采样算法中以适应数据流. 该模型通过重放内存中的少量数据来解决过拟合现象，这一过程与基于正则化的方法高度相关，该方法使用源自一般终身学习算法 EWC[19] 的知识保存泛化，保持当前模型参数与先前模型参数之间的距离有助于保持 GNN 中先前任务的知识和性能. 此外，ContinualGNN 主要使用均值聚合器在 GraphSAGE 上进行实验. 随着扩展到其他 GNN，新的模式检测模块需要相应地进行修改.

8.2 TrafficStream

基于特定任务的方法是为了在特定的任务领域实现终身学习而设计的. 这种方法通常专注于解决特定任务，并且旨在在期望的领域中取得良好的性能. 然而，这类方法的局限性在于对特定任务的依赖性，因此不能保证适用于其他领域. 在终身学习中，存在一些特定于任务的方法的例子，比如永无止境的语言学习者(NELL)[83] 和终身情感分类(LSC)[100]. 在图数据领域，一个典型的特定任务方法的例子就是 TrafficStream.

Chen 等人[31] 提出了一个名为 TrafficStream 的流式交通流(streaming traffic flow)预测框架以预测数据结构在其节点和数据结构中不断演变的特定交通流量. 为预测不断变化的交通流量，最简单的方法是在特定的时间(每周、每月或每年)从头开始重新训练模型. 但按顺序重新训练模型，将会合并大量数据，是昂贵且低效的. 此外，由于网络的扩展，以前的模型的拓扑和架构会有不同的模式，以往模型的空间依赖性将不再相关. 因此，我们需要巩固和保留从以前学习到的知识中获得的洞察力，帮助模型提高性能. 基于这些挑战，Yu 等人[59] 提出的 TrafficStream 框架快速且有效地捕获新的图结构的模式，并整合旧图结构和新图结构之间的知识. 该框架使用了简单的交通流预测模型 Sur-model 来表示智能交通系统中复杂的图数据模型. 为有效更新当前网络模式，使用在 TrafficStream 方法中的 2 个组成部分：扩展网络训练和评估模式检测. 扩展训练过程的主要功能是降低捕获图中节点之间依赖关系的复杂性. 由于大量的属性和节点数量增加了 GNN 层的处理任务，TrafficStream 引入了通过将包含的节点数量减少到节点中仅 2 跳相邻节点的方法来生成子图的模型，有效地提高了 GNN 的处理速度. 进化模式检测的目的是在一个完全不同于前一个图的图上检测不同的模式，利于识别兴趣点(POI)的转移，以便模型可以更有针对性地选择有显著变化的节点进行学习.

使用 TrafficStream 这种终身学习方法的本质是为了避免灾难性遗忘，该方法通过回顾先前的数据并应用信息重放和参数平滑 2 种策略来实现这一目标. 信息重放策略利用先前信息中的样本数据来进行训练. 在样本数据的选择方面，一方面采用一种简单的方法，即从多个节点中随机选择. 具体而言，该模型使用 JS 散度(JDS)算法，选择 JDS 得分较低的节点来构建子图. 另一方面，参数平滑策略旨在维护先前的信息以防止灾难性遗忘，该方法借鉴了 EWC(elastic weight consolidation)的思想[36] 来执行参数平滑. 通过这种机制，模型中那些具有较少基本参数的权重将会受到更小的影响，使其能够适应新的模式；而那些被认为重要的参数在权重平滑过程中将获得更高的权重，从而限制了参数的更改，以保留先前的知识.

9 应 用

LGL 是一种强大的学习范式，其潜在的应用涵盖了各个领域. 通过在不同时间点持续学习和适应新的数据、任务和环境，LGL 能够为许多现实世界的场景提供解决方案. 从社交网络分析到医疗健康领域，再到智能交通系统，LGL 都具有广泛的应用前景. 在本节中，我们将深入探讨 LGL 在不同领域的应用，并分析其可能带来的影响和潜在价值.

9.1 交通预测

交通预测在现代交通系统中扮演着至关重要的角色，直接影响着人们的日常生活. 其挑战在于，交通网络和流量模式会随着时间的推移而不断演化和变化，而且还会受到突发事件的影响. LGL 为预测动态图网络上的交通流量提供了巨大的潜力.

目前已有关于交通预测的 LGL 研究，其中一项研究[31] 通过回放新图的邻节点和不稳定节点来适应新的模式，并通过回放稳定节点来巩固已有的模式，同时采用了 EWC 正则化. 尽管该研究取得了一定的成果，但忽略了样本级别的依赖性和时空依赖性. 为了保持样本级别的依赖性，一个可行的解决方案是显式地编码节点相似性并进行蒸馏或参数平滑. 此外，提出可以使用 JSD 来计算节点之间的相似度. 另外，时间上的依赖性可能会导致相似性的变化，因此

需要考虑相似性的稳定性，确保其不会过时. 因此，对于稳定的节点，其相似性也应该是稳定的.

除了这些挑战，交通预测还需要考虑到交通网络的拓扑结构、交通流量的周期性和季节性变化，以及各种交通事件的影响. 未来的研究需要更深入地探索如何利用 LGL 方法来解决这些复杂的交通预测问题，以实现更准确和可靠的预测能力，从而为交通管理和规划提供更好的支持.

9.2 推荐系统

推荐系统领域涉及到多种图结构，如用户-项目二部图、用户行为序列图、用户社交关系图、产品共同购买图和项目知识图等. GNN 在推荐系统中得到了广泛的应用，其优势在于能够处理复杂的图结构，并通过节点聚合来有效地编码多跳关系的拓扑结构，同时结合来自社交关系或知识图等的额外信息，以提高了用户和物品的表示能力. 改进后的用户表示被广泛应用于推荐系统中的各种任务，包括用户-项目协同过滤、顺序推荐、社交推荐以及基于知识图的推荐等. 此外，GNN 还在推荐系统中的其他任务上发挥着重要作用[40,103-106]，如点击率（CTR）预测、兴趣点推荐、群体推荐和捆绑推荐等.

持续学习对于推荐系统尤为重要，特别是对于电子商务推荐系统而言. 这些系统处于不断变化的环境中，用户的偏好和物品的特征可能会随着时间的推移而发生变化. 因此，推荐系统中的持续图学习主要集中在 Top-K 推荐任务[8,26-30,41-43,63,65-66,91,104,107-119]上，即不断更新和维护用户和物品的表示，以适应环境和用户需求的变化. 通过持续学习，推荐系统能够更好地适应不断变化的市场和用户行为，从而提供更准确、更个性化的推荐服务.

9.3 生物信息学

生物信息学领域涉及到分子和化合物的研究，这些分子和化合物可以自然地表示为以原子为节点，化学键为边. 这种结构特点使得 GNN 在生物信息学领域的应用成为可能，涵盖了诸如药物发现、性质/结构预测等多个方面.

在药物发现中，常常观察到具有相似药理活性的药物往往具有相似的分子结构. 基于 GNN 的分子表示学习和分子图生成等方法，能够对分子结构的内部进行考虑，有效地预测分子的化学性质和活性，从而加速了药物发现的过程. 此外，基于 GNN 的模型也被广泛应用于医学诊断领域[107]. 然而，在生物化学和生物医学方面的持续学习仍然鲜有研究，具有持续学习能力的模型对于生物信息学的进步具有重要意义. 例如，一个药物发现模型可以不断地发现新的靶向分子，从而推动新药物的开发；而一个疾病诊断模型则可以通过不断地学习来识别新的疾病或病毒，提高医学诊断的准确性和及时性. 因此，持续的图结构学习为生物信息学研究提供了一个充满前景的方向.

9.4 金融学习

在复杂的金融系统中，各种组件之间错综复杂的关系使得图成为一种被广泛用于表示这种复杂关系数据的数据结构. 根据研究结果[108]表明，金融体系中的组成部分和关系通常会发生演变，财务图也随之改变，这使得 LGL 成为该领域的一个潜在发展方向. 然而，现有的图和数据属性给应用 LGL 方面带来了挑战. 图的异构性是普遍存在的，包括账户设备图、异构信息网络和金融知识图. 尽管一些 LGL 工作已经研究了异构图的存在[26-30]，但它们几乎没有探索来自不同类型节点之间的相互作用.

根据目前的研究成果，异构 GNN 已经在多个领域得到了广泛应用，例如在支付宝金融风控中的恶意账号检测、供应链金融中的欺诈检测以及蚂蚁金服的套现欺诈检测等方面. 这些应用充分展示了异构 GNN 在处理具有不同属性和权重的边时的有效性. 此外，多层网络的研究进展表明，对异构性的刻画，包括对不同类型节点以及属于不同网络层节点之间相互作用模式的描述，对于理解复杂系统的结构至关重要. 在金融领域，研究金融机构关联网络、风险溢出与系统性风险，也强调了不同类型金融机构之间关联关系对系统性风险的影响. 这进一步证明了在金融领域中，考虑图的异构性和复杂性是至关重要的. 金融知识图谱的构建与应用，以及基于区块链的跨域异构应用间可信数据交换及逻辑执行体系的研究，都突显了在金融领域中利用图和数据结构来处理和分析复杂信息的重要性.

尽管面临挑战，但通过识别图特征并提出新的 GNN 模型来克服同构图和异构图之间的障碍，以及通过构建银行和企业间的多重金融关联网络模型来分析不同金融关联下的银行和企业间金融网络拓扑结构特征[7]，我们可以看到在金融学习领域中，利用图和数据结构来理解和分析复杂金融系统是有前景的. 因此，尽管现有的图和数据属性在应用 LGL 方面带来了挑战，但通过不断的研究和技术创新，这些挑战是可以被克服的.

9.5 计算机视觉

计算机视觉（computer vision, CV）是一个研究如

何使计算机能够“看”和理解图像的领域. 近年来，随着深度学习技术的发展，CNN 成为了处理 CV 任务的主要工具之一. 然而，除了 CNN 之外，GNN 也在 CV 领域显示出了其独特的价值和潜力，最近的一些工作也显示了 GNN 在计算机视觉任务中的可行性，如 ML-GCN[50]，Curve-GCN[41] 和 Vision GNN[42] 等.

GNN 通过将图像中的不同对象或特征表示为节点，并利用节点之间的连接关系构建图形结构，从而在图像分类、目标检测、图像生成等方面取得了显著的成果. 这种基于图的方法能够更好地捕捉图像中对象之间的语义关系，从而提高模型的性能. 例如，在场景图生成中，使用 GNN 识别物体之间的语义关系有助于理解视觉场景背后的信息. GNN 的应用不仅限于传统的 2D 自然图像处理，还扩展到了视频、3D 数据、多模态(视觉和语言)以及医学图像等领域. 这表明 GNN 具有很强的泛化能力，能够在多种不同的数据类型和应用场景中发挥作用.

此外，GNN 在持续学习方面的应用也逐渐受到关注. 尽管目前大多数持续学习的工作集中在视觉分类任务上，但 GNN 在 CV 中的应用也为其他视觉领域提供了新的可能性. 特别是在图像中构造图以保存结构知识进行持续学习的方法，如范例关系图和记忆知识图，展示了 GNN 在保持长期学习能力方面的潜力.

总之，GNN 在 CV 领域的应用展现了其强大的能力和广泛的适用性. 通过构建图像的图表示并利用节点间的拓扑结构，GNN 能够有效地捕捉图像中的复杂信息和对象间的关系，从而在多个 CV 任务中取得优异的性能，例如，最近的一些工作通过建模特征[119]、范例[8] 和任务[39] 之间的拓扑结构来构造图. 然而，LGL 在 CV 中的应用仍处于起步阶段. CV 中现有的 LGL 工作通常从图像中构造图，保存结构知识以进行持续学习，如范例关系图[8,29,77] 和记忆知识图[43]. 未来，随着研究的深入和技术的进步，LGL 在 CV 领域的应用将会变得更加广泛和深入，我们相信未来这一领域将更加关注持续的图学习.

10 待解决的问题

从本质上来说，目前大多数连续图学习方法都在可塑性和稳定性之间做出妥协和权衡. 虽然灾难性遗忘问题已经得到有效解决，但对于持续性能改进的系统性研究仍然不足，尽管目前的努力(例如文献 [32])已经取得了一些积极的向后转移结果. 然而，正向后转移仅仅意味着 $BWT=\frac{1}{Y-1}\sum_{j=1}^{T-1}R_{T,j}-R_j$，$j>0$，而持续的性能改进要求 $R_{T,j}>R_{T-1,j}$ 一直成立，这一要求比正向后转移更为严格. 为了实现持续的绩效改进，我们概述了一些可能影响实现这一目标的考虑因素，并提出了一些未解决的问题和相应的未来方向.

10.1 LGL 的收敛性

在讨论连续性能时，模型何时收敛是一个不可避免的问题. 而知识增强和优化控制则为实现持续的绩效改进提供了路线图，实现这一目标可能需要施加更严格的条件，例如拥有足够的模型容量. 从理论上讲，具有固定架构的神经网络在容量上是有限的. LGL 在本质上通过学习为所有任务找到一个共享的参数解决方案，以及查找最优解的复杂性主要取决于参数空间的结构. 在 LGL 的设置中，学习新的任务相当于在有限的、不断缩小的参数空间中学习新任务的同时仍然要克服灾难性遗忘. 然而，事实证明，这个问题通常是非确定性多项式(nondeterministic polynomially，NP)问题[106]，因为随着更多任务的引入，可行的参数空间往往是不规则的，并且可行参数空间的范围将逐渐变窄. 最终，模型将难以避免地陷入无法学习新任务的情况.

实际上，我们可以通过扩展网络架构来扩展学习任务的数量. 这种方法可以不在任务间共享参数，而是分配特定于任务的参数，并实现摆脱模型容量的限制. 通过这种方式，我们也可能会优化持续学习的性能. 然而，这种方式可能会受到联合训练的限制，该联合训练用于联合优化所有任务的 LGL 性能的上限. 理想情况下，模型在性能达到或接近此上限时会收敛. 然而，问题在于性能达到这个上限需要多长时间以及在达到上限后可以持续多长时间.

未来的趋同方向可能包括 2 个方面. 一是性能如何尽快达到上限，以及模型如何能够长时间保持最佳性能. 另一个方向是如何尽可能多地扩展模型学习的任务数量.

10.2 可扩展性

随着任务数量的增加，持续性能改进也面临着更大的挑战. LGL 模型旨在应对多样的任务，但随着任务的增加，可行参数空间会迅速缩小，这使得为所有任务找到共享解决方案变得困难. 因此，实现连续性的性能改进随着任务数量的增加而变得更加具有挑战性. 一种直观的方法是增加每个任务的数据量，以此减少任务的数量.

然而，现实世界中的图通常是大规模且高度复杂的. 例如，一个社交网络可能包含数千万甚至数十亿个节点和边. 尽管 LGL 将图划分为多个任务并减少每次训练的内存需求，但对于 LGL 来说，这仍然是一个巨大的挑战. 正如前文所述，随着任务数量的增加，可行的参数空间往往会迅速缩小，并且为所有任务找到一个共享解决方案是一个 NP 难题. 因此，在任务数量和数据量的选择上，我们面临着一个两难的境地.

减小图的规模的典型方法包括邻居采样（如 GraphSAGE[50] 中所示）或随机游走抽样（如 PinSage[120] 中所示）. 然而，采样过程不可避免地会导致信息丢失，进而降低模型性能. 另一种选择是设计和开发能够有效处理大型图的图算法，例如简化的 GCN[108-109]. 尽管这些简化模型因为其简洁性而更具有可扩展性，但它们在聚合器或更新选择方面存在一定的限制. 因此，为了应对任务数量的增加，未来我们需要继续研究并发展更加高效的 LGL 方法，以便在大规模和高度复杂的图上实现持续性能改进.

10.3 无监督的 LGL

标签稀疏性同样对持续性能改进产生影响. 在现实中，图上的大多数节点都是未标记的，且对 GNN 的大多数研究都是半监督的. 然而，半监督图学习严重依赖标签，导致不良泛化和弱鲁棒性[112]. 一种很有前途的方法是无监督图学习，尤其是自监督图学习. 自监督图学习可以通过各种方法增强图，如对比学习[63] 或自生成标签[113] 来学习更好的图表示. 在 LGL 环境中，自监督图学习可以利用额外的知识，就像互信息一样，这可能会提高性能. 此外，更好的图表示还可以提高 GNN 的泛化能力，从而降低在参数空间发现共享解决方案的难度.

10.4 可解释性

在持续性能改进的背景下，可解释性的重要性不言而喻. 良好的可解释性不仅有助于我们理解图模型中哪些知识对于学习任务至关重要，而且还能帮助图模型保留关键知识，从而有效克服灾难性遗忘并实现持续的性能改进. 目前，关于 GNN 的可解释性[114-115] 和持续学习[116] 方面的研究已经取得了一些进展.

GNN 的解释方法主要分为实例级和模型级 2 种类型. 实例级方法通过针对每个输入图提供与其预测相关的解释来识别重要特征，例如基于梯度/特征的方法、基于扰动的方法、分解方法和代理方法等[114-115]. 这些方法通过监测不同扰动的预测变化及使用梯度或隐藏特征图来表示不同输入特征的重要性，从而为每个输入图提供依赖于输入的解释. 模型级方法则提供更高层次的见解和一般性理解，专门研究哪些输入图模式可以导致 GNN 进行某种预测[12]. 例如，XGNN 方法[121] 通过训练一个图生成器来解释 GNN，使生成的图模式最大化模型的某种预测，从而为 GNN 的工作方式提供高层次的见解和一般性的理解. PAGE 方法[38] 则是通过发现人类可解释的原型图来解释底层 GNN 模型为图分类学到了什么，为给定的类生成解释，因此能够提供比实例级更简洁和全面的解释[115].

尽管在 GNN 的可解释性方面已经有了一些进展，但持续图学习的可解释性仍然是一个待解决的问题. 这需要进一步地研究和探索，以开发出更加高效和有效的可解释性方法，从而更好地理解和改进 GNN 在持续学习中的表现. 此外，考虑到持续学习算法必须在保留旧知识与学习新知识之间取得平衡，可解释性方法的发展也将对如何实现这一平衡提供重要的指导和支持.

10.5 数据集的稀缺性

尽管图学习已经被广泛应用于各种领域，但是针对终身学习的图数据集却相对稀缺. 过去的工作提供了一些图数据集用于基于终身学习的研究. 这些数据集通常是基于一般图学习数据集构建的，其中包括一些被广泛使用的数据集. 最近也出现了一些针对终身学习而建立的数据集，但它们仍然无法涵盖当前图持续学习所需的全部范围.

当前用于 LGL 研究的数据集在各个领域中都有应用. 这些数据集涵盖了各种类型的图，包括社交网络、生物网络、交通网络等. 它们被用于许多不同的应用场景，例如节点分类、链接预测、图生成等. 然而，尽管这些数据集已经为研究提供了基础，但它们的数量和多样性仍然有限.

在当前情况下，数据集的稀缺性是限制 LGL 发展的一个重要因素. 因此，有必要进一步研究和开发新的数据集，以更好地支持 LGL 的发展. 这些新数据集应该更广泛地涵盖各种不同领域的图，并且应该具有更丰富的标签和元数据，以便更全面地评估和验证 LGL 模型的性能.

11 总　　结

本文全面审视了 LGL，深入探讨了图学习与终身学习的交叉点，并深入探讨了动机、最新方法和可能的解决方案. 我们将 LGL 方法分为基于回放、正

则化、表示、预演、架构和混合等多类,旨在为研究人员提供清晰的分类和指导.这些方法在克服灾难性遗忘和实现持续性能改进方面起着关键作用.在介绍 LGL 方法时,我们不仅关注其理论基础和技术细节,还特别强调了其在实际应用中的可行性和有效性.当前的研究已经取得了一定的进展,例如在社交网络分析、推荐系统和生物信息学等领域展示了 LGL 的潜力.这些实际应用为进一步推动 LGL 的研究和应用提供了宝贵的经验和启示.除了对已有研究的综述外,本文还提出了一些开放性问题,如 LGL 的收敛性、可扩展性、可解释性和无监督学习等,这些问题为未来的研究方向提供了有益的启示.我们鼓励研究人员在探索新的 LGL 方法的同时,也要注重解决开放性问题,以推动该领域的进一步发展.总的来说,本文旨在为读者提供关于 LGL 的最新进展和研究方向,希望能够帮助读者深入探索 LGL 的潜力,并为未来的研究和应用奠定坚实的基础.

作者贡献声明: 刘壮负责整体研究框架的构建和论文的撰写;董子宸负责图连续学习方法和技术部分的撰写,并参与讨论;董宜琳负责对 LGL 理论结合实践的调研,以及对 LGL 分析和总结,并提供案例和实证等支持;尚家名、张帆、陈雨然、楼佩妍、孙欣然、王昱、赵军、Wayne Lin 共同参与了文献综述、数据的搜集和整理以及部分撰写等工作,并对论文提供补充和修改意见.董子宸和董宜琳为共同第 2 作者.

参 考 文 献

[1] Sun Suyuan, Wu Chaoyun, Wu Changyao, et al. An empirical study of civil servants ' lifelong E-learning continuance intention[C]//Proc of ICEB 2009 Proceedings. Macau, SAR China: ICEB, 2009: 420−430

[2] Zhang Ziwei, Cui Peng, Zhu Wenwu. Deep learning on graphs: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(1): 249−270

[3] Zhuang Fuzhen, Qi Zhiyuan, Duan Keyu, et al. A comprehensive survey on transfer learning[J]. Proceedings of the IEEE, 2020, 109(1): 43−76

[4] Liu Bing. Lifelong machine learning: A paradigm for continuous learning[J]. Frontiers of Computer Science, 2017, 11(3): 359−361

[5] Hong Xianbin, Wong Prudence, Liu Dawei, et al. Lifelong machine learning: Outlook and direction[C]//Proc of the 2nd Int Conf on Big Data Research. New York: ACM, 2018: 76−79

[6] Mahesh B. Machine learning algorithms−a review[J]. International Journal of Science and Research, 2020, 9(1): 381−386

[7] Chen Zhiyuan, Liu Bing. Lifelong Machine Learning[M]. San Rafael: Morgan & Claypool Publishers, 2018

[8] Dong Songlin, Hong Xiaopeng, Tao Xiaoyu, et al. Few-shot class-incremental learning via relation knowledge distillation[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 1255−1263

[9] Wang Kuansan, Shen Zhihong, Huang Chiyuan, et al. Microsoft academic graph: When experts are not enough[J]. Quantitative Science Studies, 2020, 1(1): 396−413

[10] Eckhoff M, Reiher M. Lifelong machine learning potentials[J]. Journal of Chemical Theory and Computation, 2023, 19(12): 3509−3525

[11] Zhang Peiyan, Yan Yuchen, Li Chaozhuo, et al. Continual learning on dynamic graphs via parameter isolation[C]//Proc of the 46th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2023: 601−611

[12] Verstaevel N, Boes J, Nigon J, et al. Lifelong machine learning with adaptive multi-agent systems[C]//Proc of the 9th Int Conf on Agents and Artificial Intelligence. Porto, Portugal: ICAART, 2017: 275−286

[13] Ruvolo P, Eaton E. Active task selection for lifelong machine learning[C]//Proc of the 27th AAAI Conf on Artificial Intelligence. Bellevue, Washington: AAAI, 2013: 862−868

[14] Xia Feng, Sun Ke, Yu Shuo, et al. Graph learning: A survey[J]. IEEE Transactions on Artificial Intelligence, 2021, 2(2): 109−127

[15] Yang Menglin, Zhou Min, Kalander M, et al. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space[C]//Proc of the 27th ACM SIGKDD Conf on Knowledge Discovery & Data Mining. New York: ACM, 2021: 1975−1985

[16] Skarding J, Gabrys B, Musial K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey[J]. IEEE Access, 2021, 9: 79143−79168

[17] Goyal P, Kamra N, He X, et al. DynGEM: Deep embedding method for dynamic graphs[J]. arXiv preprint, arXiv: 1805.11273, 2018

[18] Zhou Lekui, Yang Yang, Ren Xiang, et al. Dynamic network embedding by modeling triadic closure process[C]//Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2018: 571−578

[19] Xue Guotong, Zhong Ming, Li Jianxin, et al. Dynamic network embedding survey[J]. Neurocomputing, 2022, 472: 212−223

[20] Wang Chen, Qiu Yuheng, Gao Dasong, et al. Lifelong graph learning[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 13719−13728

[21] Tian Zonggui, Zhang Du, Dai Hongning. Continual learning on graphs: A survey[J]. arXiv preprint, arXiv: 2402.06330, 2024

[22] Kazemi S M, Goel R, Jain K, et al. Representation learning for dynamic graphs: A survey[J]. Journal of Machine Learning Research, 2020, 21(70): 1−73

[23] Barros C D T, Mendonça M R F, Vieira A B, et al. A survey on embedding dynamic graphs[J]. ACM Computing Surveys, 2021, 55(1): 1−37

[24] Lopez-Paz D, Ranzato M A. Gradient episodic memory for continual learning[J]. Advances in Neural Information Processing Systems, 2017, 30: 6470−6479

[25] Han Y, Karunasekera S, Leckie C. Graph neural networks with continual learning for fake news detection from social media[J]. arxiv preprint, arxiv: 2007.03316, 2020

[26] Wang Yuening, Zhang Yingxue, Coates M. Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2021: 3518−3522

[27] Ahrabian K, Xu Yishi, Zhang Yingxue, et al. Structure aware experience replay for incremental learning in graph-based recommender systems[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2021: 2832−2836

[28] Xu Yishi, Zhang Yingxue, Guo Wei, et al. GraphSAIL: Graph structure aware incremental learning for recommender systems[C]// Proc of the 29th ACM Int Conf on Information &Knowledge Management. New York: ACM, 2020: 2861−2868

[29] Ding Sihao, Feng Fuli, He Xiangnan, et al. Causal incremental graph convolution for recommender system retraining[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(4): 4718−4728

[30] Jha K, Xun G, Zhang A. Continual representation learning for evolving biomedical bipartite networks[J]. Bioinformatics, 2021, 37(15): 2190−2197

[31] Chen Xu, Wang Junshan, Xie Kunqing. TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning[J]. arXiv preprint, arXiv: 2106.06273, 2021

[32] Wang Junshan, Song Guojie, Wu Yi, et al. Streaming graph neural networks via continual learning[C]//Proc of the 29th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2020: 1515−1524

[33] Zhou Fan, Cao Chengtai. Overcoming catastrophic forgetting in graph neural networks with experience replay[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2021: 4714−4722

[34] Zhang Xikun, Song Dingjin, Tao Dacheng. Continual learning on graphs: Challenges, solutions, and opportunities[J]. arXiv preprint, arXiv: 2402.11565, 2024

[35] Kou Xiaoyu, Lin Yankai, Liu Shaobo, et al. Disentangle-based continual graph representation learning[J]. arXiv preprint, arXiv: 2010.02565, 2020

[36] Febrinanto F G, Xia F, Moore K, et al. Graph lifelong learning: A survey[J]. IEEE Computational Intelligence Magazine, 2023, 18(1): 32−51

[37] Cai Jie, Wang Xin, Guan Chaoyu, et al. Multimodal continual graph learning with neural architecture search[C]//Proc of the ACM Web Conf 2022. New York: ACM, 2022: 1292−1300

[38] Kim S, Yun S, Kang J. DyGRAIN: An incremental learning framework for dynamic graphs[C]//Proc of the 31st Int Joint Conf on Artificial Intelligence. Messe Wien, Vienna, Austria: IJCAI, 2022: 3157−3163

[39] Luo Yadan, Huang Zi, Zhang Zheng, et al. Learning from the past: Continual meta-learning with Bayesian graph neural networks[C]//Proc of the 34th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 5021−5028

[40] Gao Chen, Zheng Yu, Li Nian, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions[J]. ACM Transactions on Recommender Systems, 2023, 1(1): 1−51

[41] Ling Huan, Gao Jun, Kar A, et al. Fast interactive object annotation with curve-GCN[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 5257−5266

[42] Han Kai, Wang Yunhe, Guo Jianyuan, et al. Vision GNN: An image is worth graph of nodes[J]. Advances in Neural Information Processing Systems, 2022, 35: 8291−8303

[43] Liu Y, Hong X, Tao X, et al. Structural knowledge organization and transfer for class-incremental learning[C]//Proc of the 3rd ACM Int Conf on Multimedia in Asia. New York: ACM, 2021: 1−7

[44] Galke L, Franke B, Zielke T, et al. Lifelong learning of graph neural networks for open-world node classification[C]//Proc of 2021 Int Joint Conf on Neural Networks (IJCNN). Piscataway, NJ: IEEE, 2021: 1−8

[45] Gallardo J, Hayes T L, Kanan C. Self-supervised training enhances online continual learning[J]. arXiv preprint, arXiv: 2103.14010, 2021

[46] Mehta S V, Patil D, Chandar S, et al. An empirical investigation of the role of pre-training in lifelong learning[J]. Journal of Machine Learning Research, 2023, 24(214): 1−50

[47] Madaan D, Yoon J, Li Y, et al. Representational continuity for unsupervised continual learning[J]. arXiv preprint, arXiv: 2110.06976, 2021

[48] Purushwalkam S, MorgadoP, Gupta A. The challenges of continuous self-supervised learning[G]//LNCS 13686: Proc of European Conf on Computer Vision. Berlin: Springer, 2022: 702−721

[49] Sen P, Namata G, Bilgic M, et al. Collective classification in network data[J]. AI magazine, 2008, 29(3): 93−93

[50] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in Neural Information Processing Systems, 2017: 1024−1034

[51] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]//Proc of the 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2014: 701−710

[52] Grover A, Leskovec J. Node2Vec: Scalable feature learning for networks[C]//Proc of the 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855−864

[53] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arxiv preprint, arxiv: 1609.02907, 2016

[54] Lu Bin, Gan Xiaoying, Yang Lina, et al. Geometer: Graph few-shot class-incremental learning via prototype representation[C]//Proc of the 28th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2022: 1152−1161

[55] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint, arXiv: 1412.6980, 2014

[56] Tieleman T. Lecture 6.5 - RMSprop: Divide the gradient by a

running average of its recent magnitude[J]. COURSERA: Neural Networks for Machine Learning, 2012, 4(2): 26−31

[57] Perini M, Ramponi G, Carbone P, et al. Learning on streaming graphs with experience replay[C]//Proc of the 37th ACM/SIGAPP Symp on Applied Computing. New York: ACM, 2022: 470−478

[58] Rakaraddi A, Siew Kei L, Pratama M, et al. Reinforced continual learning for graphs[C]//Proc of the 31st ACM Int Conf on Information & Knowledge Management. New York: ACM, 2022: 1666−1674

[59] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting[J]. arxiv preprint, arxiv: 1709.04875, 2017

[60] Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, 2017, 114(13): 3521−3526

[61] Maltoni D, Lomonaco V. Continuous learning in single-incremental-task scenarios[J]. Neural Networks, 2019, 116: 56−73

[62] Parisi G I, Tani J, Weber C, et al. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization[J]. Frontiers in Neurorobotics, 2018, 12: 78

[63] Zeng Jiaqi, Xie Pengtao. Contrastive self-supervised learning for graph classification[C]//Proc of the AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2021, 35(12): 10824−10832

[64] Liu Qing, Majumder Orchid, Achille Alessandro, et al. Incremental few-shot meta-learning via indirect discriminant alignment[C]//Proc of Computer Vision–ECCV 2020. Berlin: Springer, 2020: 685−701

[65] Yuan Hao, Yu Haiyang, Gui Shurui, et al. Explainability in graph neural networks: A taxonomic survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(5): 5782−5799

[66] Ko J, Kang S, Kwon T, et al. Begin: Extensive benchmark scenarios and an easy-to-use framework for graph continual learning[J]. arxiv preprint, arxiv: 2211.14568, 2022

[67] Ruder S. An overview of gradient descent optimization algorithms[J]. arxiv preprint, arxiv: 1609.04747, 2016

[68] Mallya A, Lazebnik S. Packnet: Adding multiple tasks to a single network by iterative pruning[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 7765−7773

[69] Su Junwei, Zou Difan, Zhang Zijun, et al. Towards robust graph incremental learning on evolving graphs[C]//Proc of Int Conf on Machine Learning. New York: PMLR, 2023: 32728−32748

[70] De Cao N, KipfT. MolGAN: An implicit generative model for smallmolecular graphs[J]. arXiv preprint, arXiv: 1805.11973, 2018

[71] Simonovsky M, Komodakis N. GraphVAE: Towards generation of small graphs using variational autoencoders[C]//Proc of the 27th Int Conf on Artificial Neural Networks. Berlin: Springer, 2018: 412−422

[72] Shi Chence, Xu Minkai, Zhu Zhaocheng, et al. GraphAF: A flow-based autoregressive model for molecular graph generation[J]. arXiv preprint, arXiv: 2001.09382, 2020

[73] You Jiaxuan, Liu Bowen, Ying Zhitao, et al. Graph convolutional policy network for goal-directed molecular graph generation[J]. Advances in Neural Information Processing Systems, 2018, 31: 6412−6422

[74] You Jiaxuan, Ying Rex, Ren Xiang, et al. GraphRNN: Generating realistic graphs with deep auto-regressive models[C]//Proc of Int Conf on Machine Learning. Stockholmsmässan, Stockholm Sweden: PMLR, 2018: 5708−5717

[75] Dai Hanjun, Nazi A, Li Yujia, et al. Scalable deep generative modeling for sparse graphs[C]//Proc of Int Conf on Machine Learning. Vienna, Austria: PMLR, 2020: 2302−2312

[76] Liu Huihui, Yang Yiding, Wang Xinchao. Overcoming catastrophic forgetting in graph neural networks[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 8653−8661

[77] Dineva K, Atanasova T. Systematic look at machine learning algorithms–advantages, disadvantages and practical applications[J]. International Multidisciplinary Scientific GeoConference: SGEM, 2020, 20(2.1): 317−324

[78] Yuan H, Tang J, Hu X, et al. XGNN: Towards model-level explanations of graph neural networks[C]//Proc of ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2020: 430−438

[79] Sun Li, Ye Junda, Peng Hao, et al. Self-supervised continual graph learning in adaptive riemannian spaces[C]//Proc of the 37th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2023, 37: 4633−4642

[80] You Yuning, Chen Tianlong, Sui Yongduo, et al. Graph contrastive learning with augmentations[J]. Advances in Neural Information Processing Systems, 2020, 33: 5812−5823

[81] Ruvolo P, Eaton E. ELLA: An efficient lifelong learning algorithm [C]//Proc of Int Conf on Machine Learning. Atlanta, GA: PMLR, 2013: 507−515

[82] Rusu A A, Rabinowitz N C, Desjardins G, et al. Progressive neural networks[J]. arXiv preprint, arXiv: 1606.04671, 2016

[83] Lomonaco V, Maltoni D. Core50: A new dataset and benchmark for continuous object recognition[C]//Proc of Conf on Robot Learning. Mountain View, CA: PMLR, 2017: 17−26

[84] Golkar S, Kagan M, Cho K. Continual learning via neural pruning[J]. arXiv preprint, arXiv: 1903.04476, 2019

[85] Ahn H, Cha S, Lee D, et al. Uncertainty-based continual learning with adaptive regularization[J]. Advances in Neural Information Processing Systems, 2019, 32: 1−11

[86] Gao Yang, Yang Hong, Zhang Peng, et al. GraphNAS: Graph neural architecture search with reinforcement learning[J]. arXiv preprint, arXiv: 1904.09981, 2019

[87] Yoon J, Yang E, Lee J, et al. Lifelong learning with dynamically expandable networks[J]. arXiv preprint, arXiv: 1708.01547, 2017

[88] Mallya A, Davis D, Lazebnik S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights[C]//Proc of Computer Vision (ECCV 2018). Berlin: Springer, 2018: 67−82

[89] Wang Liyuan, Zhang Xingxing, Su Hang, et al. A comprehensive survey of continual learning: Theory, method and application[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024: 1−20

[90] Xia Jiafeng, Li Dongsheng, Gu Hansu, et al. Incremental graph convolutional network for collaborative filtering[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New

York: ACM, 2021: 2170−2179

[91] Qin Haiou, Zhang Du. A perpetual learning algorithm that incrementally improves performance with deliberation[J]. IEEE Access, 2020, 8: 131425−131438

[92] Zhang Xikun, Song Dongjin, Tao Dacheng. Hierarchical prototype networks for continual graph representation learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(4): 4622−4636

[93] Fortunato S, Bergstrom C T, Börner K, et al. Science of science[J]. Science, 2018, 359(6379): eaao0185

[94] Rebuffi S A, Kolesnikov A, Sperl G, et al. iCaRL: Incremental classifier and representation learning[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 2001−2010

[95] Cha H, Lee J, Shin J. Co2l: Contrastive continual learning[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Montreal. Piscataway, NJ: IEEE, 2021: 9516−9525

[96] Zhang J O, Sax A, Zamir A, et al. Side-tuning: A baseline for network adaptation via additive side networks[C]//Proc of Computer Vision (ECCV 2020). Glasgow, UK: SIP, 2020: 698−714

[97] Shon H, Lee J, Kim S H, et al. Dlcft: Deep linear continual fine-tuning for general incremental learning[G]//LNCS 13693: Proc of European Conf on Computer Vision. Berlin: Springer, 2022: 513−529

[98] Boschini M, Bonicelli L, Porrello A, et al. Transfer without forgetting[G]//LNCS 13661: Proc of European Conference on Computer Vision. Berlin: Springer, 2022: 692−709

[99] Cossu A, Tuytelaars T, Carta A, et al. Continual pre-training mitigates forgetting in language and vision[J]. arXiv preprint, arXiv: 2205.09357, 20221

[100] Chen Z, Ma N, Liu B. Lifelong learning for sentiment classification[J]. arxiv preprint, arxiv: 1801.02808, 2018

[101] Kmoblauch J, Husain H, Diethe T. optimal continual learning has perfect memory and is NP-haed[C]//proc of Int conf on Machine Learning, New York: PMLR, 2020: 5327−5337

[102] Tang J, Zhang J, Yao L, et al. Arnetminer: Extraction and mining of academic social networks[C]//Proc of the 14th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2008: 990−998

[103] Liu Jin, Tan Guanxin, Lan Wei, et al. Identification of early mild cognitive impairment using multi-modal data and graph convolutional networks[J]. BMC Bioinformatics, 2020, 21: 1−12

[104] Wang J, Zhang S, Xiao Y, et al. A review on graph neural network methods in financial applications[J]. arxiv preprint, arxiv: 2111.15367, 2021

[105] Wu Shiwen, Sun Fei, Zhang Wentao, et al. Graph neural networks in recommender systems: A survey[J]. ACM Computing Surveys, 2022, 55(5): 1−37

[106] Sharma K, Lee Y C, Nambi S, et al. A survey of graph neural networks for social recommender systems[J]. arxiv preprint, arxiv: 2212.04481, 2022

[107] Liu zhuang, Huang Degen, Huang Kaiyu, et al. Per−trained financial language representation model for financial text mining[C]//Proc of the 29 th Int Toint wnf on Artificial Intelligenle. IJ CAI: ijcai·org, 2020, 4513−4519

[108] He Xiangnan, Deng Kuan, Wang Xiang, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]// Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 639−648

[109] Wu F, Souza A, Zhang T, et al. Simplifying graph convolutional networks[C]//Proc of Int Conf on Machine Learning. PMLR, 2019: 6861−6871

[110] Zheng Yizhen, Pan Shirui, Lee V, et al. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination[J]. Advances in Neural Information Processing Systems, 2022, 35: 10809−10820

[111] Wu Lirong, Lin Haitao, Tan Cheng, et al. Self-supervised learning on graphs: Contrastive, generative, or predictive[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(4): 4216−4235

[112] Huang Qiang, Yamada M, Tian Yuan, et al. Graphlime: Local interpretable model explanations for graph neural networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(7): 6968−6972

[113] Miao Siqi, Liu Mia, Li Pan. Interpretable and generalizable graph learning via stochastic attention mechanism[C]//Proc of Int Conf on Machine Learning. New York: PMLR, 2022: 15524−15543

[114] Rymarczyk D, van de Weijer J, Zieliński B, et al. Icicle: Interpretable class incremental continual learning[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 1887−1898

[115] Chen Z M, Wei X S, Wang P, et al. Multi-label image recognition with graph convolutional networks[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 5177−5186

[116] Chen Chaoqi, Wu Yushuang, Dai Qiyuan, et al. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective[J]. arxiv preprint, arxiv: 2209.13232, 2022

[117] Lei C H, Chen Y H, Peng W H, et al. Class-incremental learning with rectified feature-graph preservation[C]//Proc of the Asian Conf on Computer Vision. Berlin: springer, 2020: 358−374

[118] Cai H, Zheng VW, Chang KCC. A comprehensive survey of graphembedding: Problems, techniques, and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1616−1637

[119] de Bruin T, Kober J, Tuyls K, et al. Improved deep reinforcement learning for robotics through distribution-based experience retention[C]//Proc of 2016 IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS). Piscataway, NJ: IEEE, 2016: 3947−3952

[120] Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems[C]//Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 974−983

[121] Javed K, White M. Meta-learning representations for continual learning[J]. Advances in Neural Information Processing Systems, 2019, 32: 1−1



Liu Zhuang, born in 1982. PhD, associate professor, master supervisor. His main research interests include natural language processing, fintech, graph neural networks, and LLM.

刘　壮, 1982 年生. 博士，副教授，硕士生导师. 主要研究方向为自然语言处理、金融科技、图神经网络、大语言模型.



Dong Zichen, born in 2003. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural network.

董子宸, 2003 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Dong Yilin, born in 2004. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

董宜琳, 2004 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



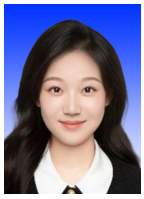
Shang Jiaming, born in 2003. Undergraduate. His main research interests include natural language processing, fintech, machine learning, and graph neural network.

尚家名, 2003 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Zhang Fan, born in 2002. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

张　帆, 2002 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Chen Yuran, born in 2005. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

陈雨然, 2005 年生，本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Lou Peiyan, born in 2004. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

楼佩妍, 2004 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Sun Xinran, born in 2004. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

孙欣然, 2004 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Wang Yu, born in 2003. Undergraduate. Her main research interests include natural language processing, fintech, machine learning, and graph neural networks.

王　昱, 2003 年生. 本科生，主要研究方向为自然语言处理、金融科技、机器学习、图神经网络.



Zhao Jun, born in 1990. PhD. His main research interests include natural language processing, machine translation, and graph neural networks.

赵　军, 1990 年生. 博士，主要研究方向为自然语言处理、机器翻译、图神经网络.



Wayne Lin, born in 1991. PhD candidate. Her main research interests include natural language processing, QA, and LLM.

Wayne Lin, 1991 年生. 博士研究生. 主要研究方向为自然语言处理、大语言模型.