GraphZoom：基于谱方法的高效多级图嵌入框架

摘要

图嵌入技术已经越来越多地应用于涉及非欧几里德数据学习的众多不同应用中。然而，现有的图嵌入模型要么在训练过程中无法纳入节点属性信息，要么存在节点属性噪声，影响了模型的准确性。此外，由于它们的高计算复杂度和内存使用，很少有它们可以扩展到大型图形。在本文中，我们提出了 GraphZoom，这是一个多级框架，用于提高无监督图嵌入算法的准确性和可扩展性。GraphZoom 首先进行图融合，生成一个新的图，该图有效地编码了原图的拓扑结构和节点属性信息。然后通过合并具有高光谱相似性的节点，将融合后的图反复粗化成更小的图。GraphZoom 允许将任何现有的嵌入方法应用于粗化的图，然后逐步将在最粗级别获得的嵌入细化为越来越精细的图。我们已经在一些流行的图数据集上评估了我们的方法，用于转换和归纳任务。我们的实验表明，与目前最先进的无监督嵌入方法相比，GraphZoom 可以大幅提高分类精度，并显著加快整个图嵌入过程，速度可达 40.8 倍。

关键词：图嵌入；谱方法；图融合；图节点分类

1 引言

图是一种数据结构，它用于表示对象之间的关系。这种数据结构主要由节点（或称为顶点）和边组成。节点可以代表实体，如人、物品或概念，而边则表示这些实体之间的关系或连接。图可以是有向的或无向的，有向图中的边具有方向性，表示从一个节点到另一个节点的特定关系；无向图则不区分边的方向。图广泛存在于社交网络、通信网络、生物信息学等各种领域，例如，在社交网络 [13] 中，图可以用来表示人与人之间的关系，其中每个人是一个节点，如果他们之间存在社交联系则两节点连边。在交通网络 [5] 中，节点可以用来表示交通枢纽（如汽车站、火车站、机场等），边则代表从这一点到另一点的交通路线。此外，还有生物学中的蛋白质相互作用网络 [14] 和语言学中的词共现网络 [16]。在这些情况中，图提供了一种直观的方式来理解和分析这些系统中的复杂关系。

图形数据在现实世界无处不在，因此，图分析逐渐受到了越来越多的关注，然而，处理和分析图形数据并不是简单的。有一个非常重要的问题就是数据规模的大小，大多数真实的网络都很大，包含数百万个节点和边，而且随着数据量增加，也就是点边数增加，数据规模大小至少是以平方级的增长。这意味着需要高性能的计算和存储系统才能处理这种大规模图数据，同时，有许多图算法和图查询方法是计算密集型的，对系统的处理能力提出了很高的要求。此外，由于图是复杂的和多样的，存在各种类型的图，如异质图、动图等，也增加了处理的难度。随着人工智能技术的愈加成熟，机器学习，尤其是深度学习方法，在很多领域展

现出强大的力量。然而，由于图数据是一种高维的、非欧几里得结构的数据，但大多数现成的机器学习方法以及深度学习方法仅支持向量数据，并不适用于图数据。

随着图的不断发展，在 21 世纪初，研究者们提出了图嵌入方法，很好的解决了上面的问题。图嵌入是一种将图中的节点映射到低维空间中的技术，它旨在保留图中节点之间的相似性或关系，使得存在连接关系的节点彼此靠近，不存在连接的节点之间相对远离，从而捕捉图的原有特征。这种映射使得图数据可以直接利用机器学习算法来进行处理和分析。图嵌入技术的出现为图形数据的处理和分析提供了一种新的思路，广泛应用于图的各种分析处理任务中，如节点分类、链接预测、节点聚类、网络可视化等任务并提升其性能，如在微信社交网络中根据好友的社交关系为用户推荐可能认识的好友或者感兴趣的内容等，具有很强的现实意义。

2 相关工作

早期的图嵌入方法通常使用矩阵分解方法 [1]，当时是作为一种降维的方法，但这种方法难以用于大规模的图。基于图拉普拉斯特征映射（Laplacian Eigenmaps，LE）的图嵌入方法 [2] 的核心思想是如果两个节点在原始图中彼此相邻，那么它们在降维后的目标子空间中应该尽量接近，也就是它们的低维嵌入向量应该彼此接近。算法首先构建一个图的表示，其中节点和边分别代表数据点和它们之间的关系。然后，算法通过计算图中节点之间的相似性来定义局部结构。这种相似性可以通过节点之间的局部距离或其他度量来衡量。接下来，算法利用这些相似性信息来优化低维嵌入空间中的节点位置，使得在保持局部结构的同时，尽可能减少高维空间中的冗余信息。局部线性嵌入（Locally Linear Embedding，LLE） [8] 的核心思想是将数据点视为一个连续的、邻域限制的高维空间中的线性关系，并通过最小化重构误差来找到低维空间中的映射。局部保留投影 (Locality Preserving Projections，LPP） [12] 通过构建空间中各样本对之间的远近亲疏关系，并在投影中保持这种关系。其权值矩阵的设置反映了样本之间的近邻关系，对于近邻样本赋予非零权值，而对于相距较远的样本则赋零。这样，LPP 可以在投影过程中保留样本的局部邻域结构。多维缩放（Multiple Dimensional Scaling，MDS） [15] 主要思想是通过构造低维空间的内积矩阵，通过优化算法调整节点的位置，使得在低维空间中节点之间的距离或相似度与高维空间中的相应值尽可能接近。高阶临近性保留嵌入算法（High-Order Proximity Embedding，HOPE） [10] 是一种基于高阶邻近性的图嵌入方法，其核心思想在于利用节点之间的高阶邻近关系来捕捉图的结构信息。与传统的只考虑一阶或二阶邻近性的图嵌入方法不同，HOPE 能够考虑更高阶的邻近性，从而更全面地保留图的结构信息。使用全局结构信息学习图表示（Learning Graph Representations with Global Structural Information，GraRep） [3] 通过操纵不同的全局转移矩阵，捕获顶点的 k-step relational information（即顶点的 k 阶关系信息），得到 k 阶局部信息，整合这些局部信息，得到全局表达。具体来说，GraRep 将 k 阶关系投影到独立的子空间中，而不是像 skip-gram 模型那样将节点的所有 k 阶关系综合反映到一个向量中。

随着随机游走技术的兴起，研究者开始探索使用随机游走来捕捉图的局部和全局结构，并将其用于图嵌入。DeepWalk [11] 是基于 Word2vec [9] 提出的一种图嵌入方法，是语言模型和无监督学习从单词序列到图上的一种扩展，DeepWalk 将网络节点类比为文本中的单词，将节点的邻居节点类比为单词的上下文，将随机游走采样到的节点序列类比为文本中的句子，再

用 Skip-Gram 模型对生成的定长节点序列映射成低维嵌入向量。Node2vec [4] 是一种综合考虑 DFS 邻域和 BFS 邻域的图嵌入方法，可以看作是 DeepWalk 的一种扩展，可以学习到更加全面和丰富的网络结构信息。

随着深度学习研究的快速推进，越来越多的深度神经网络技术应用于图分析中。GCN [7]，即图卷积网络，借鉴了卷积神经网络（CNN）的思想，通过定义图上的卷积操作来提取图的结构特征和节点信息。在 GCN 中，每个节点都与其邻居节点相连，并通过边进行信息传递。通过堆叠多个 GCN 层，模型可以捕获节点在图上的高阶邻域信息。每一层 GCN 都根据当前节点及其邻居节点的表示来计算新的节点表示，最终生成一个包含丰富信息的节点嵌入矩阵。基于图采样和聚合的图嵌入（Graph SAmple and aggreGatE，GraphSAGE） [6] 通过学习一个聚合函数，聚合邻居节点的特征得到一个节点的信息。再根据各个节点的特征和邻居关系，可以很方便地得到一个新节点的表示，具体步骤包括：类似广度优先遍历的方式对图中每个节点的 k 深度邻近节点进行采样，用聚类函数对每个节点的邻近节点特征信息进行聚类，用非线性激活函数生成新的嵌入向量。

图嵌入技术目前处于不断发展和完善的状态，作为一种高效便捷的图表示方法，其具有广泛的应用前景和巨大的研究潜力。

3 本文方法

3.1 本文方法概述

文章提出了一个无监督的多级学习框架 GraphZoom，能够在现有无监督图嵌入方法的基础上提高其质量和效率。如图 1所示，GraphZoom 包含以下四个阶段：

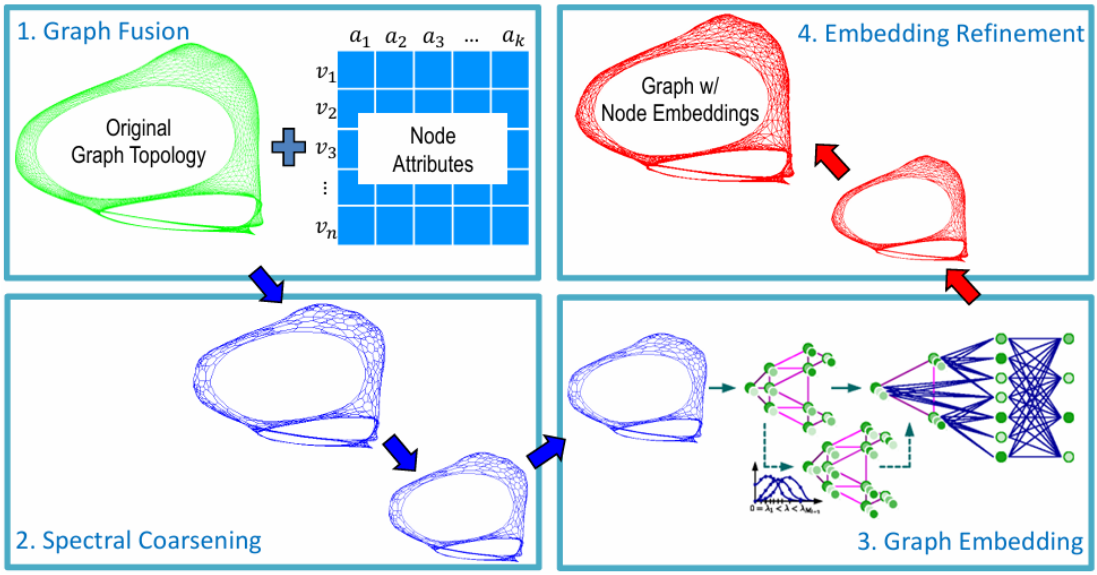

图 1. GraphZoom 框架

阶段一：图融合（Graph fusion）

利用节点属性的特征矩阵，与邻接矩阵结合得到包含结构特征和属性特征的混合邻接矩阵。简单来说，就是构建了一个与原始图具有相同节点数的加权图，但边集是不同的，边的

权重也是不同的。阶段二：频谱粗化（Spectral Coarsening）基于新兴的图信号处理技术，采用高效的局部频谱嵌入方法识别节点簇。简单来说，就是对个随机向量应用简单平滑（低通图滤波）函数，得到维图嵌入的平滑向量，这样可以在线性的时间内实现。考虑一个由图拉普拉斯矩阵（图拉普拉斯矩阵 = 度矩阵-邻接矩阵）的特征向量线性组合表示的随机向量（图信号），采用低通图滤波器快速滤除随机图信号的高频分量或图拉普拉斯矩阵的高特征值对应的特征向量，通过在上应用平滑函数，可以得到一个平滑向量，他基本上是前几个特征向量的线性组合。具体来说，使用高斯迭代将线性方程组求解到一组个初始随机向量，这组向量正交于全 1 向量，满足，其中是原图或融合图的拉普拉斯矩阵。根据中的平滑向量，作者将每个节点都嵌入 T 维空间，这样如果节点 p 和 q 的低维嵌入向量和高度相关，那么可以认为他们在频谱上相似。这里的节点之间的距离采用相邻节点 p 和 q 的频谱节点亲和度来衡量。

在确定了节点聚合方法后，就可以在两个粗化级别和之间获得图映射算子，其中大小为。如果中的节点聚合到中的节点上，则，否则为 0。利用来构造一系列频谱缩减图，较粗的图拉普拉斯矩阵可以通过公式计算。这能高效地保留图的频谱特性，或者说是全局特性。

阶段三：图嵌入（Graph Embedding）

在构造了粗化后的图之后，可以通过任意无监督的图嵌入方法进行嵌入。

阶段四：嵌入优化（Embedding Refinement）

对于给定在第级的图的节点嵌入，使用相应的投影算子将投影到上，即得到第级的精细图，由于投影算子的性质，在粗图中嵌入的节点会直接复制到上一级细图中同一聚合集的节点中，在这种情况下，精细图中频谱相似的节点如果在粗化阶段聚合成单个节点，将具有相同的嵌入结果。为了进一步提高映射嵌入的质量，文章应用了一个由 Tikhonov 正则化驱动的局部细化过程，由于这种方式涉及矩阵求逆，因此以这种方式获得精确的嵌入可能效率不高。为了解决上述问题，采用更有效的频谱图过滤器来平滑嵌入，最后使用低通图滤波器来平滑映射得嵌入矩阵。通过迭代地得到原始图的嵌入。

4 复现细节

4.1 与已有开源代码对比

使用了复现论文的源代码，https://github.com/cornell-zhang/GraphZoom。此外使用两种基准方法的源代码：

（1）deepwalk，源代码地址为：https://github.com/phanein/deepwalk；

（2）node2vec，源码地址为：https://github.com/eliorc/node2vec。

4.2 实验环境搭建

（1）GraphZoom 的实验环境配置过程

创建虚拟环境并激活：

conda create -n graphzoom python=3.6

conda activate graphzoom

安装 graphzoom 的依赖包：

pip install -r requirements.txt

更改参数，运行 graphzoom：

python graphzoom.py –dataset citeseer –searchratio 12 –numneighs 10 –embedmethod deepwalk –coarse simple

（2）DeepWalk 的实验环境搭建过程

创建虚拟环境并激活：

conda create -n deepwalk python=3.8

conda activate deepwalk

安装 deepwalk 的依赖包：

pip install wheel

pip install Cython

pip install six

pip install gensim

pip install scipy

pip install psutil

pip install networkx

安装 DeepWalk 库:

python setup.py install

更改参数，运行 DeepWalk：

deepwalk –input datapath –output dataembeddingspath

（3）node2vec 的实验环境搭建过程

创建虚拟环境并激活：

conda create -n node2vec python=2.7

conda activate node2vec

安装 node2vec 的依赖包：

pip install gensim

pip install numpy

pip install networkx

安装 DeepWalk 库:

python setup.py install

更改参数，运行 node2vec：

python src/main.py –input datapath –output dataembeddingspath

5 实验结果分析

实验对 GraphZoom 框架与几种最先进的无监督图嵌入方法和多级嵌入框架基于 5 个标准图数据集进行了评估。实验中用到的数据集下图 2所示。基准方法为两种模型 DeepWalk、node2vec 和两个多层次的图嵌入框架 HARP 和 MILE。

Dataset	Type	Task	Node	Edges	Classes	Features
Cora	Citation	Transductive	2708	5429	7	1433
Citeseer	Citation	Transductive	3327	4732	6	3703
Pubmed	Citation	Transductive	19717	44338	3	500
PPI	Biology	Inductive	14755	222055	121	50
Reddit	Social	Inductive	232965	57307946	210	5414
Friendster	Social	Transductive	7944949	4466733688	5000	N/A

图 2. 数据集

图 3展示了分类任务的平均分类准确率，以及所有基线方法和 GraphZoom 的执行时间，关于超参数，文章使用 10 次游走，游走深度为 80，窗口大小为 10，DeepWalk 和 Node2vec 的嵌入维度均为 128。

	Cora		Citeseer		Pubmed	
Method	Accuracy(%)	Time(secs)	Accuracy(%)	Time(secs)	Accuracy(%)	Time(mins)
DeepWalk	71.4	97.8	47.0	120.0	69.9	14.1
HARP(DW)	71.3	296.7 (0.3×)	43.2	272.4 (0.4×)	70.6	33.9 (0.4×)
MILE(DW, l=1)	71.9	68.7 (1.4×)	46.5	53.7 (2.2×)	69.6	7.0 (2.0×)
MILE(DW, l=2)	71.3	30.9 (3.2×)	47.3	22.5 (5.3×)	66.7	4.4 (2.3×)
MILE(DW, l=3)	70.6	15.9 (6.1×)	47.1	9.9 (12.1×)	64.5	2.5 (5.8×)
GZoom_F+MILE(DW)	73.8	70.6 (1.4×)	48.9	24.7 (4.9×)	72.1	7.0 (2.0×)
GZoom(DW, l=1)	76.9	39.6 (2.5×)	49.7	19.6 (2.1×)	75.3	4.0 (3.6×)
GZoom(DW, l=2)	**77.3**	15.6 (6.3×)	**50.8**	6.7 (6.0×)	75.9	1.7 (8.3×)
GZoom(DW, l=3)	75.1	**2.4 (40.8×)**	49.5	**1.3 (30.8×)**	**77.2**	**0.6 (23.5×)**
node2vec	71.5	119.7	45.8	126.9	71.3	15.6
HARP(N2V)	72.3	171.0 (0.7×)	44.8	174.3 (0.7×)	70.1	46.1 (0.3×)
MILE(N2V, l=1)	72.1	57.3 (2.1×)	46.1	60.9 (2.1×)	70.8	7.3 (2.1×)
MILE(N2V, l=2)	71.8	30.0 (4.0×)	45.7	28.8 (4.4×)	67.3	4.3 (3.6×)
MILE(N2V, l=3)	68.5	16.5 (7.2×)	45.2	15.6 (8.1×)	61.8	1.8 (8.0×)
GZoom_F+MILE(N2V)	74.3	59.2 (2.0×)	48.3	62.3 (2.0×)	72.9	7.3 (2.1×)
GZoom(N2V, l=1)	**77.3**	43.5 (2.8×)	**54.7**	38.1 (3.3×)	77.0	3.0 (5.2×)
GZoom(N2V, l=2)	77.0	13.5 (8.9×)	51.7	15.3 (8.3×)	**77.8**	1.5 (10.4×)
GZoom(N2V, l=3)	75.3	**3.0 (39.9×)**	50.7	**4.5 (28.2×)**	77.4	**0.4 (39.0×)**

图 3. 原文实验结果

从结果中可以看到，对于直推式学习任务，GraphZoom 可以使 DeepWalk 和 node2vec 在 Cora、Pubmed 和 Citeseer 上的分类准确率分别提高 8.3%、10.4%和 19.4%，同时达到减少 40.8 倍运行时间的目的。当增加粗化级别时，GraphZoom 仍可以以较短的 CPU 时间产生可比的甚至更好的嵌入精度。实验结果表明 GraphZoom 与底层的嵌入方法无关，能够提高各种数据集上最新的监督式嵌入方法的准确性和速度。这些结果表明，多级频谱方法可以提高填充的速度和质量，所以 GraphZoom 的运行速度要快得多。除了减小图形尺寸之外，作者的粗化方法还可以从原始图形中过滤出多余的信息，同时保留嵌入的关键频谱特性，因此分类准确率方面的嵌入质量也得到提高。

复现实验采用的数据集为 Cora 和 Citeseer，基线方法为 DeepWalk 和 node2vec，实验对比了所有基线方法和 GraphZoom 在节点分类任务上的平均准确率以及平均执行时间。关于 DeepWalk 和 node2vec 的超参数，我们使用 10 次游走，游走深度为 80，窗口大小为 10，DeepWalk 和 Node2vec 的嵌入维度均为 128。

methed	Cora		Citeseer	
	Accuracy(%)	Time(s)	Accuracy(%)	Time(s)
Deepwalk	71.5	42.656	46.3	44.098
GZoom(DW,l=1)	75.1	41.998	47.7	39.046
GZoom(DW,l=2)	76	34.583	48.5	37.873
GZoom(DW,l=3)	74.2	28.282	50.9	32.124
node2vec	72.6	10.230	47.1	9.6234
GZoom(N2V,l=1)	75.8	15.363	49.6	11.646
GZoom(N2V,l=2)	75.3	9.867	50.3	9.635
GZoom(N2V,l=3)	74.6	6.722	49.1	7.891

图 4. 复现实验结果

复现结果如图 4所示，当粗化级别为 3 时，GraphZoom 的表现最好，GraphZoom 可以使 DeepWalk 在 Cora 和 Citeseer 上的分类准确率分别提高 6.2%、9.9%，同时运行时间能缩短 33.6%、27.15%，使 node2vec 在 Cora 和 Citeseer 上的分类准确率分别提高 4.4%、6.3%，同时运行时间能缩短 32.7%、18.00%。当增加粗化级别时，GraphZoom 可以缩短运行时间，而且也能产生优于基线方法的甚至更好的嵌入精度。这些结果表明，多级频谱方法可以提高填充的速度和质量，所以 GraphZoom 的运行速度要快得多。除了减小图形尺寸之外，作者的粗化方法还可以从原始图形中过滤出多余的信息，同时保留嵌入的关键频谱特性，因此分类准确率方面的嵌入质量也得到提高。

6 总结与展望

本文介绍了一种图嵌入框架 GraphZoom，用于提高无监督图嵌入任务的准确性和可扩展性。GraphZoom 首先融合原图的节点属性和拓扑结构，构造新的加权图。然后，它使用光谱粗化来生成粗化图的层次结构，其中嵌入在最粗的级别上对最小的图进行嵌入。然后，使用合适的图滤波器对图嵌入进行迭代细化，得到最终结果。实验表明，GraphZoom 在许多流行的数据集上提高了分类精度和嵌入速度。复现结果在性能方面与原论文给出的基本一致，问题在于时间开销上与原文不符，这可能是硬件的差距所带来的现象。以后的工作可以从以下方面进行展开：

（1）图融合方面：GraphZoom 模型通过融合节点属性矩阵和拓扑图来生成一个包含更丰富信息的融合图。为了进一步提高准确性，可以考虑引入更复杂的特征融合技术，如深度学习中的特征提取和融合方法。

（2）频谱粗化方面：GraphZoom 模型通过合并具有高频谱相似性的节点来生成一系列连续的粗化图。为了优化这一过程，可以尝试结合其他图处理技术，如图分割和聚类，以进一步简化图结构并保留关键信息。

（3）嵌入优化方面：GraphZoom 模型通过使用适当的图过滤器将嵌入图重新映射回到原始图上，以保证嵌入的平滑性。为了进一步优化这一过程，可以考虑引入更先进的图过滤器技术，如基于图卷积网络的过滤器。

参考文献

[1] Amr Ahmed, Nino Shervashidze, Shravan M. Narayanamurthy, Vanja Josifovski, and Alex Smola. Distributed large-scale natural graph factorization. *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural Information Processing Systems*, 2001.

[3] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[5] S. Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI Conference on Artificial Intelligence*, 2019.

[6] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Neural Information Processing Systems*, 2017.

[7] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2016.

[8] Li Liu, William Kwok-Wai Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *International Joint Conference on Artificial Intelligence*, 2016.

[9] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[10] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[11] Bryan Perozzi, Rami Al-Rfou, and Steven S. Skiena. Deepwalk: online learning of social representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[12] Baoxu Shi and Tim Weninger. Proje: Embedding projection for knowledge graph completion. In *AAAI Conference on Artificial Intelligence*, 2016.

[13] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI Conference on Artificial Intelligence*, 2017.

[14] Xu-Wen Wang, Lorenzo Madeddu, Kerstin Spirohn, Leonardo Martini, Adriano Fazzone, Luca Becchetti, Thomas P. Wytock, István A. Kovács, Olivér M. Balogh, Bettina Benczik, Mátyás Pétervári, Bence Ágg, Péter Ferdinandy, Loan Vulliard, Jörg Menche, Stefania Colonnese, Manuela Petti, Gaetano Scarano, Francesca Cuomo, Tong Hao, Florent Laval, Luc Willems, Jean-Claude Twizere, Marc Vidal, Michael A. Calderwood, Enrico Petrillo, Albert Ĺaszló Barabási, Edwin K. Silverman, Joseph Loscalzo, Paola Velardi, and Yang-Yu Liu. Assessment of community efforts to advance network-based prediction of protein–protein interactions. *Nature Communications*, 14, 2023.

[15] Fei Wu, Jun Song, Yi Yang, Xi Li, Zhongfei Zhang, and Yueting Zhuang. Structured embedding via pairwise relations and long-range interactions in knowledge base. In *AAAI Conference on Artificial Intelligence*, 2015.

[16] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *AAAI Conference on Artificial Intelligence*, 2017.