



上海交通大学学位论文

排列型组合优化问题的机器学习求解 方法研究

姓 名：汪润中

学 号：019033910050

导 师：杨小康、严骏驰

学 院：电子信息与电气工程学院

学科/专业名称：计算机科学与技术

申请学位层次：博士

2023 年 3 月

**A Dissertation Submitted to
Shanghai Jiao Tong University for Doctoral Degree**

**MACHINE LEARNING SOLVERS FOR
PERMUTATION-BASED COMBINATORIAL
OPTIMIZATION PROBLEMS**

**Author: Runzhong Wang
Supervisor: Prof. Xiaokang Yang and Prof. Junchi Yan**

School of Electronic Information Electrical Engineering
Shanghai Jiao Tong University
Shanghai, P.R.China
March 30th, 2023

摘要

组合优化问题是计算机科学、运筹学、应用数学等学科共同研究的基础性问题。本文研究的排列型组合优化问题是匹配、排序、路径规划等常见组合优化问题的集合。近年来,随着机器学习(尤其是深度学习)技术的发展,考虑到组合优化问题的重要科学和应用价值,利用机器学习方法求解组合优化问题已经成为了一个前沿的研究方向。面向排列型组合优化问题,本文研究并完善了两种主流的方法论:1)纯机器学习求解方法,2)机器学习与经典算法融合的求解方法。本文设计并实现了完善的开源框架以供科学研究和工程应用。本文的研究成果集中在如下几个方面:

首先,本文对纯机器学习求解方法的神经网络设计进行了理论分析,设计了三种方法并分别用于求解图匹配问题、二次指派问题和具有基数约束的组合优化问题。这类纯机器学习求解方法的特点是神经网络的输出符合对应问题的特定排列。纯机器学习求解方法首先将数学形式的问题输入转化成等价的图结构,随后利用图神经网络等模型抽取其特征,最后采用可微分的约束层(例如 Sinkhorn 层)对神经网络求解器的输出施加约束。作者对图神经网络、可微分约束层的设计开展了理论和实验研究,针对排列型组合优化问题设计了专用网络,实验结果证明了神经网络求解器广泛、灵活的应用场景,特别在数学优化求解的效率和效果上超越了传统优化算法。作者发现,基于数据驱动的神经网络求解方法减轻了对复杂数学知识的依赖,使求解方法的设计、调优更为灵活;由于神经网络本身非常适合采用 GPU 加速,与纯 CPU 的传统求解算法相比,神经网络求解器可以充分利用 GPU 的强大算力、效率更高;与不可微分的传统求解方法相比,天然可微的神经网络模块可以进一步支持端到端的联合预测、优化,实现在噪声环境和不确定条件下的优化求解。

其次,本文研究了融合传统算法与神经网络的机器学习框架。此类融合方法同时具有神经网络的高效性和传统算法的强先验知识,可充分发掘其协同效应。作者认为,与纯机器学习求解方法相比,融合传统算法的机器学习框架具有更精细、更可靠、更易学的特点。其中,更精细体现在神经网络设计需要考虑更多的算法细节,需要对具体问题、算法有深入的理解和认知;更可靠来源于引入了传统算法作为性能兜底,即使神经网络性能下降依然可以正常工作;更易学体现在神经网络的专才专用,只需学习简化后的子任务。作者共设计了三种融合学习框架,分别利用了不同问题的不同特性。其中,第一种融合范式采用机器学习指导传统算法,并用于求解图编辑距离问题。融合方法显著降低了图编辑距离求解的计算开销,同时精度损失几乎可以忽略不计。第二种融合范式采用传统算法指导机器学习,构造了图匹配神经网络的无监督训练框架。无监督模型在小规模数据集上与监督学习性能相当,在大规模数据集上可以为监督训练提供初始化。第三种融合范式中,通过对等地融合传统算法与机器学习,实现了通用的双层优化机

器学习融合框架，融合框架的效果超越了单独的强化学习或启发式求解器。

最后，本文设计并实现了面向排列型组合优化问题（特别是图匹配和二次指派问题）的开源框架与系统。相关技术的开源旨在促进该领域的研究，并简化相关算法的工程部署和下游应用。特别地，本文开发了通用图匹配算法库 `pygmtools`，通过一套统一的编程接口，支持线性匹配、二图匹配、多图匹配、神经网络图匹配等算法的跨平台、跨设备部署。本文还设计了一套面向图匹配与二次指派问题的深度学习系统 `ThinkMatch`，该系统涵盖了国内外相关领域最新的研究进展，提供了 8 个最先进的神经网络模型在 5 个基准数据集上的训练、测试功能，具有配置灵活、功能完善的特点。本文为国内自主研发的深度学习框架“计图”开发了图匹配求解功能，支持传统算法求解和深度学习求解，并开源了典型应用场景的具体实现。

综上所述，本文对排列型组合优化的机器学习求解方法进行了深入的研究。针对不同问题，本文设计了三种纯机器学习求解器，通过理论和实验证明了其优越性；本文提出了三种机器学习与传统算法的融合方式，分别侧重于把握不同问题的不同层面；开源项目总结、凝练了上述技术。大量理论分析和实验结果表明，本文提出的一系列方法深入发掘了组合优化机器学习求解的本质，显著提升了机器学习在排列型组合优化问题上的求解能力。

关键词：组合优化，排列型问题，深度学习，图神经网络，优化求解器，图匹配，二次指派问题，图编辑距离，基数约束，哈密尔顿回路，图神经网络

ABSTRACT

Combinatorial optimization (CO) is the fundamental problem at the intersection of computer science, operations research, and applied math. The permutation-based combinatorial optimization studied in this thesis is the superset of real-world problems including matching, ranking, routing, etc. Seeing the recent ground-breaking achievements of machine learning (especially deep learning), and being aware of the values of CO in its scientific and engineering applications, applying machine learning to tackle CO has become a trending research topic. Within the scope of permutation-based CO, this thesis focuses on and extends two mainstream methodologies: 1) pure machine learning solvers, and 2) machine learning fused with traditional solvers. Besides, the author designs and implements open-source repositories for scientific research and other downstream applications. The contributions of this thesis are summarized as follows.

Firstly, this thesis studies the theoretical insights for designing neural networks, and tailors three pure machine learning solvers for three permutation-based CO problems including graph matching, quadratic assignment, and cardinality-constrained combinatorial optimization problems. The output of the neural network meets certain permutation constraints. These pure machine learning solvers accept problem inputs in math form, then transform them into equivalent graph structures. Features are further extracted by graph neural networks, followed by a differentiable layer (e.g. a Sinkhorn layer) to enforce the constraints. Experiment results show the wide applicability and flexibility of neural network solvers, surpassing traditional solvers in terms of efficiency and/or efficacy in pure optimization tasks. This thesis concludes the following advantages of pure machine learning solvers over traditional solvers: 1) The data-driven manner of machine learning mitigates the requirement of in-depth math knowledge, allowing easier and more flexible solver design; 2) Compared to traditional solvers which are mostly CPU-based, neural networks are more friendly to GPUs, leading to higher efficiency by exploiting the power of GPUs; 3) Compared to traditional solvers which are usually non-differentiable, the natural differentiability of neural networks allows joint end-to-end learning of predictor networks and solver networks, enabling important applications such as optimization with noises and decision-making under uncertainty.

Secondly, this thesis studies the paradigms to fuse machine learning with traditional solvers, taking the best of the two worlds offered by neural networks' high efficiency and traditional solvers' strong prior knowledge. There are three unique properties of fused solvers compared to pure machine learning solvers: 1) The fused solvers are more specialized, meaning that more domain knowledge is required when designing a fusion paradigm of neural networks and traditional solvers; 2) The fused solvers are

more reliable because the traditional algorithm can ensure the worst-case performance thus being robust to issues such as network degeneration; 3) The neural networks are easier to learn in those fused solvers because the neural networks can focus on learning simplified subproblems. Technically, this thesis designs three fusion paradigms, by exploiting different natures of different problems. The first paradigm guides the traditional A* solver by machine learning to solve the graph edit distance problem, and the experiment shows a significant reduction in the computational cost, with a negligible loss in accuracy. The second paradigm develops an unsupervised deep graph matching learning pipeline by guiding the neural network with traditional graph matching solvers. The experiment shows that the unsupervised model is comparative to supervised state-of-the-arts on smaller-scaled datasets, and the unsupervised model could serve as an initialization and improve supervised finetuning on larger-scaled datasets. Thirdly, a bi-level fusion framework is developed for general permutation-based CO problems. Experiment results show the effectiveness and the fused solvers are superior to stand-alone reinforcement learning or heuristic solvers.

Finally, this thesis presents open-source frameworks and systems for permutation-based CO (especially graph matching and quadratic assignment). The open-source projects are meant to further facilitate future research, and simplify the deployment and application of those novel solvers. Specifically, a general graph matching library named *pygmtools* is developed, equipping a universal API with the support of linear matching, two-graph matching, multi-graph matching, and neural graph matching. *pygmtools* also supports the deployment of the aforementioned solvers across different platforms and devices. Additionally, a deep learning system tailored for scientific research on graph matching and quadratic assignment named *ThinkMatch* is developed. It covers the state-of-the-art research progress and offers 8 novel neural networks with training/testing protocols on 5 benchmark datasets. *ThinkMatch* has a flexible configuration system and full-featured utilities. Besides, graph matching functionalities are developed for the deep learning framework named Jittor whereby both traditional solvers and neural network solvers are supported, and examples are released featuring typical applications.

In summary, this thesis studies machine learning methods for solving permutation-based combinatorial optimization problems. Tailored for different problems, three pure learning solvers are developed, and their superiorities are proved both theoretically and empirically. This thesis also develops three paradigms fusing learning and traditional solvers, with focuses on different aspects varying among problems. The open-source projects summarize and distill the aforementioned methodologies. A large amount of theoretical and empirical study shows that the methods proposed in this thesis reflect the inherent nature of learning for CO, significantly improving the solution quality of

learnable CO methods.

Key words: combinatorial optimization, permutation-based problem, deep learning, graph neural networks, optimization solvers, graph matching, quadratic assignment problem, graph edit distance, cardinality constraint, Hamiltonian cycle problem, graph neural networks

主要符号与英文缩略词对照表

主要符号对照：除非另行说明，以字母 x 为例，普通小写字母 x 代表标量，黑体小写字母 \mathbf{x} 代表矢量，黑体大写字母 \mathbf{X} 代表矩阵。 \mathbf{X}^\top 代表矩阵 \mathbf{X} 的转置， $\text{diag}(\mathbf{x})$ 代表将向量 \mathbf{x} 放置于对角的矩阵， $\text{vec}(\mathbf{X})$ 代表将矩阵 \mathbf{X} 进行按列向量化的操作。 $\text{tr}(\mathbf{X})$ 代表矩阵的迹。本文中， \mathbb{R} 代表实数域， $\mathbf{1}_{n \times m}, \mathbf{0}_{n \times m} \in \mathbb{R}^{n \times m}$ 分别代表元素是全 1、全 0 的 $n \times m$ 矩阵。 $\mathbf{1}_n$ 是 $\mathbf{1}_{n \times 1}$ 的简写形式，是一个列向量。花体字母 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ 代表由点集合 \mathcal{V} 、边集合 \mathcal{E} 构成的图 \mathcal{G} 。 $\mathbf{X} \otimes \mathbf{Y}$ 代表矩阵的克罗内克积 (Kronecker product)， $\mathbf{X} \odot \mathbf{Y}$ 代表矩阵的哈达玛积 (Hadamard product)。 $\|\mathbf{X}\|_y$ 代表矩阵的 y -范数，其中 $\|\mathbf{X}\|_F = \text{tr}(\mathbf{X}^\top \mathbf{X})$ 代表 Frobenious 范数。同样的定义对向量也成立。 $[\mathbf{x} \ \mathbf{y}]$ 表示拼接 (concat) 两个向量， $[\mathbf{X} \ \mathbf{Y}]$ 表示按特征维度拼接两个矩阵或张量。 \mathbf{x}_i 、 $\mathbf{x}[i]$ 、 $\mathbf{X}_{i,j}$ 和 $\mathbf{X}[i,j]$ 均代表对向量 (或矩阵) 取元素，元素下标从 1 开始。

主要数学符号:

X	排列矩阵
x 或 $\text{vec}(\mathbf{X})$	按列向量化的排列矩阵
n_1	排列矩阵的行数目
n_2	排列矩阵的列数目
ω	排列型组合优化的问题参数
$J(\mathbf{X}, \omega)$	排列型组合优化的目标函数
S	双随机矩阵
A	图的邻接矩阵
M	线性指派问题的 (一阶) 相似度矩阵
K	图匹配问题和二次指派问题的 (二阶) 相似度矩阵
H	超图匹配问题的 (高阶) 相似度张量
$\mathcal{G}^1 = (\mathcal{V}^1, \mathcal{E}^1)$	图 1 和它的节点集合、边集合
$\mathcal{G}^A = (\mathcal{V}^A, \mathcal{E}^A)$	伴随图和它的顶点集合、边集合
\mathbb{R}	实数集
$\mathbb{R}_{\geq 0}$	非负实数集
$\mathcal{U}(a, b)$	实数 a, b 之间的均匀分布
$\mathcal{N}(a, b)$	均值为 a 、方差为 b 的正态分布

主要英文缩略词和中文对照：

AI	Artificial Intelligence	人工智能
ML	Machine Learning	机器学习
NP	Non-deterministic Polynomial	非确定性多项式复杂度
CO	Combinatorial Optimization	组合优化
LAP	Linear Assignment Problem	线性指派问题
GM	Graph Matching	图匹配问题
MGM	Multi-Graph Matching	多图匹配问题
MGM ³	Multi-Graph Matching with a Mixture of Modes	混合模态多图匹配问题
QAP	Quadratic Assignment Problem	二次指派问题
FLP	Facility Location Problem	设施选址问题
MCP	Mac Coverage Problem	最大覆盖问题
GED	Graph Edit Distance	图编辑距离问题
TSP	Traveling Salesman Problem	旅行商问题
HCP	Hamiltonian Cycle Problem	汉密尔顿回路问题
OT	Optimal Transport	最优传输问题
CNN	Convolution Neural Network	卷积神经网络
GNN	Graph Neural Network	图神经网络
GCN	Graph Convolutional Network	图卷积神经网络
LSTM	Long-Short Term Memory	长短时记忆力网络
CV	Constraint Violation	约束违反量
A*	A-star Algorithm	A 星搜索算法
GA	Graduated Assignment	渐近指派算法
SIFT	Scale-Invariant Feature Transform	尺度不变特征变换
RL	Reinforcement Learning	强化学习
MDP	Markov Decision Process	马尔可夫决策过程
DQN	Deep Q-Network	深度 Q 值网络
PPO	Proximal Policy Optimization	近端策略优化
POMO	Policy Optimization with Multiple Optima	多最优解的策略优化
API	Application Programming Interface	程序编程接口

目 录

摘 要.....	I
ABSTRACT.....	III
主要符号与英文缩略词对照表.....	VI
第一章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 排列型组合优化问题描述.....	3
1.3 主要研究挑战.....	6
1.4 本文主要贡献.....	7
1.5 本文组织结构.....	10
第二章 排列型组合优化求解文献综述.....	13
2.1 排列型组合优化的方法论.....	13
2.1.1 传统求解算法.....	13
2.1.2 纯机器学习算法.....	15
2.1.3 融合传统算法的学习方法.....	16
2.2 排列型组合优化的典型问题和现有求解方法.....	17
2.2.1 匹配问题.....	17
2.2.2 路径问题.....	18
2.2.3 基数约束优化问题.....	19
2.3 排列型组合优化的开源框架.....	20
2.4 本章小结.....	20
第三章 排列型组合优化问题的纯机器学习求解.....	21
3.1 本章引言.....	21
3.2 面向图匹配求解的神经网络设计.....	21
3.2.1 图匹配问题描述.....	21
3.2.2 研究动机.....	23
3.2.3 算法描述.....	24
3.2.4 实验分析.....	31
3.3 面向通用二次指派求解的神经网络设计.....	39
3.3.1 二次指派问题描述.....	39
3.3.2 研究动机.....	40
3.3.3 算法描述.....	41
3.3.4 实验分析.....	51
3.4 面向基数约束求解的神经网络设计与理论分析.....	61

3.4.1	基数约束优化问题描述	62
3.4.2	研究动机	62
3.4.3	算法描述和理论分析	64
3.4.4	具体实现和实验分析	68
3.5	本章小结	76
第四章	融合传统算法的排列型组合优化学习求解	79
4.1	本章引言	79
4.2	与树搜索融合的图编辑距离机器学习框架	79
4.2.1	图编辑距离问题介绍	79
4.2.2	研究动机	81
4.2.3	算法描述	82
4.2.4	实验分析	87
4.3	与图匹配求解器融合的无监督训练框架	91
4.3.1	混合模态、部分匹配和外点介绍	92
4.3.2	研究动机	92
4.3.3	算法描述	94
4.3.4	实验分析	101
4.4	与启发式融合的双层优化通用机器学习框架	107
4.4.1	研究动机	107
4.4.2	组合优化问题的单/双层形式	108
4.4.3	算法描述	110
4.4.4	具体实现和实验分析	111
4.5	本章小结	119
第五章	排列型组合优化问题的学习求解开源	121
5.1	本章引言	121
5.2	pygmtools: 面向下游应用的图匹配通用算法库	121
5.2.1	开发动机与竞品比较	121
5.2.2	设计细节与工具用法	122
5.3	ThinkMatch: 面向科研的深度图匹配系统	124
5.3.1	开发动机与系统特色	125
5.3.2	系统使用方法	126
5.4	面向计图的图匹配求解框架设计与实现	127
5.4.1	技术特点	127
5.4.2	应用案例	129
5.5	本章小结	130

第六章 总结与展望	131
6.1 工作总结	131
6.2 未来展望	133
参考文献	135
附录 A 3.4 节定理的详细证明	155
A.1 命题 3.4 的证明	155
A.2 推论 3.5 的证明	161
A.3 推论 3.6 的证明和注记	162
致 谢	165
作者简介及攻读学位期间科研成果目录	167

第一章 绪论

排列 (permutation) 作为一个重要的离散数学概念, 许多实际问题可以被刻画为 n 个候选对象的排列。基于给定的目标函数求解最优排列具有丰富的理论和应用价值。特别地, 在以排列型问题为代表的组合优化问题中, 近年来机器学习、尤其是深度学习方法在国内外研究中不断涌现, 机器学习已经被证明能够提升传统非学习求解方法的性能, 相关技术已应用于计算机视觉、数据挖掘、运筹优化等领域。本文聚焦于排列型组合优化问题的机器学习求解方法, 研究内容围绕纯机器学习求解方法、融合传统算法的学习求解方法、开源求解框架设计展开。

1.1 研究背景和意义

组合优化问题 (combinatorial optimization problems, CO) 是计算机科学、运筹学、应用数学等学科共同研究的基础性问题。根据目标解空间是否完全连续, 优化问题可分为连续优化和组合优化。解空间的离散性为组合优化的求解带来了技术上的挑战——多数组合优化问题具有 NP 难的特性, 难以在短时间内精确求解; 但是, 由于离散是真实世界的基本属性, 研究组合优化求解具有极强的科学价值和现实意义——通过在离散的解空间优化一个目标函数, 组合优化实现了对真实世界目标函数 (例如成本、收益、资源消耗) 的最优化。一方面, 组合优化的 NP 难特性使得传统算法遇到瓶颈, 在单一数据集的过度调参很可能导致其他场景的性能下降; 另一方面, 广泛存在的真实场景数据以及由数据驱动的机器学习技术带来了另一种可能性: 利用机器学习的自适应能力, 弥补传统算法的不足。采用神经网络求解排列型优化问题可以追溯到 20 世纪 80 年代采用 Hopfield 网络求解旅行商问题的工作^[1]。可惜的是, 受限于当时的计算资源, 相关研究随着 AI 的第二次低谷一起沉寂。近年来随着算力和神经网络结构的发展, 国内外学者陆续注意到了采用机器学习求解组合优化的研究价值。图灵奖得主、“深度学习三巨头”之一的 Yoshua Bengio 教授曾为此撰文第一作者的综述文章^[2], 中国科学院大学的郭田德、韩丛英教授在同时期提出了该概念^[3]并撰写专著《组合优化问题的机器学习求解方法》^[4], 机器学习、计算理论、数学等相关领域的顶级会议和组织均认识到了该问题的重要性 (详见表1-1)。

表 1-1 近期国际上组织的组合优化、机器学习的相关研讨会。

组织者	所属领域	研讨会主题	时间
国际机器学习大会 (ICML)	机器学习	感知、推断与图神经网络	2020
计算理论国际年会 (STOC)	计算理论	带预测的经典算法	2020
纯数学与应用数学协会 (IPAM)	数学	深度学习与组合优化	2021
神经信息处理系统大会 (NeurIPS)	机器学习	深度学习与整数规划	2021
国际先进人工智能协会 (AAAI)	人工智能	人工智能与决策优化	2022

特别地,本文关注的是排列型组合优化问题(permutation-based combinatorial optimization problems),对该问题的研究最早可以追溯到19世纪中叶数学家雅可比用德语写成的著作^[5]。但是,受限于时代背景和语言差异^[6],对排列型组合优化的系统性研究直到20世纪40年代才正式开始。伯克霍夫^[7]、冯·诺依曼^[8]、丹齐格^[9]等现代计算机理论和应用数学的先驱均对线性目标函数下的最优排列开展过研究。作为排列型组合优化中最基础的形式,当目标函数为线性时,该问题又被称为线性指派问题(linear assignment problem, LAP)。线性指派问题最终在1955年被解决,Harold Kuhn^[10]总结了另外两位匈牙利数学家的工作^[11-12],将线性指派算法命名为匈牙利算法。James Munkres^[13]发现了匈牙利算法的多项式时间复杂度,目前我们已知该算法的时间复杂度是 $\mathcal{O}(n^3)$ 。对于排列型组合优化问题中更一般、更复杂的形式,通常不存在已知的多项式时间复杂度的精确算法:例如,图匹配和二次指派问题被证明是NP难^[14]、哈密顿回路问题被证明是NP完全^[15]。计算复杂性的理论瓶颈为传统算法的研究带来了挑战,而近年来涌现的机器学习方法为这个历久弥新的研究领域注入了新的活力。

在排列型组合优化问题的求解上,与传统算法相比,机器学习方法具有独特的技术特点和性能优势。本文将其归纳为“可训练、可并行、可微分、可融合”的四大优势。可训练反映了机器学习的数据驱动特点,无需过多的人工干预即可让求解模块适配数据集分布。作为对比,传统求解算法的设计、调参通常需要大量的专家知识。可并行是机器学习、尤其是深度学习方法的天然优势,它们能够充分利用现代GPU的强大算力。作为对比,传统求解算法通常是专为串行设计的,且通常只能运行在CPU上。例如,在求解二次指派问题时,机器学习方法成功地学习到了如何将组合优化问题映射到一个较优解(本文3.3节)。同时,与运行在CPU上的传统算法相比,运行在GPU上的机器学习方法展现了更出色的计算效率。可微分的优势体现在优化求解与上游的预测、感知步骤高度耦合的场景中。当预测、感知模块采用机器学习时,同为机器学习的可微分求解方法自然比一般不可微的传统算法更具有优势。例如,对于计算机视觉中的图匹配问题,基于卷积神经网络和机器学习求解器的方法(本文3.2节)相比卷积神经网络和传统求解器的组合^[16]具有更高的匹配精度;对于金融资产的价格预测和(带约束的)投资决策,本文3.4节中基于长短时记忆网络和机器学习求解器的预测、决策一体化方法展现出了更高的决策质量。此外,传统求解算法和深度学习模型实际上可以互相提供指导,构造可融合的求解新范式,弥补单独传统算法或单独神经网络的性能缺陷。

由于机器学习在求解排列型组合优化问题上的诸多优越性,相关技术已经被广泛应用于计算机视觉、模式识别、数据挖掘、运筹优化等场景。典型的排列型组合优化问题如图1-1所示。视觉任务是深度学习和排列型组合优化的重要场景,Sarlin等^[17]利用图神经网络和注意力机制构造了一个匹配求解网络,将该网络用于图像的稠密关键点匹配以及三维重建任务;He等^[18]观察到了多目标跟踪

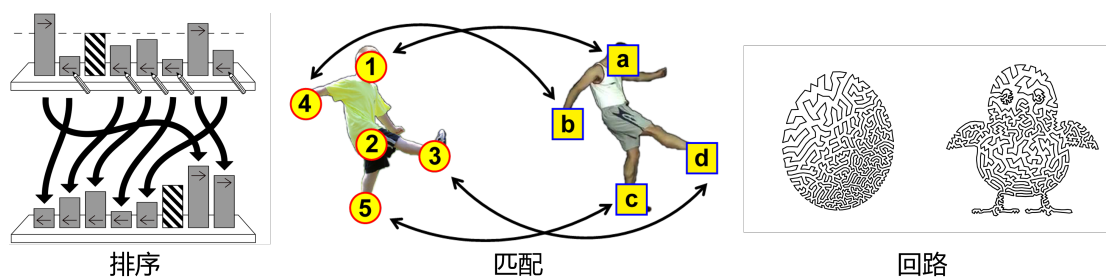


图 1-1 排列型组合优化问题的代表性例子。

Figure 1-1 Representative examples of permutation-based combinatorial optimization problems.

任务与匹配求解的等价性，通过在神经网络中嵌入一个可微分的匹配求解模块实现端到端学习；Song 等^[19]将排列型组合优化用于视频超分辨率任务，通过机器学习方法处理特征对齐问题；Li 等^[20]将跨域目标检测任务视为排列型优化问题下的图匹配和超图匹配问题，利用神经网络方法求解。对于视觉之外其他模态的模式识别、数据挖掘问题，排列型组合优化的求解也是重要技术之一：Xu 等^[21]在自然语言处理中通过机器学习求解排列型组合优化，将语义相近的语句对齐；Peng 等^[22]利用机器学习处理了模式识别中经典的图编辑距离问题，在一般的图结构上得到了应用；Chen 等^[23]将机器学习辅助的 A* 搜索和求解技术用于化学分子的逆合成分析。此外，更精确、更高效的排列型组合优化求解技术还可以反哺机器学习领域。联邦学习是一种强调用户隐私的深度学习技术，Liu 等^[24]将联邦学习中的模型融合问题看作排列型组合优化的求解，基于匹配的模型融合方法可以显著提升联邦学习模型的收敛速度和测试精度。最后，对于各种真实场景下的运筹优化问题，基于机器学习的排列型组合优化求解方法已经有了广泛应用。Vinyals 等^[25]在经典的旅行商问题中，提出了一个输入城市序列、输出目标排列的指针网络对优化问题进行求解。在计算机网络调度领域，基于现有的机器学习排列型组合优化求解方法，Zhang 等^[26]提出了一种在边缘计算场景下的细粒度服务迁移算法。

综上所述，研究排列型组合优化的机器学习算法一方面具有广泛的应用前景，可以推动关联学科的发展；另一方面与数十年的理论积累紧密关联，而机器学习的引入为这个领域提供了新的视角，有望突破传统算法目前的性能瓶颈。正因如此，排列型组合优化的机器学习求解方法才得到了国内外研究者的广泛关注，并在近年逐渐成为机器学习领域的一个研究热点。

1.2 排列型组合优化问题描述

诸多实际问题的求解可以看作寻找前 n 个自然数的最优排列。例如，我们希望最优化 n 台机器完成 n 个任务的总成本，每台机器只能处理一项任务，且每台机器完成每项任务都对应一个成本。如图 1-2 所示，对于 $n = 3$ ，排列 $(2, 3, 1)$ 即

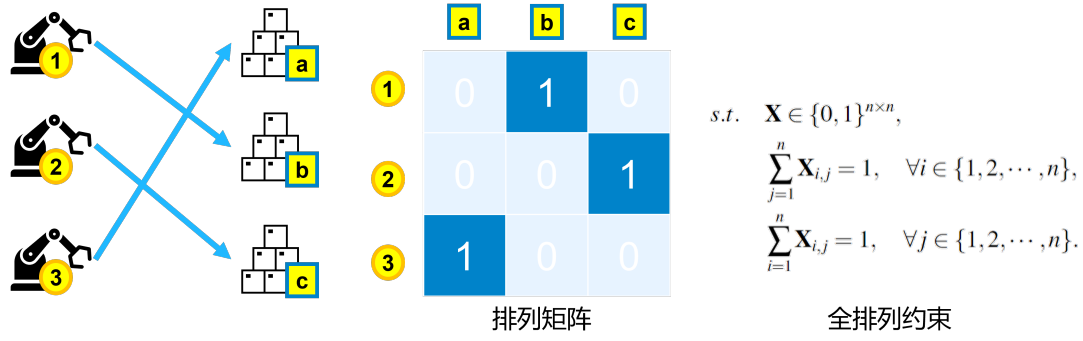


图 1-2 一个具体的排列型组合优化任务。

Figure 1-2 An example of a permutation-based combinatorial optimization task.

是一个可行解，其中机器 1 被指派到任务 b、机器 2 被指派到任务 c、机器 3 被指派到任务 a。本文对排列型组合优化问题定义如下，

$$\min_{\mathbf{X}} J(\mathbf{X}|\omega), \quad (1.1a)$$

$$s.t. \quad \mathbf{X} \in \{0, 1\}^{n_1 \times n_2}, \quad (1.1b)$$

$$\sum_{j=1}^{n_2} \mathbf{X}_{i,j} = a_i, \quad \forall i \in \{1, 2, \dots, n_1\}, \quad (1.1c)$$

$$\sum_{i=1}^{n_1} \mathbf{X}_{i,j} = b_j, \quad \forall j \in \{1, 2, \dots, n_2\}. \quad (1.1d)$$

其中，0/1 矩阵 \mathbf{X} 为排列型组合优化问题的决策变量，例如 $\mathbf{X}_{i,j} = 1$ 代表指派第 i 台机器完成第 j 个任务； ω 代表问题参数，例如对于所有 i, j ， $\omega_{i,j}$ 代表让机器 i 完成任务 j 的成本； $J(\mathbf{X}|\omega)$ 为最小化的目标函数，例如完成所有任务的总成本 $J(\mathbf{X}|\omega) = \sum_{i,j} \omega_{i,j} \mathbf{X}_{i,j}$ （若为最大化问题，将目标函数取反即可）。公式 (1.1c) 和公式 (1.1d) 分别代表排列型问题的行约束和列约束， a_i, b_j 可根据问题需求进行配置，等号约束可根据需要替换为“ \leq ”。特别地，求解得到的决策变量须满足

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{X}_{i,j} = n_1 \quad \text{或} \quad \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{X}_{i,j} = n_2, \quad (1.2)$$

以确保符合排列型问题的性质（而非其他的线性约束）。举例说明，公式 (1.1) 覆盖了以下典型的排列型问题：

1) **全排列**。作为最经典的排列型组合优化问题，全排列为将前 n 个自然数全部进行排列，例如，将 n 台机器指派至 n 个任务。 $(1, 2, \dots, n)$ 是一个全排列。全排列型组合优化问题的数学形式为

$$\min_{\mathbf{X}} J(\mathbf{X}|\omega), \quad (1.3a)$$

$$s.t. \quad \mathbf{X} \in \{0, 1\}^{n \times n}, \quad (1.3b)$$

$$\sum_{j=1}^n \mathbf{X}_{i,j} = 1, \quad \forall i \in \{1, 2, \dots, n\}, \quad (1.3c)$$

$$\sum_{i=1}^n \mathbf{X}_{i,j} = 1, \quad \forall j \in \{1, 2, \dots, n\}. \quad (1.3d)$$

上述问题形式确保了矩阵 \mathbf{X} 的行、列元素之间一一对应的关系。

2) 部分排列。 在一些实际场景中，我们无需排列完整的候选集合。对于前 n_1 个自然数，只需将其中 n_2 个进行排列 ($n_2 < n_1$)。例如，我们有 n_1 台机器，但只需完成 n_2 个任务。对于 $n_1 = 5, n_2 = 3$, $(1, 2, 4)$ 是一个部分排列。部分排列的组合优化问题的数学形式为

$$\min_{\mathbf{X}} J(\mathbf{X}|\omega), \quad (1.4a)$$

$$s.t. \quad \mathbf{X} \in \{0, 1\}^{n_1 \times n_2}, \quad (1.4b)$$

$$\sum_{j=1}^{n_2} \mathbf{X}_{i,j} \leq 1, \quad \forall i \in \{1, 2, \dots, n_1\}, \quad (1.4c)$$

$$\sum_{i=1}^{n_1} \mathbf{X}_{i,j} = 1, \quad \forall j \in \{1, 2, \dots, n_2\}. \quad (1.4d)$$

上述问题形式确保了矩阵 \mathbf{X} 的列元素到行元素的单一对应，但允许部分行元素的空缺。

3) 前 k 个排列。 从 n 个候选自然数中选取 k 个，也属于排列型组合优化问题的范畴。例如，从 n 位员工中选择 k 位共同完成一项任务。前 k 个排列的组合优化问题的数学形式为

$$\min_{\mathbf{X}} J(\mathbf{X}|\omega), \quad (1.5a)$$

$$s.t. \quad \mathbf{X} \in \{0, 1\}^{n \times 2}, \quad (1.5b)$$

$$\sum_{j=1}^2 \mathbf{X}_{i,j} = 1, \quad \forall i \in \{1, 2, \dots, n\}, \quad (1.5c)$$

$$\sum_{i=1}^{n_1} \mathbf{X}_{i,1} = k, \quad \sum_{i=1}^{n_1} \mathbf{X}_{i,2} = n - k. \quad (1.5d)$$

上述问题形式确保了有且只有 k 个元素出现在 \mathbf{X} 的第一列中。

基于上述定义，排列型组合优化问题的求解可以描述为：已知目标函数 J 、问题参数 ω 和约束条件 \mathcal{C} ，输出最优的决策变量 \mathbf{X} ，并符合约束条件 $\mathbf{X} \in \mathcal{C}$ 。本文研究的排列型组合优化问题的机器学习方法构造了一个机器学习模型 $f: (J, \omega, \mathcal{C}) \rightarrow \mathbf{X}$ ，尽可能最优化目标函数 $J(\mathbf{X}|\omega)$ 。问题参数 ω 可以是确定的，直接由问题输入定义（此类问题称为纯优化问题，pure optimization problem）； ω 也可以是不确定的，来自其他机器学习模型的预测（此类问题称为联合预测-优化

问题, joint predict-and-optimize problem, 亦有学者称之为不确定条件下的决策问题, decision-making under uncertainty)。

1.3 主要研究挑战

机器学习方法在处理排列型组合优化问题时, 同时存在理论层面和技术层面的挑战, 能否妥善处理这些挑战深刻地影响着最终效果的优劣。具体地, 作者认为, 将机器学习用于求解排列型组合优化时, 需要解决以下挑战。

1) 针对纯机器学习求解方法, 如何设计神经网络求解器? 作为一类组合优化问题, 排列天然具有离散的性质。但是, 神经网络的输入、输出和中间层均具有连续性。特别地, 在训练过程中需要梯度进行反向传播, 因此所有神经网络参数都必须是连续的。如何设计神经网络求解器, 在实现网络输出符合约束的同时保留梯度? 一种可行的策略是直接离散域处理排列, 利用强化学习方法估计策略梯度, 在允许离散输出的同时依然可以得到梯度进行训练。不过, 根据机器学习界的普遍共识, 强化学习方法在样本效率、训练稳定性等方面存在缺陷。另一种方法借鉴了经典优化领域的思路, 将排列从离散域放松到连续域, 进而使用神经网络输出松弛问题在连续域的解。不过, 松弛问题与原问题的近似程度很可能不受控制, 造成机器学习方法在原问题上的求解性能不佳。上述挑战还反映了神经网络的黑箱特性, 需要从理论、实践两方面解决。

2) 针对融合传统算法的学习求解方法, 如何设计可融合的范式? 机器学习方法在分类、回归等模式识别问题中表现更佳, 传统算法擅长针对问题精确求解。研究者们希望设计可融合的求解范式, 同时具有机器学习的高性能和传统算法的高精度。但是, 处理具体问题时, 通常需要面对专用的传统优化算法。由于传统算法种类繁多、细节各异, 在具体设计融合框架时依然缺乏框架性的指导: 机器学习或传统算法谁应该占据主导地位、如何定位可用神经网络替换的传统算法模块、如何设计高效的训练方法, 都是亟待解决的科学问题。

3) 针对所有机器学习方法, 如何为 NP 难的排列型组合优化设计训练方法? 除了少数属于 P 问题的排列型组合优化 (例如线性指派问题), 多数排列型组合优化问题属于 NP 难问题。基于通常的假设 $P \neq NP$, NP 难复杂性意味着无法在多项式时间内精确求解。这一特点一方面造成了传统算法的瓶颈, 给机器学习方法留下了提升空间; 另一方面也为机器学习的训练带来了障碍——由于无法通过传统算法大量地获得最优解标签, 简单且稳定的监督学习方法不再适用。因为得不到大量监督标签, 需要特别设计无监督或弱监督的学习框架。此外, 排列型组合优化本身的难度也对模型结构设计和模型容量提出了挑战——如何在有限的计算资源下合理地设计模型结构、高效地建模问题特征, 进而实现基于机器学习的问题求解。

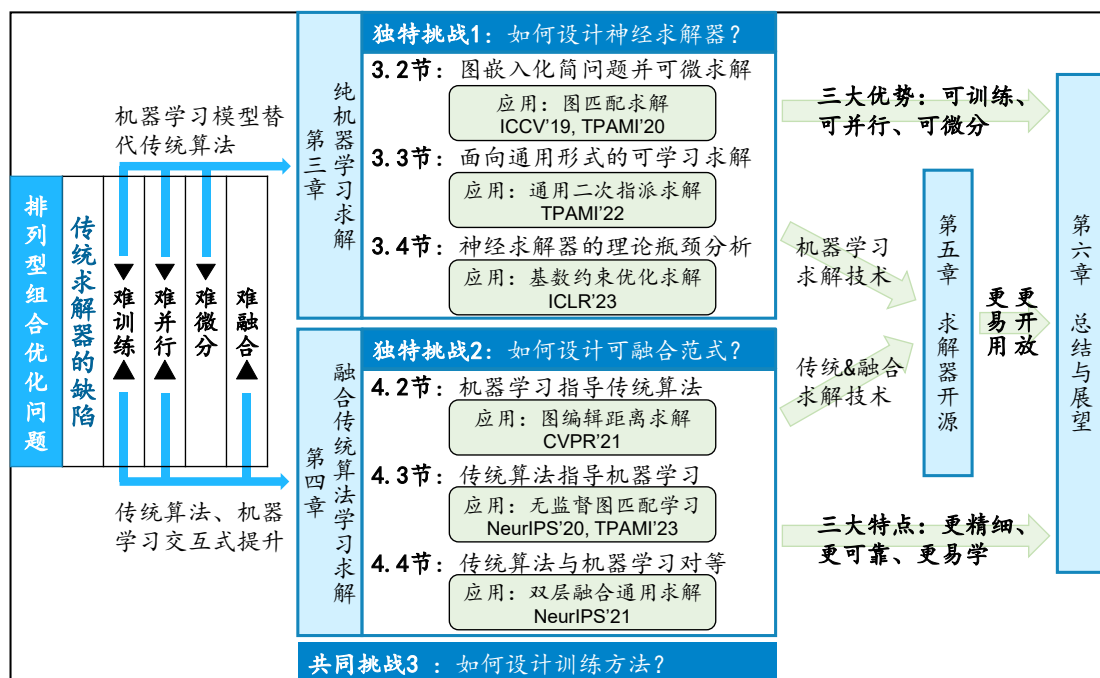


图 1-3 本文的研究挑战和主要贡献总结。

Figure 1-3 Summary of challenges and main contributions of this thesis.

1.4 本文主要贡献

如图1-3所示，从传统求解器不可训练、不可并行、不可微分的缺陷出发，本文的研究主要基于纯机器学习求解、融合传统算法学习求解这两种现有的方法论^[2]展开。通过机器学习模型替代传统算法，解决了传统求解器难训练、难并行、难微分的缺陷，这也是纯机器学习方法突出的优势所在。通过传统算法和机器学习交互式提升，解决了传统求解器难融合的缺陷，同时神经网络模块带来了可训练、可并行的特性。与纯机器学习求解方法相比，融合方法具有更精细、更可靠、更易学的三大特点。通过将上述研究中积累的求解技术进一步整合、二次开发，形成了本文发布的开源求解框架。

作者认为，机器学习可训练、可微分、可并行的优势在纯机器学习求解算法中表现得尤为突出。面向主要技术难点——如何为排列型组合优化问题的纯机器学习求解方法设计神经求解器，同时考虑下游任务的实际挑战，本文在纯机器学习求解方向有以下贡献：

- 提出了首个面向图匹配求解的神经网络，利用图嵌入化简问题并可微求解。图匹配问题是模式识别、数据挖掘等领域的基础性问题，其本质是一个 NP 难的排列型组合优化问题。本文首先设计了一种基于图嵌入的机器学习图匹配算法，其中图嵌入算法由单个图的图内卷积和同时考虑两个图的跨图卷积组成。通过图嵌入，图的结构信息被嵌入到了节点信息中，原本 NP 难的图匹配问题被化简为可在多项式时间内求解的线性指派。本文采用可

微分的 Sinkhorn 方法求解线性指派，并结合排列损失函数构成了一个端到端、可微分的神经网络图匹配求解器。这是学术界首个面向图匹配求解的神经网络方法。实验证明了图嵌入图匹配算法的高精度。

- **提出了首个面向二次指派问题的神经网络，探索面向通用数学形式的可学习求解。**二次指派问题是图匹配在数学形式上的扩展，除了覆盖一般的图匹配，还覆盖了布局规划、电路设计等运筹优化场景。作者发现，求解二次指派问题等价于在一个“伴随图”上求解顶点分类。自然地，作者采用图神经网络学习顶点分类，结合可微的 Sinkhorn 算法对神经网络输出添加约束，构造了首个直接求解二次指派问题的神经网络。实验结果证明了神经网络求解器广泛、灵活的应用场景：与目前效果最好的预测网络结合，神经网络求解器在图匹配场景达到了最高的匹配精度；在运筹优化问题中，神经网络求解器在求解效率、求解效果两个方面超越了传统优化算法。
- **提出了首个面向基数约束优化问题的神经网络，并分析了神经求解器的理论瓶颈。**作为一种常见的排列约束，基数约束的优化问题通常具有 NP 难的复杂性。在神经网络中引入基数约束并非易事，本文基于最优传输理论、Sinkhorn 算法和 Gumbel 方法设计了一种可微分、可调控的基数约束网络层。理论分析表明，本文提出的方法可以任意程度地逼近离散的基数约束；作为对比，现有方法在最坏情况下无法逼近。在三个应用场景的实验表明，本文的神经网络求解器性能与商用求解器相当，并且具有处理联合预测-优化问题的能力。

作者认为，融合传统算法的排列型组合优化学习方法与纯机器学习方法相比具有三大特点：更精细、更可靠、更易学。更精细体现在算法框架需要对具体问题针对性设计，对设计者要求更高；更可靠体现在传统算法能够提供性能兜底，避免神经网络退化；更易学体现在神经网络的专才专用，只需学习某个具体模块。特别地，面向融合方法的主要技术难点——如何设计融合范式，本文有以下贡献：

- **机器学习指导传统算法，在经典的 A* 树搜索中融合深度学习以实现提速。**图编辑距离是一种常用的图结构度量，计算图编辑距离需要求解排列型组合优化。从图编辑距离的经典求解算法——A* 树搜索算法出发，本文利用神经网络预测模块替代传统的人工启发式预测。本文提出的神经网络预测更加精确，同时增加了对搜索树的剪枝频率。实验结果表明，将神经网络模块融合进 A* 树搜索算法可以显著降低图编辑距离求解的计算开销，同时求解的精度损失几乎可以忽略不计。
- **传统算法指导机器学习，在深度图匹配中融合传统图匹配求解器以实现无监督预训练。**以传统的监督学习方式训练图匹配神经网络需要大量的、昂

贵的节点级标注，导致图匹配方法的应用范围受限。本文遵循计算机视觉领域无监督学习的研究脉络，在深度图匹配网络中融合传统图匹配求解器，提出了一种“差异最小化”的无监督学习框架：在训练过程中，通过将神经网络分支与传统求解器分支的输出差异最小化，实现图匹配神经网络的无监督预训练。实验结果证明了传统算法与神经网络融合可以组成强大的无监督学习框架。

- **传统算法与机器学习对等融合，将启发式求解器嵌入双层优化的强化学习框架以提升求解精度。**常见的组合优化问题通常存在一系列高效的启发式求解器，能够在短时间内获得较高质量的解。但是，即使实际情况允许求解器利用更多的求解时间，启发式求解器通常很难获得更好的解。本文提出一种融合机器学习的双层优化框架，上层优化采用强化学习模型优化问题结构，下层优化采用传统的启发式求解器进行高效求解。在图编辑距离、汉密尔顿回路、计算任务调度等排列型组合优化问题上，该融合框架的效果超越了单独的强化学习模型或启发式求解器。

在前述研究中，作者积累了大量的求解技术，包括传统求解算法和最新的机器学习求解方法。注意到目前的排列型组合优化问题缺乏高质量、应用范围广的开源求解框架，作者将积累的技术整合、扩展后，形成了本文在排列型组合优化求解开源方面的贡献：

- **开发了面向下游应用的图匹配通用算法库 `pygmtools`，方便开发者快速、高效的跨平台部署。**针对图匹配问题的算法研究已经持续了二十余年，但是，国际上仍缺乏一个通用的求解框架。本文作者开发的 `pygmtools` 通过一套统一的编程接口，支持线性匹配、二图匹配、多图匹配、神经图匹配等成熟的图匹配算法在不同开源框架和不同计算设备上的开发和部署。
- **开发了面向科研的深度学习排列型组合优化系统 `ThinkMatch`，方便研究人员快速试错和模型迭代。**为了促进国内外科研社区对排列型组合优化问题，尤其是图匹配和二次指派问题的研究，本文作者开发的 `ThinkMatch` 系统涵盖了国际上最新的研究进展，提供了 8 个最先进的神经网络模型、4 种常见实验配置和 6 个基准数据集的训练测试接口。`ThinkMatch` 还提供了研究所需的训练可视化、日志文件、数据集预处理、模型文件存取等功能，支持灵活的配置文件设置和参数调整，已经成为了国际上开展排列型组合优化相关研究的基准框架之一。
- **开发了面向国内自主研发的深度学习框架“计图”的图匹配框架，实现与计图的相互赋能。**计图是清华大学图形学实验室开发的深度学习与科学计算框架，保留了用户友好的编程接口同时支持更高效的动态编译。本文作

者基于计图算子开发的图匹配框架一方面提升了算法的求解效率，另一方面也拓展了计图框架的应用场景。

1.5 本文组织结构

本文内容按照以下结构展开：

第一章首先阐述了本文的研究内容、研究重点以及应用背景，包括排列型组合优化问题的定义、相关方法的应用场景以及在排列型组合优化中引入机器学习的意义。同时给出了排列型组合优化问题的数学形式，罗列了研究挑战，给出了本文的主要创新点和贡献点。本文主要从机器学习的视角出发，重点关注求解排列型组合问题的过程中遇到的机器学习挑战。本章为阐述具体研究内容的后续章节做了铺垫，使得读者对本文在排列型组合优化和机器学习领域的贡献有了整体认识。

第二章对排列型组合优化的机器学习求解方法开展了文献综述，从传统算法到最新的机器学习算法和开源框架，总结了该领域研究现状，回顾了研究进展。本章旨在使读者了解现有研究中的优势与不足，与后续章节的技术讨论形成呼应。

第三章面向纯机器学习的排列型组合优化求解方法，首先从图匹配问题出发，提出了一种基于图嵌入算法的图匹配求解网络，其中图嵌入将原先 NP 难的问题简化为容易求解的线性指派问题。扩展到更一般的二次指派问题，继续提出将二次指派问题视作等价的图结构，通过图神经网络求解顶点分类实现对二次指派问题的学习求解。最后讨论了基数约束优化问题，从理论层面证明了在神经网络中处理基数约束的可行性，并提出了一种可微分、可调控的基数约束网络层，能够任意程度地逼近离散的基数约束。本章从理论和实验两方面证明了纯机器学习的求解方法在排列型组合优化问题上可微分、可并行、可训练的优势。

第四章面向融合传统算法的排列型组合优化学习求解方法，首先验证了机器学习指导传统算法的可行性，深度学习可以在经典的 A* 树搜索算法中替代传统启发式，通过更准确的路径预测和更激进的剪枝策略提升搜索效率。在另一个角度，传统算法也可以指导神经网络，通过“差异最小化”的框架指导深度图匹配模型实现无监督预训练。最后，本章提出了机器学习与传统算法的双层优化融合范式，在实验效果上分别超越了单独的机器学习模型以及单独的传统算法。本章从实验上证明了将传统算法与机器学习融合的优越性，并且面向具体问题场景提出了 3 种新的融合方式，丰富了方法论，验证了融合方法更精细、更可靠、更易学的特点。

第五章讨论排列型组合问题的开源框架，分别介绍了面向图匹配应用的算法库 `pygmtools`、面向排列型组合优化领域科研人员的 `ThinkMatch` 以及面向国内自主研发的深度学习框架“计图”的图匹配求解器。通过开源工作，作者一方面

希望回馈曾经给予作者无私帮助的开源社区，另一方面也希望促进相关领域的科研和应用。

第六章总结了本文贡献，并讨论了后续研究方向。

第二章 排列型组合优化求解文献综述

排列型组合优化问题具有深刻的科学价值和现实意义，在过去的数十年内受到了数学、运筹学、计算机科学等学科的广泛关注。特别是近几年，与机器学习结合的排列型组合优化求解方法频繁发表于国际机器学习大会 International Conference on Machine Learning (ICML)、神经信息处理大会 Neural Information Processing System (NeurIPS)、国际表征学习大会 International Conference on Learning Representations (ICLR)、模式识别与计算机视觉大会 Computer Vision and Pattern Recognition (CVPR) 等机器学习、人工智能领域的顶级会议，以及 IEEE 会刊模式分析和机器智能 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 等顶级期刊中。本章首先总结了近 10 年排列型组合优化的方法论，特别是机器学习方法在该类问题的进展，随后根据具体问题讨论方法的发展脉络，最后讨论排列型组合优化的开源求解框架。

2.1 排列型组合优化的方法论

本节分别对机器学习时代之前的传统求解算法、基于纯机器学习的方法、融合传统算法的学习方法进行讨论。

2.1.1 传统求解算法

对排列型组合优化问题持续数十年的研究已经形成了许多成熟的求解方法，包括整数规划求解、连续松弛求解、启发式方法、元启发式方法等。传统算法的一大优势是无需训练，一旦开发完成即可直接用于任意同类型问题的求解。这些传统算法由研究人员手工设计、调参，凝聚了研究人员对问题结构和数据分布的深刻理解，通常具有较优秀的求解性能。

整数规划是求解组合优化问题（特别是线性目标函数的组合优化问题）的通用方法。整数规划方法广泛存在于上世纪 80-90 年代的排列型组合优化研究中^[27-29]。通过对所有变量和约束条件构造一棵分支定界（branch-and-bound）搜索树，整数规划方法在整个可行域中寻找组合优化问题的最优解。其中，分支策略决定了搜索方向，定界方法通过在每个节点计算目标函数的下界实现搜索树的剪枝（假设求解的是最小化问题）。虽然分支定界的基本原理是简单的树搜索，但是一个高效的求解器内部有非常复杂的实现细节，其求解效率严重依赖于预求解、分支策略等专家知识。所幸，目前市面上已有多个整数规划求解器可供直接调用，包括开源的 SCIP^[30]和商用求解器 Gurobi^[31]。在实际求解时，首先将问题进行数学建模，随后直接调用现有的求解器即可。整数规划的优点是可以保证输出问题的最优解，但是由于问题本身的 NP 难复杂性，整数规划方法通常耗时较长。

由于直接在离散域进行整数规划效率较低，将原问题松弛到连续域成为了一类颇具竞争力的求解方法。通过将约束条件放松至连续域，更高效的连续优化方法可以被用于求解。在求解完成后，再将连续解投影回离散域，作为原问题的一个可行解。特别地，以排列型组合优化中的“全排列”问题（定义请见1.2节）为例，一种常见的松弛方法是双随机松弛，即将公式(1.3)中的0/1约束舍弃，松弛到如下形式

$$\min_{\mathbf{X}} J(\mathbf{X}|\omega), \quad (2.1a)$$

$$s.t. \quad \sum_{j=1}^n \mathbf{X}_{i,j} = 1, \quad \forall i \in \{1, 2, \dots, n\}, \quad (2.1b)$$

$$\sum_{i=1}^n \mathbf{X}_{i,j} = 1, \quad \forall j \in \{1, 2, \dots, n\}. \quad (2.1c)$$

上述松弛形式被广泛应用于求解旅行商问题^[32]和图匹配问题^[33-34]。其他的松弛形式，例如谱松弛^[35]，也在图匹配等排列型组合优化问题中得到了一定的应用。

启发式算法（heuristics）是针对特定问题特别设计的算法，通常具有快速、高效的特点。例如，针对旅行商问题，研究者发现了 k -最优（ k -Opt）方法可以快速改进一个现有的回路^[36]。基于此发现，Lin、Kernighan、Helsgaun等人开发了以他们的首字母命名的LKH算法^[37]。LKH算法的开源版本持续更新了20余年，在求解旅行商等路径问题时兼具速度和精度。对于图编辑距离问题，研究者发现可以将边信息通过匹配算法映射到点信息中，进而进行快速匹配和距离计算^[38-39]。对于计算任务的排列，研究者发现了它与俄罗斯方块的联系，并提出了对应的“Tetris”启发式算法^[40]。

元启发式算法（meta-heuristics）具有更强的通用性，通常是通过计算机模拟某种自然现象，实现对可行解的搜索以及对目标函数的最优化。常见的元启发式算法有模拟退火^[41]、蚁群算法^[42]、禁忌搜索^[43]、遗传算法^[44]等。元启发式算法通常从某个初始解出发，通过模仿某种自然现象进行搜索，在求解时限前返回搜索过程中的最优解。元启发式算法具有通用性与高效性，受到了工业界的广泛欢迎，在多个排列型组合优化问题上均有应用，例如图匹配问题^[45-46]、排班排产问题^[47]、旅行商问题^[48]等。近期国内对排列型组合优化问题的研究多基于元启发式，例如广东财经大学、中山大学的周雅兰等^[49]研究了求解排列型组合优化的粒子群优化算法，清华大学的宁秀^[50]研究了交互熵算法用于求解排列型组合优化的性能，武汉科技大学的李章洪等^[51]提出了针对不同元启发式的通用解空间坍塌策略。

2.1.2 纯机器学习算法

将机器学习引入排列型组合优化的求解时，一种思路是用机器学习模型完全替代传统算法，利用机器学习模型的表达能力和神经网络在 GPU 上的加速实现速度和精度的双重提升。需要指出的是，神经网络在优化领域的应用远早于最近一波深度学习浪潮^[52]。早在 1985 年，Hopfield 和 Tank 就提出采用 Hopfield 神经网络，将约束纳入能量函数，将网络的训练直接转化成旅行商问题的求解^[1]。但是，这一类先驱工作与如今的机器学习方法存在显著不同——先驱工作将优化过程看作神经网络的训练，神经网络训练完毕即意味着优化完成；如今的机器学习方法分为训练和测试步骤，期望模型学得问题求解的共性策略，在测试步骤直接加载预训练权重即可。

在神经网络设计层面，序列神经网络、图神经网络和注意力机制是纯机器学习求解方法重要结构。以长短时记忆网络 (long-short term memory, LSTM) 为代表的序列神经网络起源于 Vinyals 等的指针网络论文^[25]，论文中将排列问题的所有候选元素看作一个序列输入 LSTM 编码器，利用 LSTM 网络中的长时记忆能力构造问题的特征向量。随后，使用另一个 LSTM 解码器，配合注意力机制^[53]输出一个排列，作为问题的解。但是，这种将无序集合看作有序输入的处理方法缺乏稳定性，任意打乱输入顺序即可造成输出的变化。后续 Khalil 等^[54]采用的图神经网络方法解决了序列模型的稳定性缺陷，图神经网络已经成为了纯机器学习方法的默认模型结构。最近的研究工作还引入了在自然语言、视觉领域获得广泛成功的 Transformer 结构，Kwon 等^[55]提出了一种 Transformer 模型，通过将矩阵转化成等价的二分图来处理矩阵形式的输入。

在学习方法层面，早期工作在简单的小规模问题上采用了监督学习^[25]，不过由于最优解时常不可得，近年来人们对于使用强化学习求解组合优化的关注度呈上升趋势^[54,56-58]。强化学习的基本思想是通过智能体不断地与环境交互，让智能体学到最优的行为方式。其中，智能体对当前环境的观察称为状态。智能体依据特定的策略，基于当前状态选择一个动作。在智能体执行一个动作后，环境将转移到另一个状态。同时，环境会以奖励的形式对智能体的动作作出反馈。在求解排列型组合优化问题时，问题本身通常被视作环境，决策变量被视为动作，目标函数被视为奖励函数。一般而言，组合优化领域的研究者会选择直接调用成熟的强化学习算法，例如 DQN^[59]和 PPO^[60]。此外也有部分专为组合优化开发的强化学习算法，例如 POMO^[61]考虑了组合优化问题最优解不唯一的特点。一般来说，强化学习是逐步求解组合优化问题的。但是，逐步求解可能造成误差累积，如果智能体在早期犯了错误，后期决策步骤无法进行修正。为了应对上述挑战，Chen 等^[62]提出让智能体学习如何更好地修改现有解，而不是直接从零开始输出解。此外，Barrett 等^[63]改进了强化学习流程，这使得让智能体修改之前的动作成为可能。

考虑到强化学习的低训练效率，一些最新的工作正在探索更高效的无监督

学习方法。Karalias 等^[64]提出将网络的输出直接看作解，将目标函数看作训练的损失函数，直接进行端到端的自监督训练。为了让输出符合约束，将约束保持作为惩罚项添加在自监督损失函数中。这种自监督方法的训练效率远高于强化学习，对于 100 节点的问题最多只需几小时即可训练完成（强化学习需要几天）。但是，上述方法的约束保持只能作用在训练中，在测试时可能出现严重违反约束条件的情况。本文的3.4章节将对此进行深入的理论和实验分析。

2.1.3 融合传统算法的学习方法

尽管纯机器学习求解算法在不断发展，到目前为止，传统的排列型组合优化算法依然在许多场景中占主导地位。因此，融合传统算法的学习方法应运而生：将机器学习作为一个模块，与现有的传统优化算法融合，有望获得较好的效果。由于排列型组合优化问题是对真实世界的高级数学抽象，传统算法中较为僵化的专家规则和启发式策略限制了算法性能的进一步提升。将机器学习方法与传统算法结合意味着将复杂的现实世界与高度抽象的优化问题联系起来，实现 $1+1>2$ 的效果。

虽然强化学习的训练过程往往非常耗时^[65]，但在缺乏先验知识和专家经验的场合，其相比传统启发式算法常常更加有效。因此，将强化学习与传统的启发式算法相结合已经成为一个重要研究方向。例如，Cai 等^[66]提出强化学习的输出结果可以作为启发式方法的初始解。Khalil 等^[67]提出将强化学习模型作为整数规划算法中的分支策略。此外，模仿学习^[68]也是一种融合强化学习和传统启发式算法的技术路径。在训练开始阶段，智能体模仿传统的启发式算法，这可以明显减少模型训练的收敛时间。例如，Gasse 等^[69]在整数规划中，让神经网络模仿经典的强分支（strong branching）策略。强分支策略具有决策效果好但是计算效率低的特点，用更高效的机器学习模型将其替代可以提升整数规划算法的运行效率。

构造神经网络与传统优化算法融合的框架可以扩展深度神经网络的应用范围。将深度神经网络应用在排列型组合优化（包括其他优化）场景时，通常会遇到传统优化算法不可导的挑战。研究者已经对优化算法的可导性开展了深入的研究。对于凸优化问题及其求解算法，Amos 等^[70-71]发现了凸优化问题在最优解处的可导性，并可通过 KKT (Karush-Kuhn-Tucker) 条件进行反向传播。对于更一般的非凸优化问题（排列型组合优化正属于此类），目前的研究认为其求解过程虽然理论上可导，但是由于在决策边界的非连续性，其真实的梯度无法指导神经网络训练^[72]。虽然该问题具有很大挑战，但最近的研究也在许多特定问题的（近似）可导性上有了实质性突破，例如次模最优化^[73]、线性目标函数问题^[72,74]。特别地，文献^[72]提出了一种将优化转换为可微分且不局部恒定操作的方法。它依赖于随机扰动优化器，有望与现有求解器一起使用。文献^[75]提出了 CombOptNet，其将线性整数规划问题视作黑盒，并作为一个可微组合层。它利

表 2-1 机器学习求解排列型组合优化问题的代表性现有工作总结。

	匹配问题	路径问题	基数约束问题
纯机器学习求解	[77-79]	[25,54,80]	[81]
融合传统算法的学习求解	[16,82-83]	[84]	[72,74]

用欧几里德距离进行约束更新，并利用文献^[76]提出的方案进行代价更新。本文的第四章将对机器学习与传统算法的融合范式做进一步扩展。

2.2 排列型组合优化的典型问题和现有求解方法

本小节以匹配问题、路径问题、基数约束优化问题为例，分别讨论传统算法与机器学习算法在求解具体排列型组合问题的研究进展。特别地，机器学习方法的代表性现有工作总结在表格2-1中。

2.2.1 匹配问题

匹配问题的基础、在多项式时间内可解的线性指派问题在 1955 年被匈牙利算法所解决^[10]。线性指派问题只考虑了点与点之间的一阶相似度，忽视了潜在的连边关系。本文关注的是考虑边信息以及更高阶信息的匹配问题，包括图匹配、图编辑距离、超图匹配等，以及更一般化的数学形式例如二次指派问题。上述问题均具有 NP 难复杂性。

在匹配问题的传统方法中，考虑到问题的 NP 难复杂性，多种近似求解成对图结构匹配（即二图匹配）的算法被相继提出，其中包括渐近指派算法^[33,85]、谱匹配方法^[86]、随机游走^[34]、投影算法^[87-88]等。在国内，安徽大学的江波^[89]、武汉大学王福东^[90]对图匹配问题的传统算法开展了系统性研究。图编辑距离的问题设定类似于二图匹配问题，一类启发式算法将问题简化到一组线性指派进行快速求解^[38-39]。为了处理噪声、外点存在时图相似度建模中的误差，多个图结构也可以被联合匹配^[91-93]。多个图结构的联合匹配不仅是一个更现实的问题设定，同时也可在一定程度上减轻二图匹配时图相似度建模的局限性。在多数多图匹配工作中，循环一致性被视作一个有效的正则项，用于增强对噪声和外点的鲁棒性^[94-95]。另一种提升匹配鲁棒性的方向是提升连边的复杂度，引入更高阶的超图进行超图匹配^[96-97]。超图匹配能够同时建模旋转和缩放不变性（图匹配只能选择其一），但超图匹配需要更大的计算开销。对于匹配问题中传统算法的技术细节，可以参考综述^[98]。

通过机器学习求解匹配问题正在受到越来越多研究人员的关注^[99]，包括早期的传统机器学习方法^[100-101]以及最新的深度学习方法^[16]。在匹配问题中引入机器学习的主要动机是基于学习的特征能够为匹配提供更多的信息，尤其是考虑到匹配任务与一般的识别任务存在差异。将常规任务（例如 ImageNet 分类任

任务^[102]) 中训练得到的特征直接用于匹配通常效果不佳。研究者们认为, 在匹配任务上的端到端训练可以进一步提升精度。例如, 卷积神经网络 (CNN) 和图神经网络 (GNN) 是常用的深度学习模型^[77-78,103]。在最新的研究中, 除了考虑匹配任务, 研究者还考虑了对潜在图结构的学习, 以降低视觉任务中对预先给定图结构的依赖^[79,83]。此外, 对于模式识别任务中的匹配问题, 通常需要标注节点层面的匹配关系进行监督训练, 这带来了过于昂贵的标注代价。本文的4.3小节将致力于降低匹配学习中对标注数据的需求。

2.2.2 路径问题

排列型组合优化中的路径问题可以分为判别形式 (如汉密尔顿回路问题) 和优化形式 (如旅行商问题)。判别形式中, 需要回答一个判断问题: 输入的图中是否存在汉密尔顿回路? 优化形式中, 需要求解得到最优解, 例如找到总长度最短的回路。判别和优化形式在一定程度上可以统一: 若将存在的边长设为 0、不存在的边长设为 1, 构造一个全连接的图, 判别形式的汉密尔顿回路问题即可规约到优化形式——若找到了长度为 0 的回路, 该回路即是汉密尔顿回路。鉴于旅行商问题更具一般性, 且拥有广泛的应用场景, 旅行商问题的求解在过去得到了充分研究。

求解路径问题的经典算法包括整数规划、启发式算法、元启发式算法等。利用整数规划求解旅行商问题并不直观, 需要采用特殊的数学建模形式才可获得较高的求解效率。一种较为高效的整数规划建模称为 MTZ 方法, 其命名来自于三位学者 Miller、Tucker、Zemlin^[104]。MTZ 方法的高效性来自于消除了旅行商问题中的子回路, 但是受限于问题本身的 NP 难复杂性, 即使是合理建模的整数规划算法依然效率较低。旅行商问题还可放松至双随机约束进行求解^[32]。另一方面, 由于旅行商问题的广泛应用场景, 存在许多性能强大的专用求解器。基于线性规划的 Concorde 求解器¹是研究中常用的求解方法。此外, 启发式算法通常更加高效, 基于 k -Opt 局部改进策略的 LKH 算法^[37]能够快速、精确地完成旅行商问题的求解。最后, 以遗传算法为例的元启发式算法同样具有求解旅行商问题的能力^[48]。

在现有的神经网络求解路径问题的工作中, 多数工作聚焦于预测一个动作序列以构成最终的解^[25,54,80]。由于路径问题的广泛影响力, 机器学习求解排列型组合优化的先驱工作^[25]首先提出了在监督学习的范式下, 利用时序神经网络和注意力机制^[53]求解旅行商问题。在后续工作中^[54], 研究者提出了一种从图到序列的学习、求解框架, 其中的关键技术包括基于图神经网络的图嵌入^[105]以及基于 Q 值学习 (Q-learning) 的深度强化学习^[106]。Li 等^[107]通过引入问题简化和树搜索方法, 进一步扩展了从图到序列的框架。Kool 等^[80]为路径求解的研究提

¹<https://www.math.uwaterloo.ca/tsp/concorde.html>

出了基础的学习框架，包括基于注意力机制^[53]的特征提取网络、基于搜索的后处理步骤和基于强化学习的训练方法。该框架被后续的工作不断完善，包括改进强化学习算法^[61]、改进特征提取网络^[55]、改进后处理步骤^[108]等。不过，这些纯神经网络的路径求解方法尚缺乏和传统算法结合的灵活性，同时也会面临强化学习的典型挑战：稀疏的奖励函数和过大的动作空间。文献^[84]和本文的4.4小节展示了如何通过融合传统算法应对这些挑战。

2.2.3 基数约束优化问题

基数约束 (cardinality constraint) 的组合优化问题是一类真实世界中广泛存在的排列型问题，其形式等价于1.2小节讨论的前 k 个排列——在大小为 n 的候选集合中，只能选择其中的 k 个。换言之，在最终的解中只能出现 k 个非零项，即 $\|\mathbf{x}\|_0 \leq k$ 。基数约束广泛存在于许多真实的优化场景中，例如运筹领域的设施选址问题^[109] (寻找 k 个位置放置新设施)、数据挖掘中的影响力最大化问题^[110] (选取社交网络中的 k 个种子用户使得影响力最大)、金融领域的投资组合优化问题^[111] (选取至多 k 个资产进行投资，以控制运营费用)。基数约束优化中，次模优化 (submodular optimization)^[112] 是一个绕不开的概念。简单概括，次模优化是“边际效益递减”这一经济学属性在数学上的描述，许多基数约束问题都属于次模优化。次模的性质带来了 NP 难排列型组合优化问题中少有的可近似性，即贪心算法可以达到 $(1 - \frac{1}{e})$ 的近似率 (假设是最小化问题)。因此，贪心算法是求解此类问题的一种简单、有效的算法。

针对基数约束的可微分学习，存在两种技术路径。一类工作从可微分的排序方法出发，构造可微分的“前 k 个选取” (Topk) 算法。对排序算法的可微分化研究开始于^[113]，对 n 个数字的排序等价于一个最优传输问题^[114]：设定 $(1, 2, 3, \dots, n)$ 共 n 个目的地，输入的 n 个数字作为出发地，出发地与目的地之间的差值作为距离。对 n 个数字的排序等价于寻找一个最优的运输方案，最小的数字被运送到 1，最大的数字被运送到 n 。上述最优传输问题等价于排列型组合优化中经典的线性指派问题，一方面可以用匈牙利算法^[10]进行求解，另一方面也可以用可微分的 Sinkhorn 算法^[115]进行近似求解。最新的研究工作^[81,116]遵循了这个思路，基于 Sinkhorn 算法构造了可微分的 Topk 选取算法，可用于基数约束的学习和求解。另一类工作提出直接采用不可微分的排序算法 (例如快速排序)，对算法添加扰动以估计其近似梯度。站在神经网络的视角，可以将排序算法看作一个黑盒模块，利用黑盒微分方法^[72]添加扰动并估计其梯度。另外，Berthet 等^[74]提出了在排序前的集合中添加大量随机噪声，根据随机扰动后的排序分布进行梯度估计。

2.3 排列型组合优化的开源框架

现有的排列型组合优化开源算法可分为基于整数规划的通用算法和面向独立问题的专用算法。总体而言，基于传统算法的开源框架在过去有一定的发展；基于机器学习的求解算法尚无成熟的开源框架。

由于整数规划形式的泛用性，开源的整数规划求解器可以被用于求解多数排列型组合优化问题。但是，如2.1.1小节所讨论，开发一个整数规划算法是一个复杂的系统工程，需要软件工程、优化算法等多方面的专家知识。整数规划求解领域在过去一直由商业软件主导，例如属于 IBM 公司的 CPLEX 求解器²、由 Gurobi 公司开发的 Gurobi 求解器^[31]以及国内杉树科技公司开发的 COPT 求解器³。来自柏林祖斯研究所 (Zuse Institute Berlin) 的 Tobias Achterberg 博士打破了该领域商业软件的垄断，他基于 C 语言开发了整数规划求解器 SCIP^[30]，并将源代码开源。SCIP 拥有一个活跃的开源社区，是目前性能最强的开源整数规划求解器。

针对本文所关注的排列型组合优化问题，以图匹配问题为例，目前存在若干开源工具，其中多数是随着技术论文一起发表的开源代码。在这些开源代码中，作者通常同时实现了其他方法以进行对比实验。例如，ZAC_GM 是传统图匹配论文^[117]的官方开源代码，提供了若干经典二图匹配算法的 Matlab 实现；multiway 是多图匹配论文^[91]的官方开源代码，提供了若干经典多图匹配算法的 Matlab 实现。需要注意的是，虽然上述两个代码库中的算法是开源的，它们所依赖的 Matlab 依然是一个商业软件。因此，匹配问题的（纯正）开源工作尚有发展、完善的空间。机器学习图匹配方法的作者 Rolinek^[82]和 Fey^[78]分别开源了他们的求解模型，但没有与其他算法进行比较。对于更一般的二次指派问题，其开源求解框架在国际上依然是空白。

2.4 本章小结

本章分别讨论了排列型组合优化求解的方法论、典型问题以及开源框架，为下文中的技术创新提供了背景知识。在方法论层面，本章讨论了机器学习时代之前的传统算法、纯机器学习的求解算法、机器学习与传统算法融合的方法。在问题层面，本章讨论了三类典型的排列型组合问题：匹配问题、路径问题和基数约束优化问题，并讨论了相关求解方法的发展现状。最后，排列型组合优化的开源框架尚未成熟，还有进一步发展、完善的空间。

²<https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer>

³<https://www.shanshu.ai/solver>

第三章 排列型组合优化问题的纯机器学习求解

3.1 本章引言

利用机器学习、尤其是神经网络方法直接求解排列型组合优化是一个经典的研究方向。早在 1985 年，Hopfield 和 Tank 就提出将网络的训练直接转化成旅行商问题的求解^[1]。随着算力尤其是 GPU 算力的提升，伴随着深度学习技术的发展，利用神经网络方法直接求解排列型组合优化逐渐成为了一个热门研究方向。本章分三节介绍本文在纯机器学习求解方法的理论和技术贡献：面向图匹配问题的求解，3.2 节介绍了一种（跨）图嵌入的神经网络方法；面向更一般的二次指派问题，3.3 节介绍了首个基于神经网络的通用二次指派求解器；以基数约束问题的求解为例，3.4 节研究了纯神经网络求解方法中约束满足的理论保证，并且以理论为指导，改进了现有的基数约束神经网络。

3.2 面向图匹配求解的神经网络设计

图匹配 (graph matching) 旨在构造两个图结构之间的节点匹配关系。通过引入机器学习，可以在考虑噪声的同时高效地建模图匹配中的相似度函数，使得图匹配问题在数学上的最优解具有正确的物理意义。在技术上，本节采用了神经网络学习节点、边的特征以及图匹配问题的相似度函数，通过图嵌入方法将问题简化，进而进行可微的精确求解。所有神经网络模块在具有组合性质的排列损失函数下进行端到端的监督训练。本节介绍的神经网络方法具有较强的灵活性，其中排列损失函数与节点数目无关、图神经网络也可处理任意节点数目的图。此外，同一个神经网络可以匹配来自多个类别的图。多个评测集上的实验结果印证了本方法的性能优势，特别是与传统图匹配算法的比较优势。本节实验还展现了深度图匹配方法在不同数据集和类别之间的泛化性，以及匹配外点时的鲁棒性。

3.2.1 图匹配问题描述

作为一个经典的排列型组合优化问题，图匹配问题具有 NP 难的复杂性^[14]，因此在研究中通常采用近似方法求解得到非精确的结果。考虑经典的二图匹配设定，给定图结构 \mathcal{G}^1 和 \mathcal{G}^2 ，图匹配具有如下数学形式，被称为二次指派问题 (quadratic assignment problem, QAP)，

$$\begin{aligned} \max J(\mathbf{X}) &= \text{vec}(\mathbf{X})^\top \mathbf{K} \text{vec}(\mathbf{X}), \\ \text{s.t. } \mathbf{X} &\in \{0, 1\}^{n_1 \times n_2}, \quad \mathbf{X} \mathbf{1}_{n_2} = \mathbf{1}_{n_1}, \quad \mathbf{X}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}. \end{aligned} \quad (3.1)$$

其中 \mathbf{X} 是包含点匹配信息的 0/1 排列矩阵, 被称为“相似度矩阵”的 $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ 包含了两个图结构中点到点、边到边的相似度信息。 \mathbf{K} 即为该问题的参数。一种计算 \mathbf{K} 的方法是采用高斯核函数 $\mathbf{K}_{ia,jb} = \exp\left(-\frac{(\mathbf{f}_{ij}-\mathbf{f}_{ab})^2}{\sigma^2}\right)$, 其中 \mathbf{f}_{ij} 是边 ij 所对应的特征向量。当 $ia = jb$ 时, 节点的相似度也可以用上述公式计算。公式 (3.1) 假设了 $n_1 \leq n_2$ 。

特别地, 公式 (3.1) 被称作 Lawler 形式的二次指派问题 (Lawler's QAP)^[118]。作为最一般的问题形式, 它可以包含其他的特殊形式, 例如 Koopmans-Beckmann 形式的二次指派问题 (Koopmans-Beckmann's QAP, 简称为 KB-QAP)^[14]:

$$J(\mathbf{X}) = \text{tr}(\mathbf{X}^\top \mathbf{F}^1 \mathbf{X} \mathbf{F}^2) + \text{tr}(\mathbf{K}^P \mathbf{X}). \quad (3.2)$$

其中 $\mathbf{F}^1 \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{F}^2 \in \mathbb{R}^{n_2 \times n_2}$ 是 \mathcal{G}^1 , \mathcal{G}^2 对应的带权重的邻接矩阵, 矩阵 $\mathbf{K}^P \in \mathbb{R}^{n_1 \times n_2}$ 包含点到点的相似度权重。Koopmans-Beckmann 形式可以经过如下公式转换为 Lawler 形式的二次指派问题: $\mathbf{K} = \mathbf{F}^2 \otimes \mathbf{F}^1 + \text{diag}(\text{vec}(\mathbf{K}^P))$ 。

此外, 最近的研究工作尝试研究了比二阶更高阶的相似度信息。在这类被称为超图匹配 (hypergraph matching) 的工作中, 目标函数通常被写作张量-向量的按维度归一化^[119-122]:

$$\begin{aligned} \mathbf{x}^* &= \arg \max(\mathbf{H} \otimes_1 \mathbf{x} \otimes_2 \mathbf{x} \dots \otimes_m \mathbf{x}) \\ \text{s.t. } \mathbf{X} \mathbf{1}_{n_2} &= \mathbf{1}_{n_1}, \quad \mathbf{X}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}, \quad \mathbf{x} = \text{vec}(\mathbf{X}) \in \{0, 1\}^{n_1 n_2 \times 1} \end{aligned} \quad (3.3)$$

其中 m 是相似度的阶数, \mathbf{H} 是包含 m 阶的相似度张量, 包含了高阶的超边相似度。其中, \otimes_k 代表了张量乘法^[123]。张量乘法定义如下:

$$(\mathbf{H} \otimes_k \mathbf{x})_{\dots, i_{k-1}, i_{k+1}, \dots} = \sum_{i_k=1}^{n_1 n_2} \mathbf{H}_{\dots, i_{k-1}, i_k, i_{k+1}, \dots} \cdot \mathbf{x}_{i_k} \quad (3.4)$$

其中, \otimes_k 可以看作是在第 k 维上进行张量的边缘化 (marginalization)。张量乘法的详细细节可以参考文献^[120] 的 3.1 节。为了提高优化算法的效率, 现有的超图匹配工作均假设相似度张量与超边的下标顺序无关, 即超边是无向的。

最后, 多图匹配也是一个重要的研究方向。多图匹配的目标旨在同时匹配多个图结构。过去的研究主要打造了两类典型的多图匹配方法: 第一类方法将多图匹配问题转化为多轮迭代, 每轮迭代中只需求解一个二图匹配问题^[95, 124]; 第二类方法首先获取成对的预匹配结果, 随后采用平滑化的后处理方法进行匹配修正^[91, 125-127]。和二图匹配相比, 多图匹配的优势在于融合了多图信息后的匹配结果对噪声、外点更加鲁棒。

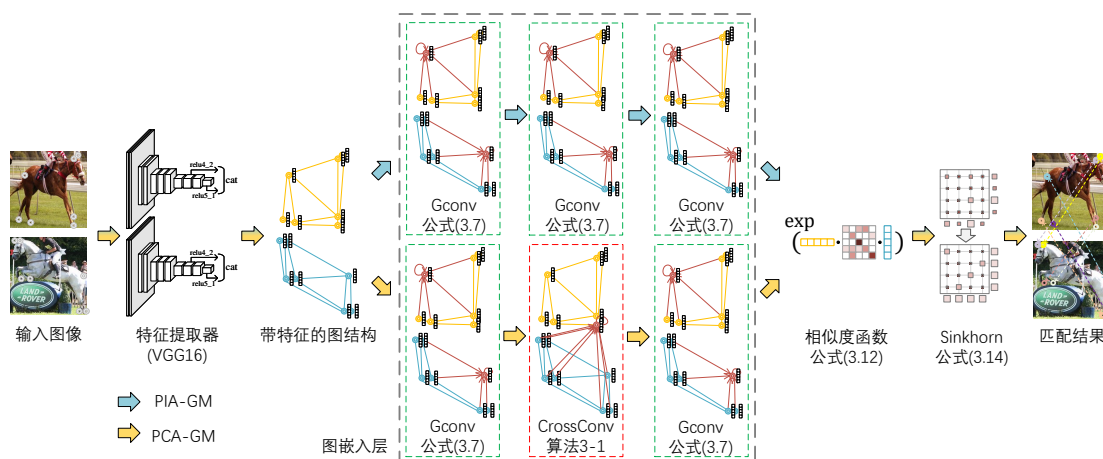


图 3-1 单次前向计算的深度学习图匹配方法 (PIA-GM、PCA-GM) 概览。
 Figure 3-1 Overview of vanilla feed-forward methods (PIA-GM, PCA-GM) for deep learning of graph matching.

3.2.2 研究动机

基于上一节的描述，不论是二图、超图还是多图匹配算法，都需要面对一个基础性问题：如何合理地设计点特征、边特征提取器以及相似度模型，将真实世界的匹配任务转化为等价的数学形式。这个数学形式的最优解需要对应到正确且合理的匹配结果。但是，人工设计的相似度函数（例如基于欧氏距离的高斯核函数）可能缺乏足够的模型表达力，会导致数学形式的最优解偏离出正确的匹配结果。在考虑噪声、外点等干扰因素时，这种挑战更加明显。

为了解决上述挑战，一类最近的工作聚焦于学习图匹配的相似度函数，以提升相似度建模阶段的模型容量。此类工作的愿景是完全取代现有的、手工设计的相似度函数。因为图匹配问题的目标函数由相似度决定，通过合适的相似度函数，图匹配问题在数学上的最优解可以对应到正确的匹配关系。在一定程度上，上述基于学习的特征和相似度方法和现有的求解方法^[33-34,120,123]是正交的，因为现有的求解方法通常假设存在给定的相似度。

基于上述观察，本章节的研究动机来自如下两个方面：

1) **将图匹配问题转换为线性指派问题**。通过图嵌入网络，特别是图卷积网络，将图结构嵌入到节点特征向量中，可以将图匹配问题转换为线性指派问题。嵌入的节点特征将包含节点周围的结构信息，使得匹配过程中可以包含高阶（超过二阶）的信息。通过这种方式，该模型规避了非常具有挑战性的 QAP 问题求解。这种设计也允许在训练和测试中处理具有不同节点数目的图。据作者所知，这是首次在图匹配学习中采用深度图嵌入网络。

2) **设计面向组合优化的排列损失函数**。面向排列型组合优化，本节开发了一种针对 Sinkhorn 网络的排列损失函数。输入非负矩阵，模型通过 Sinkhorn 算法得到了双随机矩阵。将其视为预测的匹配结果，分类任务中常用的交叉熵损失可以衡量双随机矩阵与匹配真值之间的差异。这种面向组合优化的损失函数还

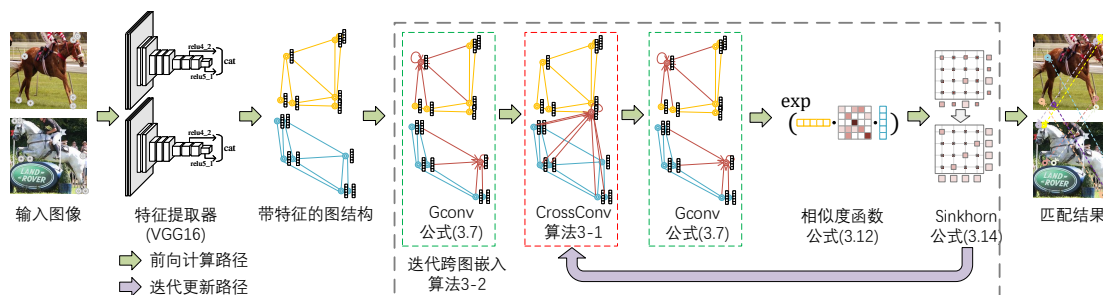


图 3-2 迭代跨图嵌入的深度学习图匹配方法 (IPCA-GM) 概览。

Figure 3-2 Overview of iterative cross-graph embedding method (IPCA-GM) for deep learning of graph matching.

可以处理大小不同的图。图的规模变化一直是图匹配领域的挑战之一，例如，在基于结构支持向量机的学习模型中^[101]，图的节点数目在训练时必须固定。然而，这种假设在实践中通常是不现实的。据作者所知，这是首次在图匹配中采用排列损失函数。

3.2.3 算法描述

在本节中，作者将介绍三种匹配图结构 $\mathcal{G}^1 = (V_1, \mathcal{E}_1)$ 与图结构 $\mathcal{G}^2 = (V_2, \mathcal{E}_2)$ 的模型：i) PIA-GM (**p**ermutation loss and **i**ntra-graph **a**ffinity based **g**raph **m**atching learning)，基于排列损失与图内相似度特征的图匹配学习模型；ii) PCA-GM (**p**ermutation loss and **c**ross-graph **a**ffinity based **g**raph **m**atching learning) 基于排列损失与图间相似度特征的图匹配学习模型；iii) IPCA-GM (**i**terative **p**ermutation loss and **c**ross-graph **a**ffinity based **g**raph **m**atching learning) 基于排列损失与迭代图间相似度特征的图匹配学习模型。这些神经网络模型都能同时建模图像特征与图结构特征。模型中采用的 Sinkhorn 网络层以可微分的方式预测匹配关系，同时允许损失函数的反向传播。PIA-GM 采用了图内特征更新的策略；PCA-GM 使用了一层额外的跨图传播模块，允许特征在多个图之间传播；IPCA-GM 在跨图传播中采用迭代更新。图3-1展示了基于排列损失函数的图内相似度学习方法 (PIA-GM) 和跨图相似度学习方法 (PCA-GM)，图3-2展示了基于跨图迭代更新的相似度学习方法 (IPCA-GM)。

本节提出的模型均包括了一个（可选的）CNN 网络作为图像特征提取器、一个节点嵌入模块、一个相似度函数以及排列预测模块。处理图像输入时，CNN 网络（此处为 VGG16^[128]）提取 RGB 图片输入中的特征信息，并将它们组成图结构节点的特征向量。节点特征由一个（跨图的）节点嵌入网络更新，得到包含图结构信息的特征向量。本节的 CNN 网络、图嵌入网络、相似度函数权重均可端到端地学习。

图像特征提取。若问题输入为图像，本节使用了一个 CNN 网络来提取关键点的特征向量，其中关键点的特征向量由 CNN 输出的特征图像 (feature map) 上

插值得到。对于图片 I_s ，从特征图像中为关键点 P_{si} 提取的特征向量为：

$$\mathbf{h}_{si}^{(0)} = \text{Interp}(P_{si}, \text{CNN}(I_s)), \quad (3.5)$$

其中 $\text{Interp}(P, X)$ 代表在高维张量 X 代表的特征空间中，基于关键点位置 P 进行双线性插值得到的结果。 $\text{CNN}(I)$ 则代表将 RGB 图片 I 输入卷积神经网络，输出对应特征空间的张量。受孪生网络^[129] (Siamese network) 的启发，两张输入的图片均由同一个 CNN 网络处理，得到图像特征空间的特征图像。为了同时利用 CNN 网络提取出的局部结构信息与全局的语义特征，从 CNN 的不同深度中抽取，随后拼接得到节点的特征向量。为了与前序工作^[16]的公平地对比，在实验中采用了在 ImageNet 分类任务^[130]上预训练的 VGG16 网络^[128]作为主干网络。

图内节点嵌入。 本节提出的模型包含了一个图内节点嵌入 (intra-graph node embedding) 模块。过去的研究已经工作证明，与只使用一阶信息的点匹配^[131-132]相比，充分利用图结构信息能够建立更加鲁棒的匹配结果^[98]。PIA-GM 利用一个多层的节点嵌入模型构建了二阶和高阶的图相似度信息。节点嵌入模型采用类似图卷积网络^[133]的消息传递机制，节点的特征由其邻接节点以及该节点自身作为信息源，进行高效的特征更新：

$$\mathbf{m}_{si}^{(k)} = \frac{1}{|(i, j) \in \mathcal{E}_s|} \sum_{j: (i, j) \in \mathcal{E}_s} f_{msg}(\mathbf{h}_{sj}^{(k-1)}), \quad (3.6a)$$

$$\mathbf{n}_{si}^{(k)} = f_{node}(\mathbf{h}_{si}^{(k-1)}), \quad (3.6b)$$

$$\mathbf{h}_{si}^{(k)} = f_{update}(\mathbf{m}_{si}^{(k)}, \mathbf{n}_{si}^{(k)}), \quad (3.6c)$$

其中公式 (3.6a) 代表沿着边的信息传播， $f_{msg}(\cdot)$ 是其信息传递函数。根据 GCN 中的常规操作，为了避免由于每个节点的邻接节点总数不同而造成偏差，由邻接节点传播而来的累加信息被当前节点的邻接节点总数归一化。公式 (3.6b) 代表每个节点内部的特征更新方式，它包含了节点的自更新函数 $f_{node}(\cdot)$ 。基于更新函数 $f_{update}(\cdot)$ ，公式 (3.6c) 收集了所有更新到节点 i 的特征信息。其中， $f_{msg}(\cdot)$, $f_{node}(\cdot)$, $f_{update}(\cdot)$ 可以取任意可微分的、向量到向量的映射。本节采用了带 ReLU 激活函数的神经网络来实现 $f_{msg}(\cdot)$, $f_{node}(\cdot)$ ， $f_{update}(\cdot)$ 是简单的求和函数。在本节的其他部分，公式 (3.6) 被写作 k 至 $(k+1)$ 层 GNN 的图卷积 (GConv)：

$$\{\mathbf{h}_{si}^{(k+1)}\} = \text{GConv}(\mathbf{A}_s, \{\mathbf{h}_{si}^{(k)}\}), \quad i \in \mathcal{V}_s, \quad (3.7)$$

它代表了节点嵌入网络。消息传递路径由图的邻接矩阵 $\mathbf{A}_s \in \{0, 1\}^{n_s \times n_s}$ 所表示。通过多次迭代公式 (3.7) 中的图卷积层，构建了一个多层的 GNN：

$$\{\mathbf{h}_{si}\} = \text{GNN}(\mathbf{A}_s, \{\mathbf{h}_{si}^{(0)}\}), \quad i \in \mathcal{V}_s, \quad (3.8)$$

算法 3-1 单次前向计算的跨图嵌入 (CrossEmb)

输入: 第 $(k-1)$ 层特征 $\{\mathbf{h}_{1i}^{(k-1)}, \mathbf{h}_{2j}^{(k-1)}\}_{i \in \mathcal{V}_1, j \in \mathcal{V}_2}$

- 1: # 预测相似度, 见公式 (3.12, 3.14)
- 2: 根据公式 (3.12), 从 $\{\mathbf{h}_{1i}^{(k-1)}, \mathbf{h}_{2j}^{(k-1)}\}$ 构造 $\hat{\mathbf{M}}$;
- 3: $\hat{\mathbf{S}} \leftarrow \text{Sinkhorn}(\hat{\mathbf{M}})$;
- 4: # 跨图特征更新, 见公式 (3.9)
- 5: $\{\mathbf{h}_{1i}^{(k)}\} \leftarrow \text{CrossConv}(\hat{\mathbf{S}}, \{\mathbf{h}_{1i}^{(k-1)}\}_{i \in \mathcal{V}_1}, \{\mathbf{h}_{2j}^{(k-1)}\}_{j \in \mathcal{V}_2})$;
- 6: $\{\mathbf{h}_{2j}^{(k)}\} \leftarrow \text{CrossConv}(\hat{\mathbf{S}}^\top, \{\mathbf{h}_{2j}^{(k-1)}\}_{j \in \mathcal{V}_2}, \{\mathbf{h}_{1i}^{(k-1)}\}_{i \in \mathcal{V}_1})$;

输出: 第 k 层特征 $\{\mathbf{h}_{1i}^{(k)}, \mathbf{h}_{2j}^{(k)}\}_{i \in \mathcal{V}_1, j \in \mathcal{V}_2}$

其中 $\mathbf{h}_i^{(0)}$ 是由公式 (3.5) 中所示的 CNN 网络得到的图像特征向量。

跨图节点嵌入。在图内节点嵌入的基础上, 跨图的节点嵌入 (cross-graph node embedding) 能够进一步提升匹配模型的性能。算法3-1总结了跨图节点嵌入的流程。在跨图节点嵌入中, 首先使用了浅层嵌入网络中预测得到的相似度信息, 通过构建相似度函数以及 Sinkhorn 迭代 (详见三段后对 Sinkhorn 的正式介绍), 预测一个双随机的相似度矩阵。由此预测得到的相似度矩阵 $\hat{\mathbf{S}}$ 编码了两个图的节点之间的相似度信息。跨图的信息传播方式与公式 (3.7) 中的图内卷积类似, 其中邻接矩阵被 $\hat{\mathbf{S}}$ 所取代, 且特征向量的信息从一个图向另一个图传递。

$$\mathbf{m}_{1i}^{(k)} = \sum_{j \in \mathcal{V}_2} \hat{\mathbf{S}}_{i,j} f_{\text{msg-cross}}(\mathbf{h}_{2j}^{(k-1)}), \quad (3.9a)$$

$$\mathbf{n}_{1i}^{(k)} = f_{\text{node-cross}}(\mathbf{h}_{1i}^{(k-1)}), \quad (3.9b)$$

$$\mathbf{h}_{1i}^{(k)} = f_{\text{update-cross}}(\mathbf{m}_{1i}^{(k)}, \mathbf{n}_{1i}^{(k)}), \quad (3.9c)$$

其中, $f_{\text{msg-cross}}(\cdot), f_{\text{node-cross}}(\cdot)$ 为恒等映射 (identical mapping), $f_{\text{update-cross}}(\cdot)$ 首先将两个输入的特征张量拼接 (concatenate), 再将其输入全连接网络。考虑一对图 $\mathcal{G}^1 = (\mathcal{V}_1, \mathcal{E}_1), \mathcal{G}^2 = (\mathcal{V}_2, \mathcal{E}_2)$, 公式 (3.9) 中刻画的跨图消息传递形式可以表示为

$$\{\mathbf{h}_{1i}^{(k+1)}\}_{i \in \mathcal{V}_1} = \text{CrossConv}(\hat{\mathbf{S}}, \{\mathbf{h}_{1i}^{(k)}\}_{i \in \mathcal{V}_1}, \{\mathbf{h}_{2j}^{(k)}\}_{j \in \mathcal{V}_2}), \quad (3.10a)$$

$$\{\mathbf{h}_{2j}^{(k+1)}\}_{j \in \mathcal{V}_2} = \text{CrossConv}(\hat{\mathbf{S}}^\top, \{\mathbf{h}_{2j}^{(k)}\}_{j \in \mathcal{V}_2}, \{\mathbf{h}_{1i}^{(k)}\}_{i \in \mathcal{V}_1}), \quad (3.10b)$$

其中 $\hat{\mathbf{S}}$ 是由浅层网络预测的 \mathcal{G}^2 与 \mathcal{G}^1 之间的对应关系; 同时 $\hat{\mathbf{S}}^\top$ 代表同样的、从 \mathcal{G}^1 到 \mathcal{G}^2 的对应关系。

迭代跨图节点嵌入。在算法3-1中讨论的跨图嵌入方法考虑了基于浅层网络特征的跨图匹配信息, 在实验中展现出了优越性。但是, 算法3-1的局限性在于它只考虑了简单的单次前向计算, 仍有进一步提升的空间。考虑到更精确的跨图匹配精度很可能会带来更优质的图嵌入特征, 反之亦然, 此处进一步提出了迭代跨图节点嵌入, 更精确地预测跨图更新矩阵 $\hat{\mathbf{S}}$ 。算法3-2的 $\text{IterCrossEmb}(\cdot)$ 函数

算法 3-2 迭代更新的跨图嵌入 (IterCrossEmb)

输入: CNN 特征 $\{\mathbf{h}_{1i}^{(0)}, \mathbf{h}_{2j}^{(0)}\}_{i \in \mathcal{V}_1, j \in \mathcal{V}_2}$; 迭代次数 K

- 1: # 首次图内消息传递, 见公式 (3.6)
- 2: $\{\mathbf{h}_{si}^{(1)}\} \leftarrow \text{GConv}_1(\mathbf{A}_s, \{\mathbf{h}_{si}^{(0)}\})$;
- 3: 将 $\hat{\mathbf{S}}^{(0)}$ 初始化为全 0 矩阵;
- 4: $\hat{\mathbf{S}}^{(0)} \leftarrow \mathbf{0}^{n_1 \times n_2}$;
- 5: **for** $k \leftarrow \{1..K\}$ **do**
- 6: # 跨图消息传递, 见公式 (3.9)
- 7: $\{\mathbf{h}_{1i}^{(2)}\} \leftarrow \text{CrossConv}(\hat{\mathbf{S}}^{(k-1)} \{\mathbf{h}_{1i}^{(1)}\}, \{\mathbf{h}_{2j}^{(1)}\})$;
- 8: $\{\mathbf{h}_{2j}^{(2)}\} \leftarrow \text{CrossConv}(\hat{\mathbf{S}}^{(k-1)\top}, \{\mathbf{h}_{2j}^{(1)}\}, \{\mathbf{h}_{1i}^{(1)}\})$;
- 9: # 第二次图内消息传递, 见公式 (3.6)
- 10: $\{\mathbf{h}_{si}^{(3)}\} \leftarrow \text{GConv}_2(\mathbf{A}_s, \{\mathbf{h}_{si}^{(2)}\})$;
- 11: # 预测匹配关系, 见公式 (3.12, 3.14)
- 12: 根据公式 (3.12), 从 $\{\mathbf{h}_{1i}^{(3)}\}, \{\mathbf{h}_{2j}^{(3)}\}$ 构造 $\hat{\mathbf{M}}$;
- 13: $\hat{\mathbf{S}}^{(k)} \leftarrow \text{Sinkhorn}(\hat{\mathbf{M}})$;
- 14: **end for**

输出: 节点嵌入特征 $\{\mathbf{h}_{1i}^{(3)}, \mathbf{h}_{2j}^{(3)}\}_{i \in \mathcal{V}_1, j \in \mathcal{V}_2}$

总结了跨图节点嵌入方法中图嵌入层（包括图内的嵌入层）的计算方式。

该迭代算法中, $\hat{\mathbf{S}}^{(0)}$ 首先被初始化为全 0 矩阵, 随后从 $\hat{\mathbf{S}}^{(k-1)}$ 预测 $\hat{\mathbf{S}}^{(k)}$, 利用了包括跨图嵌入层（算法第6行至第8行）和图内嵌入层（算法第10行）。基于嵌入的特征可以计算一个相似度矩阵（算法第12行），进而通过 Sinkhorn 算法得到一个双随机矩阵 $\hat{\mathbf{S}}$ （算法第13行）。双随机矩阵（doubly stochastic matrix）的定义是行和、列和均为 1 的矩阵, 是排列矩阵在连续域上的松弛形式。需要注意的是, 此处预测的匹配阵 $\hat{\mathbf{S}}$ 被视为跨图更新的路径。实验结果表明, 上述迭代跨图嵌入方法可以进一步提升匹配精度。

相似度函数。通过前述的节点嵌入模型, 两个图之间结构的相似度信息被编码在了节点嵌入空间内。因此, 模型允许将公式 (3.1) 中传统的二阶相似度矩阵 \mathbf{K} 简化成线性的形式。定义 \mathbf{h}_{1i} 为第一个图中点 i 的特征向量, \mathbf{h}_{2j} 为第二个图中点 j 的特征向量:

$$\mathbf{M}_{i,j}^{(0)} = f_{\text{aff}}(\mathbf{h}_{1i}, \mathbf{h}_{2j}), \quad i \in \mathcal{V}_1, j \in \mathcal{V}_2. \quad (3.11)$$

相似度矩阵 $\mathbf{M}^{(0)} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$ 包含了二图所有节点之间的相似度得分。 $\mathbf{M}_{i,j}^{(0)}$ 代表图 1 的节点 i 与图 2 的节点 j 之间, 包含了高阶图结构信息相似度。

模型中, $f_{\text{aff}}(\cdot)$ 被设定为一个双线性 (bi-linear) 函数, 随后对它加以一个指

数函数，以保证相似度矩阵中所有的元素均为非负的⁴：

$$\mathbf{M}_{i,j}^{(0)} = \exp\left(\frac{\mathbf{h}_{1i}^\top \mathbf{A} \mathbf{h}_{2j}}{\tau}\right). \quad (3.12)$$

考虑 d 维的特征向量，即 $\forall i \in \mathcal{V}_1, j \in \mathcal{V}_2, \mathbf{h}_{1i}, \mathbf{h}_{2j} \in \mathbb{R}^{d \times 1}$ 。 $\mathbf{A} \in \mathbb{R}^{d \times d}$ 包含了相似度函数中可学习的权重。 τ 是 Sinkhorn 算法中熵正则项对应的超参数^[114]，对于 $\tau > 0$ ，若 $\tau \rightarrow 0^+$ ，那么公式 (3.12) 的判别能力就会增强，但同时训练过程中梯度传播的精度和稳定程度都会降低。

Sinkhorn 网络与线性指派问题求解。 给定由公式 (3.12) 得到的相似度矩阵，Sinkhorn 网络可以求解对应的、基于相似度矩阵的线性指派问题。Sinkhorn 算法接受任何非负的方阵作为输入，输出一个双随机矩阵 (doubly stochastic matrix)。双随机矩阵是离散化的排列矩阵的松弛形式。Sinkhorn 网络已经在研究中被证实具有学习并预测排列形式的能力^[113,134]。对于一个方阵 $\mathbf{M}^{(k-1)} \in \mathbb{R}^{n \times n}$ ，Sinkhorn 运算符可以写为

$$\mathbf{M}^{(k)'} = \mathbf{M}^{(k-1)} \oslash (\mathbf{M}^{(k-1)} \mathbf{1} \mathbf{1}^\top), \quad (3.13a)$$

$$\mathbf{M}^{(k)} = \mathbf{M}^{(k)'} \oslash (\mathbf{1} \mathbf{1}^\top \mathbf{M}^{(k)'}), \quad (3.13b)$$

\oslash 代表矩阵中逐元素的除法， $\mathbf{1} \in \mathbb{R}^{n \times 1}$ 是一个元素全为 1 的列向量。Sinkhorn 算法的步骤即为交替迭代公式 (3.13a) 的行归一化、以及公式 (3.13b) 的列归一化。若输入的 $\mathbf{M}^{(k-1)}$ 不是方阵，亦可添加虚拟变量 (dummy variable)，例如补 0，将其补全至方阵后再进行计算。

通过不断迭代公式 (3.13) 直至收敛，我们最终得到一个双随机矩阵。在训练过程中，该双随机矩阵 \mathbf{S} 是模型的预测输出。

$$\mathbf{S} = \text{Sinkhorn}(\mathbf{M}^{(0)}). \quad (3.14)$$

在测试过程中，作为后处理步骤，作者在矩阵 \mathbf{S} 上运行匈牙利算法^[10]将输出结果离散化，得到排列矩阵。在 Sinkhorn 算法中，只存在矩阵乘法与按元素除法两种操作，这两种操作均可微分。因此，Sinkhorn 具有完全可微的特性，利用 PyTorch 等深度学习框架的自动求导功能^[135]，能够在工程上高效地实现它的反向传播。

排列-交叉熵损失函数。 本节提出的深度图匹配方法直接利用了点与点的真值对应关系，即排列矩阵，作为图匹配网络端到端训练的监督信息。由于公式 (3.14) 中所述的 Sinkhorn 网络能够将任何非负矩阵变换为双随机矩阵，此处提出

⁴作者还尝试了更加灵活的全连接层作为相似度函数，但实验结果显示该指数函数的形式更加简单有效，且训练过程更加稳定。

了一种基于线性指派的排列损失函数 (permutation loss), 在训练过程中, 用于衡量预测的匹配结果与真值匹配结果之间的一致性程度。

交叉熵损失函数被用于端到端的模型训练。已知代表真实匹配情况的排列矩阵 \mathbf{X} , 即可计算预测值 \mathbf{S} 与真值 \mathbf{X} 之间的交叉熵。该函数被命名为排列损失函数。排列损失函数是用于训练本节模型的主要方法:

$$L_{perm} = - \sum_{i \in V_1, j \in V_2} (\mathbf{X}_{i,j} \log \mathbf{S}_{i,j} + (1 - \mathbf{X}_{i,j}) \log(1 - \mathbf{S}_{i,j})) \quad (3.15)$$

需要注意的是, 另一项深度图匹配工作 GMN^[16] 采用了一种基于像素偏移量的损失函数 (offset loss), 在文中被称为“偏移损失” (displacement loss)。特别地, 偏移损失函数使用模型预测的匹配关系, 加权计算了一个偏移向量 \mathbf{d} 。偏移损失函数即为模型预测的偏移向量与使用真值对应关系计算的偏移向量的差值。

$$\mathbf{d}_i = \sum_{j \in V_2} (\mathbf{S}_{i,j} P_{2,j}) - P_{1,i} \quad (3.16a)$$

$$L_{off} = \sum_{i \in V_1} \sqrt{\|\mathbf{d}_i - \mathbf{d}_i^{gt}\|^2} + \epsilon \quad (3.16b)$$

其中 $\{P_{1,i}\}, \{P_{2,j}\}$ 分别是关键点在图片 1 与图片 2 中的坐标集合, 而 ϵ 则是为了数值稳定性设定的一个小的常数。作为对比, 本节采用的基于交叉熵的排列损失函数可以使模型在端到端训练的过程中直接学习求解线性指派问题。完整的训练流程可参考算法3-3。

算法设计细节的进一步讨论。

相似度矩阵与节点嵌入。现存的图匹配方法主要致力于使用预先明确定义的相似度矩阵或相似度张量, 建模二阶^[34-35]以及高阶^[121,123]的相似度特征。对于使用至多二阶信息的图匹配而言, 相似度信息可以被编码在一个 $n_1 n_2 \times n_1 n_2$ 大小的相似度矩阵中。研究者们提出了许多基于优化的方法, 以最大化匹配结果所对应的相似度。作为对比, 本节将相似度建模的问题诉诸节点嵌入技术, 主要基于两个考量: 首先, 通过节点嵌入技术, 可以将相似度矩阵的空间复杂度降低至 $n_1 \times n_2$ (作为对比, 传统的二阶相似度矩阵需要占用 $n_1 n_2 \times n_1 n_2$ 的空间); 其次, 二阶形式的相似度函数只能够表达至多二阶的信息, 而节点嵌入技术能够直接编码更高阶的图结构信息乃至跨图信息。

Sinkhorn 网络与谱匹配。作为已发表的图匹配工作, GMN^[16] 采用了可以微分的谱匹配 (spectral matching, SM^[35]) 求解二次指派问题。与之对比, 本文采用了 *Sinkhorn* 网络完成了对应的工作。事实上, *Sinkhorn* 网络的输入具有 $\mathcal{O}(n_1 n_2)$ 的复杂度, 而谱匹配的输入具有 $\mathcal{O}(n_1^2 n_2^2)$ 的复杂度。此外, 在实验中谱匹配需要更多次迭代才能收敛, 而更多的迭代次数会为梯度的反向传播带来负面影响。实际上, 谱匹配算法是为了求解图匹配问题 (二次指派问题) 提出的, 而 *Sinkhorn*

算法 3-3 基于排列损失的图内/图间相似度图匹配学习 (PIA/PCA/IPCA-GM)

输入: 数据集 \mathcal{D} 包含点对应关系的图对; 初始模型权重 \mathbf{W} ; 迭代次数 K ; 学习率 lr

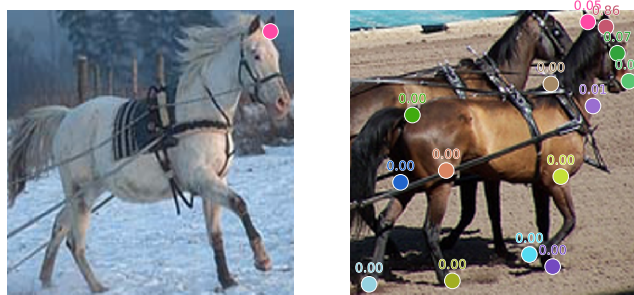
```

1: repeat
2:   # 从数据集中获取图以及真值标注, 下标  $s = 1, 2$  代表两个输入的图
3:    $\{(I_s, \{P_{si}\}_{i \in \mathcal{V}_s}, \mathbf{A}_s) | s = 1, 2\}, \mathbf{S}^{gt} \in \mathcal{D}$ ;
4:   # 提取 CNN 特征, 见公式 (3.5)
5:    $\{h_{si}^{(0)}\} \leftarrow \text{Interp}(\{P_{si}\}, \text{CNN}(I_s))$ ;
6:   if PIA-GM then
7:     # 图内特征更新, 见公式 (3.6a, 3.6b, 3.6c)
8:     for  $k \leftarrow \{1, 2, 3\}$  do
9:        $\{\mathbf{h}_{si}^{(k)}\} \leftarrow \text{GConv}_k(\mathbf{A}_s, \{\mathbf{h}_{si}^{(k-1)}\})$ ;
10:    end for
11:  end if
12:  if PCA-GM then
13:    # 第一层图内特征更新, 见公式 (3.6a, 3.6b, 3.6c)
14:     $\{\mathbf{h}_{si}^{(1)}\} \leftarrow \text{GConv}_1(\mathbf{A}_s, \{\mathbf{h}_{si}^{(0)}\})$ ;
15:    # 跨图特征更新, 见算法3-1
16:     $\{\mathbf{h}_{1i}^{(2)}, \mathbf{h}_{2j}^{(2)}\} \leftarrow \text{CrossEmb}(\{\mathbf{h}_{1i}^{(1)}, \mathbf{h}_{2j}^{(1)}\})$ ;
17:    # 第二层图内特征更新, 见公式 (3.6a, 3.6b, 3.6c)
18:     $\{\mathbf{h}_{si}^{(3)}\} \leftarrow \text{GConv}_2(\mathbf{A}_s, \{\mathbf{h}_{si}^{(2)}\})$ ;
19:  end if
20:  if IPCA-GM then
21:    # 迭代跨图特征更新, 见算法3-2
22:     $\{\mathbf{h}_{1i}^{(3)}, \mathbf{h}_{2j}^{(3)}\} \leftarrow \text{IterCrossEmb}(\{\mathbf{h}_{1i}^{(0)}, \mathbf{h}_{2j}^{(0)}\}, K)$ ;
23:  end if
24:  # 预测匹配关系, 见公式 (3.12, 3.14)
25:  build  $\mathbf{M}^{(0)}$  from  $\{\mathbf{h}_{1i}^{(3)}\}, \{\mathbf{h}_{2j}^{(3)}\}$  by Eq. (3.12);
26:   $\mathbf{S} \leftarrow \text{Sinkhorn}(\mathbf{M}^{(0)})$ ;
27:  # 基于公式 (3.15) 的端到端训练
28:   $\mathbf{W} \leftarrow -lr \times \frac{\partial L_{perm}(\mathbf{S}, \mathbf{X})}{\partial \mathbf{W}} + \mathbf{W}$ ;
29: until convergence
输出: 学习得到的模型参数  $\mathbf{W}$ .

```

网络则是为了求解线性指派问题。节点嵌入网络将图匹配的二次指派问题转化为一个能够借助 Sinkhorn 算法高效求解的线性指派问题。

偏移损失与排列损失。GMN^[16]所采用的损失函数是一个在原文中被称为“displacement loss”的偏移损失函数。该损失函数接受所有待匹配节点之间的匹配关系作为权重, 加权计算一个从源图片到目标图片的偏移向量。在训练过程中, GMN 试图最小化预测的偏移向量与实际偏移向量之间的误差 (即欧式距离)。作为对比, 借助 Sinkhorn 网络, 本节采用了一个面向组合优化的排列函数, 其中损失值被计算为预测的排列矩阵与真值排列矩阵之间的交叉熵。该排列损失函数直接利用了真值排列结果作为监督信息, 进而将该信息应用于端到端的训练中。



$$L_{perm} = 5.139, \quad L_{off} = 0.070$$

图 3-3 偏移损失的失败案例 (L_{perm} 排列损失, L_{off} 偏移损失)。

Figure 3-3 Failure case of the displacement loss.

图 3-3 给出了一个偏移损失函数的失败案例。源图片 (左图) 和目标图片 (右图) 中标注了所有候选的匹配, 右图中的数字代表模型预测出匹配的可能性。两个真实的匹配点均用粉色标出 (在这次失败的预测中, 只获得了 0.05 的可能性)。偏移损失由所有候选匹配加权得到, 计算得到了一个具有误导性的低损失 0.070。本节提出的排列损失, 与之相比, 得到了一个更加合理的高损失 5.139。在这个例子中, 偏移损失函数呈现了一个它不应有的低值, 但排列损失给出了准确的信息。实验结果同时显示, 使用排列损失训练出的模型性能显著地超越了由偏移损失训练得到的模型。

3.2.4 实验分析

测评标准。 实验通过两个图之间的匹配精度评价算法的表现。在测评阶段, 作者对 \mathcal{G}^1 与 \mathcal{G}^2 的关键点进行了过滤: 如果保证 \mathcal{G}^1 与 \mathcal{G}^2 的节点一一对应, 则称为交集过滤 (intersection filtering), 即无外点的实验设定; 如果有节点在 \mathcal{G}^2 中出现、但不出现在 \mathcal{G}^1 , 则称为子集过滤 (inclusive filtering), 即单边有外点的实验设定。上述设定可通过过滤数据集中的外点进行控制。参与测试的所有模型预测输入的两个图结构之间节点的对应关系, 这种对应关系由一个排列矩阵表示。模型的匹配精度由预测得到的排列矩阵与真值排列矩阵计算得到。

匹配精度的计算方法: 正确匹配的关键点对数目除以真值的关键点对总数。对于一个预测得到的排列矩阵 $\mathbf{X} \in \{0, 1\}^{n_1 \times n_2} = \text{Hungarian}(\mathbf{S})$ 以及真值排列矩阵 $\mathbf{X}^{gt} \in \{0, 1\}^{n_1 \times n_2}$, 假设 $n_1 \leq n_2$, 匹配精度可由如下公式计算:

$$\text{acc} = \frac{1}{n_1} \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} \mathbf{X}_{i,j} \mathbf{X}_{i,j}^{gt}. \quad (3.17)$$

对比方法。 在实验中, 以下算法参与了评测和对比。

GMN. 图匹配神经网络 (Graph Matching Network, GMN) 是深度图匹配学习领域的先驱工作^[16]。GMN 采用了 VGG16^[128] 提取图像中的特征, 其中构建了图

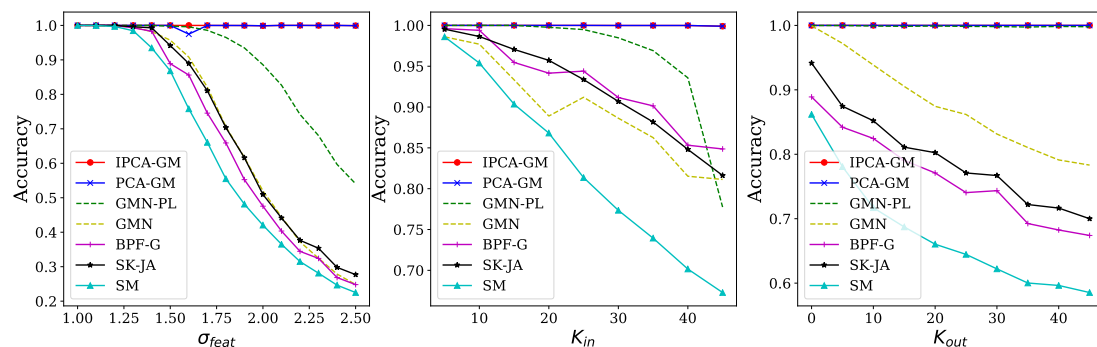


图 3-4 不同程度噪声、内点数目、外点数目的仿真实验结果。

Figure 3-4 Synthetic test with different levels of noises, inlier numbers and outlier numbers.


















匹配相似度矩阵的一阶特征与二阶特征分别来自于 VGG16 网络的浅层 (relu4_2) 与深层 (relu5_1)。GMN 通过一个不可学习的谱匹配 (spectral matching, SM) 求解器处理图匹配的相似度。该模型与输入图片中的物体类别无关, 因而可以为所有类别的物体学习一个统一的匹配模型。对于输入的两个图结构, 其中一个图 (源图, source graph) 由德罗内三角剖分 (Delaunay triangulation) 构建, 另一个图 (目标图, target graph) 则具有全连接 (fully-connected) 的拓扑结构。GMN 是图匹配领域提出的第一个端到端的深度学习模型。需要注意的是, GMN 在训练过程中采用了公式 (3.16b) 中的偏移损失作为损失函数。由于作者没有公开源代码, 本文使用 PyTorch 框架复现了 GMN 的代码。

HARG-SSVM. 由文献^[101]提出的基于结构化支持向量机 (structured support vector machine, SSVM) 的图匹配学习算法, 在实验中作为基于传统机器学习 (而非深度学习) 的图匹配学习算法的性能基准。*HARG-SSVM* 模型与输入的物体类型有关, 它为每个特定的输入类别学习了单独的图模型。本文在征得原作者的允许后使用了他们公开的代码。文献^[101]中的实验设定假定待匹配的关键点未知, 待匹配的关键点由 Hessian 检测法^[136]得到。在本节的实验设定中, 所有待匹配的关键点都是对模型已知的。本节的实验简单地调整了作者公开的源代码, 以适应新的实验条件: 对于所有由 Hessian 检测器得到的候选点, 选择真值对应点的最近邻作为输入的待匹配节点。在作者的原始代码中, 这一做法在训练 *HARG-SSVM* 的过程中就已经被采用。该算法中, 图结构由被称为 *HARG* 的手工边特征算子构建。

PIA/PCA/IPCA-GM. 为了与 GMN^[16]公平地对比, *PIA/PCA/IPCA-GM* 模型同样采用了 VGG16^[128]作为主干 CNN 网络, 从 relu4_2 和 relu5_1 提取图像的特征向量。从 CNN 网络层中提出的两个特征向量被拼接 (concatenate) 在一起组成新的特征向量, 进而能够同时刻画图像中局部的结构信息以及全局的语义信息。在 *PIA-GM* 中, 相似度由三层的 (图内) GNN 网络构建; 在 *PCA-GM* 中, 该模块由一层 (图内) GNN、一层跨图 GNN 以及一层 (图内) GNN 网络组成; 在 *IPCA-GM*

表 3-1 *Pascal VOC* 关键点数据集无外点（上半部分）和单边有外点（下半部分）情况下的匹配精度（%）。

Table 3-1 *Matching accuracy (%) on Pascal VOC Keypoint, without outliers (upper half) and with outliers in one graph (lower half).*

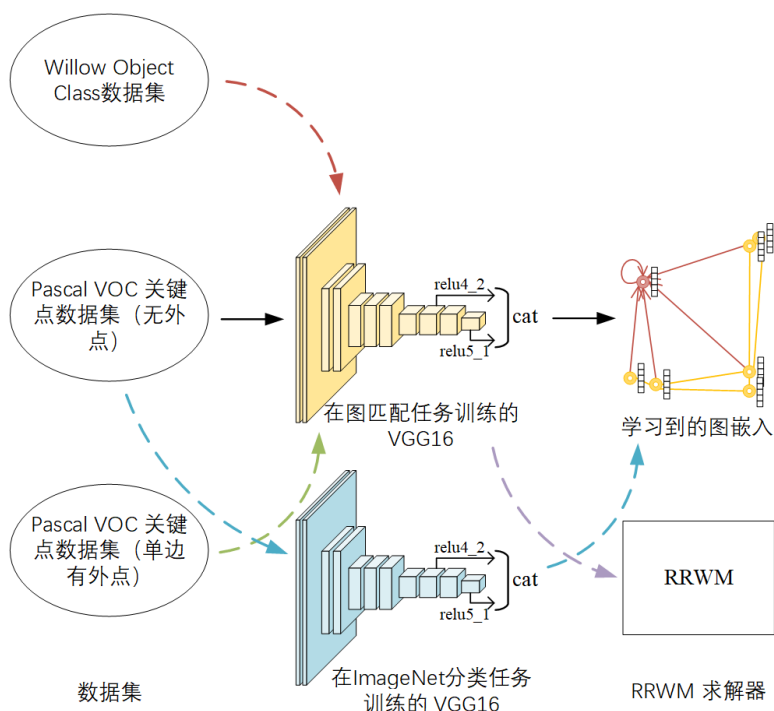
图匹配方法																		平均			
GMN ^[16]	31.9	47.2	51.9	40.8	68.7	72.2	53.6	52.8	34.6	48.6	72.3	47.7	54.8	51.0	38.6	75.1	49.5	45.0	83.0	86.3	55.3
GMN-PL	31.1	46.2	58.2	45.9	70.6	76.4	61.2	61.7	35.5	53.7	58.9	57.5	56.9	49.3	34.1	77.5	57.1	53.6	83.2	88.6	57.9
PIA-GM	41.5	55.8	60.9	51.9	75.0	75.8	59.6	65.2	33.3	65.9	62.8	62.7	67.7	62.1	42.9	80.2	64.3	59.5	82.7	90.1	63.0
PCA-GM	51.2	61.3	61.6	58.4	78.8	73.9	68.5	71.1	40.1	63.3	45.1	64.4	66.4	62.2	45.1	79.1	68.4	60.0	80.3	91.9	64.6
IPCA-GM	51.0	64.9	68.4	60.5	80.2	74.7	71.0	73.5	42.2	68.5	48.9	69.3	67.6	64.8	48.6	84.2	69.8	62.0	79.3	89.3	66.9
GMN ^[16]	34.2	55.0	46.4	39.6	77.0	60.5	46.9	54.5	31.7	51.0	48.0	48.0	48.5	50.8	28.8	73.8	49.8	38.3	69.4	83.9	51.8
GMN-PL	38.9	56.1	47.9	41.0	79.1	66.5	49.0	57.9	33.7	54.4	43.7	49.5	53.5	55.4	31.2	76.6	53.0	37.8	71.3	86.4	54.1
PIA-GM	43.8	60.6	51.5	43.5	75.4	70.6	58.9	62.0	35.3	54.4	44.3	57.1	56.1	58.6	40.0	76.5	60.1	36.5	76.1	86.3	57.4
PCA-GM	44.6	63.6	53.7	45.9	78.0	69.5	52.7	63.1	37.6	56.4	44.4	58.3	56.2	57.5	39.0	80.1	59.6	40.2	69.4	87.1	57.8
IPCA-GM	44.5	63.9	54.6	47.6	79.9	69.8	54.7	64.4	37.9	59.4	55.6	57.5	57.5	57.4	40.2	80.1	60.0	41.2	71.4	86.9	59.2

中，跨图 GNN 采用迭代形式。在经过图网络处理之后，PIA/PCA/IPCA-GM 均采用了公式 (3.12) 所示的相似度映射。每层 GNN 网络均拥有 2048 维的特征维度。公式 (3.15) 中的排列损失函数被用于端到端的监督训练。两个输入的图结构均使用德罗内三角剖分构建。基于经验，公式 (3.12) 的参数设定为 $\tau = 0.05$ (PIA-GM 和 PCA-GM) 以及 $\tau = 0.005$ (IPCA-GM)。上述模型均采用 PyTorch 实现。

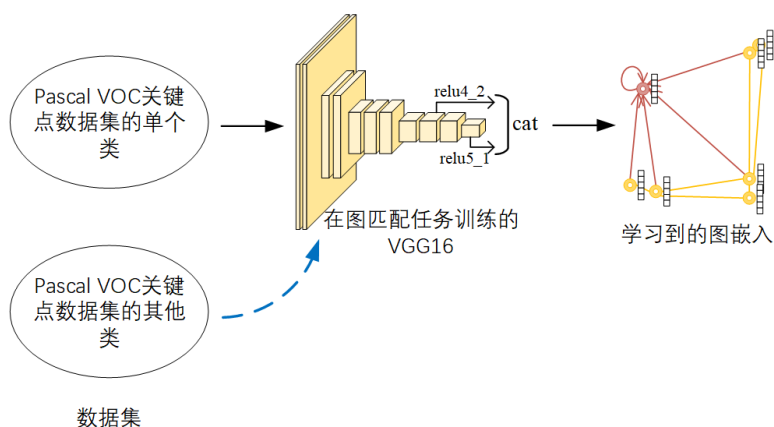
GMN-PL & PIA/PCA-GM-OL. GMN-PL 是 GMN^[16] 的变体，PIA/PCA-GM-OL 是 PIA/PCA-GM 的变体。其中，GMN-PL 将原始 GMN 模型中的偏移损失函数替换成了排列损失函数，同时保持其他设定不变。同样的，PIA/PCA-GM-OL 将排列损失函数替换成了偏移损失函数，同时维持其他模块不变。

真实图片数据集上的实验每次从数据集中随机选取两张图片，在图片中分别构建具有相同节点数目的图结构。在本节的实验设定中，图结构对模型是未知的，而不同的模型采取了不同的图结构构建策略（参见上文中的讨论）。CNN 网络的初始化参数由在 ImageNet^[130] 分类任务上预训练的模型参数得到。

仿真实验。 作为一种常用的图匹配评测方法，作者在仿真数据集上测试了深度图匹配方法的性能。作者仿照文献^[34]中采用的方法，生成了仿真实验所需的随机数据。首先，给定内点数目 K_{in} ，随机生成一个真值图，其中每个节点具有 1024 维随机分布于 $U(-1, 1)$ 的特征向量（512 维点特征，512 维边特征）。此外，每个节点具有一个在 2D 平面 $U(0, 256) \times U(0, 256)$ 上随机分布的坐标。根据外点数目 K_{out} ，在 2D 平面上随机添加外点以模仿干扰。在训练与测试阶段，从中采样带有加性高斯噪声的图样本，其中高斯噪声服从分布 $N(0, \sigma_{feat}^2)$ ，同时关键点坐标被随机的仿射变换和加性坐标噪声 $N(0, \sigma_{coo}^2)$ 所扰动。需要注意，在仿真实验中没有使用 CNN 提取图像特征，只对图相似度建模方法以及监督信息（即损失函数）进行了横向比较。在实验中，通过改变 K_{in} 、 K_{out} 、 σ_{feat} 三个参数，比较不同情况下深度图匹配算法以及不包含可学习参数的传统图匹配方法的精度（默认参数： $K_{in} = 20, K_{out} = 0, \sigma_{feat} = 1.5, \sigma_{coo} = 10$ ）。每一次随机实验随机生成



(a) 表3-2、表3-3、表3-4中的迁移性和泛化性实验。



(b) 图3-6、图3-7中的跨类别迁移学习实验。

图 3-5 迁移学习实验说明。

Figure 3-5 Illustration of transfer learning experiments.

了 10 个不同的图结构，并计算了它们的平均匹配精度。图3-4中的实验结果表明，相比于其他方法，PCA/IPCA-GM 对噪声和复杂的图结构具有更强的鲁棒性。

Pascal VOC 关键点数据集。本文还在带有 Berkeley 额外关键点标注^[137]的 Pascal VOC 数据集^[138]上衡量了不同算法的表现。该数据集包含有 20 类不同的物体，及其对应的关键点位置标注。模仿其他工作的数据预处理步骤^[16]，作者将原始数据集过滤，得到 7,020 张训练图片以及 1,682 张测试图片组成的数据集。在被作为网络输入之前，所有物体都被沿着它们的边界框 (bounding box) 裁切，随后重新调整至 256×256 的大小。Pascal VOC 关键点数据集被认为是一个较困难的语义关键点匹配数据集，因为数据集中的物体可能会在大小、姿态、光照等

表 3-2 在 Pascal VOC 关键点数据集上对抗外点的迁移学习匹配精度 (%)。

Table 3-2 Matching accuracy (%) for transfer learning test in the presence of outliers on Pascal VOC Keypoint.

图匹配方法	从无外点迁移	在单边外点微调	在单边外点训练
GMN ^[16]	51.7	51.8	51.3
GMN-PL	53.2	54.1	50.2
PIA-GM	55.2	57.4	–
PCA-GM	56.0	57.8	57.6
IPCA-GM	54.1	59.2	–

表 3-3 Willow Object Class 数据集的匹配精度 (%)。

Table 3-3 Accuracy (%) on Willow Object Class.

图匹配方法	face	mbike	car	duck	wbottle
HARG-SSVM ^[101]	91.2	44.4	58.4	55.2	66.6
GMN-VOC ^[16]	98.1	65.0	72.9	74.3	70.5
PCA-GM-VOC	100.0	69.8	78.6	82.4	95.1
IPCA-GM-VOC	100.0	67.1	73.3	82.1	91.7
GMN-Willow ^[16]	99.3	71.4	74.3	82.8	76.7
PCA-GM-Willow	100.0	76.7	84.0	93.5	96.9
IPCA-GM-Willow	100.0	77.7	90.2	84.9	95.2

方面有很大差别。此外，数据集中内点的数目会在 6 至 23 之间变动。

Pascal VOC 关键点数据集^[137]的实验测试了模型分别在 20 个类别上的匹配精度。该数据集上的详细实验结果请见表3-1，包括了有外点和无外点两种实验条件下的结果。当排列损失函数替代偏移损失函数后，GMN-PL 几乎在所有类别上均超过了 GMN^[16]。考虑 20 个类的平均匹配精度，PIA-GM 的性能超越了现有的 GMN 方法，并且 PCA-GM 和 IPCA-GM 分别利用了单次前向计算的跨图嵌入和迭代的跨图嵌入方法，进一步提升了匹配精度。并且，PCA-GM 和 IPCA-GM 在对抗外点时展现了更强的鲁棒性（下文中将详细讨论）。配备 2 块 2080Ti 显卡，PCA-GM 在训练过程中的速度约为 18 个图像对每秒，IPCA-GM 的训练速度约为 16 个图像对每秒。上述结果证明了在真实图片匹配场景中，排列损失相比于偏移损失有优势、图嵌入 +Sinkhorn 的求解组合相比于固定的 SM 求解器^[35]有优势以及（迭代）跨图嵌入相比于普通图内嵌入方法有优势。

迁移学习实验。如图3-5总结了验证 PIA/PCA/IPCA-GM 模型的鲁棒性和泛化能力的实验。图3-5中，黑色箭头表示基础的模型和训练方式。作者研究了从无外点的训练数据迁移到同一数据集（不同样本）、单边有外点的测试数据（图5(a)的绿色箭头），从训练数据集迁移至另一个不同的测试数据集（图5(a)的红色箭头），从训练类别迁移到其他的类别进行测试（图5(b)），以及学习到的 CNN、图嵌入

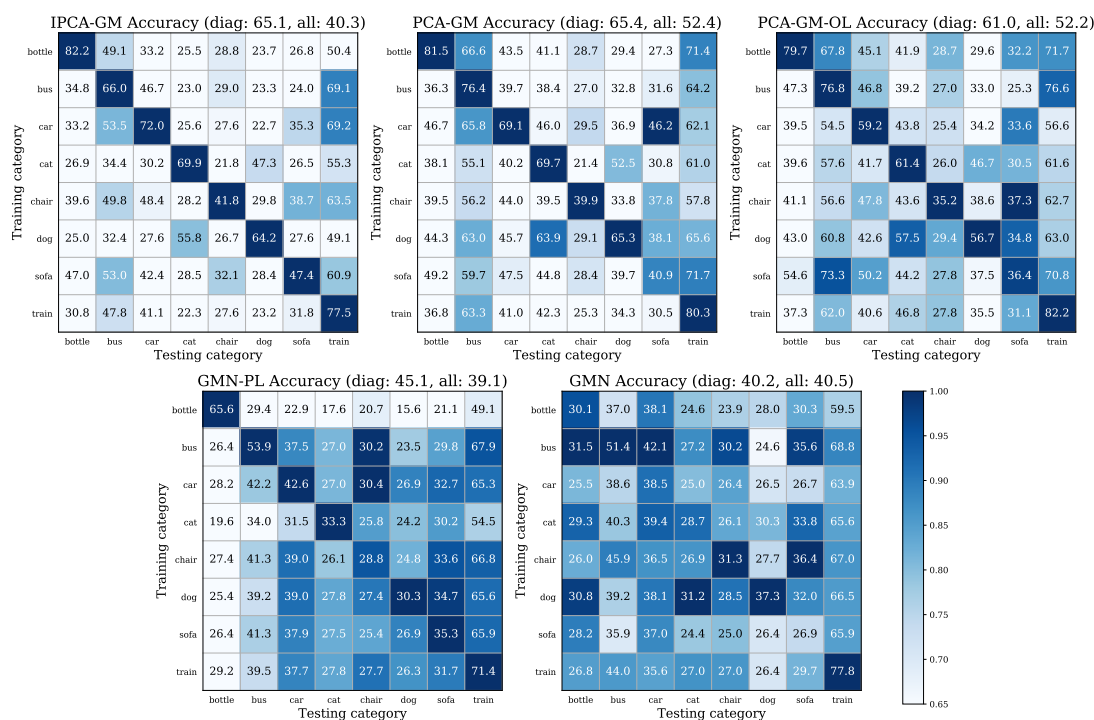


图 3-6 Pascal VOC 关键点数据集上 8 个类别混淆矩阵的精度结果。

Figure 3-6 Confusion matrix of accuracy of eight categories of objects from Pascal VOC Keypoint.

模块的可迁移性（图 5(a) 的蓝色和紫色箭头）。

对抗外点的迁移能力。如图 5(a) 中绿色箭头所示，作者首先在无外点的 Pascal VOC 关键点数据集上训练了图匹配模型，随后直接将该模型用于有外点的情况进行测试（一个图中单边有外点、一个图中没有）。实验结果如表 3-2 所示，“从无外点迁移”代表直接将无外点情况训练的模型迁移到有外点进行测试，“在有外点微调”代表在有外点的数据集上进行微调。最后一列“在有外点训练”代表在有外点的数据集上从零开始训练，“-”该模型在这个实验条件下没有收敛。直接在单边有外点的数据集上训练似乎对 IPCA-GM 比较有挑战，其中一个可能的原因是当外点存在时跨图传播权重矩阵不满足双随机性，即每列的加和不等 1，可能在训练的早期误导了迭代更新的跨图匹配模块。PIA-GM 可能因为较低模型容量无法在单边有外点的数据集上直接训练。作为对比，从无外点训练的模型向单边有外点进行迁移学习会带来更高的性能，表格 3-1 中的有外点匹配结果均是基于迁移学习得到的。

跨数据集迁移能力。本节在 Willow Object Class 数据集上测试了模型跨数据集进行知识迁移的能力。Willow Object Class 数据集是由文献^[101]的作者收集的真实图匹配数据集。该数据集包含了 5 类物体，每类物体至少包含了 40 张图片。数据集中的图片是从 Caltech-256 (face、duck 和 wine bottle) 数据集以及 Pascal VOC 2007 (car 和 motorbike) 数据集中收集得到。在实验中，图片首先被缩放到 256×256 ，之后再输入 CNN 网络提取特征。对该数据集而言，它的匹配难度应

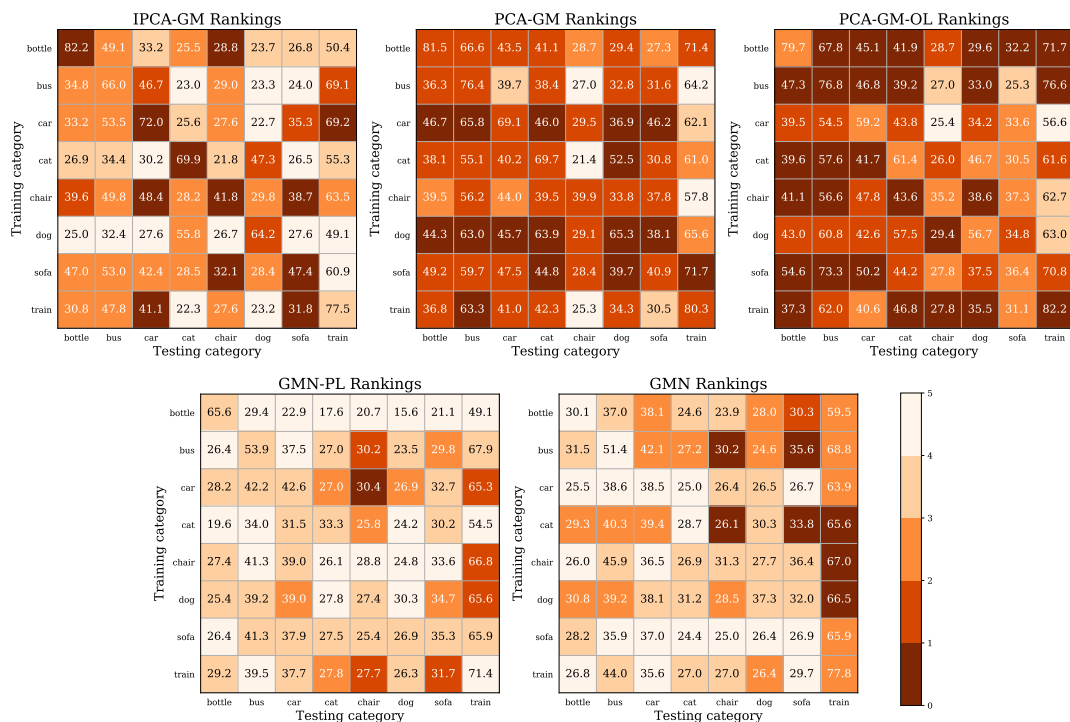


图 3-7 Pascal VOC 关键点数据集上 8 个类别混淆矩阵的排序结果。

Figure 3-7 Confusion matrix of rankings of eight categories of objects from Pascal VOC Keypoint.

当低于 Pascal VOC 关键点数据集，因为数据集作者将所有隶属于同一类的物体在姿态上对齐，并且该数据集中的图片相对缺乏尺寸、背景以及光照上的差异。

本文仿照文献^[101]构建了测评工具，基于原作者公开的 Matlab 源代码训练和测试了传统机器学习方法 HARG-SSVM。对于其他方法，按照如图 5(a) 中红色箭头所示的评测方法，测试了模型的跨数据集迁移学习能力，结果如表 3-3 所示。对于深度图匹配模型，它们的模型权重首先在剔除了重复类别的 Pascal VOC 关键点数据集上预训练。这些通过迁移得到的模型在表 3-3 中标注为 GMN-VOC、PCA-GM-VOC 和 IPCA-GM-VOC。上述模型可以进一步在 Willow Object Class 的训练集上微调，得到 GMN-Willow、PCA-GM-Willow 和 IPCA-GM-Willow。需要注意，传统机器学习方法 HARG-SSVM 对每个类都学习了一个模型，但深度学习学习方法只需一个模型即可处理所有类别。PCA-GM 和 IPCA-GM 展现出了强大的跨数据集迁移能力，并且在 Willow Object Class 数据集的所有类上均超越了其他方法。

跨类别迁移能力。为了测试模型在不同类别之间的迁移能力，作者在 8 个随意选取的 Pascal VOC 类别上测试了 IPCA-GM、PCA-GM、PCA-GM-OL、GMN-PL 和 GMN，并采用图 3-6、图 3-7 对结果进行了可视化。每个类别中训练集、测试集的分割方法与原数据集保持一致。图 3-6 代表了该结果的混淆矩阵 (confusion matrix)，其中 y 轴标签代表训练的类，x 轴标签代表测试的类。蓝色的深浅代表了每个类上的精度高低。图 3-7 代表了混淆矩阵的每一格在 5 种测试方法之间的

表 3-4 不同模型配置在 *Pascal VOC* 关键点数据集的匹配精度 (%)。Table 3-4 Matching accuracy (%) on *Pascal VOC* Keypoint with different module configurations.





















CNN	匹配求解																					平均
图匹配	图嵌入	51.2	61.3	61.6	58.4	78.8	73.9	68.5	71.1	40.1	63.3	45.1	64.4	66.4	62.2	45.1	79.1	68.4	60.0	80.3	91.9	64.6
图匹配	RRWM	44.0	61.4	53.5	51.7	72.1	67.7	64.5	59.6	41.0	55.8	53.9	58.6	60.0	57.2	41.3	78.8	61.8	45.2	74.2	88.1	59.5
ImgNet	图嵌入	36.5	46.9	51.9	43.0	42.7	52.0	53.4	57.7	33.5	47.5	44.9	54.7	54.6	48.1	28.0	61.8	57.2	39.9	72.6	58.3	49.3
ImgNet	RRWM	29.9	44.7	41.4	41.6	34.7	44.9	45.2	38.8	27.0	34.3	36.7	37.7	41.4	41.7	24.9	57.2	41.2	30.7	68.3	53.3	40.8

表 3-5 在 *Pascal VOC* 关键点数据集上，各个模块对模型精度的影响。Table 3-5 Ablation study on proposed components on *Pascal VOC* Keypoint.

图匹配	微调 CNN	图内嵌入	跨图嵌入	迭代跨图	距离度量	匹配精度
✓		✓	✓	✓	✓	66.9
✓		✓	✓	✓	×	66.3
✓		✓	✓	×	×	64.2
✓		✓	×	×	×	62.1
✓		×	×	×	×	54.8
×		×	×	×	×	41.9

性能排序。实验结果表明，本节提出的深度学习方法能够在相似的类（例如“猫”和“狗”）之间进行迁移。基于图嵌入的深度学习方法可以在混淆矩阵的非对角线元素上提供更高的迁移精度，同时排列损失函数可以在对角线元素上提供更强的监督信号。IPCA-GM 在对角线元素上的精度更加突出，一种可能的解释是 IPCA-GM 具有更高的模型容量，因此会更大程度地拟合到训练类上。

CNN 和图嵌入模型的泛化能力。本节讨论的深度图匹配模型可以同时学习图像特征（通过 CNN 网络）以及匹配求解（通过图嵌入和 Sinkhorn）。此处探索了 CNN 模型与匹配求解模块能否与其他现有技术相互替换、以及上述两个模块在多大程度上相互耦合。表 3-4 所示的实验将 PCA-GM 模型分为两部分：一个 CNN 主干网络（在“图匹配”任务上训练）、一个基于“图嵌入”的图匹配求解模块。上述两个模块可以和其他现有技术进行组合，包括现有的图匹配求解器 RRWM^[34]以及在 ImageNet 图像分类任务上预训练的普通 CNN 网络。在图 5(a)中，蓝色箭头代表了“ImgNet+ 图嵌入”的组合，紫色箭头代表了“图匹配+RRWM”的组合。实验结果表明，在图匹配任务上学习得到的 CNN 网络在迁移到 *bike, chair, table* 等类别后依然有较强的匹配精度。作为对比，基于图嵌入的匹配求解模块似乎和 CNN 存在更高程度地耦合。不过，“ImgNet+ 图嵌入”的组合依然在所有类上超越了“ImgNet+RRWM”，在部分类上与“图匹配+RRWM”性能相当。因此，学习到的图嵌入匹配求解模块在迁移到其他 CNN 网络时，依然具有求解图匹配的能力。

消融实验 (ablation study)。作者研究了 IPCA-GM 网络中采用的各个模块对模型性能的影响。在端到端的模型训练中引入/不引入各个模块，实验结果如表格 3-5 所示。“图匹配微调 CNN”代表在图匹配任务上训练 CNN，不然则采用

ImageNet 数据集上预训练的权重。实验结果显示, IPCA-GM 模型中各个模块均对匹配精度起到正向作用。在实验中, GNN 网络被随机初始化, 相似度映射关系的权重矩阵则使用单位矩阵加上随机噪声进行初始化。

3.3 面向通用二次指派求解的神经网络设计

Lawler 形式的二次指派问题 (Lawler's QAP) 作为一类常见的排列型组合优化问题, 覆盖了图匹配、布局规划、电路设计等重要的应用场景。从图匹配出发、面向更一般的二次指派问题, 本节提出了一种纯神经网络直接求解 Lawler's QAP 的方法, 其中 QAP 问题的相似度矩阵 (及其等价的伴随图) 可以直接输入神经网络, 随后 QAP 的求解被转化为在伴随图上的顶点分类问题。伴随图通过一个图嵌入网络学习如何将顶点分类, 与随后的 Sinkhorn 算法和排列损失函数一起构成了端到端的学习框架。为了进一步扩展模型的适用范围, 本节引入了具有排列意识 (permutation-aware) 的图嵌入方法, 以及引入虚拟节点以处理两边节点数目不一致的情况。这是首次将神经网络直接用于求解最一般的 Lawler 形式二次指派问题。本节的求解网络非常灵活, 可以扩展到超图匹配、多图匹配等场景。在图匹配仿真数据以及 QAPLIB 基准数据集⁵上开展的纯 QAP 优化实验表明, QAP 的神经网络求解器与目前最优秀的传统图匹配和 QAP 求解器性能相当、甚至更优, 同时时间开销显著降低。

3.3.1 二次指派问题描述

二次指派问题的数学形式已在3.2.1小节中和图匹配问题一起进行了讨论。作为图匹配问题更一般的数学形式, 二次指派问题在计算机视觉、机器学习等领域具有广泛的应用场景。实际上, 二次指派问题起源于运筹领域对设施布局的研究: 若要将 n 个设施布局在 n 个候选位置, 已知任意两个位置之间的距离和任意两个设施之间的流量 (人流量或物流量), 目标是找到一个排列矩阵, 使得总的流量 \times 距离最小。上述目标可以采用类似公式 (3.2) 中 Koopmans-Beckmann 形式二次指派问题的描述:

$$\begin{aligned} \max J(\mathbf{X}) &= \text{tr}(\mathbf{X}^\top \mathbf{F}^1 \mathbf{X} \mathbf{F}^2), \\ \text{s.t. } \mathbf{X} &\in \{0, 1\}^{n \times n}, \quad \mathbf{X} \mathbf{1}_n = \mathbf{1}_n, \quad \mathbf{X}^\top \mathbf{1}_n = \mathbf{1}_n. \end{aligned} \quad (3.18)$$

其中 $\mathbf{F}^1 \in \mathbb{R}^{n \times n}$, $\mathbf{F}^2 \in \mathbb{R}^{n \times n}$ 分别代表候选位置的距离和设施之间的流量。Lawler 形式的二次指派问题更具一般性, 不仅可以覆盖设施布局等运筹问题, 还可以处理图匹配等问题。通过 $\mathbf{K} = \mathbf{F}^2 \otimes \mathbf{F}^1$, 可以将 Koopmans-Beckmann 形式转化为

⁵<https://coral.ise.lehigh.edu/data-sets/qaplib/>

表 3-6 现有研究工作在两种 QAP 形式下的总结。

Table 3-6 Summary of existing methods on two popular formulations of QAP.

	Koopmans-Beckmann's QAP	Lawler's QAP
非学习方法	[27,139-141]	[28,34-35,142-143]
机器学习方法	[144]、3.2节	3.3节

Lawler 形式：

$$\begin{aligned} \max J(\mathbf{X}) &= \text{vec}(\mathbf{X})^\top \mathbf{K} \text{vec}(\mathbf{X}), \\ \text{s.t. } \mathbf{X} &\in \{0,1\}^{n \times n}, \quad \mathbf{X} \mathbf{1}_n = \mathbf{1}_n, \quad \mathbf{X}^\top \mathbf{1}_n = \mathbf{1}_n. \end{aligned} \quad (3.19)$$

本节关注的是利用机器学习求解最一般的 Lawler 形式的二次指派问题，其中相似度矩阵 \mathbf{K} 是问题输入。为了方便描述，上述公式假设了问题的可行域是一个全排列（即 $n_1 = n_2 = n$ ），本节探讨的 QAP 求解方法也可扩展到部分排列的场景。

3.3.2 研究动机

3.2节所描述的神经网络求解方法专为图匹配问题设计。面向更一般的 Lawler 形式二次指派问题，本节提出了更一般的神经网络结构进行求解。如表格3-6所总结的，在机器学习发展之前，传统的求解算法在两种形式的 QAP 问题中均有广泛的研究。最近的研究工作^[144]开始注意到了机器学习方法用于求解 Koopmans-Beckmann's QAP 的潜力。本文3.2节的 PCA/IPCA-GM 方法需要输入两个待匹配图结构的邻接矩阵，可以看作特殊形式的 Koopmans-Beckmann's QAP。本节首次针对最一般的 Lawler's QAP 问题的机器学习求解开展研究，并探索了相关方法在超图匹配、多图匹配的扩展。面向二次指派问题，表格3-7从多个角度归纳了本节的研究内容和前序工作（KB's QAP 是 Koopmans-Beckmann's QAP 的简写），包括学习求解的问题类型、多图匹配的处理方法、GNN 模块、GNN 学习的图类型、相似度函数以及损失函数的设计。可以看到，本节研究内容的应用范围与技术细节和前序工作存在显著差异。

本章节的研究动机来自如下几个方面：

1) **设计一个神经网络模型直接处理最一般的 Lawler 形式二次指派问题。** 作为一个二次指派问题的神经网络求解器，神经网络允许 Lawler's QAP 中的相似度矩阵 \mathbf{K} 直接作为输入。这是通过将相似度矩阵视为一个等价的伴随图 (association graph)，其中伴随图的节点可以通过图神经网络^[145]进行分类。作为对比，现有的二次指派（更准确地说是图匹配）学习方法^[16,101,146]需要单个图的节点特征和边特征作为神经网络输入，无法直接处理相似度矩阵。直接处理相似度矩阵是适用范围最广的情况，因为相似度矩阵可以基于节点特征和边特征构造。

表 3-7 机器学习图匹配和二次指派方法的技术总结。

Table 3-7 Summary of existing literature in learning graph matching and QAP.

方法	学习求解类型	多图匹配	GNN	学习的图	相似度函数	损失函数
Nowak 等 ^[144]	KB's QAP 的特殊情况	无	图卷积	独立的图	内积	多分类交叉熵
GMN ^[16]	无	无	无	带权指数函数	偏移损失	
Zhang 等 ^[146]	无	无	消息传递网络 CMPNN	独立的图	内积	多分类交叉熵
PCA/IPCA-GM(3.2节)	KB's QAP 的特殊情况	无	图卷积 + 跨图卷积	独立的图	带权指数函数	排列损失
CIE-H ^[77]	KB's QAP 的特殊情况	无	边嵌入 + 跨图卷积	独立的图	带权指数函数	带匈牙利注意力的排列损失
LCS ^[103]	Lawler's QAP 的特殊情况	无	图卷积	伴随图	全连接网络	排列损失
BBGM ^[82]	无	非学习的求解器 ^[93]	样条卷积	独立的图	带权内积	汉明距离
NGM (本节)	最一般的 Lawler's QAP	无	样条 + 排列图卷积	独立的图 + 伴随图	带权内积	排列损失
NHGM (本节)	高阶指派问题	无	样条 + 排列超图卷积	独立的图 + 伴随超图	带权内积	排列损失
NMGM (本节)	最一般的 Lawler's QAP	端到端的谱方法	样条 + 排列图卷积	独立的图 + 伴随图	带权内积	排列损失

2) **神经网络同时支持监督学习和非监督学习。** 图像等领域的匹配问题通常具有节点级别的标注信息，可以采用监督学习方法进行端到端的训练。对于优化领域的问题（例如 QAPLIB 中的问题），其目标函数就是监督信号，因此可以进行无标注数据的非监督训练。

3) **将神经网络求解器扩展到高阶（三阶）指派。** 二次指派问题建模了图结构中的二阶相似度。更进一步，超图（存在同时连接多个节点的超边）中存在更高阶的结构，因此超图匹配问题需要建模更高阶的相似度信息。对于超图匹配问题，本节仿照伴随图的形式构造了伴随超图（association hypergraph）。这是首次利用深度学习处理超图匹配，特别是将超边的相似度信息进行显式建模。

4) **将神经网络求解器扩展到多图匹配。** 将端到端的谱方法^[127]作为后处理步骤，可在二次指派求解器的基础上学习多图匹配。此外，本节开发了额外技术以稳定学习过程。这是首次在端到端的神经网络中学习多图匹配。

3.3.3 算法描述

本节将首先介绍神经图匹配网络（Neural Graph Matching, NGM），它具有直接求解 Lawler 形二次指派问题的能力。其次介绍神经超图匹配网络（Neural Hyper Graph Matching, NHGM），将神经网络求解器的应用范围扩展到了超图匹配领域。最后，介绍神经多图匹配网络（Neural Multi-Graph Matching, NMGM），将神经网络求解器的应用范围扩展到了多图匹配。图3-8总结了这些算法。

背景知识。 图匹配神经网络 NGM 实际求解了如下优化问题：对公式 (3.1)

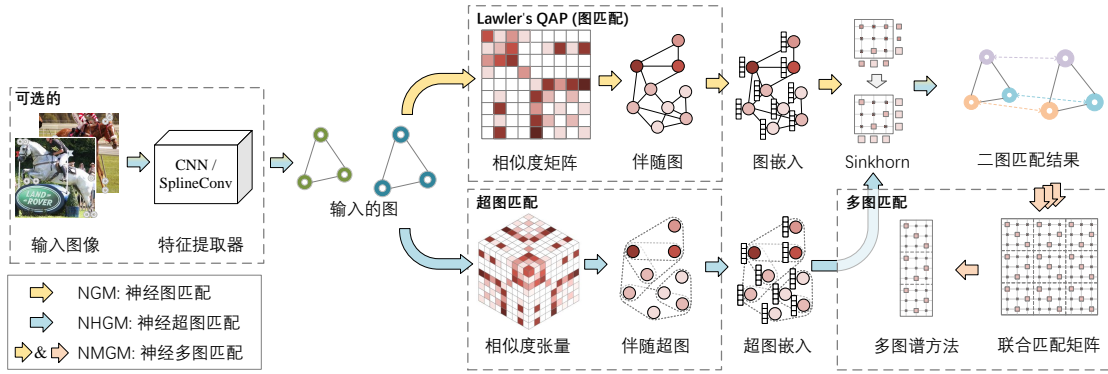


图 3-8 神经网络 QAP 求解器（神经图匹配算法，NGM）概览。

Figure 3-8 Overview of our neural QAP solver named neural graph matching (NGM).

中的 Lawler's QAP，通过常见的双随机约束进行放松，

$$\begin{aligned} \max J(\mathbf{S}) &= \text{vec}(\mathbf{S})^\top \mathbf{K} \text{vec}(\mathbf{S}), \\ \text{s.t. } \mathbf{S} &\in [0, 1]^{n_1 \times n_2}, \quad \mathbf{S} \mathbf{1}_{n_2} = \mathbf{1}_{n_1}, \quad \mathbf{S}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}. \end{aligned} \quad (3.20)$$

其中 \mathbf{S} 是（部分）双随机矩阵，其所有行的总和为 1，所有列的总和 ≤ 1 。双随机矩阵是排列矩阵在连续域的松弛形式，为保证梯度存在，神经网络的输出必须是连续的。图匹配问题中，对于相似矩阵 $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ ，对角元素 $\mathbf{K}_{ia,ia}$ 代表了 \mathcal{G}^1 的节点 i 和 \mathcal{G}^2 的节点 a 的一阶相似度，而非对角线元素 $\mathbf{K}_{ia,jb}$ 代表了 \mathcal{G}^1 的边 ij 和 \mathcal{G}^2 的边 ab 的二阶相似度。其中， ia 实际上代表了 \mathbf{K} 的第 $i \cdot n_2 + a$ 行、 jb 实际上代表了 \mathbf{K} 的第 $j \cdot n_2 + b$ 列。

如图3-9所示，通过所谓的伴随图 $\mathcal{G}^A = (\mathcal{V}^A, \mathcal{E}^A)$ [34-35]，可以从一个全新的角度来看待图匹配和二次指派问题。为了避免图和伴随图之间的歧义，作者将图中的点称为“节点”，将伴随图中的点称为“顶点”。读者在阅读本节时应注意区分这两个概念，因为它们将反复出现。图3-9(b)展示了分别具有相似性矩阵 \mathbf{K} 和相似性张量 \mathbf{H} 的图匹配和超图匹配问题，与 (a) 伴随图和 (c) 伴随超图相对应。图3-9(b)中的节点到节点匹配问题可以被表示为在伴随图上的顶点分类任务，伴随图的边权可以由相似性矩阵表示。这一视角在过去的文献中被广泛采用，例如图匹配的文献 [34] 和超图匹配的文献 [123]。图3-9(d)展示了一个示例，从独立的图到最终匹配结果：相似性矩阵 \mathbf{K} 是从独立的图构建的，匹配问题等价于在伴随图上进行顶点分类。具有排列意识的图卷积和顶点分类器被用于伴随图上生成顶点得分，接着进行矩阵变形和 Sinkhorn 归一化以获得双随机矩阵。

伴随图的顶点 $\mathcal{V}^A = \mathcal{V}^1 \times \mathcal{V}^2$ 编码了与排列矩阵 $\mathbf{X}_{i,a}$ 相对应的节点匹配候选，因此向量化的匹配矩阵 $\text{vec}(\mathbf{X})$ 等价于伴随图的顶点集。边 \mathcal{E}_A 表示两对对应关系之间的一致性， $(ia, jb) \in \mathcal{E}_A$ 由 $\mathbf{K}_{ia,jb}$ 建模，相似矩阵 \mathbf{K} 的非对角部分等价于伴随图的邻接矩阵。因此，二次指派问题可以转化为伴随图上的顶点分类问题，这也是文献 [34-35] 的做法。在本文中，为了更好地利用一阶相似性，对角线元素

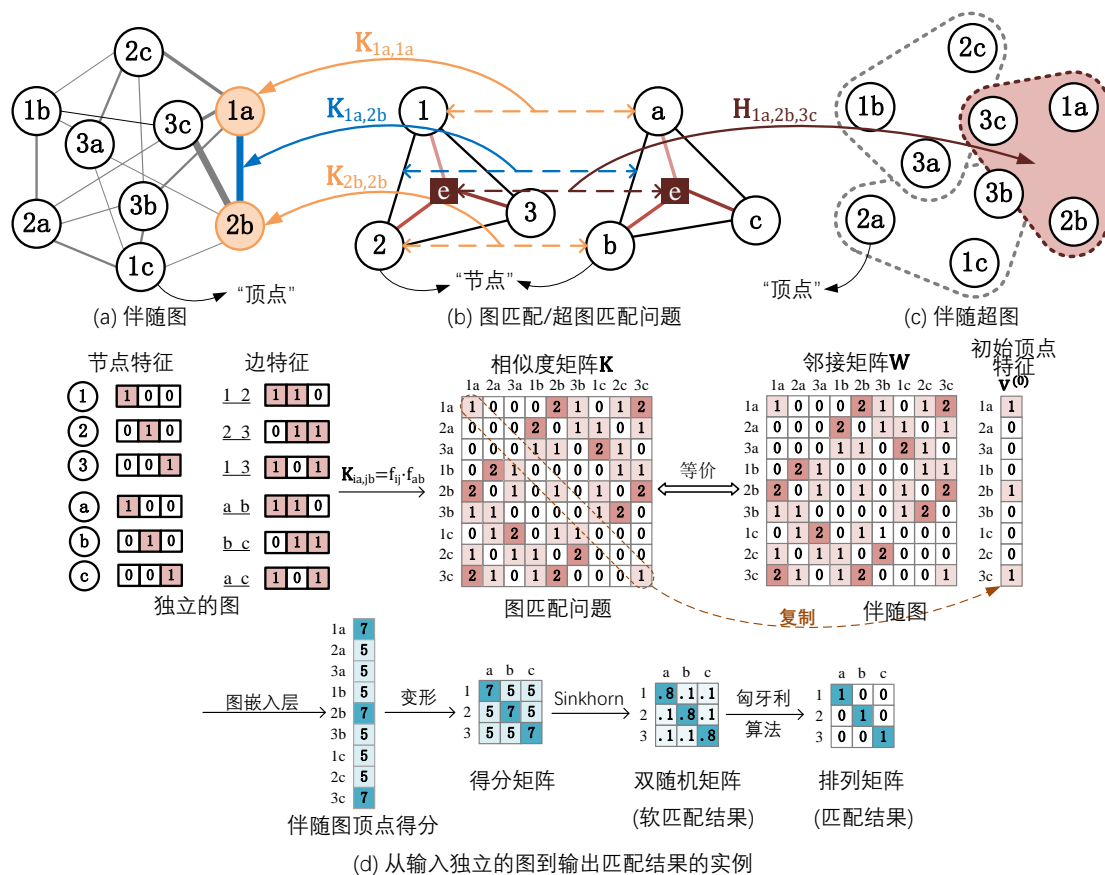


图 3-9 机器学习 QAP/超图匹配求解器的形式化及完整流程介绍。
 Figure 3-9 Formulation of machine learning QAP/hyper graph matching solver and an illustration of the overall pipeline.

$K_{ia,ia}$ 被视为顶点属性 V^A 。这种公式也可以推广到超图匹配问题，将边替换为超边，如图 3-9(c) 所示。伴随图中不会出现违反排列约束的边（例如在图 3-9(a) 中，顶点“1a”和“1c”之间没有边）。

神经图匹配网络 NGM。 面向上一节介绍的问题形式和伴随图，基于图卷积网络（Graph Convolutional Networks, GCN）^[133] 及具有排列意识的图嵌入模块，即可通过顶点分类求解松弛后的 Lawler's QAP 问题，即公式 (3.20)。顶点分类在与相似度矩阵等价的伴随图上进行，并接着采用 Sinkhorn 算法以添加排列约束。根据现有的传统图匹配研究^[34-35]，图匹配问题等价于在伴随图上进行顶点分类。NGM 可以接受原始图像（通过联合学习的 CNN 和相似度函数）或相似度矩阵（无 CNN 及相似度函数）作为输入，并可端到端地从真值标签学习或通过纯优化步骤学习求解 QAPLIB 的问题。

从真实图像中构造相似度矩阵。NGM 允许输入相似度矩阵或原始图像。此处介绍的图像处理模块是可选的，被视为用于处理图像的特殊插件。如图 3-11 的上半部分所示，图像特征通过可学习的 CNN 层（例如 VGG16^[128]）提取。与 3.2 节类似，给定带有关键点标记的两个输入图像，采用 CNN 层提取 VGG16 的 relu4_2 和 relu5_1 特征，拼接得到每个节点的特征 $\bar{U}^1 \in \mathbb{R}^{n_1 \times d}$ 和 $\bar{U}^2 \in \mathbb{R}^{n_2 \times d}$ ，其中 d 是

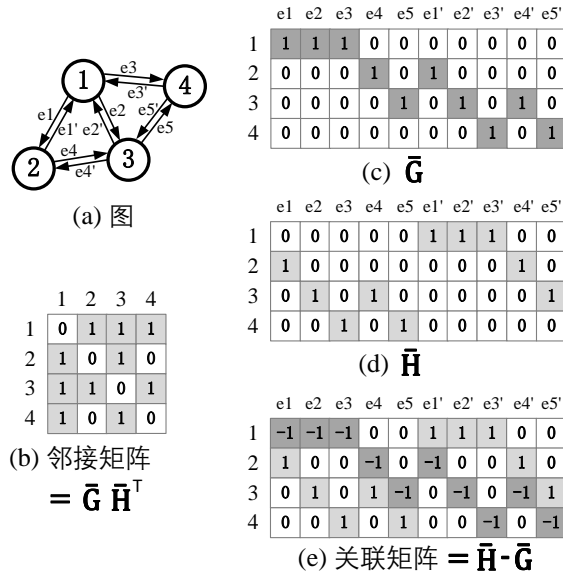


图 3-10 矩阵 $\bar{\mathbf{G}}, \bar{\mathbf{H}}$ 和其他常用图表示方法的联系。

Figure 3-10 The connection between matrices $\bar{\mathbf{G}}, \bar{\mathbf{H}}$ and other common representations of graphs.

特征维度。然后在 $\bar{\mathbf{U}}^1, \bar{\mathbf{U}}^2$ 上采用两次样条卷积 (SplineConv)^{6[147]} 层产生精细化的特征 $\bar{\mathbf{F}}^1 \in \mathbb{R}^{n_1 \times d}$, $\bar{\mathbf{F}}^2 \in \mathbb{R}^{n_2 \times d}$ 。SplineConv^[147] 是一种强大的图卷积操作符，利用 B-Spline 核将几何特征编码为节点特征。因此，SplineConv 适用于图像匹配数据集上的特征精炼。有关 SplineConv 的详细信息，请参阅原始论文。边特征由特征空间中两个节点之差得到：

$$\bar{\mathbf{X}} = \bar{\mathbf{G}}^{1T} \bar{\mathbf{F}}^1 - \bar{\mathbf{H}}^{1T} \bar{\mathbf{F}}^1, \bar{\mathbf{Y}} = \bar{\mathbf{G}}^{2T} \bar{\mathbf{F}}^2 - \bar{\mathbf{H}}^{2T} \bar{\mathbf{F}}^2, \quad (3.21)$$

其中 $\bar{\mathbf{G}}, \bar{\mathbf{H}}$ 的定义如图 3-10 所示：两个图的连通性由 $\bar{\mathbf{G}}^1, \bar{\mathbf{H}}^1 \in \{0, 1\}^{n_1 \times n_{e1}}$ 和 $\bar{\mathbf{G}}^2, \bar{\mathbf{H}}^2 \in \{0, 1\}^{n_2 \times n_{e2}}$ 表示，其中 $\bar{\mathbf{A}}^1 = \bar{\mathbf{G}}^1 \bar{\mathbf{H}}^{1T}, \bar{\mathbf{A}}^2 = \bar{\mathbf{G}}^2 \bar{\mathbf{H}}^{2T}$ 是两个图的邻接矩阵， $\bar{\mathbf{G}}_{i,k} = \bar{\mathbf{H}}_{j,k} = 1$ 表示边 k 将节点 i 连接到节点 j 。

此外，图像特征提取网络还包含一个分支，通过对 relu5_3 层的输出进行最大池化产生全局特征。来自两个图的池化全局特征随后被拼接起来，传递到一个全连接 (fc) 层，通过 tanh 激活函数，产生全局特征 $\bar{\mathbf{g}} \in \mathbb{R}^d$ ：

$$\bar{\mathbf{g}} = \tanh(\text{fc}([\bar{\mathbf{g}}^1 \quad \bar{\mathbf{g}}^2])), \quad (3.22)$$

其中 $[\cdot \quad \cdot]$ 表示拼接。节点特征和边特征分别对应有独立的 $\bar{\mathbf{g}}_{\text{node}}, \bar{\mathbf{g}}_{\text{edge}}$ 。基于加权内积，通过使用 $\bar{\mathbf{g}}$ 作为权重，构建节点相似度矩阵 $\mathbf{K}^P \in \mathbb{R}^{n_1 \times n_2}$ 和边相似度矩

⁶SplineConv 在其原始论文^[147] 中被称为 SplineCNN。不过，术语“CNN”通常代表图像上的卷积，而 SplineCNN 是在图上的卷积。本文将其写作 SplineConv，以避免歧义。

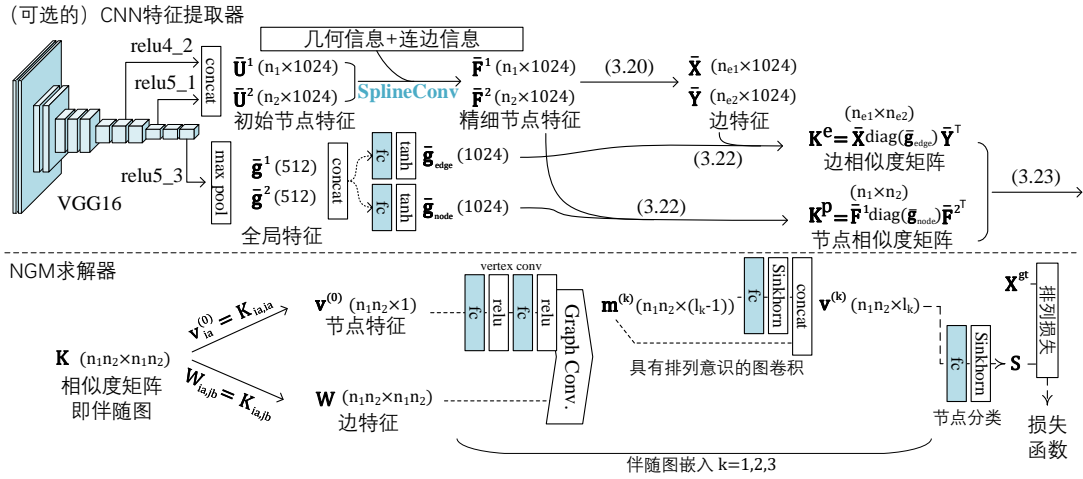


图 3-11 用于求解二图匹配问题的 NGM 神经网络结构。

Figure 3-11 Network structure of NGM to solve two-graph matching problem.

阵 $\mathbf{K}^e \in \mathbb{R}^{n_{e1} \times n_{e2}}$:

$$\mathbf{K}^e = \bar{\mathbf{X}} \text{diag}(\bar{\mathbf{g}}_{\text{node}}) \bar{\mathbf{Y}}^\top, \quad \mathbf{K}^p = \bar{\mathbf{F}}^1 \text{diag}(\bar{\mathbf{g}}_{\text{edge}}) \bar{\mathbf{F}}^{2\top}. \quad (3.23)$$

QAP 的相似度矩阵是根据 \mathbf{K} 的分解式构建的^[148]:

$$\mathbf{K} = \text{diag}(\text{vec}(\mathbf{K}^p)) + (\bar{\mathbf{G}}^2 \otimes \bar{\mathbf{G}}^1) \text{diag}(\text{vec}(\mathbf{K}^e)) (\bar{\mathbf{H}}^2 \otimes \bar{\mathbf{H}}^1)^\top, \quad (3.24)$$

其中 $\text{diag}(\cdot)$ 表示从输入向量构建对角矩阵, \otimes 表示 Kronecker 积。所有上述操作都允许反向传播。

伴随图构建。从相似度矩阵 \mathbf{K} 中可以得到伴随图。伴随图的带权邻接矩阵 \mathbf{W} 来自于 \mathbf{K} 的非对角线元素。将 $\mathbf{v}^{(k)} \in \mathbb{R}^{n_1 n_2 \times l_k}$ 表示为第 k 层 (从 $k=0$ 开始) 的 l_k 维节点嵌入。在 $k=0$ 时刻, 初始嵌入特征是标量, 即 $l_0=1$, 取自 \mathbf{K} 的对角线。

$$\mathbf{W}_{ia,jb} = \mathbf{K}_{ia,jb}, \quad \mathbf{v}_{ia}^{(0)} = \mathbf{K}_{ia,ia}, \quad (3.25)$$

\mathbf{W} 包含了伴随图中的连通性和权重信息。如果一阶相似度 $\mathbf{K}_{ia,ia}$ 不存在, 可以为所有 $\mathbf{v}^{(0)}$ 分配一个常数。

具有排列意识的伴随图嵌入。如图 3-9 所示, 图匹配问题等价于在伴随图中选择对应的顶点。由于 GCN 在图学习任务中展现出了简单且有效的优势, 本节采用了 GCN^[133] 作为基础架构对伴随图上的顶点进行分类。定义伴随图的 (无权) 邻接矩阵为 $\mathbf{A} \in \{0, 1\}^{n_1 n_2 \times n_1 n_2}$, 其中如果 $\mathbf{K}_{ia,jb} > 0$, 则 $\mathbf{A}_{ia,jb} = 1$, 否则为 $\mathbf{A}_{ia,jb} = 0$ 。 $\mathbf{A}_{ia,jb}$ 表示伴随图中顶点 ia 和 jb 之间是否存在一条边。回想一下, 顶点 ia 表示 \mathcal{G}^1 的节点 i 和 \mathcal{G}^2 的节点 a 之间的匹配 (见图 3-9)。在伴随图中, 当且仅当 i 匹配到 a 、 j 匹配到 b 可以共存, 且边 ij 和边 ab 分别在存在于 \mathcal{G}^1 、 \mathcal{G}^2 时, 伴随图中 ia 和 jb 之间才存在一条边。由于 \mathbf{A} 是对称的, 作者计算它的度作

为归一化矩阵:

$$\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}_{n_1 n_2}). \quad (3.26)$$

顶点特征更新的步骤如下所示:

$$\mathbf{m}^{(k)} = \mathbf{D}^{-1} \mathbf{W} f_m(\mathbf{v}^{(k-1)}) + f_v(\mathbf{v}^{(k-1)}), \quad \mathbf{v}^{(k)} = \mathbf{m}^{(k)}, \quad (3.27)$$

其中, 消息传递函数 $f_m: \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ 和顶点自更新函数 $f_v: \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ 均通过具有两个全连接层和 ReLU 激活函数的网络来实现。

上述的顶点嵌入过程在公式 (3.27) 中并没有考虑排列型问题的约束。为此, 作者开发了一个具有排列意识的嵌入模型: 在每一层中, 通过分类层和 Sinkhorn 层得到一个双随机矩阵, $\text{Classifier}: \mathbb{R}^{n_1 n_2 \times l_k} \rightarrow [0, 1]^{n_1 \times n_2}$, 接着进行向量化 $\text{vec}(\cdot)$ 。预测的双随机矩阵被拼接入顶点嵌入中, 从而在嵌入层中考虑排列约束。调整网络层的输出维度以容纳新的嵌入信息 $f_m, f_v: \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{(l_k-1)}$, 这种具有排列意识的嵌入算法在本文中被称为 **Sinkhorn 嵌入**。

$$\mathbf{v}^{(k)} = [\mathbf{m}^{(k)} \quad \text{vec}(\text{Classifier}(\mathbf{m}^{(k)}))], \quad (3.28)$$

其中 $[\cdot \quad \cdot]$ 表示拼接。本节的实验中比较了式 (3.27) 中的普通节点嵌入方法和式 (3.28) 中具有排列意识的 Sinkhorn 嵌入方法, 以验证添加排列约束的必要性。

基于 Sinkhorn 层的节点分类。由于图匹配等价于关联图上的顶点分类 (见图 3-9), 因此采用带有 Sinkhorn 网络的顶点分类器来预测匹配结果。具体而言, 使用一个单层全连接分类器 $f_c: \mathbb{R}^{l_k} \rightarrow \mathbb{R}$, 随后采用指数激活函数和熵正则化系数 τ :

$$\mathbf{s}_{ia}^{(k)} = \exp\left(\frac{f_c(\mathbf{v}_{ia}^{(k)})}{\tau}\right). \quad (3.29)$$

将分类得分变形为 $\mathbb{R}^{n_1 \times n_2}$ 后, 通过 Sinkhorn 网络层^[113,149]将一对一匹配的排列约束添加至 \mathbf{s} 。如 3.2.3 所讨论的, Sinkhorn 层可以将非负方阵作为输入并输出双随机矩阵^[113,134]。由于不同图的大小不同, 得分矩阵可能是非正方形的, 需要将输入矩阵 $\mathbf{S} \in \mathbb{R}^{n_1 \times n_2}$ 填充为一个方形矩阵 (假设 $n_1 \leq n_2$) 并用虚拟元素 (例如 $\epsilon = 10^{-3}$) 填充。3.2 节已讨论过, Sinkhorn 算子完全可微分, 并且可以通过自动微分技术^[135]进行高效实现。本节所提出的带有 Sinkhorn 网络的顶点分类器通过以下表示 $\text{Classifier}: \mathbb{R}^{n_1 n_2 \times l_k} \rightarrow [0, 1]^{n_1 \times n_2}$, 它也在公式 (3.28) 中被提及。

端到端训练的损失函数。对于存在监督信号的情况, 可以采用 3.2 节中介绍的排列损失函数进行监督训练:

$$L_{\text{perm}} = - \sum_{i=1}^{n_1} \sum_{a=1}^{n_2} \mathbf{X}_{i,a}^{gt} \log \mathbf{S}_{i,a} + (1 - \mathbf{X}_{i,a}^{gt}) \log(1 - \mathbf{S}_{i,a}). \quad (3.30)$$

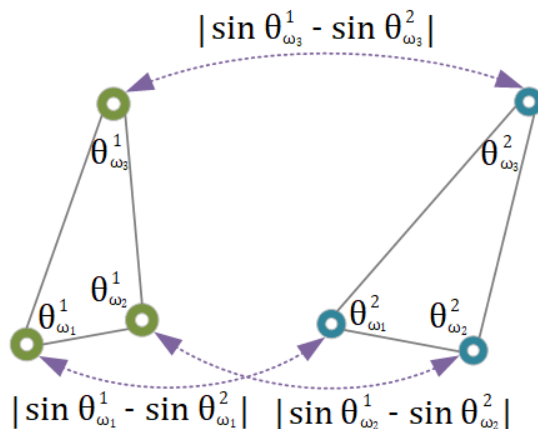


图 3-12 超图匹配考虑的三阶相似度

Figure 3-12 Third-order affinity adopted in hypergraph matching.

此外,NGM 方法还允许直接优化 QAPLIB 问题的目标函数,这将在实验中(3.3.4节)详细讨论。以上讨论的所有组件都是可微的。因此,NGM 求解器和可选的 CNN 特征提取器都可以通过反向传播和梯度下降进行学习。

神经超图匹配网络 NHGM。神经超图匹配 (NHGM) 利用高阶结构进行更加鲁棒的匹配。NHGM 与 NGM 的流程几乎相同,面向超图中的顶点分类采用了一种更一般的消息传递方式,例如^[150-151]。由于计算代价的爆炸性增长(随着阶数 t 的增加,参数复杂度为 $\mathcal{O}((n_1 n_2)^t)$),本文将超图限制为三阶,这也与大多数现有工作^[121,123]保持一致。此外,本方案在技术上具有推广到任意阶的灵活性。

构造伴随超图。在超图匹配中,二阶的相似度矩阵 \mathbf{K} 扩展到了 t 阶的相似度张量 $\mathbf{H}^{(t)}$ 。本节中,超图相似度在高维特征空间中计算得到,其定义如图 3-12 所示。作者将图 3-12 中的平面视为由 $\bar{\mathbf{F}}^1$ 和 $\bar{\mathbf{F}}^2$ 构成的特征空间,图 3-12 中的每个点代表具有特征的节点。第三阶相似度张量被定义为特征空间中角度的差异:

$$\mathbf{H}_{\omega_1, \omega_2, \omega_3}^{(3)} = \exp \left(- \left(\sum_{q=1}^3 |\cos \theta_{\omega_q}^1 - \cos \theta_{\omega_q}^2| \right) / \sigma_3 \right). \quad (3.31)$$

上述超图相似度的灵感来自于现有的超图匹配工作^[121,123],现有的超图相似度更多地来自几何信息。除了利用几何信息,本节的超图相似度还可以在高维特征空间中计算得到。

与二阶的伴随图类似,基于 \mathbf{H} 可以构建一个伴随超图 (association hypergraph)。伴随超图 $\mathcal{H}^A = (\mathcal{V}^A, \mathcal{E}^A)$ 将待匹配的节点 $\omega = (V_i^1, V_j^2)$ 视为顶点 \mathcal{V}^A (这与二阶的伴随图一致),而将 $(V_{\omega_1}^1, V_{\omega_1}^2), \dots, (V_{\omega_t}^1, V_{\omega_t}^2)$ 中的高阶相似性作为超边 \mathcal{E}^A ,如图 3-9(c) 所示。 \mathbf{H} 的元素对应到伴随超图的邻接权重。在 NHGM 中,作者定义了超图卷积来进行顶点分类。

具有排列意识的伴随超图嵌入。作为式 (3.27) 的扩展,顶点嵌入在伴随超图

中连接的所有顶点之间进行更新。首先计算用于归一化的 t 阶张量：

$$\mathbf{D}_{\omega_1, \dots, \omega_t}^{(t)-1} = \mathbf{A}_{\omega_1, \dots, \omega_t}^{(t)} / (\mathbf{A}^{(t)} \otimes_2 \mathbf{1} \cdots \otimes_t \mathbf{1})_{\omega_1}, \quad (3.32)$$

接着采用扩展自公式 (3.27) 的特征更新方式：

$$\begin{aligned} \mathbf{p}^{(k)} &= f_m^{(t)}(\mathbf{v}^{(k-1)}), \mathbf{H}^{(t)'} = (\mathbf{D}^{(t)-1} \odot \mathbf{H}^{(t)})_{t+1}, \\ \mathbf{m}^{(k)} &= \sum_t \lambda_t \mathbf{H}^{(t)'} \otimes_t \mathbf{p}^{(k)} \cdots \otimes_2 \mathbf{p}^{(k)} + f_v(\mathbf{v}^{(k-1)}), \\ \mathbf{v}^{(k)} &= \left[\mathbf{m}^{(k)} \quad \text{vec}(\text{Classifier}(\mathbf{m}^{(k)})) \right], \end{aligned} \quad (3.33)$$

其中， \otimes_i 表示按维度 i 进行张量乘法（参见公式 (3.4)）， \odot 表示逐元素的乘法， $(\cdot)_{t+1}$ 表示沿着 $(t+1)$ 维度进行扩展。 $f_m^{(t)}: \mathbb{R}^{k-1} \rightarrow \mathbb{R}^{(k-1)}$ 是第 t 阶的信息传递函数。通过 λ_t 的加权求和融合不同阶的特征。

NHGM 的其他模块，包括分类器和排列损失函数，与 NGM 相同。因此，NGM 可以被视为 NHGM 的特殊情况，其中阶数被限制为 2。在具体实现中， \mathbf{H} 的稀疏性也被用于提高时间和内存效率。

神经多图匹配网络 NMGM。 本节还探索了学习多图匹配，其中信息通过循环一致性 (cycle-consistency) 在多图之间进行融合。循环一致性的定义是，任意两个图的匹配关系在经过第三图传递后仍然是一致的，即对于任意 i, j, k ，有 $\mathbf{X}_{ij} = \mathbf{X}_{ik} \mathbf{X}_{kj}$ 。这可以被看作每个图通过 $\mathbf{X}_i \in \{0, 1\}^{n \times n}$ 匹配到一个大小为 n 的中心集上。 \mathcal{G}^i 和 \mathcal{G}^j 之间的匹配可以用 $\mathbf{X}_{ij} = \mathbf{X}_i \mathbf{X}_j^\top$ 表示。目前，学术界已经有一系列基于初始匹配的多图匹配工作^[125,127,152-153]。由于其有效性、简单性以及支持端到端训练的能力，NMGM 使用了谱方法^[127]。本节假设所有图的大小相同。

构造联合匹配矩阵。首先通过 NGM 获取初始的二图匹配，以构建一个对称的联合匹配矩阵 $\mathcal{S} \in \mathbb{R}^{nm \times nm}$ (joint matching matrix)。对于每一对 n 个节点的图 \mathcal{G}^i 和 \mathcal{G}^j ， $\mathbf{S}_{ij} \in [0, 1]^{n \times n}$ 是由 NGM 计算的双随机矩阵。对于 m 个图，可以在两两匹配的基础上构建 \mathcal{S} ：

$$\mathcal{S} = \begin{pmatrix} \mathbf{S}_{00} & \cdots & \mathbf{S}_{0m} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{m0} & \cdots & \mathbf{S}_{mm} \end{pmatrix}. \quad (3.34)$$

对于 \mathcal{S} 的对角部分， \mathbf{S}_{ii} 都是单位矩阵。所有 \mathbf{S}_{ij} 都是方阵。谱方法的目标是获得一个具有循环一致性的联合匹配矩阵 $\hat{\mathcal{S}}$ ，其与 \mathcal{S} 的内积被最大化：

$$\max_{\hat{\mathcal{S}}} \text{tr}(\hat{\mathcal{S}}^\top \mathcal{S}). \quad (3.35)$$

为确保循环一致性，可以选择任意一个图 k 作为中心集，省略 k 的下标，将 $\hat{\mathcal{S}}$ 分

解为相对于中心集的匹配:

$$\hat{\mathbf{U}}\hat{\mathbf{U}}^\top = \hat{\mathcal{S}}, \quad \text{where} \quad \hat{\mathbf{U}} = \begin{pmatrix} \hat{\mathbf{S}}_0 \\ \vdots \\ \hat{\mathbf{S}}_m \end{pmatrix}. \quad (3.36)$$

在理想条件下, 所有 $\hat{\mathbf{S}}_i$ 都是排列矩阵, $\hat{\mathbf{U}}$ 的每一列都是线性无关的, $\hat{\mathbf{U}}/\sqrt{m}$ 是 $\hat{\mathcal{S}}$ 的 m 个特征值为 m 的特征向量。由于计算复杂度, $\hat{\mathbf{S}}_i$ 中的排列约束被放松, 因此公式 (3.35) 变成了一个广义瑞利问题 (generalized Rayleigh problem), 并通过谱方法求解, 具体如下所示。

可微分的多图匹配方法——谱方法。多图匹配可以通过对 \mathcal{S} 进行特征向量分解 (即谱方法) 来实现。基于广义瑞利问题, 在每个图中给定 n 个节点, 提取对称矩阵 \mathcal{S} 的前 n 个特征值所对应的特征向量:

$$\mathbf{U}\Sigma\mathbf{U}^\top = \mathcal{S}, \quad (3.37)$$

其中对角矩阵 $\Sigma \in \mathbb{R}^{n \times n}$ 包含前 n 个特征值, $\mathbf{U} \in \mathbb{R}^{mn \times n}$ 是对应的 n 个特征向量。过去工作已经证明特征向量分解是可微的^[154], 这使它们能够成为端到端深度学习框架中的组件。基于 \mathcal{S} 的多图匹配问题将会输出如下结果, 可以看作对公式 (3.35) 的松弛解:

$$\hat{\mathcal{S}} = m\mathbf{U}\mathbf{U}^\top. \quad (3.38)$$

公式 (3.37) 中特征向量分解的梯度是^[154]

$$\frac{\partial L}{\partial \mathcal{S}} = \mathbf{U} \left(4m \left(\mathbf{Y}^\top \odot \left(\mathbf{U}^\top \left(\frac{\partial L}{\partial \hat{\mathcal{S}}} \right)_{sym} \mathbf{U} \right)_{sym} \right) \right) \mathbf{U}^\top, \quad (3.39)$$

$$\text{where} \quad \mathbf{Y}_{ij} = \begin{cases} 1/(\sigma_i - \sigma_j) & i \neq j \\ 0 & i = j \end{cases}$$

其中 L 表示损失函数, \odot 表示逐元素乘法, $\mathbf{A}_{sym} = (\mathbf{A} + \mathbf{A}^\top)/2$, $\sigma_i = \Sigma_{ii}$ 表示第 i 个特征值。根据这个反向传播公式, 如果存在相同的特征值, 即当 $i \neq j$ 且 $\sigma_i = \sigma_j$ 时, 将会在反向传播中出现除以 0 的数值错误。特别地, 当输入 \mathbf{S} 已经满足循环一致性 (即 $\mathbf{S}_{ij} = \mathbf{S}_{ik}\mathbf{S}_{kj}$) 时, 经常会出现此问题——在这种情况下, 多图匹配结果与原始匹配结果几乎相同。为避免数值错误, 可以在这种情况下跳过特征分解, 直接赋值 $\hat{\mathbf{S}}_{ij} = \mathbf{S}_{ij}$ 。在具体算法中, 如果前 n 个特征值中的残差小于 δ , 例如 $\delta = 10^{-4}$, 即可跳过特征分解。实验发现, 这种策略可以有效地稳定学习过程。

最终的匹配结果通过再次调用 Sinkhorn 算法得到：

$$\bar{\mathbf{S}}_{ij} = \text{Sinkhorn} \left(\exp \left(\frac{\hat{\mathbf{S}}_{ij}}{\hat{\tau}} \right) \right) \quad (3.40)$$

其中经过多图匹配处理后的结果 $\hat{\mathbf{S}}_{ij}$ 来自矩阵 $\hat{\mathbf{S}}$ ， $\exp(\cdot / \hat{\tau})$ 为 Sinkhorn 算法提供了熵正则项^[114]。和二图匹配的监督类似，公式 (3.15) 介绍的排列损失函数可作用于所有的 $\bar{\mathbf{S}}_{ij}$ 。

进一步讨论。

学习排列型组合优化的问题结构。作者认为，本节介绍的神经网络求解器的内在工作模式实际上学习了二次指派问题的基本结构。由于二次指派问题与伴随图之间的联系，Lawler's QAP 的原始数学形式被转化为了现代深度学习模型可处理的结构。问题结构由图神经网络学习，从而化简成了可通过 Sinkhorn 算法可微求解的线性指派问题。与本节类似，文献^[54]考虑了一些图上的组合优化问题，其中问题结构通过 GNN 学习、简化，并通过贪心的形式求解。

具有排列意识的图嵌入。本节介绍了具有排列意识的 Sinkhorn 嵌入方法，可以在浅层的图嵌入中添加排列约束。不然，排列约束直到最后的输出层——Sinkhorn 层才会被考虑。过去的研究已经证明，更早地考虑排列信息对非学习的求解器 (RRWM^[34] vs SM^[35]) 和基于学习的求解器 (PCA-GM vs PIA-GM, 3.2节) 都是有效的。本节通过仿真实验和在真实图像上的消融实验揭示了 Sinkhorn 嵌入的重要性；此外，通过引入多头 Sinkhorn 嵌入可以进一步提高性能，但会带来额外的计算量。

在优化问题中引入 *Gumbel* 采样。作为一种常见的后处理步骤，匈牙利算法^[10] 以确定性的方式将双随机矩阵 \mathbf{S} 转换为一个排列矩阵 \mathbf{X} 。从概率的角度看， \mathbf{S} 代表了在排列矩阵空间上的分布，监督学习中的排列损失函数最小化了概率 \mathbf{S} 和真实分布 \mathbf{X}^{gt} 之间的距离。匈牙利算法选择了概率最高的排列矩阵。在处理模式识别、计算机视觉领域的匹配时，这种贪心策略是成功的。然而，在优化目标函数时，分布内很可能存在一个更好的解。因此，本节采用 Gumbel-Sinkhorn^[149]，将公式 (3.29) 替换为

$$\mathbf{s}_{ia}^{(k)} = \exp \left(\frac{(f_c(\mathbf{v}_{ia}^{(k)}) + g)}{\tau_g} \right), \quad (3.41)$$

随后采用标准的 Sinkhorn 算法进行处理。其中，添加的噪声 g 从标准的 Gumbel 分布中采样，其累积分布函数 (cumulative distribution function, CDF) 是

$$G(x) = e^{-e^{-x}}, \quad (3.42)$$

该公式代表了一种极端值的分布。通过公式 (3.41)，可以从原始分布中采样得到更稀疏的双随机矩阵，重复采样可以得到一批双随机矩阵。这些矩阵接下来

表 3-8 总结：3.3节提出的所有方法及其变体。

Table 3-8 Notation of all proposed methods and their variants in Sec 3.3.

模型名称	求解能力	介绍
NGM	QAP/二图匹配	3.3.3节介绍的神经图匹配模型
NHGM	超图匹配	3.3.3节介绍的神经超图匹配模型
NMGM	多图匹配	3.3.3节介绍的神经多图匹配模型
NGM-V	QAP/二图匹配	将 NGM 的 Sinkhorn 嵌入 (3.28) 替换为普通嵌入 (3.27)
NGM-MH	QAP/二图匹配	具有多头 (8 头) Sinkhorn 嵌入的 NGM 模型
NGM-SF	多图匹配	训练完毕的 NGM 模型配合非学习的多图谱方法
NGM-GX	QAP/二图匹配	利用 Gumbel-Sinkhorn 采样 X 次, 返回目标函数最优的解

通过匈牙利离散化, 计算目标分数并选择其中的最优解作为最终解。通过调整 Gumbel 采样次数, 可以平衡探索程度和计算速度。Gumbel 分布在排列型组合优化的理论、应用将在3.4节进一步阐述。

3.3.4 实验分析

本节的实验测试了两种 Lawler's QAP 优化场景: 1) 仿真的点配准实验, 它以相似度矩阵/相似度张量作为输入; 2) QAPLIB 中来自真实世界的大规模 QAP 问题, 其中神经网络学习如何最小化目标函数。本节还测试了神经网络求解器在 Lawler's QAP 的视觉应用场景, 即学习图像关键点的 CNN 特征并在真实图像上进行匹配。

本节还进行了超图匹配和多图匹配的测试, 以评估 NHGM 和 NMGM 的表现。基于 PyTorch 实现的 NGM、NHGM、NMGM 包括一个三层 GNN, 其中隐层特征维度 $l_1 = l_2 = l_3 = 16$ 。其他的超参数设置为 $\tau = \hat{\tau} = 0.05, \lambda_2 = 1, \lambda_3 = 1.5$ 。批 (batch) 的大小设置为 8。NGM、NHGM 和 NMGM 使用 Adam 优化器进行训练^[155]。详细的学习率配置在详细实验部分列出。表 3-8总结了本节提出的所有方法及其变体。

仿真数据上的纯 QAP 优化。 仿真实验的任务是对二维平面上随机生成的点集进行匹配, 作者与其他非学习的图匹配求解器进行了比较。每次试验生成了 10 组坐标在平面 $\mathcal{U}(0,1) \times \mathcal{U}(0,1)$ 内的点集。通过添加 $\mathcal{U}(1 - \delta_s, 1 + \delta_s)$ 的随机缩放和随机噪声 $\mathcal{N}(0, \sigma_n^2)$, 得到扰动后的仿真点集。从每个点集中, 采样 200 个图用于训练、100 个图用于测试, 共得到 2,000 个训练样本和 1,000 个测试样本。假设图结构对于图匹配求解器是未知的, 实验采用德罗内 (Delaunay) 三角剖分构建参考图, 目标图 (可能包含外点) 具有全连接的结构。外点也是从 $\mathcal{U}(0,1) \times \mathcal{U}(0,1)$ 中随机采样。默认情况下, 有 10 个内点、没有外点, $\delta_s = 0.1, \sigma_n = 0$ 。实验中, 对所有方法构造了相同的相似度矩阵, 即不同方法求解的 Lawler's QAP 相同。需要注意的是, 本实验直接学习输入的相似度矩阵/相似度张量, 不学习任何相似度函数。GMN^[16]、PCA-GM (3.2节) 等现有的图匹配机器学习方法均不支持这

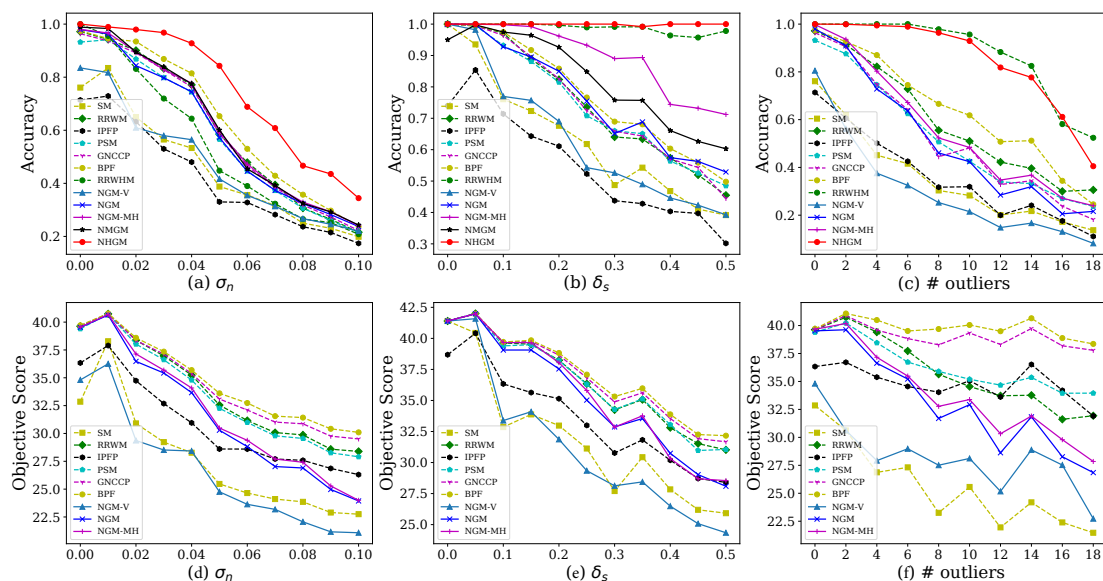


图 3-13 基于不同的扰动 σ_n , δ_s 和外点数目的仿真实验结果。

Figure 3-13 Synthetic test by varying deformation σ_n , δ_s and number of outliers.

一实验设定，因此它们没有出现在对比方法中。

由于现有的学习方法^[16,146]不能处理给定相似度矩阵的 Lawler's QAP 问题，本实验主要比较非学习的 QAP 求解方法：**1) SM**^[35] 将图匹配视为聚类，通过谱方法求解；**2) RRWM**^[34] 采用随机游走和跳跃方法来进行图匹配；**3) IPFP**^[87] 通过整数投影迭代地改进给定的解；**4) PSM**^[156] 通过概率视角改进了 SM；**5) GNCCP**^[157] 是用于图匹配的凸凹路径跟随方法；**6) BPF**^[142] 通过分支切换改进了路径跟随技术，在图匹配上达到了最先进的性能。此外，超图匹配算法 **7) RRWHM**^[123] 将强大的 RRWM 扩展到超图场景。为了处理超图匹配，RRWHM 跟随文献^[123]将二阶相似度整合到了三阶张量中。在这个实验中，二阶的相似度定义为 $\mathbf{K}_{ia,jb} = \exp(-(\mathbf{f}_{ij} - \mathbf{f}_{ab})^2 / \sigma_2^2)$ ，其中 \mathbf{f}_{ij} 是边 ij 的长度。根据经验，设定 $\sigma_2^2 = 5 \times 10^{-7}$ 。第三阶相似度模型遵循公式 (3.31)，其中 $\sigma_3^2 = 0.01$ 。

为了公平地比较计算效率，实验使用 GPU 重新实现了原理上支持并行的 SM、RRWM 和 RRWHM 求解器，这比它们在单线程 CPU 上的原始版本效率更高。其他方法由于涉及迭代计算和复杂的分支策略，不适合用 GPU 实现。因此，实验比较了文献^[142]发布的 CPU 版本。NGM-V、NGM-MH、NGM-SF 的定义可见表 3-8。仿真实验中，学习率从 10^{-2} 开始，每 5,000 步降低 10 倍。多图匹配算法默认同时匹配 4 个图。匈牙利算法^[10]被用作共同的离散化方法。

结果和分析。图 3-13(a-c) 表明，本节提出的 NGM 在匹配精度上与目前最先进的求解器相当，并且在严重的随机缩放情况下可以超越传统求解器（如图 3-13(b)）。多头的 Sinkhorn 嵌入模型 NGM-MH 进一步提高了匹配精度。在第三阶相似度的帮助下，NHGM 在面对噪声、缩放和外点的鲁棒性上表现出最优的性能。与非学习的超图匹配 RRWHM^[123] 相比，NHGM 在存在外点的情况下与

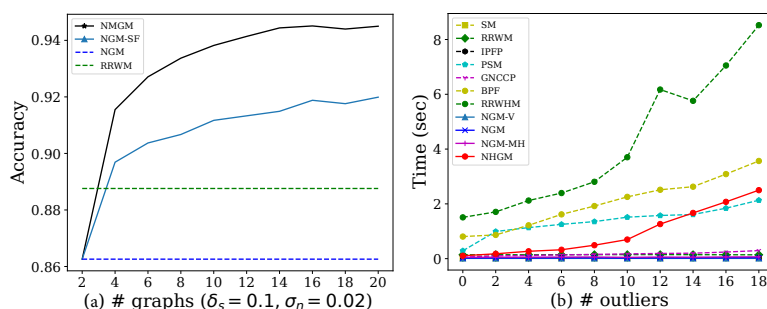


图 3-14 基于不同图数目/外点数目的仿真实验结果。

Figure 3-14 Synthetic test by varying number of graphs/outliers.

RRWHM 表现相当，如图 3-13(c) 所示。而在图 3-13(a&b) 中，NHGM 对噪声和缩放更加鲁棒。

通过融合多个图的匹配信息，NMGM 稳定地提升了 NGM，并且在图 3-14(a) 中展示了通过引入更多图来学习联合匹配的必要性——NMGM 的性能显著优于 NGM-SF，而后者的权重来自在二图匹配问题上训练的 NGM 模型。如图 3-14(b) 所示，NGM、NGM-V 和 NGM-MH 属于最快的图匹配算法。由于高效的 GPU 并行化，NHGM 比最先进的非学习方法 PSM^[156] 和 BPF^[142] 还要快。作为对比，传统的超图匹配算法如 RRWHM^[123] 通常比二阶图匹配慢得多，并且不可扩展到更大的问题规模。

图 3-13(d-f) 报告了通过二图匹配方法求解 QAP 的目标函数值。有趣的是，相对于较高的准确度，学习求解器的目标函数值相对较低。正如在 3.3.2 节讨论的那样，QAP 目标函数在噪声条件下可能存在偏差，即 QAP 的最优解可能与真实匹配不符。本节的机器学习求解器在监督信号的引导下忽略了相似度矩阵中的噪声。这种现象在匹配真实世界图像时变得更加严重，基于学习的求解器的优势变得更加明显，详细情况将在稍后的真实图像任务中展示。

通过 NGM-V 和 NGM 之间的精度差距可以看出具有排列意识的 Sinkhorn 嵌入的有效性。在 NGM-MH 中通过多头 Sinkhorn 嵌入实现了进一步提升，特别是在随机缩放的情况下，如图 3-13(b) 所示。特别地，没有 Sinkhorn 嵌入的 NGM-V 的工作方式类似于 SM，嵌入过程不考虑排列约束，而它们的表现也非常接近；另一方面，基于 Sinkhorn 嵌入的 NGM 和 NGM-MH 在概念上类似于 RRWHM，它们都试图在浅层网络考虑排列约束。它们的性能也非常接近。

真实世界 QAP 问题的机器学习求解。 NGM 可以求解最通用的 Lawler's QAP 问题，该问题具有视觉领域之外广泛的应用场景。在 QAPLIB^[158] 数据集上进行的实验评估展示了 NGM 学习优化 QAP 目标函数的能力（在图匹配中，目标函数应该被最大化，而在 QAP 中是最小化）。QAPLIB 包含了 15 个类别、134 个来自真实世界的 QAP 问题，例如规划医院的设施布局^[159]。问题大小 $n_1 = n_2 = n$ ，其范围从 12 到 256。本文的实验在 133 个实例上进行，其中 $12 \leq n \leq 150$ 。最具挑战性的 tai256c 问题对目前的计算机来说复杂度太高了（存储中间结果需要

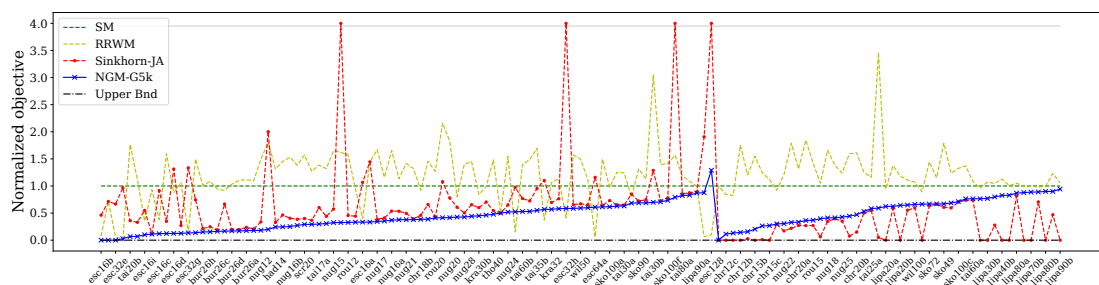


图 3-15 性能最好的 *NGM-G5k* 与其他非学习求解器在归一化目标函数下的对比 (越低越好)。

Figure 3-15 Normalized objective score (lower is better) of our best-performing *NGM-G5k* against learning-free QAP solvers.

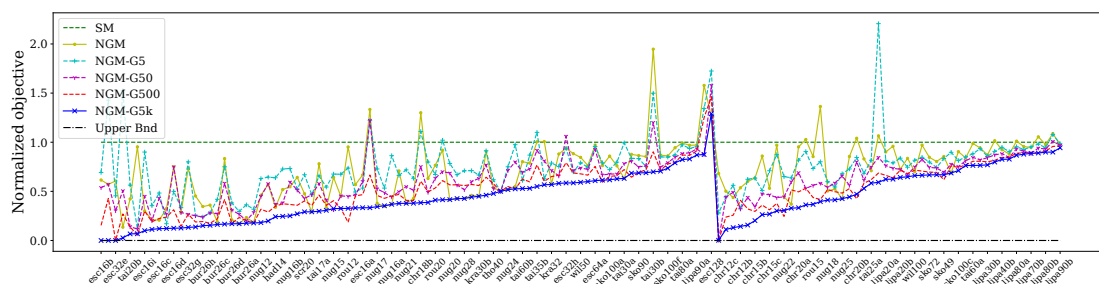


图 3-16 所有 *NGM* 变种在归一化目标函数下的对比 (越低越好)。

Figure 3-16 Normalized objective score (lower is better) comparing different *NGM* variants.

275GB 显存)。将损失函数设置为 QAP 的目标函数，并保持模型架构不变，作者制定了一个优化任务，使目标函数最小化：

$$L_{obj} = \text{vec}(\mathbf{S})^\top \mathbf{K} \text{vec}(\mathbf{S}) \quad (3.43)$$

其中， \mathbf{S} 是 *NGM* Sinkhorn 层的输出。当达到最优的 L_{obj} 时，学习得到的 \mathbf{S} 是原始 QAP 问题在双随机松弛下的解。为了探索可行域的其他部分，实验在测试中采用了 Gumbel-Sinkhorn 方法。

QAPLIB 问题的命名规则如下：前缀是问题类别，即提出问题的作者名字缩写，后面的数字表示问题的规模。如果有多个相同规模的问题，QAPLIB 会添加字母后缀以进行区分：

$$\begin{array}{ccc} \text{bur} & - & 26 & - & \text{a} \\ \text{作者名} & & \text{问题规模} & & \text{后缀 (可选)} \end{array}$$

实验为所有具有相同前缀 (即来自相同类别) 的问题训练一个网络，因为它们通常具有相同的结构 (例如，所有的 bur 问题都是键盘布局设计问题)。同一类别的问题的规模可能会有所不同 (例如，esc 问题的规模从 16 到 128 不等)。

实验结果由归一化的目标函数表示。为了简洁、直观，归一化的目标函数

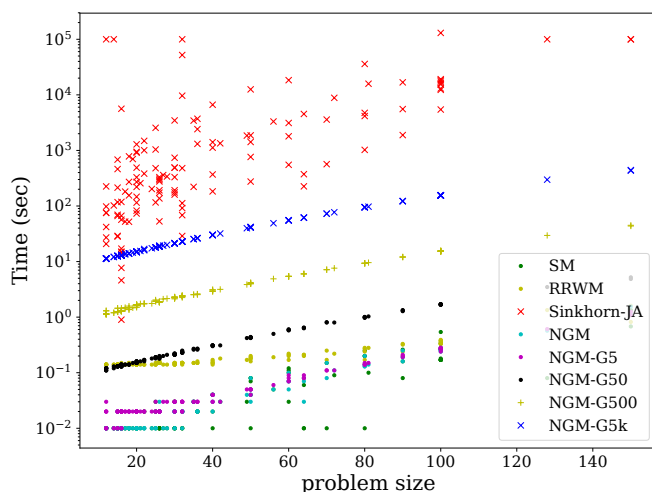


图 3-17 运行时间（对数尺度）与问题规模（每个图的节点数目）之间的关系。
Figure 3-17 Run time (log-scale) against problem size i.e. number of nodes for each graph.

基于最新的在线测评结果⁷中的上界（原始上界）计算，并面向基准方法谱匹配 (SM)^[35]进行归一化：

$$\text{归一化分数} = \frac{\text{求解的目标函数} - \text{目标函数上界}}{\text{SM 目标函数} - \text{目标函数上界}} \quad (3.44)$$

实验对标准的 NGM 模型和具有不同 Gumbel 采样数量的 NGM (NGM-GX, X=样本数) 性能进行了验证。学习率初始化为 10^{-4} ，每 5,0000 步降低 10 倍。批的大小设置为 1，Gumbel Sinkhorn 的正则化参数 $\tau_g = 1$ 。实验公平地比较了 NGM 方法与 GPU 实现下的 RRWM^[34]和 SM^[35]，并提供了 Sinkhorn-JA 的结果^[143]（在 Intel Xeon CPU @ 2.40 GHz 上运行）。对于文献^[143]中缺失的问题，作者假设 Sinkhorn-JA 未能取得任何可行解，因为原始论文中没有解释部分问题结果缺失的原因。

结果和分析。在图 3-15 中，本节的方法超过了 RRWM^[34]和 SM^[35]，并且在与最先进的 Sinkhorn-JA^[143]比较中表现出色。由于不存在基于学习的 QAP 求解器，实验只比较了非学习方法。图 3-16 验证了 Gumbel 采样的有效性，其中基于 Gumbel 的 NGM-G5k 始终优于确定性的 NGM，而后者总是通过匈牙利算法选择概率最高的排列。随着采样数量的减少，基于 Gumbel 的方法的性能逐渐退化。这表明，对采样空间进行更多的探索可以提高解的质量。

进一步的评测结果如表 3-9 和图 3-17 所示。通过学习和 Gumbel 采样，本节的 NGM-G5k 在 133 个实例中有 72 个实例找到了最佳解，而非学习的最先进求解器 Sinkhorn-JA^[143]在 46 个实例上表现更好。这个结果表明，相比于传统的求解器，基于学习的求解器例如 NGM 可以适应更广泛的问题。更重要的是，表现

⁷<http://anjos.mgi.polymtl.ca/qaplib/inst.html>

表 3-9 在所有 *QAPLIB* 问题上，达到最佳性能的频次统计。

Table 3-9 Best-performing occurrence count on *QAPLIB* among all instances and all tested solvers.

问题类别	bur	chr	els	esc	had	kra	lipa	nug	rou	scr	sko	ste	tai	tho	wil	总计
问题总数	8	14	1	19	5	3	16	15	3	3	13	3	25	3	2	133
SM ^[35]	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
RRWM ^[34]	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	9
Sinkhorn-JA ^[143]	0	14	1	1	0	0	15	4	1	0	1	0	7	1	1	46
NGM	0	0	0	1	0	2	0	5	0	0	0	3	1	0	0	12
NGM-G5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
NGM-G50	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	2
NGM-G500	1	0	0	3	0	0	0	1	1	0	5	0	1	0	0	12
NGM-G5k	8	1	0	11	5	3	1	10	1	3	7	2	17	2	1	72

表 3-10 *QAPLIB* 中，若干统计量和 *NGM-G5k*、*Sinkhorn-JA* 的 *gap* (公式 (3.45)) 之间的皮尔森相关系数。

Table 3-10 Pearson correlation coefficient between the listed statistics in *QAPLIB* and the corresponding *gap* (Eq. (3.45)) of *NGM-G5k* and *Sinkhorn-JA*.

	nz	nz/n^4 (sparsity)	$\mathbf{K}_{std}/\mathbf{K}_{max}$	$d_{min}/\bar{\mathbf{K}}$	$d_{max}/\bar{\mathbf{K}}$	$d_{std}/\bar{\mathbf{K}}$	d_{max}/\bar{d}	d_{std}/\bar{d}
$gap(\text{NGM-G5k})$	0.130	0.631	0.212	-0.222	0.230	0.286	0.230	0.280
$gap(\text{Sinkhorn-JA})$	0.270	0.545	-0.090	-0.220	0.291	0.355	0.291	0.475
$gap_{\text{NGM}} - gap_{\text{SJA}}$	-0.184	-0.121	0.234	0.071	-0.137	-0.163	-0.137	-0.288

最佳的神经网络 *NGM-G5k* 比 *Sinkhorn-JA* 快一个数量级，并且通过调整 Gumbel 方法的采样数量可以平衡解的质量和计算开销。最后，图 3-18 中的混淆矩阵展示了在不同实例之间的泛化能力，其中 *NGM-G5k* 在 *y* 轴的实例上进行了训练、在 *x* 轴的实例上进行测试。更深的颜色、更低的目标函数代表了更好的性能。*NGM* 模型可以有效地泛化到不同问题规模且未在训练中见过的问题实例。总之，机器学习 *QAP* 求解器在 *QAPLIB* 上实现了最佳的准确性和速度的权衡，并且可以在不同问题之间进行泛化。

如表 3-9 和图 3-15 所示，非学习的 *Sinkhorn-JA*^[143] 和机器学习的 *NGM-G5k* 在 *QAPLIB* 的不同类别上表现更好，例如 *Sinkhorn-JA* 在 *chr* 和 *lipa* 类上表现更好，而机器学习方法在 *bur*、*esc*、*nug*、*scr* 和 *tai* 类上更具优势。本文对此开展了一些统计研究，以发现模型表现和问题结构之间的关系，以期为未来的机器学习和非学习的求解器研究提供启示。

对于每个问题，作者从每个实例的相似度矩阵中统计了一些指标：问题规模 n ，平均值 $\bar{\mathbf{K}}$ ，最小值 \mathbf{K}_{min} ，最大值 \mathbf{K}_{max} ，标准差 \mathbf{K}_{std} ，零元素数量 nz ，平均度数（伴随图的度） \bar{d} ，最小度数 d_{min} ，最大度数 d_{max} 和度的标准差 d_{std} 。算法的

Training instance	bur26a	5621774	5882	123120	4406	227404	1439332	327610
	had18	5953371	5558	127730	4962	274630	1483226	350332
	kra30a	5801101	5894	114410	5086	206570	1422984	330654
	nug25	5945588	5998	137840	4420	180220	1428212	334760
	scr20	5873473	5778	138020	4966	154636	1427996	331832
	tai25a	5847762	5868	125850	4938	239580	1352912	346560
	tho40	5895581	5944	127910	5190	235470	1484768	304878
		bur26a	had18	kra30a	nug25	scr20	tai25a	tho40
		Testing instance						

图 3-18 用于测试 *QAPLIB* 问题上泛化能力的混淆矩阵。
Figure 3-18 Generalization test by confusion matrix for *QAPLIB*.

表 3-11 *NGM* 等方法 *Pascal VOC* 关键点数据集 (无外点) 的匹配精度 (%)。
Table 3-11 Matching accuracy (%) of *NGM* on *Pascal VOC Keypoint (no outliers)*.

图匹配方法	✈	🚲	🦋	🐼	🔪	🚗	🚘	🚙	🚒	🚓	🚔	🚕	🚖	🚗	🚘	🚙	🚒	🚓	🚔	🚕	🚖	平均
GMN ^[16]	41.6	59.6	60.3	48.0	79.2	70.2	67.4	64.9	39.2	61.3	66.9	59.8	61.1	59.8	37.2	78.2	68.0	49.9	84.2	91.4	62.4	
PCA-GM (3.2节)	49.8	61.9	65.3	57.2	78.8	75.6	64.7	69.7	41.6	63.4	50.7	67.1	66.7	61.6	44.5	81.2	67.8	59.2	78.5	90.4	64.8	
IPCA-GM (3.2节)	53.8	66.2	67.1	61.2	80.4	75.3	72.6	72.5	44.6	65.2	54.3	67.2	67.9	64.2	47.9	84.4	70.8	64.0	83.8	90.8	67.7	
CIE-H ^[77]	49.9	63.1	70.7	53.0	82.4	75.4	67.7	72.3	42.4	66.9	69.9	69.5	70.7	62.0	46.7	85.0	70.0	61.8	80.2	91.8	67.6	
LCS ^[103]	46.9	58.0	63.6	69.9	87.8	79.8	71.8	60.3	44.8	64.3	79.4	57.5	64.4	57.6	52.4	96.1	62.9	65.8	94.4	92.0	68.5	
BBGM ^[82]	61.9	71.1	79.7	79.0	87.4	94.0	89.5	80.2	56.8	79.1	64.6	78.9	76.2	75.1	65.2	98.2	77.3	77.0	94.9	93.9	79.0	
NGM	61.8	71.2	77.6	78.8	87.3	93.6	87.7	79.8	55.4	77.8	89.5	78.8	80.1	79.2	62.6	97.7	77.7	75.7	96.7	93.2	80.1	
NHGM	59.9	71.5	77.2	79.0	87.7	94.6	89.0	81.8	60.0	81.3	87.0	78.1	76.5	77.5	64.4	98.7	77.8	75.4	97.9	92.8	80.4	

表现通过求解得到的目标函数与上界的 *gap* 来表示:

$$gap = \frac{\text{求解的目标函数} - \text{目标函数上界}}{\text{求解的目标函数}} \tag{3.45}$$

gap 表示当前方法相较已知的最优解 (通常在极高的复杂度下求解得到) 可以提高的百分比。皮尔森 (Pearson) 相关系数 *r* 计算每个 *gap* 和相应统计量之间的相关性, 以及一些有意义的统计量组合。表 3-10 列出了 $|r| \geq 0.2$ 的项目, 其中正相关意味着对求解器的性能有负面影响, 因为较低的 *gap* 更好。表 3-10 还包含了统计量和两种方法 *gap* 之差的相关性。

在表 3-10 的前两列中, 更高的稀疏性 (nz/n^4 , 相似度矩阵中零元素的比例)、更多的零元素 *nz* 会使 QAP 问题变得更难。然后, 归一化的标准差 $\mathbf{K}_{std}/\mathbf{K}_{max}$ 对 NGM-G5k 表现出一些微弱的负面影响, 但对 Sinkhorn-JA 的影响很小。在最后五列中, 两种方法都受到伴随图更高度数的影响, 而 Sinkhorn-JA 似乎对度数的归一化标准差 $d_{std}/\bar{\mathbf{K}}$ 、 d_{std}/\bar{d} 更敏感。总之, 稀疏性是 NGM-G5k (其中消息传递路径被阻塞) 和 Sinkhorn-JA (难以找到更紧的下界) 的关键挑战。此外, 与伴随图学习相结合的学习可以抑制度数的波动。未来的改进方向可能包括设计更高容量的图学习模型、针对稀疏伴随图设计全局通信机制等。

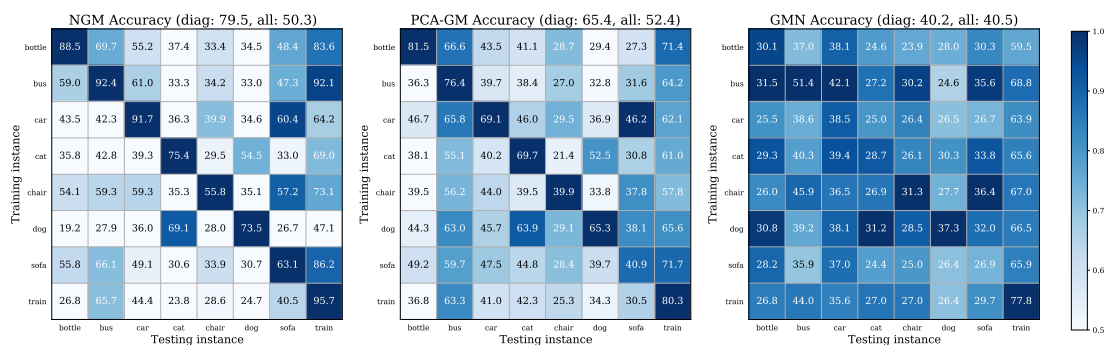


图 3-19 NGM 方法在 *Pascal VOC* 关键点数据集上的泛化性实验。
Figure 3-19 Generalization study of NGM on *Pascal VOC* Keypoint dataset.

在真实图像上联合学习 CNN 和 QAP 求解。本节的匹配网络允许使用原始图像输入，其中包括一个卷积神经网络（见图 3-11）。用于评测的 *Pascal VOC* 关键点数据集和 Willow ObjectClass 数据集已在 3.2 节中介绍。

在 *Pascal VOC* 关键点数据集上的结果。这个真实图像数据集^[137] 包含了 20 个具有关键点标签的实例类别。此处遵循 3.2 节的实验方法，将带有内点位置的图像输入模型进行关键点匹配。*Pascal VOC* 关键点是一个颇具挑战性的数据集，因为物体可能因其尺度、姿态和照明而变化，每个图像中的关键点数量从 6 到 23 不等。由于 NMG 的多图匹配机制需要保证图之间的节点数量相同，没有在 *Pascal VOC* 关键点数据集上测试 NMG。

遵循文献^[16]和 3.2 节的实验设定，删除了匹配无意义且标注质量低的图像，最终共有 7,020 个训练样本和 1,682 个测试样本被用于实验。在将物体沿着其边界框裁剪并调整为 256×256 大小后，再输入网络。如 3.3.3 节所讨论的，作者采用 VGG16 主干网络^[128]，并从不同 CNN 层抽取特征构造相似度矩阵：使用 `relu4_2` 来表示节点特征，使用 `relu5_1` 来表示边特征，使用 `relu5_3` 来表示全局特征。实验中，VGG16 模块的学习率设置为 2×10^{-3} ，其他模块设置为 2×10^{-5} 。对于两幅图片，其中一个图结构采用德内三角剖分构建，另一个图采用全连接。NHGM 的三阶相似度参数设置为 $\sigma_3 = 10^{-4}$ 。

实验比较了通过机器学习学习构建图匹配相似度函数的方法：GMN^[16]，PCA-GM（3.2 节），LCS^[103]，BBGM^[82]。值得注意的是，BBGM^[82] 原论文中使用的实验设定比本文的实验设定更简单，因为他们过滤掉了超出边界框的关键点，而本文没有。对此，本文作者重新实现了 BBGM^[82] 以适应本文的实验设定，因此本文中报告的 BBGM 的精度略差于 BBGM 的原始论文。

表 3-11 中的结果表明，在将 CNN 特征和 QAP 求解器联合训练的帮助下，NGM 和 NHGM 在大多数类别上超过了其他方法，特别是在平均准确度方面表现最佳。与 3.2 节的 PCA-GM 方法相比，由于更换了更强的特征提取网络、并显式地构造了二阶相似度，NGM 的匹配精度更高。此外，NGM 在准确度方面超过了现有最先进的 BBGM 方法，这并不奇怪，因为 NGM 和 BBGM 共享相同的特

表 3-12 NGM 等模型在 Willow ObjectClass 数据集上的匹配精度 (%)
Table 3-12 NGM matching accuracy (%) on Willow ObjectClass dataset.

图匹配方法	图数目	car	duck	face	m-bike	w-bottle	平均精度
GMN ^[16]	2	67.9	76.7	99.8	69.2	83.1	79.3
PCA-GM (3.2节)	2	87.6	83.6	100.0	77.6	88.4	87.4
IPCA-GM (3.2节)	2	90.4	88.6	100.0	83.0	88.3	90.1
BBGM ^[82]	2	96.8	89.9	100.0	99.8	99.4	97.2
LCS ^[103]	2	91.2	86.2	100.0	99.4	97.9	94.9
NGM	2	97.4	93.4	100.0	98.6	98.3	97.5
NHGM	2	97.4	93.9	100.0	98.6	98.9	97.8
HiPPI ^[92]	≥40	74.0	88.0	100.0	84.0	95.0	88.2
MGM-Floyd ^[160]	32	85.0	79.3	100.0	84.3	93.1	88.3
NMGM	10	97.6	94.5	100.0	100.0	99.0	98.2

征提取器，而 NGM 同时学习了特征提取器和图匹配求解器，而 BBGM 只学习了特征提取器。NHGM 通过利用超图相似度进一步提升了 NGM 的性能。训练期间的显存消耗和运行速度如下：NGM (5707MB, 每秒 14.1 对图像)；NHGM (38060MB, 每秒 11.0 对图像)。NGM 与 PCA-GM (5165MB, 每秒 14.4 对图像) 的计算代价相当。

模型的泛化能力进一步通过混淆矩阵加以验证，如图 3-19 所示，仅使用横轴上的类别进行训练，并在所有类别上进行测试。单元格的色由当前单元格的精度除以同一列中的最高精度来确定。如混淆矩阵所示，NGM 在视觉相似的类别之间具有一定的泛化能力，例如“椅子”和“沙发”、“猫”和“狗”、“公交车”和“火车”。与其他深度图匹配方法相比，NGM 在混淆矩阵对角线上的类别上拟合得更好，但在考虑所有类别时，NGM 似乎性能不及 PCA-GM，可能是因为大多数类与训练类无关。匹配的可视化结果如图 3-20 所示（绿线为正确匹配，红线为错误匹配）。NGM 和 NHGM 的错误模式 (error pattern) 相似，但与 PCA-GM 不同，因为它们图像特征提取器不同。NHGM 通过纠正一些现有的错误，如“猫”、“摩托车”、“人”和“电视”，改进了 NGM 的结果。

在 Willow ObjectClass 数据集上的结果。遵循 3.2 节的实验设定，Willow ObjectClass 数据集中每个类的前 20 张图片被用于构造训练集，其余图片作为测试集。实验中的超参数设定和 Pascal VOC 关键点数据集保持一致。本实验与 HARG-SSVM^[101]、GMN^[16] 和 PCA-GM (3.2 节) 进行了比较，并将两种最新的多图匹配算法 HiPPI^[92] 和 MGM-Floyd^[160] 视为非学习的基准算法。表 3-12 展示了 NGM 和 NHGM 在刚性物体上的优异表现，性能超过了最先进的 BBGM^[82]。表 3-12 的“图数目”代表同时用于协同匹配的图数目。与非学习的多图匹配求解方法 HiPPI^[92] 和 MGM-Floyd^[160] 相比，NMGM 使用的图数量更少，但得益于端到端的多图匹配学习，达到了更高的匹配精度。NMGM 在所有方法中表现最佳。

在 Pascal VOC 关键点数据集上的消融实验。通过对一项对照实验，本文验证了在整个图匹配和 QAP 学习框架中引入机器学习的优势。为了公平比较，本实



图 3-20 *Pascal VOC* 关键点数据集中 12 个类匹配结果的可视化。
Figure 3-20 Visualization of 12 categories in *Pascal VOC* Keypoint dataset.

验采用了无 SplineCov 的标准 VGG16 模型^[128]。在表 3-13 中，第一列表示 CNN 权重是来自预训练的 ImageNet 分类任务 (ImageNet)^[130]还是来自于学习后的图匹配模型 (NGM)；第二列表示是采用非学习求解器 RRWM 还是学习型求解器 NGM 来求解 QAP 问题。本实验验证了在 CNN 和求解器上的联合学习的必要性，其中同时学习 CNN 和求解器的方法表现最佳。CNN 的学习显然可以弥补分类任务和匹配任务之间的差异，但值得注意的是，学习的求解器在使用 ImageNet CNN 时几乎比 RRWM 准确率高出一倍 (44.0% vs 24.0%)，这表明机器学习求解器学到了可泛化的求解策略。

在 *Pascal VOC* 关键点数据集上，作者对 NHGM 中提出的模块进行了消融实验，如表 3-14 所示。基准模型是按照 NGM 构建的，但在模型输入中忽略了节点相似度，即 $\mathbf{v}_{ia}^{(0)} = 1$ ，并排除了 Sinkhorn 嵌入。通过依次添加这些组件，验证了节点相似度、Sinkhorn 嵌入以及 SplineConv 特征和加权内积相似度的有效性。最后，为 NHGM 添加了超图相似度。表 3-14 显示，SplineConv 特征对匹配准确性有重要贡献，并且特征提取的网络可以任意修改，因为本节的方法处理了最一般

表 3-13 将 *NGM* 的 *CNN*、图匹配求解器模块替换成非学习的 *ImageNet CNN* 和 *RRWM* 的对比试验。

Table 3-13 *Controlled experiment by replacing NGM with unlearned ImageNet CNN and RRWM solver.*

CNN	求解器	✈️	🚲	🦅	🐼	🦊	🚗	🚘	🐱	🐶	🐻	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	🐘	mean		
ImgNet	RRWM	16.1	22.3	20.8	21.8	21.3	31.0	23.2	25.4	18.6	20.5	20.6	21.7	18.8	21.9	13.5	28.6	21.7	18.3	50.5	42.8	24.0											
ImgNet	NGM	30.8	42.5	44.3	33.8	39.8	52.2	49.2	53.9	27.5	42.4	29.3	49.1	45.1	45.1	24.0	48.3	49.9	29.9	70.2	73.3	44.0											
NGM	RRWM	41.5	54.7	54.3	50.3	67.9	74.3	70.3	60.6	42.3	59.1	48.1	57.3	59.1	56.2	40.6	69.6	63.1	52.2	76.3	87.8	59.3											
NGM	NGM	50.1	63.5	57.9	53.4	79.8	77.1	73.6	68.2	41.1	66.4	40.8	60.3	61.9	63.5	45.6	77.1	69.3	65.5	79.2	88.2	64.1											

表 3-14 在 *Pascal VOC* 关键点数据集上对 *NHGM* 的消融实验。

Table 3-14 *Ablation study of NHGM on Pascal VOC Keypoint dataset.*

模型	匹配精度	相对精度
基准 <i>NGM</i> 模型	58.7%	
+ 使用节点相似度作为顶点特征	59.6%	+0.9%
+ Sinkhorn 图嵌入	64.1%	+4.5%
+ SplineConv 特征	74.3%	+10.2%
+ 带权内积的相似度函数	80.1%	+5.8%
+ 超图相似度	80.4%	+0.3%

形式的 QAP。

3.4 面向基数约束求解的神经网络设计与理论分析

拥有单次前馈计算 (one-shot)、非自回归 (non-autoregressive) 特点的神经网络已经广泛应用于求解组合优化问题 (例如文献^[64]、3.2和3.3节), 并可以面向目标函数进行自监督训练 (可训练)。在过去的研究中, 这些方法还展现出了效率上的优越性 (可并行) 和处理非确定性决策问题的潜力 (可微分)。在端到端的神经网络求解器中, 如何处理离散约束是一个常见挑战, 对此目前存在三种典型的处理方法:

1. 在目标函数中添加惩罚项。但是, 这种做法无法保证有限程度的约束违反, 这限制了神经网络在许多约束敏感场景的应用;
2. 扰动输入并以黑盒的方式生成近似梯度。尽管约束条件可以被严格遵守, 但近似的梯度可能会限制算法的性能;
3. 作为折衷方案, 开发柔性算法, 从而使神经网络的输出满足松弛后的约束。但是, 这种做法仍然可能任意程度地违反约束。

面向终极目标——实现一个能够任意程度保持约束的神经网络求解器, 本节关注了一个常见的排列型组合优化问题: 基数约束问题。基数约束实际上可以通过可微分的最优传输 (optimal transport, OT) 层进行处理。基于这个观察, 本节提

表 3-15 不同组合优化求解网络的理论、实验性能比较。

Table 3-15 Theoretical and empirical comparison among CO networks.

神经网络求解器名称	Erdos Goes Neural ^[64]	CardNN-S	CardNN-GS/HGS
在网络结构中添加约束	否(添加惩罚项)	是(通过 Sinkhorn)	是(通过 Gumbel-Sinkhorn)
约束违反的理论上界	无法控制	$\tilde{O}\left(\frac{m\tau}{ \phi_k - \phi_{k+1} }\right)$	$\tilde{O}\left(\frac{m\tau(\phi_i - \phi_j + \sigma)}{ \phi_i - \phi_j ^2 + \sigma^2}\right)_{\forall i \neq j}$
实验中的约束违反	8.44	6.71	0.09
实验中与最优解的 gap (↓)	0.152	0.139	0.023

出了基于 OT 的基数约束神经网络用于端到端的组合优化学习。本节研究了两种变体：分别使用 Sinkhorn 和 Gumbel-Sinkhorn 处理 OT。基于本节的理论推导，它们对约束的违反程度可以被精确地描述和调控。在纯优化问题中，本节的方法与商用求解器性能相当甚至更好；此外，本节还展示了纯机器学习求解器应用于联合预测-优化任务的可行性。

3.4.1 基数约束优化问题描述

基数约束的优化问题 (cardinality-constrained optimization) 中，解的非零元素必须小于等于 k 个：

$$\min_{\mathbf{x}} J(\mathbf{x}|\omega) \quad s.t. \quad \|\mathbf{x}\|_0 \leq k. \quad (3.46)$$

基数约束优化的应用场景非常广泛，例如求解运筹优化领域的设施选址问题^[109]，发现社交网络中最具影响力的种子用户^[110]以及构造运营成本可控的投资组合^[111]。在基数约束下，研究者旨在寻找大小为 k 的最优子集。如公式 (1.5) 所描述的，基数约束优化可以被等价地刻画为排列型组合优化中的“前 k 个排列”问题，即寻找并确定 k 个非零元素。非零元素确定后，即可采用数值优化方法对非零的决策变量进行赋值。下文的研究内容进一步印证了基数约束优化和其他排列型组合优化问题的关联性——可以采用类似的方法论（基于 Sinkhorn 算法），构造端到端的神经网络进行求解。

3.4.2 研究动机

最近的一类组合优化神经网络^[64,161-162]通过单次前馈计算和松弛不可微约束，改进了现有的基于强化学习的自回归网络^[84,105]，形成了一种端到端的学习框架。单次前馈网络的优势体现在三个方面：1) 相较于基于 CPU 的传统求解器^[30]和繁琐的自回归网络，在 GPU 上运行的单次前馈网络效率更高；2) 通过直接优化目标函数实现自监督学习，避免了监督学习^[25]的局限性和强化学习^[60]的复杂性；3) 端到端的架构使得求解预测性优化问题成为可能，即求解不确定条件下的决策问题^[76,163]。本节遵循了文献^[64]提出的单次前馈求解范式，问题编

码网络接受问题的原始数据并预测决策变量。然后，将决策变量传递到可微的公式中，估算目标函数，并最终将目标函数视为自监督损失。所有模块都必须支持微分以进行端到端学习。作为组合优化的神经网络求解器，网络输出应当满足问题的约束条件，同时保留梯度。但由于组合问题的输入-输出映射通常是分段常数 (piece-wise constant) 形式，实际梯度在多数情况为 0，在输出改变时为无穷大，因此在神经网络中处理组合约束非常困难。

与其他组合问题的约束类似，基数约束也难以进行微分。现有的神经网络方法在处理基数约束时遇到了理论和技术上的瓶颈，如表3-15所示⁸。本节提出通过对编码网络输出的概率分布进行前 k 个排列，进而将基数约束编码到神经网络中。一种直观的做法是对所有概率进行排序并选择前 k 个，但这种方法得到的近似梯度缺乏足够的信息量。受 Cuturi 等^[116]、Xie 等^[81]研究工作的启发，本节将前 k 个选择 (topk selection) 重新表述为最优传输 (optimal transport, OT) 问题^[164]，并通过可微的 Sinkhorn 算法^[165]高效求解。最后，通过自监督损失，本节提出了一种组合优化神经网络，其输出具有柔性的基数约束，并支持端到端学习。

然而，对 Sinkhorn 算法的理论分析表明，如果第 k 个和 $(k+1)$ 个概率的值太过接近，它对基数约束的违反可能会显著增长。作者注意到基于扰动的近似可微方法^[72,74-75]和 Gumbel 方法^[149,166-167]在构造输出接近离散的神经网络上已经取得了成功，在本节中，作者进一步结合了 Gumbel 技巧进而使神经网络更严格地满足约束。网络兼顾了柔性算法^[16]的高效率和扰动方法^[72,166]的约束保持能力。本节还开发了一种同伦 (homotopy) 方法^[168]，在求解过程中逐步加强约束。在文献^[64]的自监督学习框架下，本节在两个确定性的排列型组合优化问题（设施选址问题和最大覆盖问题）上验证了网络的有效性。

此外，本节还研究了一个重要应用——预测性组合优化，其中问题参数在决策时是未知的。本节提出了一个“联合预测-优化”网络，该网络同时学习预测器和神经网络求解器，通过最终的目标函数进行端到端的学习，而不是两阶段的“先预测、后优化”方法。若是先学习预测器、然后单独进行优化，则存在预测误差影响优化性能的风险。具体而言，针对一个重要的实际任务：不确定条件下的投资组合优化，本节构建了一个端到端的预测性投资组合优化模型。在真实数据上的实验结果表明，它优于经典的“先预测、后优化”范式。

具体而言，本节的研究动机来自如下几个方面：

1) 一种单次前馈求解基数优化的神经网络架构。本节提出了首个端到端的基数约束神经网络，通过单次前馈网络高效地求解带有基数约束的排列型组合优化问题，即将约束条件纳入网络架构中，而非将其作为惩罚项直接加入损失函数。

2) 具有理论优势的神经网络求解器。基于可微分的 Topk 选择技术^[81]，基

⁸表3-15的理论结果忽略了对数项、实验结果来自图 23(a)对应的实验。

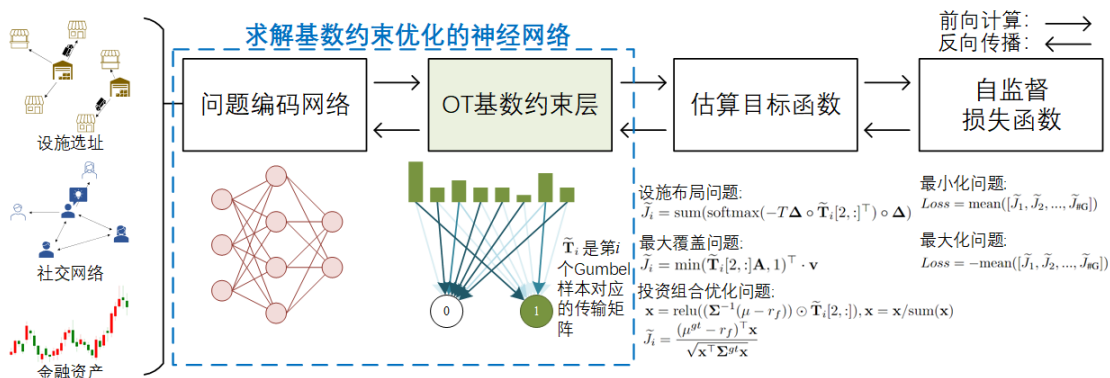


图 3-21 通过自监督学习求解基数约束优化的 CardNN 框架概览。

Figure 3-21 Overview of CardNN pipeline tackling cardinality-constrained optimization via self-supervised learning.

数约束通过可微的最优传输层编码在神经网络中。通过进一步引入黑盒微分方法^[72,75]中采用的扰动方法，利用 Gumbel 技术加强约束满足，本节的理论结果保证了神经网络求解器的约束满足性可被任意程度地调控。

3) 实现“联合预测-优化”范式。与传统的“先预测、后优化”框架相比，本节可微分的神经网络求解器支持一个端到端的“联合预测-优化”框架。通过在真实世界的金融资产数据上开展预测性投资组合优化研究，作者展示了端到端“联合预测-优化”框架的应用潜力。与“先预测、后优化”的基准模型 LSTM+Gurobi 相比，“联合预测-优化”模型的夏普比率从 1.1 提高到了 2.0。

3.4.3 算法描述和理论分析

本节提出的 CardNN 方法如图 3-21 所示。遵循单次前馈求解组合优化问题的通用范式^[64]，本节在神经网络求解器中引入了一个基于最优传输的基数约束层，使问题编码网络输出满足约束条件。这种在网络层中显式添加约束的方法在理论和实践上均具有优势。特别地，问题编码网络和本节主要讨论的 OT 基数约束层组成了求解器网络，随后采用神经网络输出的决策变量来估计目标函数（等价于自监督损失）。在本节中，使用可微分的网络求解最优传输问题^[114]，可实现对神经网络添加基数约束。设 $\mathbf{s} = [s_1, \dots, s_m]$ 为编码网络预测的概率，最优传输层通过将 k 个元素移到一个目的地（选中）同时将其他 $(m - k)$ 个元素移到另一个目的地（未选中），实现对 \mathbf{s} 中前 k 个元素的选取。在接下来的部分，作者将介绍两种最优传输层的实现方式和理论分析。

CardNN-S: 处理基数约束的 Sinkhorn 网络。面向基数约束问题的求解，最优传输问题定义如下。最优传输的源是 m 个候选概率，目的地是 \mathbf{s} 的最小/最大值。使用最优传输处理基数约束，等价于将前 k 个概率移动到 s_{\max} ，其余概率移

动到 s_{\min} 。最优传输的边缘分布 (\mathbf{c}, \mathbf{r}) 和距离矩阵 (\mathbf{D}) 定义如下：

$$\mathbf{c} = \underbrace{[1 \quad 1 \quad \dots \quad 1]}_{m \text{ 个概率}}, \mathbf{r} = \begin{bmatrix} m-k \\ k \end{bmatrix}, \mathbf{D} = \begin{bmatrix} s_1 - s_{\min} & s_2 - s_{\min} & \dots & s_m - s_{\min} \\ s_{\max} - s_1 & s_{\max} - s_2 & \dots & s_{\max} - s_m \end{bmatrix}. \quad (3.47)$$

对应的最优传输问题是一个线性整数规划问题，具有如下形式：

$$\min_{\mathbf{T}} \text{tr}(\mathbf{T}^\top \mathbf{D}) \quad s.t. \quad \mathbf{T} \in \{0, 1\}^{2 \times m}, \mathbf{T}\mathbf{1} = \mathbf{r}, \mathbf{T}^\top \mathbf{1} = \mathbf{c}, \quad (3.48)$$

其中， \mathbf{T} 是传输矩阵，也代表了面向基数约束的一个可行解（即前 k 个排列矩阵）。公式 (3.48) 的最优解 \mathbf{T}^* 等价于首先对所有概率进行排序，然后选择前 k 项。但是，上述方法会遇到难以微分的问题。为了利用柔性算法求梯度，本节将 \mathbf{T} 的排列约束放松至连续的 $[0, 1]$ 值，并在公式中引入熵正则项。公式 (3.48) 变换为：

$$\min_{\mathbf{T}^\tau} \text{tr}(\mathbf{T}^{\tau \top} \mathbf{D}) + \tau h(\mathbf{T}^\tau) \quad s.t. \quad \mathbf{T}^\tau \in [0, 1]^{2 \times m}, \mathbf{T}^\tau \mathbf{1} = \mathbf{r}, \mathbf{T}^{\tau \top} \mathbf{1} = \mathbf{c}, \quad (3.49)$$

其中 $h(\mathbf{T}^\tau) = \sum_{i,j} \mathbf{T}_{ij}^\tau \log \mathbf{T}_{ij}^\tau$ 是熵正则项^[114]。对任何实数矩阵 \mathbf{D} ，可以按如下步骤求解公式 (3.49)：首先施加熵正则项 $\mathbf{T}^\tau = \exp(-\mathbf{D}/\tau)$ 。随后，对 \mathbf{T}^τ 依次进行行归一化、列归一化⁹，

$$\mathbf{D}_r = \text{diag}(\mathbf{T}^\tau \mathbf{1} \oslash \mathbf{r}), \mathbf{T}^\tau = \mathbf{D}_r^{-1} \mathbf{T}^\tau; \quad \mathbf{D}_c = \text{diag}(\mathbf{T}^{\tau \top} \mathbf{1} \oslash \mathbf{c}), \mathbf{T}^\tau = \mathbf{T}^\tau \mathbf{D}_c^{-1}, \quad (3.50)$$

其中 \oslash 是逐元素除法。本节中， $\mathbf{T}^{\tau*}$ 表示上式收敛的结果，它是公式 (3.49) 的最优解。 $\mathbf{T}^{\tau*}$ 的第二行被视为面向基数约束的松弛后的解： $\mathbf{T}^{\tau*}[2, i]$ 代表 x_i 是非零的概率。接下来， $\mathbf{T}^{\tau*}$ 将被用于估计目标函数。请注意， $\mathbf{T}^{\tau*}$ 是一个松弛解、通常在原问题中是不可行的，需要定义约束违反的程度来衡量 $\mathbf{T}^{\tau*}$ 的质量。

定义 3.1 (约束违反量, Constraint Violation, 以下简称为 CV). CV 代表了松弛解 \mathbf{t} (来自分布 \mathcal{T}) 和任意一个可行解 (来自可行解集合 \mathcal{H}) 之间最小距离的期望： $CV = \mathbb{E}_{\mathbf{t} \in \mathcal{T}} [\min_{\mathbf{h} \in \mathcal{H}} \|\mathbf{t} - \mathbf{h}\|_F]$. 显然， \mathbf{h} 是距离 \mathbf{t} 最近的可行解。

注记 3.2 (CV 的意义). 以自监督的 CardNN-S 为例，在训练过程中需要基于 $\mathbf{T}^{\tau*}$ 估计目标函数，作为自监督的损失函数。在测试过程中，解必须在原问题中可行。天然地，可以返回最近的可行解 \mathbf{T}^* 。实际上，在训练中，神经网络学习的是一个松弛后、更容易的原始问题，而 $CV = \|\mathbf{T}^* - \mathbf{T}^{\tau*}\|_F$ 是一个重要的指标，用于衡量训练中的松弛问题和测试中的原始问题之间的差距。此处 \mathcal{T} 表示所有 CardNN-S 输出的分布，为简洁起见已省略。这种 CV 的意义同样适用于其他自监督的组合

⁹该 Sinkhorn 算法是公式 (3.14) 的一般形式，将边缘分布从全 1 向量扩展到了任意的非负向量。

算法 3-4 CardNN-GS: 处理基数约束的 Gumbel-Sinkhorn 网络

输入: 具有 m 个元素的向量 \mathbf{s} ; 基数 k ; Sinkhorn 参数 τ ; 噪声参数 σ ; 样本数目 $\#G$.

- 1: **for** $i \in \{1, 2, \dots, \#G\}$ **do**
- 2: 对于所有 s_j , 有 $\tilde{s}_j = s_j - \sigma \log(-\log(u_j))$, 其中 u_j 采样自 $\mathcal{U}(0, 1)$;
- 3: $\tilde{\mathbf{D}} = \begin{bmatrix} \tilde{s}_1 - s_{\min} & \dots & \tilde{s}_m - s_{\min} \\ s_{\max} - \tilde{s}_1 & \dots & s_{\max} - \tilde{s}_m \end{bmatrix}$;
- 4: 遵循公式 (3.47) 构造 \mathbf{c}, \mathbf{r} ;
- 5: $\tilde{\mathbf{T}}_i = \exp(-\tilde{\mathbf{D}}/\tau)$;
- 6: **while not converged do**
- 7: $\tilde{\mathbf{D}}_r = \text{diag}(\tilde{\mathbf{T}}_i \mathbf{1} \odot \mathbf{r})$; $\tilde{\mathbf{T}}_i = \tilde{\mathbf{D}}_r^{-1} \tilde{\mathbf{T}}_i$;
- 8: $\tilde{\mathbf{D}}_c = \text{diag}(\tilde{\mathbf{T}}_i^\top \mathbf{1} \odot \mathbf{c})$; $\tilde{\mathbf{T}}_i = \tilde{\mathbf{T}}_i \tilde{\mathbf{D}}_c^{-1}$;
- 9: **end while**
- 10: **end for**

输出: $\#G$ 个传输矩阵 $[\tilde{\mathbf{T}}_1, \tilde{\mathbf{T}}_2, \dots, \tilde{\mathbf{T}}_{\#G}]$.

优化神经网络。接下来, 作者将在理论上刻画 CardNN-S 的 CV:

命题 3.3. 假设 Sinkhorn 算法已收敛。CardNN-S 的约束违反量是

$$CV_{\text{CardNN-S}} = \|\mathbf{T}^* - \mathbf{T}^{\tau*}\|_F \leq \frac{2m\tau \log 2}{|\phi_k - \phi_{k+1}|}. \quad (3.51)$$

不失一般性, ϕ 代表 \mathbf{s} 的降序排列, 即 ϕ_k 和 ϕ_{k+1} 分别是 \mathbf{s} 中第 k 大和第 $(k+1)$ 大的元素。命题 3.3 是基于论文^[81]中定理 2 的简单扩展, 是一个可调控的 CV, 比 Karalias 等论文^[64]中完全不受控的 CV 更好。但是, 从式 (3.51) 中可以看出, 如果 $|\phi_k - \phi_{k+1}|$ 变小, CardNN-S 的 CV 会逐渐增长, 并在 $\phi_k = \phi_{k+1}$ 的极端情况下变为发散——在这种极端情况下, 它的 CV 不能通过调整超参数 τ 来减小。出现这种发散并不意外, 因为如果 $\phi_k = \phi_{k+1}$, 显然无法决定应该选择它们之中的哪个。实际上, 如果存在直接作用于 $\mathbf{T}^{\tau*}$ 的监督信号, 这种发散并非不能接受。但是, 正如注记 3.2 所讨论的, 一个可调控的 CV 在自监督组合优化网络中的重要性不容忽视。由于仅使用 Sinkhorn 方法达到了其理论瓶颈, 在下文中, 作者提出了改进版本, 通过引入随机扰动^[72,166]来进一步减小 CV。

CardNN-GS: 处理基数约束的 Gumbel-Sinkhorn 网络。为了突破 CardNN-S 方法的理论瓶颈, 本节提出了基于 Gumbel-Sinkhorn 的基数约束网络层, 如算法 3-4 所示¹⁰。接下来, 作者将从理论角度分析它的 CV。本文采用了重参数技巧^[166] (reparameterization trick) 对神经网络预测的概率添加随机变量, 可以规避掉不可微分的采样过程。Gumbel 分布可以写作

$$g_\sigma(u) = -\sigma \log(-\log(u)), \quad (3.52)$$

其中 σ 控制了方差, u 从均匀分布 $\mathcal{U}(0, 1)$ 中采样得到。对应地, \mathbf{s} 和 \mathbf{D} 需要更

¹⁰基于 Sinkhorn 的 CardNN-S 算法只需将 $\#G$ 设为 1, σ 设为 0。

新为如下形式：

$$\tilde{s}_j = s_j + g_\sigma(u_j), \quad \tilde{\mathbf{D}} = \begin{bmatrix} \tilde{s}_1 - s_{\min} & \tilde{s}_2 - s_{\min} & \cdots & \tilde{s}_m - s_{\min} \\ s_{\max} - \tilde{s}_1 & s_{\max} - \tilde{s}_2 & \cdots & s_{\max} - \tilde{s}_m \end{bmatrix}. \quad (3.53)$$

对于在 Gumbel 噪声下的最优传输问题，其线性整数规划的形式为

$$\min_{\mathbf{T}^\sigma} \text{tr}(\mathbf{T}^{\sigma\top} \tilde{\mathbf{D}}) \quad s.t. \quad \mathbf{T}^\sigma \in \{0, 1\}^{2 \times m}, \mathbf{T}^\sigma \mathbf{1} = \mathbf{r}, \mathbf{T}^{\sigma\top} \mathbf{1} = \mathbf{c}, \quad (3.54)$$

其中，公式 (3.54) 的最优解写作 $\mathbf{T}^{\sigma*}$ 。与 CardNN-S 类似，为了使线性整数规划适用于基于梯度的深度学习方法，作者放松了整数约束，并添加了熵正则化项：

$$\min_{\tilde{\mathbf{T}}} \text{tr}(\tilde{\mathbf{T}}^\top \tilde{\mathbf{D}}) + h(\tilde{\mathbf{T}}) \quad s.t. \quad \tilde{\mathbf{T}} \in [0, 1]^{2 \times m}, \tilde{\mathbf{T}} \mathbf{1} = \mathbf{r}, \tilde{\mathbf{T}}^\top \mathbf{1} = \mathbf{c}, \quad (3.55)$$

上式可以用公式 (3.50) 中介绍的 Sinkhorn 算法进行求解。在这里，公式 (3.55) 的最优解表示为 $\tilde{\mathbf{T}}^*$ 。由于 $\mathbf{T}^{\sigma*}$ 是最接近 $\tilde{\mathbf{T}}^*$ 的可行解，CardNN-GS 的 CV 即为 $\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F$ 的期望。在实践中，可以并行地生成多个 $\tilde{\mathbf{T}}$ 以克服随机性（根据上文中的定义， ϕ 是 \mathbf{s} 的降序排列版本）：

命题 3.4. 在至少 $(1 - \varepsilon)$ 的概率下，CardNN-GS 的约束违反量是

$$CV_{CardNN-GS} = \mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq (\log 2) m \tau \sum_{i \neq j} \Omega(\phi_i, \phi_j, \sigma, \varepsilon), \quad (3.56)$$

$$\text{其中 } \Omega(\phi_i, \phi_j, \sigma, \varepsilon) = \frac{2\sigma \log \left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1 - \varepsilon)} \right) + |\phi_i - \phi_j| \left(\frac{\pi}{2} + \arctan \frac{\phi_i - \phi_j}{2\sigma} \right)}{(1 - \varepsilon)((\phi_i - \phi_j)^2 + 4\sigma^2)(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})}.$$

证明思路：该命题的证明可通过对命题3.3的结论进行泛化来实现。定义 $\phi_{\pi_k}, \phi_{\pi_{k+1}}$ 是经过 Gumbel 噪声扰动后第 k 大、第 $(k+1)$ 大的元素，CV 的上界等价于推导 $\mathbb{E}_u [1/(|\phi_{\pi_k} + g_\sigma(u_{\pi_k}) - \phi_{\pi_{k+1}} - g_\sigma(u_{\pi_{k+1}})|)]$ 的上界，其中 $g_\sigma(u_{\pi_k}) - g_\sigma(u_{\pi_{k+1}})$ 的概率密度函数具有上界 $f(y) = 1/(y^2 + 4)$ 。随后，CV 的上界即可通过积分得到。详细的推导步骤可见附录A.1。□

推论 3.5. 为了简化，省略公式中的对数项（详细证明见附录A.2），

$$CV_{CardNN-GS} \leq \tilde{O} \left(\frac{m\tau(|\phi_i - \phi_j| + \sigma)}{|\phi_i - \phi_j|^2 + \sigma^2} \right)_{\forall i \neq j}. \quad (3.57)$$

通过图 3-22 中的简单示例可以直观地比较 CardNN-S 和 CardNN-GS 的差异：找出 $[1.0, 0.8, 0.601, 0.6, 0.4, 0.2]$ 的前 3 个值，图 3-22 记录了不同的 τ, σ 值对应的

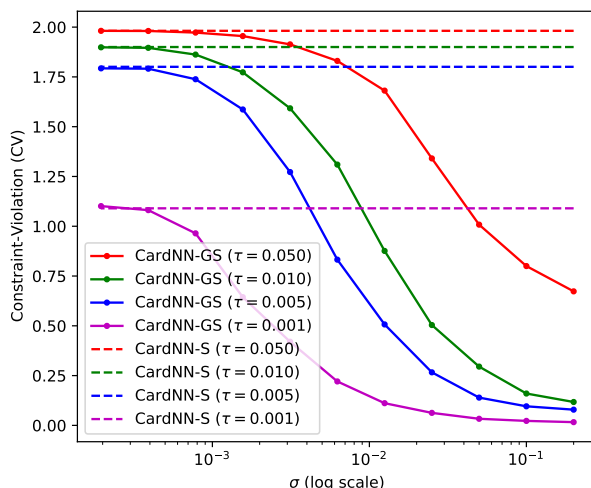


图 3-22 对比 Sinkhorn 和 Gumbel-Sinkhorn 约束违反量的简单示例。

Figure 3-22 Toy example comparing the CVs between Sinkhorn and Gumbel-Sinkhorn.

CV。对于 CardNN-GS，更大的 σ 和更小的 τ 可以收紧 CV；而对 CardNN-S 而言，其 CV 值普遍更高、且只能通过 τ 来调控。该结果与命题 3.3 和命题 3.4 的结论相符合。

同伦版本的 Gumbel-Sinkhorn。推论 3.5 表明，通过调整 τ 和 σ 可以收紧 CV，这促使作者研究了 Gumbel-Sinkhorn 的同伦 (homotopy) 方法^[168-169]，其中约束条件逐渐收紧 (即逐渐降低 τ 和 σ 的值)。在实践中，作者未考虑调整 σ ，因为较大的 σ 意味着增加了方差，需要更多的 Gumbel 样本，带来了额外的计算开销。同伦版本的神经网络求解器被命名为 CardNN-HGS。

最后，CardNN-S 和 CardNN-GS 在理论上可以统一：

推论 3.6. 当 $\sigma \rightarrow 0^+$ 时，CardNN-S 是 CardNN-GS 的一种特殊情况 (详细证明见附录 A.3)。

3.4.4 具体实现和实验分析

确定性排列型组合优化实验。首先讨论通过机器学习，以单次前馈的形式求解两个确定性的排列型组合优化问题：设施选址问题 (facility location problem, FLP) 和最大覆盖问题 (max covering problem, MCP)。确定性优化意味着所有问题参数在决策时都是已知的。

设施选址问题。考虑如下问题：给定 m 个地点，我们要在其中新增 k 个设施 (例如仓储)，为其余的邻近地点提供服务^[109]。FLP 问题的目标函数是最小化每个地点和其最近设施之间距离的总和。使用 $\Delta \in \mathbb{R}_{\geq 0}^{m \times m}$ 表示地点之间的距离矩

算法 3-5 CardNN-GS/HGS 用于求解设施选址问题

输入: 距离矩阵 Δ ; 学习率 α ; softmax 温度 β ; CardNN-GS 参数 $k, \tau, \sigma, \#G$.

- 1: **if** 训练阶段 **then**
- 2: 随机初始化神经网络权重 θ ;
- 3: **end if**
- 4: **if** 测试阶段 **then**
- 5: 加载预训练的神经网络权重 θ ; $J_{best} = +\infty$;
- 6: **end if**
- 7: **while** not converged **do**
- 8: $\mathbf{s} = \text{SplineCNN}_{\theta}(\Delta)$;
- 9: $[\tilde{\mathbf{T}}_1, \tilde{\mathbf{T}}_2, \dots, \tilde{\mathbf{T}}_{\#G}] = \text{CardNN-GS}(\mathbf{s}, k, \tau, \sigma, \#G)$;
- 10: 对所有的 i , 计算 $\tilde{J}_i = \text{sum}(\text{softmax}(-\beta\Delta \circ \tilde{\mathbf{T}}_i[2, :]) \circ \Delta)$;
- 11: $J = \text{mean}([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}])$;
- 12: **if** 训练阶段 **then**
- 13: 基于梯度 $\frac{\partial J}{\partial \theta}$ 和学习率 α , 通过梯度下降更新 θ ;
- 14: **end if**
- 15: **if** 测试阶段 **then**
- 16: 基于梯度 $\frac{\partial J}{\partial \mathbf{s}}$ 和学习率 α , 通过梯度下降更新 \mathbf{s} ;
- 17: 对所有的 i , 计算 $\tilde{J}_i = \text{sum}(\min(\Delta \circ \text{TopK}(\tilde{\mathbf{T}}_i[2, :]^{\top}))$;
- 18: $J_{best} = \min([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}], J_{best})$;
- 19: **end if**
- 20: **end while**
- 21: **if** 同伦方法 **then**
- 22: 降低 τ 的值, 随后跳转至第7行;
- 23: **end if**

输出: 学习到的神经网络权重 θ (训练阶段) / 最优的目标函数 J_{best} (测试阶段).

阵, FLP 的数学形式是

$$\min_{\mathbf{x}} \sum_{j=1}^m \min(\{\Delta_{i,j} \mid \forall \mathbf{x}_i = 1\}) \quad s.t. \quad \mathbf{x} \in \{0, 1\}^m, \|\mathbf{x}\|_0 \leq k. \quad (3.58)$$

问题编码网络: 首先基于问题输入构建图结构, 对于具有二维坐标的地点, 如果两个地点间的距离小于阈值 (例如 0.02), 则添加一条边。实验中利用一个 3 层的 SplineConv^[147]网络, 从上述图结构中提取特征。

目标函数估计: 公式 (3.58) 中的 \min 运算符会导致梯度稀疏。因此, 采用带有负温度 $-\beta$ 的 softmax 替换 \min , 其中 $\mathbf{A} \circ \mathbf{b}$ 表示矩阵 \mathbf{A} 与向量 \mathbf{b} 平铺后的矩阵计算逐元素乘积:

$$\tilde{J}_i = \text{sum}(\text{softmax}(-\beta\Delta \circ \tilde{\mathbf{T}}_i[2, :]^{\top}) \circ \Delta), \quad J = \text{mean}([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}]). \quad (3.59)$$

算法3-5总结了用于求解 FLP 的 CardNN-GS/HGS 算法的训练、测试流程。CardNN-S 即为 $\tau = 0, \#G = 1$ 的特例。

算法 3-6 CardNN-GS/HGS 用于求解最大覆盖问题

输入: 二分图邻接矩阵 \mathbf{A} ; 物品价值 \mathbf{v} ; 学习率 α ; CardNN-GS 参数 $k, \tau, \sigma, \#G$.

```

1: if 训练阶段 then
2:   随机初始化神经网络权重  $\theta$ ;
3: end if
4: if 测试阶段 then
5:   加载预训练的神经网络权重  $\theta$ ;  $J_{best} = 0$ ;
6: end if
7: while not converged do
8:    $\mathbf{s} = \text{GraphSage}_{\theta}(\mathbf{A})$ ;
9:    $[\tilde{\mathbf{T}}_1, \tilde{\mathbf{T}}_2, \dots, \tilde{\mathbf{T}}_{\#G}] = \text{CardNN-GS}(\mathbf{s}, k, \tau, \sigma, \#G)$ ;
10:  对所有的  $i$ , 计算  $\tilde{J}_i = \min(\tilde{\mathbf{T}}_i[2, :]\mathbf{A}, 1)^{\top} \cdot \mathbf{v}$ ;
11:   $J = \text{mean}([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}])$ ;
12:  if 训练阶段 then
13:    基于梯度  $\frac{\partial J}{\partial \theta}$  和学习率  $\alpha$ , 通过梯度下降更新  $\theta$ ;
14:  end if
15:  if 测试阶段 then
16:    基于梯度  $\frac{\partial J}{\partial \mathbf{s}}$  和学习率  $\alpha$ , 通过梯度下降更新  $\mathbf{s}$ ;
17:    对所有的  $i$ , 计算  $\tilde{J}_i = (\text{TopK}(\tilde{\mathbf{T}}_i[2, :])\mathbf{A})^{\top} \cdot \mathbf{v}$ ;
18:     $J_{best} = \max([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}], J_{best})$ ;
19:  end if
20: end while
21: if 同伦方法 then
22:   降低  $\tau$  的值, 随后跳转至第7行;
23: end if

```

输出: 学习到的神经网络权重 θ (训练阶段) / 最优的目标函数 J_{best} (测试阶段).

最大覆盖问题。给定 m 个集合和 n 个物品, 每个集合可以覆盖任意数量的物品, 每个物品都与一个价值相关联。最大覆盖问题^[170] (max covering problem, MCP) 的目标是找到 k 个集合 ($k \ll m$), 使得覆盖物品的总价值最大。该问题反映了许多现实场景, 例如在社交网络中发现具有影响力的种子用户^[110]。作者为集合和物品建立一个二分图, 其中覆盖关系被编码为边。记 $\mathbf{v} \in \mathbb{R}^n$ 为价值, $\mathbf{A} \in \{0, 1\}^{m \times n}$ 为二分图的邻接矩阵, $\mathbb{I}(\mathbf{x})$ 为指示函数 (如果 $\mathbf{x}_i \geq 1$ 则 $\mathbb{I}(\mathbf{x})_i = 1$, 否则 $\mathbb{I}(\mathbf{x})_i = 0$)。MCP 的数学形式是

$$\max_{\mathbf{x}} \sum_{j=1}^n \left(\mathbb{I} \left(\sum_{i=1}^m \mathbf{x}_i \mathbf{A}_{ij} \right) \cdot \mathbf{v}_j \right) \quad s.t. \quad \mathbf{x} \in \{0, 1\}^m, \|\mathbf{x}\|_0 \leq k, \quad (3.60)$$

问题编码网络: 为了使用神经网络处理覆盖问题的二分图, 本节采用了三层的 GraphSage^[171] 网络和一个采用 Sigmoid 激活的全连接层。全连接层的输出即为选每个集合的概率。

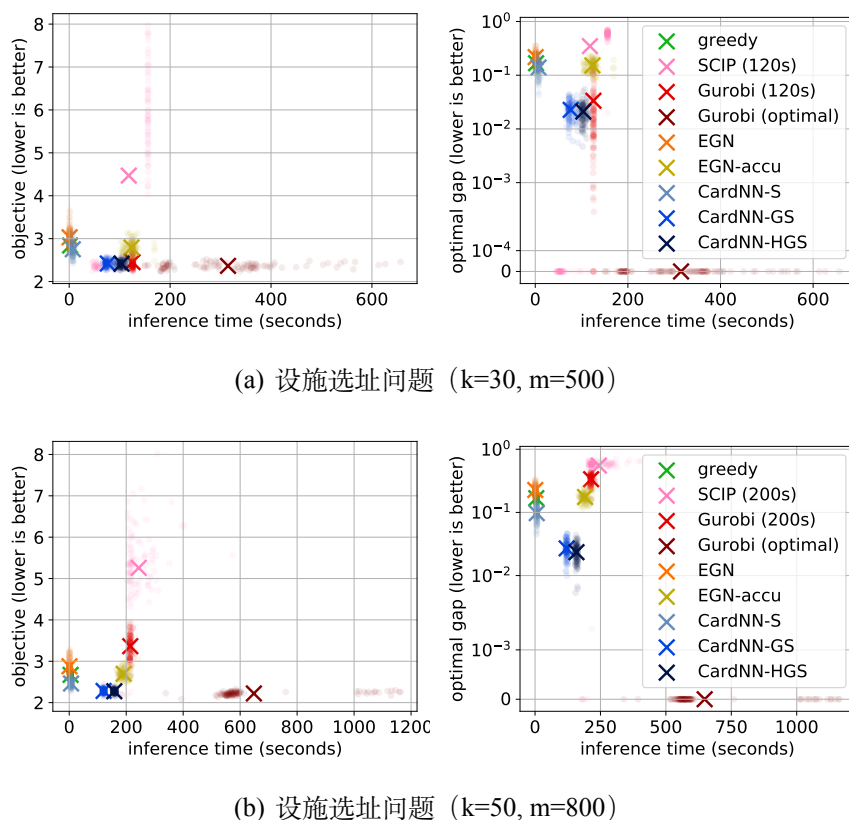


图 3-23 设施选址问题 (FLP) 的仿真实验结果。

Figure 3-23 Synthetic experiment results for facility location problem (FLP).

目标函数估计: 基于公式 (3.60), 目标函数的估计方法是

$$\tilde{J}_i = \min(\tilde{\mathbf{T}}_i[2, :]\mathbf{A}, 1)^\top \cdot \mathbf{v}, J = \text{mean}([\tilde{J}_1, \tilde{J}_2, \dots, \tilde{J}_{\#G}]). \quad (3.61)$$

算法3-6总结了用于求解 MCP 的 CardNN-GS/HGS 算法的训练、测试流程。CardNN-S 即为 $\tau = 0, \#G = 1$ 的特例。

学习和优化方法。鉴于优化目标是最小化或最大化, 可以分别将 J 或 $-J$ 作为自监督损失函数。本节使用 Adam 优化器^[155]进行训练。在测试阶段, \mathbf{s} 中的概率采用神经网络的预测初始化, 随后通过梯度更新 \mathbf{s} 进而最大化目标函数。

实验设定。纯优化问题的实验设定遵循了 Karalias 等人^[64]提出的自监督学习框架, 实验中考虑了仿真数据和真实数据。对于仿真数据, 本节构建了包含 100 个样本的独立训练集和测试集: 对于 FLP 问题, 在单位正方形上随机生成了坐标数据; 对于 MCP 问题, 作者模仿了 ORLIB^[172]的数据分布生成了仿真数据。由于缺乏大规模的真实数据集, 真实数据仅用于测试 (即在仿真数据上训练, 在真实数据上测试)。实验收集了上海、首尔、伦敦、纽约的星巴克门店位置 (166-569 家门店) 用于测试 FLP, 采用了 Rozemberczki 等^[173]收集的 6 个 Twitch 社交网络 (1912-9498 个节点) 用于测试 MCP。

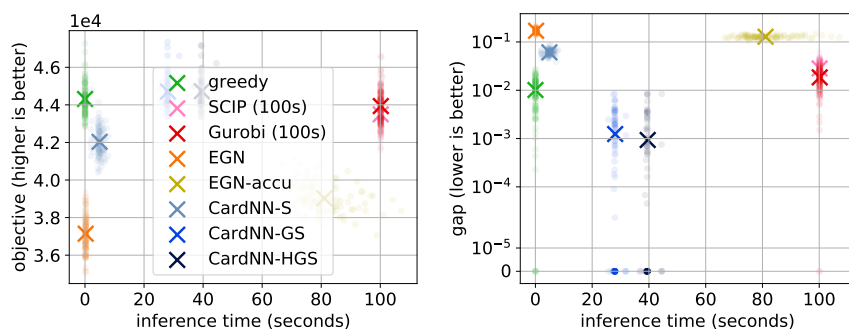
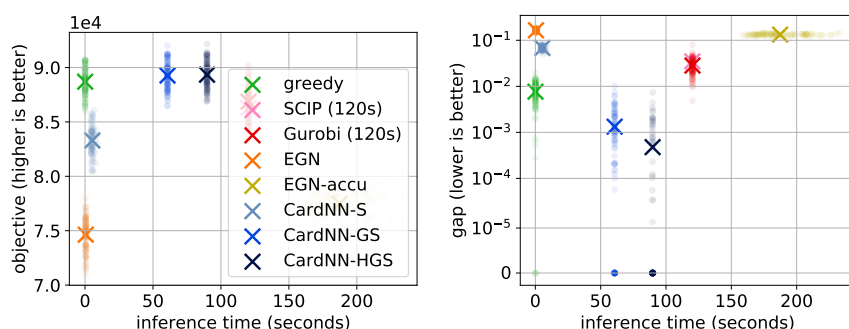
(a) 最大覆盖问题 ($k=50, m=500, n=1000$)(b) 最大覆盖问题 ($k=100, m=1000, n=2000$)

图 3-24 最大覆盖问题 (MCP) 的仿真实验结果。

Figure 3-24 Synthetic experiment results for max covering problem (MCP).

对比方法。1) 贪心算法是一类容易实现并且非常有效的算法。由于 FLP 和 MCP 问题的次模 (submodular) 性质^[112], 贪心算法在这类问题上拥有至少 $(1 - 1/e)$ 的近似比。2) 整数规划求解器包括最先进的商用求解器 Gurobi 9.0^[31]和最先进的开源求解器 SCIP 7.0^[30]。实验为它们设定的时间上限大于本节神经网络求解器。对于 3) 组合优化神经网络, 实验与最先进的 Erdos Goes Neural (EGN)^[64]进行比较, 该方法基于其官方实现¹¹修改以应用于基数约束优化。EGN 和本节方法的主要区别在于, EGN 没有在它的网络结构中添加约束条件。此外, 实验发现所有自监督学习方法的训练过程在几十分钟内就可以收敛。由于强化学习组合优化方法^[54,80]需要更多的训练时间, 因此实验没有与强化学习方法进行比较。

评价指标和实验结果。本节主要采用“gap”指标:

$$\text{gap} = \frac{|J - J^*|}{\max(J, J^*)} \quad (3.62)$$

其中 J 是预测的目标函数, J^* 是所有方法中最优的目标函数。如果其中一个整数规划求解器证明了解的最优性, 则称其为“optimal gap”。图 3-23 和图 3-24 分别报告了仿真数据集上的 FLP 和 MCP 结果, 其中包括目标函数、gap 与计算时

¹¹https://github.com/Stalence/erdos_neu

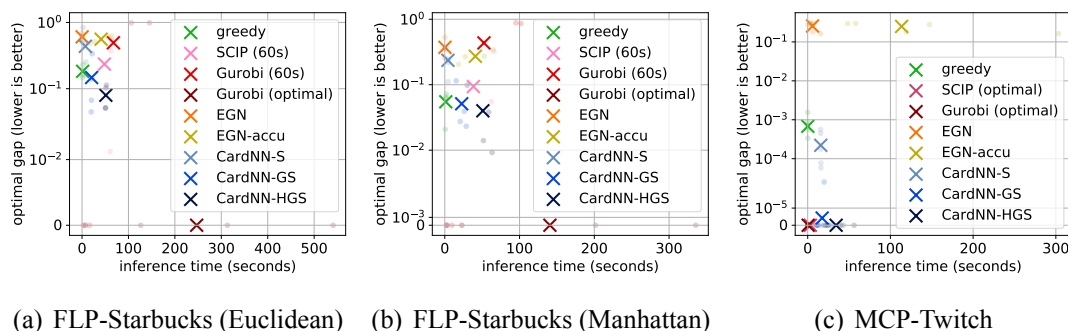


图 3-25 FLP、MCP 问题的真实数据实验结果。

Figure 3-25 Real-world experiment results for FLP and MCP.

间的关系。每个散点表示一个问题实例，平均性能由“×”标出。在效率和效果方面，本节的 CardNN-S 网络优于现有的 EGN 网络，因为 EGN 对约束的违反不可调控。CardNN-GS 和 CardNN-HGS 进一步提升了求解效果，甚至超过了最先进的商用求解器 Gurobi（花费更少的计算时间并获得更好的结果）。在 MCP 的仿真实验中，Gurobi 求解器未能在 24 小时内返回最优解，因此没有报告其结果。图 3-25 总结了真实数据集上的结果。在真实数据中，本节的 CardNN 模型性能始终优于 EGN，并且与最先进的 SCIP/Gurobi 求解器相当，部分场景甚至可以超越传统求解器。在 FLP-Starbucks 问题中，CardNN-GS/HGS 花费了相近的预测时间，达到了更低的“gap”；在 MCP-Twitch 问题中，CardNN-HGS 比 SCIP/Gurobi 更慢，但找到了所有最优解。

同时考虑效率和效果，组合优化神经网络的性能排名是 CardNN-HGS > CardNN-GS > CardNN-S > EGN。这与第 3.4.3 节中的理论结果一致：更好地保持约束会使得神经网络求解器拥有更好的性能。此外，可微分的 topk 求解并非只有一种技术路径，为了验证选择 Xie 等人^[81]的 Sinkhorn 技术作为基础的合理性，作者还比较了其他可微分的 topk 求解方法，并在表 3-16 中汇总了不同可微分 topk 方法在 MCP 问题的实验结果。

预测性排列型组合优化实验。本节还研究了一个有趣且重要的预测性排列型组合优化问题，即在决策时间问题参数未知的情况下，如何预测在未来能够平衡风险和回报的投资组合（portfolio），同时满足基数约束以控制运营成本。传统上，处理类似问题需要经历两个独立的步骤：1) 通过深度学习模型预测未来的资产价格；2) 基于预测结果，求解一个优化问题以寻找最优的投资组合。本文称这种做法为“先预测、后优化”（predict-then-opt）。然而，由于预测模型中不可避免地存在误差，优化过程很可能会被误导。为了解决这个问题，Solín 等^[174]提出对无约束的投资组合目标函数进行求导^[70]，实现“联合预测-优化”。但对于更实际的、具有基数约束的投资组合优化问题，相关研究依然是空白。

问题形式。考虑投资组合优化中持有资产数量不超过 k 的实际情景^[111]，本实验的优化目标是在基数约束下同时追求高收益（由均值向量 $\mu \in \mathbb{R}^m$ 衡量）与低

表 3-16 在 MCP ($k=50, m=500, n=1000$) 问题对 CardNN-S topk 技术路径的验证。
Table 3-16 Validation of the topk methodology used in CardNN-S on MCP ($k=50, m=500, n=1000$).

求解方法	CardNN+[72]	CardNN+[74]	CardNN+[175]	CardNN-S	CardNN-GS
目标函数 (\uparrow)	32499.7	37618.9	38899.6	42034.9	44710.3

表 3-17 预测性投资组合优化在 2021 年 S&P500 数据的实验结果。
Table 3-17 Predictive portfolio optimization result on S&P500 assets in 2021.

方法名称	预测方法 + 优化方法	预测误差 \downarrow	夏普比率 \uparrow
历史最优	无 +Gurobi	(无预测)	0.673
先预测、后优化	LSTM+Gurobi	0.153	1.082
联合预测-优化	LSTM+CardNN-GS	1.382	1.968

风险 (由协方差矩阵 $\Sigma \in \mathbb{R}^{m \times m}$ 衡量)。本实验采用了夏普比率 (Sharpe ratio)^[176] 作为最大化的目标函数。该问题被表述为:

$$\max_{\mathbf{x}} \frac{(\boldsymbol{\mu} - r_f)^\top \mathbf{x}}{\sqrt{\mathbf{x}^\top \Sigma \mathbf{x}}}, \quad s.t. \quad \sum_{i=1}^m x_i = 1, \mathbf{x} \geq 0, \|\mathbf{x}\|_0 \leq k, \quad (3.63)$$

其中 \mathbf{x} 代表了每一项资产的权重, r_f 代表无风险收益 (risk-free return), 例如国债利率。需要注意的是, 真实的 $\boldsymbol{\mu}$ 、 Σ 在决策时是未知的, 它们需要用神经网络进行预测。

问题编码网络。实验采用基于长短时记忆 (LSTM) 的编码器-解码器结构作为神经网络模块 (即价格预测网络)。将历史每日价格序列输入 LSTM 编码器, 另一个 LSTM 解码器将输出对未来资产价格的预测。作者在 LSTM 的隐藏状态后附加一个全连接层来学习基数约束的概率, 将概率输入本节的 CardNN-GS 层。

目标函数估计。基于网络的输出 $\boldsymbol{\mu}, \Sigma, \tilde{\mathbf{T}}$, 利用无约束的公式 (3.63) 的闭式解来估计 \mathbf{x} 的值:

$$\mathbf{x} = \Sigma^{-1}(\boldsymbol{\mu} - r_f), \quad (3.64)$$

随后通过如下步骤添加约束:

$$\mathbf{x} = \text{relu}(\mathbf{x} \odot \tilde{\mathbf{T}}_i[2, :]), \mathbf{x} = \mathbf{x} / \text{sum}(\mathbf{x}). \quad (3.65)$$

在获得 \mathbf{x} 后, 即可通过基于真实资产价格的 $\boldsymbol{\mu}^{gt}$ 、 Σ^{gt} , 估计真实的夏普比率, 并将该夏普比率作为监督信号:

$$\tilde{J}_i = \frac{(\boldsymbol{\mu}^{gt} - r_f)^\top \mathbf{x}}{\sqrt{\mathbf{x}^\top \Sigma^{gt} \mathbf{x}}}. \quad (3.66)$$

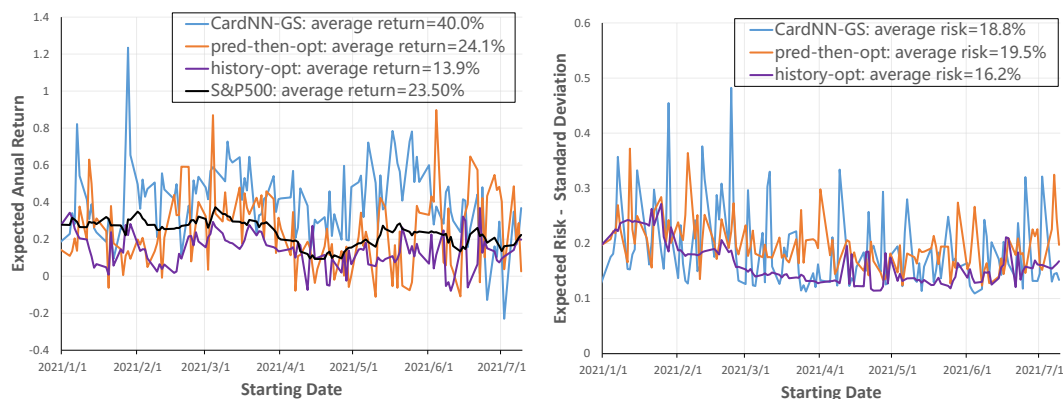


图 3-26 在 S&P500 资产上的投资组合优化实验结果：回报曲线（左）和风险曲线（右）。

Figure 3-26 Portfolio prediction results on S&P500 assets about return (left) and risk (right).

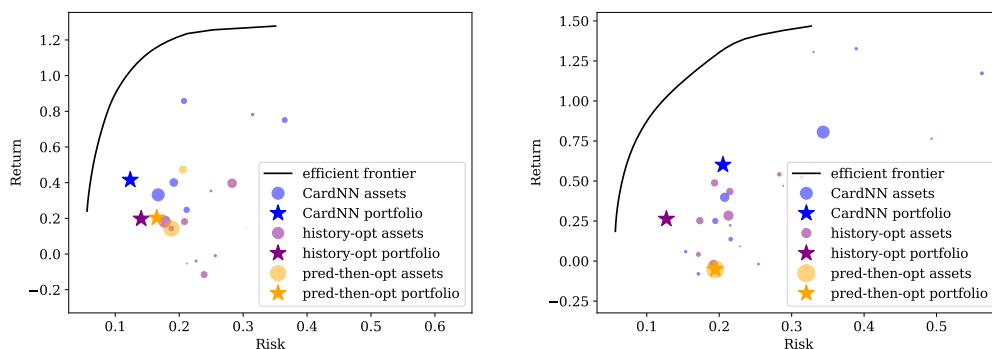


图 3-27 2021-03-25 和 05-21 投资组合的可视化。

Figure 3-27 Visualization on 2021-03-25&05-21 data of predicted portfolios.

实验设定和基准方法。实验中，价格预测网络采用真实的资产价格进行监督，但优化部分是通过自监督训练的（即没有最优解标签）。网络需要预测接下来的 120 个交易日（约 24 周）内具有最佳夏普比率的投资组合，并使用 2021 年的实际数据进行回测。训练集由 2018 年 1 月 1 日至 2020 年 12 月 30 日标普 500 (S&P500) 指数中 494 个资产组成。实验将年化无风险收益率设为 3%，基数约束 $k = 20$ 。经典的“先预测、后优化” (predict-then-opt) 学习与本节提出的“联合预测-优化” (predict-and-opt, 求解器为 CardNN-GS) 使用相同的 LSTM 模型，其中“先预测、后优化”的学习目标是 minimized 资产价格预测的平方误差，并基于预测的价格使用 Gurobi 进行优化求解。此外，实验还比较了一种“历史最优” (history-opt) 的投资方法，即遵循历史数据中的最优投资组合进行投资决策。

实验结果。作者在实际数据集（从 2021 年 1 月 1 日到 2021 年 12 月 30 日）上测试了投资组合方法，结果如图 3-26 和表 3-17 所示。图 3-26 中，基于经典的“先预测、后优化”方法（使用 LSTM 进行预测、使用 Gurobi 进行优化）和本节的 CardNN-GS 端到端的“联合预测-优化”，得到的投资组合的收益率（左）与风

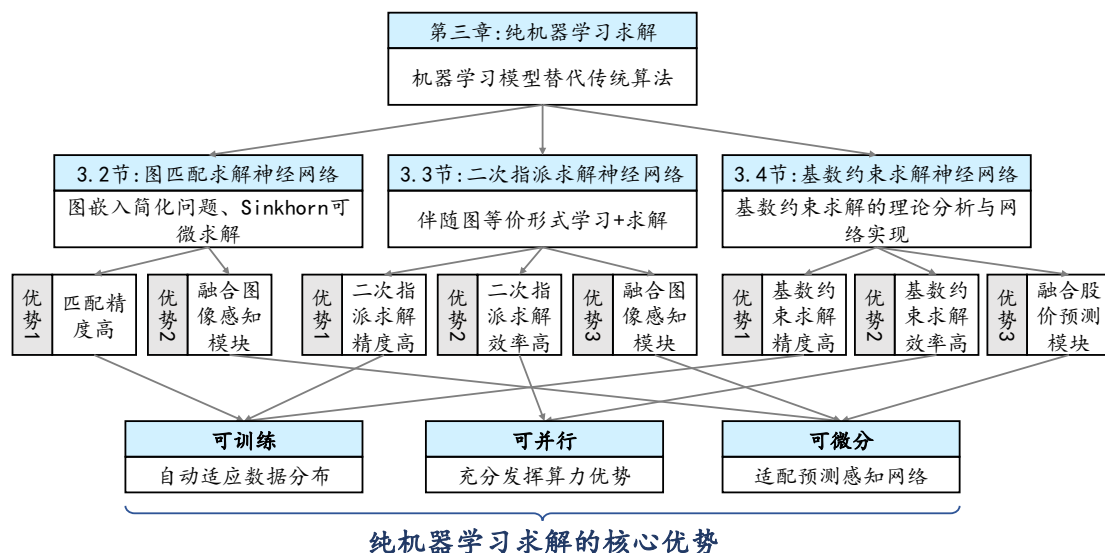


图 3-28 第三章纯机器学习求解排列型组合优化研究总结。

Figure 3-28 Summary of research on pure machine learning solvers for permutation-based combinatorial optimization.

险（右）的对比。总体而言，CardNN-GS 预测的投资组合收益更高、风险更低，将投资组合的年回报率从 24.1% 提高到了 40%。此外，由于 S&P500 指数不符合基数约束，与其直接比较风险是不公平的。表 3-17 中的“预测误差”代表价格预测的均方误差，需要注意的是更准确的价格预测并不一定带来更好的投资决策。图 3-27 展示了预测的投资组合并将其与有效边界（efficient frontier）进行了比较。有效边界是金融领域的概念，代表权衡风险和回报后最优投资组合的集合，更接近该边界代表了更好的投资组合。此外，需要注意的是，由于预测总是存在误差，在预测性投资组合优化任务中达到有效边界几乎是不可能的。

3.5 本章小结

本章研究在图匹配问题、二次指派问题、基数约束优化问题上，初步证明了纯机器学习方法用于求解排列型组合优化的可行性，对问题建模方式、网络结构设计、约束理论保证等方面开展了研究。本章的研究内容由图 3-28 总结。研究证明了纯机器学习方法的核心优势：可训练、可并行、可微分。其中，可训练体现在端到端训练的神经网络能够自适应数据分布，在 3.2 节的匹配任务、3.3 和 3.4 节的纯优化任务中，神经网络的求解精度均持平或超过了对应问题的传统求解算法。可并行体现在神经网络模型可以在 GPU 上高效地推理，例如在 3.3 和 3.4 节的纯优化任务中，神经网络方法在求解速度上相比在 CPU 上运行的传统算法更具优势。最后，可微分体现在纯神经网络求解方法能够与上下游其他深度学习模块结合，组成功能强大的端到端学习框架，例如 3.2 和 3.3 节的图像感知模块和 3.4 的股价预测模块都可以和后续的神经网络求解器一起端到端地学习。

在模型设计、理论分析等技术层面，本章还有以下贡献：3.2节主要关注图匹配问题，通过图嵌入神经网络，将图匹配问题中的高阶结构信息嵌入至节点特征，进而将原先 NP 难的组合优化简化为可在多项式时间内求解的线性指派问题。在图匹配研究中常用的仿真实验和真实图像数据集上，图嵌入神经网络求解器的有效性得到了充分证明。3.3节更进一步，考虑了图匹配问题的超集——Lawler 形式的二次指派问题。基于伴随图的等价形式，作者对最一般的二次指派问题实现了机器学习建模和特征提取，并通过在伴随图上学习一个标准的图学习任务——顶点分类任务——学习了如何求解二次指派。在纯优化数据集和图像匹配任务上，神经网络求解器均展现了优秀的性能并超越了传统算法。3.4节尝试结合理论与实践，通过对基数约束违反量的理论分析，提出了纯神经网络求解方法的设计准则：神经网络在可微的前提下，其输出应尽可能地满足约束。基于该准则设计的基数约束优化求解网络，在仿真和真实数据上超越了传统算法，和商用求解器表现相当。

第四章 融合传统算法的排列型组合优化学习求解

4.1 本章引言

求解排列型组合优化问题时，成熟的传统算法通常具有高效、可靠等特征。因此，复用传统算法中的优势模块、引入机器学习融合求解是一个颇具潜力的研究方向。图灵奖得主 Yoshua Bengio 教授的第一作者综述^[2]就重点论述了融合传统算法的学习方法的可行性和发展前景。面向不同具体问题的专用传统算法，本章分别设计了三种融合范式：面向图编辑距离问题，4.2小节使用神经网络预测模块替代传统树搜索算法中的启发式预测，由神经网络指导传统算法的搜索方向；面向图匹配问题的神经网络，4.3小节利用传统图匹配算法的高精度特性，构造图匹配的无监督学习框架，由传统算法指导神经网络；面向通用的排列型组合优化问题，4.4小节融合了高效的启发式求解和灵活的强化学习，在双层优化的框架下实现优化求解，其中传统算法和神经网络相互对等。

4.2 与树搜索融合的图编辑距离机器学习框架

图编辑距离 (graph edit distance, GED) 是一种常用的图相似度度量方法，计算图编辑距离需要恢复从源图 (source graph) 到目标图 (target graph) 的编辑路径，其本质是求解一个以最小化路径长度为目标排列型组合优化问题。传统的 A* 算法由于其穷举性质而存在效率问题，其搜索中的启发式模块¹²严重依赖于人类先验知识。本节提出了一种融合方法，利用传统搜索方法生成编辑路径以提供可解释性，同时结合了深度学习模型的高效率和适应性，以实现一种经济、高效的 GED 求解器。受动态规划的启发，作者提出了更高效的动态图嵌入方法，并鼓励对次优的搜索分支进行剪枝。为此，本节的方法可以轻松以动态方式集成到 A* 的算法流程中，并通过机器学习模块替换启发式算法，显著地减少了计算开销。在多个图数据集上的实验结果表明，本节的融合方法可以显著简化 A* 搜索过程，同时又不牺牲太多的精度。本节的工作也是第一个基于深度学习的 GED 方法，特别是能够输出编辑路径的深度学习 GED 方法。

4.2.1 图编辑距离问题介绍

图编辑距离 (GED) 是一种常用的图相似度度量，它在许多视觉和模式识别任务中都处于核心地位，应用场景包括图像匹配^[101]，签名验证^[177]，场景图编辑^[178]，药物发现^[179] 和基于案例的推理^[180]等。图4-1给出了一个具体的图编辑

¹²4.2节中，“启发式”代表精确算法中人工设计的预测模块（不直接输出解）。本文的其他章节中，“启发式”代表人工设计的快速求解算法（直接输出解）。优化领域的文献通常不区分这两种应用场景下“启发式”的命名，特在此说明。

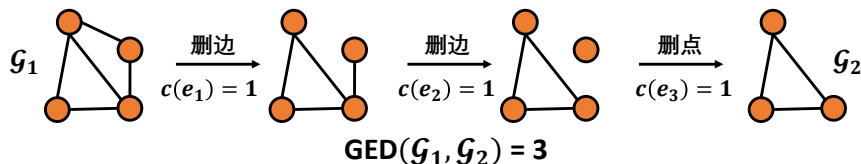


图 4-1 面向简单图 $\mathcal{G}_1, \mathcal{G}_2$ 的图编辑路径。

Figure 4-1 An edit path between two simple graphs $\mathcal{G}_1, \mathcal{G}_2$.

路径的例子，一般来说，GED 算法的目标是找到从源图到目标图的最佳编辑路径，使得编辑代价最小：

$$\text{GED}(\mathcal{G}_1, \mathcal{G}_2) = \min_{(e_1, \dots, e_l) \in \gamma(\mathcal{G}_1, \mathcal{G}_2)} \sum_{i=1}^l c(e_i), \quad (4.1)$$

其中 $\gamma(\mathcal{G}_1, \mathcal{G}_2)$ 代表了将源图 \mathcal{G}_1 编辑成目标图 \mathcal{G}_2 的所有可能的编辑路径。 $c(e_i)$ 代表路径 e_i 对应的代价。上述问题本质上是一个 NP 难的排列型组合优化问题^[181]，编辑路径等价于节点匹配关系的排列。

在任何情况下，精确的 GED 求解器^[38,181]均能保证找到最优解，但在大规模的图上效率较低。这些精确求解器依赖启发式方法来估计当前部分解（partial solution）对应的图相似度。近年来，许多深度图学习方面的努力^[161-162,182]采用图神经网络^[133,183]直接回归一个图相似度分数。这些深度学习方法没有显式地考虑 GED 的组合性质，无法输出编辑路径。然而，在许多应用中，求解编辑路径往往才是问题的关键^[101,178]，且大多数现有的 GED 工作^[38,181,184-186]仍然更加专注于求解编辑路径本身。

传统的 GED 求解器。精确的 GED 求解器可以处理小规模的问题，通过穷举搜索找到全局最优解。精确方法的主要基础是树搜索，例如 A* 算法^[38]，其中维护了一个所有待搜索状态的优先队列，并且搜索顺序由一个启发式分数所控制。启发式分数由两部分组成——当前部分编辑路径的代价（精确值）和剩余子图之间编辑距离（预测值）^[184,186]，如图4-2所示。A* 算法可以看作是在所有可能的编辑操作组合上进行广度优先搜索，而其他组合优化技术，如深度优先的分支定界^[181]和基于线性规划下界方法^[187]也可被采用，进而剪掉搜索树中的非必要分支。然而，精确的 GED 方法计算时间太长，在大型图上效率较差^[188]。近似的 GED 求解器在通常情况下，可以在多项式时间内得到一个近似解，较为高效地求解 GED。迄今为止，基于二分图匹配的方法^[184-186]能够较好地权衡时间与精度^[189]，其中边编辑代价被编码进节点代价，得到的二分图匹配问题可以通过匈牙利算法^[184,190]或 Volgenant-Jonker 算法^[185,191]来求解。集束搜索^[192]是精确 A* 算法的贪心版本。另一条研究路线，即近似图匹配^[34,88,98,193-194]与近似的 GED 密切相关，已有一些研究将图匹配方法（例如 IPFP^[87]）应用于求解 GED 问题^[39]。

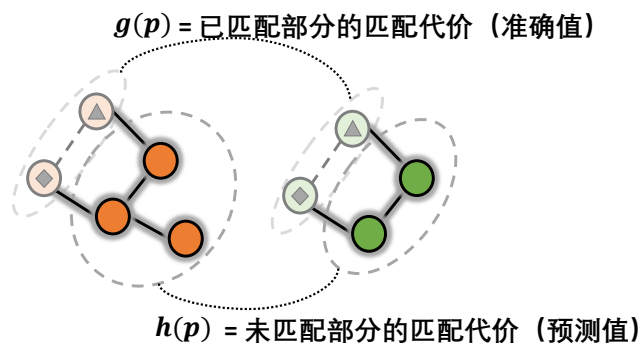


图 4-2 A^* 搜索树中的一个状态（对应一个部分编辑路径）的示意图。
Figure 4-2 Illustration of a partial edit path as one state of A^* search tree.

近似求解 GED 的两个缺点是它们严重依赖人类知识，且解的质量相对较差。

深度学习的 GED 求解器。最近，机器学习方法在非欧几里得数据（即图）上的成功^[133,147,183,195]促使研究人员设计了若干回归形式的神经网络，用于预测 GED 等图相似度度量。SimGNN^[162] 首先将图相似度学习视作回归任务，其中的图卷积层^[133]和注意力层^[53]受到监督训练，监督信号来自精确的 A^* ^[192]算法。Bai 等^[182]通过使用 CNN 处理多尺度节点相似性，扩展了他们的先前工作。Li 等^[161]在前馈图神经网络中提出了跨图模块，将其用于图相似度学习。上述方案也被应用在信息检索中，其中 Dai 等^[196]采用卷积网络预测文本之间的编辑代价。然而，所有这些回归模型都无法预测编辑路径，无法处理必须输出编辑路径的应用场景。

4.2.2 研究动机

随着图规模的增长，需要 GED 求解器更加高效的同时，也需要求解器能够输出准确的编辑路径。然而，现有方法无法同时具备这两个优点。如上所述，基于深度学习的求解器难以恢复编辑路径，而非学习的方法则存在效率问题。本节旨在结合两种方法的优点，设计一个融合求解器。

具体而言，本节的融合求解器主要基于 A^* 算法^[38]，这是一种在开源 GED 软件^[192,197]中广泛使用的求解算法；同时，本节采用神经网络预测相似度分数以指导 A^* 搜索，替代传统 A^* 算法中手工设计的启发式方法。在技术层面，本节提出的图编辑神经网络（Graph Edit Neural Network, GENN）在两个方面受到了经典的动态规划算法的启发：首先，在图被修改（例如节点被删除）时复用现有的嵌入信息，在 A^* 搜索树状态中逐步删除图节点¹³；其次，通过学习更高效的启发式方法，可以避免探索不必要的次优分支，从而实现显著的加速。值得注意的是，与原始的 A^* 算法相比，本节的学习方法不再是一个精确的 GED 求解器；

¹³为了区分图中的“节点”和搜索树中的“节点”，本节将搜索树的节点命名为“状态”。

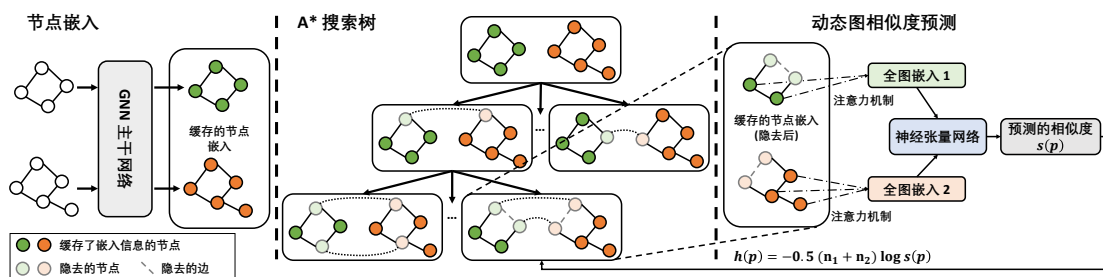


图 4-3 4.2节提出的 GENN-A* 方法概览。

Figure 4-3 Overview of the proposed GENN-A* in Sec. 4.2.

但实验结果显示，本节的融合算法在极小的精度损失代价下，显著减少了 A* 算法的运行时间。

本节的研究动机具体总结如下：

1) 设计一种融合深度网络求解 GED 的求解器，其中搜索树的状态选择启发式由图嵌入学习模块替代。通过融合传统算法的高精度和深度神经网络的高效率，融合算法在效率上优于传统的启发式方法。

2) 设计一种基于动态规划的图嵌入方法，最大程度地复用先前的计算。在这个意义上，本节的方法可以自然地与 A* 算法集成，在每次图修改后动态地预测图相似度。与普通的图嵌入方法相比，基于动态规划的算法复杂度更低。

4.2.3 算法描述

前序知识：用于求解 GED 的 A* 算法。为了精确求解 GED 问题，研究人员通常采用基于树搜索的算法来遍历所有可能的编辑操作。其中，A* 算法相当受欢迎^[38,184,192,197]，包括本节的学习方法也是基于 A* 算法。GED 旨在寻找最小编辑代价对应的最优编辑路径，将源图 $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ 转换为目标图 $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ ，其中 $|\mathcal{V}_1| = n_1, |\mathcal{V}_2| = n_2$ 。将 $\mathcal{V}_1 = \{u_1, \dots, u_{n_1}\}$ ， $\mathcal{V}_2 = \{v_1, \dots, v_{n_2}\}$ 表示为源图和目标图中的节点， ε 表示“虚拟节点”。可能的节点编辑操作包括节点替换 $u_i \rightarrow v_j$ ，节点插入 $\varepsilon \rightarrow v_j$ 和节点删除 $u_i \rightarrow \varepsilon$ ，每个操作的代价可以根据问题进行定义。如图 4-2 所示，给定节点编辑后，边的编辑方式也固定了，例如给定 $p = (u_\blacklozenge \rightarrow v_\blacklozenge, u_\blacktriangle \rightarrow v_\blacktriangle)$ ，即可得到边的编辑方式 $(u_\blacklozenge u_\blacktriangle \rightarrow v_\blacklozenge v_\blacktriangle)$ 。因此在 A* 算法中只明确考虑节点的编辑¹⁴。

算法 4-1 展示了一种标准的 A* 算法，与现有工作^[38,184]保持一致。它维护了一个优先队列，其中每个搜索树状态都包含了 GED 问题的一个部分编辑路径。如图 4-2 所示，每个状态的优先级定义为两个指标之和： $g(p)$ 表示当前部分路径的代价，可以准确计算； $h(p)$ 表示未匹配子图之间 GED 的启发式预测。A* 总是探索

¹⁴节点替换可以视为两个图之间的节点匹配，而节点插入/删除可以视为将源/目标图中的节点与虚节点匹配，因此，在本节中，“匹配”和“编辑”这两个概念可以互换使用。

算法 4-1 精确求解 GED 的 A* 算法

输入: 图 $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, 其中 $\mathcal{V}_1 = \{u_1, \dots, u_{n_1}\}$, $\mathcal{V}_2 = \{v_1, \dots, v_{n_2}\}$

- 1: 将 OPEN 初始化为空的优先队列;
- 2: 对于所有 $w \in \mathcal{V}_2$, 将 $(u_1 \rightarrow w)$ 插入 OPEN;
- 3: 将 $(u_1 \rightarrow \varepsilon)$ 插入 OPEN;
- 4: **while** 未找到完整解 **do**
- 5: 从 OPEN 中选择 $(g(p) + h(p))$ 最小的 p ;
- 6: **if** p 是一个完整解 **then**
- 7: **return** 完整的解 p ;
- 8: **else**
- 9: 定义 p 包含了以下节点: $\mathcal{A} = \{u_1, \dots, u_k\} \subseteq \mathcal{V}_1$ 和 $\mathcal{B} \subseteq \mathcal{V}_2$;
- 10: **if** $|\mathcal{A}| \leq n_1$ **then**
- 11: 对所有的 $v_i \in \mathcal{V}_2 \setminus \mathcal{B}$, 向 OPEN 中插入 $p \cup (u_{k+1} \rightarrow v_i)$;
- 12: 向 OPEN 插入 $p \cup (u_{k+1} \rightarrow \varepsilon)$;
- 13: **else**
- 14: 向 OPEN 中插入 $p \cup \bigcup_{v_i \in \mathcal{V}_2 \setminus \mathcal{B}} (\varepsilon \rightarrow v_i)$;
- 15: **end if**
- 16: **end if**
- 17: **end while**

输出: 从 \mathcal{G}_1 到 \mathcal{G}_2 的最优编辑路径。

每次迭代中 $g(p) + h(p)$ 最小的状态, 如果对于所有部分编辑路径, $h(p) \leq h^{opt}(p)$ 始终成立, 则最优性可以得到保证^[184], 其中 $h^{opt}(p)$ 表示未匹配子图之间的最优编辑代价。

以 A* 算法的三种变体为例, 作者希望说明合适的 $h(p)$ 对于 A* 算法的效率而言非常重要: 1) 如果 $h(p) = h^{opt}(p)$, 则可以贪心地找到最优路径。然而, 计算 $h^{opt}(p)$ 需要另一个指数时间的求解器, 因此这在实际中是不可行的。2) 可以利用启发式来预测 $h(p)$, 其中 $0 \leq h(p) \leq h^{opt}(p)$ 。基于匈牙利算法的二分图匹配启发式是表现最好的启发式之一, 其时间复杂度为 $\mathcal{O}((n_1 + n_2)^3)$ 。本节的实验也考虑了这一传统求解器 Hungarian-A*^[38]。3) Plain-A* 是最简单的算法, 始终保持 $h(p) = 0$, 并且这种策略在计算 $h(p)$ 时不会增加任何开销。然而, 因为缺乏对未来代价的预测, 搜索树可能变得过大, 导致效率降低。

深度学习在图相似度学习的成功^[161-162,182]表明, 可以通过高效率的学习方法预测高质量的 $h(p)$, 其中预测值与 $h^{opt}(p)$ 接近, 进而缓解 A* 算法的效率问题。本节提出的动态图嵌入网络能够预测 $h(p)$, 并通过剪枝 A* 中的次优分支来实现对 $h(p)$ 的高效学习和预测。值得注意的是, 本节的方法破坏了最优性条件 $h(p) \leq h^{opt}(p)$, 但实验证明, 这种精度损失是有限的。

节点嵌入神经网络。如图4-3所示, 本节提出的 GENN 神经网络的整体流程与现有的神经网络方法 SimGNN^[162]类似, 但为了提升效率, 作者去除了 SimGNN 中冗余的直方图模块。给定输入的图, 节点嵌入通过图神经网络 (GNN) 计算得到。GENN 的实现细节介绍如下。

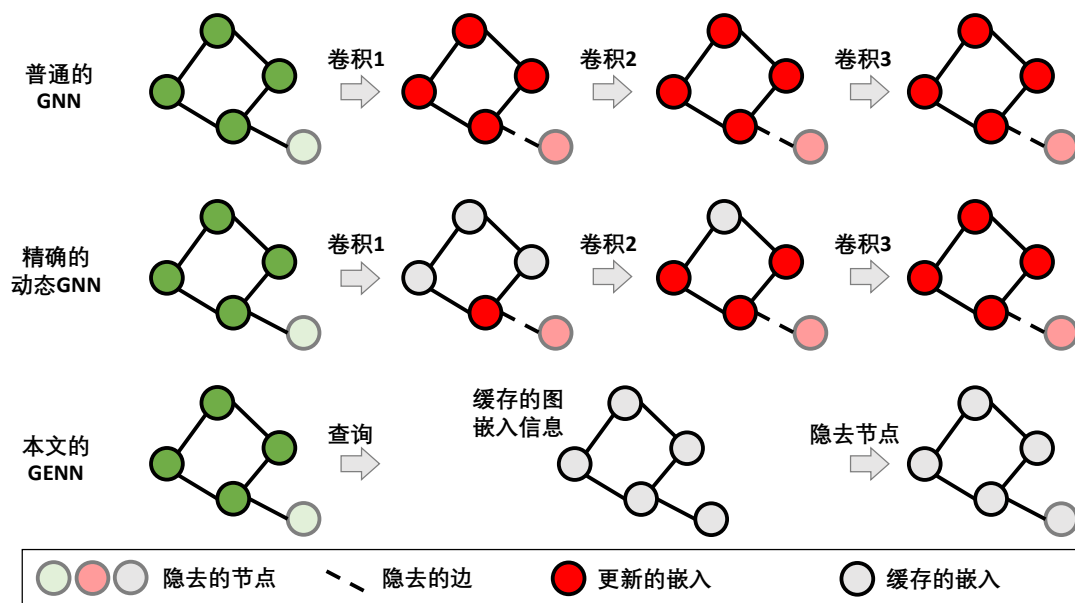


图 4-4 三种用于 A^* 算法动态嵌入的图神经网络的对比。

Figure 4-4 Comparison of three graph neural network variants for dynamic graph embedding in A^* algorithm.

初始化。首先，节点嵌入被初始化为节点度数（即节点连边数目）的独热（one-hot）编码。对于具有节点标签的图（例如分子图），作者使用 one-hot 向量对节点标签进行编码，并将其拼接到度数上。根据不同的图定义，边可以初始化成加权或不加权的。

GNN 主干网络。考虑到不同的图数据类型，对于普通的图结构（例如分子图和程序图）采用图卷积网络（GCN）^[133]；考虑到样条卷积在学习几何特征时的优越性^[78,82]，对于从 2D 图像构建的图，采用 SplineConv^[147]。GENN 的节点嵌入由 GNN 主干网络得到，并通过缓存以进行更高效的动态图嵌入。遵循 SimGNN^[162]的网络结构，GENN 拥有三层 GNN 。

A^* 本质上是一种动态规划（dynamic programming, DP）算法，其中部分编辑路径中已匹配的节点在算法流程中被逐步隐去。在求解 GED 时， A^* 的每个状态包含一个部分编辑路径，本节采用 GENN 网络来预测两个未匹配子图之间的距离。在每个状态下，与其父状态相比，未匹配子图中多隐去了一个节点。这种动态规划场景不同于现有的所谓动态图嵌入（dynamic embedding）研究^[198-200]。在 A^* 算法中，由于在搜索树的每个状态中都需要预测 $h(p)$ ，因此需要更高效的实现以提升整体效率。接下来，本节将讨论并比较三种可能的动态嵌入方法。

面向 A^* 算法的动态图嵌入。 A^* 算法的动态规划性质提供了一种提升效率的方向——动态图嵌入。图4-4总结了三种不同的动态图嵌入策略，假设共有 3 个图卷积层，其中，普通 GNN 的所有节点需要进行完整的前向传递，其中包含了冗余操作；精确的动态 GNN 缓存了所有中间嵌入，只有隐去节点的 3 跳邻居会被更新；最后，GENN 的动态嵌入不需要进行卷积操作，是效率最高的动态图

嵌入方法。接下来将分别讨论三种动态嵌入方法的优劣。

普通的 *GNN* 网络。处理动态情况的一种简单方式是，在图被修改时，为新图中的所有节点调用完整的前向传递。然而，这种做法含有冗余的计算。将节点数表示为 n ，嵌入维度表示为 F ，*GNN* 层数表示为 K 。在最坏情况——图是全连接的时候，普通 *GNN* 的时间复杂度为 $\mathcal{O}(n^2FK + nF^2K)$ 。普通 *GNN* 的优点是无需缓存。

精确的动态 *GNN* 网络。如图4-4的第二行所示，通过分析图神经网络的消息传递路径，当节点被隐去时，只有其相邻节点的嵌入会受到影响。因此，如果缓存所有前向传递的中间结果，可以更高效地计算出精确的图嵌入。基于图神经网络的消息传递特性，在第 k 个卷积层，只有隐去节点的 k 跳邻居需要更新。然而，精确的动态 *GNN* 在最坏情况的时间复杂度仍然为 $\mathcal{O}(n^2FK + nF^2K)$ （对于完全连接的图），并且它需要 $\mathcal{O}(nFK)$ 的内存用于缓存所有卷积层。如果为了最佳的时间效率而缓存所有可能的子图，则内存开销将增长到无法接受的 $\mathcal{O}(n2^nFK)$ 。实验表明，这种方法的提速效果几乎可以忽略不计。

本节提出的 *GENN*。如图4-4的最后一行所示，*GENN* 首先进行前向卷积，并缓存最后一层的嵌入特征。在 A^* 算法中，如果某些节点被隐去，*GENN* 只需将它们从最后一个卷积层中删除，并将剩余的节点输入相似度预测模块。*GENN* 只涉及单次前向传递，其开销可以忽略不计；加载缓存的时间复杂度为 $\mathcal{O}(1)$ ，内存复杂度为 $\mathcal{O}(nF)$ 。

GENN 的设计主要受到动态规划的启发：给定输入图修改操作（在本节的 A^* 搜索中为删除节点），动态规划算法复用了已有的计算结果，以获得最佳的计算效率。*GENN* 缓存了节点嵌入特征用于子图的相似性计算。此外，动态规划算法倾向于最小化搜索空间，而本节学到的 $h(p)$ 是一种比传统的启发式方法更激进的剪枝策略，从而加速了 A^* 算法。

预测图相似度。从缓存中获取嵌入向量后，调用注意力模块和神经张量网络即可预测相似度得分。为了行文的简洁，不失一般性，这里仅讨论了完整大小的原始输入图。

用于全图嵌入的注意力模块。给定节点级的嵌入，通过注意机制^[53]可以得到全图级的嵌入。将 $\mathbf{H}_1 \in \mathbb{R}^{n_1 \times F}$, $\mathbf{H}_2 \in \mathbb{R}^{n_2 \times F}$ 表示为来自 *GNN* 主干网络的节点嵌入，全局的键值向量通过平均聚合后进行非线性变换得到：

$$\bar{\mathbf{H}}_1 = \text{mean}(\mathbf{H}_1), \bar{\mathbf{H}}_2 = \text{mean}(\mathbf{H}_2), \quad (4.2)$$

$$\mathbf{k}_1 = \tanh(\bar{\mathbf{H}}_1 \mathbf{W}_1), \mathbf{k}_2 = \tanh(\bar{\mathbf{H}}_2 \mathbf{W}_1), \quad (4.3)$$

其中 $\text{mean}(\cdot)$ 在第一维（节点维度）上执行， $\mathbf{W}_1 \in \mathbb{R}^{F \times F}$ 是可学习的注意力权重。

聚合系数由 $\mathbf{k}_1, \mathbf{k}_2 \in \mathbb{R}^{1 \times F}$ 和 $\mathbf{H}_1, \mathbf{H}_2$ 计算得到:

$$\mathbf{c}_1 = \delta(\mathbf{H}_1 \mathbf{k}_1^\top \cdot \alpha), \mathbf{c}_2 = \delta(\mathbf{H}_2 \mathbf{k}_2^\top \cdot \alpha). \quad (4.4)$$

其中 $\alpha = 10$ 是缩放因子, $\delta(\cdot)$ 表示 Sigmoid 函数。基于聚合系数 $\mathbf{c}_1 \in \mathbb{R}^{n_1 \times 1}, \mathbf{c}_2 \in \mathbb{R}^{n_2 \times 1}$, 通过节点嵌入特征的加权求和得到全图嵌入特征:

$$\mathbf{g}_1 = \mathbf{c}_1^\top \mathbf{H}_1, \mathbf{g}_2 = \mathbf{c}_2^\top \mathbf{H}_2. \quad (4.5)$$

用于预测相似度的神经张量网络。神经张量网络 (Neural Tensor Network, NTN)^[201] 被用于计算 $\mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^{1 \times F}$ 之间的相似度分数:

$$s(\mathcal{G}_1, \mathcal{G}_2) = f(\mathbf{g}_1 \mathbf{W}_2^{[1:t]} \mathbf{g}_2^\top + \mathbf{W}_3 [\mathbf{g}_1 \quad \mathbf{g}_2] + \mathbf{b}), \quad (4.6)$$

其中 $\mathbf{W}_2 \in \mathbb{R}^{F \times F \times t}, \mathbf{W}_3 \in \mathbb{R}^{t \times 2F}, \mathbf{b} \in \mathbb{R}^t$ 是可学习的权重, 第一项代表为所有的 $i \in [1 \dots t]$ 计算 $\mathbf{g}_1 \mathbf{W}_2[:, :, i] \mathbf{g}_2^\top$ 、并将结果拼接起来。 $f: \mathbb{R}^t \rightarrow (0, 1)$ 代表一个由 Sigmoid 激活的全连接层。 t 代表神经张量网络的特征通道数, 本文按经验将其设为 $t = 16$ 。

与前序工作^[162]一致, 神经网络模型的预测在 $(0, 1)$ 之间, 其含义是归一化后的 GED 分数, 它和实际 GED 的关联是:

$$s(\mathcal{G}_1, \mathcal{G}_2) = \exp(-GED(\mathcal{G}_1, \mathcal{G}_2) \times 2 / (n_1 + n_2)). \quad (4.7)$$

对于 A* 算法中遇到的部分编辑路径, 神经网络预测的相似度分数 $s(p)$ 可通过公式 (4.7) 转换为 $h(p)$:

$$h(p) = -0.5(n'_1 + n'_2) \log s(p), \quad (4.8)$$

其中 n'_1, n'_2 代表未匹配子图中的节点数目。注意力层和神经张量网络的时间复杂度分别是 $\mathcal{O}((n'_1 + n'_2)F^2)$ 和 $\mathcal{O}(n'_1 n'_2 F t)$ 。由于在求解 GED 过程中, 图卷积层只被调用了一次, 其开销可以忽略不计; 加载已缓存的 GENN 嵌入的时间复杂度是 $\mathcal{O}(1)$ 。因此, 每次调用神经网络预测的时间复杂度是 $\mathcal{O}((n'_1 + n'_2)F^2 + n'_1 n'_2 F t)$ 。该时间复杂度和传统的启发式^[38, 202]相近, 传统启发式的时间复杂度一般为 $\mathcal{O}(\min(n'_1, n'_2)^2 \max(n'_1, n'_2))$ 。

面向动态图神经网络的监督学习。 GENN 的训练包括两个步骤: 首先, GENN 权重由训练数据集中的图相似度分数初始化。其次, 模型通过 A* 算法求解得到的最优编辑路径进行微调。详细的训练过程请参见算法 4-2。

类似于其他深度图相似度学习方法^[162, 182], GENN 的权重在数据集标签的监督下训练。对于相对较小的图数据集, GED 问题的最优解能够穷举得到, 可以将精确的 GED 作为监督信号。在无法精确求解 GED 的情况下, 可以根据其他有意义的标准来构建训练集, 例如基于语义节点匹配计算 GED 标签。

算法 4-2 GENN-A* 的训练流程

输入: 训练集的图 $\{(G_i, G_j)\}$ 以及相似度分数标签 $\{s^{gt}(G_i, G_j)\}$.

```

1: while 未收敛 do # 使用训练集标签监督训练
2:   从数据集中随机采样  $(G_i, G_j)$ ;
3:   利用普通 GENN 计算  $s(G_i, G_j)$ ;
4:   通过损失函数  $MSE(s(G_i, G_j), s^{gt}(G_i, G_j))$  更新网络权重;
5: end while
6: while 未收敛 do # 使用最优路径微调
7:   从训练集中随机采样  $(G_i, G_j)$ ;
8:   利用 A* 算法求解最优的编辑路径  $p^*$  和  $GED(p^*)$ ;
9:   在  $(G_i, G_j)$  上调用 GENN, 并缓存节点嵌入结果;
10:  for 部分编辑路径  $p \subseteq p^*$  do
11:    计算  $g(p)$  和  $h^{opt}(p) = GED(p^*) - g(p)$ ;
12:     $s^{opt}(p) = \exp(-2h^{opt}(p)/(n'_1 + n'_2))$ ;
13:    基于缓存的 GENN 嵌入计算  $s(p)$ ;
14:    通过损失函数  $MSE(s(p), s^{opt}(p))$  更新网络权重;
15:  end for
16: end while
输出: 训练完毕的 GENN 网络。

```

本节进一步提出了一种微调 GENN 的方案, 以更好地适应 A* 搜索。然而, 使用搜索树状态对 GENN 进行微调意味着需要 $h^{opt}(p)$ 的标签, 而对于任意部分编辑路径, 求解 $h^{opt}(p)$ 仍然是 NP 难的。为了尽可能地降低求解 $h^{opt}(p)$ 的次数, 作者提出了一种通过仅求解一次 GED 来获得多个 $h^{opt}(p)$ 标签的有效方法。

定理 4.1 (最优的部分代价). 给定最优的编辑路径 p^* 和对应的 $GED(p^*)$, 对任意的部分编辑路径 $p \subseteq p^*$, 有 $g(p) + h^{opt}(p) = GED(p^*)$ 。

证明. 如果 $g(p) + h^{opt}(p) > GED(p^*)$, 那么基于路径 p 得到的最小编辑代价大于 $GED(p^*)$, 因此 p 不是一条最优的部分路径, 这和条件 $p \subseteq p^*$ 冲突。如果 $g(p) + h^{opt}(p) < GED(p^*)$, 这意味着存在一条更好的编辑路径, 其代价小于 $GED(p^*)$ 。这和 p^* 是最优编辑路径的条件冲突。因此, $g(p) + h^{opt}(p) = GED(p^*)$ 。□

根据定理 4.1, 对于任意最优的部分编辑路径 p , $h^{opt}(p) = GED(p^*) - g(p)$ 都成立。因此, 如果求解得到了一个包含 m 个节点的最优路径 p^* , 那么可以得到 $(2^m - 1)$ 个最优的部分编辑路径用于微调。实验中, 作者随机选择了 200 对图进行微调, 产生的微调数据集已经足够使网络收敛。

4.2.4 实验分析

实验数据集和实验设定. 本节用于实验分析的数据集介绍如下:

AIDS 数据集包含了若干用于评估抗 HIV 活性的有机化合物¹⁵。前序工作^[162]对 AIDS 数据集进行了预处理, 移除了超过 10 个节点的图, 并计算了任意两个图

¹⁵<https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data>

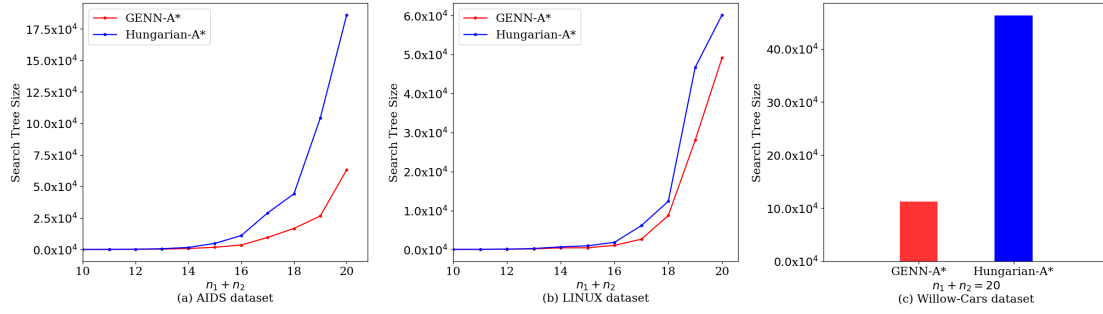
图 4-5 平均搜索树规模和问题规模 ($n_1 + n_2$) 的关系。Figure 4-5 Average search tree size w.r.t. problem size ($n_1 + n_2$).

表 4-1 在 AIDS, LINUX, Willow-Cars 的测试结果。

Table 4-1 Evaluation on benchmarks AIDS, LINUX and Willow-Cars.

方法名称	恢复编辑路径	AIDS			LINUX			Willow-Cars		
		mse ($\times 10^{-3}$) ↓	ρ ↑	p@10 ↑	mse ($\times 10^{-3}$) ↓	ρ ↑	p@10 ↑	mse ($\times 10^{-3}$) ↓	ρ ↑	p@10 ↑
SimGNN ^[162]	×	1.189	0.843	0.421	1.509	0.939	0.942	-	-	-
GMatch ^[161]	×	1.886	0.751	0.401	1.027	0.933	0.833	-	-	-
GraphSim ^[182]	×	0.787	0.874	0.534	0.058	0.981	0.992	-	-	-
GENN	×	1.618	0.901	0.467	0.438	0.955	0.428	-	-	-
Beam Search ^[192]	✓	12.090	0.609	0.481	9.268	0.827	0.973	1.820	0.815	0.725
Hungarian ^[184]	✓	25.296	0.510	0.360	29.805	0.638	0.913	29.936	0.553	0.650
VJ ^[185]	✓	29.157	0.517	0.310	63.863	0.581	0.287	45.781	0.438	0.512
GENN-A*	✓	0.839	0.953	0.866	0.324	0.991	0.962	0.599	0.928	0.938

之间的最优 GED。仿照前人的工作，实验中的节点编辑代价定义为：如果 u_i, v_j 是不同的原子，则 $c(u_i \rightarrow v_j) = 1$ ；否则 $c(u_i \rightarrow v_j) = 0$ 。节点插入和删除的代价都定义为 1。边被视为无属性的边，因此替换边的代价 = 0，插入/删除边的代价 = 1。

LINUX 数据集由 Wang 等^[203]提出，包含了来自 LINUX 内核的程序依赖图。前序工作^[162]也提供了经过预处理的版本，其中最大的图具有 10 个节点，数据集中包含了最优的 GED 值。所有节点和边都没有属性，因此替换代价为 0，插入/删除代价为 1。

Willow 数据集是用于匹配图像语义关键点的数据集，在 3.2 节和 3.3 节的图匹配实验中均有涉及。本节使用 Willow ObjectClass 数据集验证了 GENN-A* 在计算机视觉问题上的性能。实验选择了“汽车”数据集，通过德罗内三角剖分构建了带有 2D 关键点位置的图，其中边的编辑代价为 $c(\mathcal{E}_i \rightarrow \mathcal{E}_j) = |\mathcal{E}_i - \mathcal{E}_j|$ ，其中 $\mathcal{E}_i, \mathcal{E}_j$ 是两条边的长度。 \mathcal{E}_i 的插入/删除代价定义为 $|\mathcal{E}_i|$ 。为了数值上的考虑，所有边长都用 300 归一化。节点替换代价为 0，禁止插入/删除节点 $c(u_i \rightarrow \varepsilon) = c(\varepsilon \rightarrow v_j) = \infty$ 。实验基于语义关键点匹配关系来构建训练集标签，值得注意的是，使用该方法得到的监督信号与最优 GED 不同。不过，实验结果表明，这种监督足以初始化 GENN 的模型权重。

作者认为，在三个数据集中，LINUX 数据集的问题定义最简单、最容易求解。相比之下，AIDS 数据集具有带属性的节点，而 Willow 数据集具有带属性的

表 4-2 求解单个 GED 问题的平均耗时 (秒)。

Table 4-2 Averaged time (sec) for solving one GED problem.

方法名称	AIDS	LINUX	Willow-Cars
Hungarian-A*[38]	30.542	2.332	188.234
GENN-A*	16.235	2.177	78.481

表 4-3 Linux 数据集上, 预测 $h(p)$ 的平均耗时 (毫秒)。Table 4-3 Averaged time (msec) to predict $h(p)$ on LINUX dataset.

	普通 GNN	精确的动态 GNN	GENN	Hungarian 启发式[38]
耗时	2.329	3.145	0.417	0.358

边, 这使得这两个数据集比 LINUX 数据集更具挑战性。与 Bai 等[162]一致, 本节将所有数据集的 60% 用于训练, 20% 用于验证, 20% 用于测试。

实验框架采用 Python 编写, 图神经网络部分使用 Pytorch-Geometric[204]实现。为性能考虑, A* 算法使用 Cython[205]实现。在图神经网络的选择上, GCN[133]被用于 AIDS 和 LINUX 数据集, SplineCov[147]被用于处理 Willow-Cars 数据集中的 2D 欧几里得数据 (核数目 = 16)。三层 GNN 的特征通道数分别定义为 64、32、16。训练时, 采用了 0.001 学习率的 Adam 优化器[155]和 5×10^{-5} 的权重衰减。LINUX 和 AIDS 数据集的批大小设为 128, Willow 数据集的批大小为 16。本实验没有考虑多线程和 GPU 并行技术。

对比方法。精确的 GED 求解器 Hungarian-A*[38]是主要对比方法, 它使用了匈牙利二分图匹配算法来预测 $h(p)$ 。为了对比的公平性, 实验基于 Cython 重新实现了 Hungarian-A*。Hungarian 求解器[184]也被采用, 作为一种传统的近似求解算法。值得注意的是, 匈牙利二分图匹配算法既可以被用于 A* 算法的启发式方法 (即 Hungarian-A* 的启发式), 也可以快速地计算 GED 问题的近似解 (即 Hungarian 求解器), 读者应注意区分这两种方法。其他近似求解器也被考虑在内, 包括集束搜索 (beam search)[192], 它是 A* 的贪婪版本, 以及 VJ[185], 它是 Hungarian 求解器的一种变体。对于回归形式的深度图相似度学习方法, 实验比较了 SimGNN[162], GMatch[161]和 GraphSim[182]。本节的 GENN 主干网络是这些模型的简化版本, 因为在融合算法中神经网络的计算效率至关重要。

实验结果和分析。对标文献[182]的实验结果请见表4-1。在该实验中, GED 方法需要完成图检索任务, 其中数据库由训练集组成, 检索项来自训练集。相似度分数的定义请见公式 (4.7)。本节提出的方法可以基于传统 GED 求解器[38,184-185]的方式进行工作, 显式地输出编辑路径 (GENN-A*), 也可以直接回归式地预测相似度[161-162,182], 不输出编辑路径 (GENN)。此处采用了前序工作定义的评测指标[162,182]: **mse** (mean square error) 表示预测的相似度分数与真实相似度分数之间的均方误差。 ρ 表示预测和真实相似度之间的斯皮尔曼 (Spearman) 相关性。

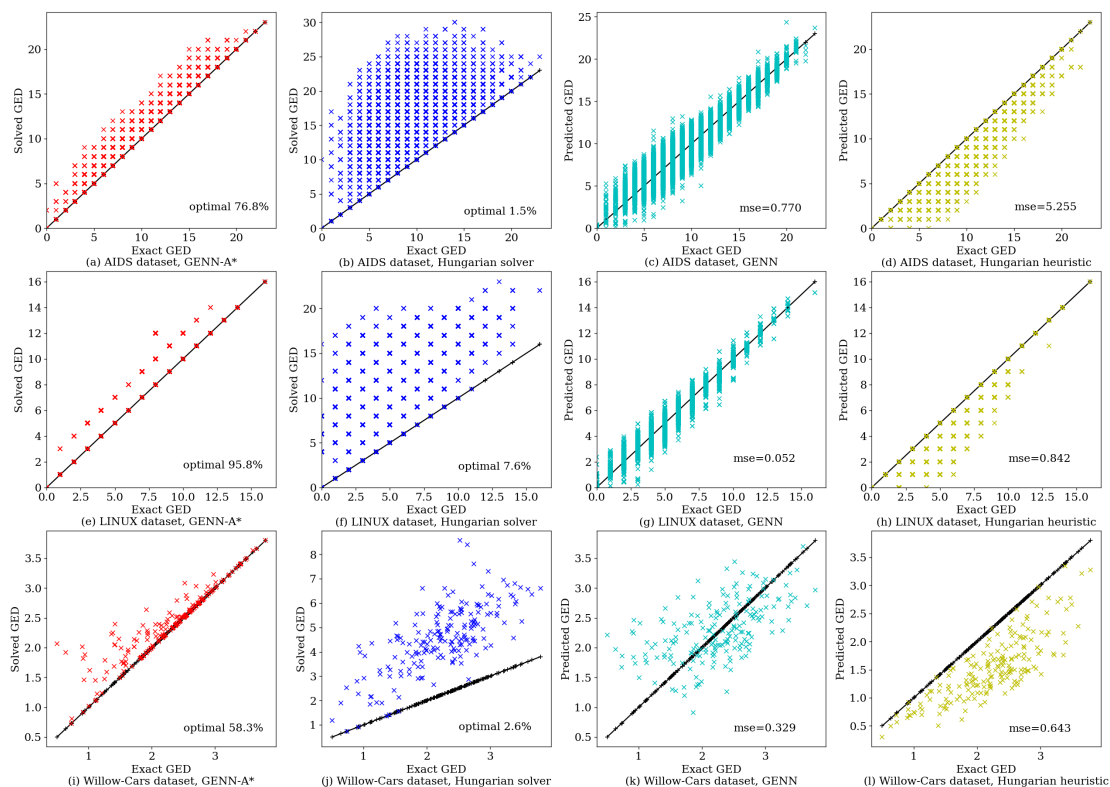


图 4-6 GENN-A* (红色)、近似的 Hungarian 求解器^[184] (蓝色)、GENN (黄色)、精确 A* 求解器中的 Hungarian 启发式^[38] (黄色) 在 AIDS、LINUX、Willow-Cars 数据集的散点图。

Figure 4-6 The scatter plots of our proposed GENN-A* (red), inexact Hungarian solver^[184] (blue), our GENN network (cyan) and Hungarian heuristic in exact A* solver^[38] (yellow) on AIDS, LINUX and Willow-Cars datasets.

p@10 (precision@10) 表示在预测的前 10 个图中包含最相似图的精度。Willow-Cars 没有与深度学习方法进行比较, 因为训练集中没有最优的 GED 标签。AIDS 和 LINUX 数据集中, 对比方法的结果引用自论文^[182]。表 4-1 的结果表明, 本节的回归模型 GENN 虽然结构更加简单, 但是性能与其他最先进的回归模型相当; 本节的 GENN-A* 在所有近似的 GED 求解器中表现最佳。需要指出的是, 当比较 GED 求解器和基于回归的深度学习模型^[161-162,182]时, “均方误差 (mse)” 可能不是一个公平的指标: 首先, GED 求解器可以预测编辑路径, 而回归模型不支持此功能。其次, GED 求解器的解是最优值的上界, 而基于回归的图形相似度模型可以预测最优值两侧的 GED 值。事实上, 只需对预测的 GED 值增加一个偏移量, 即可降低 GED 求解器的均方误差, 而这正是回归模型的特点。

图 4-5 汇总了 A* 算法 (算法 4-1) 中添加到搜索队列 OPEN 的状态总数。与 Hungarian-A* 相比, GENN-A* 的搜索树规模显著降低, 特别是当问题规模扩大时、或是当数据集从较简单的 LINUX 变为更难的 AIDS 和 Willow-Cars 时, 搜索树规模的降低更加明显。这一效应进一步缩短了 A* 算法的耗时, 如表 4-2 所示。上述实验结果表明, 本节提出的 GENN 算法学到了比匈牙利启发式更强的

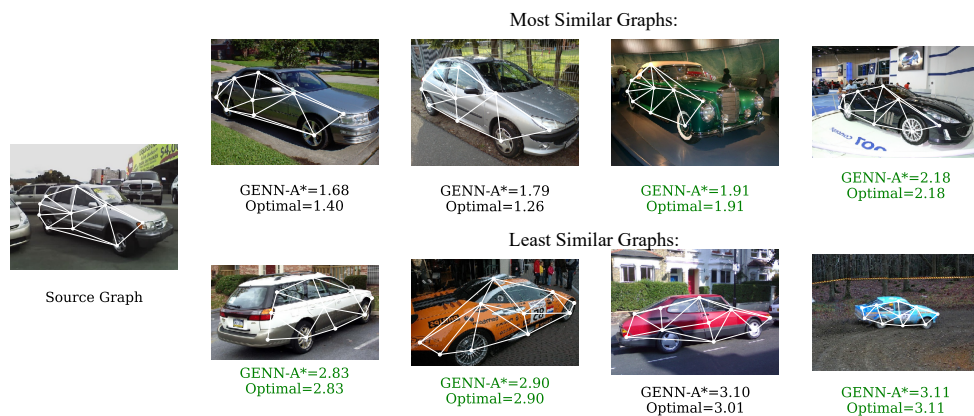


图 4-7 在 Willow-汽车数据集上使用 GENN-A* 检索的可视化结果。

Figure 4-7 The visualization of a query on Willow-Cars dataset by GENN-A*.

$h(p)$, 对冗余的搜索路径进行了剪枝。通过进一步比较三种动态图嵌入方法的推理时间, 如表4-3所示, 本节的 GENN 与传统的匈牙利启发式的单次调用耗时相当。由于过多的缓存与访存操作, 精确的动态 GNN 甚至比普通 GNN 更慢。

图 4-6比较了不同 GED 求解方法和启发式方法的散点图, 其中 x 轴代表最优 GED 值, y 轴代表求解 (或预测) 的 GED 值。最优解应当位于 $y = x$ 的黑色对角线上。对比两种求解器, GENN-A* (红色) 取得了比 Hungarian 近似求解器^[184] (蓝色) 更紧的上界, 且有相当数量的问题被求到了最优解。对比两种启发式预测方法, 本节的回归模型 GENN (青色) 也比 Hungarian 启发式^[38] (黄色) 更精确。图 4-7展示了采用 GENN-A* 查询 Willow-Car 图像的可视化结果, 查询到的 4 个最相似的图在朝向和结构上均与源图相近, 4 个最不相似的图在朝向和外观上变化很大。绿色代表 GENN-A* 求解到了最优的 GED。

4.3 与图匹配求解器融合的无监督训练框架

图匹配问题的重要性已在3.2节中得到阐述。近年来, 在图匹配研究中, (深度) 学习方法已经展示出相对传统求解器的优越性 (包括本文的3.2节、3.3节)。然而, 为了实现较高的匹配精度, 这些方法基本上都是基于监督学习, 这可能带来昂贵的标注成本。本节开发了一个统一的无监督学习框架, 涵盖二图匹配到多图匹配, 无需节点对应关系的标注即可进行训练。具体而言, 在“传统求解器指导深度神经网络”的融合范式下, 本节设计了一种孪生网络 (Siamese network) 形式的无监督学习方法, 通过最小化传统图匹配求解器和 (可微的) Sinkhorn 网络之间的差异, 进行端到端的无监督训练。在视觉图匹配任务中, 这两个分支共享相同的 CNN 主干网络。本节的框架进一步支持混合模式的无监督学习, 这种设定在现实中非常普遍。在传统算法层面, 本节开发了一种统一的渐近指派 (Graduated Assignment, GA) 算法, 它同时适用于二图匹配、多图匹配和来自混合模式的多图匹配。在自然图像匹配等真实世界场景上的实验结果表明, 与监督

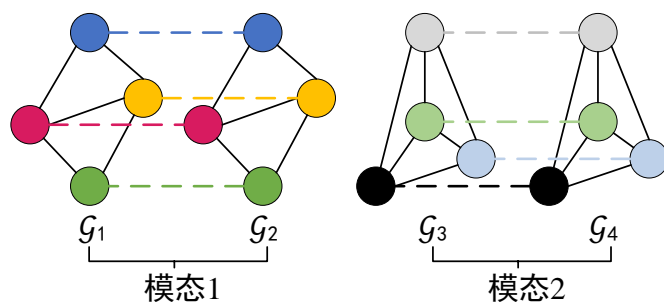


图 4-8 混合两种模态的匹配问题示例。

Figure 4-8 Example of matching with a mixture of two modes.

学习的二图匹配方法相比，本节的无监督方法表现相当、甚至更好。

4.3.1 混合模态、部分匹配和外点介绍

现有图匹配研究中广泛考虑的二图匹配、多图匹配场景已在3.2.1节进行了介绍。在此基础上，本节进一步考虑了多图匹配的扩展——混合模态多图匹配 (multi-graph matching with a mixture of modes, MGM³)。这是一个真实且颇具挑战的场景，其中图可能属于不同的类别，我们需要联合求解匹配问题和聚类问题，如图 4-8所示。在这个例子中，模态 1 和模态 2 内部的图可以完全匹配，模态 1 和模态 2 之间的匹配是无效的。除去非机器学习的前序工作^[206]，这一重要问题并未得到充分的研究。二图匹配、多图匹配和混合模态的多图匹配均是本节无监督学习方法的研究场景。

由于噪声、自遮挡等因素的存在，外点 (outlier) 和部分匹配 (partial matching) 也是实际匹配任务中常见的挑战。图4-9总结了在多图场景下，四种图匹配任务的差别 (相同颜色表示匹配关系)，包括：(a) 经典的完全匹配，其中任意图中的任意节点都能在所有图中找到对应；(b) 部分匹配，其中一部分内点被遮挡，所有节点都能在一个全集中找到对应 (此处全集有 4 个节点)；(c) 外点匹配，其中由噪声引起的外点无法在全集中被找到；(d) 同时包含部分匹配和外点匹配，这是最具挑战性、但是最真实的实验场景。多数现有的研究工作^[95,160]只考虑了 (a)，一些最新的研究工作^[82]考虑了 (b)。本节的方法能够同时处理 (a) (b) (d)。其中，(c) 比较少见，因为外点通常与部分匹配同时出现。

4.3.2 研究动机

在现有研究中已经取得重大突破的深度学习图匹配方法通常包含以下模块：感知模块 (例如 VGG16^[128])、图结构学习模块 (例如 GCN^[133]) 和相似度函数 (例如向量空间的相似度)。与固定的相似度函数和图像特征 (例如 SIFT 描述符^[207]) 相比，这些深度学习模块在颇具挑战的真实数据集上展现出了卓越的性能。然而，当前监督式的深度图匹配需要对大规模训练数据进行昂贵的标注，这

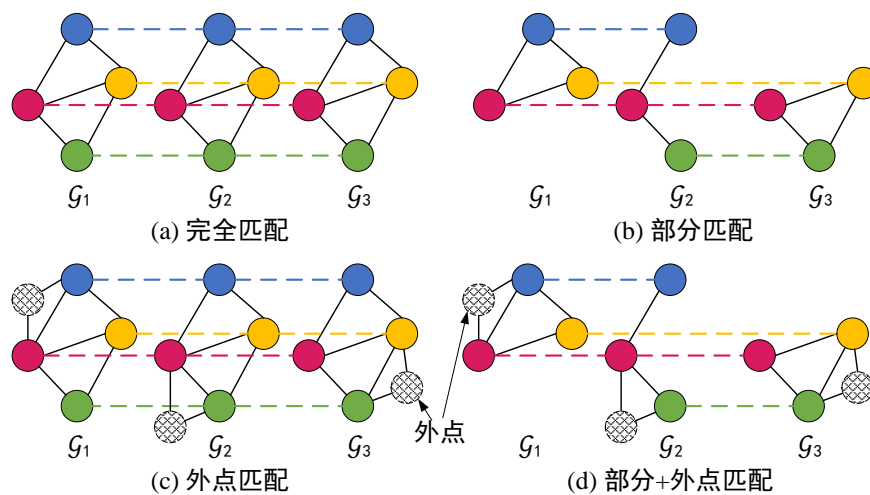


图 4-9 含外点、部分匹配的图匹配任务。

Figure 4-9 Graph matching with outliers and partial matching.

限制了深度图匹配方法的实际应用。尽管颇具吸引力，但是开发图匹配的无监督学习算法非常困难，因为图匹配方法通常涉及不可微分的离散化步骤。此外，对于大多数机器学习任务来说，进行无监督学习都是具有挑战性的。

观察到最近无监督图像分类模型^[208-209]的成功，作者发现它们具有一个相通的设计理念，即通过最小化两个分支在同一数据上预测值的差异实现无监督学习。受此启发，本节提出了一个简单而有效的无监督学习框架，它融合了传统图匹配求解器和深度学习模型：在差异最小化的框架下，以孪生网络^[210]的形式，不可微的传统求解器和可微分 Sinkhorn 分支共享一个 CNN 主干网络，训练的目标是最小化传统求解器与 Sinkhorn 分支的差异。此外，由于图匹配求解器同时利用了节点信息和结构信息进行匹配^[95]，它们的精度更高。为了形成一个统一的框架，本节采用了经典的渐近指派算法^[33]作为传统求解器，后文中将展示，它可以扩展到多种复杂的图匹配场景。在多图匹配中，通过联合考虑多个图结构可以得到更多的信息，这一信息同样可以在本节的无监督学习框架下得到利用。

具体而言，本节的研究动机来自如下几个方面：

1) 提出一个无监督的深度图匹配框架。与大多数基于监督学习的深度图匹配方法相反，本节通过融合传统图匹配求解算法和深度学习模型，实现无需监督的深度图匹配训练。虽然本节的实验主要在现有的测评数据集上进行，且这些数据集通常涉及视觉场景，但本节的无监督框架本身是通用的、与每个模块的具体技术独立。

2) 开发广泛适配的传统图匹配求解器。本节开发的传统图匹配求解器同时支持二图匹配、多图匹配、混合模态的多图匹配，具有一个统一的算法框架——渐近指派算法。渐近指派算法是求解排列型组合优化问题的经典算法^[32-33]，本节进一步扩展了渐近指派算法在多图匹配、混合模态匹配的应用场景。

3) 适配混合模态的多图匹配 (MGM³)。这是一个新兴且重要的图匹配场

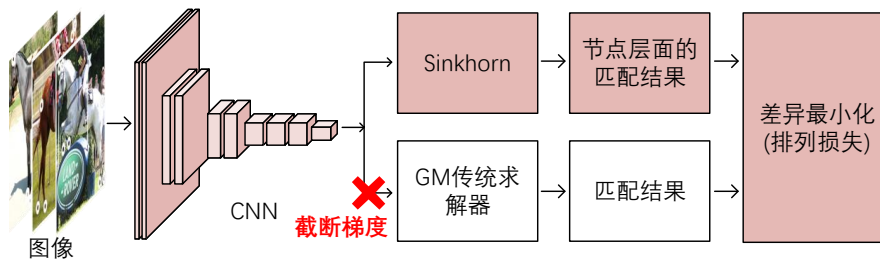


图 4-10 无监督的图匹配学习框架概览 (以视觉任务为例)。

Figure 4-10 An overview of the unsupervised (visual) GM learning pipeline.

景, DPMC^[206]作为目前仅有的混合模态多图匹配方法, 在方法中未考虑机器学习, 并且他们的聚类是在匹配之后进行的。相反, 观察到匹配和聚类本质上可以交互提升, 本节开发了一种 MGM³ 方法, 其中聚类和匹配随着逐渐增加的置信度交替进行。

4.3.3 算法描述

基于差异最小化的图匹配无监督学习框架。通过融合传统求解器和深度学习模型, 本节提出了一种基于差异最小化的无监督学习框架, 可用于通用的深度图匹配模型训练, 其中训练过程不需要匹配标注信息。具体而言, 本节考虑了三种图匹配问题的变体, 即二图匹配 (GM), 多图匹配 (MGM) 和混合模态的多图匹配 (MGM³)。这些问题设定涵盖了大多数现有的图匹配研究工作。

本节的无监督学习流程如图4-10所示, 其中白色框代表无梯度的模块、红色框代表有梯度的模块。白色框中的图匹配求解器可以是任意 GM、MGM 或 MGM³ 的求解器。在无监督框架中, 梯度截断可以有效地避免模型退化^[211]。本节考虑了 $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$, 并将 n_i 定义为 \mathcal{G}_i 中关键点的数量。本节的无监督框架包括以下模块:

- 特征提取。本节的无监督图匹配学习框架是一个面向计算机视觉的模型, 接受图像作为输入。图像经过 VGG16^[128]网络处理, VGG16 的权重在 ImageNet^[102]上预训练。与3.2节、3.3节的现有工作保持一致, VGG16 的 relu4_2 和 relu5_1 特征被拼接后用于表征节点特征。对每张图片, $\mathbf{F}_i \in \mathbb{R}^{n_i \times l}$ 代表使用 CNN 网络为了 n_i 个节点抽取得到的 l 维特征。
- 利用 Sinkhorn 计算节点匹配。首先, \mathcal{G}_i 和 \mathcal{G}_j 节点层面的匹配采用节点特征的内积计算得到:

$$\mathbf{W}_{ij} = \text{Sinkhorn}(\mathbf{F}_i \mathbf{F}_j^\top, \tau_w). \quad (4.9)$$

其中 $\text{Sinkhorn}(\mathbf{M}, \tau)$ 代表先作用熵正则项 $\mathbf{S} = \exp(\mathbf{M}/\tau)$, 随后采用公式 (3.14) 的流程, 依次进行行归一化和列归一化得到双随机矩阵。这一方法得到的解可被视为对匈牙利算法的可微分近似^[114]。其中, Sinkhorn 和匈牙利

利算法的差异由退火参数 τ 调控：当 τ 很小时，Sinkhorn 算法的输出和匈牙利相近（以收敛速度减慢为代价），当 τ 增大时，Sinkhorn 算法的输出将更加平滑^[32]。

在本文的无监督学习中，Sinkhorn 算法的输出 \mathbf{W}_{ij} 一方面被视为节点层面的匹配结果，用于计算无监督损失函数；另一方面被视为节点层面的相似度，进而被图匹配求解器接受和处理。

- 编码结构信息。如图4-10所示，图匹配求解器接受图像中的 CNN 特征和结构信息，并且将结构信息编码为图的边。作者模仿 Bernard 等^[92]的方法构建边：对于 G_i 的加权邻接矩阵，首先计算每对关键点之间的欧几里得距离 $l_{ab} = |p_a - p_b|$ ，其中 p_a, p_b 是关键点的坐标。 $\mathbf{A}_i \in \mathbb{R}^{n_i \times n_i}$ 是 G_i 的邻接矩阵，相应的 $\mathbf{A}_i[a, b]$ 的值计算如下：

$$\mathbf{A}_i[a, b] = \exp\left(-\frac{l_{ab}^2}{\sigma \hat{l}^2}\right), \quad (4.10)$$

其中 \hat{l} 是所有 l_{ab} 的中位数， \mathbf{A}_i 的对角线元素全是 0。 σ 是缩放参数。

- 融合图匹配求解器的差异最小化方法。许多计算机视觉领域的无监督学习模型采用了与本节类似的差异最小化框架^[208-209]。作者将这一范式进一步扩展到图匹配中。基于节点级的 CNN 特征和额外的结构信息（包括可能存在的多图一致性信息），图匹配求解器能够提供一个匹配结果，该匹配与 Sinkhorn 预测的节点级匹配不同。

由于 Sinkhorn 分支的匹配精度纯粹依赖于 CNN 特征的质量，利用 Sinkhorn 直接匹配可能是整个网络中最简单的匹配方式。图匹配求解器利用了更多的信息，因此理论上它的精度会比 Sinkhorn 更高。本节无监督学习的目标函数即为最小化这两个分支之间的差异（采用公式 (3.15) 中的排列损失函数衡量），即经过学习后，Sinkhorn 分支的匹配精度应该与图匹配求解器一样准确。由于 Sinkhorn 分支的精度纯粹基于 CNN 特征，更准确的 Sinkhorn 匹配意味着 CNN 特征的质量更高；而更优质的 CNN 特征也会促进求解器的性能提升，因为两个分支共享了一个 CNN 网络。通过最小化较弱的一阶匹配方法（Sinkhorn）和较强的二阶匹配方法（图匹配求解器）之间的差异，本节证明了这种无监督学习范式的有效性。最后，Sinkhorn 分支能够反向传播梯度，且大多数图匹配求解器是不可微的，求解器分支的梯度天然地被截断了。

值得注意的是，本节的无监督学习框架没有限制图匹配求解器的具体实现。本节主要采用了渐近指派算法，因为这种经典方法可以适用于所有目前已知的图匹配任务，包括二图匹配（GM）、多图匹配（MGM）和混合模态多图匹配

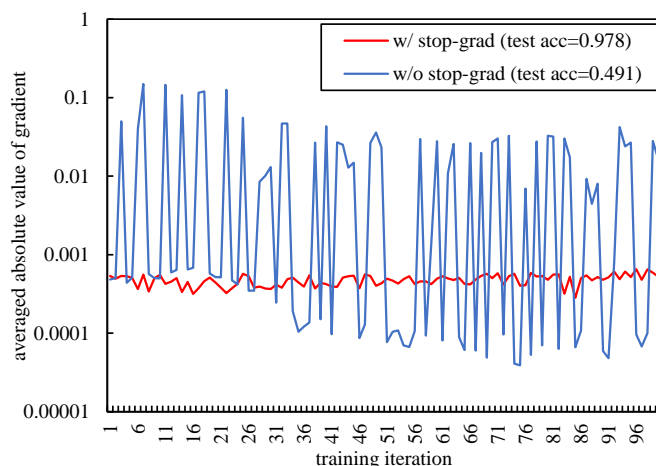


图 4-11 4.3节的无监督图匹配学习框架中，梯度截断的必要性。

Figure 4-11 The necessity of stop-gradient in unsupervised learning of graph matching in Sec. 4.3.

(MGM³)。由于截断梯度的必要性（见下面的讨论），求解器中的可学习参数无法得到更新，因此作者未采用基于机器学习的求解器。

讨论：梯度截断的必要性。本节提出的双分支结构可以看作是一种孪生网络（根据定义，孪生网络中两个分支的结构可以有所不同）。孪生网络在机器学习中的应用最早可以追溯到 Bromley 等的工作^[210]。在本节中，输入样本是一对待匹配的图，输出是匹配矩阵。这与许多图像分类任务中的嵌入特征不同。特别地，防止网络输出坍塌是无监督学习研究中的主要目标之一，本文的无监督方法既不需要负样本对^[208]，也无需使用动量编码器^[209]，即可避免无监督网络的坍塌。

如图4-10所示，本节的框架中只有 Sinkhorn 分支允许梯度反传，而求解器分支不存在可利用的梯度。从技术上讲，求解器的梯度只能通过额外的开销进行估计，例如借用 BBGM^[82]中的梯度近似技术。作者发现，本节的设计可以通过无监督学习获得高精度，而不会遇到网络坍塌的问题。作者认为，固定一个分支的无监督学习学习会更加稳定。当然，在技术上也可以对求解器分支求导，例如如图4-11中，作者参考了 BBGM^[82]的梯度近似技术在 Willow ObjectClass 数据集上进行了实验，未采取梯度截断的模型梯度波动更大，且最终的收敛精度更低。有趣的是，同期的另一项工作^[211]也采用了与本节类似的差异最小化的方法无监督地学习图像分类，他们的实验也证实了梯度截断思想的有效性。

基于渐近指派的图匹配求解器。上述无监督学习框架覆盖了经典的二图匹配^[33-35,88,212]、多图匹配^[91,94-95,127,213]、以及新兴的混合模态多图匹配^[206]任务。接下来，作者将在统一的渐近指派框架下，讨论如何利用传统算法求解 GM、MGM 和 MGM³ 问题。

二图匹配算法 GA-GM。渐近指派算法在图匹配的应用可以追溯到 1996 年，Gold 和 Rangarajan^[33]提出了通过迭代投影求解图匹配的方法。其中，每次投影都在最优化目标函数的一阶泰勒展开，并通过渐近因子调控约束的满足程度。基

算法 4-3 渐近指派式二图匹配算法 (GA-GM)

输入: 输入图 $\mathcal{G}_i, \mathcal{G}_j$; 节点相似度 \mathbf{W}_{ij} ; 初始退火参数 τ_0 ; 降温系数 γ ; 下界 τ_{min} ; 全集大小 d ; 外点阈值 ϕ .

```

1: 随机初始化  $\mathbf{X}_{ij}$ ; projector  $\leftarrow$  Sinkhorn;  $\tau \leftarrow \tau_0$ ;
2: if 存在部分匹配 then
3:    $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$ ; # 处理部分匹配
4: end if
5: while True do
6:   while  $\mathbf{X}_{ij}$  未收敛 AND #iter  $\leq$  #GMIter do
7:      $\mathbf{V}_{ij} \leftarrow \lambda \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j + \mathbf{W}_{ij}$ ; # 更新  $\mathbf{V}_{ij}$ 
8:      $\mathbf{X}_{ij} \leftarrow$  projector( $\mathbf{V}_{ij}, \tau$ ); # 将  $\mathbf{V}_{ij}$  投影到 (放松后的) 可行域
9:     if projector == Hungarian AND  $\phi > 0$  then # 处理外点匹配
10:       $\mathbf{Q}_{ij} \leftarrow \lambda \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j + \mathbf{W}_{ij}$ ;
11:       $\mathbf{X}_{ij} \leftarrow (\mathbf{Q}_{ij} < \phi) \odot \mathbf{X}_{ij}$ ;
12:    end if
13:   end while
14:   if projector == Sinkhorn AND  $\tau \geq \tau_{min}$  then # 渐近指派控制
15:      $\tau \leftarrow \tau \times \gamma$ ;
16:   else if projector == Sinkhorn AND  $\tau < \tau_{min}$  then
17:     projector  $\leftarrow$  Hungarian;
18:   else
19:     break;
20:   end if
21: end while

```

输出: 二图匹配矩阵 \mathbf{X}_{ij} 。

于深度学习模型提取得到的特征，本节采用公式 (3.2) 中的 KB-QAP 形式作为渐近指派的目标函数，其中带权的邻接矩阵 \mathbf{A}_i 已在公式 (4.10) 中定义，公式 (3.2) 中的节点相似度矩阵 \mathbf{K}_p 由 \mathbf{W}_{ij} 表达。 \mathbf{X}_{ij} 中的每个元素都通过 $1/d + 10^{-3}z$ 进行初始化，其中 $z \sim \mathcal{N}(0, 1)$ 采样自标准正态分布。

渐近指派算法通过渐近式的 Sinkhorn 和匈牙利指派，将公式 (3.2) 中的目标函数逐步投影到可行空间中。对于非方阵，可以用负无穷填充成方阵。当 τ 减小时，Sinkhorn 算法的输出会越来越接近匈牙利算法。因此，通过逐步减小 τ ，Sinkhorn 可以让输入矩阵逐步接近排列矩阵，再通过匈牙利方法得到匹配结果。这种流程类似于模拟退火。在算法 4-3 中，退火速度由 $\gamma < 1$ 调控， $\tau \leftarrow \tau \times \gamma$ ，且存在一个下限 τ_{min} 。

多图匹配算法 GA-MGM。Solé 和 Serratos^[213] 曾讨论过将渐近指派算法用于多图匹配，在他们的论文中，需要一个“原型”图作为锚定（他们使用第一个图作为“原型”）。这意味着他们假设任意二图之间存在双射（即完全匹配），但这在实践中很难满足。相比之下，本节的改进方法完全是去中心化的，不需要这一假设，并且可以处理部分匹配和外点匹配的情况。此外，本节的方法还能够扩展到混合模态的匹配问题。这些灵活性使得本算法更加实用。特别地，通过对所

算法 4-4 渐近指派式的多图匹配算法 (GA-MGM)

输入: 输入图 $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$; 节点相似度 $\{\mathbf{W}_{ij}\}$; 初始退火参数 τ_0 ; 降温系数 γ ; 下界 τ_{min} ; 全集大小 d ; 外点阈值 ϕ ; 聚类权重 \mathbb{B} (如果不考虑聚类, 则所有 $\mathbb{B}_{ij} = 1$).

```

1: 随机初始化  $\{\mathbf{U}_i\}$ ; projector  $\leftarrow$  Sinkhorn;  $\tau \leftarrow \tau_0$ ;
2: if 存在部分匹配 then
3:    $\mathbf{A}_i = \mathbf{A}_i \times \frac{d}{n_i}$ ; # 处理部分匹配
4: end if
5: while True do
6:   while  $\{\mathbf{U}_i\}$  未收敛 AND #iter  $\leq$  #GMIter do
7:      $\forall i \in [m], \mathbf{V}_i \leftarrow \mathbf{0}$ ;
8:     for  $\mathcal{G}_i, \mathcal{G}_j$  in  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$  do
9:        $\mathbf{V}_i \leftarrow \mathbf{V}_i + (\lambda \mathbf{A}_i \mathbf{U}_i \mathbf{U}_j^\top \mathbf{A}_j \mathbf{U}_j + \mathbf{W}_{ij} \mathbf{U}_j) \times \mathbb{B}_{ij}$ ; # 更新  $\mathbf{V}_i$ 
10:    end for
11:    for  $\mathcal{G}_i$  in  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$  do
12:       $\mathbf{U}_i \leftarrow$  projector( $\mathbf{V}_i, \tau$ ); # 将  $\mathbf{V}_i$  投影到 (放松后的) 可行域
13:    end for
14:    if projector == Hungarian AND  $\phi > 0$  then # 处理外点匹配
15:      for  $\mathcal{G}_i$  in  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$  do
16:         $\mathbf{Q}_i \leftarrow \sum_{j \neq i} \lambda \mathbf{A}_i \mathbf{U}_i \mathbf{U}_j^\top \mathbf{A}_j \mathbf{U}_j + \mathbf{W}_{ij} \mathbf{U}_j$ ;
17:         $\mathbf{U}_i \leftarrow (\mathbf{Q}_i < \phi) \odot \mathbf{U}_i$ ;
18:      end for
19:    end if
20:    if projector == Sinkhorn AND  $\tau \geq \tau_{min}$  then # 渐近指派控制
21:       $\tau \leftarrow \tau \times \gamma$ ;
22:    else if projector == Sinkhorn AND  $\tau < \tau_{min}$  then
23:      projector  $\leftarrow$  Hungarian;
24:    else
25:      break;
26:    end if
27:  end while
28: end while

```

输出: 多图匹配矩阵的集合 $\{\mathbf{U}_i\}$.

有二图匹配的 KB-QAP 目标函数求和, MGM 的目标函数定义如下:

$$\max_{\mathbf{X}_{i,j}, i,j \in [m]} \sum_{i,j \in [m]} \left(\lambda \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) + \operatorname{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij}) \right). \quad (4.11)$$

在实践中, \mathbf{X}_{ij} 被分解为 $\mathbf{X}_{ij} = \mathbf{U}_i \mathbf{U}_j^\top$, $i, j \in [m]$ 代表迭代所有 i, j 的组合。为了行文整洁, 公式 (4.11) 没有包括排列约束。采取这种分解形式是为了利用多图匹配中的循环一致性:

定义 4.2 (循环一致性^[95], cycle-consistency). 在部分匹配场景下, 我们称 $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m$

算法 4-5 渐近指派式混合模态的多图匹配算法 (GA-MGM³)

输入: 输入图 $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$; 节点相似度 $\{\mathbf{W}_{ij}\}$; 聚类权重 $\{\beta\}$.

- 1: 所有 $\mathbb{B}_{ij} \leftarrow 1$; # 初始化聚类权重矩阵
- 2: **for** β in $\{\beta\}$ **do**
- 3: **while** $\{\mathbf{U}_i\}, \mathbb{C}$ 未收敛 AND #iter \leq #ClsIter **do**
- 4: $\{\mathbf{U}_i\} \leftarrow$ GA-MGM($\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}, \{\mathbf{W}_{ij}\}, \mathbb{B}$); # 多图匹配
- 5: **for** $\mathcal{G}_i, \mathcal{G}_j$ in $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$ **do**
- 6: build \mathbf{X}_{ij} from \mathbf{U} ;
- 7: $\Delta_{ij} \leftarrow \lambda_c \exp(-\|\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} - \mathbf{A}_j\|) + \text{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij})$; # 图间相似度
- 8: **end for**
- 9: $\mathbb{C} \leftarrow$ 在 \mathbf{A} 上 k-means++ 的聚类结果;
- 10: $\mathbb{B} \leftarrow \mathbb{C} \times (1 - \beta) + \beta$; # 聚类权重
- 11: **end while**
- 12: **end for**

输出: 多图匹配矩阵的集合 $\{\mathbf{U}_i\}$; 聚类矩阵 \mathbb{C} .

的匹配是循环一致的, 当且仅当

$$\mathbf{X}_{ij} \geq \mathbf{X}_{ik} \mathbf{X}_{kj}, \quad \forall i, j, k \in [m], \quad (4.12)$$

其中 \geq 代表逐元素的比较, $[m]$ 代表从 1 到 m 所有图的下标。

如果存在一个大小为 d 的全集, 那么所有匹配关系可以表达为每个图对全集的匹配, 即 $\mathbf{U}_i \in \{0, 1\}^{n_i \times d}$ 。如果任意的两两匹配都可分解为 $\mathbf{X}_{ij} = \mathbf{U}_i \mathbf{U}_j^\top$, 那么公式 (4.12) 定义的循环一致性即可得到满足。因此, 本节的多图匹配算法在 \mathbf{U}_i 上进行处理。

算法4-4中的渐近指派多图匹配 (GA-MGM) 可以看作 GA-GM 在多图上的泛化。其中, 匹配矩阵 \mathbf{X}_{ij} 采用面向全集的匹配 $\mathbf{U}_i, \mathbf{U}_j$ 取代, 保证了循环一致性。聚类权重矩阵 \mathbb{B} 在多图匹配问题中未使用, 可以使用全 1 填充。 $\{\mathbf{U}_i\}$ 中的每个元素采用 $1/d + 10^{-3}z$ 随机初始化, 其中 $z \sim \mathcal{N}(0, 1)$ 。作为对比, 现有的多图匹配方法需要两两匹配结果初始化^[125,127,152], 或使用多图匹配结果初始化^[92], 这些更复杂的初始化方法需要更大的计算开销。

混合模态多图匹配算法 GA-MGM³。本节提出的渐近指派式混合模态多图匹配算法 (GA-MGM³) 以 GA-MGM 为基础, 它的目标函数在公式 (4.11) 的基础上添加了聚类变量:

$$\max_{\mathbf{X}_{ij}, \mathbb{C}} \sum_{i,j \in [m]} \mathbb{C}_{ij} \left(\lambda \text{tr}(\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} \mathbf{A}_j) + \text{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij}) \right). \quad (4.13)$$

其中 \mathbb{C} 是聚类的指示变量, 当 $\mathcal{G}_i, \mathcal{G}_j$ 属于同一类时 $\mathbb{C}_{ij} = 1$, 否则 $\mathbb{C}_{ij} = 0$ 。GA-MGM³ 算法设计了两个退火参数, 分别针对匹配和聚类, 能够同时完成模态区分 (即聚类) 和匹配。其中, 聚类步骤使用了经典的聚类算法 k-means++^[214]。作者

注意到,更精确的多图匹配将提高聚类准确性,反之亦然。受此启发,GA-MGM³将聚类和多图匹配交替进行,并逐步增加聚类的置信度直至收敛。GA-MGM³的算法流程总结在算法4-5中,主要包含以下两步:

- 基于匹配的聚类。处理 MGM³ 场景的核心挑战是构建一种合适的图间相似度量,随后即可采用标准的谱聚类方法。面对这一挑战,本节提出了一种基于匹配的图间相似度。

给定来自多个模态的多个图结构,可以先假设它们属于同一模态,随后采用现有的多图匹配求解器(例如算法4-4)求得匹配关系。对于来自同一模态的图,它们点、边的匹配程度应该更高;对于来自不同模态的图则相反。对于 $\mathcal{G}_i, \mathcal{G}_j$, 它们的匹配程度采用如下公式计算:

$$\Delta_{ij} = \underbrace{\lambda_c \exp(-\|\mathbf{X}_{ij}^\top \mathbf{A}_i \mathbf{X}_{ij} - \mathbf{A}_j\|_1)}_{\text{边匹配程度}} + \underbrace{\text{tr}(\mathbf{X}_{ij}^\top \mathbf{W}_{ij})}_{\text{节点匹配程度}}. \quad (4.14)$$

其中第一项衡量了邻接矩阵(即边)的匹配程度,第二项衡量了节点的匹配程度。此处还引入了一个权重系数 λ_c 以平衡这两项。得到 Δ 后,即可使用基于 k-means++^[214]的谱聚类方法区分不同模态。基于公式(4.14),对于同模态的图 $\mathcal{G}_i, \mathcal{G}_j$, 更精确的 \mathbf{X}_{ij} 将会使 Δ_{ij} 更大,因此更精确的匹配结果将会使提升聚类的精度。

- 面向聚类的匹配。基于公式(4.13)中为 MGM³ 设计的 KB-QAP 目标函数,聚类权重 \mathbb{B}_{ij} 可以在 GA-MGM 算法的投影步骤得到考虑(见算法4-4中的第9行)。若假设聚类的精度为 100%, 则只需设定 $\mathbb{B} = \mathbb{C}$, 投影步骤即符合公式(4.13)的目标函数。不过,在实际情况下,聚类很可能不是最优的,作者进一步为 \mathbb{B} 添加一个退火参数 β : $\mathbb{B}_{ij} = \begin{cases} 1 & \text{if } \mathbb{C}_{ij} = 1 \\ \beta & \text{if } \mathbb{C}_{ij} = 0 \end{cases}$ 该式等价于 $\mathbb{B} = \mathbb{C} \times (1 - \beta) + \beta$ 。退火参数 $\beta \in [0, 1]$ 可以被看作聚类的置信度,通过逐渐将 β 从 1 下降到 0, 即可求解 MGM³ 问题。

渐近指派算法的理论分析。此处以 GA-MGM 为例,讨论渐近指派算法的理论基础。GA-GM 可以看作是 GA-MGM 的特殊情况,将图的数量限制为 2; 对 GA-MGM³ 的推导可以简单地将聚类系数相乘实现。公式(4.11)的多图 KB-QAP 目标函数简写为 J 。设定 $\mathbf{X}_{ij} = \mathbf{U}_i^\top \mathbf{U}_j$, 对于一组可行的多图匹配解 \mathbf{U}_i^0 , 对 J 进行泰勒展开:

$$J = \sum_{i,j \in [m]} \lambda \text{tr}(\mathbf{U}_j^0 \mathbf{U}_i^{0\top} \mathbf{A}_i \mathbf{U}_i^0 \mathbf{U}_j^{0\top} \mathbf{A}_j) + \text{tr}(\mathbf{U}_j^0 \mathbf{U}_i^{0\top} \mathbf{W}_{ij}) + \sum_{i \in [m]} \text{tr}(\mathbf{V}_i^\top (\mathbf{U}_i - \mathbf{U}_i^0)) + \dots, \quad (4.15)$$

表 4-4 Willow ObjectClass 数据集上, 非学习 MGM 算法、监督学习方法和无监督方法的匹配精度。

Table 4-4 Matching accuracy with both learning-free MGM methods, supervised learning peer methods and unsupervised learning methods on Willow ObjectClass dataset.

图匹配方法	学习方式	car	duck	face	mbike	wbottle	平均
MatchLift ^[125]	非学习	0.665	0.554	0.931	0.296	0.700	0.629
MatchALS ^[152]	非学习	0.629	0.525	0.934	0.310	0.669	0.613
MSIM ^[91]	非学习	0.750	0.732	0.937	0.653	0.814	0.777
MGM-Floyd ^[160]	非学习	0.850	0.793	1.000	0.843	0.931	0.883
HiPPI ^[92]	非学习	0.740	0.880	1.000	0.840	0.950	0.882
VGG16+Sinkhorn	非学习	0.776	0.732	0.990	0.525	0.824	0.769
GA-GM	非学习	0.814	0.885	1.000	0.809	0.954	0.892
GA-MGM	非学习	0.746	0.900	0.997	0.892	0.937	0.894
HARG-SSVM ^[101]	监督学习	0.584	0.552	0.912	0.444	0.666	0.632
GMN ^[16]	监督学习	0.743	0.828	0.993	0.714	0.767	0.809
PCA-GM (3.3节)	监督学习	0.840	0.935	1.000	0.767	0.969	0.902
DGMC ^[78]	监督学习	0.903	0.890	1.000	0.921	0.971	0.937
CIE-H ^[77]	监督学习	0.822	0.812	1.000	0.900	0.976	0.902
BBGM ^[82]	监督学习	0.968	0.899	1.000	0.998	0.994	0.972
NMGM (3.3节)	监督学习	0.976	0.945	1.000	1.000	0.990	0.982
GANN (with HiPPI ^[92])	无监督	0.845	0.868	0.992	0.911	0.954	0.914
GANN-GM	无监督	0.854	0.898	1.000	0.886	0.964	0.920
GANN-MGM	无监督	0.964	0.949	1.000	1.000	0.978	0.978

其中

$$\mathbf{V}_i = \frac{\partial J}{\partial \mathbf{U}_i} \Big|_{\mathbf{U}_i = \mathbf{U}_i^0} = \sum_{j \in [m]} \left(2\lambda \mathbf{A}_i \mathbf{U}_i^0 \mathbf{U}_j^{0\top} \mathbf{A}_j \mathbf{U}_j^0 + \mathbf{W}_{ij} \mathbf{U}_j^0 \right). \quad (4.16)$$

上式中, 除了 \mathbf{V}_i , 所有项都是常数。用一阶泰勒展开逼近 J , 则最大化 J 等价于最大化 $\sum_{i \in [m]} \text{tr}(\mathbf{V}_i^\top \mathbf{U}_i)$, 这等价于求解 m 个独立的线性指派问题。因此, 给定初始的 \mathbf{U}_i , GA-MGM 最优化了目标函数的一阶泰勒展开, 并通过公式 (4.15) 计算 \mathbf{V}_i 。随后得到的线性指派问题可以通过 Sinkhorn 算法 (由 τ 调控) 或匈牙利算法^[190]来求解, 求解器的近似程度由退火参数 τ 调控^[32], 以确保 \mathbf{U}_i 逐渐收敛到高质量的离散解。关于渐近指派的收敛性分析, 请参见文献^[215]。在实现中省略了常数 2, 因为它可以被参数 λ 吸收。

4.3.4 实验分析

实验设定和评测标准。与 3.2 节、3.3 节一致, 本实验以匹配精度作为评价指标。特别地, 已知预测的排列阵 \mathbf{X} 和真值 \mathbf{X}^{gt} , 实验考虑了如下指标:

$$\text{准确率 precision} = \frac{\text{tr}(\mathbf{X}^\top \mathbf{X}^{gt})}{\text{sum}(\mathbf{X})}, \quad (4.17)$$

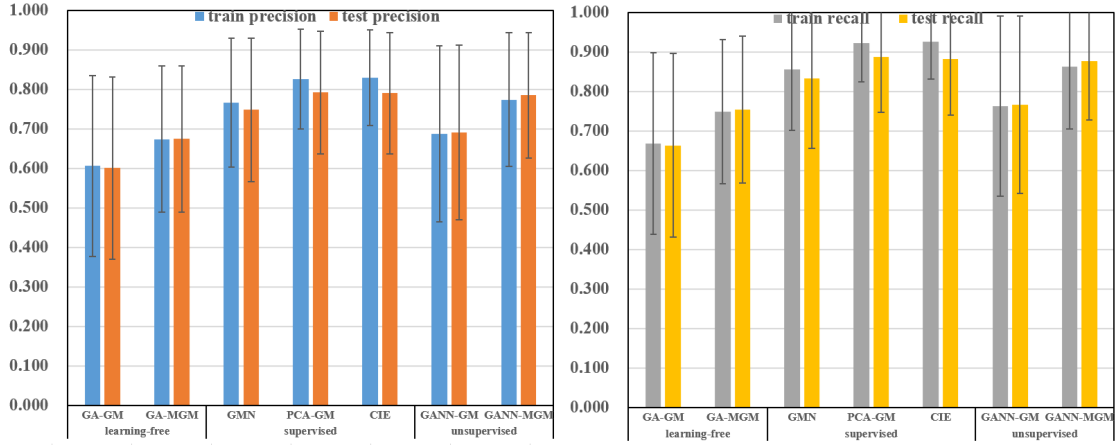


图 4-12 CUB2011 数据集上的单模态图匹配的准确率和召回率。

Figure 4-12 Precision and recall of graph matching with a single mode on CUB2011 dataset.

$$\text{召回率 recall} = \frac{\text{tr}(\mathbf{X}^\top \mathbf{X}^{gt})}{\text{sum}(\mathbf{X}^{gt})}, \quad (4.18)$$

$$\text{F1 得分 f1} = \frac{\text{precision} + \text{recall}}{2}. \quad (4.19)$$

需要注意的是，如果不存在外点或部分匹配，则有“准确率” = “召回率” = “F1 得分”。本文将这种情况下的指标统称为“匹配精度”。

对于 MGM³ 任务，实验考虑了类内的匹配精度。此外，还考虑了如下的聚类指标：

1. 聚类纯度^[216] (Clustering Purity, CP)，其中 C_i 代表预测的类 i ， C_j^{gt} 代表真实的类 j ：

$$\text{CP} = \frac{1}{m} \sum_{i=1}^k \max_{j \in \{1, \dots, k\}} |C_i \cap C_j^{gt}| \quad (4.20)$$

2. 兰德系数^[217] (Rand Index, RI) 的计算对以下两项求和：实际来自同一个类且被分到了同一类的数目 n_{11} ，以及实际来自不同类、被分到了不同类的数目 n_{00} ，最后除以总的图数目 n ：

$$\text{RI} = \frac{n_{11} + n_{00}}{n} \quad (4.21)$$

3. 聚类精度^[206] (Clustering Accuracy, CA) 中，定义 $\mathcal{A}, \mathcal{B}, \dots$ 是真实的类集合， $\mathcal{A}', \mathcal{B}', \dots$ 是预测的类集合， k 是类的数目：

$$\text{CA} = 1 - \frac{1}{k} \left(\sum_{\mathcal{A}} \sum_{\mathcal{A}' \neq \mathcal{B}'} \frac{|\mathcal{A}' \cap \mathcal{A}| |\mathcal{B}' \cap \mathcal{A}|}{|\mathcal{A}| |\mathcal{A}'|} + \sum_{\mathcal{A}'} \sum_{\mathcal{A} \neq \mathcal{B}} \frac{|\mathcal{A}' \cap \mathcal{A}| |\mathcal{A}' \cap \mathcal{B}|}{|\mathcal{A}'| |\mathcal{B}|} \right) \quad (4.22)$$

表 4-5 *Pascal VOC* 关键点数据集 (不过滤节点) 的 *F1* 分数 (%) 结果。
Table 4-5 *F1* scores (%) on *Pascal VOC* Keypoint dataset (without filtering).

模型名称	学习?	主干网络	参数量	✈️	🚗	🚲	👤	👶	👦	👧	👨	👩	👪	👶	👦	👧	👨	👩	👪	👶	👦	👧	👨	👩	👪	平均精度
GMN ^[16]	监督	VGG16	12.9M	28.0	55.0	33.1	27.0	79.1	52.2	26.0	40.2	28.4	36.0	29.8	33.7	39.4	43.0	22.1	71.8	30.8	25.9	58.8	78.0		41.9	
PCA-GM ^[219]	监督	VGG16	41.7M	27.5	56.5	36.6	27.7	77.8	49.2	23.9	42.3	27.4	38.2	38.7	36.5	39.3	42.8	25.6	74.3	32.6	24.7	51.5	74.3		42.4	
GANN-GM	无监督	VGG16	12.4M	12.6	19.5	16.6	18.5	41.1	32.4	19.3	12.3	24.3	17.2	38.0	12.2	15.9	18.2	19.4	35.5	14.8	15.4	41.5	60.8		24.3	
GANN-MGM	无监督	VGG16	12.4M	18.1	33.4	20.2	28.2	71.7	33.9	22.0	24.7	23.5	22.4	50.2	19.6	20.9	27.7	19.5	74.5	19.3	26.5	39.8	72.7		33.4	
		ResNet34	21.3M	16.4	37.4	23.2	29.5	80.8	33.6	24.1	17.8	26.9	22.6	53.4	16.9	21.8	30.1	23.3	86.0	19.7	31.7	39.0	71.5		35.3	
NGM	监督	VGG16	71.4M	45.9	66.6	57.2	47.3	87.4	64.8	50.5	59.9	39.7	60.9	42.1	58.3	58.5	61.9	44.6	94.5	50.1	35.2	73.1	82.1		59.0	
		ResNet34	77.9M	45.1	65.5	52.7	44.0	87.3	69.4	56.1	62.2	45.7	63.6	61.9	59.6	59.2	67.8	54.4	96.9	57.0	45.9	74.3	83.6		62.6	
BBGM ^[82]	监督	VGG16	71.4M	41.3	65.7	55.0	43.4	86.8	61.1	35.5	59.0	40.2	60.0	29.7	57.1	57.5	65.9	37.7	95.8	52.6	30.3	74.4	84.1		56.7	
		ResNet34	77.9M	38.1	69.1	54.2	45.0	87.0	74.7	43.3	62.3	48.3	63.7	63.8	60.9	60.4	65.4	50.2	97.1	56.2	45.9	78.4	82.2		62.3	
GANN-MGM+NGM	无监督	VGG16	71.4M	47.0	69.4	53.7	46.3	85.7	67.6	59.0	60.2	45.9	61.0	29.9	57.9	59.5	63.2	47.4	92.2	51.5	39.9	71.6	78.3		59.4 (+0.4)	
	+监督	ResNet34	77.9M	46.5	66.2	56.5	46.5	85.9	73.8	57.4	61.4	47.3	65.7	63.9	59.4	60.1	70.6	54.7	94.3	57.0	51.8	74.9	82.4		63.8 (+1.2)	
GANN-MGM+BBGM ^[82]	无监督	VGG16	71.4M	43.0	69.2	55.3	46.3	85.4	66.7	53.3	61.4	46.7	64.0	33.7	61.7	60.7	64.0	43.9	94.0	54.9	52.3	78.0	80.0		60.7 (+4.0)	
	+监督	ResNet34	77.9M	39.6	68.9	56.2	46.7	87.4	71.9	44.5	62.0	47.9	65.8	50.8	62.6	61.6	62.9	49.4	96.9	59.6	49.7	80.6	85.3		62.5 (+0.2)	

本节的方法有两个版本, GA-GM/MGM/MGM³ 代表非学习的纯求解器, GANN-GM/MGM/MGM³ 代表经过无监督训练后的神经网络融合求解器。实验同时汇报了两个版本的精度。

实验数据集。本节的实验考虑了图匹配任务常用的 *Pascal VOC* 关键点数据集和 *Willow ObjectClass* 数据集, 这两个数据集已在3.2节中介绍。此外, 还考虑了 *CUB2011* 数据集^[218], 它是一个大规模的图像匹配数据集, 包含了 200 种、11,788 张鸟类照片, 每个类的训练、测试数据按 50%/50% 划分。*CUB2011* 中的关键点可能被自遮挡, 因此该数据集需要处理部分匹配的问题 (平均每张图片有 12 个点, 全集有 15 个点)。数据集中的鸟类可能处在飞行、站立、游泳等不同姿态, 图像的光照、背景均有差别。因此, *CUB2011* 数据集的匹配难度较高。

单模态下的图匹配实验结果。单模态下的图匹配是经典的实验设定, 其中所有图属于相同的类别, 该设定覆盖了 GM、MGM 问题。实验比较了无监督学习方法 GANN-GM 和 GANN-MGM、非学习的求解器^[91-92,125,152,160]以及基于监督学习的图匹配方法^[16,77-78,82,101] (包括3.2节、3.3节)。

Willow ObjectClass 的实验结果。本节在 *Willow ObjectClass* 数据集上的评测方式遵循了现有工作^[160], 同时比较了非学习和机器学习方法, 结果如表 4-4 所示。表 4-4 的上半部分是非学习的传统求解器, 下半部分是机器学习求解器。GANN-MGM 模型支持无监督训练, 在训练阶段每个类别随机选择了 8 个图, 在测试阶段同时匹配了单个类中的所有图 (因为数据集规模相对较小)。多数现有的机器学习图匹配方法都是面向二图匹配的, 除了 NMGM (3.3节) 外, 没有更多的机器学习多图匹配方法。与其他非学习的求解器相比, GA-MGM 与其他最先进的多图求解器的表现一样出色。最重要的是, 通过无监督学习, GANN-MGM 超过了多数监督学习的二图匹配方法, 与最精确的监督学习方法 NMGM 精度接近。由于没有限制图 4-10 中图匹配求解器的具体实现方式, 作者还实现并比较了另一种无监督学习方法——使用 HiPPI^[92] 替换 GA-MGM, 它的匹配精度也超过了几种机器学习图匹配方法。

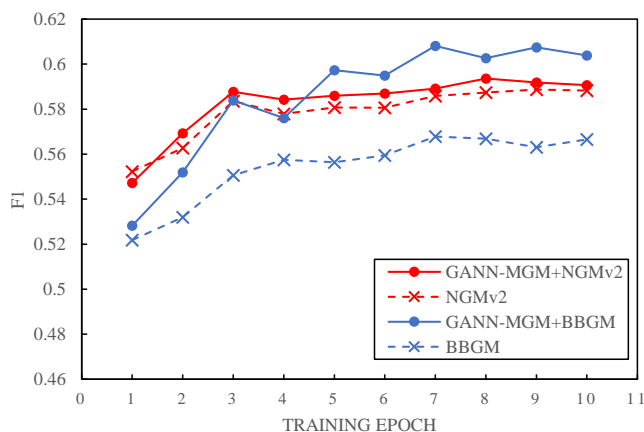


图 4-13 通过不同方式初始化 VGG16 网络后, NGM 和 BBGM^[82] 模型的训练曲线。

Figure 4-13 Training curves of NGM and BBGM^[82] after being initialized by different VGG16 networks.

CUB2011 的实验结果。CUB2011 比 Willow ObjectClass 难度更高, 因为图的节点数目更多, 且需要应对部分匹配的挑战。由于非学习的 GA-MGM 与其他最新的 MGM 方法^[91-92,160]在 Willow ObjectClass 上性能相当, 实验中主要考虑了基于监督学习的深度图匹配方法 (仅适用于二图, 因为多图机器学习方法 NMGM 不支持部分匹配)。对于 GANN-MGM 方法, 训练阶段在每个类随机选择 8 个图, 以节省时间和内存, 在测试阶段, 同一类别的所有图会被一起匹配。实验遵循了原始数据集的训练/测试集分割方式。如图4-12所示, GANN-GM 的性能稍逊色一些, 而 GANN-MGM 在测试集上与需要监督训练的 PCA-GM (3.2节)、CIE^[77]表现相当。此外, 无监督学习的方法在训练样本和未见过的测试样本上均有较好的性能, 而监督学习方法会在训练数据上过拟合。

Pascal VOC 关键点的实验结果。与本节的其他实验保持一致, Pascal VOC 关键点数据集基于原始的数据集标签、不过滤关键点, 在部分匹配的设定下进行实验。实验考虑了至多 5 个图进行多图联合匹配。GMN^[16]、BBGM^[82]、PCA-GM (3.2节)、NGM (3.3节) 等最新的 (监督学习) 深度二图匹配方法也被一并比较, 结果如表 4-5 所示。实验发现, 由于背景、光照、姿态等变化较大, Pascal VOC 关键点数据集的匹配难度较高, 训练标签在这种情况下非常重要。本节的无监督学习方法直接测试的表现不如监督学习。实验进一步研究了一种新的图匹配学习范式: 首先通过无监督学习方法预训练 CNN 权重, 然后再通过标签进行有监督微调。实验考虑了最先进的监督学习模型 BBGM^[82]和 NGM (3.3节)。对于 GANN-MGM+NGM 和 GANN-MGM+BBGM, 它们的 CNN 主干网络通过无监督学习模型进行初始化, 而其他网络层 (SplineConv^[147]层和 NGM 的求解器层) 则进行随机初始化。原始的 BBGM、NGM 的训练流程直接采用了在 ImageNet^[102]分类任务上预训练的 CNN 权重进行初始化, 而分类任务似乎与匹配任务之间存在一定

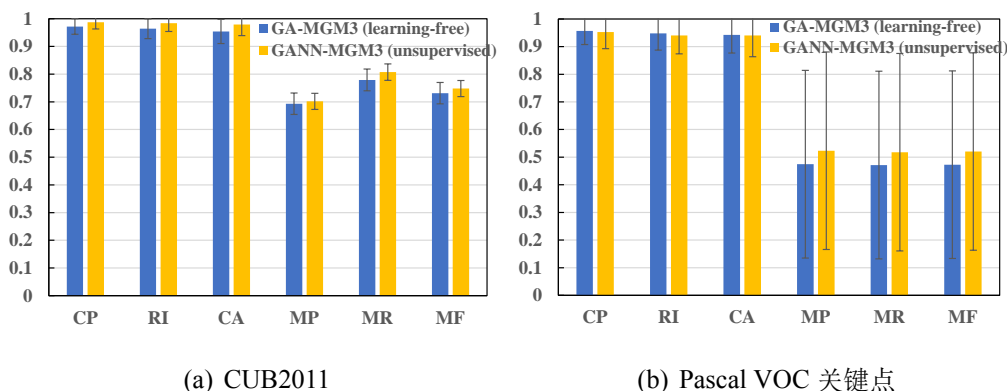


图 4-14 在 CUB2011 和 Pascal VOC 关键点数据集上 GA/GANN-MGM³ 的实验结果。

Figure 4-14 Results of GA/GANN-MGM³ on CUB2011 and Pascal VOC Keypoint datasets

表 4-6 混合模态多图匹配 (MGM³) 在 Willow ObjectClass 数据集上的实验结果。

Table 4-6 Multiple graph matching with mixture of modes (MGM³) evaluation on Willow ObjectClass dataset.

方法名称	学习	8 Cars, 8 Ducks, 8 Motorbikes					40 Cars, 50 Ducks, 40 Motorbikes				
		CP	RI	CA	MA	时间 (s)	CP	RI	CA	MA	时间 (s)
RRWM ^[34]	非学习	0.879	0.871	0.815	0.748	0.4	0.962	0.949	0.926	0.751	8.8
CAO-C ^[95]	非学习	0.908	0.903	0.860	0.878	3.3	0.971	0.960	0.956	0.906	1051.5
CAO-PC ^[95]	非学习	0.887	0.883	0.831	0.870	1.8	0.971	0.960	0.956	0.886	184.0
DPMC ^[206]	非学习	0.931	0.923	0.890	0.872	1.2	0.969	0.959	0.948	0.941	97.5
GA-MGM ³	非学习	0.921	0.905	0.893	0.653	10.6	0.890	0.871	0.850	0.669	107.8
GANN-MGM ³	无监督	0.976	0.970	0.963	0.896	5.2	0.974	0.968	0.956	0.906	80.7

的差异。图 4-13 汇报了基于 VGG16 主干网络的模型在训练过程中的匹配精度。可以看到，采用 GANN-MGM 初始化的模型收敛得更快、更好。此外，目前现有的图匹配深度学习方法均采用了 VGG16^[128] 作为主干网络。为了比较的公平性，本节的主要模型均基于 VGG16 开发。在此基础上，作者还考虑了 ResNet34^[220] 作为 CNN 主干网络。一方面，ResNet（残差网络）提供了更高的模型容量，从而实现了更好的匹配精度；另一方面，无监督学习在 ResNet 上的成功也证明了本节的自监督框架较广泛的适用范围。

混合模态下的图匹配实验结果。进一步地，实验探究了多个图分别归属于多个模态的情况。在这种情况下，求解器需要同时处理匹配和聚类任务。据作者所知，到目前为止不存在其他基于机器学习的 MGM³ 算法，本实验主要比较了 DPMC^[206] 方法，并采用了该论文的实验环境。

Willow ObjectClass 的实验结果。遵循论文^[206] 的实验方案，作者考虑了 3 个模态（Cars、Ducks、Motorbikes）混合的问题。除了专用于 MGM³ 的求解器 DPMC^[206]，实验还比较了两种常用的单模态图匹配求解器，它们分别面向于二图匹配^[34] 和多图匹配^[95]。对于这两种单模态方法，在实验中先求解所有图之间

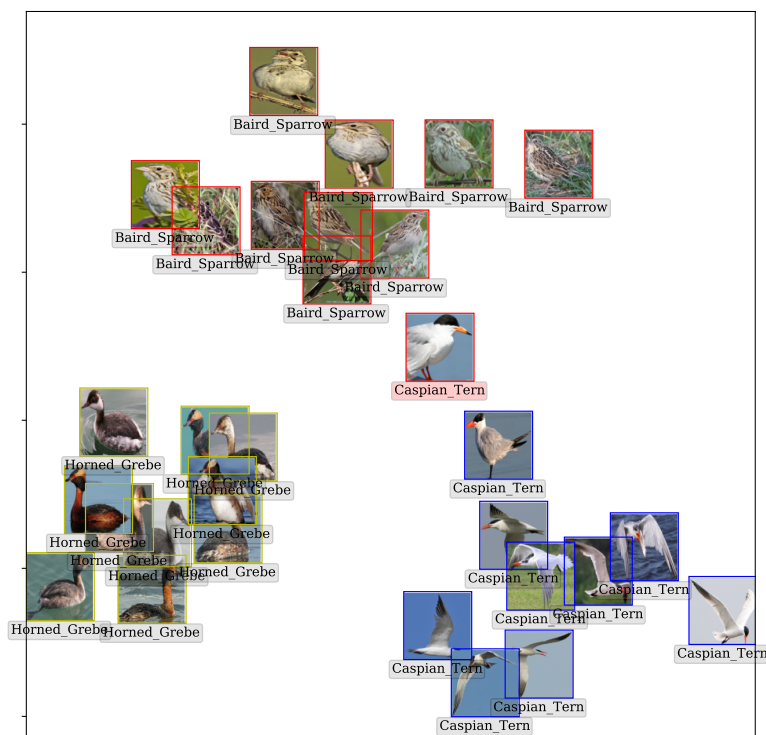


图 4-15 CUB2011 数据集上 GANN-MGM³ 聚类结果的可视化。

Figure 4-15 Visualization of clustering by GANN-MGM³ on CUB2011.

的匹配，再进行谱聚类。表4-6总结了实验结果，其中 CP、RI、CA 是聚类指标，MA 是类内匹配的精度。实验测试了较小规模（24 个图）和较大规模（130 个图）的 MGM³ 问题，GANN-MGM³ 在较小规模问题上表现出色，并且可扩展性较好。对于 MGM³ 任务的无监督学习，聚类精度可能比匹配精度更重要。非学习的 GA-MGM³ 聚类精度较高，但匹配精度不及其他方法。但是凭借 GA-MGM³ 的监督，匹配和聚类性能都得到了改善。此外，GANN-MGM³ 在较小规模问题上相对较慢，但在较大规模问题上比 DPMC^[206] 更快，这可能是因为在小规模问题上的开销在小规模问题上显得更大。此外，实验还发现，无监督学习提升了渐近指派算法的收敛速度。

CUB2011 的实验结果。对于 MGM³ 任务，实验在 10 只角鸬鹚 (Horned Grebe)、10 只贝尔德麻雀 (Baird Sparrow) 和 10 只里海燕鸥 (Caspian Tern) 上进行了 50 次测试。与 Willow ObjectClass 相比，CUB2011 上的 MGM³ 问题更具挑战性，因为所有图像都是鸟类下的不同种属，更难区分。几种 MGM³ 算法在训练集、测试集上的聚类指标以及匹配准确率 (MP)、召回率 (MR) 和 F1 分数 (MF) 如图 4-14(a) 所示。与非学习版本相比，无监督学习有助于提高匹配和聚类的精度。图 4-15 展示了 GANN-MGM³ 在 CUB2011 数据集上的可视化聚类结果，其中图间的距离通过多维标度分析 (multi-dimensional scaling) 降维到了二维平面上并进行了可视化。图像下方的标签代表真实的类别，图像外边框的颜色 (红/蓝/黄) 代表了预测的模式。在这个例子中，仅有一张“里海燕鸥”被错误分类，并在图中

用红色背景标出。如图4-15所示，多数角鸬鹚在游泳、多数贝尔德麻雀在站立、多数里海燕鸥在飞行。在图间距离组成的空间中，这三种姿态分别位于三个不同的方向上。可视化结果显示，不同鸟类的不同姿态可能是实验中正确划分不同模态的关键——甚至对于唯一错误分类的角鸬鹚图像，如果以姿态分类，也是正确的。姿态信息主要存在于边，而本节的算法正是着重考虑了 MGM^3 任务中广泛存在点 + 边的信息。

Pascal VOC 关键点的实验结果。作者考虑了三种模态混合的 MGM^3 问题：从数据集中随机选取 5 张 bicycle、5 张 bottle 和 5 张 tvmonitor 图片。图 4-14(b) 汇报了 200 次随机实验的均值和标准差。实验结果显示，在 *Pascal VOC* 关键点数据集上，无监督学习对聚类指标的提升不如对匹配精度的提升显著，这可能是因为 *Pascal VOC* 中同一类图像在姿态和外观上仍然存在很大差异，这提升了聚类的难度。

4.4 与启发式融合的双层优化通用机器学习框架

启发式求解器 (heuristic solvers) 是在实践中求解排列型组合优化等问题时常用的近似算法。这类算法通常速度快，但可能会牺牲解的质量。目前，大多数现有的机器学习求解组合优化方法将 CO 视为单层优化，直接让神经网络端到端地学习一个解。这种框架可能无法直接扩展到更高复杂性的 CO 问题中，且求解性能受限于机器学习有限的模型容量。本节提出了一种融合算法，以期结合启发式求解和机器学习求解的优点。基于双层优化的形式，该框架将传统求解器与神经网络进行对等的融合，其中上层机器学习方法的目标是优化图结构（例如，在图中添加、删除或修改边），下层的启发式算法求解的是优化图结构后的新问题。对机器学习而言，这种双层优化框架简化了学习任务，模型无需直接学习求解过程，能够有效地缓解对神经网络容量的需求。在三个主流排列型组合优化问题上的实验结果显示，与手工设计的启发式算法和单层的机器学习算法相比，本方法的求解精度更高。

4.4.1 研究动机

目前利用机器学习求解 CO 问题的主流工作多是单层的^[25,54,62,64,144,221-223] (包括第三章)，其中神经网络的输出需要落在可行域中，且这些工作假定了有足够的模型容量来学习 CO 问题的输入-输出映射。然而，在实践中满足该假设将会面临以下两个方面的挑战：一方面，在有限的计算资源下设计并训练一个具有足够容量的神经网络并不容易。现有的模型结构通常为特定问题量身定制，并在设计阶段需要大量的试错^[25,221]；另一方面，训练一个容量足够的大模型要么需要高质量的标签作为监督^[25,67]，但由于（排列）组合优化问题的 NP 难特性，这对于大规模问题来说是不可行的；要么需要采用强化学习 (Reinforce Learning,

RL)^[54,62,84,222], 但由于大规模问题上动作空间过大、奖励稀疏 (sparse reward) 等问题, 这可能会导致训练的不稳定^[224]。

为了利用传统优化算法的稳定性、可靠性, 现有工作^[67,69,225-228] (包括4.2节) 已经证明了将机器学习与传统算法进行融合在求解纯优化问题时的有效性。然而, 设计一种通用的融合算法仍然很难, 因为模型设计通常需要特定的领域知识^[225]。同时, 如何获取高质量的监督标签也是一个挑战, 现有的监督学习方法只能针对特定问题^[67,69]; 不然就需要使用强化学习^[225,227], 其中强化学习的技术难点依然无法回避。

面向以图为输入的 (排列型) 组合优化问题, 本节期望提出一种通用融合方法。具体而言, 本节的研究动机如下:

1) 结合传统启发式和深度学习算法的优势。 作者将原始的 (单层) 组合优化问题重构为一个双层优化问题, 上层优化中的强化学习智能体会自适应地修改图结构, 下层优化采用现有的启发式算法来求解修改后的组合优化任务。该方法不需要监督信息。

2) 降低深度学习模型的复杂性、提升训练稳定性。 通过将下层求解步骤交给传统的启发式算法, 深度学习的一部分模型容量得到解放。此外, 双层优化框架下智能体的每次决策都可以得到奖励函数, 缓解了现有强化学习求解器训练阶段奖励稀疏的问题。

3) 提升启发式算法的灵活性和求解精度。 启发式算法通常由人工设计, 其求解性能与算法设计水平、领域知识背景强关联。本节希望通过机器学习方法, 在灵活的应用场景中通用地提升启发式算法的求解精度。

4.4.2 组合优化问题的单/双层形式

本节主要考虑以图 \mathcal{G} 为参数的 (排列型) 组合优化问题的学习与求解。不失一般性, 考虑参数是单个图的单层优化形式:

$$\min_{\mathbf{x}} f(\mathbf{x}|\mathcal{G}) \quad s.t. \quad h_i(\mathbf{x}, \mathcal{G}) \leq 0, \text{ for } i = 1 \dots I \quad (4.23)$$

其中 \mathbf{x} 代表决策变量 (即问题的解), $f(\mathbf{x}|\mathcal{G})$ 是给定参数 \mathcal{G} 的目标函数, $h_i(\mathbf{x}, \mathcal{G}) \leq 0$ 代表所有约束的集合。以公式 (1.3) 举例说明, 对全排列的组合优化问题, 约束代表了将 \mathbf{x} 变形为矩阵后, 它的每行、每列的求和都必须等于 1。现有的单层优化方法将公式 (4.23) 看作一个端到端的学习任务, 辅以不同的学习方式, 包括: **1) 监督学习**^[25,144], 其中需要利用求解器在小规模问题上得到标签, 但是, 精确求解大规模问题几乎不可能; **2) 无监督学习**^[64] (3.3节、3.4节) 将公式 (4.23) 在连续域的松弛版本作为目标函数, 但现有方法无法通用地处理复杂约束; **3) 强化学习**^[54,62] 通过序列决策预测 \mathbf{x} , 但是直到获得一个完整的解 \mathbf{x} 之前, 都无法获知奖励函数, 这带来了强化学习中的稀疏奖励问题^[224]。

为了改进单层优化形式，作者在经典优化中找到了灵感：可以通过修改问题结构辅助问题求解。例如，在线性整数规划中，添加割平面是一种常用的求解方法^[229-230]。对于图上的（排列）组合优化问题，如果所有的约束都可以在图结构中表达，作者提出了如下假说：

假说 4.3. 通过修改 \mathcal{G} ，即可找到问题参数 \mathcal{G} 对应的最优解 \mathbf{x}^* 。

直接证明这一假说的难度可能太高，但存在一种使上述假说成立的充分条件：

命题 4.4. 定义 \mathbb{G} 是所有从 \mathcal{G} 修改得到的图的集合， \mathbb{X} 是 \mathcal{G} 对应的所有可行解的集合。如果算法 A 是从 \mathbb{G} 到 \mathbb{X} 的满射 (*surjection*)，对于 \mathcal{G} 和它的最优解 \mathbf{x}^* ，一定存在 $\mathcal{G}^* \in \mathbb{G}$ ，其中 \mathbf{x}^* 是算法 A 在 \mathcal{G}^* 上运行得到的结果。

证明. 根据满射的定义，由于 $\mathbf{x}^* \in \mathbb{X}$ ，一定存在一个图 $\mathcal{G}^* \in \mathbb{G}$ ，使得 \mathbf{x}^* 是利用算法 A 求解 \mathcal{G}^* 的结果。□

以有向无环图下的计算调度问题为例，作者进一步说明上述命题的含义。不失一般性，假定依次处理节点 1 到节点 n 是最优解 \mathbf{x}^* ，可以按如下方式修改图结构：对于所有 $i \in \{1, \dots, n-1\}$ ，如果 i 到 $(i+1)$ 的边不存在，则添加这条边。随后，依次处理节点 1 到 n 成了唯一的可行解，而它也会是任意一个算法的输出结果。上述构造方法对 \mathbb{X} 中的所有可行解均适用。

注记 4.5. 本节讨论的假说和命题为后续基于图修改的求解框架做了铺垫。需要注意的是， \mathcal{G}^* 的存在仅仅表明，通过修改图辅助求解是一个有潜力的研究方向。在实际中，给定 \mathcal{G} 、寻找 \mathcal{G}^* 依然具有 NP 难的复杂性。本节的融合求解方法的目的是通过双层优化形式，寻找更好的图（并不一定是最好的图）。考虑实际操作的可行性，作者还限制了修改图的次数。

为了将单层的原问题改写为双层优化，作者引入了“优化后的图” \mathcal{G}' ：

$$\begin{aligned} & \min_{\mathbf{x}', \mathcal{G}'} f(\mathbf{x}' | \mathcal{G}) \\ \text{s.t. } & H_j(\mathcal{G}', \mathcal{G}) \leq 0, \text{ for } j = 1 \dots J, \\ & \mathbf{x}' \in \arg \min_{\mathbf{x}'} \{f(\mathbf{x}' | \mathcal{G}') : h_i(\mathbf{x}', \mathcal{G}') \leq 0, \text{ for } i = 1 \dots I\}. \end{aligned} \quad (4.24)$$

其中 $f(\mathbf{x}' | \mathcal{G}), f(\mathbf{x}' | \mathcal{G}')$ 分别是上层、下层优化的目标函数。下层优化代表了在优化后的 \mathcal{G}' 上的 CO 问题，采用启发式算法进行求解。下层的解 \mathbf{x}' 随后被上层接收，上层优化的目标函数 $f(\mathbf{x}' | \mathcal{G})$ 代表了 \mathbf{x}' 在原问题上计算得到的目标函数。上层优化的约束 $H_j(\mathcal{G}', \mathcal{G}) \leq 0$ 保证了 \mathcal{G}' 的可行域必须是 \mathcal{G} 的子集，且 \mathcal{G}' 至多从 \mathcal{G} 出发后修改了 K 步。上层优化的输出是 \mathcal{G}' ，采用一个强化学习智能体进行求解，其中公式 (4.24) 被视为强化学习的环境。

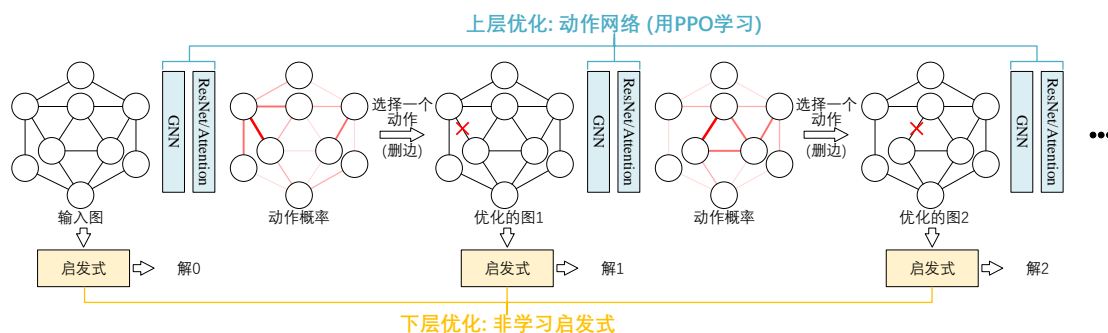


图 4-16 双层融合的机器学习 CO 求解器示意图。
Figure 4-16 An overview of bi-level hybrid MLCO solver.

4.4.3 算法描述

本节详细介绍双层融合算法 (**Bi-level Hybrid**, 下称 BiHyb), 算法框架如图 4-16 所示。该算法采用强化学习预测 \mathcal{G}' , 进而求解公式 (4.24)。这种在机器学习范式下交替求解的方式可以看作传统双层优化求解方法^[231]的扩展。

马尔可夫决策过程 (Markov Decision Process, MDP) 的定义。公式 (4.24) 是强化学习算法的优化目标。算法 4-6 总结了 MDP 的策略推演步骤, $\mathcal{G}^0 = \mathcal{G}$ 代表原图, $\mathcal{G}^k (k \neq 0)$ 代表 \mathcal{G}' , 即是经过第 k 次修改后的图。

- 状态: 当前的图 \mathcal{G}^k 被视为 MDP 的状态, \mathcal{G}^k 的节点、边包含了完整的问题参数和当前约束。初始状态 \mathcal{G}^0 代表了初始的组合优化问题。
- 动作: 动作空间的范围可包括增加、删除或修改 \mathcal{G}^k 中的一条边。由于具有 m 个节点的图 \mathcal{G}^k 至多有 m^2 条边, 通过将边的选择分解成两次节点选择, 可以将每次的动作空间减小到 $\mathcal{O}(m)$ 。特别说明, 在动作设计时, 需要保证修改后的图对应的可行域是原问题可行域的子集, 即可行域不扩大。
- 状态转移: 在完成一次动作后, \mathcal{G}^k 变为 \mathcal{G}^{k+1} , 且它们仅有一条边不同。新的图 \mathcal{G}^{k+1} 被视作新的状态, 且同时被用于计算奖励。当触及动作的上限次数 K 时, MDP 即结束。在实验中, 对于上千节点的图, 只需设定 $K \leq 20$ 即可得到不错的效果, 因此 RL 训练中稀疏奖励的问题得到了极大改善 (20 次动作 vs 1000+ 次动作)。
- 奖励: 新的图 \mathcal{G}^{k+1} 代表了一个修改后的优化问题, 它的目标函数变为 $f(\mathbf{x}^{k+1} | \mathcal{G}^{k+1})$, \mathbf{x}^{k+1} 是利用现有的启发式求解器得到的一个解。奖励函数 $= f(\mathbf{x}^k | \mathcal{G}) - f(\mathbf{x}^{k+1} | \mathcal{G})$, 代表了得到 \mathbf{x}^{k+1} 后, 上层目标函数的提升程度 (假设原问题是最小化问题)。

近端策略优化 (Proximal Policy Optimization, PPO)。本节采用了流行的 PPO 算法^[60]作为强化学习框架。PPO 是信任区域策略优化^[232] (Trust Region Policy Optimization, TRPO) 的简化版本, 其中模型更新被限制在了一个“信任区域”

算法 4-6 双层融合组合优化机器学习算法的策略推演 (policy roll-out)

输入: 初始的图 \mathcal{G} ; 最大修改次数 K 。

- 1: $\mathcal{G}^0 \leftarrow \mathcal{G}$;
- 2: **for** $k \leftarrow 0 \dots (K-1)$ **do**
- 3: 基于 \mathcal{G}^k 预测 $P(\mathbf{a}_1), P(\mathbf{a}_2|\mathbf{a}_1)$ 并采样 $\mathbf{a}_1, \mathbf{a}_2$; # 上层优化
- 4: $\mathcal{G}^{k+1} \leftarrow$ 增/删/改图 \mathcal{G}^k 的边 $(\mathbf{a}_1, \mathbf{a}_2)$; # 状态转移
- 5: $\mathbf{x}^{k+1} \leftarrow$ 利用启发式算法求解 $\arg \min_{\mathbf{x}^{k+1}} f(\mathbf{x}^{k+1}|\mathcal{G}^{k+1})$; # 下层优化
- 6: $r_k \leftarrow f(\mathbf{x}^k|\mathcal{G}) - f(\mathbf{x}^{k+1}|\mathcal{G})$; # 奖励函数
- 7: **end for**

输出: 一系列奖励函数 $\{r_0 \dots r_{K-1}\}$ 。

中, 以避免模型坍塌。PPO 比 TRPO 更容易实现、且性能相当^[60], 因此在实际中更加常用。PPO 最大化的目标函数是

$$J(\theta) = \min(r_\theta \cdot A, \text{clip}(r_\theta, 1 - \epsilon, 1 + \epsilon) \cdot A), \quad (4.25)$$

其中 r_θ 是重要性采样的比率 (importance sampling ratio), 它与模型参数 θ 有关。A 是当前策略的优势值 (advantage), 它等于累积的奖励函数减去评价网络 (critic network) 的预测值, ϵ 是控制信任区域大小的超参数。此外, 强化学习中几个常用的技巧也被采用: 在模型更新时, 累积的奖励函数被归一化; 作者还添加了一个额外的熵正则项, 以鼓励智能体跳出局部最优解, 探索更多的解空间。

4.4.4 具体实现和实验分析

接下来展示双层融合框架在三个具有挑战性的 CO 问题的具体实现和实验结果: 基于有向无环图的计算任务调度 (Directed Acyclic Graph scheduling, DAG 调度)、图编辑距离 (Graph Edit Distance, GED) 和哈密尔顿回路问题 (Hamiltonian Cycle Problem, HCP)。实验比较了双层强化学习方法 PPO-BiHyb、无需学习的启发式方法和单层的强化学习方法 PPO-Single。PPO-Single 的设计遵循 Khalil 等^[54]提出的通用框架, 该框架涵盖了大部分现有的强化学习方法^[56,80,221-223,233]。实验还考虑了在双层优化框架下进行随机图修改的 Random-BiHyb。为了公平比较, PPO-Single 的模型容量和训练/推理时间与 PPO-BiHyb 保持一致。

DAG 调度问题。有向无环图 (DAG) 能够刻画具有数据依赖关系的计算任务, DAG 调度问题是计算机集群中并行作业调度的抽象: 每个节点表示一个计算任务, 每个任务对应一个运行时间和资源需求。每个节点可能有多个父节点和子节点, 表示数据依赖关系。集群的总资源有限, 如果空闲资源足够且并发任务不存在数据依赖关系, 则这些任务可以并行地执行。DAG 调度是一个 NP 难的优化问题^[234], 目标是最小化完成所有任务的总时间。采用 PPO-BiHyb 求解 DAG 调度问题的具体实现如下:

- *MDP* 定义。如图 4-17 所示, 状态定义为当前的 DAG \mathcal{G}^k , 动作定义为向 \mathcal{G}^k

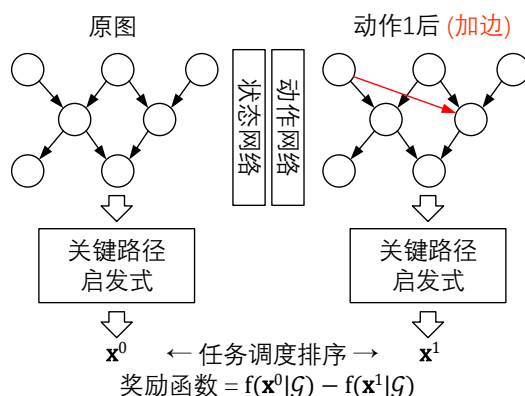


图 4-17 DAG 调度问题的状态、动作、奖励定义。

Figure 4-17 Definition of state, action and reward for DAG scheduling.

添加一条边，从而得到一个新的 DAG \mathcal{G}^{k+1} 。添加的边增加了额外的约束条件，保证了修改后的问题可行域是原问题的子集。这里 \mathbf{x} 表示任务的执行顺序，在 \mathcal{G}^{k+1} 上采用关键路径启发式来计算 \mathbf{x}^{k+1} 。奖励函数的计算方式为旧的任务耗时减去新的任务耗时： $f(\mathbf{x}^k|\mathcal{G}) - f(\mathbf{x}^{k+1}|\mathcal{G})$ 。

- 状态编码网络。采用 GCN^[133] 对 DAG 的状态进行编码。考虑到 DAG 的结构，本节设计了两个 GCN：第一个 GCN 处理原始 DAG，第二个 GCN 处理所有边被翻转的 DAG。来自两个 GCN 模块的节点嵌入被拼接在一起，并采用注意力池化（attention pooling）层^[162]得到图级别的嵌入向量：

$$\mathbf{n} = \left[\text{GCN}_1(\mathcal{G}^k) \quad \text{GCN}_2(\text{reverse}(\mathcal{G}^k)) \right], \mathbf{g} = \text{Att}(\mathbf{n}), \quad (4.26)$$

其中 \mathbf{n} （节点数目 \times 特征维度）代表节点嵌入。

- 动作网络。为了减小动作空间，边的选择被分解为两步：首先选择起始节点，然后选择结束节点。起始节点和结束节点的选择概率分别由独立的 3 层残差网络^[220]预测，第二个残差网络还将所选起始节点的特征向量拼接作为输入：

$$P(\mathbf{a}_1) = \text{softmax}(\text{ResNet}_1([\mathbf{n} \quad \mathbf{g}])), \quad (4.27a)$$

$$P(\mathbf{a}_2|\mathbf{a}_1) = \text{softmax}(\text{ResNet}_2([\mathbf{n} \quad \mathbf{n}[\mathbf{a}_1] \quad \mathbf{g}])), \quad (4.27b)$$

其中 $\mathbf{n}[\mathbf{a}_1]$ 代表节点 \mathbf{a}_1 的嵌入向量。训练阶段， $\mathbf{a}_1, \mathbf{a}_2$ 分别根据对应的概率 $P(\mathbf{a}_1), P(\mathbf{a}_2|\mathbf{a}_1)$ 进行采样。在测试阶段进行宽度为 3 的集束搜索（beam search）：搜索将覆盖具有前三高概率的动作，在所有候选动作中，只保留奖励函数最高的 3 个动作进入下一轮搜索。

- 评价网络。将最大池化作用于所有节点嵌入，池化后的特征与状态编码网

表 4-7 TPC-H 数据集上 DAG 调度问题的实验结果。

Table 4-7 Experimental results on DAG scheduling problems from TPC-H dataset.

方法名称	数据集	TPC-H-50 (节点数=467.2)		TPC-H-100 (节点数=929.8)		TPC-H-150 (节点数=1384.5)	
		目标函数↓	相对值↓	目标函数↓	相对值↓	目标函数↓	相对值↓
Shortest Job First		12818±2214	30.5%	19503±3260	15.3%	27409±2748	12.2%
Tetris ^[40]		12113±1398	23.3%	18291±2223	8.1%	25325±2842	3.7%
Critical Path		9821±1176	0.0%	16914±2499	0.0%	24429±2484	0.0%
PPO-Single ^[54,222]		10578±2092	7.7%	17282±3821	2.2%	24822±2707	1.6%
Random-BiHyb		9270±1143	-5.6%	15580±2409	-7.9%	22930±2408	-6.1%
PPO-BiHyb		8906±922	-9.3%	15193±2275	-10.2%	22371±2538	-8.4%

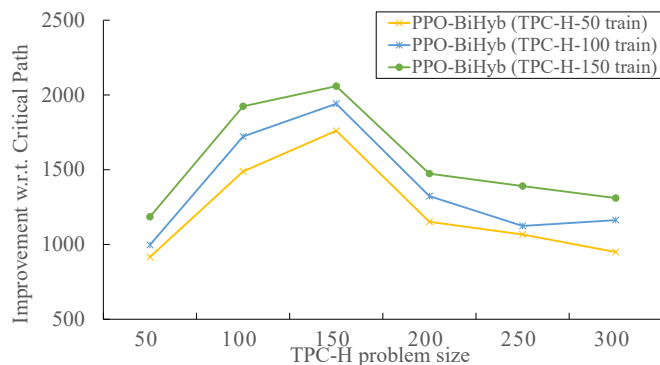


图 4-18 TPC-H 数据集上的泛化性研究

Figure 4-18 Generalization result on TPC-H dataset.

络得到的全图特征拼接，最后采用另一个残差网络预测当前状态的价值 (value)：

$$\tilde{V}(\mathcal{G}^k) = \text{ResNet}_3([\text{maxpool}(\mathbf{n}) \quad \mathbf{g}]). \quad (4.28)$$

- 启发式方法。对于数百节点的 DAG 调度问题，使用现有的商用求解器直接求解已几乎不可行。实际的调度问题通常通过快速的启发式算法来求解，例如最短任务优先 (*Shortest Job First*) 算法，该算法贪心地选择耗时最短的任务，以及关键路径 (*Critical Path*) 算法，该算法优先考虑关键路径上的任务。此外，作者还考虑了新颖的 *Tetris* 调度算法^[40]，该方法将任务调度看作资源、耗时二维空间上的俄罗斯方块游戏。由于在实验中关键路径算法性能最好，它被选作 PPO-BiHyb 中的下层优化算法。
- 机器学习方法。目前已有一些相关工作，使用强化学习在数据中心进行任务调度^[222]以及使用强化学习调度车间任务^[223]。它们可以被视为单层优化模型^[54]在特定问题上的具体实现。由于现有工作^[222-223]的问题设定与本节不同，作者重新实现了单层优化 RL 模型 PPO-Single，它的神经网络配置与 PPO-BiHyb 一致。为了验证 PPO 强化学习的有效性，实验还比较了 Random-BiHyb，它和 PPO-BiHyb 的搜索步骤相等，但在修改图时使用随机搜索而不是 PPO。

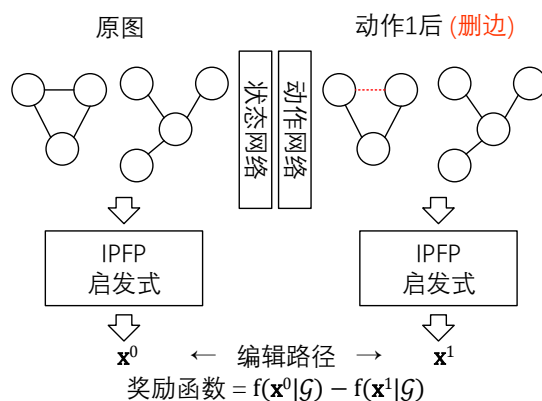


图 4-19 图编辑距离问题的状态、动作、奖励定义。

Figure 4-19 Definition of state, action and reward for Graph Edit Distance.

DAG 调度实验在 TPC-H 数据集¹⁶上进行，该数据集源自真实业务场景中的查询和并发修改，每个 DAG 都代表一个计算任务。DAG 中的每个节点具有两个属性：运行时间和资源需求。TPC-H 数据集中的 DAG 平均具有 9.18 个节点，最小的任务有 2 个节点，最大的任务有 18 个节点。节点的平均资源需求为 125.8，最小值为 1，最大值为 593。节点的平均运行时间为 1127.2 秒，最短时间为 16.3 秒，最长时间为 4964.5 秒。实验同时调度多个 DAG，并假设总共有 6000 单位的资源（即最重负载的节点将占用约 10% 的总资源），以反映真实的业务场景。实验有 50 个训练样本和 10 个测试样本，样本的生成通过从 TPC-H 中随机抽取 X 个 DAG 实现（X=50,100,150），随机种子固定。

表 4-7 总结了 TPC-H 数据集上 DAG 调度任务的结果，每个数据集的平均节点数目在括号中标出。表格的上半部分是非学习的启发式求解器，下半部分是基于 RL 的求解器。“目标函数”代表完成所有任务的耗时，“相对值”代表了当前方法和关键路径方法的相对差距。其中，PPO-BiHyb 胜过了非学习启发式算法和单层强化学习方法。PPO-Single 的搜索宽度被控制在 10，作者允许它的测试耗时比 PPO-BiHyb 长 10 倍，但其性能仍然不及关键路径启发式。随机的 Random-BiHyb 可以改进下层的启发式，但其性能仍然不及 PPO-BiHyb。上述结果证明了双层优化框架和 PPO 训练的有效性。此外，作者还研究了 PPO-BiHyb 方法的泛化能力，因为这对实际应用非常重要。图 4-18 总结了当模型分别在 TPC-H-50/100/150 数据集上训练、在其他大小的数据集上测试的实验结果。可以看出，PPO-BiHyb 可以很好地推广到大小不一的数据，甚至在具有多达 300 个 DAG（约 2800 个节点）的大规模问题上也是如此。根据作者分析，PPO-BiHyb 具有良好泛化性的原因是学习图优化策略比直接学习问题求解更容易，并且 PPO-BiHyb 充分利用了启发式算法的性能优势。

图编辑距离问题。作为一个 NP 难的排列型组合优化问题，图编辑距离(GED)问题定义和应用场景已在 4.2 节中充分讨论。在本节中，采用 PPO-BiHyb 求解图

¹⁶<http://tpc.org/tpch/default5.asp>

表 4-8 AIDS 数据集上图编辑距离 (GED) 问题的实验结果。

Table 4-8 Results on graph edit distance (GED) problems from AIDS dataset

方法名称	数据集	AIDS-20/30 (节点数=22.6)		AIDS-30/50 (节点数=37.9)		AIDS-50+ (节点数=59.6)	
		目标函数↓	相对值↓	目标函数↓	相对值↓	目标函数↓	相对值↓
	Hungarian ^[184]	72.9±19.2	94.9%	153.4±28.0	117.9%	225.6±33.9	121.4%
	RRWM ^[34]	72.1±23.7	92.8%	139.8±31.9	98.6%	214.6±41.3	110.6%
	Hungarian-Beam Search ^[38]	44.6±8.5	19.3%	103.9±22.7	47.6%	143.8±31.5	41.1%
	IPFP ^[39]	37.4±8.5	0.0%	70.4±15.1	0.0%	101.9±13.1	0.0%
	PPO-Single ^[56]	56.5±14.4	51.1%	110.0±19.2	56.3%	183.9±16.9	80.5%
	Random-BiHyb	33.1±9.0	-11.5%	66.0±15.2	-6.3%	82.4±20.3	-19.1%
	PPO-BiHyb	29.1±8.9	-22.2%	61.1±14.2	-13.2%	77.0±19.4	-24.4%

编辑距离问题的具体实现如下：

- 双层优化定义。由于 GED 问题涉及两个图，它的双层优化形式和其他单图问题不同：

$$\begin{aligned}
 & \min_{\mathbf{x}', \mathcal{G}'_1} f(\mathbf{x}' | \mathcal{G}_1, \mathcal{G}_2) \\
 & s.t. \quad H_j(\mathcal{G}'_1, \mathcal{G}_1) \leq 0, \text{ for } j = 1 \dots J \\
 & \quad \mathbf{x}' \in \arg \min_{\mathbf{x}'} \{f(\mathbf{x}' | \mathcal{G}'_1, \mathcal{G}_2) : h_i(\mathbf{x}', \mathcal{G}'_1, \mathcal{G}_2) \leq 0, \text{ for } i = 1 \dots I\}
 \end{aligned} \tag{4.29}$$

其中 $f(\mathbf{x}' | \mathcal{G}_1, \mathcal{G}_2)$ 是上层目标函数：基于 $\mathcal{G}_1, \mathcal{G}_2$ 和 \mathbf{x}' 的图编辑距离。决策变量 \mathbf{x}' 是一个排列阵，代表了从 \mathcal{G}_1 到 \mathcal{G}_2 的节点编辑方案。边的编辑方案可以由节点编辑推导得到。上层约束条件 $H_j(\mathcal{G}'_1, \mathcal{G}_1) \leq 0$ 代表了 \mathcal{G}_1 经过至多 K 次修改后可得到 \mathcal{G}'_1 。GED 的下层优化定义为修改后的 \mathcal{G}'_1 与 \mathcal{G}_2 之间的 GED 问题。

- MDP 定义。如图4-19所示，GED 问题中固定了 \mathcal{G}_2 ，智能体只能修改 \mathcal{G}_1 。状态的定义为当前的 \mathcal{G}_1^k ，动作的定义为添加或删除 \mathcal{G}_1^k 中的一条边。算法设计中，新的图 \mathcal{G}_1^{k+1} 更容易通过下层的 IPFP 启发式^[39]与 \mathcal{G}_2 对齐。
- 状态编码网络。状态编码网络采用 GCN^[133]更新图特征，采用 Sinkhorn 网络^[115]进行跨两个图的传播。与3.2节类似，Sinkhorn 网络接受一个相似度矩阵（由 \mathbf{n}_1 和 \mathbf{n}_2 的内积得到），并输出一个双随机矩阵，该矩阵被用作跨图传播的权重。由于动作仅在第一个图上执行，计算图 1 的节点特征与来自图 2 特征之间的差异作为最终的节点特征。与 DAG 模型类似，通过注意力池化来获得图级嵌入：

$$\mathbf{n}_1 = \text{GCN}(\mathcal{G}_1^k), \mathbf{n}_2 = \text{GCN}(\mathcal{G}_2), \mathbf{n} = \mathbf{n}_1 - \text{Sinkhorn}(\mathbf{n}_1 \mathbf{n}_2^\top) \cdot \mathbf{n}_2; \tag{4.30a}$$

$$\mathbf{g}_1 = \text{Att}(\mathbf{n}_1), \mathbf{g}_2 = \text{Att}(\mathbf{n}_2). \tag{4.30b}$$

training size	20/30	29.1	63.2	78.6
	30/50	33.3	61.1	84.4
	50+	32.0	61.6	77.0
		20/30	30/50	50+
		testing size		

图 4-20 在不同规模的 AIDS 数据集上训练/测试后的泛化结果。

Figure 4-20 Generalization study by varying training and testing sizes on AIDS.

- 动作网络。添加/删除图 1 中的一条边的动作被分解为两步节点选择。起始点的选择采用 3 层残差网络，中止点的选择采用了注意力查询机制。如果这条边已经存在，则将它删除；不然就添加一条边。测试时的集束搜索宽度设为 3。
- 评价网络。评价网络的结构模仿了受到了 4.2 节图相似度学习工作的启发，其中图级特征采用神经张量网络^[201]处理，最后经过两层全连接后，得到一维输出：

$$\tilde{V}(\mathcal{G}_1^k, \mathcal{G}_2) = \text{fc}(\text{NTN}(\mathbf{g}_1, \mathbf{g}_2)). \quad (4.31)$$

- 启发式方法。基于 Blumenthal 等^[235]对 GED 启发式方法的详细评测结果，实验选择了其中 4 种性能最优的启发式算法：匈牙利算法^[184] (Hungarian) 将原问题简化为二分图匹配问题，RRWM^[34]采用松弛后的二次规划求解 GED，匈牙利集束搜索算法^[38] (Hungarian-Beam Search) 采用匈牙利启发式引导搜索，在 4.2 节已有讨论，IPFP^[39]结合了搜索和二次规划。因为在实验中，IPFP 是性能最好的启发式算法，因此它被选为 PPO-BiHyb 的启发式算法。
- 机器学习方法。如 4.2 节讨论，现有的深度学习方法能够以回归的形式预测一个 GED 分数^[161-162]，但忽略了 GED 问题的组合优化本质。4.2 节的神经网络融合 A* 搜索方法与本节的方法功能相似，但它需要监督学习，无法直接比较。实验基于 PPO-Single，主要比较了 Liu 等的强化学习框架^[56]。PPO-Single 和 PPO-BiHyb 的网络设计细节保持一致，且 PPO-Single 的集束搜索宽度被设为 20，从而保持和 PPO-BiHyb 接近的预测效率。Random-BiHyb 与 PPO-BiHyb 的搜索步数相同，但是搜索方向是随机的。

GED 实验在 AIDS 数据集上进行，该数据集包含研究中的抗 HIV 化合物分子^[179]，已在 4.2 节的实验分析中介绍。遵循 4.2 节的实验设定，作者将原子视为节

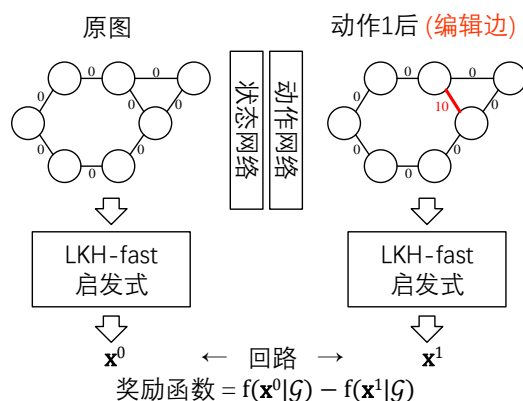


图 4-21 汉密尔顿回路问题的状态、动作、奖励定义。

Figure 4-21 Definition of state, action and reward for Hamiltonian Cycle Problem.

点，使用 one-hot 特征对编码原子种类，并定义不同类节点的编辑代价为 1。此外，对点、边的增、删操作代价均为 1。根据图的大小，将 AIDS 数据集三个子集，即 AIDS-20/30、AIDS-30/50 和 AIDS-50+，其中 AIDS-X/Y 代表数据集中的节点数目在 X、Y 之间。因为小于 20 个节点的图编辑距离问题相对简单，可以在几小时内精确求解，实验考虑了更大规模的图。对每个数据集，使用固定的随机种子选取 50 对训练样本和 10 对测试样本。

表4-8总结了 AIDS 数据集上 GED 问题的实验结果。“目标函数”代表求解得到的编辑代价，“相对值”代表了当前方法和 IPFP 方法的相对差距。PPO-BiHyb 超过了所有无需学习的启发式方法，并且也比单层的 RL 方法 PPO-Single 表现更好。此外，作者还研究了针对不同问题规模的泛化性，图4-20显示了在不同大小的 GED 问题上进行训练/测试的结果。颜色深浅代表相对 IPFP 的改进百分比，颜色越深改进越多。PPO-BiHyb 模型可以泛化到训练时未见过的问题规模，但泛化性能通常不如在相同规模下训练得到的模型。

汉密尔顿回路问题。作为图论中一个经典的排列型组合优化问题，汉密尔顿回路问题 (HCP) 的研究起源于由欧拉提出的戈尼斯堡 (Königsberg) 七桥问题^[236]。汉密尔顿回路的定义是一条途径所有点的闭合回路。给定一个图，HCP 需要回答：这个图中是否存在汉密尔顿回路？这是一个 NP 完全的决策问题^[237]。本节将 HCP 转化为全连接图，用更通用的旅行商问题 (TSP) 算法来求解：在对应的 TSP 问题中，对于 HCP 中存在的边，其长度定义为 0；不存在的边，其长度定义为 1。因此，如果 TSP 求解器找到了一个长度为 0 的回路，它就是一个汉密尔顿回路。值得注意的是，多数研究机器学习 TSP 的工作^[25,80,233,238]都没有考虑 HCP，而是更关注二维坐标下欧氏距离 TSP 这一特殊情况。采用 PPO-BiHyb 求解 HCP 的具体实现如下：

- **MDP 定义。**如上文所述，为了充分利用现有的 TSP 启发式算法，HCP 问题被转化成了 TSP 问题。如图4-21所示，求解 HCP 问题时，状态定义为当前图 \mathcal{G}^k ，动作定义为增加 \mathcal{G}^{k+1} 中一条边的权重。通过增加边权，可以柔性

地避免启发式算法经过这条边。LKH 启发式^[37]被用于下层优化,进而计算新的回路 \mathbf{x}^{k+1} 。奖励函数定义为当前回路相比前一回路的长度缩短量,即 $f(\mathbf{x}^k|\mathcal{G}) - f(\mathbf{x}^{k+1}|\mathcal{G})$ 。

- 状态编码网络。实验采用 GCN^[133]编码 HCP 的图,GCN 的节点嵌入随后被注意力池化机制处理,以获得全图的嵌入向量。

$$\mathbf{n} = \text{GCN}(\mathcal{G}^k), \mathbf{g} = \text{Att}(\mathbf{n}) \quad (4.32)$$

- 动作网络。在使用启发式算法求解回路后,PPO-BiHyb 模型将选择一条边,将其长度增加 10 (实验证明,本方法对该数值不敏感),将这一程度上避免后续步骤中选择这条边。选择起始点的第一个动作由 3 层残差网络^[220]预测,第二个动作是选择结束节点,该节点需要与起始节点在当前回路上相邻,通过注意力查询得到。

$$P(\mathbf{a}_1) = \text{softmax}(\text{ResNet}_1([\mathbf{n} \ \mathbf{g}])), \quad (4.33a)$$

$$P(\mathbf{a}_2|\mathbf{a}_1) = \text{softmax}\left(\mathbf{n} \cdot \tanh(\text{Linear}(\mathbf{n}[\mathbf{a}_1]))^\top\right). \quad (4.33b)$$

训练中, $\mathbf{a}_1, \mathbf{a}_2$ 根据 $P(\mathbf{a}_1), P(\mathbf{a}_2|\mathbf{a}_1)$ 采样得到。测试阶段,作者采用了宽度 12 的集束搜索。

- 评价网络。对所有节点特征采用最大池化操作,池化后的特征与状态编码网络的全图特征拼接,最后输入三层残差网络预测当前状态的价值:

$$\tilde{V}(\mathcal{G}^k) = \text{ResNet}_2([\text{maxpool}(\mathbf{n}) \ \mathbf{g}]). \quad (4.34)$$

- 启发式方法。启发式方法的性能随着问题规模、问题性质变化而变化。实验考虑了三种启发式算法进行比较:最近邻算法^[239] (Nearest Neighbour) 每一步都会选择距离最近的点,最远插入算法^[36] (Farthest Insertion) 会插入不在当前路径中、且离路径最远的点,LKH3 启发式^[37]由作者 Lin、Kernighan 命名,基于 k -最优算法流程,LKH3 是一个强大的启发式求解器,可以解得许多问题的最优解。通过调整 LKH3 的随机尝试次数,能够权衡耗时与精度(默认 5 次随机尝试)。实验中,LKH3 的默认版本被命名为 *LKH3-fast*,这也是 PPO-BiHyb 模型的下层优化算法。实验还考虑了另一种更耗时的版本,即 100 次随机尝试的 *LKH3-accu*。
- 机器学习方法。目前多数现有的 TSP 神经网络求解器只能处理二维平面上欧式距离的 TSP^[25,80,233,238],这和本节的 HCP 使用条件不一致。因此,作

表 4-9 FHCP 数据集上哈密顿回路问题 (HCP) 的实验结果。

Table 4-9 Results on Hamiltonian cycle problem (HCP) from FHCP dataset.

方法名称	数据集	FHCP-500/600 (节点数 = 535.1)	
		TSP 目标函数 ↓	发现哈密顿回路 ↑
Nearest Neighbor ^[239]		79.6±13.4	0%
Farthest Insertion ^[36]		133.0±31.7	0%
LKH3-fast ^[37]		13.8±25.2	0%
LKH3-accu ^[37]		6.3±13.0	20%
PPO-Single ^[54]		9.5±45.6	0%
Random-BiHyb		10.0±21.9	0%
PPO-BiHyb		6.7±14.0	25%

者基于目前最一般的框架^[54]重新实现了 PPO-Single，其中神经网络的设计细节与 PPO-BiHyb 保持一致。随机的 Random-BiHyb 方法也被一起比较。

HCP 实验在包含了 1001 个难解 HCP 实例的 FHCP 数据集^[240]上进行。已知该数据集的所有实例都存在哈密顿回路，但对于标准的 HCP/TSP 启发式算法而言，找到它们并不容易。实验结果总结在表 4-9 中，数据集名称 FHCP-X/Y 代表了节点数目在 X 到 Y 之间，“TSP 目标函数”基于 HCP 转化得到的 TSP 问题计算，训练集由 FHCP 数据集中节点数目在 250 到 500 之间的图组成。PPO-BiHyb 的性能与最强的启发式算法 *LKH3-accu* 相当，且在发现哈密顿回路的能力上甚至可以超越 *LKH3-accu*，而发现更多的哈密顿回路正是 HCP 的最终目标。PPO-Single 的集束搜索宽度设置为 12，它的推理时间比 PPO-BiHyb 长 10 倍，但其性能仍然不如双层的 PPO-BiHyb。此外，基于随机搜索的双层优化方法 Random-BiHyb 也可以提升下层算法 *LKH3-fast* 的性能。

4.5 本章小结

传统算法与机器学习方法是求解排列型组合优化问题时两种重要的技术路径。融合算法的核心优势是在求解纯优化问题时，性能通常比单独的神经网络或传统求解器更高。本节的初步研究证明，将机器学习与传统算法融合，能够同时利用机器学习算法的高效率以及传统算法高精度、强理论保证等优势。特别地，为了与第三章的纯机器学习方法区分，作者在图 4-22 中总结了与传统算法融合的机器学习方法的三大技术特点：更精细、更可靠、更易学。其中，更精细体现在算法设计时需要充分理解传统算法和机器学习的内部原理和具体优势、劣势。由于不同具体问题、不同传统算法的组合非常多，较难产生通用的融合求解范式，而是需要针对具体场景精细化设计。例如，4.2 节专为 A* 算法和图编辑距离设计，4.3 节专为无监督图匹配任务设计，4.4 节需要了解每个问题的最优启发式方法。更可靠体现在传统算法能够提供性能兜底，规避神经网络可能的泛化性差、

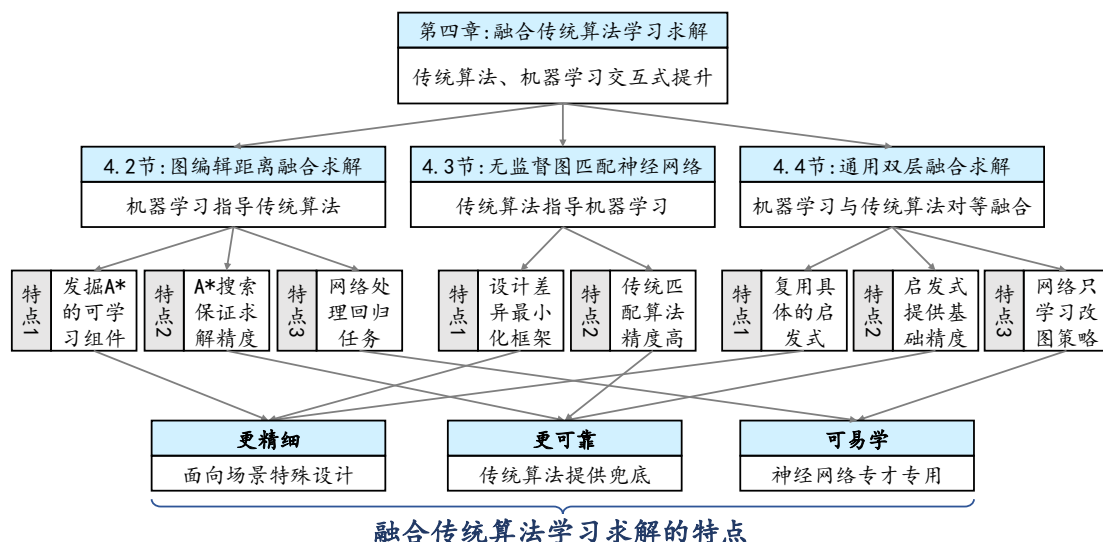


图 4-22 第四章融合传统算法的学习求解排列型组合优化研究总结。

Figure 4-22 Summary of research on fused traditional solvers and machine learning for permutation-based combinatorial optimization.

网络坍塌等问题，对实际应用更加友好。例如，4.2节和4.3节的传统算法能够提供比神经网络更精确的结果（虽然耗时更长），4.4节的框架保证了融合算法的性能不会劣于启发式本身。更易学体现在神经网络在融合算法中只需完成某个具体任务，不需要学习问题参数到解的映射。例如，4.2节中神经网络只需回归相似度分数，4.4节的神经网络只需学习改图策略。

在技术层面，本章总结了三种融合范式，分别是4.2节的神经网络指导传统算法、4.3节的传统算法指导神经网络、4.4节的神经网络于传统算法对等融合。面向具体问题，本章还有如下技术贡献：4.2节以图编辑距离问题中传统的A*算法为基础，通过在经典的树搜索步骤中引入神经网络预测模块，显著提升了传统算法的效率。4.3节以图匹配问题中最新的深度学习求解方法为基础，融合了多种传统图匹配求解算法，通过差异最小化框架进行无监督学习，在二图匹配、多图匹配、混合模态多图匹配等实际任务中实现了无监督预训练。4.4节将机器学习和传统算法分别看成双层优化的上、下层，设计了一种强化学习框架求解通用的（排列型）组合优化问题。在三个经典问题上的实验结果表明，相比于单独的机器学习模型和单独的传统启发式算法，双层融合算法的性能更强。当然，本章的研究只是对三种融合范式（传统算法指导、机器学习指导、传统算法和机器学习对等）的初探，面向丰富的具体问题，作者相信融合框架还有广阔的发展空间和应用前景。

第五章 排列型组合优化问题的学习求解开源

5.1 本章引言

排列型组合优化问题具有广泛的应用场景和丰富的理论内涵，在技术层面涵盖了传统的优化算法到最新的机器学习、深度学习方法。排列型组合优化方法论的丰富性一方面带来了深刻的研究价值，但另一方面，过于复杂、专业的技术细节实际上限制了该领域的发展和應用。为了一定程度上解决排列型组合优化实践中的挑战，本文作者提供了若干排列型组合优化的算法实现，统一了编程接口，编写了详细文档，开源了面向排列型组合优化（特别是图匹配）设计的开源算法框架。其中，面向下游应用场景的图匹配通用算法库（pygmtools）面向广大科研、工程人员设计，无需了解过多的算法细节即可调用成熟的传统/机器学习求解器。面向科研的深度图匹配系统（ThinkMatch）专为本方向的科研人员设计，系统支持完善的训练、测试功能，能够快速验证算法、调整超参数等。面向国内自主研发的深度学习框架“计图”的图匹配求解复用了计图高效率的框架，同时也为计图带来了图匹配建模、求解等功能。一方面，这些开源工作是作者到目前为止研究工作的总结和凝练；另一方面，作者希望这些开源工作可以进一步壮大开源社区，促进国内外相关领域的深入研究。

5.2 pygmtools：面向下游应用的图匹配通用算法库

鉴于图的强大表达能力以及匹配任务的广泛需求，图匹配问题的重要性已被广泛认可^[98-99]，图匹配技术具有广泛的应用场景^[17-18,24,212,241-242]。图匹配的技术优势来自两方面：一方面，图匹配明确的数学形式（公式(3.1)）显式地建模了点和边的特征，而多数匹配方法（例如点匹配方法^[10,243]）只考虑了节点特征。图匹配方法的理论上限显然更高。另一方面，最新的深度图匹配方法^[16,77,244]提供了可微的图匹配求解器，更容易集成到端到端的深度学习框架中。本节介绍了面向多种深度学习和科学计算框架的图匹配通用算法库 pygmtools¹⁷，它的核心优势在于易用性和通用性。

5.2.1 开发动机与竞品比较

随着图匹配受到的关注日益增加，作者希望建立一个易于使用的图匹配工具，为广大科研人员、开发人员提供最常用的图匹配求解算法。在框架设计时，选择 Python 作为编程语言，一方面可适配日益发展的深度学习技术（Python 已经是深度学习领域事实上的标准编程语言），另一方面也可以支持使用 Python 编

¹⁷代码：<https://github.com/Thinklab-SJTU/pygmtools>，文档：<https://pygmtools.readthedocs.io/>

表 5-1 不同开源图匹配算法库的功能比较。

Table 5-1 Feature comparison among different open-source GM libraries.

算法库名称	pygmtools	LPMP	ZAC_GM	multiway
线性求解器数	2	1	2	1
二图求解器数	3	1	9	1
多图求解器数	3	1		4
神经求解器数	4	1		
支持新算法开发	✓	✓	✓	✓
支持深度学习	✓	✓		
支持 GPU	✓			
在线文档	✓	✓		
支持 pip install	✓	(✓)		
支持编程语言	Python	C++/Python	Matlab	Matlab

程的科学计算工作。此外，基于 Python 的软件生态也得到了很好的发展，已经有了 PyTorch^[245]、Numpy^[246]等优秀的开源项目。

表5-1总结了现有的开源图匹配算法库，表格中“(✓)”符号表示该功能还有继续完善的空间。目前现有的图匹配库中，大部分都是学术论文对应的开源代码（其中包括了论文实验中的比较方法）。这些开源代码更多聚焦于新算法的实现、评测以及论文实验结果的可复现性，而一定程度上忽略了代码的易用性。此外，在论文正式发表多年后，这些开源代码通常缺乏积极的更新、维护。这些局限性都为新接触相关领域的研究人员造成了一定阻碍。例如，ZAC_GM 是传统图匹配求解器论文^[117]的官方代码，multiway 则是多图求解器^[91]的官方代码。这两个 Matlab 代码仓库都提供了几种其他的二图匹配或多图匹配算法。需要注意的是，Matlab 是一个商业软件，因此它们实际上不属于“纯粹”的开源项目。

在作者看来，LPMP 是目前最符合开源精神的图匹配算法库。LPMP 包含了同一作者撰写的若干论文的具体实现，涵盖了二图匹配^[247]、多图匹配^[93]和深度学习图匹配^[82]等具体技术。然而，LPMP 包含的求解器不够全面，并且它的易用性、GPU 支持性有待改善。例如，LPMP 支持 pip install，但需要在本地配置编译器，抬高了用户的使用门槛。LPMP 的所有求解器代码仅支持 CPU 运行。作为对比，本节介绍的 pygmtools 可以通过简单的 pip 命令安装，对深度学习和 GPU 的支持可以根据需要灵活配置。此外，作者鼓励社区用户通过 GitHub 的合并请求（pull request）提交新的求解器，完善 pygmtools 的功能。

5.2.2 设计细节与工具用法

基于 Python 完善的机器学习和科学计算生态，pygmtools 工具能够覆盖尽可能多的用户和场景。pygmtools 旨在提供最先进的图匹配算法，包括二图匹配、多图匹配和最新的深度图匹配算法，支持多种计算后端（包括国内自主研发的深度

图 5-1 开源项目 *pygmtools* 的五层架构。Figure 5-1 Architecture of *pygmtools* project with 5 layers.

学习框架计图、飞桨和昇思，以及科研中常用的 Numpy、PyTorch 和 TensorFlow)。对于不同的图匹配算法和不同的计算后端，作者开发了统一的 API 接口。本工具遵循白玉兰开源倡议¹⁸，基于 Mulan PSL v2 许可证¹⁹，不存在任何闭源的依赖。作者提供了一个面向初学者的图匹配指南以及具体实例，涵盖了匹配一般图、匹配图像、对齐神经网络等任务。

作为一个标准的 Python 软件包，*pygmtools* 的安装配置非常简单，支持标准的 `pip install` 命令：

```
1 pip install pygmtools
```

这一命令会自动安装图匹配核心功能和其他依赖。安装完毕后，即可基于 Numpy 后端开展工作。若需要其他后端，也可按需进行配置。

处理实际的图匹配问题包括了问题建模、问题求解两个步骤，这两个步骤均可采用 *pygmtools* 处理。假设 A_1, A_2 是邻接矩阵， n_1, n_2 是节点数目，只需 9 行代码即可实现图匹配：

```
1 import numpy as np
2 import pygmtools as pygm
3 from functools import partial
4 conn1, edge1 = pygm.utils.dense_to_sparse(A1)
5 conn2, edge2 = pygm.utils.dense_to_sparse(A2)
6 aff = partial(pygm.utils.gaussian_aff_fn, sigma=1.) # 相似度函数
7 K = pygm.utils.build_aff_mat(None, edge1, conn1, None, edge2, conn2,
8     edge_aff_fn=aff) # 问题建模
9 S = pygm.rrwm(K, n1, n2) # 图匹配求解
X = pygm.hungarian(S) # 转换成离散解
```

¹⁸<http://www.baiyulan.org.cn/>

¹⁹<https://opensource.org/license/mulanpsl-2-0/>

问题建模步骤将节点和边的特征转换为相似度矩阵 $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ 。其中，函数 `pygm.utils.build_aff_mat` 根据用户配置，计算所有点到点、边到边的相似度分数，并将所有相似度得分映射到正确位置，得到了公式 (3.1) 中的目标函数。在这个例子中，问题求解步骤采用了 RRWM 求解器^[34]。此外，`pygmtools` 还支持调用其他求解器^[35,77,87,95]，包括3.2、3.3、4.3节的求解器。

`pygmtools` 的一个设计动机是为了满足科研社区不断增长的图匹配需求。在不同的场景下，考虑到计算设备、现有工具链、政策法规等因素，所需的计算框架（即后端）可能是不同的。`pygmtools` 设计了一个通用的前端 API 来适配大多数后端。其中，在科学计算领域广泛使用的 Numpy 能够适配最多的计算设备和操作系统，是默认后端；此外，还支持计图 (Jittor)、飞桨 (Paddle)、昇思 (MindSpore)、PyTorch、Tensorflow 等深度学习框架。开发者可以根据需要选择后端，并可基于后端调用 GPU。在代码设计中，前端 API 保持一致，后端框架可根据需要导入。

```
1 pygm.BACKEND = 'pytorch' # 全局修改默认后端
2 S = pygm.ngm(pygm.utils.from_numpy(K), n1, n2) # S是一个PyTorch张量
3 X = pygm.hungarian(S.detach().numpy(), backend='numpy') # X是一个
   Numpy数组
```

除了核心的问题建模、求解功能之外，`pygmtools` 还包含了以下功能：

- 评测集准。`pygmtools` 提供了图匹配科研的评测基准，以提供统一的数据接口和评估函数，创建公平的图匹配评测基准。它支持五个常用的图匹配数据集。
- 集成测试。为了确保开源代码质量，作者利用了 GitHub actions 功能运行自动测试，并确保不同后端的函数能够输出相同的结果。集成测试还使用了静态测试工具 `codecov`²⁰ 以保证覆盖率。本文写成时，自动集成测试已覆盖了 94% 的代码。

5.3 ThinkMatch：面向科研的深度图匹配系统

排列型组合优化问题的具体种类繁多，在研究一个新问题时，通常缺乏可复用的开源算法和测评基准。尤其是进入深度学习时代后，开发一个深度模型需要实现训练、测试、数据集接口、超参数调整、训练可视化等功能，比传统的非学习求解器更具挑战性。上述工程实践的负担实际上造成了对机器学习排列型组合优化研究的障碍，容易“劝退”新接触该领域的研究者。为了降低其他研究者的学习成本，本文的所有技术均已开源（包括完整的训练、测试代码和预训练模型）。特别地，对于3.2、3.3、4.3节关注的图匹配和二次指派问题，面向排列型组

²⁰<https://about.codecov.io/>

合优化与机器学习领域的科研人员，作者通过整合现有技术框架，开源了一个深度图匹配系统——ThinkMatch²¹。

5.3.1 开发动机与系统特色

实际上，ThinkMatch 的初始版本是作者在开展图匹配和二次指派研究时的实验平台。随着研究的开展，作者一方面认识到了整合代码、维护科研系统的重要性，另一方面，作者也了解到开源社区对最先进的深度学习模型、不同模型的公平评测等需求。因此，作者重新整理了现有代码，并发布了首个深度图匹配的科研系统。具体而言，ThinkMatch 的开发和开源主要出于以下目的：

- 为了未来的深度学习排列型组合优化研究提供基础性的模块；
- 提供学术界最先进、最前沿的深度图匹配模型的实现；
- 在不同数据集、不同参数设定下，公平地比较不同模型的性能。

与 pygmtools 的区别和联系。ThinkMatch 与 pygmtools 的目标使用场景不同。ThinkMatch 面向的是排列型组合优化和机器学习领域的科研人员，协助他们更高效地开发下一代深度图匹配算法。ThinkMatch 支持算法修改、模型配置、训练测试等科研中的常用功能。pygmtools 面向更广泛的下游应用，用户无需了解过多的算法细节即可方便地调用。pygmtools 更强调安装容易、用户友好、适配自主框架等特点，突出易用性。不过，ThinkMatch 的许多功能依赖于 pygmtools，包括数据集加载、测评，传统求解器调用等。另一方面，pygmtools 中的预训练神经网络权重是在 ThinkMatch 系统中训练的。作者希望，这两个项目可以在开源社区中持续发展、相互促进。

具体而言，面向科研的 ThinkMatch 系统具有以下特色：

- **支持多种深度图匹配神经网络。**具体而言，包括了3.2节介绍的 PIA-GM、PCA-GM 和 IPCA-GM 模型，3.3介绍的 NGM、NHGM、NMGGM 模型，4.3节介绍的 GANN-GM、GANN-MGM、GANN-MGM³ 模型。此外，还支持学术界的其他基准模型，例如 GMN^[16]和 BBGM^[82]。
- **支持优化求解、图像匹配的应用。**该系统支持优化数据集（例如 QAPLIB）的数据加载、预处理、训练测试等功能，同时支持深度图匹配方法在图像匹配问题的应用。在图像任务中，支持不同的 CNN 主干网络的灵活配置，例如 VGG 网络^[128]和 ResNet 网络^[220]，支持并行的图像加载、预处理等功能。

²¹代码：<https://github.com/Thinklab-SJTU/ThinkMatch>，文档：<https://thinkmatch.readthedocs.io/>

- **支持深度神经网络的训练和测试。**深度学习的训练和测试需要依赖较复杂的代码框架。ThinkMatch 提供了完整的训练、测试框架，其中训练流程支持灵活地配置优化器、学习率策略，同时支持监督学习和非监督学习。测试功能基于 pygtools 的基准评测函数，可以为所有方法提供公平的测试结果。
- **支持预训练模型加载功能。**训练过程中的模型权重会自动保存在特定目录下。如果训练程序中断，可以随时加载模型权重重启训练；此外，ThinkMatch 还支持迁移学习等研究方向。
- **支持模型、数据集的灵活定制。**通过修改 yaml 格式的配置文件，科研人员可以灵活地修改模型参数、数据集设定；通过仿照现有模型的源码，可以快速开发迭代新的模型，进行科研探索。
- **环境配置方便快捷。**深度学习研究的一大痛点是环境配置，处理 GPU 依赖、编译器等复杂的工具链需要耗费研究人员的大量精力。ThinkMatch 采用了容器技术，提供了预先打包的 docker 镜像，用户只需拉取在线镜像即可完成复杂的环境配置。

5.3.2 系统使用方法

本节介绍 ThinkMatch 系统的安装方式，以及主要功能（训练、测试）的使用方法。

安装环境。若设备上已安装 docker²²或 singularity²³等容器环境，推荐使用 docker 镜像安装。作者提供了一个预配置镜像，该镜像可在 docker 和其他兼容的容器（如 singularity）上运行。若使用 docker 作为运行容器，在 Linux 系统中，执行以下命令获取镜像：

```
1 docker pull runzhongwang/thinkmatch:torch1.6.0-cuda10.1-cudnn7-pygl  
   .6.3-pygtools0.3.5
```

针对不同的 GPU 型号和驱动版本，ThinkMatch 提供了多个 docker 镜像以针对不同的 PyTorch/CUDA/PyTorch-Geometric 版本组合。请根据用户的 GPU 型号及 CUDA 版本做出选择：

```
1 thinkmatch:torch1.6.0-cuda10.1-cudnn7-pygl.6.3-pygtools0.3.5 # GTX10  
   and RTX20 GPUs  
2 thinkmatch:torch1.7.1-cuda11.0-cudnn8-pygl.6.3-pygtools0.3.5  
3 thinkmatch:torch1.10.0-cuda11.3-cudnn8-pyg2.0.3-pygtools0.3.5 #  
   RTX30 GPUs
```

²²<https://www.docker.com/>

²³<https://docs.sylabs.io/guides/3.11/user-guide/>

训练与测试模型。运行命令

```
1 python train_eval.py --cfg path/to/your/yaml
```

其中“path/to/your/yaml”是实验配置的yaml文件地址，例如

```
1 experiments/vgg16_pca_voc.yaml
```

默认的配置文件均为yaml格式并存储于“experiment/”路径下。

加载预训练模型进行测试。ThinkMatch提供所包含的模型与数据集的预训练模型。下载模型到本地后，将以下内容添加到配置文件：

```
1 PRETRAINED_PATH: path/to/your/pretrained/weights
```

模型文件的命名规则为“前缀_CNN主干网络_模型名称_数据集名称.pt”。随后运行命令

```
1 python eval.py --cfg path/to/your/yaml
```

即可加载预训练模型并执行测试。

5.4 面向计图的图匹配求解框架设计与实现

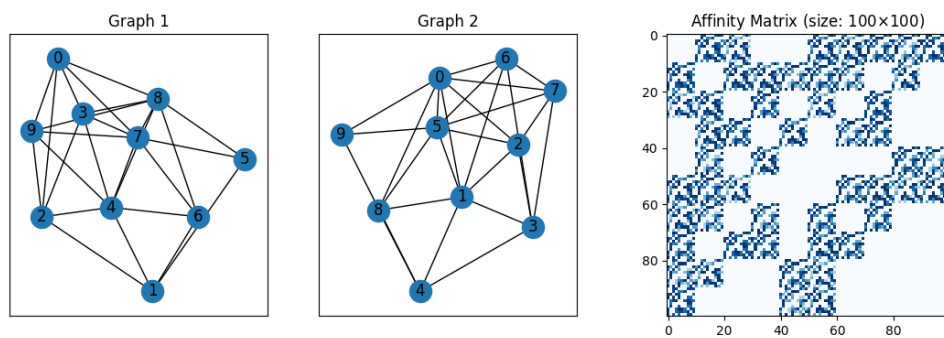
计图(Jittor)是清华大学图形学实验室开发的开源深度学习和高性能计算框架²⁴。计图吸收了PyTorch等成熟框架易编程、用户友好的优点，在计算层面引入动态编译(just-in-time, JIT)技术，在保留易用性的同时提高了框架的计算效率。在应用层面，计图可适配国内外多种主流的计算设备和操作系统。考虑到图匹配任务在深度学习、高性能计算等领域的广泛需求，面向计图日益壮大的用户社群，作者为计图设计并开发了图匹配求解功能。这是首个面向国内自主框架、支持完整图匹配功能的设计与实现。

5.4.1 技术特点

面向计图的图匹配求解功能作为pygmtools算法库的功能之一，具有以下技术特点：

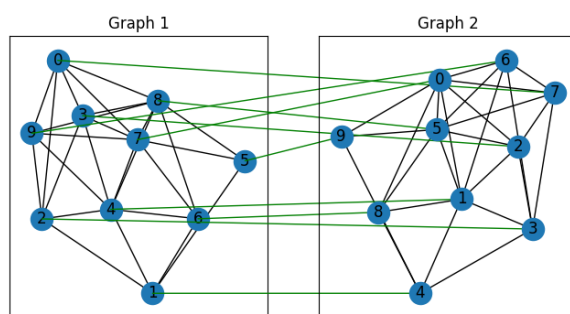
- **计图原生API调用。**图匹配求解器中，所有计算操作均调用计图的原生API，避免了额外的数据转换开销。基于原生API调用，用户可以最大化利用计图的计算效率，且在深度学习应用中可以最大程度的保留梯度信息。
- **支持动态编译。**动态编译(JIT)技术是计图框架的主要优势，作者开发的图匹配功能将这一优势完整地保留。在实际操作中，首先图匹配算子和其他算子一起被快速定义，随后在运行时执行动态编译，提升计算效率。

²⁴<https://cg.cs.tsinghua.edu.cn/jittor/>

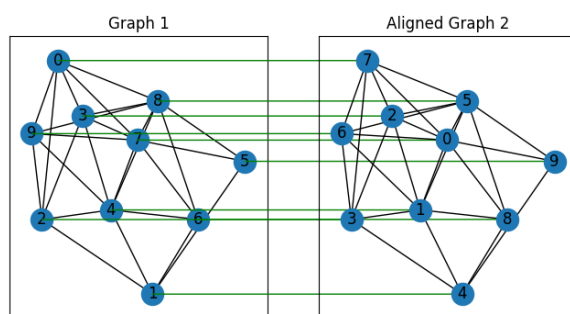


(a) 随机生成的同构图

(b) 相似度矩阵



(c) 匹配结果



(d) 对齐后的同构图

图 5-2 计图图匹配框架的应用案例可视化。**Figure 5-2** Visualization of an application of Jittor's graph matching framework.

- **完整的求解算法支持。** 对标 PyTorch、Numpy 等其他主流后端，本文作者为计图开发了完整的线性匹配、二图匹配、多图匹配、神经网络图匹配等功能。完整的求解算法可以支持计图在众多复杂场景下的应用。
- **支持科学计算与深度学习。** 高性能科学计算和深度学习是计图框架设计的两大应用场景，也是图匹配算法的主要应用领域。图匹配算法与计图的结合将进一步拓宽相关框架的应用场景，助力下游领域的研究和应用。

5.4.2 应用案例

在线文档²⁵中，作者提供了若干基于计图编写的图匹配应用案例。此处以经典的同构图对齐任务为例进行说明：

第一步：导入计图等依赖库。

```
1 import jittor as jt # 导入计图
2 import pygtools as pygm
3 pygm.BACKEND = 'jittor' # 设置计图后端
4 jt.flags.use_cuda = jt.has_cuda # 设置GPU支持
```

第二步：随机生成两个同构的图，但将节点顺序随机打乱。可视化的结果如图5-2(a)所示，从视觉上看，这两个图结构毫无关联。

```
1 X_gt = jt.zeros((num_nodes, num_nodes))
2 X_gt[jt.arange(0, num_nodes, dtype=jt.int64), jt.randperm(num_nodes)]
   = 1
3 A1 = jt.rand(num_nodes, num_nodes) # 图1的邻接矩阵
4 A2 = jt.matmul(jt.matmul(X_gt.t(), A1), X_gt) # 图2的邻接矩阵
5 n1 = n2 = jt.Var([num_nodes])
```

第三步：基于连边关系计算相似度并构造相似度矩阵。相似度矩阵的可视化的结果如图5-2(b)所示，该矩阵是公式(3.1)中的参数 \mathbf{K} 。

```
1 conn1, edge1 = pygm.utils.dense_to_sparse(A1)
2 conn2, edge2 = pygm.utils.dense_to_sparse(A2)
3 import functools
4 gaussian_aff = functools.partial(pygm.utils.gaussian_aff_fn, sigma
   =.1) # 设置相似度函数
5 K = pygm.utils.build_aff_mat(None, edge1, conn1, None, edge2, conn2,
   n1, None, n2, None, edge_aff_fn=gaussian_aff)
```

第四步：调用计图框架下的求解算法，求解匹配关系。节点匹配关系如图5-2(c)中的绿色连线所示。

```
1 X = pygm.rrwm(K, n1, n2)
```

随后，根据匹配结果将图2的节点重新排列，得到图5-2(d)。通过图匹配算法对齐后，两个图的同构关系在可视化结果中非常明显。

上述应用案例希望通过具体实例，向用户展示计图框架下的图匹配求解器用法。此外，在线文档还包含了子图匹配 (subgraph matching)、种子图匹配 (seeded graph matching)、图像匹配 (image matching)、深度学习图像匹配 (deep image matching)、神经网络融合 (neural network fusion) 等主题。

²⁵https://pygtools.readthedocs.io/en/latest/auto_examples/index.html#jittor-backend-examples

5.5 本章小结

分别面向一般的图匹配应用，以及面向专业的科研人员，本文作者分别开发了 `pygmtools` 与 `ThinkMatch` 工具。其中，`pygmtools` 强调易用性，安装简单，为常用的二图匹配、多图匹配、深度学习图匹配算法开发了统一的编程接口。`ThinkMatch` 强调科研工具的完备性，涵盖了目前为止学术界广泛认可的最先进模型与算法，并提供其在各种经典数据集上的训练与评估功能。此外，本文面向国内自主研发的开源框架“计图”开发了完善的图匹配功能，并提供了详细的应用案例。开源后，上述工具受到了开源社区的一定关注。截至本文写成时（2023年3月），上述开源仓库在 GitHub 上共收获了 877 次收藏（star）、119 次派生（fork），共有 12 位独立贡献者。`pygmtools` 被全球用户总计下载了 3.1 万次。此外，`ThinkMatch` 已经成为深度图匹配研究事实性的基础系统，国内外一系列已公开代码的论文^[82,248-250]均是基于 `ThinkMatch` 进一步开发、扩展的。

上述开源工具还存在许多待提高的地方，开源社区的讨论和反馈也促进了这些工具的不断完善。作者希望，这些开源工作可以进一步促进机器学习与排列型组合优化领域的深入研究，同时带动图匹配技术与自主深度学习框架在各种下游场景的应用。

第六章 总结与展望

6.1 工作总结

本文主要研究了机器学习用于求解排列型组合优化的理论和方方法论。由于排列型组合优化刻画了匹配、排序、路径规划等常见场景，研究排列型组合优化的求解具有广泛的现实意义；此外，求解最优解通常具有 NP 难的复杂性，在理论层面，研究排列型组合优化具有足够的挑战性。近年来，机器学习尤其是深度学习技术的快速发展，为排列型组合优化这一历久弥新的研究领域带来了新的机遇。数据驱动的机器学习方法有潜力替代传统求解算法中冗杂的人工设计，这已经成为国际学术界的普遍共识。本文分别针对纯机器学习的求解方法，以及机器学习与传统求解器融合的方法开展了研究，初步证明了这两种方法论的可行性，在若干实际问题中展示了机器学习方法的优越性。

特别地，针对纯机器学习的排列型组合优化求解方法：

- 3.2节面向图匹配问题首次设计了一种图嵌入方法，将图匹配的高阶图结构信息嵌入到节点特征。基于此方法，原本 NP 难的图匹配问题简化为了可在多项式时间内精确求解的线性指派问题，采用 Sinkhorn 网络进行可微分求解。在仿真数据和真实图像的匹配任务上，基于深度图嵌入的图匹配求解方法超越了现有的深度图匹配以及传统算法。
- 3.3节面向最一般的 Lawler 形二次指派问题，设计了一种纯神经网络求解器。通过将问题参数转化成等价的伴随图形式，原问题转变成了在伴随图上求解顶点分类问题。利用图神经网络求解顶点分类的能力，作者设计了一种求解 Lawler 形二次指派的纯神经网络方法。在优化任务上，神经网络求解器超越了传统算法；此外，神经网络求解器还拥有嵌入其他端到端学习框架的灵活性，在图像匹配任务上达到了最高的匹配精度。
- 3.4节面向基数约束的排列型组合优化问题，对纯神经网络求解方法的理论和实验性质开展了研究。研究揭示了一定条件下，神经网络的约束满足程度与求解性能的正相关性，并在理论上推导了三种输出层网络对应的约束满足程度。在纯优化问题的实验结果显示，合理设计的神经网络求解器可与商用求解器性能持平甚至更好；此外，在预测性投资组合优化任务上，作者再次展示了纯神经网络可嵌入端到端深度学习框架的优势。

针对融合传统算法的排列型组合优化学习求解方法：

- 4.2节以图编辑距离的传统 A* 求解器为基础，设计了一种神经网络模块以替代传统算法流程中的人工启发式预测。与 A* 算法的动态规划特性相对应，神经网络能够提供更高效的剪枝策略，神经网络本身也可通过复用

中间结果进行提速，这些特点带来了算法效率的显著提升。此外，神经网络的高预测精度使得实验中融合求解器的精度损失几乎可以忽略不计。

- 4.3节以现有的图匹配深度学习框架为基础，设计了一种将传统求解器分支与深度图匹配分支差异最小化的无监督学习框架。该框架降低了深度图匹配对标注数据的依赖，且具有灵活性，对不同的求解器、不同的匹配条件均适用。实验中，将无监督学习的模型直接用于小规模数据集上，可以达到与监督学习方法接近的精度；在大规模数据集上通过预训练+微调模式，匹配精度超越了无预训练的监督学习方法。
- 4.4节设计了一个通用的双层优化融合框架，将强化学习智能体与传统启发式求解器分别视为上、下层优化，上层的强化学习智能体在不扩大问题可行域的前提下修改问题结构，下层的传统启发式求解器修改后的问题。这种双层融合框架降低了对深度学习模型容量的需求，减轻了强化学习的稀疏奖励问题，还提升了现有启发式方法的求解性能。实验证明，这种双层融合框架的求解性能优于单独的单层强化学习模型或启发式求解器。

在本文的研究视角下，纯机器学习方法与融合方法的核心优势分别如下：纯机器学习方法具有嵌入其他端到端深度学习框架的能力（即可微分），例如3.2节和3.3节中，纯机器学习求解器可以与深度感知模块端到端地结合。3.4节中，纯机器学习求解器可以和资产价格预测网络结合，进行端到端的学习。在这种前端的预测/感知与后端的决策一体化的任务中，纯机器学习的算法可凭借其端到端的能力、与前端模块高度耦合，展现其性能优势。融合方法在求解纯优化问题时性能更强，例如，4.2节中，A*搜索算法与精确的神经网络预测共同带来了融合算法的高精度；4.4节中，传统的启发式算法更是为双层融合框架带来了性能的兜底保证。在纯优化问题的求解中，凭借传统算法的理论保证和先验知识，辅助以深度学习方法的高效性，通常可以获得更高的求解精度。此外，特定的融合方式也可提升纯机器学习方法的性能，例如4.3节的无监督学习框架。在方法论的角度，纯机器学习方法更加灵活，定义训练集、目标函数后即可端到端地训练；融合方法更加精细化，需要深入理解具体问题的传统求解算法，设计的融合方法才会高效。

分别面向图匹配的下游科研与应用、面向排列型组合优化与机器学习的深入研究、面向国内自主研发的深度学习框架“计图”，本文整理并开源了排列型组合优化（特别是图匹配和二次指派）问题的求解框架。这些开源框架已经受到了来自开源社区和科研人员的广泛欢迎。

最后，作者也认识到，站在优化理论的角度，机器学习并不能改变多数排列型组合优化问题的NP难属性。本文关注的是在具体实践中，如何提升困难问题、大规模实例的求解效率。这种实践上的性能提升是可行的，以现有的传统优化算法为例，它们的理论近似比通常刻画的是最坏情况的近似程度，在实践中，这些

传统算法的性能通常远超它们的理论近似比。此外，发掘组合优化问题的最优结构也是引入机器学习的重要动机，本文4.4节在这个方向进行了初步尝试，未来还有更多空间值得研究、探索。

6.2 未来展望

本文分别在技术、理论和开源算法层面，对排列型组合优化问题的机器学习求解做出了一定程度的研究和贡献。作者认为，通用性是机器学习求解方法在下阶段的发展重点，与其他 AI 领域热点问题的交叉、结合也是重要的发展方向。具体而言，还有以下方向值得在后续工作中继续研究和探索：

- **更通用的纯机器学习求解范式。**本文研究的纯机器学习求解范式更多地针对特定的排列型问题（例如图匹配、二次指派、基数约束）设计专用的网络结构，特别地，上述排列约束均可采用 Sinkhorn 算法进行处理。如何将这一方法论进一步泛化，更通用地处理（排列型）组合优化问题，是一个有潜力的研究方向。此外，在扩展到更通用场景后，如何保持理论上可控的约束违反程度，也是下一步研究的挑战。
- **更通用的整数规划融合求解。**在所有组合优化问题中，（混合）整数规划是一类最典型的问题，许多实际问题可以写成整数规划形式，进而采用先进的商用求解器进行求解。虽然在整数规划求解中融合机器学习已经得到了学术界的广泛关注^[67,69,230,251]，但是，商用求解器依然是目前实际落地的首选。作者认为，进一步发掘整数规划求解器中的可融合流程（包括但不限于预求解、分支策略、割平面、启发式等），将会是开源的融合算法突破通用性瓶颈、迈向实际应用的发展方向。
- **与大语言模型的相互赋能。**2022 年底，ChatGPT 的横空出世引爆了公众对新一轮人工智能革命的无限遐想。目前“大语言模型”所展现的语言组织、意图理解、对话能力有望成为新的 AI “基础设施”，但大语言模型缺乏数学、逻辑能力的缺陷亦有目共睹。作者认为，本文所关注的优化、求解能力可以与大语言模型形成优势互补。具体地，对真实问题进行数学建模需要大量的专业知识，或可利用大语言模型从自然语言和多轮对话中完成自动化建模；此外，目前的大语言模型缺乏逻辑、数学计算等能力，优化求解结果或可作为下一轮输入，赋能大语言模型的智能分析和决策。
- **下游领域的典型应用。**本文探索了排列型组合优化求解在机器视觉、运筹优化、数据挖掘等计算机相关场景的应用。作者认为，人工智能驱动的科学（AI for Science）是未来人工智能研究的重要方向之一。考虑到优化问题在基础科学中广泛存在，在未来，作者计划将机器学习求解排列型组合优化的范式用于以生命科学为代表的基础科学领域。例如，蛋白质的

对接 (docking) 可以视为匹配问题, 联合用药可以视为 (目标函数黑盒的) 基数约束优化问题。

- **进一步发​​展开源社区。** 开源社区以及网络上的公开资源曾为作者的研究提供了极大的帮助。开源工作的未来任务包括积极维护代码、进一步增强易用性、紧跟研究热点等。作者希望, 本文的开源工作可以在开源社区和科研人员中形成良性循环, 进一步促进机器学习排列型组合优化领域的发展以及相关开源工具链的完善。

参考文献

- [1] HOPFIELD J J, TANK D W. "Neural" computation of decisions in optimization problems[J]. *Biological cybernetics*, 1985, 52(3): 141-152.
- [2] BENGIO Y, LODI A, PROUVOST A. Machine learning for combinatorial optimization: a methodological tour d' horizon[J]. *European Journal on Operational Research*, 2020.
- [3] 郭田德, 韩丛英. 从数值最优化方法到学习最优化方法[J]. *运筹学学报*, 2019(1-12).
- [4] 郭田德, 韩丛英, 唐思琦. 组合优化问题的机器学习求解方法[M]. 北京: 科学出版社, 2019.
- [5] JACOBI C G J. Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*[J]., 1846.
- [6] KUHN H W. A tale of three eras: The discovery and rediscovery of the Hungarian Method[J]. *European Journal on Operational Research*, 2012, 219(3): 641-651.
- [7] BIRKHOFF G. Tres observaciones sobre el algebra lineal[J]. *Univ. Nac. Tucuman, Ser. A*, 1946, 5: 147-154.
- [8] VON NEUMANN J. A certain zero-sum two-person game equivalent to the optimal assignment problem[J]. *Contributions to the Theory of Games*, 1953, 2(0): 5-12.
- [9] DANTZIG G B. Application of the simplex method to a transportation problem[J]. *Activity analysis and production and allocation*, 1951.
- [10] KUHN H W. The Hungarian method for the assignment problem[J]. *Naval research logistics quarterly*, 1955, 2(1-2): 83-97.
- [11] KÖNIG D. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre[J]. *Mathematische Annalen*, 1916, 77(4): 453-465.
- [12] EGERVÁRY J. Matrixok kombinatorius tulajdonságairól.[J]. *Matematikai és Fizikai Lapok*, 1931.
- [13] MUNKRES J. Algorithms for the assignment and transportation problems[J]. *Journal of the Society for Industrial & Applied Mathematics*, 1957, 5(1): 32-38.
- [14] LOIOLA E M, de ABREU N M M, BOAVENTURA-NETTO P O, et al. A survey for the quadratic assignment problem[J]. *European Journal on Operational Research*, 2007, 176(2): 657-690.

- [15] DILLEN COURT M B. Finding Hamiltonian cycles in Delaunay triangulations is NP-complete[J]. Discrete Applied Mathematics, 1996, 64(3): 207-217.
- [16] ZANFIR A, SMINCHISESCU C. Deep Learning of Graph Matching[C]//Computer Vision and Pattern Recognition. 2018: 2684-2693.
- [17] SARLIN P E, DETONE D, MALISIEWICZ T, et al. Superglue: Learning feature matching with graph neural networks[C]//Computer Vision and Pattern Recognition. 2020: 4938-4947.
- [18] HE J, HUANG Z, WANG N, et al. Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking[C]//Computer Vision and Pattern Recognition. 2021: 5299-5309.
- [19] SONG H, JIN Y, CHENG Y, et al. Learning interlaced sparse Sinkhorn matching network for video super-resolution[J]. Pattern Recognition, 2022, 124: 108475.
- [20] LI W, LIU X, YUAN Y. SIGMA++: Improved Semantic-complete Graph Matching for Domain Adaptive Object Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.
- [21] XU C, XU J, DONG Z, et al. Semantic Sentence Matching via Interacting Syntax Graphs[C]//International Conference on Computational Linguistics. 2022: 938-949.
- [22] PENG Y, CHOI B, XU J. Graph Edit Distance Learning via Modeling Optimum Matchings with Constraints.[C]//International Joint Conference on Artificial Intelligence. 2021: 1534-1540.
- [23] CHEN B, LI C, DAI H, et al. Retro*: Learning Retrosynthetic Planning with Neural Guided A* Search[C]//International Conference on Machine Learning. 2020.
- [24] LIU C, LOU C, WANG R, et al. Deep neural network fusion via graph matching with applications to model ensemble and federated learning[C]//International Conference on Machine Learning. 2022: 13857-13869.
- [25] VINYALS O, FORTUNATO M, JAITLY N. Pointer networks[C]//Neural Information Processing Systems. 2015: 2692-2700.
- [26] ZHANG J, YU P, ZHOU F, et al. Resource and delay aware fine-grained service offloading in collaborative edge computing[J]. Computer Networks, 2022, 218: 109383.

- [27] EDWARDS C. A branch and bound algorithm for the Koopmans-Beckmann quadratic assignment problem[G]//Combinatorial optimization II. Springer, 1980: 35-52.
- [28] HAHN P, GRANT T, HALL N. A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method[J]. EJOR, 1998, 108(3): 629-640.
- [29] BALAS E, TOTH P. Branch and bound methods for the traveling salesman problem[J]., 1983.
- [30] GAMRATH G, ANDERSON D, BESTUZHEVA K, et al. The SCIP Optimization Suite 7.0[R/OL]. Optimization Online. 2020. http://www.optimization-online.org/DB_HTML/2020/03/7705.html.
- [31] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual[EB/OL]. 2021. <https://www.gurobi.com>.
- [32] GOLD S, RANGARAJAN A. Softmax to softassign: neural network algorithms for combinatorial optimization[J]. J. Artif. Neural Netw, 1996, 2(4): 381-399.
- [33] GOLD S, RANGARAJAN A. A graduated assignment algorithm for graph matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(4): 377-388.
- [34] CHO M, LEE J, LEE K M. Reweighted random walks for graph matching[C] //European Conference on Computer Vision. 2010: 492-505.
- [35] LEORDEANU M, HEBERT M. A Spectral Technique for Correspondence Problems using Pairwise Constraints[C]//International Conference on Computer Vision. 2005: 1482-1489.
- [36] ROSENKRANTZ D J, STEARNS R E, LEWIS P M, II. An analysis of several heuristics for the traveling salesman problem[J]. SIAM journal on computing, 1977, 6(3): 563-581.
- [37] HELSGAUN K. An effective implementation of the Lin–Kernighan traveling salesman heuristic[J]. European Journal of Operational Research, 2000, 126(1): 106-130.
- [38] RIESEN K, FANKHAUSER S, BUNKE H. Speeding Up Graph Edit Distance Computation with a Bipartite Heuristic.[C]//Mining and Learning with Graphs. 2007: 21-24.
- [39] BOUGLEUX S, GAÜZERE B, BRUN L. Graph edit distance as a quadratic program[C]//International Conference on Pattern Recognition. 2016: 1701-1706.

- [40] GRANDL R, ANANTHANARAYANAN G, KANDULA S, et al. Multi-Resource Packing for Cluster Schedulers[J]. SIGCOMM Comput. Commun. Rev., 2014, 44(4): 455-466.
- [41] VAN LAARHOVEN P J, AARTS E H, van LAARHOVEN P J, et al. Simulated annealing[M]. Springer, 1987.
- [42] DORIGO M, BIRATTARI M, STUTZLE T. Ant colony optimization[J]. IEEE computational intelligence magazine, 2006, 1(4): 28-39.
- [43] GLOVER F, LAGUNA M. Tabu search[M]. Springer, 1998.
- [44] DAVIS L. Handbook of genetic algorithms[J]., 1991.
- [45] AUWATANAMONGKOL S. Inexact graph matching using a genetic algorithm for image recognition[J]. Pattern Recognition Letters, 2007, 28(12): 1428-1437.
- [46] ADAMCZEWSKI K, SUH Y, LEE K. Discrete Tabu Search for Graph Matching[C]//International Conference on Computer Vision. 2015.
- [47] IYER S K, SAXENA B. Improved genetic algorithm for the permutation flow-shop scheduling problem[J]. Computers & Operations Research, 2004, 31(4): 593-606.
- [48] POTVIN J Y. Genetic algorithms for the traveling salesman problem[J]. Annals of Operations Research, 1996, 63: 337-370.
- [49] 周雅兰, 王甲海, 黄聪. 求解排列问题的分布估计离散粒子群优化算法[J]. 电子学报, 2014(561-571).
- [50] 宁秀. 置换型组合优化问题的交互熵算法研究[D]. 清华大学, 2019.
- [51] 李章洪, 梁晓磊, 田梦丹, 等. 求解排列组合问题的解空间动态缩减策略[J]. 计算机应用, 2020(2016-2020).
- [52] SMITH K A. Neural networks for combinatorial optimization: a review of more than a decade of research[J]. Informs journal on Computing, 1999, 11(1): 15-34.
- [53] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. ArXiv preprint, 2017.
- [54] KHALIL E, DAI H, ZHANG Y, et al. Learning combinatorial optimization algorithms over graphs[C]//Neural Information Processing Systems. 2017: 6351-6361.
- [55] KWON Y D, CHOO J, YOON I, et al. Matrix encoding networks for neural combinatorial optimization[J]. Neural Information Processing Systems, 2021, 34: 5138-5149.

- [56] LIU C, WANG R, JIANG Z, et al. Revocable deep reinforcement learning with affinity regularization for outlier-robust graph matching[C]//International Conference on Learning and Representations. 2023.
- [57] MAO H, ALIZADEH M, MENACHE I, et al. Resource management with deep reinforcement learning[C]//Proceedings of the 15th ACM workshop on hot topics in networks. 2016: 50-56.
- [58] BAI Y, XU D, WANG A, et al. Fast detection of maximum common subgraph via deep q-learning[J]. ArXiv preprint, 2020.
- [59] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. ArXiv preprint, 2013.
- [60] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. ArXiv preprint, 2017.
- [61] KWON Y D, CHOO J, KIM B, et al. Pomo: Policy optimization with multiple optima for reinforcement learning[J]. Neural Information Processing Systems, 2020, 33: 21188-21198.
- [62] CHEN X, TIAN Y. Learning to perform local rewriting for combinatorial optimization[J]. Neural Information Processing Systems, 2019, 32: 6281-6292.
- [63] BARRETT T, CLEMENTS W, FOERSTER J, et al. Exploratory combinatorial optimization with reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 34: 04. 2020: 3243-3250.
- [64] KARALIAS N, LOUKAS A. Erdos Goes Neural: an Unsupervised Learning Framework for Combinatorial Optimization on Graphs[C]//Neural Information Processing Systems. 2020.
- [65] JOSHI C K, CAPPART Q, ROUSSEAU L M, et al. Learning TSP requires rethinking generalization[J]. ArXiv preprint, 2020.
- [66] CAI Q, HANG W, MIRHOSEINI A, et al. Reinforcement learning driven heuristic optimization[J]. ArXiv preprint, 2019.
- [67] KHALIL E, LE BODIC P, SONG L, et al. Learning to branch in mixed integer programming[C]//AAAI Conference on Artificial Intelligence: vol. 30: 1. 2016.
- [68] KOBER J, PETERS J. Imitation and reinforcement learning[J]. IEEE Robotics & Automation Magazine, 2010, 17(2): 55-62.

- [69] GASSE M, CHÉTELAT D, FERRONI N, et al. Exact combinatorial optimization with graph convolutional neural networks[J]. *Neural Information Processing Systems*, 2019.
- [70] AMOS B, KOLTER J Z. Optnet: Differentiable optimization as a layer in neural networks[C]//*International Conference on Machine Learning*. 2017: 136-145.
- [71] AGRAWAL A, AMOS B, BARRATT S, et al. Differentiable convex optimization layers[J]. *Advances in neural information processing systems*, 2019, 32.
- [72] POGANČIĆ M V, PAULUS A, MUSIL V, et al. Differentiation of blackbox combinatorial solvers[C]//*International Conference on Learning and Representations*. 2019.
- [73] SAKAUE S. Differentiable greedy algorithm for monotone submodular maximization: Guarantees, gradient estimators, and applications[C]//*International Conference on Artificial Intelligence and Statistics*. 2021: 28-36.
- [74] BERTHET Q, BLONDEL M, TBOUL O, et al. Learning with differentiable perturbed optimizers[J]. *Neural Information Processing Systems*, 2020, 33: 9508-9519.
- [75] PAULUS A, ROLÍNEK M, MUSIL V, et al. Comboptnet: Fit the right np-hard problem by learning integer programming constraints[C]//*International Conference on Machine Learning*. 2021: 8443-8453.
- [76] ELMACHTOUB A N, GRIGAS P. Smart “predict, then optimize”[J]. *Management Science*, 2022, 68(1): 9-26.
- [77] YU T, WANG R, YAN J, et al. Learning deep graph matching with channel-independent embedding and Hungarian attention[C]//*International Conference on Learning and Representations*. 2020.
- [78] FEY M, LENSSEN J E, MORRIS C, et al. Deep graph matching consensus[C]//*International Conference on Learning and Representations*. 2020.
- [79] LIU H, WANG T, LI Y, et al. Joint Graph Learning and Matching for Semantic Feature Correspondence[J]. *Pattern Recognition*, 2023, 134: 109059.
- [80] KOOL W, van HOOFF H, WELLING M. Attention, Learn to Solve Routing Problems![C]//*International Conference on Learning Representations*. 2019: 1-25.
- [81] XIE Y, DAI H, CHEN M, et al. Differentiable Top-k with Optimal Transport[C]//*Neural Information Processing Systems: vol. 33*. Curran Associates, Inc., 2020: 20520-20531.

- [82] ROLÍNEK M, SWOBODA P, ZIETLOW D, et al. Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers[C]//European Conference on Computer Vision. 2020: 407-424.
- [83] YU T, WANG R, YAN J, et al. Deep Latent Graph Matching[C]//International Conference on Machine Learning. 2021: 12187-12197.
- [84] LU H, ZHANG X, YANG S. A learning-based iterative method for solving vehicle routing problems[C]//International Conference on Learning and Representations. 2019.
- [85] TIAN Y, YAN J, ZHANG H, et al. On the convergence of graph matching: Graduated assignment revisited[C]//European Conference on Computer Vision. 2012: 821-835.
- [86] COUR T, SRINIVASAN P, SHI J. Balanced Graph Matching[C]//Neural Information Processing Systems. 2006.
- [87] LEORDEANU M, HEBERT M, SUKTHANKAR R. An integer projected fixed point method for graph matching and map inference[C]//Neural Information Processing Systems. 2009: 1114-1122.
- [88] YU T, YAN J, WANG Y, et al. Generalizing graph matching beyond quadratic assignment model[C]//Neural Information Processing Systems. 2018: 853-863.
- [89] 江波. 基于图理论与稀疏约束模型的图像表示、匹配与识别[D]. 安徽大学, 2015.
- [90] 王福东. 基于泛函表示的图匹配研究[D]. 武汉大学, 2021.
- [91] WANG Q, ZHOU X, DANIILIDIS K. Multi-Image Semantic Matching by Mining Consistent Features[C]//Computer Vision and Pattern Recognition. 2018: 685-694.
- [92] BERNARD F, THUNBERG J, SWOBODA P, et al. HiPPI: Higher-Order Projected Power Iterations for Scalable Multi-Matching[C]//International Conference on Computer Vision. 2019: 10284-10293.
- [93] SWOBODA P, MOKARIAN A, THEOBALT C, et al. A convex relaxation for multi-graph matching[C]//Computer Vision and Pattern Recognition. 2019: 11156-11165.
- [94] YAN J, WANG J, ZHA H, et al. Consistency-driven alternating optimization for multigraph matching: A unified approach[J]. IEEE Transactions on Image Processing, 2015, 24(3): 994-1009.

- [95] YAN J, CHO M, ZHA H, et al. Multi-Graph Matching via Affinity Optimization with Graduated Consistency Regularization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(6): 1228-1242.
- [96] YAN J, LI C, LI Y, et al. Adaptive Discrete Hypergraph Matching[J]. IEEE Transactions on Cybernetics, 2017, 48(2): 765-779.
- [97] NGUYEN Q, GAUTIER A, HEIN M. A Flexible Tensor Block Coordinate Ascent Scheme for Hypergraph Matching[C]//Computer Vision and Pattern Recognition. 2015: 5270-5278.
- [98] YAN J, YIN X C, LIN W, et al. A Short Survey of Recent Advances in Graph Matching[C]//International Conference on Multimedia Retrieval. 2016: 167-174.
- [99] YAN J, YANG S, HANCOCK E R. Learning for Graph Matching and Related Combinatorial Optimization Problems[C]//International Joint Conference on Artificial Intelligence. 2020: 4988-4996.
- [100] CAETANO T, MCAULEY J, CHENG L, et al. Learning graph matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(6): 1048-1058.
- [101] CHO M, ALAHARI K, PONCE J. Learning Graphs to Match[C]//International Conference on Computer Vision. 2013: 25-32.
- [102] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database[C]//Computer Vision and Pattern Recognition. 2009: 248-255.
- [103] WANG T, LIU H, LI Y, et al. Learning Combinatorial Solver for Graph Matching[C]//Computer Vision and Pattern Recognition. 2020: 7568-7577.
- [104] MILLER C E, TUCKER A W, ZEMLIN R A. Integer programming formulation of traveling salesman problems[J]. Journal of the ACM, 1960, 7(4): 326-329.
- [105] DAI H, DAI B, SONG L. Discriminative embeddings of latent variable models for structured data[C]//International Conference on Machine Learning. 2016: 2702-2711.
- [106] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. ArXiv preprint, 2013.
- [107] LI Z, CHEN Q, KOLTUN V. Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search[C]//BENGIO S, WALLACH H, LAROCHELLE H, et al. Neural Information Processing Systems: vol. 31. Curran Associates, Inc., 2018.

- [108] QIU R, SUN Z, YANG Y. DIMES: A Differentiable Meta Solver for Combinatorial Optimization Problems[J]. ArXiv preprint, 2022.
- [109] LIU B. Facility Location Problem[M]//Theory and Practice of Uncertain Programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 157-165.
- [110] CHEN W, SUN X, ZHANG J, et al. Network Inference and Influence Maximization from Samples[C]//International Conference on Machine Learning. 2021.
- [111] CHANG T J, MEADE N, BEASLEY J E, et al. Heuristics for cardinality constrained portfolio optimisation[J]. Computers & Operations Research, 2000, 27(13): 1271-1302.
- [112] FUJISHIGE S. Submodular Functions and Optimization[M]. Elsevier, 1991.
- [113] ADAMS R, ZEMEL R. Ranking via Sinkhorn Propagation[J]. ArXiv:1106.1925, 2011.
- [114] CUTURI M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport[J]. Neural Information Processing Systems, 2013: 2292-2300.
- [115] SINKHORN R, RANGARAJAN A. A relationship between arbitrary positive matrices and doubly stochastic matrices[J]. Ann. Math. Statistics, 1964, 35(2): 876-879.
- [116] CUTURI M, TEBOUL O, VERT J P. Differentiable Ranking and Sorting using Optimal Transport[C]//Advances in Neural Information Processing Systems: vol. 32. 2019: 6858-6868.
- [117] WANG F, XUE N, YU J G, et al. Zero-Assignment Constraint for Graph Matching With Outliers[C]//Computer Vision and Pattern Recognition. 2020.
- [118] LAWLER E L. The quadratic assignment problem[J]. Management Science, 1963, 9(4): 586-599.
- [119] CHERTOK M, KELLER Y. Efficient high order matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(12): 2205-2215.
- [120] DUCHENNE O, BACH F, KWEON I, et al. A Tensor-Based Algorithm for High-Order Graph Matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12): 2383-2395.
- [121] YAN J, ZHANG C, ZHA H, et al. Discrete Hyper-graph Matching[C]//Computer Vision and Pattern Recognition. 2015: 1520-1528.
- [122] ZASS R, SHASHUA A. Probabilistic Graph and Hypergraph Matching[C]//Computer Vision and Pattern Recognition. 2008: 1-8.

- [123] LEE J, CHO M, LEE K M. Hyper-graph Matching via Reweighted Random Walks[C]//Computer Vision and Pattern Recognition. 2011: 1633-1640.
- [124] YAN J, WANG J, ZHA H, et al. Consistency-driven alternating optimization for multigraph matching: A unified approach[J]. IEEE Transactions on Image Processing, 2015, 24(3): 994-1009.
- [125] CHEN Y, GUIBAS L, HUANG Q. Near-optimal joint object matching via convex relaxation[C]//International Conference on Machine Learning. 2014: 100-108.
- [126] HUANG Q, GUIBAS L. Consistent Shape Maps via Semidefinite Programming[C]//Proc. Eurographics Symposium on Geometry Processing (SGP). 2013.
- [127] PACHAURI D, KONDOR R, SINGH V. Solving the multi-way matching problem by permutation synchronization[C]//Neural Information Processing Systems. 2013: 1860-1868.
- [128] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//International Conference on Learning and Representations. 2014.
- [129] BROMLEY J, GUYON I, LECUN Y, et al. Signature verification using a “siamese” time delay neural network[C]//Neural Information Processing Systems. 1994.
- [130] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//Computer Vision and Pattern Recognition. 2009: 248-255.
- [131] FISCHLER M A, BOLLES R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.
- [132] ZHANG Z. Iterative point matching for registration of free-form curves and surfaces[J]. International Journal Computer Vision, 1994, 13(2): 119-152.
- [133] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[J]. International Conference on Learning and Representations, 2017.
- [134] SANTA CRUZ R, FERNANDO B, CHERIAN A, et al. Visual permutation learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(12): 3100-3114.
- [135] SERRATOSA F, SOLÉ-RIBALTA A, CORTÉS X. Automatic learning of edit costs based on interactive and adaptive graph recognition[G]//International Workshop on Graph-Based Representations in Pattern Recognition. 2011: 152-163.

- [136] MIKOLAJCZYK K, SCHMID C. Scale & affine invariant interest point detectors[J]. *International Journal Computer Vision*, 2004, 60(1): 63-86.
- [137] BOURDEV L, MALIK J. Poselets: Body part detectors trained using 3d human pose annotations[C]//*International Conference on Computer Vision*. 2009: 1365-1372.
- [138] EVERINGHAM M, VAN GOOL L, WILLIAMS C K, et al. The pascal visual object classes (voc) challenge[J]. *International Journal Computer Vision*, 2010, 88(2): 303-338.
- [139] PUNNEN A P, KABADI S N. A linear time algorithm for the Koopmans-Beckmann QAP linearization and related problems[J]. *Discrete Optimization*, 2013, 10(3): 200-209.
- [140] ERDOĞAN G, TANSEL B. A branch-and-cut algorithm for quadratic assignment problems based on linearizations[J]. *Computers & Operations Research*, 2007, 34(4): 1085-1106.
- [141] DOKEROGLU T, COSAR A. A novel multistart hyper-heuristic algorithm on the grid for the quadratic assignment problem[J]. *Engineering Applications of Artificial Intelligence*, 2016, 52: 10-25.
- [142] WANG T, LING H, LANG C, et al. Graph matching with adaptive and branching path following[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(12): 2853-2867.
- [143] KUSHINSKY Y, MARON H, DYM N, et al. Sinkhorn Algorithm for Lifted Assignment Problems[J]., 2019, 12(2): 716-735.
- [144] NOWAK A, VILLAR S, BANDEIRA A, et al. Revised note on learning quadratic assignment with graph neural networks[C]//*Data Science Workshop*. 2018.
- [145] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. *IEEE Transactions on Neural Networks*, 2008, 20(1): 61-80.
- [146] ZHANG Z, LEE W S. Deep Graphical Feature Learning for the Feature Matching Problem[C]//*International Conference on Computer Vision*. 2019: 5087-5096.
- [147] FEY M, ERIC LENSSEN J, WEICHERT F, et al. SplineCNN: Fast geometric deep learning with continuous b-spline kernels[C]//*Computer Vision and Pattern Recognition*. 2018: 869-877.
- [148] ZHOU F, DE LA TORRE F. Factorized graph matching[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 38(9): 1774-1789.

- [149] MENA G, BELANGER D, LINDERMAN S, et al. Learning latent permutations with gumbel-sinkhorn networks[J]. International Conference on Learning and Representations, 2018.
- [150] BAI S, ZHANG F, TORR P H. Hypergraph convolution and hypergraph attention[J]. Pattern Recognition, 2021, 110: 107637.
- [151] FENG Y, YOU H, ZIZHAO Z, et al. Hypergraph Neural Networks[C]//AAAI: vol. 33: 01. 2019: 3558-3565.
- [152] ZHOU X, ZHU M, DANIILIDIS K. Multi-Image Matching via Fast Alternating Minimization[C]//International Conference on Computer Vision. 2015: 4032-4040.
- [153] MASET E, ARRIGONI F, FUSIELLO A. Practical and efficient multi-view matching[C]//International Conference on Computer Vision. 2017: 4568-4576.
- [154] IONESCU C, VANTZOS O, SMINCHISESCU C. Matrix backpropagation for deep networks with structured layers[C]//International Conference on Computer Vision. 2015: 2965-2973.
- [155] KINGMA D, BA J. Adam: A Method for Stochastic Optimization[J]. International Conference on Learning and Representations, 2014.
- [156] EGOZI A, KELLER Y, GUTERMAN H. A Probabilistic Approach to Spectral Graph Matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35(1): 18-27.
- [157] LIU Z Y, QIAO H, XU L. An Extended Path Following Algorithm for Graph-Matching Problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1451-1456.
- [158] BURKARD R E, KARISCH S E, RENDL F. QAPLIB – a quadratic assignment problem library[J]. Journal of Global optimization, 1997, 10(4): 391-403.
- [159] HAHN P M, KRARUP J. A hospital facility layout problem finally solved[J]. Journal of Intelligent Manufacturing, 2001, 12(5-6): 487-496.
- [160] JIANG Z, WANG T, YAN J. Unifying Offline and Online Multi-graph Matching via Finding Shortest Paths on Supergraph[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(10): 3648-3663.
- [161] LI Y, GU C, DULLIEN T, et al. Graph Matching Networks for Learning the Similarity of Graph Structured Objects[C]//International Conference on Machine Learning. 2019: 3835-3845.

- [162] BAI Y, DING H, BIAN S, et al. Simgnn: A neural network approach to fast graph similarity computation[C]//Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 2019: 384-392.
- [163] WILDER B, DILKINA B, TAMBE M. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization[C]//AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 1658-1665.
- [164] VILLANI C. Optimal transport: old and new[M]. Springer, 2009.
- [165] SINKHORN R. A relationship between arbitrary positive matrices and doubly stochastic matrices[J]. AoMS, 1964.
- [166] JANG E, GU S, POOLE B. Categorical Reparameterization with Gumbel-Softmax[C]//International Conference on Learning and Representations. 2017.
- [167] GROVER A, WANG E, ZWEIG A, et al. Stochastic Optimization of Sorting Networks via Continuous Relaxations[C]//International Conference on Learning and Representations. 2019.
- [168] XU Y, YAN Y, LIN Q, et al. Homotopy Smoothing for Non-Smooth Problems with Lower Complexity than $O(1/\epsilon)$ [J]. Neural Information Processing Systems, 2016.
- [169] XIAO L, ZHANG T. A proximal-gradient homotopy method for the sparse least-squares problem[J]. SIAM Journal on Optimization, 2013, 23(2): 1062-1091.
- [170] KHULLER S, MOSS A, NAOR J S. The budgeted maximum coverage problem[J]. Information processing letters, 1999, 70(1): 39-45.
- [171] HAMILTON W L, YING R, LESKOVEC J. Inductive Representation Learning on Large Graphs[J]. Neural Information Processing Systems, 2017.
- [172] BEASLEY J E. OR-Library: distributing test problems by electronic mail[J]. Journal of the operational research society, 1990, 41(11): 1069-1072.
- [173] ROZEMBERCZKI B, ALLEN C, SARKAR R. Multi-Scale Attributed Node Embedding[J]. Journal of Complex Networks, 2021, 9(2).
- [174] SOLIN M M, ALAMSYAH A, RIKUMAHU B, et al. Forecasting portfolio optimization using artificial neural network and genetic algorithm[C]//2019 7th International Conference on Information and Communication Technology (ICoICT). 2019: 1-7.
- [175] AMOS B, KOLTUN V, KOLTER J Z. The Limited Multi-Label Projection Layer[J]. ArXiv preprint, 2019.

- [176] SHARPE W F. The sharpe ratio[J]. *Streetwise—the Best of the Journal of Portfolio Management*, 1998: 169-185.
- [177] MAERGNER P, PONDENKANDATH V, ALBERTI M, et al. Combining graph edit distance and triplet networks for offline signature verification[J]. *Pattern Recognition Letters*, 2019, 125: 527-533.
- [178] CHEN L, LIN G, WANG S, et al. Graph Edit Distance Reward: Learning to Edit Scene Graph[J]. *European Conference on Computer Vision*, 2020.
- [179] RIESEN K, BUNKE H. IAM graph database repository for graph based pattern recognition and machine learning[C]//*Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. 2008: 287-297.
- [180] ZEYEN C, BERGMANN R. A*-based similarity assessment of semantic graphs[C]//*International Conference on Case-Based Reasoning*. 2020: 17-32.
- [181] ABU-AISHEH Z, RAVEAUX R, RAMEL J Y, et al. An exact graph edit distance algorithm for solving pattern recognition problems[C]//*International Conference on Pattern Recognition*. 2015.
- [182] BAI Y, DING H, GU K, et al. Learning-Based Efficient Graph Similarity Computation via Multi-Scale Convolutional Set Matching.[C]//*AAAI Conference on Artificial Intelligence*. 2020: 3219-3226.
- [183] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. *Trans. on Neural Networks*, 2009.
- [184] RIESEN K, BUNKE H. Approximate graph edit distance computation by means of bipartite graph matching[J]. *Image and Vision Computing*, 2009, 27(7): 950-959.
- [185] FANKHAUSER S, RIESEN K, BUNKE H. Speeding up graph edit distance computation through fast bipartite matching[C]//*International Workshop on Graph-Based Representations in Pattern Recognition*. 2011: 102-111.
- [186] ZENG Z, TUNG A K, WANG J, et al. Comparing stars: On approximating graph edit distance[J]. *Proceedings of the VLDB Endowment*, 2009, 2(1): 25-36.
- [187] LEROUGE J, ABU-AISHEH Z, RAVEAUX R, et al. New binary linear programming formulation to compute the graph edit distance[J]. *Pattern Recognition*, 2017, 72: 254-265.

- [188] ABU-AISHEH Z, GAÜZÈRE B, BOUGLEUX S, et al. Graph edit distance contest: Results and future challenges[J]. Pattern Recognition Letters, 2017, 100: 96-103.
- [189] BLUMENTHAL D B, BORIA N, GAMPER J, et al. Comparing heuristics for graph edit distance computation[J]. The VLDB journal, 2020, 29(1): 419-458.
- [190] KUHN H W. The Hungarian method for the assignment problem[C]//Export. Naval Research Logistics Quarterly. 1955: 83-97.
- [191] JONKER R, VOLGENANT A. A shortest augmenting path algorithm for dense and sparse linear assignment problems[J]. Computing, 1987, 38(4): 325-340.
- [192] KASPAR R. Graph Matching Toolkit[Z]. <https://github.com/dzambon/graph-matching-toolkit>. 2017.
- [193] JIANG B, TANG J, DING C, et al. Graph matching via multiplicative update algorithm[C]//Neural Information Processing Systems. 2017: 3187-3195.
- [194] ZHOU F, DE LA TORRE F. Factorized Graph Matching[C]//Computer Vision and Pattern Recognition. 2012: 127-134.
- [195] ZHOU J, CUI G, ZHANG Z, et al. Graph Neural Networks: A Review of Methods and Applications[J]. ArXiv:1812.08434, 2018.
- [196] DAI X, YAN X, ZHOU K, et al. Edit Distance Embedding using Convolutional Neural Networks[C]//International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.
- [197] CHANG L, FENG X, LIN X, et al. Speeding Up GED Verification for Graph Similarity Search[C]//International Conference on Data Engineering. 2020.
- [198] PAREJA A, DOMENICONI G, CHEN J, et al. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs[J]. AAAI Conference on Artificial Intelligence, 2020, 34(4): 5363-5370.
- [199] MANESSI F, ROZZA A, MANZO M. Dynamic graph convolutional networks[J]. Pattern Recognition, 2020, 97: 107000.
- [200] ZHENG L, LI Z, LI J, et al. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.[C]//International Joint Conference on Artificial Intelligence. 2019: 4419-4425.
- [201] SOCHER R, CHEN D, MANNING C D, et al. Reasoning with Neural Tensor Networks for Knowledge Base Completion[C]//Neural Information Processing Systems. Lake Tahoe, Nevada: Curran Associates Inc., 2013: 926-934.

- [202] BOUGLEUX S, GAÜZÈRE B, BRUN L. A Hungarian Algorithm for Error-Correcting Graph Matching[C]//FOGGIA P, LIU C L, VENTO M. Graph-Based Representations in Pattern Recognition. Cham: Springer International Publishing, 2017: 118-127.
- [203] WANG X, DING X, TUNG K H A, et al. An Efficient Graph Indexing Method[J]. ICDE, 2012: 210-221.
- [204] FEY M, LENSSEN J E. Fast Graph Representation Learning with PyTorch Geometric[C]//ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019.
- [205] BEHNEL S, BRADSHAW R, CITRO C, et al. Cython: The Best of Both Worlds[J]. Computing in Science Engineering, 2011, 13(2): 31-39.
- [206] WANG T, JIANG Z, YAN J. Clustering-aware Multiple Graph Matching via Decayed Pairwise Matching Composition[J]. AAAI Conference on Artificial Intelligence, 2020: 7-12.
- [207] LOWE G D. Object Recognition from Local Scale-Invariant Features[J]. International Conference on Computer Vision, 1999: 1150-1150.
- [208] CHEN T, KORNBLITH S, NOROUZI M, et al. A Simple Framework for Contrastive Learning of Visual Representations[C]//International Conference on Machine Learning. 2020: 1597-1607.
- [209] GRILL J B, STRUB F, ALTCHÉ F, et al. Bootstrap your own latent: A new approach to self-supervised learning[C]//Neural Information Processing Systems. 2020.
- [210] BROMLEY J, GUYON I, LECUN Y, et al. Signature verification using a “Siamese” time delay neural network[C]//Neural Information Processing Systems. 1994.
- [211] CHEN X, HE K. Exploring Simple Siamese Representation Learning[C]//Computer Vision and Pattern Recognition. 2021: 15750-15758.
- [212] DYM N, MARON H, LIPMAN Y. DS++: A Flexible, Scalable and Provably Tight Relaxation for Matching Problems[J]. ACM Trans. Graph., 2017, 36(6): 1-14.
- [213] SOLÉ-RIBALTA A, SERRATOSA F. Graduated Assignment Algorithm for Finding the Common Labelling of a Set of Graphs[C]//Structural, Syntactic, and Statistical Pattern Recognition. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 180-190.

- [214] ARTHUR D, VASSILVITSKII S. K-Means++: The Advantages of Careful Seeding[C]//ACM-SIAM Symposium on Discrete Algorithms. 2007: 1027-1035.
- [215] RANGARAJAN A, YUILLE A, MJOLSNESS E. Convergence properties of the softassign quadratic assignment algorithm[J]. *Neural Computation*, 1999, 11(6): 1455-1474.
- [216] MANNING C D, RAGHAVAN P, SCHUTZE H. *Introduction to Information Retrieval*[M]. Cambridge University Press, 2005.
- [217] RAND W M. Objective Criteria for the Evaluation of Clustering Methods[J]. *Journal of the American Statistical Association*, 1971, 66(336): 846-850.
- [218] WAH C, BRANSON S, WELINDER P, et al. The Caltech-UCSD Birds-200-2011 Dataset[R]. CNS-TR-2011-001. California Institute of Technology, 2011.
- [219] WANG R, YAN J, YANG X. Learning Combinatorial Embedding Networks for Deep Graph Matching[C]//International Conference on Computer Vision. 2019: 3056-3065.
- [220] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C] //Computer Vision and Pattern Recognition. 2016: 770-778.
- [221] HU R, XU J, CHEN B, et al. TAP-Net: transport-and-pack using reinforcement learning[J]. *ACM Transactions on Graphics (TOG)*, 2020, 39(6): 1-15.
- [222] MAO H, SCHWARZKOPF M, VENKATAKRISHNAN S B, et al. Learning scheduling algorithms for data processing clusters[C]//Proceedings of the ACM Special Interest Group on Data Communication. 2019: 270-288.
- [223] ZHANG C, SONG W, CAO Z, et al. Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning[J]. *Neural Information Processing Systems*, 2020, 33.
- [224] SUTTON R S, BARTO A G. *Reinforcement learning: An introduction*[M]. MIT press, 2018.
- [225] DUAN L, HU H, QIAN Y, et al. A Multi-task Selected Learning Approach for Solving 3D Flexible Bin Packing Problem[C]//Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019: 1386-1394.
- [226] HUTTER F, HOOS H H, LEYTON-BROWN K. Automated Configuration of Mixed Integer Programming Solvers[C]//Proceedings of the Conference on Integration of Artificial Intelligence and Operations Research techniques in Constraint Programming (CPAIOR). 2010: 186-202.

- [227] SUN H, CHEN W, LI H, et al. Improving Learning to Branch via Reinforcement Learning[J]. NeurIPS Workshop, 2020.
- [228] XU L, HUTTER F, HOOS H H, et al. Hydra-MIP: Automated Algorithm Configuration and Selection for Mixed Integer Programming[C]//IJCAI Workshop. 2011.
- [229] GOMORY R E. Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem[G]//50 Years of Integer Programming 1958-2008. Springer, 2010: 77-103.
- [230] TANG Y, AGRAWAL S, FAENZA Y. Reinforcement learning for integer programming: Learning to cut[C]//International Conference on Machine Learning. 2020: 9367-9376.
- [231] TALBI E G. A Taxonomy of Metaheuristics for Bi-level Optimization[M]//Metaheuristics for Bi-level Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 1-39.
- [232] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C]//International Conference on Machine Learning. 2015: 1889-1897.
- [233] FU Z H, QIU K B, ZHA H. Generalize a Small Pre-trained Model to Arbitrarily Large TSP Instances[J]. AAAI Conference on Artificial Intelligence, 2021.
- [234] FORTI A. DAG Scheduling for Grid Computing Systems[D]. University of Udine, Italy, 2006.
- [235] BLUMENTHAL D B, BORIA N, GAMPER J, et al. Comparing heuristics for graph edit distance computation[J]. The VLDB journal, 2020, 29(1): 419-458.
- [236] EULER L. Solutio problematis ad geometriam situs pertinentis[J]. Commentarii academiae scientiarum Petropolitanae, 1741: 128-140.
- [237] GAREY M R, JOHNSON D S. Computers and Intractability; A Guide to the Theory of NP-Completeness[M]. 1990.
- [238] JOSHI C K, LAURENT T, BRESSON X. An efficient graph convolutional network technique for the travelling salesman problem[J]. ArXiv preprint, 2019.
- [239] REINELT G. The traveling salesman: computational solutions for TSP applications[M]. Springer, 2003.
- [240] HAYTHORPE M. FHCP Challenge Set: The first set of structurally difficult instances of the Hamiltonian cycle problem[J]. ArXiv preprint, 2019.

- [241] LUO S, SHI C, XU M, et al. Predicting molecular conformation via dynamic graph score matching[J]. *Neural Information Processing Systems*, 2021, 34: 19784-19795.
- [242] GUO X, BAL A B, NEEDHAM T, et al. Statistical shape analysis of brain arterial networks (BAN)[J]. *The Annals of Applied Statistics*, 2022, 16(2): 1130-1150.
- [243] FISCHLER M A, BOLLES R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. *Communications of the ACM*, 1981, 24(6): 381-395.
- [244] QIN T, TU S, XU L. IA-NGM: A bidirectional learning method for neural graph matching with feature fusion[J]. *Machine Learning*, 2022: 1-27.
- [245] PASZKE A, GROSS S, CHINTALA S, et al. Automatic differentiation in pytorch[J]., 2017.
- [246] HARRIS C R, MILLMAN K J, van der WALT S J, et al. Array programming with NumPy[J]. *Nature*, 2020, 585(7825): 357-362.
- [247] SWOBODA P, ROTHER C, ALHAIJA H, et al. A study of Lagrangean decompositions and dual ascent solvers for graph matching[C]//*Computer Vision and Pattern Recognition*. 2017.
- [248] GAO Q, WANG F, XUE N, et al. Deep Graph Matching Under Quadratic Constraint[C]//*Computer Vision and Pattern Recognition*. 2021: 5069-5078.
- [249] JIANG Z, RAHMANI H, ANGELOV P, et al. Graph-Context Attention Networks for Size-Variied Deep Graph Matching[C]//*Computer Vision and Pattern Recognition*. 2022: 2343-2352.
- [250] LIU C, ZHANG S, YANG X, et al. Self-supervised learning of visual graph matching[C]//*European Conference on Computer Vision*. 2022: 370-388.
- [251] NAIR V, BARTUNOV S, GIMENO F, et al. Solving mixed integer programs using neural networks[J]. *ArXiv preprint*, 2020.

附录 A 3.4节定理的详细证明

在正式给出3.4节定理的详细证明之前，首先再次强调3.4节给出的如下定义：

- $\mathbf{T}^* = \text{TopK}(\mathbf{D})$ 是公式 (3.48) 线性整数规划形式的 OT 问题的最优解。它等价于首先将所有元素排序，随后选取其中前 k 个。如果第 k 个和第 $(k+1)$ 个元素相等，算法会随机地选择其中一个，以严格满足基数约束；
- $\mathbf{T}^{\tau*} = \text{Sinkhorn}(\mathbf{D})$ 是公式 (3.49) 熵正则化后的 OT 问题的最优解，由 Sinkhorn 算法求解得到。它也是 CardNN-S 算法的输出；
- $\mathbf{T}^{\sigma*} = \text{TopK}(\tilde{\mathbf{D}})$ 是公式 (3.54) 中，受 Gumbel 噪声扰动后的线性整数规划形式的 OT 问题的最优解。它等价于首先添加 Gumbel 噪声，随后将所有扰动后的元素排序，并选取其中前 k 个。如果扰动后的第 k 个和第 $(k+1)$ 个元素相等，算法会随机地选择其中一个，以严格满足基数约束；
- $\tilde{\mathbf{T}}^* = \text{Sinkhorn}(\tilde{\mathbf{D}})$ 是公式 (3.55) 中，受 Gumbel 噪声扰动后，添加熵正则化项的 OT 问题的最优解，由 Sinkhorn 算法求解得到。它也是 CardNN-GS 算法的输出。

A.1 命题 3.4的证明

首先介绍一个引理，用于证明命题 3.4:

引理 A.1. 已知实数 ϕ_i, ϕ_j , 且 u_i, u_j 来自独立同分布的均匀分布 $\mathcal{U}(0,1)$ 。在添加 Gumbel 噪声后， $\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j)$ 的概率是：

$$P(\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j)) = \frac{1}{1 + \exp\left(-\frac{\phi_i - \phi_j}{\sigma}\right)}. \quad (\text{A.1})$$

证明. 因为 $g_\sigma(u_i) = -\sigma \log(-\log(u_i))$, $P(\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j))$ 等于如下不等式成立的概率:

$$\phi_i - \sigma \log(-\log(u_i)) > \phi_j - \sigma \log(-\log(u_j)) \quad (\text{A.2})$$

我们有

$$\phi_i - \phi_j > \sigma \log(-\log(u_i)) - \sigma \log(-\log(u_j)) \quad (\text{A.3a})$$

$$\frac{\phi_i - \phi_j}{\sigma} > \log\left(\frac{\log(u_i)}{\log(u_j)}\right) \quad (\text{A.3b})$$

$$e^{\frac{\phi_i - \phi_j}{\sigma}} > \frac{\log(u_i)}{\log(u_j)} \quad (\text{A.3c})$$

由于 $u_j \in (0, 1)$, $\log(u_j) < 0$, 则

$$\log(u_j) < \log(u_i) e^{-\frac{\phi_i - \phi_j}{\sigma}} \quad (\text{A.4a})$$

$$\log(u_j) < \log\left(u_i^{\exp-\frac{\phi_i - \phi_j}{\sigma}}\right) \quad (\text{A.4b})$$

$$u_j < u_i^{\exp-\frac{\phi_i - \phi_j}{\sigma}} \quad (\text{A.4c})$$

因为 u_i, u_j 来自独立同分布的均匀分布, 上式成立的概率是

$$\int_0^1 \int_0^{u_i^{\exp-\frac{\phi_i - \phi_j}{\sigma}}} du_j du_i = \int_0^1 u_i^{\exp-\frac{\phi_i - \phi_j}{\sigma}} du_i = \frac{1}{1 + \exp-\frac{\phi_i - \phi_j}{\sigma}} \quad (\text{A.5})$$

因此, 公式 $\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j)$ 成立的概率是

$$P(\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j)) = \frac{1}{1 + \exp-\frac{\phi_i - \phi_j}{\sigma}} \quad (\text{A.6})$$

□

接下来给出命题 3.4 的详细证明过程:

命题 3.4 的证明. 再次强调如下定义: $\Phi = [\phi_1, \phi_2, \phi_3, \dots, \phi_m]$ 是 \mathbf{s} 的降序排列后的版本. 对它采用独立同分布的 Gumbel 噪声进行扰动, 得到

$$\tilde{\Phi} = [\phi_1 + g_\sigma(u_1), \phi_2 + g_\sigma(u_2), \phi_3 + g_\sigma(u_3), \dots, \phi_m + g_\sigma(u_m)] \quad (\text{A.7})$$

其中 $g_\sigma(u) = -\sigma \log(-\log(u))$ 是 Gumbel 噪声, 其程度和噪声参数 σ 有关, 且 $u_1, u_2, u_3, \dots, u_m$ 是独立同分布的均匀分布. 定义排列 π 是 $\tilde{\Phi}$ 的降序排列, 即 $\phi_{\pi_1} + g_\sigma(u_{\pi_1}), \phi_{\pi_2} + g_\sigma(u_{\pi_2}), \phi_{\pi_3} + g_\sigma(u_{\pi_3}), \dots, \phi_{\pi_m} + g_\sigma(u_{\pi_m})$ 是降序的.

根据命题 3.3, 对于 $\phi_1, \phi_2, \phi_3, \dots, \phi_m$, 有

$$\|\mathbf{T}^* - \mathbf{T}^{\tau*}\|_F \leq \frac{2m\tau \log 2}{|\phi_k - \phi_{k+1}|} \quad (\text{A.8})$$

通过将 Φ 替换为 $\tilde{\Phi}$, 并计算对 u 的期望, 可以得到

$$\mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq \mathbb{E}_u \left[\frac{2m\tau \log 2}{|\phi_{\pi_k} + g_\sigma(u_{\pi_k}) - \phi_{\pi_{k+1}} - g_\sigma(u_{\pi_{k+1}})|} \right] \quad (\text{A.9})$$

基于引理 A.1, 出现 $\pi_k = i, \pi_{k+1} = j$ 的概率是

$$P(\pi_k = i, \pi_{k+1} = j) = \frac{1}{1 + \exp - \frac{\phi_i - \phi_j}{\sigma}} \sum_{\forall \pi} \left(\prod_{a=1}^{k-1} \frac{1}{1 + \exp - \frac{\phi_{\pi_a} - \phi_i}{\sigma}} \prod_{b=k+2}^m \frac{1}{1 + \exp - \frac{\phi_j - \phi_{\pi_b}}{\sigma}} \right) \quad (\text{A.10})$$

其中第一项代表 $\phi_i + g_\sigma(u_i) > \phi_j + g_\sigma(u_j)$, 第二项代表存在 $(k-1)$ 个元素大于 $\phi_i + g_\sigma(u_i)$ 且剩余的元素小于 $\phi_j + g_\sigma(u_j)$ 的概率。

接下来, 推导 $\mathbb{E}_u \left[\frac{1}{|\phi_{\pi_k} + g_\sigma(u_{\pi_k}) - \phi_{\pi_{k+1}} - g_\sigma(u_{\pi_{k+1}})|} \right]$ 的上界。定义 $\mathcal{A}_{i,j}$ 如下:

$$u_i, u_j \in \mathcal{A}_{i,j}, \quad s.t. \quad \phi_i + g_\sigma(u_i) - \phi_j - g_\sigma(u_j) > \varepsilon \quad (\text{A.11})$$

其中 ε 是一个足够小的数。我们有

$$\begin{aligned} & \mathbb{E}_u \left[\frac{1}{|\phi_{\pi_k} + g_\sigma(u_{\pi_k}) - \phi_{\pi_{k+1}} - g_\sigma(u_{\pi_{k+1}})|} \right] \\ &= \sum_{i \neq j} P(\pi_k = i, \pi_{k+1} = j) \mathbb{E}_{u_i, u_j \in \mathcal{A}_{i,j}} \left[\frac{1}{|\phi_i + g_\sigma(u_i) - \phi_j - g_\sigma(u_j)|} \right] \end{aligned} \quad (\text{A.12a})$$

$$\begin{aligned} &= \sum_{i \neq j} \left(\frac{1}{1 + \exp - \frac{\phi_i - \phi_j}{\sigma}} \sum_{\forall \pi} \left(\prod_{a=1}^{k-1} \frac{1}{1 + \exp - \frac{\phi_{\pi_a} - \phi_i}{\sigma}} \prod_{b=k+2}^m \frac{1}{1 + \exp - \frac{\phi_j - \phi_{\pi_b}}{\sigma}} \right) \right. \\ & \quad \left. \mathbb{E}_{u_i, u_j \in \mathcal{A}_{i,j}} \left[\frac{1}{|\phi_i + g_\sigma(u_i) - \phi_j - g_\sigma(u_j)|} \right] \right) \end{aligned} \quad (\text{A.12b})$$

$$\begin{aligned} &= \sum_{i \neq j} \left(\frac{1}{1 + \exp - \frac{\phi_i - \phi_j}{\sigma}} \sum_{\forall \pi} \left(\prod_{a=1}^{k-1} \frac{1}{1 + \exp - \frac{\phi_{\pi_a} - \phi_i}{\sigma}} \prod_{b=k+2}^m \frac{1}{1 + \exp - \frac{\phi_j - \phi_{\pi_b}}{\sigma}} \right) \right. \\ & \quad \left. \mathbb{E}_{u_i, u_j \in \mathcal{A}_{i,j}} \left[\frac{1}{|\phi_i - \sigma \log(-\log(u_i)) - \phi_j + \sigma \log(-\log(u_j))|} \right] \right) \end{aligned} \quad (\text{A.12c})$$

$$= \sum_{i \neq j} \left(f(\phi_i - \phi_j, \sigma, z) \sum_{\forall \pi} \left(\prod_{a=1}^{k-1} \frac{1}{1 + \exp - \frac{\phi_{\pi_a} - \phi_i}{\sigma}} \prod_{b=k+2}^m \frac{1}{1 + \exp - \frac{\phi_j - \phi_{\pi_b}}{\sigma}} \right) \right) \quad (\text{A.12d})$$

定义 $f(\delta, \sigma, z)$ 如下:

$$f(\delta, \sigma, z) = \frac{1}{1 + \exp - \frac{\delta}{\sigma}} \mathbb{E}_{u_i, u_j} \left[\frac{1}{|\delta - \sigma \log(-\log(u_i)) + \sigma \log(-\log(u_j))|} \right] \quad (\text{A.13})$$

$$s.t. \quad \delta - \sigma \log(-\log(u_i)) + \sigma \log(-\log(u_j)) > z > 0$$

对于公式 (A.12d) 中的概率项, 对任意的排列 π , 一定存在 π_a, π_b , 使得

$$\frac{1}{1 + \exp - \frac{\phi_{\pi_a} - \phi_i}{\sigma}} \leq \frac{1}{1 + \exp - \frac{\phi_k - \phi_i}{\sigma}} \quad (\text{A.14a})$$

$$\frac{1}{1 + \exp - \frac{\phi_j - \phi_{\pi_b}}{\sigma}} \leq \frac{1}{1 + \exp - \frac{\phi_j - \phi_{k+1}}{\sigma}} \quad (\text{A.14b})$$

因此,

$$\text{公式 (A.12d)} \leq \sum_{i \neq j} \left(f(\phi_i - \phi_j, \sigma, z) \frac{1}{1 + \exp - \frac{\phi_k - \phi_i}{\sigma}} \frac{1}{1 + \exp - \frac{\phi_j - \phi_{k+1}}{\sigma}} \right) \quad (\text{A.15a})$$

$$\leq \sum_{i \neq j} \frac{f(\phi_i - \phi_j, \sigma, z)}{(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} \quad (\text{A.15b})$$

基于引理A.1中的公式 (A.1), 将 $\phi_j - \phi_i$ 替换为 y , 得到

$$\text{公式 (A.1)} \Rightarrow P(g_\sigma(u_i) - g_\sigma(u_j) > \phi_j - \phi_i) = \frac{1}{1 + \exp - \frac{\phi_i - \phi_j}{\sigma}} \quad (\text{A.16a})$$

$$\Rightarrow P(g_\sigma(u_i) - g_\sigma(u_j) > y) = \frac{1}{1 + \exp \frac{y}{\sigma}} \quad (\text{A.16b})$$

$$\Rightarrow P(g_\sigma(u_i) - g_\sigma(u_j) < y) = 1 - \frac{1}{1 + \exp \frac{y}{\sigma}} = \frac{1}{1 + \exp - \frac{y}{\sigma}} \quad (\text{A.16c})$$

其中等式右边正是标准 Logistic 分布的累积分布函数 (如果 $\sigma = 1$):

$$\text{CDF}(y) = \frac{1}{1 + \exp(-y)} \quad (\text{A.17})$$

因此 $-\log(-\log(u_i)) + \log(-\log(u_j))$ 等价于 Logistic 分布, 它的概率密度函数是

$$\text{PDF}(y) = \frac{d\text{CDF}(y)}{dy} = \frac{1}{\exp(-y) + \exp y + 2} \quad (\text{A.18})$$

本证明采用了 PDF(y) 的一个上界:

$$\text{PDF}(y) = \frac{1}{\exp(-y) + \exp y + 2} \leq \frac{1}{y^2 + 4} \quad (\text{A.19})$$

基于 Logistic 分布, $-\sigma \log(-\log(u_i)) + \sigma \log(-\log(u_j))$ 可被替换为 σy , 其中 y 采样自 Logistic 分布。因此, $f(\delta, \sigma, z)$ 的上界推导如下

$$f(\delta, \sigma, z) = \frac{1}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{\int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{\exp(-y) + \exp y + 2} dy}{\int_{-\delta/\sigma+z}^{\infty} \frac{1}{\exp(-y) + \exp y + 2} dy} \quad (\text{A.20a})$$

$$= \frac{1}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{\int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{\exp(-y) + \exp y + 2} dy}{1 - \frac{1}{1 + \exp(\delta/\sigma - z)}} \quad (\text{A.20b})$$

$$= \frac{1}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{\int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{\exp(-y) + \exp y + 2} dy}{\frac{\exp(\delta/\sigma - z)}{1 + \exp(\delta/\sigma - z)}} \quad (\text{A.20c})$$

$$= \frac{1}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{\int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{\exp(-y) + \exp y + 2} dy}{\frac{1}{1 + \exp(-\delta/\sigma + z)}} \quad (\text{A.20d})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{\exp(-y) + \exp y + 2} dy \quad (\text{A.20e})$$

$$\leq \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \int_{-\delta/\sigma+z}^{\infty} \frac{1}{\delta + \sigma y} \frac{1}{y^2 + 4} dy \quad (\text{A.20f})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{2\sigma \log((z\sigma - \delta)^2 + 4\sigma^2) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) - 4\sigma \log z + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20g})$$

$$\leq \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{2\sigma \log((z\sigma + |\delta|)^2 + 4\sigma^2) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) - 4\sigma \log z + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20h})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{2\sigma \log((z\sigma + |\delta|)^2 + 4\sigma^2) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) - 2\sigma \log z^2 + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20i})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{2\sigma \log\left(\frac{(z\sigma + |\delta|)^2 + 4\sigma^2}{z^2}\right) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20j})$$

$$\leq \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{2\sigma \log\left(\frac{(z\sigma + |\delta| + 2\sigma)^2}{z^2}\right) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20k})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) - 2\delta \arctan\left(\frac{z - \delta/\sigma}{2}\right) + \pi\delta}{4\delta^2 + 16\sigma^2} \quad (\text{A.20l})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp - \frac{\delta}{\sigma}} \cdot \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) + \delta \left(\pi - 2 \arctan\left(\frac{z - \delta/\sigma}{2}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.20m})$$

$$\leq \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp-\frac{\delta}{\sigma}} \cdot \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) + |\delta| \left(\pi - 2 \arctan\left(\frac{z - \delta/\sigma}{2}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.20n})$$

$$\leq \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp-\frac{\delta}{\sigma}} \cdot \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) + |\delta| \left(\pi - 2 \arctan\left(-\frac{\delta}{2\sigma}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.20o})$$

$$= \frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp-\frac{\delta}{\sigma}} \cdot \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) + |\delta| \left(\pi + 2 \arctan\left(\frac{\delta}{2\sigma}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.20p})$$

其中公式 (A.20f) 成立的原因是 $\frac{1}{\exp(-y) + \exp y + 2} \leq \frac{1}{y^2 + 4}$, 公式 (A.20n) 成立的原因是 $\pi - 2 \arctan(\frac{z - \delta/\sigma}{2}) \geq 0$ 。在至少 $(1 - \varepsilon)$ 的概率下, 有

$$z = \log \frac{1 + \varepsilon \exp \frac{\delta}{\sigma}}{1 - \varepsilon} \geq -\log(1 - \varepsilon) \quad (\text{A.21a})$$

$$\frac{1 + \exp(-\frac{\delta}{\sigma} + z)}{1 + \exp-\frac{\delta}{\sigma}} = \frac{1}{1 - \varepsilon} \quad (\text{A.21b})$$

因此,

$$f(\delta, \sigma, z) \leq \text{公式 (A.20p)} = \frac{1}{1 - \varepsilon} \frac{4\sigma \log\left(\frac{z\sigma + |\delta| + 2\sigma}{z}\right) + |\delta| \left(\pi + 2 \arctan\left(\frac{\delta}{2\sigma}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.22a})$$

$$\leq \frac{1}{1 - \varepsilon} \frac{4\sigma \log\left(\sigma - \frac{|\delta| + 2\sigma}{\log(1 - \varepsilon)}\right) + |\delta| \left(\pi + 2 \arctan\left(\frac{\delta}{2\sigma}\right)\right)}{4\delta^2 + 16\sigma^2} \quad (\text{A.22b})$$

进一步推导得到

$$\text{公式 (A.15b)} \leq \sum_{i \neq j} \left(\frac{4\sigma \log\left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1 - \varepsilon)}\right) + |\phi_i - \phi_j| \left(\pi + 2 \arctan\left(\frac{\phi_i - \phi_j}{2\sigma}\right)\right)}{(1 - \varepsilon)(4(\phi_i - \phi_j)^2 + 16\sigma^2)(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} \right) \quad (\text{A.23})$$

综上所述, 在至少 $(1 - \varepsilon)$ 的概率下, 存在

$$\mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq \sum_{i \neq j} \frac{(2 \log 2) m \tau \left(4\sigma \log\left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1 - \varepsilon)}\right) + |\phi_i - \phi_j| \left(\pi + 2 \arctan \frac{\phi_i - \phi_j}{2\sigma}\right) \right)}{(1 - \varepsilon)(4(\phi_i - \phi_j)^2 + 16\sigma^2)(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} \quad (\text{A.24a})$$

$$= \sum_{i \neq j} \frac{(\log 2)m\tau \left(2\sigma \log \left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1-\varepsilon)} \right) + |\phi_i - \phi_j| \left(\frac{\pi}{2} + \arctan \frac{\phi_i - \phi_j}{2\sigma} \right) \right)}{(1-\varepsilon)((\phi_i - \phi_j)^2 + 4\sigma^2)(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} \quad (\text{A.24b})$$

$$= (\log 2)m\tau \sum_{i \neq j} \Omega(\phi_i, \phi_j, \sigma, \varepsilon) \quad (\text{A.24c})$$

其中 $\Omega(\phi_i, \phi_j, \sigma, \varepsilon)$ 代表了

$$\Omega(\phi_i, \phi_j, \sigma, \varepsilon) = \frac{2\sigma \log \left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1-\varepsilon)} \right) + |\phi_i - \phi_j| \left(\frac{\pi}{2} + \arctan \frac{\phi_i - \phi_j}{2\sigma} \right)}{(1-\varepsilon)((\phi_i - \phi_j)^2 + 4\sigma^2)(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} \quad (\text{A.25a})$$

□

A.2 推论 3.5 的证明

推论 3.5 是命题 3.4 的简化版本，仅考虑了其中占支配地位的项。

证明. 对于命题 3.4 中的 $\Omega(\phi_i, \phi_j, \sigma, \varepsilon)$ ，有

$$\Omega(\phi_i, \phi_j, \sigma, \varepsilon) \leq \frac{2\sigma \log \left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1-\varepsilon)} \right) + |\phi_i - \phi_j| \left(\frac{\pi}{2} + \arctan \frac{\phi_i - \phi_j}{2\sigma} \right)}{(1-\varepsilon)((\phi_i - \phi_j)^2 + 4\sigma^2)} \quad (\text{A.26a})$$

$$\leq \frac{2\sigma \log \left(\sigma - \frac{|\phi_i - \phi_j| + 2\sigma}{\log(1-\varepsilon)} \right) + |\phi_i - \phi_j| \pi}{(1-\varepsilon)((\phi_i - \phi_j)^2 + 4\sigma^2)} \quad (\text{A.26b})$$

$$= O \left(\frac{\sigma \log(\sigma + |\phi_i - \phi_j|) + |\phi_i - \phi_j|}{(\phi_i - \phi_j)^2 + \sigma^2} \right) \quad (\text{A.26c})$$

$$= \tilde{O} \left(\frac{\sigma + |\phi_i - \phi_j|}{(\phi_i - \phi_j)^2 + \sigma^2} \right) \quad (\text{A.26d})$$

其中， $(1-\varepsilon)$ 被视为常数（假设高概率成立）， $\tilde{O}(\cdot)$ 代表忽略所有对数项。

那么，

$$\mathbb{E}_u \left[\|\mathbf{T}^{\sigma^*} - \tilde{\mathbf{T}}^*\|_F \right] \leq (\log 2)m\tau \sum_{i \neq j} \tilde{O} \left(\frac{\sigma + |\phi_i - \phi_j|}{(\phi_i - \phi_j)^2 + \sigma^2} \right) \quad (\text{A.27a})$$

$$= (\log 2)m\tau \tilde{O} \left(\frac{\sigma + |\phi_i - \phi_j|}{(\phi_i - \phi_j)^2 + \sigma^2} \right)_{\forall i \neq j} \quad (\text{A.27b})$$

$$= \tilde{O} \left(\frac{m\tau(\sigma + |\phi_i - \phi_j|)}{(\phi_i - \phi_j)^2 + \sigma^2} \right)_{\forall i \neq j} \quad (\text{A.27c})$$

□

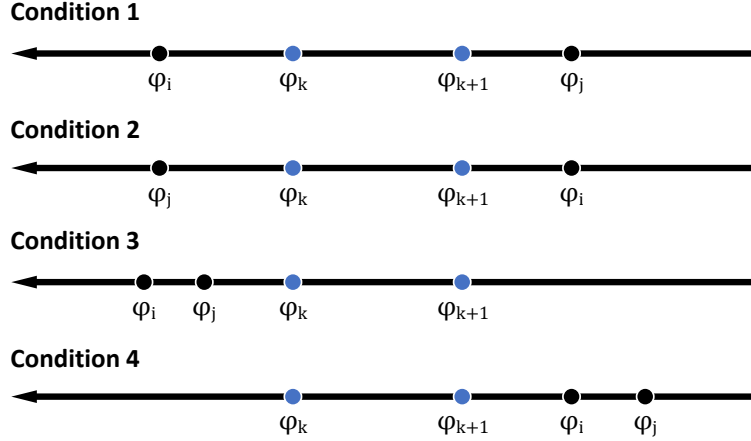


图 A-1 推论 3.6 证明中的四种情况。
Figure A-1 Four conditions to prove Corollary 3.6.

A.3 推论 3.6 的证明和注记

接下来给出推论 3.6 的证明, 并讨论 Sinkhorn 和 Gumbel-Sinkhorn 的联系: 当 $\sigma \rightarrow 0^+$, Sinkhorn 方法 (CardNN-S) 成为了 Gumbel-Sinkhorn 方法 (CardNN-GS) 特殊情况。为了更严格地表述推论 3.6, 此处给出了如下命题

命题 A.2. 假设 ϕ_k, ϕ_{k+1} 的值是唯一的²⁶, 在至少 $(1 - \varepsilon)$ 的概率下, 下式成立

$$\lim_{\sigma \rightarrow 0^+} \mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq \frac{(\pi \log 2)m\tau}{(1 - \varepsilon)|\phi_k - \phi_{k+1}|} \quad (\text{A.28})$$

上式与命题 3.3 的结论仅仅相差了一个常数项。

证明. 因为 $\sigma \rightarrow 0^+$, $\Omega(\phi_i, \phi_j, \sigma, \varepsilon)$ 分子的第一项变成了 0。分子的第二项分四种情况讨论, 如图 A-1 所示, 其中不包括 $\phi_i = \phi_k$ 且 $\phi_j = \phi_{k+1}$ (通过假设 ϕ_k, ϕ_{k+1} 的唯一性剔除)。此外, ϕ_k, ϕ_{k+1} 之间不可能存在其他项, 因为它们是排序后的相邻两项。

情况 1. 如果 $\phi_i \geq \phi_k, \phi_j \leq \phi_{k+1}$ (等号不同时成立), 一定存在 $\phi_i - \phi_k > 0$ 或 $\phi_{k+1} - \phi_j > 0$ 。因此,

$$\lim_{\sigma \rightarrow 0^+} \frac{1}{(1 + \exp \frac{\phi_i - \phi_k}{\sigma})(1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma})} = 0 \quad (\text{A.29a})$$

$$\Rightarrow \lim_{\sigma \rightarrow 0^+} \Omega(\phi_i, \phi_j, \sigma, \varepsilon) = 0 \quad (\text{A.29b})$$

²⁶为了证明的简洁性, 此处假设了 ϕ_k, ϕ_{k+1} 的值是唯一的。如果存在重复的 ϕ_k, ϕ_{k+1} , 此处的上界仅会相差一个常数项, 因此不影响结论: 当 $\sigma \rightarrow 0^+$, Sinkhorn 方法 (CardNN-S) 是 Gumbel-Sinkhorn 方法 (CardNN-GS) 的一种特殊情况。

情况 2. 如果 $\phi_i < \phi_j$, 则有 $\phi_i - \phi_j < 0$, 因此,

$$\lim_{\sigma \rightarrow 0^+} \arctan \frac{\phi_i - \phi_j}{\sigma} = -\frac{\pi}{2} \quad (\text{A.30a})$$

$$\Rightarrow \lim_{\sigma \rightarrow 0^+} \frac{\pi}{2} + \arctan \frac{\phi_i - \phi_j}{\sigma} = 0 \quad (\text{A.30b})$$

$$\Rightarrow \lim_{\sigma \rightarrow 0^+} \Omega(\phi_i, \phi_j, \sigma, \varepsilon) = 0 \quad (\text{A.30c})$$

情况 3. 如果 $\phi_i \geq \phi_j \geq \phi_k$ (等号不同时成立), 则 $\phi_i - \phi_k > 0$. 因此,

$$\lim_{\sigma \rightarrow 0^+} \frac{1}{1 + \exp \frac{\phi_i - \phi_k}{\sigma}} = 0 \quad (\text{A.31a})$$

$$\Rightarrow \lim_{\sigma \rightarrow 0^+} \Omega(\phi_i, \phi_j, \sigma, \varepsilon) = 0 \quad (\text{A.31b})$$

情况 4. 如果 $\phi_{k+1} \geq \phi_i \geq \phi_j$ (等号不同时成立), 则 $\phi_{k+1} - \phi_j > 0$. 因此,

$$\lim_{\sigma \rightarrow 0^+} \frac{1}{1 + \exp \frac{\phi_{k+1} - \phi_j}{\sigma}} = 0 \quad (\text{A.32a})$$

$$\Rightarrow \lim_{\sigma \rightarrow 0^+} \Omega(\phi_i, \phi_j, \sigma, \varepsilon) = 0 \quad (\text{A.32b})$$

综上所述, 如果 $\phi_i \neq \phi_k$ 且 $\phi_j \neq \phi_{k+1}$, $\Omega(\phi_i, \phi_j, \sigma, \varepsilon)$ 分子的第二项在 $\sigma \rightarrow 0^+$ 时候会趋向 0. 因此, $\sigma \rightarrow 0^+$ 时的上界中只需考虑 $\phi_i = \phi_k, \phi_j = \phi_{k+1}$ 这一项:

$$\lim_{\sigma \rightarrow 0^+} \mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq \frac{(\log 2)m\tau \left(|\phi_k - \phi_{k+1}| \left(\frac{\pi}{2} + \arctan \frac{\phi_k - \phi_{k+1}}{2\sigma} \right) \right)}{(1 - \varepsilon)(\phi_k - \phi_{k+1})^2} \quad (\text{A.33a})$$

$$\leq \frac{(\pi \log 2)m\tau}{(1 - \varepsilon)|\phi_k - \phi_{k+1}|} \quad (\text{A.33b})$$

□

注记。基于上述结论, 如果 $|\phi_k - \phi_{k+1}| > 0$, 当 $\sigma \rightarrow 0^+$ 时, 公式 (3.56) 退化为公式 (3.51) 中的界 (仅相差一个常数项):

$$\lim_{\sigma \rightarrow 0^+} \mathbb{E}_u \left[\|\mathbf{T}^{\sigma*} - \tilde{\mathbf{T}}^*\|_F \right] \leq \frac{(\pi \log 2)m\tau}{(1 - \varepsilon)|\phi_k - \phi_{k+1}|} \quad (\text{A.34})$$

其中暗含了一个很强的假设, 即 $|\phi_k - \phi_{k+1}| > 0$. 如果 $\phi_k = \phi_{k+1}$, 上述 CV 上界会变得发散. 由于在实践中 ϕ_k, ϕ_{k+1} 是神经网络的预测, 这一假设并不总是成立. 作为对比, 当 $\sigma > 0$ 时, 公式 (3.56) 中带 Gumbel 噪声的 CV 上界对任意的 ϕ_k, ϕ_{k+1} 永远是有界的. 上述理论结论同样在 3.4 节的实验中得到了验证, 特别是同伦算法 CardNN-HGS 进一步强调了更小的 CV 和更高的求解性能之间的关系.

致 谢

逝者如斯，不舍昼夜。八年前踏入交大校门时，我从未想过要在此待上整整八年；如今写到了博士论文的最后一段，我却不敢相信已经在交大待上了八年。感谢默默见证我成长、成熟的母校，感谢掉进塘上盛开的荷花，感谢致远湖畔如水的夜色，感谢软件大楼开阔的天台——在这个充满变化的时代，母校的一草一木几乎未曾改变，随时让我心安。

感谢我无悔的本科四年，没有钻营、没有内耗，在母校开放的平台下自由探索，只做自己感兴趣的事。感谢我从高中到大学的好友肖李然、傅家威，我们一起通宵建模、趴在地上造车，即使从未得过大奖，依然乐观、快乐。感谢交大航模队的吴俊琦老师，您带我领略了多彩的无人机世界，给予我充分的自由度进行探索、创造。感谢我的本科同学冯世祥、高洁锋，我们配合默契、合作顺利，完美交付了一个个科创项目。感谢电子系的袁焱、孟桂娥老师，您们教会了我如何实践、设计、创新，成为一名卓越的工程师。感谢本科“变态”“高压”的课程安排，让我在心思最为跳脱的年纪，学会了如何专注、如何安排时间、如何交付完美的答卷。

博士四年，有笑有泪。在此诚挚地感谢给予我无私帮助、指导的师长与同学：

感谢杨小康老师，您教会了我做学术，更教会了我如何做人。您为我提供了宽松、自由的科研环境，同时给予我充分的支持和指导。在我遭遇挫折、陷入迷茫的时候，您教导我如何走出困境，如今看来，当年的挫折已经不值一提；在我骄傲自满的时候，您鞭策我向更深、更难的方向探索，提醒我时刻反思研究的价值和意义。

感谢严骏驰老师，您在 2018 年的夏天留住了迷茫的我，带我走入学术的殿堂。感谢您将技术、思想对我倾囊相授，您对机器学习的深刻洞见、对前沿方向的精准把控令我折服。您总是能在复杂的技术框架中抽象出关键问题，在冗长的论文中写下点睛之笔。对待学生，您大方、没有架子、亲切随和，时刻为学生的发展着想，是我学习的榜样。

感谢京东的沈力博士、蚂蚁集团的华致刚博士——我们的合作非常愉快，我受益良多。感谢曾一起奋斗、并肩作战的同学（和如今的老师）们：刘畅、于天舒、张天启、任麒冰、张哲熙、方浩树、孙建华、郭子奥、陆晗、李泽楠、耿皓宇、邴洋、张昀浩、杨学、吴齐天、王晓星、吴寅初、杨晓江、蒋泽天、严格——我们流过的汗水终将结成果实。

最后，感谢我的父母和爱人，拥有你们是我无上的幸运。我从不奢求大富大贵的生活，但你们创造的物质条件让我免去了后顾之忧。感谢你们永远支持我的每一个决定，感谢你们与我一起分享成功的喜悦，也感谢你们给我无限的精神支持。你们的儿子和丈夫没有辜负你们，今后也一定不会。

时隔两年，又见北京，我又一次踏进了雍和宫。与两年前还算适中的人流量相比，这个人们口中的“龙潜福地”，如今挤满了随着社交软件而来的年轻人。站在18米高的大佛脚下，抬头仰望，我是如此的渺小。一旁的小伙问他的女伴：“明知四大皆空，又如何予取予求？”是啊，求佛便是迷信，自然是假的；但我又多么能理解那些模仿着妈妈辈，在佛像前跪下的年轻人——太多的历史进程无法阻挡，除了普照的佛光，又能将精神寄托在何处？只是这种妥协多少有些无奈，并不足以让我释然。忽然间我想到，带着心想事成的愿望，你不远万里来到此地，燃香、跪拜、许愿，你的虔诚正说明你对此事的重视非同寻常。又或许，雍和宫灵验的秘密，不过是投入正比于产出之下的幸存者偏差——世界上从来都没有佛。渡你过劫的那个人，终将是你自己。

2023年3月31日于北京

作者简介及攻读学位期间科研成果目录

作者简介

汪润中，浙江杭州人，本科毕业于上海交通大学电子系，博士就读于上海交通大学计算机系，首届吴文俊人工智能博士班班长，研究方向为机器学习求解（排列型）组合优化，导师为杨小康教授和严骏驰教授。博士期间获得中国计算机学会视觉专委会学术新锐奖（全国3人）、国家奖学金、上海交通大学“85届”计算机系奖学金（全系3人）、微软亚洲研究院奖学金提名奖（亚洲17人）等荣誉。

攻读博士学位期间发表的科研成果

- [1] **Runzhong Wang**, Junchi Yan and Xiaokang Yang. “Unsupervised Learning of Graph Matching with Mixture of Modes via Discrepancy Minimization.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.²⁷
- [2] **Runzhong Wang**, Junchi Yan and Xiaokang Yang. “Neural Graph Matching Network: Learning Lawler’s Quadratic Assignment Problem with Extension to Hypergraph and Multiple-graph Matching.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022, vol. 44, no. 9, pp. 5261-5279.
- [3] **Runzhong Wang**, Junchi Yan and Xiaokang Yang. “Combinatorial Learning of Robust Deep Graph Matching: an Embedding based Approach.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023, vol. 45, no. 6, pp. 6984 - 7000.
- [4] **Runzhong Wang**, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang and Junchi Yan. “LinSATNet: The Positive Linear Satisfiability Neural Networks.” *International Conference on Machine Learning (ICML)*, 2023.
- [5] **Runzhong Wang***, Ziao Guo*, Shaofei Jiang, Xiaokang Yang and Junchi Yan. “Deep Learning of Partial Graph Matching via Differentiable Top-K.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [6] **Runzhong Wang**, Li Shen, Yiting Chen, Xiaokang Yang, Dacheng Tao and Junchi Yan. “One-Shot Neural Combinatorial Optimization Solvers: Theoretical and

²⁷作者名字后的 * 代表共同第一作者。TPAMI 是人工智能领域的顶级期刊，ICLR、ICML、NeurIPS 是机器学习领域的顶级会议，CVPR、ICCV 是计算机视觉领域的顶级会议。其中，除 ICLR 是清华 A 类会议，其余均为 CCF-A 类会议/期刊。

- Empirical Notes on the Cardinality-Constrained Case.” *International Conference on Learning and Representations (ICLR)*, 2023.
- [7] **Runzhong Wang**, Zhigang Hua, Gan Liu, Jiayi Zhang, Junchi Yan, Feng Qi, Shuang Yang, Jun Zhou, Xiaokang Yang. “A Bi-Level Framework for Learning to Solve Combinatorial Optimization on Graphs.” *Neural Information Processing Systems (NeurIPS)*, 2021.
- [8] **Runzhong Wang**, Tianqi Zhang, Tianshu Yu, Junchi Yan and Xiaokang Yang. “Combinatorial Learning of Graph Edit Distance via Dynamic Embedding.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [9] **Runzhong Wang**, Junchi Yan and Xiaokang Yang. “Graduated Assignment for Joint Multi-Graph Matching and Clustering with Application to Unsupervised Graph Matching Network Learning.” *Neural Information Processing Systems (NeurIPS)*, 2020.
- [10] **Runzhong Wang**, Junchi Yan and Xiaokang Yang. “Learning Combinatorial Embedding Networks for Deep Graph Matching.” *International Conference on Computer Vision (ICCV Oral)*, 2019.
- [11] 汪润中, 严骏驰, 杨小康。“预测性优化的电力系统规划方法。”中国专利: CN202211231748.3, 2022。
- [12] 汪润中, 郭子奥, 蒋泽天, 刘畅, 陆嘉馨, 陈心妍, 严骏驰。“基于深度学习的图匹配问题求解软件。”软著登字第 10524202 号, 2022。
- [13] Chang Liu, Zetian Jiang, **Runzhong Wang**, Lingxiao Huang, Pinyan Lu and Junchi Yan. “Revocable Deep Reinforcement Learning with Affinity Regularization for Outlier-Robust Graph Matching.” *International Conference on Learning and Representations (ICLR)*, 2023.
- [14] Han Lu, Zenan Li, **Runzhong Wang**, Qibing Ren, Xijun Li, Mingxuan Yuan, Jia Zeng, Xiaokang Yang and Junchi Yan. “ROCO: A General Framework for Evaluating Robustness of Combinatorial Optimization Solvers on Graphs.” *International Conference on Learning and Representations (ICLR)*, 2023.
- [15] Chang Liu, Chenfei Lou, **Runzhong Wang**, Alan Yuhuan Xi, Li Shen, Junchi Yan. “Deep Neural Network Fusion via Graph Matching with Applications to Model Ensemble and Federated Learning.” *International Conference on Machine Learning (ICML)*, 2022.

- [16] Qibing Ren, Qingquan Bao, **Runzhong Wang**, Junchi Yan. “Appearance and structure aware robust deep visual graph matching: Attack, defense and beyond.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [17] Tianshu Yu, **Runzhong Wang**, Junchi Yan and Baoxin Li. “Deep Latent Graph Matching.” *International Conference on Machine Learning (ICML)*, 2021.
- [18] Tianshu Yu, **Runzhong Wang**, Junchi Yan and Baoxin Li. “Learning Deep Graph Matching with Channel-Independent Embedding and Hungarian Attention.” *International Conference on Learning and Representations (ICLR)*, 2020.
- [19] 刘畅, **汪润中**, 严骏驰。“基于强化学习的设施布局优化方法。”中国专利: CN202111405874.1, 2021。