拓扑不确定性下的探索性学习辅助社区检测统一框架

Yu Hou、CongTran、Ming Li，*IEEE* 会员，以及 Won-Yong Shin，*IEEE* 高级会员。

摘要—在社交网络中，社区结构的发现作为各种网络分析任务中的一个基础问题，受到了广泛关注。然而，由于隐私问题或访问限制，网络结构通常是不确定的，因此，如果不付出高昂的网络拓扑结构获取成本，既有的社区检测方法往往难以有效发挥作用。为应对这一挑战，我们提出了META-CODE，这是一种统一框架，用于在网络拓扑结构未知（或仅部分已知）时，通过探索性学习 辅以易于收集的节点元数据来检测重叠社区。具体而言，META-CODE 除初始网络推断步骤外，还包括三个迭代步骤：1）基于由我们提出的新型重构损失训练的图神经网络（*GNN*），生成节点级社区关联嵌入；2）通过基于社区关联的节点查询进行网络探索；3）利用来自已探索网络的基于边连通性的孪生神经网络模型进行网络推断。通过在五个真实世界数据集（包括两个大型网络）上开展大量实验，我们证明了：（a）META-CODE优于基准社区检测方法，与现有最佳竞争方法相比，增益最高达65.55%；（b）META-CODE 中各模块的影响；（c）基于实证评估和理论发现，META-CODE 中节点查询的有效性；（d）推断网络的收敛；以及（e）META-CODE 的计算效率。

索引词—社区检测、探索性学习、图神经网络、网络推断、节点查询

arXiv:2304.04497v3 [cs.SI] 5 Mar 2024

1 引言

1.1 背景与动机

COMMUNITY 检测是各种网络分析中的关键且重要任务之一，用于理解网络的基本特征。在社交网络中，社区成员归属通常允许重叠，即节点可以同时属于多个社区 [3]。因此，社区检测算法需要适应这种固有的重叠特性；为此，研究人员已开发出一系列技术，例如基于团和模块度的方法 [3], [4], 、基于非负矩阵分解（NMF）的方法 [5],[6],以及基于深度学习的方法 [7], [8], 用于检测重叠社区。

虽然上述社区检测方法始终依赖网络结构来检测社区，但底层的真实网络最初是未知的或仅部分已知，这在各种现实世界网络应用领域中都很常见 [9]。因此，除非能够获取网络结构，否则图上的下游机器学习（ML）任务（包括社区检测）将难以奏效。虽然可以在开展下游应用之前投入额外精力来揭示完整网络，但收集完整的拓扑信息成本高昂且耗费人力 [9]。另一种方法是利用从社交媒体收集的网络数据，来推断底层网络中节点之间的未知关系。尽管社交媒体平台日益普及，但这一信息来源仍有局限性，例如资源受限，以及用户设置了更严格的隐私选项。例如，2011年6月对纽约市 Facebook 用户进行的一项人口统计分析发现，52.6%的用户隐藏了好友列表 [10]。此外，网络连接信息往往包含许多弱连接 [11], ，例如过时的好友关系，这些关系不利于检测社区。因此，就网络分析而言，在实践中应假设网络结构完全不可用或仅部分可用 [12]–[15]。

一方面，已有多次尝试来解决拓扑未知的网络中的机器学习问题。例如，探索性 影响力最大化 [16]–[18]被提出，同时还提出了一种通过查询单个节点来探索其邻域、从而在未知网络上执行信息检索的解决方案。尽管探索性影响力最大化在解决影响力最大化问题方面取得了成功，但通过探索性学习进行社- 区检测 的相关研究仍尚未探索。另一方面，在网络设定中

• *Y.Hou* 任职于延世大学数学与计算学院（计算科学与工程），韩国首尔*03722*。电子邮箱：*houyu@yonsei.ac.kr*

• *C. Tran*任职于越南河内*100000*的邮电技术学院信息技术学院。电子邮箱：*congtt@ptit.edu.vn*

• *M. Li*为浙江省智能教育技术与应用重点实验室、浙江师范大学，地址：中国浙江金华*321004*。电子邮箱：*mingli@zjnu.edu.cn*

• *W.-Y. Shin* 任职于韩国首尔 *03722* 延世大学数学与计算学院（计算科学与工程系），以及韩国浦项 *37673* 浦项科技大学（*POSTECH*）人工智能研究生院。电子邮箱：*wy.shin@yonsei.ac.kr*（通讯作者：*Won-YongShin*。）

表 1 重叠社区检测方法在拓扑结构信息程度和是否存在探索性学习方面的差异总结。

方法	所需信息			探索性学习
	完整拓扑结构	部分拓扑结构	元数据	
AGM [20], BIGCLAM [5],DANMF [6], vGraph [21]	✓			
CESNA [22], NOCD [7], DGLFRM [8]	✓		✓	
HGP-EPM [23],KroMFac [14]		✓		
META-CODE（拟议）			✓	✓

结构是部分可观测的，因此人们尝试通过根据观测部分重构底层网络中未观测的部分，然后检测社区结构[14]，[19]来解决社区检测问题。在我们的研究中，我们主要关注网络结构完全未知的情况，并将部分可观测网络的情形视为一种特殊情况，原因如下。首先，由于缺乏无需结构信息即可运行的机器学习工具，初始时假设拓扑结构完全未知更具挑战性。值得注意的是，正如表1 所总结的那样，拓扑结构完全未知的网络设置带来了独特的技术难题，必须采用不同于为部分可观测网络设计的社区检测方法的方式来处理。其次，我们的拓扑结构未知设置自然涵盖了底层网络部分可观测时的社区检测问题；换句话说，解决我们的问题本身就会导向对拓扑结构部分已知的网络进行社区检测的解决方案（稍后将对此进行说明）。

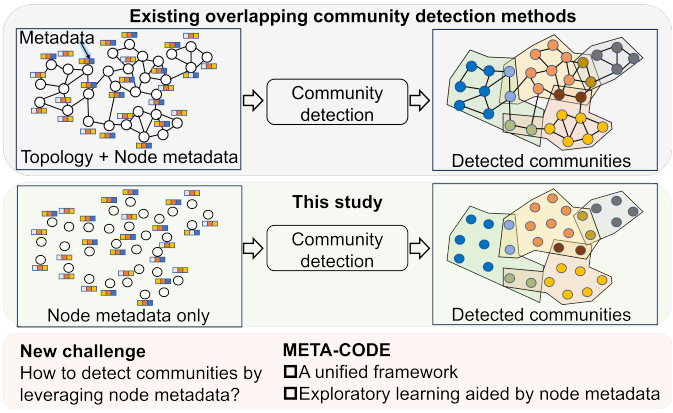


图 1. 拓扑未知的网络中重叠社区检测面临的新挑战。

在社交网络中，节点元数据 （例如用户的兴趣爱好或教育经历）可用于解决图上的许多机器学习问题并提升性能。这类节点元数据通常可以从各种来源获取，例如用户生成的内容 [24] 或从用户的行为中推断 [25]。例如，在合著网络中，只需收集每位作者发表作品中的关键词，就可以推断作者之间的主题关系。在针对无家可归青少年开展 HIV 认知宣传的情况下 [26],，个人以及无家可归者的姓名、年龄和家庭背景等元数据，可以通过福利中心为发放每月补助而要求进行的强制登记流程轻松收集。由于节点元数据 readily available [24]–[26], ，将其纳入探索性学习框架，有助于克服与拓扑结构未知（或仅部分已知）的网络相关的实际挑战。在此背景下，如图 1 所示，一个自然产生的问题是：“如何借助节点元数据，在拓扑结构不确定的情况下有效执行重叠社区检测？”

1.2 主要贡献

为回答这一问题，我们首次尝试在缺乏结构信息的情况下，通过从节点查询和节点元数据中获取信息，开发一种重叠式社区检测方法。为此，我们首先概述构建新方法时必须应对的两项设计挑战。

- **探索式学习：** 如何借助节点查询有效探索子网络；
- **节点元数据的利用：** 如何设计社区检测方法，以最大限度地利用 节点元数据。

(思路 1) 仅依据节点元数据，在拓扑结构未知的网络中准确发现社区结构是一项极具挑战性的任务。为了辅助社区检测，我们可以在给定查询预算的情况下进行节点查询，以获取结构信息。随后，可以利用节点元数据和通过节点查询得到的已探索边，推断未探索部分中两个节点之间的连接，从而更准确地识别社区关联关系。在本研究中，我们提出了一个新问题：联合求解社区归属嵌入矩阵和一系列节点查询，使得发现的社区关联关系能够让推断网络和节点元数据更为可能。

(思路 2) 值得注意的是，现有的社区检测方法已无法解决上述问题（参见表 1）。换言之，我们需要建立全新的设计原则，提出自己的重叠社区检测方法。为此，我们提出了一种可扩展且有效的社区检测方法 META-CODE，该方法借助节点元数据进行探索式学习。META-CODE 首先推断网络结构，然后迭代地执行以下三个步骤，以确保社区检测性能逐步提升：

- **步骤 1：社区归属嵌入。**人们对……的研究

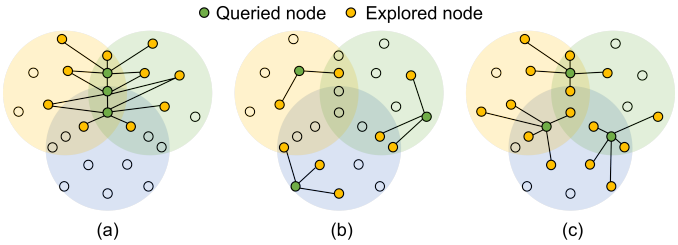


图 2. 当底层真实网络中存在三个重叠社区时，采用不同查询节点选择策略进行网络探索。(a) 选择度最高的节点作为查询节点。(b) 选择非重叠区域中的节点作为查询节点。(c) 选择属于多个社区且分布于不同社区的节点作为查询节点。

一直以来，人们对如何在属性网络中检测重叠社区的研究都较为有限，而这类网络中存在节点元数据。在本研究中，受图神经网络（GNN） [27],[28]已被广泛研究并被视为在属性网络中学习表示的强大方法这一事实启发，我们旨在设计一种由新型重构损失训练的图神经网络模型，以学习节点级社区归属嵌入。为此，我们首先刻画两类关系，即结构–社区关系和元数据–社区关系，然后利用它们训练模型。

• **步骤 2：通过节点查询进行网络探索**。遵循先前工作 [16], [18], [29],，我们使用一系列节点查询来获取更多结构信息。然而，要选择能够加快网络探索的节点查询，是一项相当具有挑战性的任务。简单的查询策略未必能实现快速网络探索；例如，在图 2a 和 2b 中，选择高度节点或非重叠区域中的节点（即仅属于一个社区的节点）可能无法实现快速网络探索。另一方面，图 2c 展示了这样一种情况：我们选择的节点 (i) 位于重叠 社区区域内，且 (ii) 分布于不同 社区。为此，我们提出了一种新的节点查询策略，并从理论上验证其有效性。

• **步骤 3：网络推断**。网络探索之后，仅有一小部分边得到进一步探索，这不足以准确检测社区。为了推断底层网络中尚未探索的部分，以实现更准确的社区检测，我们设计了一种基于边连接性的孪生神经网络（*EC-SiamNet*），该网络结合了从查询节点获取的已探索边和节点元数据。然后，在下一次迭代的步骤 1 中，将推断出的网络作为图神经网络模型的输入，以获得增强的社区归属嵌入。请注意，我们的先前工作[15] 没有这一网络推断步骤，而仅在初始阶段推断网络结构。换言之，在没有这一网络推断模块的情况下，我们

表 2 符号说明摘要。

符号表示	描述
$\mathcal{G}$	底层网络
$\mathcal{V}$	$\mathcal{G}$中的节点集合
$\mathcal{E}$	$\mathcal{G}$中的边集合
N	节点数量
$\mathcal{X}$	节点元数据
D	$\mathcal{X}$ 中每个特征向量的维度
K	社区数量
$\mathbf{F}$	非负权重隶属矩阵
S_t	t 节点查询序列
T	节点查询预算
$\mathcal{E}_t$	第 t次查询步骤后的已探索边集合
$\mathcal{G}^{(t)}$	第 t次节点查询后的推断网络
$\mathcal{E}^{(t)}$	$\mathcal{G}^{(t)}$ 中的边集合

先前工作 [15] 未能充分利用通过网络探索获得的结构信息，限制了其准确检测社区的潜力。

我们的主要贡献总结如下：

• **新颖方法：** 我们提出了 META-CODE，这是一种可扩展的端到端框架，用于解决拓扑未知网络中的社区检测问题。该框架包括：1）采用我们提出的重构损失进行训练的图神经网络模型；2）用于加速网络探索的查询节点选择策略；以及 3）用于推断网络结构的 EC-SiamNet。

• **分析与评估：** 我们通过在五个真实世界数据集（包括两个大型网络，例如，就数据集中使用的节点平均数量而言，其规模分别为 [30] 的 x61.2 倍和 x90.0倍）上开展广泛实验，验证了 META-CODE的合理性和有效性。我们证明了：1）META-CODE 优于基准重叠社区检测方法；与最佳竞争方法和 META-CODE 的一种简单变体相比，增益分别达到 65.55% 和 19.84%；2）META-CODE 中各组件的影响；3）META-CODE中节点查询策略的理论验证；4）关键超参数的影响；以及 5）可扩展性和计算复杂度分析。

1.3 组织结构与符号说明

本文其余部分安排如下。第 2 节介绍与本研究相关的已有研究。第 3 节阐述本研究的方法，包括问题定义以及 META-CODE 方法概述。第 4 节介绍所提方法的技术细节。第 5 节展示全面的实验结果。最后，第 6 节总结全文并给出结论。

表 2 总结了本文所使用的符号表示。在介绍我们的方法和技术细节时，我们将在后续章节中对这些符号表示进行形式化定义。

2 相关工作

我们提出的方法与四个更广泛的研究领域相关，分别是重叠社区检测、使用图神经网络进行社区检测、不完整网络中的社区检测，以及网络探索。

重叠社区检测。人们开发了各种技术来检测重叠社区。最初提出了基于团和模块度等基本概念的方法，用于识别重叠社区 [3], [4]。社区归属图模型（AGM） [20] 应运而生，它利用了这样一种观察：重叠社区区域内的节点往往比非重叠区域中的节点具有更密集的连接。基于 AGM，BIG-CLAM [5] 和 EPM [23] 分别采用非负矩阵分解和伯努利-泊松（BP）模型来检测重叠社区。此外，还有采用堆叠式非负矩阵分解的 DANMF [6], ，以及利用概率生成模型的 vGraph [21], 。此外， [22], [31] 还提出了少数几种方法，通过结合网络结构和节点元数据来检测重叠社区。

使用图神经网络进行社区检测。图神经网络已被广泛用于解决社区检测问题[32]–[37]。例如，AGC [32]和 SGCN [33] 等方法利用图卷积网络（GCN）[27]捕获用于社区检测的复杂特征，同时通过图卷积层聚合每个节点的邻居信息。DMGI [34] 和 MAGNN[35]采用图注意力网络（GAT） [28]自适应地学习邻域中每个节点的重要性。此外，MGAECD [36] 和 GUCD [37]在基于自编码器（AE）的技术中使用 GCN，通过重构原始信息，利用隐藏层中的嵌入表示社区。此外，文献[38], 提出了一种用于多目标进化算法的 GNN 编码技术，以解决复杂属性网络中的社区检测问题。尽管上述研究侧重于检测互不相交（即非重叠）的社区，但只有少数研究通过图神经网络处理重叠社区检测问题 [7], [8]。例如，NOCD [7]提出了一种基于图神经网络的方法，通过融合网络结构和节点元数据来检测重叠社区；DGLFRM [8]则将随机块模型 [2] 与图神经网络架构相结合，用于重叠社区检测。

不完整网络中的社区检测。一些研究探讨了在底层网络不完整（不确定）且存在缺失边的情况下识别社区的问题。其中一些研究利用了额外信息，例如节点元数据 [22] 或拓扑相似性 [39], 来进行社区检测。另有研究尝试预测不完整网络中缺失的边，随后在预测得到的拓扑结构中发现社区 [19]。一种分层伽马过程无限边划分模型 [23]不仅被设计用于识别重叠社区，还可预测缺失的边。此外，其他研究则直接从原始数据源中提取社区信息，例如缺失边网络中的影响级联 [40] 和时间序列数据[41], 。在 [14], 一项研究中，研究者将节点排序与非负矩阵分解相结合，以提高缺失边网络中社区检测的准确性。表 1 总结了采用不同程度拓扑信息（包括完整拓扑结构、部分拓扑结构和（几乎）无拓扑结构）时，各种重叠社区检测方法之间的差异。

网络探索。 网络探索旨在探索拓扑结构完全未知或部分可观测的网络，以帮助解决图上的下游机器学习任务。Jump-Crawl 模型 [42]和 BEER[43] 被提出作为识别有影响力节点的方法，用于探索未知网络。在处理缺乏拓扑结构信息的网络时，许多影响力最大化方法尝试寻找有影响力节点和查询节点，以便在未知网络中获取其邻居。遵循主动学习 [44], [45], 的理念，HEALER [46]研究了跨多个轮次的动态影响力最大化，并在每轮之后收集边信息。此外，CHANGE [16] 和 Geometric-DQN [17] 等探索式影响力最大化方法也被提出，它们通过单个节点查询来学习边。IM-META [18] 引入了一种拓扑结构感知的排序策略，以提高网络探索在执行影响力最大化任务时的有效性。

注： 尽管关于社区检测已经开展了大量研究，但现有大多数方法或多或少都需要网络结构来推断社-区结构，这在实际情境中可能需要耗费大量人力。此外，利用节点查询和节点元数据的思想尚未应用于拓扑结构未知的网络中的社区检测。

3 方法论

在本节中，作为所提出的META-CODE方法的基础，我们首先介绍网络模型，概述基本假设。然后，我们对拓扑未知的网络中的社区检测问题进行表述。我们还将概述 META-CODE 方法。

3.1 设置与基本假设

我们将底层真实网络表示为 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$，其中 $\mathcal{V}$ 是节点集合， $\mathcal{E}$ 是最初不可用的边集合。假设网络 $\mathcal{G}$ 是一个无向、无权的属性网络，不含自环和重复边，并具有可收集的节点元数据 $\mathcal{X} \in \mathbb{R}^{N \times D}$ ，即二元值节点特征，[1] 其中 D 是每个特征向量的维度。我们假设社交网络遵循 AGM [20], ，该模型认为，一对节点所属的社区越多，这对节点相连的概率就越高。社区信息可由非负权重关联矩阵 $\mathbf{F}^{(t)} \in \mathbb{R}^{N \times K}$ 表示，并将其视为节点级社区关联嵌入，[2] 其中 K 是其...

1. 非二值节点特征可以通过独热编码 [31]转换为二值。

2.为简化符号说明，在不引起歧义的情况下，省略 t 后，将表示第 t 次查询步骤中节点级社区关联嵌入的 $\mathbf{F}^{(t)}$ 记作 $\mathbf{F}$ 。

社区数量，且$\mathbf{F}_u$表示节点 u 的归属关系。

类似于主动学习 [44], [45]，我们选择节点查询序列，记为 S_t，并设定查询预算 T，以从底层网络 $\mathcal{G}$中获取额外的结构信息，其中 $0 \leq t \leq T-1$。请注意，利用查询预算的策略已被广泛用于在拓扑未知的网络 [16]–[18]上解决机器学习任务。查询单个节点 v_t后，我们可以发现其邻居，记为 $\mathcal{N}_{\mathcal{G}}(v_t)$，并相应地扩展可观测子网络，这本质上遵循了先前工作 [16], [18], [29]中的相同设定。将经过 t 次查询后已探索的边集合记为 $\mathcal{E}_t$。然后，在第$(t+1)$次节点查询期间，我们从 $\mathcal{V}$ 中选择一个节点v_t 进行扩展，并更新已探索边集合 $\mathcal{E}_{t+1} = \mathcal{E}_t \cup \mathcal{E}(\mathcal{N}_{\mathcal{G}}(v_t)\ v_t)$，其中 $\mathcal{E}(\mathcal{N}_{\mathcal{G}}(v_t)\ v_t)$是所有边的集合，这些边与 $\mathcal{N}_{\mathcal{G}}(v_t)$中的每个节点以及节点 v_t 相关联。此外，第$(t+1)$次节点查询后的推断网络 $\mathcal{G}^{(t+1)}$ 表示为 $\mathcal{G}^{(t+1)} = \left(\mathcal{V}, \mathcal{E}^{(t+1)}\right)$，其中 $\mathcal{E}^{(t+1)}$ 是 $\mathcal{G}^{(t+1)}$ 中的边集合，并且可以通过 $\mathcal{E}^{(t+1)} = \mathcal{E}_{t+1} \cup \mathcal{E}_I^{(t+1)}$ 进行更新。这里，$\mathcal{E}_I^{(t+1)} = f_I(S_{t+1}, \mathcal{E}_{t+1})$，其中 f_I 是根据查询节点与其邻居之间的关系推断网络结构的函数。

3.2 问题形式化

在本小节中，我们针对拓扑结构初始未知的网络构建社区检测问题；与部分可观测网络中的社区检测问题相比，该问题更具挑战性。为此，我们旨在通过同时获取节点元数据和查询节点的信息，在给定查询预算 T 的情况下，联合求得社区归属嵌入矩阵$\mathbf{F}$ 以及节点查询序列 $S_T = \{v_0, v_1, \cdots, v_{T-1}\}$，该序列将导向最终的已探索边$\mathcal{E}_T$。似然 $\mathbb{P}\left(\mathcal{G}^{(T)}\mathcal{X}\,|\mathbf{F}\right)$用于评估给定的推断网络 $\mathcal{G}^{(T)}$ 和节点元数据 $\mathcal{X}$更有可能对应哪个归属嵌入矩阵 $\mathbf{F}$。

如果一些节点根据 $\mathbf{F}$ 属于同一个社区，那么它们在 $\mathcal{G}^{(T)}$ 中相互连接的可能性更高，和 / 或共享来自 $\mathcal{X}$ 的共同元数据。这使我们能够从网络 $\mathcal{G}^{(T)}$和节点元数据 $\mathcal{X}$ 中发现社区 $\mathbf{F}$ 。我们将该问题表述为联合求解最优 $\mathbf{F}^*$ 和 S_T^*，以最大化似然 $\mathbb{P}\left(\mathcal{G}^{(T)}, \mathcal{X}\,|\mathbf{F}\right)$，如下所示:

$$(\mathbf{F}^*, S_T^*) = \underset{\mathbf{F}\geq 0, S_T \subseteq \mathcal{V}}{\arg\max}\ \mathbb{P}\left(\mathcal{G}^{(T)}, \mathcal{X}\,|\mathbf{F}\right), \tag{1}$$

其中 $\mathcal{G}^{(T)} = \left(\mathcal{V}, \mathcal{E}^{(T)}\right)$; $\mathcal{E}^{(T)} = \mathcal{E}_T \cup \mathcal{E}_I^{(T)}$; 以及 $\mathcal{E}_I^{(T)} = f_I\ S_T\ \mathcal{E}_T$ (,).

在我们的研究中，我们着力解决以下两个实际挑战。

- 据我们所知，此前尚无尝试在缺乏结构信息的网络中检测重叠社区。所有现有的社区检测方法在解决上述问题时都不再有效;

- 在所有节点中寻找用于估计 $\mathcal{E}$ 的一系列节点查询S_T是NP难问题，其复杂度关于 N呈指数级。

算法 1： META-CODE

输入： $\mathcal{V}, \mathcal{X}, S_0 = \emptyset, T$
输出： $\mathbf{F}$ S_T

```
函数META-CODE
    G^(0) ← 网络初始化(X)
    对于 t 从0 到 T 执行
        F ← 社区检测(G^(t), X)
        如果 t = T 则
            返回 F , S_T
        v_t ← QNodeSelect(F S_t)
        S_{t+1} ← S_t ∪ {v_t}
        E_{t+1} ← E_t ∪ E(N_G(v_t) v_t)
        E_I^(t+1) ← NetInfer(S_{t+1}, E_{t+1})
        G^(t+1) ← (V, E_{t+1} ∪ E_I^(t+1))
```

这促使我们提出一种兼具可扩展性和准确性的解决方案来解决该问题。

3.3 META-CODE 概述

在本小节中，我们将结合对 META-CODE 方法的概述，介绍我们的方法。借助从节点元数据和查询节点中获取的信息，我们生成社区归属嵌入矩阵 $\mathbf{F}$ 。META-CODE 的整体流程如算法 1 所示。该算法首先仅基于节点元数据推断初始网络结构 $\mathcal{G}^{(0)}$ （NetInitilize——参见第 2行）。随后，算法在我们的预算 T 内反复执行以下三个步骤：1）社区归属嵌入（ComDetect——参见第 4行）；2）通过节点查询进行网络探索（QNodeSelect——参见第 7 行）；以及 3）网络推断（NetInfer——参见第 10 行）。META-CODE 的示意性概览如图 3 所示。

步骤 1：社区关联嵌入。我们通过使用 GNN 架构生成节点级社区关联嵌入 $\mathbf{F}$，来应对检测重叠社区的挑战。我们的图神经网络模型从推断网络 $\mathcal{G}^{(t)}$ 和节点元数据 $\mathcal{X}$ 中提取社区关联嵌入 $\mathbf{F}$ ，其表示为:

$$\mathbf{F} := GNN_\theta(\mathcal{G}^{(t)}, \mathcal{X}), \tag{2}$$

其中 θ 是图神经网络模型中的可训练参数。

为确保模型准确捕捉结构和属性信息，我们通过建立自己的重构损失来训练 GNN 参数 θ，以保留结构和属性信息，具体内容将在第 4.2 节中说明。

步骤 2：通过节点查询进行网络探索。我们通过节点查询过程探索网络，根据步骤 1 中获得的社区归属嵌入$\mathbf{F}$ 选择一个能够实现较高网络探索程度的待查询节点。这一步的灵感来源于这样一种观点：属于多个社区且分布于不同社区的查询节点能够加速已探索网络的扩展，这将在第 5.5.3 节中通过实证进行验证。选定的节点 v_t 被添加到序列 $S_{t+1} = S_t \cup \{v_t\}$ 中，并找到其邻居

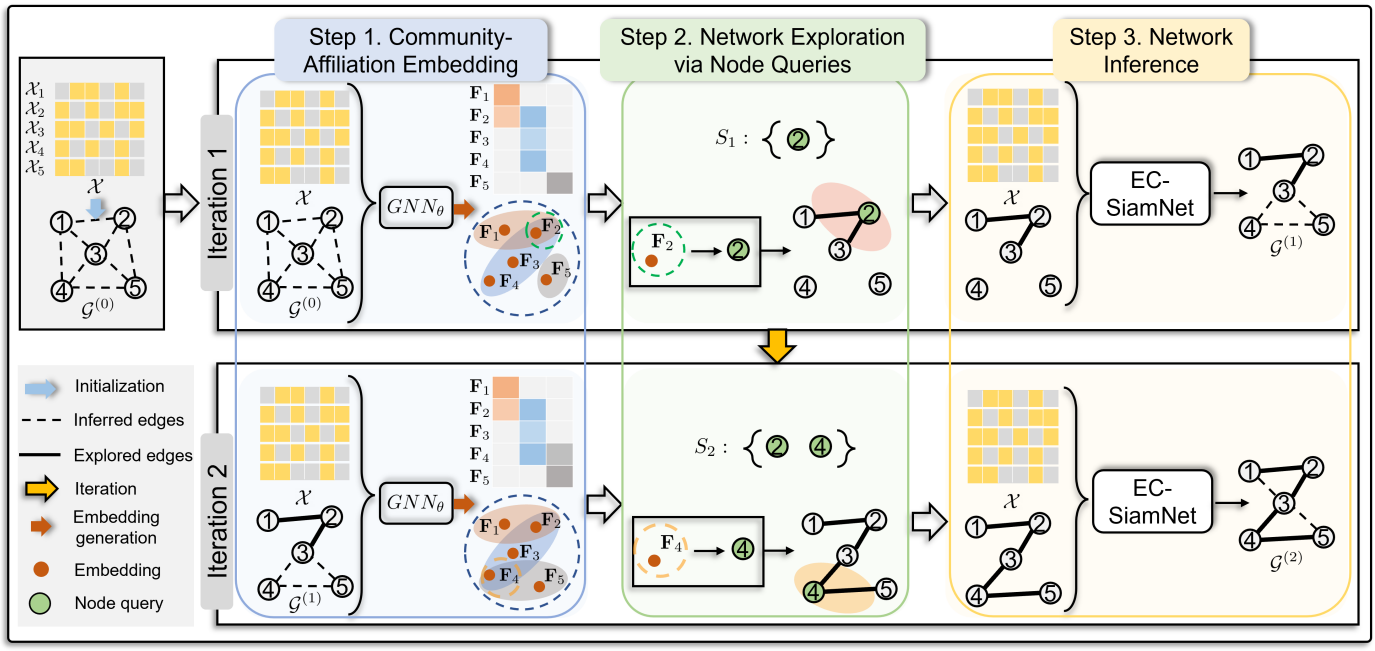


图 3. META-CODE 的示意性概览，由三个迭代步骤组成：1）社区归属嵌入，2）通过节点查询进行网络探索，以及 3）网络推断，其中执行了第一次和第二次迭代。

$\mathcal{N}_\mathcal{G}(v_t)$ 以进行网络探索。已探索边相应更新为 $\mathcal{E}_{t+1} = \mathcal{E}_t \cup \mathcal{E}(\mathcal{N}_\mathcal{G}(v_t), v_t)$。例如，如图 3 所示，在第一次迭代中查询节点 v_2 时，我们能够发现其两个邻居，即 v_1 和 v_3。

步骤 3：网络推断。我们采用 SiamNet 模型 [47]，该模型已广泛用于计算机视觉和模式分析 [48]–[50]，将其作为推断函数 $f_I(S_{t+1}, \mathcal{E}_{t+1})$ 以推断边 $\mathcal{E}_I^{(t+1)}$。在我们的研究中，我们提出 *EC-SiamNet* 作为一种强大的架构，基于由已探索边获得的连通性信息来学习两个节点之间的相似性。这通过在嵌入空间中生成吸引力和排斥力来实现。具体而言，我们的 EC-SiamNet 将节点元数据 $\mathcal{X}$ 作为输入，并返回一组相似性分数（记为 $\Psi^{(t+1)}$）作为输出；同时，该模型基于查询节点 S_{t+1}和已探索边 $\mathcal{E}_{t+1}$，使用正边和负边进行训练。例如，正边对应于与 $\mathcal{N}_\mathcal{G}(v_2)$ 中的每个节点以及节点 v_2 相连的所有边（参见图 3 中的网络推断步骤）。训练完成后，EC-SiamNet 计算相似性分数$\Psi^{(t+1)}$；当分数 $\psi_{uv}^{(t+1)} \in \Psi^{(t+1)}$ 高于某个阈值时，便会添加节点 u 和节点 v 之间的边。随后，将推断出的边 $\mathcal{E}_I^{(t+1)}$ 纳入推断网络 $\mathcal{G}^{(t+1)}$的边$\mathcal{E}^{(t+1)}$ 中，其表示为$\mathcal{E}^{(t+1)} = \mathcal{E}_{t+1} \cup \mathcal{E}_I^{(t+1)}$。

我们的方法通过迭代执行这三个步骤，在每次迭代中将更新后的推断网络 $\mathcal{G}^{(t+1)}$ 和节点元数据 $\mathcal{X}$ 作为图神经网络模型的输入，从而生成越来越准确的社区归属嵌入 $\mathbf{F}$。当达到给定的节点查询预算 T 时，这一迭代过程便会终止。值得注意的是，我们的方法也适用于网络初始时部分可观测的情况。这是因为，

在迭代执行这三个步骤时，我们的方法自然地通过节点查询解决了针对已探索子网络的社区检测问题，这可以被视为部分可观测情形。

4 所提出的 META-CODE 方法

本节将详细阐述 META-CODE 方法的各个步骤。我们还将进行理论分析，包括 META-CODE 中网络探索的有效性以及 META-CODE 的计算复杂度。

4.1 初始网络推断

起初，我们面临结构信息缺失的问题。为解决这一问题，我们需要仅依据可用的节点元数据来推断初始网络结构。

为此，作为初始网络推断的第一步，我们沿用先前工作 [15]，采用多分配聚类（MAC）方法 [51]。该方法通过确定初始社区成员归属 $\mathbf{C} \in \{0,1\}^{N \times K}$，根据节点与基于节点元数据得出的社区原型之间的相似度，将节点分配至多个社区 $\mathcal{X}$。其中，$\mathbf{C}_{uk}$ 表示节点 u 是否属于社区 k。该分配过程依赖于将 $\mathcal{X}$ 分解为 $\mathbf{C} \otimes \mathbf{U}$，其中矩阵 $\mathbf{U} \in \{0,1\}^{K \times D}$ 表示社区原型，而 $\mathbf{U}_{k\cdot}$ 表示社区 k的原型。这里，运算符 $\otimes$ 的定义使得节点 u 的第 d 个属性满足

$$\mathcal{X}_{ud} = \vee_{k=1}^{K} [\mathbf{C}_{uk} \wedge \mathbf{U}_{kd}], \tag{3}$$

这意味着具有相同社区成员归属的节点往往共享相似的节点元数据，其中$\mathcal{X}_{ud}$ 是节点 u的第 d个属性。

然后，作为初始网络推断的第二步，我们使用 AGM [20]，，根据初始社区成员归属 $\mathbf{C}$ 推断网络 $\mathcal{G}^{(0)}=\left(\mathcal{V},\mathcal{E}^{(0)}\right)$，其中 $\mathcal{E}^{(0)}$ 表示初始推断出的边集合。这里，每条边 ($u\ v\ \in$ $\mathcal{E}^{(0)}$,) 均以概率创建。

$$p\left(u,v\right)=1-\prod_{i=1}^{c_{uv}}\left(1-p\right), \tag{4}$$

其中，p 是社区内两个节点之间形成边的概率，而 c_{uv} 是根据社区成员归属 $\mathbf{C}$，节点 u 和 v 共同拥有的社区数量。这意味着，如果节点对 u 和 v 共同属于更多社区，则创建边 (u,v) 的概率更高。

4.2 社区归属嵌入

让我们回顾一下，社区关联嵌入$\mathbf{F}$可以由公式 (2) 中的图神经网络模型推导得到。利用推断网络 $\mathcal{G}^{(t)}$ 和节点元数据 $\mathcal{X}$，我们旨在设 计图神经网络模型，生成由非负权重隶属矩阵 $\mathbf{F}=GNN_{\theta}(\mathcal{G}^{(t)},\mathcal{X})$表示的节点级嵌入。为实现这一目标，我们刻画以下两种关系。

4.2.1 结构—社区关系

根据AGM，如果两个节点共同属于更多社区，那么它们在网络中相互连接的可能性就更高（参见（4））。这意味着网络结构与社区归属之间存在很强的相关性。

为刻画网络 $\mathcal{G}^{(t)}$中的网络结构与社区归属之间的关系，我们在 [5]中采用概率生成模型。这涉及基于社区归属嵌入矩阵$\mathbf{F}$，将节点 u和 v之间存在边的概率建模为 $1-\exp\left(-\mathbf{F}_u\mathbf{F}_v^T\right)$。$1-\exp\left(-\mathbf{F}_u\mathbf{F}_v^T\right)$ 的值越高，表示网络中节点 u 和 v 之间存在边的概率越高，这意味着，如果一对节点共同属于更多社区，则该节点对相互连接的概率更高。例如，如图4所示，具有相同社区成员归属的节点v_1 和 v_2，相比于可能属于不同社区的节点（如节点 v_1和 v_3），具有更高的连接概率。

在此背景下，为了确定哪种社区归属嵌入 $\mathbf{F}$ 会使给定的推断网络 $\mathcal{G}^{(t)}$ 更有可能，我们将损失函数$\mathcal{L}_1(\mathbf{F})$ 表述为：

$$\begin{aligned}\mathcal{L}_1\left(\mathbf{F}\right)&=-\log\mathbb{P}\left(\mathcal{G}^{(t)}|\mathbf{F}\right)\\&=-\sum_{(u,v)\in\mathcal{E}^{(t)}}\log\left(1-\exp\left(-\mathbf{F}_u\mathbf{F}_v^T\right)\right)+\sum_{(u,v)\notin\mathcal{E}^{(t)}}\mathbf{F}_u\mathbf{F}_v^T,\end{aligned} \tag{5}$$

其中，$\mathcal{E}^{(t)}$ 是 $\mathcal{G}^{(t)}$中的边集合。

4.2.2 元数据–社区关系

直观而言，如果节点属于同一个社区，那么它们很可能具有共同的节点属性 [22]。如图所示。

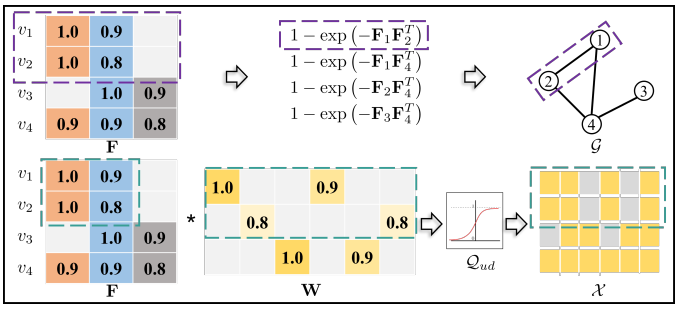


图 4. 示例说明如何根据社区归属嵌入矩阵$\mathbf{F}$重构网络结构 $\mathcal{G}$ 和节点元数据 $\mathcal{X}$。

在图 4 中，具有相同社区成员关系的节点 v_1 和 v_2 共享共同的节点元数据。接下来，我们将说明如何刻画节点元数据与社区结构之间的关系。这一关系建立在 [22] 中的属性建模之上，其假设是，将节点的社区成员归属纳入考量，有助于预测与该节点相关联的每个属性。

更正式地说，通过采用逻辑模型来构建二者之间的关系，我们可以将二值节点元数据$\mathcal{X}$中节点 $u\in\mathcal{V}$ 的第 d个属性（记为 $\mathcal{X}_{ud}$）建模为如下 sigmoid 函数：

$$\mathcal{Q}_{ud}=\frac{1}{1+\exp\left(-\sum_k\mathbf{F}_{uk}\mathbf{W}_{dk}\right)}, \tag{6}$$

其中，$\mathbf{F}_{uk}$ 是$\mathbf{F}_u$ 中的第 k个元素，$\mathbf{W}_{dk}$ 是权重矩阵$\mathbf{W}$ $\in\ \mathbb{R}^{D\times K}$中的第$(d,k)$个元素，表示社区 k 与第 d个节点属性的相关性。如图 4 所示，可以根据 $\mathcal{Q}_{ud}$重构相应的节点元数据。在此背景下，我们将损失函数 $\mathcal{L}_2(\mathbf{F})$ 定义为：

$$\begin{aligned}\mathcal{L}_2\left(\mathbf{F}\right)&=-\log\mathbb{P}\left(\mathcal{X}|\mathbf{F},\mathbf{W}\right)\\&=-\sum_{u,d}\left(\mathcal{X}_{ud}\log\mathcal{Q}_{ud}+\left(1-\mathcal{X}_{ud}\right)\log\left(1-\mathcal{Q}_{ud}\right)\right).\end{aligned} \tag{7}$$

4.2.3 模型训练

基于上述两种关系，我们建立如下重构损失：

$$\begin{aligned}\mathcal{L}\left(\mathbf{F}\right)&=\mathcal{L}_1(\mathbf{F})+\eta\mathcal{L}_2(\mathbf{F})\\&=-\sum_{(u,v)\in\mathcal{E}^{(t)}}\log\left(1-\exp\left(-\mathbf{F}_u\mathbf{F}_v^T\right)\right)+\sum_{(u,v)\notin\mathcal{E}^{(t)}}\mathbf{F}_u\mathbf{F}_v^T\\&\quad-\eta\sum_{u,d}\left(\mathcal{X}_{ud}\log\mathcal{Q}_{ud}+\left(1-\mathcal{X}_{ud}\right)\log\left(1-\mathcal{Q}_{ud}\right)\right),\end{aligned} \tag{8}$$

其中，$\eta\geq0$ 是用于平衡这两项的超参数。请注意，式 (8) 中的第一项和第二项分别旨在重构网络结构和节点元数据。我们通过最小化损失函数来更新图神经网络模型的参数 θ。

4.3 基于节点查询的网络探索

如第 3.3 节所述，在网络探索过程中，我们会尽可能多地发现每个查询节点的邻居。为此，我们关注如何选择节点查询 S_t。

尽管查询高度节点预计能使已探索网络增长得最快，但对于重叠社区检测任务而言，情况未必总是如此。如图2a所示，查询度最高的节点（即度为 6 的节点）可能只能发现数量有限的邻居。此外，由于事先不了解节点的邻居，这种拓扑感知查询策略无法实现。

在我们的研究中，节点查询策略建立在这样一个主张之上：每个查询节点不仅应属于更多社区，还应分布在多样化的社区中，以加快网络探索。如图2b和图2c所示，非重叠区域中的查询节点可能无法加快网络探索；然而，通过选择(i)位于重叠社区区域内且(ii)分布于多样化社区中的查询节点，我们能够探索每个社区内的不同区域，从而加快已探索网络的扩展。最终，这将提高社区检测的准确率，这一点已在第5.5.3节中通过实验证实。

在此背景下，我们按如下方式选择第$(t+1)$个节点进行查询，$t \in \{0, \cdots, T-1\}$：

$$v_t = \arg\max_{v} \left(\|\mathbf{F}_v\|_1 + \lambda \left(1 - \frac{1}{t+1} \sum_{u \in S_t} \text{sim}\left(\mathbf{F}_v, \mathbf{F}_u\right) \right) \right), \tag{9}$$

其中，$\|\cdot\|_1$ 表示向量的 L_1-范数，$\lambda \geq 0$ 是用于平衡这两项的超参数，$\text{sim}(\cdot, \cdot)$ 表示余弦相似度。这里，式(9)中的第二项用于控制所选节点，使其来自与先前查询节点所属社区不同的社区。

选择好查询节点 v_t 后，我们将其添加到序列 $S_{t+1} = S_t \cup \{v_t\}$ 中。随后，我们可以发现其邻居 $\mathcal{N}_{\mathcal{G}}(v_t)$，并通过更新已探索边集合 $\mathcal{E}_{t+1} = \mathcal{E}_t \cup \mathcal{E}(\mathcal{N}_{\mathcal{G}}(v_t) \ v_t)$ 来扩展可观测子网络。

4.4 网络推断

在查询过程中，已探索边集合 $\mathcal{E}_{t+1}$ 会因节点查询 S_{t+1} 而增加。随后，可以利用这些新获取的信息来更准确地预测其他边 [18]。为此，我们提出推断函数 $\mathcal{E}_I^{(t+1)} = f_I\ (S_{t+1}, \mathcal{E}_{t+1})$。请注意，我们的先前工作 [15] 不包括节点查询选择后的网络推断步骤。换言之，它没有充分利用通过网络探索获得的信息，这限制了准确检测社区的潜力。相比之下，在我们的研究中，网络 $\mathcal{G}^{(t+1)}$ 使用推断边 $\mathcal{E}_I^{(t+1)}$ 进行更新，并在每次迭代中逐渐逼近底层真实网络 $\mathcal{G}$。更新后的推断网络 $\mathcal{G}^{(t+1)}$ 可用于在下一次迭代中更好地学习社区归属嵌入矩阵 $\mathbf{F}$ 的表示。在此背景下，我们面临的主要挑战在于如何将网络推断步骤有效地整合到端到端社区检测框架中。

推断函数 $f_I\left(S_{t+1}, \mathcal{E}_{t+1}\right)$ 通过使用我们的 EC-SiamNet 模型进行设计，该模型结合网络结构和节点元数据，以在其上推断边

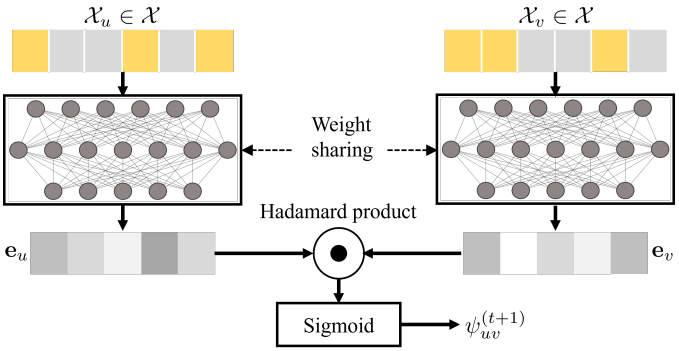


图 5. 我们用于网络推断的 EC-SiamNet 架构。

网络，而仅在初始网络推断步骤中使用节点元数据 $\mathcal{X}$（见第 4.1 节）。现在，我们解释如何训练EC-SiamNet，以及如何计算第 3.3节步骤 3 中的相似度分数 $\Psi^{(t+1)}$。在我们的研究中，我们利用了社交网络中的“同质性原则”，该原则指的是个体节点倾向于与相似节点 [52], 建立连接，并用于学习能够精确捕捉同质性效应的嵌入。经过训练的 EC-SiamNet 归纳式地映射新的输入数据（即未探索节点），这些节点与训练数据（即已探索节点）之间的连接信息是未知的。如图5所示，我们的 EC-SiamNet 架构由孪生多层感知机网络（多层感知机）组成，它们分别将对应于节点u 和 v的两个节点元数据 $\mathcal{X}_u$ 和 $\mathcal{X}_v$作为输入。多层感知机充当编码器，生成嵌入向量，分别表示为$\mathbf{e}_u$ 和 $\mathbf{e}_v$，对应于节点 u 和 v。与 [53], 中类似，我们计算两个嵌入向量$\mathbf{e}_u$ 和$\mathbf{e}_v$ 的哈达玛积，然后利用 sigmoid 输出层从该乘积中推断 $\psi_{uv}^{(t+1)} \in \Psi^{(t+1)}$。也就是说，我们计算

$$\psi_{uv}^{(t+1)} = \text{sigmoid}\left(\mathbf{e}_u \odot \mathbf{e}_v\right), \tag{10}$$

其中，$\odot$ 表示 Hadamard 积。

接下来，我们对 EC-SiamNet 的模型参数 $\mathbf{W}_{Sia}$ 进行优化。为此，我们基于 [54]中的方法构建对比损失。具体而言，对于随机梯度下降中使用的每个小批次，我们采样一对节点 $u, v \in \mathcal{V}$，根据$\mathcal{E}_{t+1}$可确定它们之间的连接关系。为利用真实网络的稀疏性，并平衡采样得到的正、负节点对，我们采用负采样 [55]。然后，我们的目标是通过最小化以下损失函数来求得参数 $\mathbf{W}_{Sia}$：

$$\begin{aligned}\mathcal{L}\left(\mathbf{W}_{Sia}, \mathcal{X}_v, \mathcal{X}_v\right) = &\left(1 - I_{uv}\right) \tfrac{1}{2} \left[\text{sim}\left(\mathbf{e}_u, \mathbf{e}_v\right)\right]^2 \\ &+ I_{uv} \tfrac{1}{2} \left[\max\left(0, r - \text{sim}\left(\mathbf{e}_u, \mathbf{e}_v\right)\right)\right]^2,\end{aligned} \tag{11}$$

其中，r 是所谓的“间隔”超参数 [54]；并且$I_{uv} = 1$ 当 $(u, v) \in \mathcal{E}_{t+1}$ 时，$I_{uv} = 0$ 在其他情况下。请注意，在 EC-SiamNet 优化阶段，我们利用了已探索边的信息（即参数 I_{uv}）。

式（11）中损失函数的设计动机有两点：1）在元数据相似的节点对的嵌入之间产生吸引力；2）在元数据不相似的节点对之间产生排斥力，以推动它们分离。

将它们分开，直到它们在嵌入空间中的距离超过某个半径 r。

通过采用所提出的 EC-SiamNet 模型计算相似度分数 $\Psi^{(t+1)}$，我们能够更精确地获得推断出的边集合 $\mathcal{E}_I^{(t+1)}$，然后将其并入网络 $\mathcal{G}^{(t+1)}$ 的边 $\mathcal{E}^{(t+1)}$ 中，即 $\mathcal{E}^{(t+1)} = \mathcal{E}_{t+1} \cup \mathcal{E}_I^{(t+1)}$。随后，在下一次迭代的步骤 **1** 中，将更新后的网络 $\mathcal{G}^{(t+1)}$ 用作图神经网络模型的输入，以获得增强的社区归属嵌入矩阵 **F** 表示。此过程会重复进行，直到达到第 T 个节点查询。

注释1。 下面说明 *EC-SiamNet* 如何在迭代过程中收敛。值得注意的是，随着查询过程的持续，我们能够从已探索网络中获取更多节点对用于训练。只要同质性原则成立，采用我们的*EC-SiamNet* 模型就能更准确地推断边，从而使网络 $\mathcal{G}^{(t)}$向其底层真实网络 $\mathcal{G}$收敛。这将带来更加精确的社区关联嵌入。

4.5 理论分析

在本小节中，我们从理论上分析 META-CODE 中通过节点查询进行网络探索的有效性。我们还从理论上展示 META-CODE 的计算复杂度。

4.5.1 META-CODE 中的理论发现

我们首先从分析角度展示节点查询策略的有效性。为便于分析，我们对底层网络 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 作出以下三个假设：

1) 底层网络 $\mathcal{G}$ 遵循 AGM，其中每个社区内形成边的概率均为 p。

2) 底层网络 $\mathcal{G}$ 由 $K(\geq 2)$个社区组成，社区规模差异受到如下约束：

$$N_{\max} - N_{\min} \leq \varepsilon, \tag{12}$$

其中， $N_{\max}$ 和 $N_{\min}$ 分别是每个社区中节点数量的最大值和最小值， $\varepsilon > 0$是一个任意小的常数。

3）我们根据节点所属的社区数量，将节点划分为 K 个子集。令 N_1、 N_2、 $\cdots$、 N_K 分别表示每个子集中的节点数量。于是，

$$\begin{cases} \frac{N_1}{N} \leq \frac{2}{3} \\ \mathbb{E}_v\left[\mathcal{D}\,|A_1\right] N_1 \geq \sum_{i=2}^{K} \mathbb{E}_v\left[\mathcal{D}\,|A_i\right] N_i, \end{cases} \tag{13}$$

其中， $\mathcal{D}$ 是节点的度分布， A_i表示某个节点 v 属于 i个社区这一事件。

假设 2 表明，所有社区的规模大致相同，这可以通过 Girvan–Newman 算法 [56] 实现。假设 3 强制要求节点不能过度属于单个社区或多个(≥ 2)社区。请注意，许多现实世界的网络（例如，Ego-Facebook [30]和引文网络 [7]）往往几乎满足假设3。现在，我们准备证明以下两个定理。

定理 4.1。 对于底层网络 $\mathcal{G} = (\mathcal{V},\ \mathcal{E})$中的任意节点 u 和 v，若它们分别属于 M 个和 M'个社区，且 $M > M'$，若 $\varepsilon \leq \frac{N_{\min}-1}{K} - 1$，则下列不等式成立：

$$\mathbb{E}_u\left[\mathcal{D}_M\right] \geq \mathbb{E}_v\left[\mathcal{D}_{M'}\right], \tag{14}$$

其中， $\mathbb{E}_u\left[\mathcal{D}_M\right]$ 和$\mathbb{E}_v\left[\mathcal{D}_{M'}\right]$ 分别是属于 M 个和 M' 个社区的节点的度分布期望。

证明。 将节点所属的第 i个社区记作 C_i。由于 AGM 通过在每个社区内独立地连接节点来生成网络$\mathcal{G}$ ，因此节点在给定社区 C_i中的度分布记为 Y_{C_i}，服从二项分布

$$Y_{C_i} \sim B\left(N_{C_i} - 1, p\right), \tag{15}$$

其中，N_{C_i} 是 C_i中的节点数。根据独立性和可加性，属于 M（分别为 M'）个社区的任意节点 u（分别为 v）的度分布 $\mathcal{D}_M$（分别为 $\mathcal{D}'_M$）仍然服从二项分布。因此，我们有

$$\begin{aligned} &\mathbb{E}_u\left[\mathcal{D}_M\right] - \mathbb{E}_v\left[\mathcal{D}_{M'}\right] \\ &= \sum_{i=1}^{M} (N_{C_i} - 1)p - \sum_{i=1}^{M'} (N_{C_i} - 1)p \\ &\geq M\left(N_{\min} - 1\right)p - M'\left(N_{\max} - 1\right)p \\ &\geq -M'\varepsilon p + \left(M - M'\right)\left(N_{\min} - 1\right)p \\ &\geq -M'\left(\tfrac{N_{\min}-1}{K} - 1\right)p + \left(M - M'\right)\left(N_{\min} - 1\right)p \\ &= \left(M - M' - \tfrac{M'}{K}\right)\left(N_{\min} - 1\right)p + M'p \\ &\geq 0, \end{aligned} \tag{16}$$

其中，第二个不等式由 (12) 得出，最后一个不等式则来自 $M' < M \leq K$。至此，该定理得证。 □

从定理 4.1 可以看出，选择属于更多社区的节点，预计能够实现更快的网络探索，因为可以发现查询节点的更多邻居。换言之，位于更多重叠社区区域内的节点与其他节点建立连接的可能性更高，从而能够实现高效的网络探索，这支持了我们在查询节点选择中的设计原则。

定理 4.2. 对于底层网络 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 中的 $M\,(\geq 2)$ 个社区中的任意节点 u 以及 $\mathcal{G}$中的任意节点 v，若 $\varepsilon \leq \frac{N_{\min}-1}{K} - 1$，则以下不等式成立：

$$\mathbb{E}_u\left[\mathcal{D}_M\right] \geq \mathbb{E}_v\left[\mathcal{D}\right], \tag{17}$$

其中， $\mathbb{E}_u\left[\mathcal{D}_M\right]$ 表示属于 M 社区的节点的度分布期望，而 $\mathbb{E}_v\left[\mathcal{D}\right]$ 表示 $\mathcal{G}$中所有节点的度分布期望。

证明。 在$\mathcal{G}$中随机选取一个节点 v ，其平均度记为 $\mathbb{E}_v[\mathcal{D}]$, 则有

$$\begin{aligned}\mathbb{E}_v[\mathcal{D}] &= \sum_{i=1}^{K}\mathbb{E}_v[\mathcal{D}\,|A_i]\,\mathbb{P}(A_i)\\ &= \frac{\sum_{i=1}^{K}\mathbb{E}_v[\mathcal{D}|A_i]N_i}{N}\\ &\le \frac{2\mathbb{E}_v[\mathcal{D}|A_1]N_1}{N}\\ &\le 2(N_{\max}-1)\,p\frac{K}{K+1}\\ &\le 2(N_{\max}-1)\,p\frac{N_{\min}-1}{N_{\min}+\varepsilon}\\ &\le 2(N_{\max}-1)\,p\frac{N_{\min}-1}{N_{\max}}\\ &\le \sum_{i=1}^{M}(N_{C_i}-1)p\\ &= \mathbb{E}_u[\mathcal{D}_M],\end{aligned} \tag{18}$$

其中，前两个不等式由（13）成立，第三个和第四个不等式分别由 $\varepsilon \le \frac{N_{\min}-1}{K}-1$ 和（12）得到。至此，该定理得证。

□

定理 4.2 进一步证实，与随机选择节点进行查询相比，所提出的节点查询策略能够更快地进行网络探索。我们的策略尝试选择重叠社区区域内的节点，这些区域的连接更为密集，从而加速已探索网络的扩展。

4.5.2 复杂度分析

为验证我们的 META-CODE 方法的可扩展性，我们通过建立以下定理，分析性地证明其计算复杂度。

定理 4.3。*所提出的 META-CODE 方法的计算复杂度为* $\mathcal{O}(|\mathcal{E}|)$。

证明。我们分析 META-CODE 中各步骤的计算复杂度。在步骤 1 中，使用图神经网络模型生成社区归属嵌入矩阵 $\mathbf{F}$，每个训练周期的复杂度为 $\mathcal{O}(L_1|\mathcal{E}|)$ [57]，其中 L_1 为图神经网络层的数量。在步骤 2 中，式 (9) 中的节点查询策略的复杂度为 $\mathcal{O}(NK+tNK)$，其中 N 和 K 分别为节点数和社区数。最后，在步骤 3 中，多层感知机的训练主导了该步骤的复杂度，其复杂度为 $\mathcal{O}(L_2|\mathcal{E}|)$ [58]，其中假设多层感知机中的模型参数 $|\mathbf{W}_{Sia}|$ 的规模不会随 $|\mathcal{E}|$增长， L_2 为多层感知机层的数量。因此，我们的 META-CODE 方法的总复杂度最终被界定为 $\mathcal{O}(|\mathcal{E}|)$。定理得证。

□

根据定理 4.3，可以看出，META-CODE的计算复杂度随底层网络中的边数线性地 扩展。

5 实验评估

在本节中，我们将介绍评估中使用的数据集以及六种用于对比的基准重叠社区检测方法，其中包括 META-CODE 的一个简单变体。在介绍性能指标和实验设置后，我们将全面评估 META-CODE 方法和这六种基准方法的性能。我们设计了广泛的实证研究，以回答以下六个关键研究问题（RQs）：

- **研究问题1**：META-CODE 相较于基准重叠社区检测方法，能在多大程度上提升社区检测的性能?
- **研究问题2**：META-CODE 中的各个模块分别如何提升社区检测的准确性?
- **RQ3**：我们的查询节点策略在快速网络探索方面有多有效?
- **RQ4**：推断网络与底层真实网络有多接近?
- **RQ5**：关键参数会如何影响 META-CODE 的性能?
- **RQ6**：META-CODE 的可扩展性如何随网络规模变化?

表 3 数据集的统计信息。

数据集	节点	边	社区	特征
Facebook	227	3,192	14	161
Twitter	249	4,372	8	1,174
Google+	1692	15,148	2	1319
工程	14,927	49,305	16	4,839
计算机科学	21,957	96,750	18	7,793

5.1 数据集

我们在涵盖多个领域的五个真实世界数据集上开展实验，其中包括三个社交网络和两个大型合著网络。为保持一致性，我们假设所有网络均为无向网络。表 3 汇总了各数据集的统计信息，包括节点数、边数、社区数和节点特征数。下面我们概述这些数据集的重要特征。[3]

Ego-Facebook、Ego-Twitter 和 Ego-Google+ [30]。Ego-Facebook、Ego-Twitter 和 Ego-Google+ 数据集分别是通过广度优先搜索（BFS）采样，从 Facebook、Twitter 和 Google+收集的、记录好友关系的社交网络[4] 。这些数据集中的节点和边分别代表用户和好友关系。节点特征是每位用户个人资料中的匿名信息。

合著网络-工程（工程）和合著网络-计算机科学（计算机科学）[7]。工程和计算机科学合著数据集是两个从微软学术图谱 [59]构建的合著网络。节点代表作者；若两位作者合著过论文，则在他们之间连一条边。节点特征表示每位作者论文的关键词，社区对应不同的研究领域。

3. 社交网络上的其他实验结果可在补充材料中找到。尽管还有其他社交网络和大型合著网络，但由于它们在所考察的数据集上呈现出类似的趋势，我们没有采用这些网络。

4. 由于从原始网络中采样了多个自我网络，我们在实验中使用 Facebook 348、Twitter 745823 和 Google $^{+}$ 100129275726588145876。为简洁起见，我们省略各自我网络的索引。

5.2 基准方法

由于尚无针对拓扑未知的网络进行重叠社区检测的先前工作，我们在实验中考虑了五种最先进的重叠社区检测方法。这些方法最初是为完全可观测网络开发的，因此需要额外的步骤来推断网络结构。为此，我们采用两种通过节点查询进行网络探索的采样策略：随机采样（RS）和深度优先搜索（DFS）。[5] 这些基准方法的实现方式与 META-CODE 类似：在给定节点查询预算的情况下，通过采样探索网络，同时迭代执行社区检测。我们按照各方法原始论文中描述的参数设置实现了所有这些基准方法。

BIGCLAM [5]。这是一种可扩展的方法，利用非负矩阵分解检测大型网络中的重叠社区。该方法通过将网络的邻接矩阵分解为两个非负矩阵，分别表示社区结构以及节点与社区的归属关系。

vGraph [21]。这是一种生成模型，可同时执行社区检测和节点表示学习。vGraph 学习社区归属和节点表示，以优化对局部网络结构和社区结构的保留。

CLARE [60]。这是一种半监督方法，首先通过在网络中寻找训练数据较少的带标签社区来定位潜在社区。随后，它使用经深度强化学习增强的社区重写器，以灵活的方式进一步调整和完善社区结构。

NOCD [7]。该方法通过结合使用BP模型和图神经网络，并最小化BP模型的负对数似然来学习社区归属关系。

DMoN [61]。这是一种受模块度启发的无监督池化方法，利用图神经网络进行图聚类。

请注意，CLARE、NOCD 和 DMoN利用网络结构和节点元数据来检测社区。此外，在我们的实验中，我们还考虑了 META-CODE的一个简单变体，即为拓扑未知的网络开发的 META-CODE$_{\text{sim}}$。

META-CODE$_{\text{sim}}$。该基线方法首先随机查询一定比例的节点，然后发现这些被查询节点的邻居，以进行网络探索。下一步，利用第 4.4节中的EC-SiamNet，基于已探索网络推断边。最后，采用第 4.2 节中的图神经网络模型生成社区关联嵌入**F**。

请注意，与原始的 META-CODE 不同，META- CODE$_{\text{sim}}$ 仅运行一次，不进行迭代。

5.3 性能指标

为了评估社区检测的性能，我们采用以下指标量化真实社区与检测到的社区之间的一致程度：

5. 虽然也可以采用偏置随机游走等其他复杂的采样策略，但本研究未采用这些策略，因为它们与随机采样和深度优先搜索具有相似的倾向。

归一化互信息（NMI） [62] 和平均 F_1 分数（AvgF_1） [5]。请注意，这两项指标的取值范围均为 $[0, 1]$，，且数值越高表示性能越好。NMI 和 AvgF_1的正式定义请参阅补充材料。

5.4 实验设置

我们介绍 META-CODE 方法中的神经网络（即图神经网络和多层感知机模型）的实验设置。我们采用具有 2 层的 GCN [27] 提取社区归属嵌入，并使用两层多层感知机作为 EC-SiamNet 的骨干网络，用于网络推断。我们使用 PyTorch Geometric [63]，（PyTorch 中用于几何深度学习的扩展库）实现 GCN 和多层感知机。我们将每个 GCN 和多层感知机的隐藏层维度分别设为 128 和 256。我们使用学习率分别为 0.001 和 0.05 的 Adam 优化器 [64] 训练图神经网络和多层感知机。此外，我们在指定范围内对超参数η 和 λ 进行调优，范围分别为 $\eta \in [1.0, 2.0]$ 和$\lambda \in [1.0, 4.0]$。

接下来，作为默认的实验设置，我们在所有数据集上对 META-CODE 开展实验，将固定的查询预算 T设置为底层网络中所有节点的百分比，从 10% 变化到 40%，增量为 10%，如表 4 所示。需要指出的是，在合著数据集（即 Engineering 和Computer Science）中，出于计算效率的考虑，我们每获得1% 的查询节点便执行一次网络推断，而不是在每次迭代中执行。这是因为，对于此类规模较大的数据集，仅从单个查询节点的邻居获取信息并不会显著提升网络推断效果。每项实验均进行 5 次，以评估平均性能。所有实验均使用 Intel (R)12-Core (TM) E5-1650v4 CPUs @ 3.60 GHz 和 64GB RAM 完成。

5.5 结果与分析

在本小节中，我们描述为回答上述研究问题（RQs）而开展的全面实验。为回答RQ1，我们提供所有数据集上的实验结果。为回答 RQ2–RQ5，鉴于其他数据集上的结果呈现出相似趋势，我们详细介绍 Facebook 和工程上的结果。

5.5.1 与基准方法的比较 *(RQ1)*

表4展示了我们 META-CODE 方法与五种最先进的重叠社区检测方法的性能比较，其中每种方法均分别结合了两种采样策略。比较基于两个性能指标——标准化互信息和AvgF_1，，在五个真实世界数据集上进行，其中从 N 个节点中查询不同比例的节点。我们注意到，前述所有方法中的超参数均根据每个具体数据集分别进行调优，以获得最佳性能。我们得出以下有意义的观察结果：

- 我们的META-CODE 方法无论针对何种数据集和性能指标，始终都显著优于所有基准方法。

表 4 META-CODE、META-CODE$_{sim}$与10 种竞争方法的性能比较：当在 N 节点中查询不同比例（%）的节点时，两个指标的结果（平均值 ± 标准差）。其中，最佳和次佳性能分别以粗体和下划线标示。

		10% 查询节点		20% 查询节点		30% 查询节点		40% 查询节点	
数据集	方法	NMI	平均F_1	NMI	平均F_1	NMI	平均F_1	NMI	平均F_1
Facebook	BIGCLAM+随机搜索	0.0263±0.0056	0.4349±0.0213	0.0539±0.0193	0.5025±0.0681	0.0703±0.0135	0.5896±0.0117	0.1083±0.0162	0.6503±0.1299
	BIGCLAM+深度优先搜索	0.0379±0.0057	0.4757±0.0115	0.0571±0.0211	0.5364±0.0351	0.0657±0.0197	0.5452±0.0733	0.0785±0.0139	0.5987±0.0448
	vGraph+RS	0.0559±0.0152	0.5700±0.0149	0.0914±0.0089	0.7046±0.0550	0.1332±0.0116	0.7233±0.0461	0.1573±0.0098	0.7708±0.0398
	vGraph+深度优先搜索	0.0601±0.0221	0.5526±0.0527	0.0943±0.0104	0.6614±0.0288	0.1030±0.0180	0.7079±0.0584	0.1272±0.0103	0.7297±0.0356
	CLARE+RS	0.0681±0.0110	0.6623±0.2354	0.0905±0.0117	0.6945±0.0277	0.1171±0.0130	0.7351±0.0204	0.1312±0.0113	0.7621±0.0209
	CLARE+深度优先搜索	0.0691±0.0131	0.6843±0.0200	0.0961±0.0137	0.7081±0.0162	0.1008±0.0121	0.7245±0.0251	0.1284±0.0122	0.7548±0.0130
	NOCD+RS	0.0704±0.0191	0.6503±0.0960	0.1022±0.0189	0.6627±0.0436	0.1422±0.0111	0.7445±0.0168	0.1958±0.0191	0.7652±0.0399
	NOCD+深度优先搜索	0.0594±0.0111	0.5471±0.0283	0.0965±0.0213	0.6568±0.0616	0.1329±0.0218	0.7352±0.0713	0.1683±0.0247	0.7885±0.0542
	DMoN+RS	0.0681±0.0297	0.6483±0.0374	0.0914±0.0209	0.7197±0.0184	0.1256±0.0119	0.7596±0.0325	0.1449±0.0132	0.7930±0.0428
	DMoN+深度优先搜索	0.0702±0.0178	0.6521±0.0374	0.0923±0.0184	0.7258±0.0446	0.1264±0.0152	07481±0.0494	0.1426±0.0152	0.7715±0.0358
	META-CODE$_{sim}$	0.0907±0.0226	0.7206±0.0473	0.1572±0.0335	0.7515±0.0282	0.2054±0.0188	0.8287±0.0408	0.2220±0.0142	0.8589±0.0428
	META-CODE	**0.0980**±0.0242	**0.7493**±0.0673	**0.1692**±0.0464	**0.8029**±0.0688	**0.2192**±0.0229	**0.8623**±0.0142	**0.2517**±0.0097	**0.8818**±0.0367
Twitter	BIGCLAM+RS	0.0307±0.0083	0.5891±0.0282	0.0341±0.0075	0.6102±0.0289	0.0399±0.0107	0.6409±0.0404	0.0450±0.0057	0.6523±0.0154
	BIGCLAM+深度优先搜索	0.0328±0.0031	0.6001±0.0162	0.0354±0.0112	0.6208±0.0206	0.0376±0.0056	0.6386±0.0099	0.0395±0.0066	0.6454±0.0281
	vGraph+RS	0.0305±0.0083	0.6102±0.0374	0.0391±0.0083	0.6321±0.0479	0.0456±0.0051	0.6605±0.0375	0.0478±0.0056	0.6703±0.0440
	vGraph+深度优先搜索	0.0337±0.0041	0.5739±0.0474	0.0431±0.0113	0.6309±0.0158	0.0465±0.0053	0.6470±0.0282	0.0474±0.0046	0.6544±0.0238
	CLARE+RS	0.0349±0.0213	0.5691±0.0139	0.0424±0.0340	0.6125±0.0309	0.0458±0.0105	0.6362±0.0261	0.0477±0.0197	0.6523±0.0347
	CLARE+深度优先搜索	0.0361±0.0112	0.5786±0.0168	0.0447±0.0107	0.6222±0.0322	0.0477±0.0112	0.6497±0.0254	0.0483±0.0120	0.6685±0.0275
	NOCD+RS	0.0402±0.0101	0.5113±0.0569	0.0566±0.0256	0.5910±0.0372	0.0568±0.0177	0.6224±0.0431	0.0894±0.0260	0.6355±0.0508
	NOCD+深度优先搜索	0.0518±0.0187	0.5487±0.0939	0.0973±0.0356	0.6314±0.0772	0.1101±0.0094	0.6518±0.0509	0.1181±0.0225	0.6645±0.0474
	DMoN+RS	0.0211±0.0039	0.4380±0.0498	0.0283±0.0085	0.4568±0.0538	0.0450±0.0183	0.5357±0.0631	0.0524±0.0156	0.5403±0.0786
	DMoN+深度优先搜索	0.0251±0.0027	0.4512±0.0751	0.0321±0.0053	0.4664±0.0587	0.0647±0.0172	0.6424±0.0413	0.0672±0.0221	0.6464±0.0395
	META-CODE$_{sim}$	0.0712±0.0172	0.6754±0.0229	0.1004±0.0135	0.7068±0.0171	0.1113±0.0106	0.7245±0.0349	0.1188±0.0125	0.7334±0.0313
	META-CODE	**0.0868**±0.0135	**0.7322**±0.0437	**0.1167**±0.0346	**0.7512**±0.0287	**0.1209**±0.0237	**0.7585**±0.0185	**0.1287**±0.0237	**0.7897**±0.0368
Google+	BIGCLAM+RS	0.0428±0.0023	0.3741±0.0318	0.0594±0.0062	0.4453±0.0146	0.0845±0.0285	0.4964±0.0365	0.1444±0.0140	0.5113±0.0365
	BIGCLAM+深度优先搜索	0.0448±0.0069	0.3756±0.0386	0.0689±0.0245	0.4540±0.0278	0.0836±0.0202	0.5069±0.0312	0.1436±0.0121	0.5102±0.0375
	vGraph+RS	0.0455±0.0051	0.4502±0.0307	0.0555±0.0126	0.4850±0.0142	0.0822±0.0119	0.4986±0.0213	0.0980±0.0131	0.5164±0.0127
	vGraph+深度优先搜索	0.0425±0.0081	0.4751±0.0099	0.0496±0.0105	0.4799±0.0167	0.0725±0.0158	0.4815±0.0188	0.0924±0.0158	0.5017±0.0144
	CLARE+RS	0.0452±0.0125	0.4813±0.0113	0.0625±0.0127	0.5009±0.0165	0.0897±0.0135	0.5239±0.0285	0.1097±0.0134	0.5354±0.0225
	CLARE+深度优先搜索	0.0467±0.0071	0.4958±0.0129	0.0677±0.0095	0.5122±0.0229	0.0889±0.0113	0.5197±0.0148	0.1009±0.0106	0.5285±0.0174
	NOCD+RS	0.1449±0.0146	0.4971±0.0272	0.1842±0.0258	0.5940±0.0292	0.2644±0.0189	0.6364±0.0362	0.2710±0.0189	0.6864±0.0409
	NOCD+深度优先搜索	0.1948±0.0395	0.5069±0.0459	0.2222±0.0264	0.5974±0.0218	0.2319±0.0107	0.6215±0.0218	0.2592±0.0183	0.6754±0.0369
	DMoN+RS	0.1452±0.0161	0.5465±0.0378	0.1681±0.0104	0.5867±0.0302	0.1698±0.0037	0.5882±0.0188	0.1707±0.0030	0.6174±0.0271
	DMoN+深度优先搜索	0.1510±0.0092	0.5457±0.0177	0.1685±0.0093	0.5868±0.0187	0.1691±0.0072	0.6209±0.0360	0.1732±0.0083	0.6551±0.0140
	META-CODE$_{sim}$	0.2058±0.0086	0.5528±0.0142	0.2563±0.0167	0.5893±0.0113	0.3024±0.0104	0.6479±0.0197	0.3226±0.0120	0.6911±0.0185
	META-CODE	**0.2327**±0.0024	**0.5715**±0.0303	**0.2766**±0.0116	**0.6056**±0.0104	**0.3359**±0.0205	**0.6651**±0.0314	**0.3478**±0.0105	**0.7269**±0.0236
Engineering	BIGCLAM+RS	0.0000±0.0000	0.1315±0.0386	0.0000±0.0000	0.2635±0.0207	0.0000±0.0000	0.3015±0.0531	0.0144±0.0133	0.4659±0.0317
	BIGCLAM+深度优先搜索	0.0029±0.0066	0.3460±0.0370	0.0194±0.0123	0.4530±0.0674	0.0406±0.0192	0.5160±0.0351	0.0590±0.0088	0.5980±0.0387
	vGraph+RS	0.0000±0.0000	0.1972±0.0282	0.0000±0.0000	0.3334±0.0080	0.0334±0.0042	0.4702±0.0188	0.0522±0.0116	0.5877±0.0913
	vGraph+深度优先搜索	0.0000±0.0000	0.3219±0.0084	0.0281±0.0114	0.4449±0.0642	0.0368±0.0054	0.4811±0.0820	0.0504±0.0194	0.5217±0.0137
	CLARE+RS	0.0109±0.0104	0.2641±0.0129	0.0194±0.0117	0.3747±0.0151	0.0541±0.0122	0.4387±0.0226	0.0674±0.0192	0.4954±0.0174
	CLARE+深度优先搜索	0.0114±0.0123	0.2954±0.0154	0.0189±0.0110	0.3528±0.0138	0.0529±0.0114	0.4219±0.0245	0.0546±0.0148	0.5789±0.0216
	NOCD+RS	0.1262±0.0219	0.7081±0.0277	0.2148±0.0113	0.7924±0.0367	0.2875±0.0182	0.8387±0.0238	0.3128±0.0175	0.8529±0.0370
	NOCD+深度优先搜索	0.1466±0.0322	0.7542±0.0598	0.2046±0.0185	0.7760±0.0638	0.2702±0.0158	0.8261±0.0304	0.3038±0.0172	0.8427±0.0178
	DMoN+RS	0.1782±0.0276	0.7562±0.0090	0.2248±0.0148	0.7854±0.0432	0.2534±0.0266	0.8095±0.0222	0.2693±0.0265	0.8242±0.0302
	DMoN+深度优先搜索	0.1822±0.0195	0.7691±0.0402	0.2204±0.0062	0.7898±0.0319	0.2396±0.0271	0.8131±0.0432	0.2431±0.0250	0.8237±0.0356
	META-CODE$_{sim}$	0.1858±0.0258	0.7584±0.0490	0.2324±0.0246	0.7941±0.0421	0.2671±0.0466	0.8105±0.0240	0.3032±0.0288	0.8316±0.0456
	META-CODE	**0.2072**±0.0198	**0.7838**±0.0366	**0.2785**±0.0430	**0.8208**±0.0341	**0.3080**±0.0256	**0.8548**±0.0207	**0.3288**±0.0326	**0.8694**±0.0280
Computer Science	BIGCLAM+RS	0.0000±0.0000	0.1164±0.0104	0.0000±0.0000	0.2167±0.0089	0.0000±0.0000	0.2914±0.0214	0.0042±0.0003	0.3624±0.0312
	BIGCLAM+深度优先搜索	0.0000±0.0000	0.1739±0.0213	0.0000±0.0000	0.2154±0.0111	0.0000±0.0000	0.3468±0.0341	0.0035±0.0004	0.3719±0.0272
	vGraph+RS	0.0017±0.0003	0.3140±0.0152	0.0028±0.0024	0.3379±0.0012	0.0050±0.0061	0.3543±0.0323	0.0106±0.0172	0.3579±0.0181
	vGraph+深度优先搜索	0.0042±0.0002	0.2942±0.0143	0.0046±0.0004	0.3151±0.0042	0.0051±0.0012	0.3308±0.0073	0.0056±0.0006	0.3419±0.0086
	CLARE+RS	0.0120±0.0118	0.3754±0.0218	0.0278±0.0141	0.4884±0.0284	0.0528±0.0101	0.5187±0.0235	0.0754±0.0147	0.5478±0.0238
	CLARE+深度优先搜索	0.0145±0.0111	0.4085±0.0176	0.0257±0.0132	0.4548±0.0213	0.0492±0.0130	0.4999±0.0255	0.0698±0.0127	0.5265±0.0247
	NOCD+RS	0.2568±0.0411	0.7125±0.0226	0.3724±0.0089	0.7810±0.0144	0.3972±0.0281	0.8174±0.0173	0.4205±0.0221	0.8438±0.0164
	NOCD+深度优先搜索	0.1741±0.0302	0.6738±0.0377	0.3018±0.0118	0.7473±0.0145	0.3617±0.0275	0.7947±0.0173	0.4074±0.0165	0.8250±0.0270
	DMoN+RS	0.2230±0.0656	0.7569±0.0541	0.2583±0.0455	0.7719±0.0533	0.2914±0.0123	0.8095±0.0411	0.3087±0.0379	0.8128±0.0420
	DMoN+深度优先搜索	0.2077±0.0550	0.7570±0.0389	0.2414±0.0438	0.7787±0.0375	0.2784±0.0243	0.7799±0.0310	0.2961±0.0210	0.8018±0.0419
	META-CODE$_{sim}$	0.3138±0.0177	0.7882±0.0258	0.3408±0.0313	0.8113±0.0289	0.3799±0.0268	0.8437±0.0379	0.4223±0.0261	0.8595±0.0254
	META-CODE	**0.3507**±0.0204	**0.8124**±0.0381	**0.4005**±0.0169	**0.8408**±0.0322	**0.4237**±0.0360	**0.8611**±0.0480	**0.4408**±0.0223	**0.8820**±0.0389

- 第二优的表现者往往是 META-CODE$_{相似}$。

META-CODE$_{相似}$ 在几乎所有情况下都优于五种最先进的方法，因为它包含经过精心设计的模块，其中包括：1）借助EC-SiamNet 进行网络推断；以及 2）由图神经网络辅助的社区归属

使用我们的重构损失进行嵌入。然而，由于缺乏通过审慎的节点查询和迭代步骤进行网络探索，META-CODE$_{sim}$ 始终劣于原始META-CODE。换言之，META-CODE$_{sim}$ 不仅忽视了节点查询选择加速网络的潜力

表5 Facebook 和工程上的消融实验。此处，表现最佳和次佳的方法分别以粗体和下划线标出。

		10% 查询节点		20% 查询节点		30% 查询节点		40% 查询节点	
数据集	方法	NMI	$AvgF_1$	NMI	$AvgF_1$	NMI	$AvgF_1$	NMI	$AvgF_1$
Facebook	META-CODE-1	0.0783±0.0199	0.5562±0.0151	0.1151±0.0312	0.6230±0.0137	0.1442±0.0329	0.6711±0.0361	0.1647±0.0081	0.8033±0.0446
	META-CODE-2	0.0669±0.0187	0.5335±0.0170	0.0755±0.0352	0.5846±0.0247	0.0981±0.0255	0.6352±0.0394	0.1368±0.0396	0.6958±0.0641
	META-CODE-3	0.0814±0.0321	0.7203±0.0516	0.1588±0.0245	0.7941±0.0280	0.2108±0.0197	0.8098±0.0696	0.2487±0.0211	0.8709±0.0231
	META-CODE-4	0.0791±0.0198	0.6842±0.0343	0.1132±0.0167	0.7221±0.0567	0.1578±0.0169	0.7923±0.0493	0.2003±0.0164	0.8218±0.0085
	META-CODE	**0.0980**±0.0242	**0.7493**±0.0673	**0.1692**±0.0464	**0.8029**±0.0688	**0.2192**±0.0229	**0.8623**±0.0142	**0.2517**±0.0097	**0.8818**±0.0367
Engineering	META-CODE-1	0.1371±0.0108	0.5729±0.0364	0.1532±0.0276	0.6031±0.0219	0.1887±0.0281	0.6598±0.0210	0.2145±0.0294	0.6946±0.0394
	META-CODE-2	0.0921±0.0135	0.5245±0.0271	0.1265±0.0217	0.5725±0.0428	0.1629±0.0297	0.6214±0.0239	0.1968±0.0204	0.6611±0.0255
	META-CODE-3	0.1874±0.0237	0.7619±0.0331	0.2584±0.0101	0.8104±0.0238	0.2934±0.0274	0.8487±0.0134	0.3207±0.0190	0.8602±0.0207
	META-CODE-4	0.1575±0.0371	0.7613±0.0193	0.2177±0.0207	0.8079±0.0193	0.2921±0.0169	0.8431±0.0177	0.3191±0.0302	0.8581±0.0175
	META-CODE	**0.2072**±0.0198	**0.7838**±0.0866	**0.2785**±0.0430	**0.8208**±0.0341	**0.3080**±0.0256	**0.8548**±0.0207	**0.3288**±0.0326	**0.8692**±0.0580

探索过程，以及通过迭代过程逐步提升社区检测性能。

- 在最先进的方法中，大多数情况下，表现最佳的竞争方法往往是使用 RS 的 NOCD，因为 NOCD 也借助节点元数据，通过图神经网络生成社区关联嵌入，能够精准地发现社区。与 NOCD 相比，所提出的 META-CODE 和 $\text{META-CODE}_{\text{sim}}$ 方法使用重构损失，同时考虑结构–社区和元数据–社区关系来生成社区关联嵌入，因此性能优于 NOCD。

- DMoN 利用模块度训练图神经网络，而模块度高度依赖网络结构。这可能导致其性能相对低于 NOCD，尤其是在仅查询少数节点时。

- 使用工程数据集时，META-CODE（X）与 $\text{META-CODE}_{\text{相似性}}$（$Y$）之间的性能差距最大；就标准化互信息而言，最大提升率为 19.84%，其中提升率（%）由 $\frac{X-Y}{Y} \times 100$ 给出。此外，使用 Facebook 数据集时，我们的META-CODE 方法与最佳先进方法之间的最大提升率在标准化互信息方面为 65.55%。

- 随着查询节点的百分比增加，META-CODE 在标准化互信息和 $AvgF_1$ 方面的提升率趋于下降。例如，在计算机科学数据集中，当查询节点的百分比分别从 10% 增至 20%、从 20% 增至 30% 以及从 30% 增至 40% 时，标准化互信息的提升率分别为 14.20%、5.79% 和 4.03%。这是因为，随着查询节点的百分比增加，网络中未探索部分的规模会缩小，已探索网络也会逐渐趋近于底层真实网络。

- 让我们比较基准方法所采用的两种网络探索采样策略。当查询的节点数量较少（即探索的网络规模较小时），将各基准方法与深度优先搜索（DFS）结合，通常比使用 RS 的效果更好。然而，当查询的节点数量足够多时，使用 RS 进行网络探索通常会取得更好的效果。例如，在 Facebook 上，当查询节点占比为 10–20% 时，采用深度优先搜索（DFS）的 vGraph 优于采用 RS 的 vGraph；而当查询节点占比超过 30% 时，采用 RS 的 vGraph 表现更好。这是因为深度优先搜索（DFS）能够更快地在局部探索单个社区内的子网络，这有利于社区检测的早期阶段。另一方面，随着查询节点数量增加，某个社区内的局部网络可能已基本探索完毕，此时最好转而探索底层网络的其他部分。这印证了这样的说法：META-CODE 的节点查询策略经过专门设计，能够选取分布在多样化社区中的节点，从而加快探索速度。

5.5.2 META-CODE 中组件的影响（**RQ2**）

为探究各组件在所提出的 META-CODE 方法成功中所起的作用，我们通过移除或替换该方法中的各个模块开展了消融研究。

- META-CODE：这对应于原始的 META-CODE 方法。
- META-CODE-1：META-CODE 中的图神经网络模型被替换为 BIGCLAM [5] 以生成社区归属嵌入。
- META-CODE-2：META-CODE 中的图神经网络模型被替换为 MLP，以生成社区关联嵌入。
- META-CODE-3：META-CODE 中的节点查询策略被替换为用于网络探索的 RS。
- META-CODE-4：移除了 META-CODE 中的网络推断步骤。也就是说，它变成了我们的先前工作 [15]。

表 5 展示了原始 META-CODE 与其四种变体在 Facebook 和工程数据集上的标准化互信息和 $AvgF_1$ 性能对比。我们的发现如下：

- 原始 META-CODE 方法始终展现出相较于其他变体的潜在增益，这表明各个模块协同作用，在发现社区的过程中都发挥着关键作用。

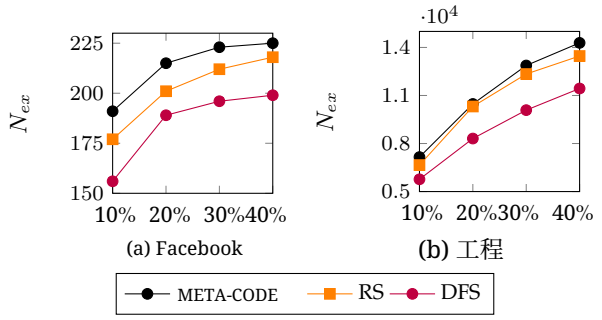


图 6. Facebook 和工程中，在 N 个节点里不同百分比（%）的查询节点对应的已探索节点数 N_{ex}。

- 对于两个数据集而言，META-CODE 与 META-CODE-2 之间的性能差距相较于其他变体要大得多。

- META-CODE-1 的表现优于 META-CODE-2，这可归因于 BIGCLAM 最初是为社区检测而设计的，而 MLP 无法准确捕捉结构与社区之间的关系。

- META-CODE 的效果显著优于 META-CODE-4（即我们的先前工作 [15]），这表明网络推断步骤对于获得更准确的社区检测结果至关重要。与 META-CODE-4 相比，META-CODE 充分利用了网络探索获取的结构信息，因此社区检测性能更优。

5.5.3 已探索子网络的增长（**RQ3**）
为通过实证验证节点查询策略的有效性，我们在 Facebook 和 Engineering 数据集上进行了实验，并绘制了图 6 中已探索节点数量（记为 N_{ex}）与查询节点百分比之间的关系。我们将所提出的查询策略与另外两种情况进行比较：在 META-CODE 中，使用两种采样策略（即 RS 和深度优先搜索）进行网络探索，而非采用文献 (9) 中提出的策略。我们的发现如下：

- 与其他采样方法相比，基于已发现的社区归属关系制定的查询策略始终更具优势，因为它能更快地探索底层网络中未知的部分。

- 随着查询节点数量的增加，Nex forRS 的增长率甚至高于深度优先搜索（DFS），因为通过 RS 查询已探索子网络之外的节点（而非通过 DFS 扩展已探索子网络）更加有效。

5.5.4 推断网络的收敛（**RQ4**）
分析推断网络收敛至底层真实网络的过程很有价值，这对于通过迭代获得更精确的社区关联嵌入至关重要。在表6中，我们给出了曲线下面积（AUC）分数，用于衡量 META-CODE-4 中的推断网络 $\mathcal{G}^{(t)}$ in（在其中提及）之间的相似性。

表6 META-CODE-4 和 META-CODE 中推断网络 $\mathcal{G}^{(t)}$ 与底层网络 $\mathcal{G}$之间的 AUC 分数。

数据集	查询比例	META-CODE-4	META-CODE	Gain
Facebook	10%	0.6015	**0.6823**	13.43%
	20%	0.6681	**0.7304**	9.32%
	30%	0.7310	**0.7864**	7.57%
	40%	0.7799	**0.8302**	6.44%
工程	10%	0.5744	**0.6220**	8.28%
	20%	0.6408	**0.6794**	6.02%
	30%	0.6907	**0.7308**	4.89%
	40%	0.7495	**0.7771**	3.68%

第 5.5.2 节）以及 META-CODE 和底层网络$\mathcal{G}$ 在使用 Facebook 和工程数据集时，对于不同比例的查询节点的结果。通过比较两个相应的邻接矩阵，可以计算 $\mathcal{G}^{(t)}$ 与 $\mathcal{G}$ 之间的 AUC 分数，其中来自 $\mathcal{G}^{(t)}$ 和 $\mathcal{G}$ 的邻接矩阵分别可视为预测结果和真实标签。较高的 AUC 表示所比较的两个网络之间具有更高的相似性。我们的研究结果如下：

- 使用 Facebook 数据集时，META-CODE-4 与 META-CODE 之间的性能差距最大；当查询 10% 的节点时，AUC 得分的最大提升率达到 13.43%。- 随着节点查询比例的增加，$\mathcal{G}^{(t)}$ 之间的 AUC 得分趋于上升。这表明 EC-SiamNet 能够有效地准确推断边，从而使推断网络快速收敛至底层真实网络，正如备注 1 中所述。

5.5.5 超参数的影响（**研究问题 5**）
图 7 和图 8 展示了两个关键超参数 η 和 λ 对 META-CODE 性能的影响，性能以使用 Facebook 和 Engineering 数据集时的标准化互信息衡量。当改变某个超参数以清晰揭示其影响时，另一个参数则设为第 5.4 节中讨论的枢轴值。我们的发现如下。

- **.....的影响** η：从图 7 可以看出，Facebook 和 Engineering 上的最高标准化互信息分别在 $\eta = 1.5$ 和 $\eta = 1.0$ 处取得。当 $\eta = 0$ 时，式（8）中的损失函数退化为 [7], 中的情形，此时仅考虑结构与社区之间的关系，而不考虑节点元数据，导致性能不尽如人意。另一方面，将 η 设为较高值会使两个数据集上的性能都相对较低，因为训练过程中过度强调了元数据与社区之间的关系。因此，恰当地确定 η 的取值对于确保令人满意的性能至关重要。

- **.....的影响** λ：从图 8 可以看出，在 Facebook 和 Engineering 数据集上，标准化互信息分别在 $\lambda = 2.0$ 和 $\lambda = 3.0$ 时达到最大值。当 $\lambda = 0$时，我们在（9）中的节点查询策略只会选择属于更多社区的节点。相反，将λ 设为较高值会对性能产生负面影响，因为这会过度强调对

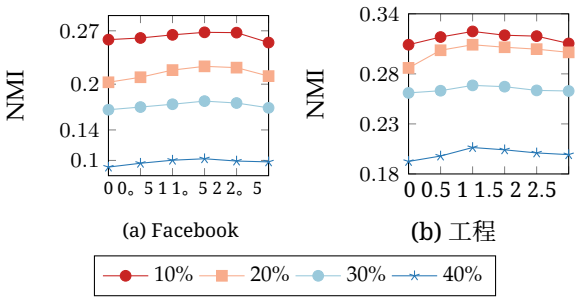


图 7. 在 Facebook和工程上查询不同百分比的节点时，META-CODE 中不同 η 对应的标准化互信息。

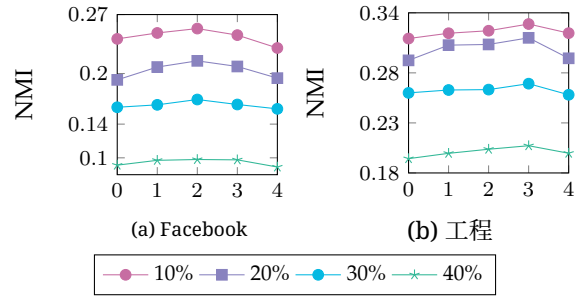


图8. 在 Facebook 和工程数据集上，不同比例的节点被查询时，META-CODE 中不同 λ 对应的标准化互信息。

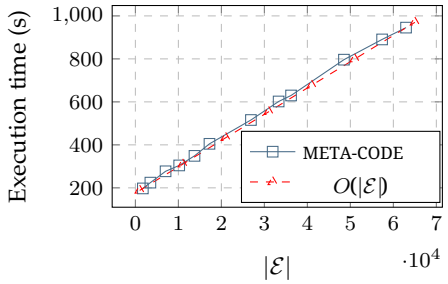


图 9. META-CODE 的计算复杂度，其中展示了执行时间与 $|\mathcal{E}|$ 的关系图。

分布在不同社区中的节点的选择。因此，需要根据数据集适当确定λ的取值。

5.5.6 计算复杂度（RQ6）

为了通过实证验证 META-CODE 方法的可扩展性，我们测量了在使用 Erd˝os–Rényi 模型 [65] 生成的合成网络上的执行时间。这些网络的节点数固定，边数各不相同。Erd˝os–Rényi 模型 $G(n,\ q)$ 可以通过以概率 q在两个节点之间创建边来生成各种网络。我们生成了一组具有 $n = 2000$ 和$q = \{0.002, 0.003, 0.006, 0.008, 0.01, 0.02, 0.03, 0.04\}$的网络。图 9 展示了随着边数增加，META-CODE 单次迭代的执行时间（以秒为单位），其中包括图神经网络和 MLP 模型的训练时间以及查询节点选择时间。虚线表示依据定理 4.3 得出的 $|\mathcal{E}|$线性扩展趋势。可以看出，我们的实证评估与理论分析一致。

6 总结

本文解决了拓扑未知的网络中的重叠社区检测这一尚未探索的问题。为应对这一挑战，我们提出了 META-CODE，这是一种用于重叠社区检测的新型统一框架，它迭代执行以下三个步骤：1）社区归属嵌入；2）通过节点查询探索网络；3）网络推断。通过在五个真实世界数据集上进行大量实验，我们证明了：（a）META-CODE 优于最先进的重叠社区检测方法，与表现最佳的现有竞争方法相比，在标准化互信息方面最高提升了 65.55%；（b）META-CODE 各模块的作用；（c）通过分析已探索子网络的增长情况，验证了我们的节点查询策略的有效性；（d）推断网络向底层真实网络收敛；（e）关键超参数的影响，包括 η 和 λ；以及（f）META-CODE 的可扩展性。我们的理论分析还揭示了：（a）所提出的节点查询策略的影响；以及（b）META-CODE 的可扩展性。

未来研究的潜在方向包括：针对不确定网络环境下的重叠社区检测，设计更精妙的图神经网络架构，从而提升节点级社区归属嵌入的表达能力。

致谢

Won-Yong Shin 的研究工作得到了韩国政府（MSIT）资助的韩国国家研究基金会（NRF）项目支持（编号：2021R1A2C3004345、RS-2023-00220762）。Ming Li 感谢浙江省重点研发计划（编号：2024C03262）和中国国家自然科学基金（编号：62172370、U21A20473）的资助。本文部分内容曾发表于 10 月在美国佐治亚州亚特兰大举行的 ACM信息与知识管理国际会议，年份为 2022 [15]。

参考文献

[1] X.Su、S. Xue、F.Liu、J. Wu、J. Yang、C. Zhou、W. Hu、C. Paris、S. Nepal、D.Jin等，“基于深度学习的社区检测综合调查”，*IEEE*神经网络与学习系统汇刊，第1–21页，2022年3月。[2] E. Abbe，“社区检测与随机块模型：近期进展”，《机器学习研究杂志》，第18卷，第1期，第6446–6531页，2017年4月。[3] G. Palla、I. Derényi、I.Farkas和T. Vicsek，“揭示自然界和社会中复杂网络的重叠社区结构”，《自然》，第435卷，第7043期，第814–818页，2005年6月。[4] V. Nicosia、G. Mangioni、V.Carchiolo和M.Malgeri，“将模块度的定义扩展到具有重叠社区的有向图”，《统计力学：理论与实验》，第2009卷，第03期，第P03024页，2009年3月。[5] J.Yang和J. Leskovec，“大规模重叠社区检测：一种非负矩阵分解方法”，载于*WSDM*论文集，2013年，第587–596页。[6] F. Ye、C.Chen和Z. Zheng，“用于社区检测的深度自编码器式非负矩阵分解”，载于*CIKM*论文集，2018年，第1393–1402页。[7] O.Shchur和S. Günnemann，“利用图神经网络进行重叠社区检测”，载于第*25*届*ACM SIGKDD*图深度学习研讨会，2019年。

[8] N. Mehta、L. C.Duke 和 P. Rai，“随机块模型与图神经网络相遇”，载于《ICML 会议录》，2019年，第 4466–4474 页。[9] T.W. Valente 和 P. Pumpuang，“识别意见领袖以促进行为改变”，《健康教育与行为》，第 34卷，第6期，第 881–896 页，2007年 12 月。[10] R. Dey、Z. Jelveh和K. Ross，“Facebook 用户变得更加注重隐私：一项大规模研究”，载于《第 10 届IEEE 国际普适计算与通信会议研讨会论文集》，2012年，第 346–352 页。[11] M.S.Granovetter，“弱联系的力量”，《美国社会学杂志》，第78 卷，第6期，第 1360–1380 页，1973 年 5月。[12] M.Kim 和J.Leskovec，“网络补全问题：推断网络中缺失的节点和边”，载于《第11届 SIAM国际数据挖掘会议论文集》，2011 年，第47–58 页。[13] C. Tran、W.-Y. Shin、A. Spitz 和 M. Gertz，“DeepNC：深度生成式网络补全”，IEEE《模式分析与机器智能汇刊》，第44 卷，第4期，第 1837–1852 页，2020 年 4 月。[14] C.Tran 、 W.-Y. Shin 和 A. Spitz，“部分可观测社交网络中的社区检测”，《ACM 知识发现与数据汇刊》，第16卷，第 2 期，第 1–24 页，2021年 7 月。[15] Y. Hou、C. Tran和 W.-Y.Shin，“META-CODE：通过探索性学习在拓扑未知的网络中进行社区检测”，载于《CIKM会议录》，2022年，第 4034–4038页。[16] B.Wilder、L.Onasch-Vera、J. Hudson、J. Luna、N. Wilson、R. Pe-tering、D.Woo、M. Tambe和 E. Rice，“在实际场景中端到端地最大化影响力”，载于《AAMAS会议录》，2018年，第 1414–1422页。[17] H.Kamarthi、P. Vijayan、B. Wilder、B. Ravindran和M. Tambe，“未知社交网络中的影响力最大化：学习有效图采样的策略”，载于《AAMAS 会议录》，2020年，第 575–583 页。[18] C. Tran、W.-Y. Shin 和 A. Spitz，“IM-META：利用未知拓扑网络中的节点元数据进行影响力最大化”，《IEEE 网络科学与工程汇刊》，2024 年。[19] B.Yan 和S.Gregory，“在不完整网络中寻找缺失的边和社区”，《物理学杂志 A：数学与理论》，第 44卷，第 49期，第 495102 页，2011 年 11 月。[20] J. Yang 和 J.Leskovec，“用于重叠网络社区检测的社区归属图模型”，载于《ICDM会议录》，2012年，第 1170–1175 页。[21] F.-Y. Sun、M. Qu、J. Hoffmann、C.-W. Huang 和J. Tang，“v Graph：用于联合社区检测和节点表示学习的生成模型”，载于《NeurIPS 会议录》，2019年，第 512–522 页。[22] J. Yang、J. McAuley 和J. Leskovec，“具有节点属性的网络中的社区检测”，载于《ICDM会议录》，2013年，第 1151–1156 页。[23] M.Zhou，“用于重叠社区检测和链接预测的无限边划分模型”，载于《AISTATS会议录》，2015年，第 1135–1143 页。[24] V. Leroy、B. B. Cambazoglu和 F. Bonchi，“冷启动链接预测”，载于《KDD 会议录》，2010年，第 393–402 页。[25] M. Kosinski、D. Stillwell和T. Graepel，“从人类行为的数字记录中可以预测私人特征和属性”，《美国国家科学院院刊》，第110 卷，第 15期，第 5802–5805 页，2013年 3 月。[26] E. Rice、E. Tulbert、J. Cederbaum、A. BarmanAdhikari和N. G.Milburn，“动员无家可归青年参与 HIV 预防：对面对面和在线社交网络干预可接受性的社会网络分析”，《健康教育研究》，第27卷，第 2 期，第226–236 页，2012 年。[27] T.N. Kipf和 M. Welling，“利用图卷积网络进行半监督分类”，载于《ICLR 会议录》（海报），2017 年。[28] P. Veli˘ckovi´c、G.Cucurull、A. Casanova、A. Romero、P. Lio和 Y. Bengio，“图注意力网络”，载于《ICLR会议录》（海报），2018年。[29] B.Wilder、N. Immorlica、E. Rice 和 M. Tambe，“在未知社交网络中最大化影响力”，载于《AAAI会议录》，2018年，第4743–4750 页。[30] J.Mcauley和 J. Leskovec，“在自我网络中发现社交圈”，《ACM知识发现与数据汇刊》，第8 卷，第 1 期，第 1–28 页，2014 年 2 月。[31] J. Leskovec和 J. Mcauley，“学习在自我网络中发现社交圈”，载于《NeurIPS 会议录》，2012 年，第539–547 页。[32] X. Zhang、H. Liu、Q. Li和 X.-M. Wu，“通过自适应图卷积进行属性图聚类”，载于《IJCAI会议录》，2019年，第4327–4333 页。[33] X.Wang、J. Li、L. Yang 和H. Mi，“基于图卷积网络的属性网络社区检测无监督学习”，《神经计算》，第 456 卷，第 147–155 页，2021 年 5 月。

[34] C. Park、J. Kim、J. Ham 和 J. Yu，“属性网络嵌入的监督式多重图聚合”，载于《AAAI会议论文集，2020年，第5371–5378页。[35] X. Fu、J. Zhang、Z. Meng 和 I. King，《MAGNN：用于异构图嵌入的元路径聚合图神经网络》，载于 WWW 会议论文集，2020年，第2331–2341页。[36] J. J. Choong、X. Liu 和 T. Murata，《利用变分自编码器学习社区结构》，载于 ICDM 会议论文集，2018年，第69–78页。[37] D. He、Y. Song、D. Jin、Z. Feng、B. Zhang、Z. Yu 和 W. Zhang，《以社区为中心的图卷积网络：用于无监督社区检测》，载于 IJCAI 会议论文集，2021年，第3515–3521页。[38] J. Sun、W. Zheng、Q. Zhang 和 Z. Xu，《用于属性网络社区检测的图神经网络编码》，IEEE《控制论汇刊》，第52卷，第8期，第7791–7804页，2021年2月。[39] B. Yan 和 S. Gregory，《使用边预测方法检测网络中的社区结构》，《统计力学：理论与实验》，第2012卷，第09期，文章编号P09008，2012年9月。[40] N. Barbieri、F. Bonchi 和 G. Manco，《基于影响力的网络无关社区检测的高效方法》，ACM《智能系统与技术汇刊》，第8卷，第2期，第1–31页，2016年12月。[41] T. Hoffmann、L. Peel、R. Lambiotte 和 N. S. Jones，《未观测边的网络社区检测》，《科学进展》，第6卷，第4期，文章编号eaav1478，2020年1月。[42] M. Brautbar 和 M. J. Kearns，《用于在大型网络中寻找有趣个体的局部算法》，载于创新计算机科学（ICS）会议论文集，2010年，第188–199页。[43] Y. Higashikawa、N. Katoh、S. Langerman 和 S.-i. Tanigawa，《多名搜索者针对环和树的在线图探索算法》，《组合优化杂志》，第28卷，第2期，第480–495页，2014年12月。[44] B. Settles，《主动学习文献综述》，2009年。[45] A. J. Joshi、F. Porikli 和 N. P. Papanikolopoulos，《用于多类别图像分类的可扩展主动学习》，IEEE《模式分析与机器智能汇刊》，第34卷，第11期，第2259–2273页，2012年1月。[46] A. Yadav、H. Chan、A. X. Jiang、H. Xu、E. Rice 和 M. Tambe，《利用社交网络帮助无家可归者收容所：不确定性下的动态影响力最大化》，载于 AAMAS 会议论文集，2016年，第740–748页。[47] J. Bromley、I. Guyon、Y. LeCun、E. Säckinger 和 R. Shah，《使用“孪生”时延神经网络进行签名验证》，载于 NeurIPS 会议论文集，1993年，第737–744页。[48] S. Chopra、R. Hadsell 和 Y. LeCun，《通过判别式学习相似度度量及其在人脸验证中的应用》，载于 CVPR 会议论文集，2005年，第539–546页。[49] X. Liu、J. Van De Weijer 和 A. D. Bagdanov，《利用自监督排序学习发掘卷积神经网络中的未标注数据》，IEEE《模式分析与机器智能汇刊》，第41卷，第8期，第1862–1878页，2019年2月。[50] Y. Qi、S. Zhang、L. Qin、Q. Huang、H. Yao、J. Lim 和 M.-H. Yang，《利用深层特征进行视觉跟踪》，IEEE《模式分析与机器智能汇刊》，第41卷，第5期，第1116–1130页，2018年4月。[51] A. P. Streich、M. Frank、D. Basin 和 J. M. Buhmann，《布尔数据的多重分配聚类》，载于 ICML 会议论文集，2009年，第969–976页。[52] J. Tang、H. Gao、X. Hu 和 H. Liu，《利用同质性效应进行信任预测》，载于 WSDM 会议论文集，2013年，第53–62页。[53] Z. Wang、C. Chen 和 W. Li，《用于链接预测的预测性网络表示学习》，载于 SIGIR 会议论文集，2017年，第969–972页。[54] R. Hadsell、S. Chopra 和 Y. LeCun，《通过学习不变映射实现降维》，载于 CVPR 会议论文集，2006年，第1735–1742页。[55] M. Armandpour、P. Ding、J. Huang 和 X. Hu，《网络嵌入的鲁棒负采样》，载于 AAAI 会议论文集，2019年，第3191–3198页。[56] M. Girvan 和 M. E. Newman，《社会网络和生物网络中的社区结构》，《美国国家科学院院刊》，第99卷，第12期，第7821–7826页，2002年4月。[57] Z. Wu、S. Pan、F. Chen、G. Long、C. Zhang 和 S. Y. Philip，《图神经网络综合综述》，IEEE《神经网络与学习系统汇刊》，第32卷，第1期，第4–24页，2021年3月。[58] F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Van-derplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay，《Scikit-learn：Python 机器学习》，《机器学习研究杂志》，第12卷，第2825–2830页，2011年10月，https://scikit-learn.org/stable/modules/neural_networks_supervised.html。

[59]微软，《微软学术图谱发布：研究就此开始！》，2015年，
h
t
t
p
s://
w
w
w.microsoft.com/en-us/research/blog/announcing-the-microsoft-academic-graph-let-the-research-begin/。
[60] X. Wu、Y. Xiong、Y. Zhang、Y. Jiao、C. Shan、Y. Sun、Y. Zhu 和 P. S. Yu，《CLARE：一种半监督社区检测算法》，载于 *KDD* 会议论文集，2022年，第2059–2069页。[61] A. Tsitsulin、J. Palowitch、B. Perozzi 和 E. Müller，《利用图神经网络进行图聚类》，《机器学习研究杂志》，第24卷，第127期，第1–21页，2023年。[62] A. F. McDaid、D. Greene 和 N. Hurley，《用于评估重叠社区发现算法的归一化互信息》，*arXiv* 预印本 *arXiv:1110.2515*，2011年。[63] M. Fey 和 J. E. Lenssen，《使用 PyTorch Geometric 快速进行图表示学习》，载于 第7届 *ICLR* 图与流形表示学习研讨会论文集，2019年。[64] D. P. Kingma 和 J. Ba，《Adam：一种随机优化方法》，载于 *ICLR* 会议论文集（海报），2015年。[65] P. Erd˝os、A. Rényi <style id='v19'>等，《随机图的演化》，匈牙利科学院数学出版物，第5卷，第1期，第17–60页，1960年。