上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：图匹配问题的机器学习求解方法

学生姓名：陆嘉馨

学生学号：518030910412

专　　业：计算机科学与技术（致远荣誉计划）

指导教师：张宁

学 院(系)：电子信息与电气工程学院、致远学院

上海交通大学

本科生毕业设计（论文）任务书

课题名称：图匹配问题的机器学习求解方法

执行时间：2022 年 2 月 至 2022 年 5 月

教师姓名：张宁　　职称：助理研究员

学生姓名：陆嘉馨　　学号：518030910412

专业名称：计算机科学与技术（致远荣誉计划）

学院(系)：电子信息与电气工程学院、致远学院

毕业设计（论文）基本内容和要求：

背景和研究内容：图匹配问题作为计算机视觉中一个重要问题，在包括图片配准、形状匹配、三维重建等问题上有广泛应用。图匹配问题通过对两个或多个不同事物间一阶、二阶，甚至更高阶的相似度建模，寻找两个或多个图之间的节点对应关系。更复杂的图匹配问题包括多图匹配问题、协同聚类匹配问题、部分图匹配问题等。多图匹配问题中，输入往往是多张图，利用多张图匹配之间的一致性等更多的信息，可以找到更好的匹配结果。协同聚类匹配问题中，输入的多张图中包含多种类别的图，需要在寻找匹配的同时进行图的聚类。部分图匹配指匹配不是双射的情况，也就是图和图不能完全匹配的情况。此时，如何处理外点，是该问题的重点。

随着机器学习技术的发展，机器学习方法在图匹配问题中的应用得到了广泛的研究。包括 GMN、PCA、NGM、BBGM 在内的多种机器学习求解方法被提出，并在二图匹配这一问题上得到了优秀的结果。但这些模型往往在训练和应用上只适用于二图匹配的场景，在多图匹配、多图匹配聚类、部分图匹配等各类更加困难的问题上的表现较差。同时，这些方法也无法利用这些复杂问题中更多的信息来进行训练。因而，它们的表现与实际应用仍有距离。

本研究的主要内容是为这些更复杂的图匹配问题构建合适的机器学习流程和算法，使用有监督学习、无监督学习、对比学习等机器学习方法，结合现有的传统求解器，提出具有高鲁棒性的，能够适用于这些复杂场景的机器学习方法，并期望在几个主流数据集上获得优秀的结果。

毕业设计（论文）进度安排：

序号	毕业设计（论文）各阶段内容	时间安排	备 注
1	文献调研	2 月	
2	方案设计	3 月	
3	实验验证	4 月	
4	撰写论文	5 月	
5	毕业答辩	6 月	

课题信息：

课题性质 ： 设计☐　　论文 √

课题来源*：国家级☐　　省部级☐　　校级☐　　横向☐　　预研 √

项目编号 _____

其他_____

指导教师签名：张宁

2022 年 2 月 8 日

学院（系）意见：

审核通过

院长（系主任）签名：_____

年　　月　　日

学生签名：陆嘉韬

2022 年 2 月 8 日

上海交通大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文《图匹配问题的机器学习求解方法》，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：陆嘉馨

日期：2022 年 5 月 24 日

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在___年解密后适用本授权书。

本学位论文属于

不保密√。

（请在以上方框内打“√”）

学位论文作者签名：陆嘉馨　　指导教师签名：张宁

日期：2022 年 5 月 24 日　　日期：2022 年 5 月 24 日

图匹配问题的机器学习求解方法

摘　要

图匹配问题作为计算机视觉中的一个重要 NP 难问题，通过对两张或多张图之间的一阶二阶甚至更高阶的相似度建模，寻找点和点、边和边之间的匹配。除了传统的二图匹配，还有包括多图匹配、多图聚类匹配、在线匹配在内的多种拓展场景。其中多图匹配在多张图的情况下进一步强调了循环一致性，多图聚类匹配问题的输入为多张不同类型的图，在匹配的同时需要对图进行聚类，在线匹配问题要求更高的匹配效率和对增量式输入的适配。随着机器学习技术被引入图匹配问题的研究，前作多着眼于二图完全匹配的场景，对于多图、多类、多外点的场景鲁棒性极低，且难以直接应用到不同的问题设置中。

我们首先针对其中的多图聚类匹配问题进行研究，提出了一个基于最小化-最大化框架的收敛性算法 M3C，并提出松弛聚类指标来拓展优化空间。基于 M3C，进一步开发了一个具有基于边的相似度学习、损失函数和伪标签选择的无监督学习模型 UM3C。我们还从中间图的角度对图匹配问题中节点的语义信息进行建模，首次将中间图这一概念引入机器学习的图匹配问题中，并提出端到端的学习框架 UPM。我们提出中间图相似度学习和外点感知的损失函数来提高模型对部分图匹配问题的鲁棒性和对多应用场景的适应能力，使其成为第一个可以同时处理二图匹配、多图匹配、多图聚类匹配、增量式匹配问题的机器学习算法。各数据集上大量实验表明，M3C，UM3C，和 UPM 在准确性和效率上都有最先进的性能，且消融实验、泛化性测试、敏感性测试都证明每一个组成部分的有效性。

关键词：图匹配，机器学习，多图匹配，多图匹配聚类，部分图匹配，组合优化

MACHINE LEARNING METHODS ON GRAPH MATCHING

ABSTRACT

Graph Matching has been a standing NP-Hard problem in computer vision. The problem refers to finding the node and edge correspondences between two or multiple graphs by maximizing the affinity score. There are multiple extension cases on graph matching problems, including multiple graph matching, mixture graph matching and clustering, and online matching. The multiple graph matching problem stresses the importance of cycle consistency among multiple graphs. Mixture graph matching and clustering aim to distinguish graph categories while matching. The online matching problem requires higher efficiency and special treatment for incoming graphs. Machine learning has shown its potential in graph matching for real-world data, while most previous works only focus on two graph matching problems and have low robustness against multiple graphs, categories, and outliers. They can hardly transfer to different settings.

We first study the mixture graph matching and clustering problem by designing an algorithm M3C with a theoretical convergence guarantee brought by the Minorize-maximization framework and flexible optimization space via the relaxed cluster indicator. Based on M3C, we develop an unsupervised learning pipeline UM3C with edge-wise affinity learning and loss design, as well as a pseudo label selection scheme. We also model inlier matching and outlier detection through a universe graph perspective and build an end-to-end learning pipeline UPM. Universe level affinity metric learning and outlier-aware loss enable UPM with high robustness and the ability to cope with various extension settings. To our best knowledge, this is the first deep learning network that can cope with two-graph matching, multiple-graph matching, online matching, and mixture graph matching simultaneously. Extensive experiments on several real-world datasets show the state-of-the-art performance of M3C, UM3C, and UPM on both accuracy and efficiency.

Key words: Graph Matching, Machine Learning, Multiple Graph Matching, Mixture Graph Matching and Clustering, Partial Matching, Combinatorial Optimization

目 录

第一章 绪论 …… 1
1.1 图匹配问题介绍 …… 1
1.1.1 二图匹配与多图匹配 …… 1
1.1.2 多图匹配聚类问题 …… 1
1.1.3 机器学习图匹配模型 …… 2
1.2 研究内容与贡献综述 …… 2
第二章 相关研究 …… 4
2.1 二图匹配问题 …… 4
2.2 多图匹配问题 …… 4
2.3 多图匹配聚类问题 …… 4
2.4 图匹配的机器学习模型 …… 5
2.5 部分图匹配问题与最大公共子图问题 …… 5
2.6 本章小结 …… 5
第三章 协同聚类匹配求解器与非监督学习模型 …… 6
3.1 相关符号表述及定义 …… 6
3.2 多图聚类匹配问题定义与分析 …… 6
3.3 最小化-最大化算法框架 …… 7
3.4 松弛聚类指标 …… 8
3.5 基于松弛的算法：M3C …… 10
3.5.1 初始化 …… 10
3.5.2 构造步骤 …… 10
3.5.3 最大化步骤 …… 11
3.6 非监督学习模型：UM3C …… 11
3.6.1 模型总览 …… 12
3.6.2 基于边的相似度学习 …… 12
3.6.3 基于边的相似度损失函数 …… 13
3.6.4 伪标签选择 …… 13
3.6.5 实现细节 …… 13
3.7 本章小结 …… 15
第四章 协同聚类匹配求解器与模型实验 …… 16
4.1 数据集与对比方法 …… 16
4.2 测试指标 …… 16
4.3 测试规则与流程 …… 17
4.4 多图匹配聚类问题表现 …… 18
4.4.1 多图聚类匹配问题实验 …… 18
4.4.2 不同图片和类别数目下实验 …… 20
4.5 M3C 与 UM3C 方法分析 …… 21
4.5.1 松弛聚类指标构造方法比较 …… 21

4.5.2 消融实验 ····· 21
4.5.3 收敛性测试 ····· 22
4.5.4 相似度矩阵泛化性测试 ····· 23
4.5.5 超参数敏感性测试 ····· 23
4.6 时间复杂度分析 ····· 24
4.7 本章小结 ····· 24
第五章 基于中间图的图匹配模型 ····· 25
5.1 中间图视角设计动机 ····· 25
5.1.1 全局视角下的节点区分 ····· 25
5.1.2 图匹配的统一模型 ····· 27
5.2 中间图相似度学习 ····· 27
5.2.1 中间图相似度构造 ····· 27
5.2.2 中间图结构分析 ····· 29
5.3 外点聚类的损失函数设计 ····· 30
5.3.1 基础损失函数设计 ····· 31
5.3.2 外点感知的损失函数设计 ····· 32
5.4 对比学习视角的特征优化 ····· 33
5.5 算法局限性与拓展空间讨论 ····· 34
5.6 本章小结 ····· 34
第六章 基于中间图的图匹配模型实验 ····· 35
6.1 数据集与对比方法 ····· 35
6.2 测试指标与流程 ····· 36
6.2.1 测试指标 ····· 36
6.2.2 训练与测试流程 ····· 37
6.3 部分图匹配表现 ····· 37
6.4 UPM 方法的探索与分析 ····· 40
6.4.1 中间图结构 ····· 40
6.4.2 中间图相似度与二图相似度 ····· 41
6.4.3 泛化性测试 ····· 42
6.4.4 超参数敏感性测试 ····· 43
6.5 在线匹配与多图匹配聚类拓展 ····· 44
6.5.1 在线多图匹配 ····· 45
6.5.2 多图匹配聚类 ····· 45
6.6 本章小结 ····· 46
全文总结 ····· 48
参考文献 ····· 49
致 谢 ····· 54

插图索引

图 3–1 多图聚类匹配求解器 M3C-Hard、M3C 与前作 DPMC[20]的展示与对比 …… 7
图 3–2 非监督学习模型 UM3C 流程概览。不同类型的多张图片和其中的特征点坐标作为输入给定到模型中。图结构由 Delaunay 三角剖分构造[51]。点和边特征分别由 VGG-16[52]提取并经过 SplineConv[47]优化。相似度矩阵 $\mathbf{K}$ 通过公式 3–25 将原始相似度矩阵 $\mathbf{K}^{raw}$ 和学习相似度矩阵 $\mathbf{K}^{learn}$ 相加构造。求解器 M3C 求解多图聚类匹配问题并得到伪匹配 $\mathbf{X}^{psd}$ 和伪相似度矩阵 $\mathbf{K}^{psd}$。相似度损失函数根据松弛聚类指标 $\hat{\mathbf{C}}$ 选择伪标签并被鼓励向伪相似度矩阵 $\mathbf{K}^{psd}$ 靠近。反向传播部分通过箭头展示。 …… 12
图 4–1 在不同类数、图数场景下对松弛聚类指标三种构建方式的比较。 …… 20
图 4–2 M3C 的消融实验。在 3×8 的场景下，我们依次添加各个组件。左：在匹配和聚类上的测试。右：伪标签在训练过程中按照迭代次数变化的准确性以及对应的测试结果。 …… 20
图 4–3 M3C 的算法总体收敛性测试与学习模型的相似度矩阵泛化性测试。 …… 22
图 4–4 M3C 中超参数 r 在不同松弛聚类指标构建方法中的敏感性测试。实验在 5×20 和 $5 \times 10, 2$ 外点的场景下进行。标记为我们加边到超图连通情况下的 r 的均值。 …… 23
图 5–1 基于中间图的部分图匹配学习模型（Universe Partial Matching, UPM）概览以及与传统二图匹配学习模型 BBGM 的对比。 …… 25
图 5–2 UPM 的总体学习流程。整个流程主要分成三个部分：前端特征提取、中间图相似度学习、中间图匹配求解器。在前端特征提取中，我们为每一个节点，使用 ResNet50[61]提取局部、中间、全局特征，并将他们输入 SplineConv[47]来获得最终的点特征。在中间图相似度学习中，我们构建输入图和中间图之间的中间图相似度，并使用外点感知的损失函数来训练中间图，具体方法在 5.2 和 5.3 中介绍。在测试阶段，我们使用外点过滤器和内点求解器来从连续解中求解离散解。外点过滤器会去除求得相似度矩阵的最后一列和对应外点所在行，内点求解器使用匈牙利算法[62]来计算输入图和中间图之间的匹配结果，最后从中间图匹配结果中重建二图匹配结果。 …… 28
图 5–3 中间图的两种构造方法说明。如第一张图所示，假设有三个子中间图，其节点和特征由不同颜色表示。第二、三张图展示了中间图是如何将这些子中间图进行融合的。中间图特征中空白的矩形表示中间图锚点特征 $\mathbf{F}_i$ 中被浪费的通道。基于特征的中间图锚点中多种颜色表示子中间图中的几个节点被聚合在一起。我们意在展现点特征是如何合并在一起的，边和邻接关系在这里没有意义。 …… 29
图 6–1 PascalVOC 上不同外点下匹配类型的分布。作为代表，我们测试了 BBGM 和 UPM 的表现对比。图上的数字代表不同匹配类型在各个类中的出现次数之和，我们也汇报了其占比。从图中可见，UPM 在区分节点类型上有出色的表现。 …… 39

图 6–2 WillowObject 上不同部分图匹配场景下的 F1 分数、精度、召回率。左侧的图片用于展现 UPM 对于未匹配内点的鲁棒性，右侧的图片用于展现 UPM 对于外点的鲁棒性。 …… 40
图 6–3 WillowObject 上带有 2 个外点的中间图匹配相似度分布。我们在每个类中随机取样 200 张图并计算每个点与对应中间图上锚点的匹配关系。X 轴代表基于特征的中间图和基于节点的中间图，其大小分别为 20 和 90。Y 轴代表不同类中有不同语义信息的内点和随机添加的外点，其用虚线隔开。中间图匹配点对出现的次数越多在图中的颜色越深。由于中间图上的锚点与其置换关系无关，我们在可视化的过程中调整了节点列的顺序。 …… 41
图 6–4 PascalVOC 上中间图相似度和二图相似度对比。中间图相似度 $\mathbf{K}^u$ 由 UPM 学习到的 $\boldsymbol{S}$ 按照方程 6–8 构造而成。二图相似度由 BBGM 使用 ResNet50 提取器学习得到。‘-UA’ 表示使用中间图相似度 $\mathbf{K}^u$ 的非学习求解器，‘-PA’ 表示使用二图相似度的非学习求解器。平均的 F1 分数在最后一列显示。 … 42
图 6–5 UPM 在 PascalVOC 和 WillowObject 上对于外点和未匹配内点的泛化性测试。模型在 Y 轴坐标的设置上进行训练，在 X 轴坐标的设置上进行测试。我们汇报了所有的 F1 分数。图中颜色根据 F1 分数关于其所在列的最大值进行归一化，更深的颜色代表同一个测试场景中更好的表现。一行的颜色越深代表训练模型的泛化能力越强。 …… 43
图 6–6 在线部分图匹配的 F1 分数、匹配准确率和时间开销比较图。我们在 PascalVOC 带有 0/2/4/8 个外点的设置下分别测试 UPM 和 BBGM 的结果。 …… 45
图 6–7 PascalVOC 上三种算法的可视化比较。图例与图 5–1 保持一致：蓝、橙、红色的点分别表示匹配内点、未匹配内点、外点。蓝线表示正确匹配，三种红线表示三种错误匹配。我们同时比较 UPM、BBGM、ZAC 的匹配结果，并汇报其 F1 分数和准确率。 …… 47
图 6–8 PascalVOC 上 UPM 在不同类型图上的可视化。图例与图 5–1 保持一致：蓝、橙、红色的点分别表示匹配内点、未匹配内点、外点。蓝线表示正确匹配，三种红线表示三种错误匹配。我们展现 UPM 对抗旋转、遮挡、大量外点的能力。 …… 47

表格索引

表 3–1 UM3C 中使用的 vgg16_bn 的网络结构。 …… 14
表 4–1 UM3C 参数。 …… 18
表 4–2 WillowObject 数据集下不同算法在多图聚类匹配场景中的测试结果。其中 MA 代表匹配结果，CA,CP,RI 表示聚类结果，算法时间消耗也附在表格中。 19
表 4–3 WillowObject 数据集下各算法在不同类数、图数设置中的表现结果。其中，MA 和 CA 分别表现匹配和聚类准确度。 …… 19
表 4–4 $3 \times 8, 2$ 外点下，谱聚类[37]和多割算法[43]对于非学习求解器的影响。带有 '-MC' 的算法表示使用多割算法作为聚类算法。 …… 21
表 4–5 每一轮迭代下超图结构变化。实验在 $3 \times 8, 2$ 外点的场景下进行。对于 M3C，超图结构即为求解的聚类指标，对于 DPMC，超图的结构为求解的最大生成树。这里的数字代表与上一轮相比变化的边的数目。 …… 21
表 5–1 部分图匹配中不同节点类型划分。 …… 26
表 6–1 三种错误匹配类型：误匹配、错匹配、过匹配。错误匹配类型由某个点对中两个点的类型来决定，第一行第一列分别表示两个点的类型与其对应的匹配类型。 …… 36
表 6–2 PascalVOC 上带有未匹配内点和 2 个外点的实验的 F1 分数。所有机器学习方法都使用 ResNet50 作为特征提取器。非学习方法的相似度矩阵通过 BBGM 学习到的一阶和二阶相似度构造而成。我们采用原论文、代码中提供的参数，或者对其进行微调，以得到为部分图匹配场景最合适的参数。 .. 38
表 6–3 PascalVOC 上带有未匹配内点和无外点的实验的 F1 分数比较。所有机器学习方法除了 DLGM，均使用 ResNet50 作为特征提取器。非学习方法的相似度矩阵通过 BBGM 学习到的一阶和二阶相似度构造而成。我们采用原论文、代码中提供的参数，或者对其进行微调，以得到为部分图匹配场景最合适的参数。 …… 38
表 6–4 基于特征的中间图（上）和基于节点的中间图（下）的超参数敏感性测试。基于特征的中间图在同一类的输入图上进行训练和测试，基于节点的中间图在训练时有一半的数据输入为来自不同类的图，这些输入的真值匹配设成零。UPM-24 和 UPM-244 是两类中间图的精确大小。我们在表中汇报 F1 分数。 …… 44
表 6–5 PascalVOC 上多图匹配聚类测试结果。我们保留所有未匹配内点并随机添加 2 个外点急性测试。表上的场景设置 3×8 表示随机取样 3 个类每个类取样 8 张图的情况。我们汇报 F1C 结果由于其相较于 MAC 指标更适用于部分图匹配的场景。 …… 45
表 6–6 WillowObject 上多图匹配聚类问题的测试结果。我们遵循前作[21] 中的测试设置，不添加额外的外点也不去除内点，并随机选用 8 张汽车、8 张鸭子、8 张摩托车作为测试数据。 …… 46

算法索引

算法 3–1 非学习算法：M3C ····· 11
算法 4–1 聚类算法流程 ····· 17
算法 5–1 基于中间图的部分匹配模型 (Universe Partial Matching, UPM)····· 31

符号对照表

N	待匹配的图的总数
N_c, N_g	N_c 表示聚类个数，每个类中的图数目为 $\{N_{g_1}, N_{g_2}, \dots, N_{g_{N_c}}\}$，当所有类中的图数目一致时，我们简写为 N_g
n	图中点的个数
$\mathbb{G}$	$=\{\mathcal{G}_1 \dots \mathcal{G}_N\}$，$N$ 张图的集合
$\mathcal{V}$	图顶点点集
$\mathcal{E}$	图边集
v	$\in \mathcal{V}$ 图中的点
e	$\in \mathcal{E}$ 图中的边
$\mathcal{G}$	$=(\mathcal{V}, \mathcal{E})$ 表示由点集 $\mathcal{V}_\mathcal{G}$ 和边集 $\mathcal{E}_\mathcal{G}$ 组成的图
$\mathcal{U}$	$=\{\mathcal{F}_u^{(1)} \dots \mathcal{F}_u^{(n)}\}$，带有 n 个节点的中间图
$\mathbb{X}$	$=\{\mathbf{X}_{ij}\}_{1\le i,j\le N}$，图集 $\mathbb{G}$ 中所有（可能的）匹配结果
$\mathbf{X}_{ij}(\mathbf{X}_{\mathcal{G}_i,\mathcal{G}_j})$	$\in \{0,1\}^{n_i \times n_j}$，$\mathcal{G}_i$ 和 $\mathcal{G}_j$ 之间的匹配结果
$\hat{\mathbf{X}}$	模型预测的匹配
$\mathbf{K}_{ij}$	$\in \mathbb{R}^{n_i n_j \times n_i n_j}$，$\mathcal{G}_i$ 与 $\mathcal{G}_j$ 之间的相似度矩阵
$J(\mathbf{X}), J_{ij}$	匹配 $\mathbf{X}$ 的相似度数值，或图 $\mathcal{G}_i, \mathcal{G}_j$ 之间的相似度数值
$\mathbf{F}$	图 $\mathcal{G}$ 中的点特征
$\mathbf{E}_{ij}$	图 $\mathcal{G}$ 中点 v_i, v_j 之间的边特征
$\mathbf{C}$	$\mathbf{C} \in \{0,1\}^{N\times N}$，$N$ 张图间的聚类指标。$c_{ij}=1$ 如果图 $\mathcal{G}_i$ 和图 $\mathcal{G}_j$ 来自于同一类，反之 $c_{ij}=0$
$\hat{\mathbf{C}}$	$\in \{0,1\}^{N\times N}$，松弛聚类指标
$\mathbf{A}$	$\in \{0,1\}^{N\times N}$，超图的邻接矩阵
$\mathrm{vec}(\cdot)$	对矩阵取列向量化
$\mathrm{SCC}(\cdot)$	给定聚类指标的强连通分量
r	松弛聚类指标时的超参数，表示选取的边比例
α	相似度构建时的超参数，表示传统相似度的比重
$\mathcal{S}$	点相似度
p_i	点 v_i 与中间图 $\mathcal{U}$ 之间匹配概率分布
p_{ij}	点 v_i, v_j 之间的匹配概率
$\mathcal{L}$	损失函数

第一章　绪论

1.1　图匹配问题介绍

1.1.1　二图匹配与多图匹配

图匹配（Graph Matching, GM）问题是计算机视觉领域中的一个重要问题，在图像配准[1]，图像识别[2-3]，三维图像匹配[4-5]，物体追踪[6-7]，运动结构恢复[8]中都有广泛的应用。图匹配问题通过对两张或多张图之间的一阶、二阶[9-10]，甚至更高阶的相似度[11]，来寻找点与点、边与边之间的匹配。二图匹配问题（two-graph matching）最常用的形式为二阶指派问题（Quadratic Assignment Problem, QAP）[12]，也被称为劳勒的二阶指派问题（Lawler's QAP）[13]:

$$\begin{aligned} \mathbf{X} &= \arg\max_{\mathbf{X}} \operatorname{vec}(\mathbf{X})^{\top}\mathbf{K}\operatorname{vec}(\mathbf{X}) \\ s.t.\ \ &\mathbf{X} \in \{0,1\}^{n_1\times n_2}, \mathbf{X}\mathbf{1}_{n_2} \le \mathbf{1}_{n_1}, \mathbf{X}^{\top}\mathbf{1}_{n_1} \le \mathbf{1}_{n_2} \end{aligned} \tag{1–1}$$

其中 $\mathbf{X}$ 是一个表示点与点对应关系的（局部）置换矩阵，$\operatorname{vec}(\mathbf{X})$ 是向量化的置换矩阵。$\mathbf{1}_n$ 是长度为 n 全为 1 的列向量。$\mathbf{K} \in \mathbb{R}^{n_1n_2\times n_1n_2}$ 表示相似度矩阵。其对角元素和非对角元素分别表示点与点、边与边的相似度。

二图匹配问题常见的一类拓展问题为多图匹配问题（multiple graph matching, MGM），其常见的形式为:

$$\begin{aligned} \mathbb{X} &= \arg\max_{\mathbb{X}=\{\mathbf{X}_{ij}\}_{i=1,j=1}^{N,N}} \sum \operatorname{vec}(\mathbf{X}_{ij})^{\top}\mathbf{K}_{ij}\operatorname{vec}(\mathbf{X}_{ij}) \\ s.t.\ \ &\mathbf{X}_{ij}\mathbf{1}_{n_j} \le \mathbf{1}_{n_i}, \mathbf{X}_{ij}^{\top}\mathbf{1}_{n_i} \le \mathbf{1}_{n_j}, \mathbf{X}_{ik}\mathbf{X}_{kj} \le \mathbf{X}_{ij}, \forall i,j,k \end{aligned} \tag{1–2}$$

与二图匹配相比，多图匹配更强调了循环一致性（Cycle Consistency）[14-16]，利用多张图片之间的多重关系在二图匹配的基础上提升同一类型图的匹配准确率，近年来有很多文章探讨了多图匹配问题这一场景，提出了包括在连续优化域[14, 17-18]和离散优化域[15-16, 19]上的优化方法。

1.1.2　多图匹配聚类问题

一类更复杂的拓展问题为多图匹配聚类问题（Mixture Graph Matching and Clustering, MGMC），意在匹配的同时对图进行聚类以区分不同类型的图。前两类问题在匹配过程中往往认为待匹配的图同属于一个类别，但在现实应用场景中，为图片标记标签是困难且昂贵的，甚至会需要特定的领域知识来辅助标记。比如，在药物分子生成任务中，大量的药物分子缺少标签数据且少有有效的类别区分，在匹配的过程中将不同类型的分子进行匹配是无意义的。另一方面，一些应用场景天然存在多种类型的待匹配图。比如在交通监控中，人、自行车、汽车等多种物体往往会同时出现在同一场景中，而此时进行匹配更需要区分图的类型。

两篇近期的工作[20-21]对多图聚类匹配问题进行了探索：DPMC[20]是一个非学习的算法，通过构建最大生成树，并应用基于相似度数值的衰减率来优化图与图之间的匹配；GANN[21]提出了一个仅基于点的非监督学习流程，并使用分级分配算法来预测伪标签进行训练。虽然这些算法取得了一定成效，但它们都有不少弊端：1）**收敛性**，DPMC 无法稳定收敛，而 GANN 只进行一次匹配和聚类。他们都缺少理论上的联合优化的收敛性支撑。2）

优化结构固化，DPMC 严重依赖于构建的最大生成树结构，并且只能选择 $N-1$ 条边来优化 N^2 个匹配对。3）**鲁棒性**，GANN 对噪声鲁棒性极低，其匹配精度在 2 个外点的情况下就已经大幅下降，当外点数目更多时，算法几乎无法产生有效的匹配。

1.1.3 机器学习图匹配模型

随着机器学习的发展，机器学习方法推动图匹配问题在现实图像数据集上的研究，并取得了良好的进展[22-27]。这些算法几乎都基于二图完全匹配的设置，假定参与匹配的每个点都能找到在另一张图中的对应点，此时匹配构成了双射函数，本文将其称为完全匹配。但是从现实生活中获取的数据往往无法满足此类假设：1）待匹配的图往往只是部分重合，无法保证一张图中的所有点都可以在另一张图中存在，关键点常常因为旋转、重合、剪切而缺失。2）点标签往往来自人工标记或是算法生成，而这一产生过程容易产生很大的误差。生成算法容易产生无意义的关键点，而这些关键点对于匹配问题是严重的噪音，并且无法在任何一张图中找到对应关系。3）如前文提到，极端场景中，当输入的图来自不同的类，我们就不应该将其进行匹配，而现有算法无法自动处理这一点。

因此，我们将目光放到更普适的部分图匹配[14, 18]场景中。部分图匹配是完全匹配的另一种扩展场景，与完全图匹配相比，其更具挑战并未被进行很好的研究。虽然前作 BBGM[26]通过学习负相似度来减少冗余点①的匹配，DLGM[27]通过预测高一致性的图结构来减少冗余匹配，但它们都只能处理二图匹配的场景，并且混淆不同类型的冗余点。它们的损失函数设计也容易受到数据集中无意义的外点影响。

根据以上论述，本文希望提出能够解决现有图匹配，尤其是机器学习图匹配算法模型缺点的算法和模型：

(1) 在多图匹配聚类问题中，对于匹配和聚类协同进行并具有收敛性、高鲁棒性、灵活结构的算法；

(2) 能够利用无标签数据进行非监督的多图聚类匹配机器学习的基于 QAP 问题定义的模型；

(3) 能够高效处理部分图匹配的机器学习模型；

(4) 能够同时处理包括二图匹配、多图匹配、多图聚类匹配、增量图匹配等多种图匹配问题的机器学习模型。

1.2 研究内容与贡献综述

为了解决前文的前两个问题，我们提出一个基于最小化-最大化（Minorize-Maximization, MM）框架的非学习算法 M3C（**M**inorize **M**aximization **M**atching and **C**lustering），其收敛性由算法框架保证。我们同时提出一个松弛聚类指标（relaxed cluster indicator）来对前作固化的优化结构进行松弛，以得到更合适的优化空间。为了进一步提高算法表现，同时考虑学习数据标签的难以获得，我们将 M3C 嵌入非监督学习框架，并设计基于边的相似度学习（edge-wise affinity learning）来匹配求解器，设计相似度损失函数（affinity loss）来提高相似度质量，以及伪标签选择（pseudo label selection）来提高学习模型的鲁棒性。

对于后两个问题，我们提出对部分图匹配中未匹配内点（unmatched inlier）和外点（outlier）的定义，并分析其在图匹配问题中暗含的语义信息。我们在机器学习图匹配模型中引入中间图（universe graph）这一概念，并提出机器学习算法 UPM（**U**niverse **P**artial **M**atching）。具

① 冗余点表示所有没有对应匹配的点，包括未匹配的内点和外点

体地说，所有的输入图都先与中间图进行匹配，而不是成对匹配，这一方法联合了二图和多图匹配过程，同时能够自动满足全数据集上的循环一致性。并且，中间图可以作为过滤器来区分未匹配内点和外点。我们更是设计了外点感知损失函数（outlier-aware loss）来协同区分外点。这些设计使得我们的算法不仅能够应用于二图匹配的场景，更是在多图、多图聚类匹配等场景中有极佳的表现。

综上，本文主要贡献如下：

(1) 对于多图匹配聚类问题，提出一个非学习的方法 M3C。其基于最小化-最大化算法框架并有理论上的收敛保证。同时，松弛聚类指标提供了更灵活的优化空间。该算法在实验中达到和其他的非学习算法同等甚至更好的结果。

(2) 进一步将 M3C 嵌入到非监督学习框架，提出 UM3C 算法流程。我们还提出基于边的相似度学习、相似度损失函数，以及伪标签选择算法来提高学习模型的质量和鲁棒性。其准确性大幅超过前作 GANN[21]，在 WillowObject 这一数据集上甚至超过监督学习算法 BBGM[26]和 NGM[25]。

(3) 对于部分图匹配问题，从中间图视角，我们给出未匹配内点和外点的定义，并且指出基于二图匹配的机器学习模型在区分这两类点的弊端，从而首次将中间图匹配模型引入机器学习图匹配算法中。

(4) 对于中间图匹配模型，设计一套高效的学习流程 UPM 和损失函数。具体来说，我们提出中间图相似度学习来寻找未匹配内点，以及外点感知损失函数来过滤外点，并给出详细严谨的数值分析证明这种设计如何帮助我们的模型处理部分图匹配问题。

(5) 大量实验表明，UPM 在各类主流数据集上已经超过了现有的所有机器学习模型和非学习算法，同时保有极强的泛化能力和鲁棒性。据我们所知，UPM 是第一个可以同时处理二图匹配、多图匹配、多图聚类匹配、增量式匹配问题的机器学习算法，其不仅较先前方法有更高的准确率，更显著加速匹配速度。

第二章　相关研究

2.1　二图匹配问题

传统的图匹配问题指的是二图匹配问题，其优化目标为前文所列公式1–1。GAGM[9]通过泰勒展开计算目标函数的偏导将图匹配问题转换成线性规划问题。RRWM[10]通过构建协同图（association graph）并使用随机游走（random walk）方法对图匹配问题进行求解。RRWHM[11]将 RRWM 中的协同图拓展出更高维的几何信息。DS++，DS*[28-29] 设计了不同的方式将图匹配严格放宽到凸问题(convex problem),然后将结果投射回原始解空间。LPMP[30]提出了几个图匹配问题的拉格朗日松弛（Lagrangian relaxation），并利用子梯度（sub-gradient）和对偶上升更新（dual ascent updates）来引导求解器优化拉格朗日对偶分解。

2.2　多图匹配问题

对于多图匹配问题，其求解目标如公式1–2所示。在二图匹配的基础上，多图匹配引入了循环一致性的概念来进一步利用多张图之间的匹配关系进行求解。循环一致性指的是两张图 $\mathcal{G}_i, \mathcal{G}_j$ 之间的匹配应该与两者通过某中间图 $\mathcal{G}_k$ 产生的匹配保持一致,也就是说,$\mathbf{X}_{ij} = \mathbf{X}_{ik}\mathbf{X}_{kj}$。现有的多图匹配求解器多基于循环一致性这一观察，并主要分成两类工作。

第一类工作着眼于离散求解域，其代表作包括基于组合的方法 CAO[16]，MGM-Floyd[19]，以及基于树结构的方法 MatchOpt[15]，TPMC[20]。这些方法通过寻找多张图之间的匹配组合来迭代地对初始的二图匹配进行优化，通常由机制自动满足或者通过优化目标函数来鼓励提高循环一致性。

另一类工作[14, 17, 31]尝试将多图匹配问题松弛到连续空间,再将解重新投射回来得到一个离散的匹配结果。它们通过中间图来保持全局的循环一致性，并利用一些连续域的优化方法进行求解。例如,谱分解方法(spectral clustering)[32], 半正定规划(semi-definite programming)[17], 以及基于此更鲁棒的优化方法 MatchLift[14] 使用交替方向乘子法(alternating direction methods of multipliers, ADMM）来处理部分多图匹配问题。以上列举的方法往往将基于相似度数值的匹配和基于循环一致性的优化分成两个步骤进行，另有一些工作，例如 MatchALS[18]，也探究了相似度数值和循环一致性在连续求解域上的联合优化。

增量式多图匹配（Incremental Multi-graph Matching）属于多图匹配问题的一类拓展问题，指图片增量式输入的图匹配场景，此时要求求解器有更高的效率。IMGM[33]首次提出了这一问题设置，并利用分块的思想提高了多图匹配的效率，使多图匹配算法能够适用于增量式多图匹配的问题设置中。MGM-SPFA 和 FastSPFA[19]则利用 SPFA 算法将 MGM-Floyd 中的多源最短路问题转换成更适用于增量匹配的单源最短路问题，大幅提高了算法效率。增量式多图匹配在机器学习模型中的研究目前依然是空白，现有的机器学习算法无法对增量式多图匹配问题进行很好的适配。关于二图匹配、多图匹配问题更详细的综述可见[34-35]。

2.3　多图匹配聚类问题

多图匹配聚类问题是指在求解多图匹配问题的时候，不仅找到图中节点匹配关系，还要将不同的图进行聚类，分为不同的集群。主要有两篇前作讨论了这个场景：DPMC[20] 和

GANN[21]。DPMC[20]是一个非学习的求解器。它通过在超图（supergraph）上构建一个最大生成树来得到优化结构，并根据构建的树结合基于相似度数值的衰减率更新匹配结果。GANN[21]提出一个非监督的学习流程，结合基于分级分配的算法得到的伪标签进行训练。两个方法都最后基于相似度矩阵，进行谱聚类得到最后的聚类结果。另外除了我们讨论的图层面的聚类，还有一些工作探索了联合匹配在节点层面的聚类[36]。在他们讨论的问题中，输入的图在节点匹配的同时被分割成若干个子图。

在本文中，对图聚类意指将相似的图分到同一个类别中。一种常见的方法是将每个图嵌入到希尔伯特空间（Hilbert space）随后进行聚类（例如 k-means 方法）。包括 DPMC 和 GANN 在内的前作均使用了基于相似度矩阵的谱聚类[37]方法。另一种求解方法是最大割算法（Max Cut）[38-41]或者多割算法（Multi-Cut）[42-43]。它们基于超图的视角，将每张图视作超图上的点，超图上的边为图与图之间的相似度数值，对超图进行分割。

2.4 图匹配的机器学习模型

深度学习近来越来越多地被使用在基于图像的图匹配问题中。GMN（Graph Matching Network）[22]首次将 CNN 运用到图匹配问题中。其利用 CNN 从图像中提取点特征,并结合基于谱方法的匹配算法和类似回归的匹配监督来进行学习。GNN 编码的结构[23, 44]或者几何[45]信息，结合基于交叉熵的损失函数和 Sinkhorn 微分匹配求解器，更是进一步提升了机器学习求解方法的效果。CIE（Channel-independent Embedding）[46]通过边嵌入和基于匈牙利算法的注意力机制稳定了 PCA（Permutation loss and Cross-graph Affinity GM）的端到端训练。BBGM（Blackbox Deep Graph Matching）[26]提出了一个结合了样条卷积（Spline Convolution）[47]的前端特征提取器，并通过拟合离散图匹配求解器的梯度进行反向传播[48]得到了更好的匹配结果。NGM（Neural Graph Matching Nets）[24-25]提出基于可学习的求解器的对于最一般的 Lawler's QAP 的学习方法。DLGM（Deep Latent Graph Matching）[27]基于 BBGM，利用决策和生成模型来学习图的一致性拓扑结构以提高图匹配的质量。其学习模型与其他方法正交，实则可以运用到任意的图匹配模型中。

2.5 部分图匹配问题与最大公共子图问题

最大公共子图问题（Maximum common subgraph, MCS）与二图部分匹配有着极为相似的设定。只要我们将最大公共子图中的点视作内点，其他点视为外点，最大公共子图问题就可以转换成部分图匹配问题。机器学习方法同样被引入最大公共子图问题中。MCS 极其依赖于图的同构性，并且期望寻找到图结构完全一致的子图。然而，在部分图匹配中，图的拓扑结构往往被大量的外点打乱。此前的工作[49]利用 k-hop 邻居来分解目标图，并使用 GNN 来编码子图的结构用于匹配。另一份工作[50]利用 DQN（Deep Q Learning）替代 MCS 求解器中启发式的点选择方法，得到更准确和高效的学习模型。

2.6 本章小结

本章回顾了图匹配问题相关的各类研究，包括传统的二图匹配、多图匹配、多图聚类匹配、部分图匹配和最大公共子图问题，以及现有的一些图匹配机器学习算法。我们简单介绍了图匹配问题的发展线路以及各类问题的工作路线，并介绍了不同方法的核心思想和特点。本文将基于这些工作，提出我们的图匹配问题求解器和机器学习模型。

第三章　协同聚类匹配求解器与非监督学习模型

3.1　相关符号表述及定义

定义 3.1 (匹配组合链)　匹配组合链这一概念首先被应用在 CAO、MGM-Floyd[16, 19]等算法中，指的是由一组二图匹配组合产生的匹配：$\mathbf{X}_{\mathcal{G}_1\mathcal{G}_k} = \mathbf{X}_{\mathcal{G}_1\mathcal{G}_2}\mathbf{X}_{\mathcal{G}_2\mathcal{G}_3}\dots\mathbf{X}_{\mathcal{G}_{k-1}\mathcal{G}_k}$。我们进一步定义 $\mathcal{G}_i$ 和 $\mathcal{G}_j$ 之间的匹配组合优化空间来表示二图之间所有可能的匹配组合：

$$\mathbf{P}_{\mathbb{X}}(i,j) = \{\mathbf{X}_{\mathcal{G}_i,\mathcal{G}_1}\dots\mathbf{X}_{\mathcal{G}_k,\mathcal{G}_j} | \forall \mathbf{X}_{\mathcal{G}_i,\mathcal{G}_1},\dots,\mathbf{X}_{\mathcal{G}_k,\mathcal{G}_j} \in \mathbb{X}\}. \tag{3–1}$$

定义 3.2 (超图)　超图[15-16]常用于描述多图匹配。超图 $\mathcal{G}_S = \{\mathcal{V},\mathcal{E},\mathbf{A}\}$ 中的点是待匹配的图 $\mathcal{V} = \{\mathcal{G}_1,\dots,\mathcal{G}_N\}$，边代表图与图之间的关系，边权为二图匹配的相似度值。$\mathbf{A} \in \{0,1\}^{N\times N}$ 用以表示超图的领接矩阵。

注意超图中的每一条路径都代表一个匹配组合链。因此，路径 $\mathcal{G}_1$ 到 $\mathcal{G}_k$ 的权值 $(\{e_{v_1v_2},\dots,e_{v_{k-1}v_k}\})$ 定义为匹配组合连 $\mathbf{X}_{v_1v_k} = \mathbf{X}_{v_1v_2}\mathbf{X}_{v_2v_3}\dots\mathbf{X}_{v_{k-1}v_k}$ 的相似度值。在此超图上，$\mathcal{G}_i$ 与 $\mathcal{G}_j$ 的匹配组合优化空间定义为：

$$\mathbf{P}_{\mathbf{A}}(i,j) = \{\mathbf{X}_{\mathcal{G}_i,\mathcal{G}_1}\dots\mathbf{X}_{\mathcal{G}_k,\mathcal{G}_j} | \forall a_{i1} = \dots = a_{kj} = 1\}, \tag{3–2}$$

也就是 $\mathcal{G}_i$ 与 $\mathcal{G}_j$ 之间在此超图上所有的路径组合。

定义 3.3 (聚类指标)　聚类指标 $\mathbf{C} \in \{0,1\}^{N\times N}$ 定义为:

$$c_{ij} = \begin{cases} 1 & \mathcal{G}_i,\mathcal{G}_j \text{ 来自同一类.} \\ 0 & \text{反之.} \end{cases} \tag{3–3}$$

注意 $c_{ij}c_{jk} \le c_{ik}$ 是 $\mathbf{C}$ 为（硬）聚类指标的充分必要条件。聚类的个数可以用聚类指标 $\mathbf{C}$ 的**强连通分量**来表示，这里记成 $\mathtt{SCC}(\mathbf{C})$。

3.2　多图聚类匹配问题定义与分析

本质上，多图聚类匹配问题是一个协同优化问题。匹配的结果最大化图对的相似度数值来帮助区分图片的类别，而当图片的类别被很好地区分的时候，可以反过来引导类内的匹配优化。

给定二图匹配相似度矩阵和聚类个数 N_c，多图聚类匹配问题的优化目标 $\mathcal{F}(\mathbb{X},\mathbf{C})$ 可以写成：

$$\begin{aligned} \max_{\mathbb{X},\mathbf{C}} \ & \frac{1}{\sum_{ij} c_{ij}} \cdot \sum_{ij} c_{ij} \cdot \mathrm{vec}(\mathbf{X}_{ij})^{\top}\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij}) \\ \text{s.t.} \ & \mathbf{X}_{ij}\mathbf{1}_{n_j} \le \mathbf{1}_{n_i}, \mathbf{X}_{ij}^{\top}\mathbf{1}_{n_i} \le \mathbf{1}_{n_j}, \\ & c_{ik}c_{kj} \le c_{ij}, \forall i,j,k,\ \mathtt{SCC}(\mathbf{C}) = N_c \\ & \mathbf{X}_{ij} \in \{0,1\}^{n_i\times n_j}, c_{ij} \in \{0,1\}. \end{aligned} \tag{3–4}$$

第一行约束条件要求 $\mathbf{X}_{ij}$ 是一个（部分）置换矩阵，第二行要求聚类指标严格将图片分成 N_c 个类。$\frac{1}{\sum_{ij} c_{ij}}$ 是为了防止聚类大小影响的正则项。同一类内的循环一致性可以通过 $\mathbf{X}_{ik}\mathbf{X}_{kj} \le \mathbf{X}_{ij}, \forall c_{ik} = c_{kj} = c_{ij} = 1$ 来保证，或者根据算法的设计得到增强。

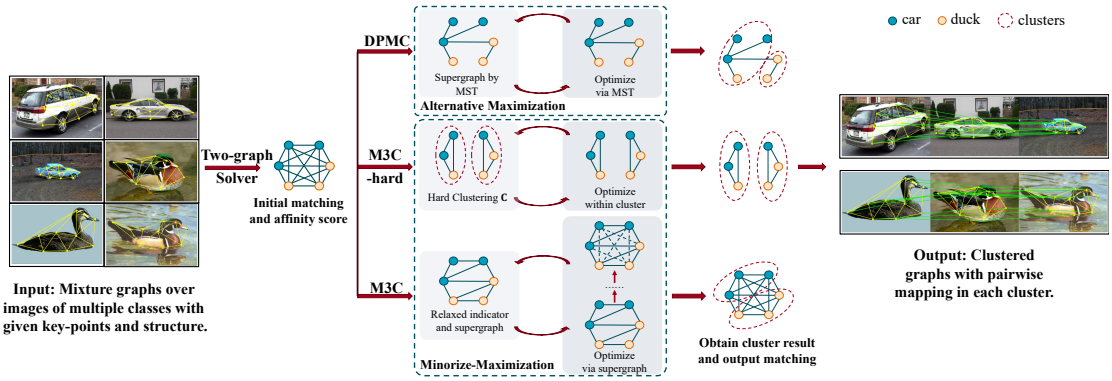


图 3–1 多图聚类匹配求解器 M3C-Hard、M3C 与前作 DPMC[20]的展示与对比。顶部：DPMC 使用基于树的优化方法并且没有收敛性保证。中部：M3C-Hard 有基于最小化-最大化算法的收敛性保证，但因为硬聚类指标，其聚类匹配的效果很差。底部：M3C 将硬聚类的条件进行松弛得到松弛聚类指标，在有收敛性保证的同时提供了更好的匹配聚类解。

Figure 3–1 Illustration and comparison of the peer method DPMC[20] for solving joint graph matching and clustering, and our M3C-Hard and M3C. Top: DPMC applies tree-based optimization and has no convergence guarantees. Mid: M3C-hard enjoys the convergence by Minorize-Maximization, but still suffers from hard clustering (see Proposition3.1). Bottom: M3C relaxes the constraints on cluster structure and could converge to a better solution.

3.3 最小化-最大化算法框架

前文指出 GANN[21]是一个一次性串联的匹配和聚类算法,其流程忽略了多图聚类匹配问题中两个问题的天然的联合关系。DPMC[20]提出了一种交替优化匹配和聚类的系统（见图3–1），但是其收敛性极差。因此为了结合多图聚类匹配问题中匹配与聚类天然的协同优化关系，同时提供收敛性保证，我们这里提出一个基于最小化-最大化算法（Minorize-Maximization,, MM）的框架。

注意到我们的优化目标 $\mathcal{F}(\mathbb{X}, \mathbf{C})$ 中有两个变量，所以不能直接使用 MM 算法。因此我们使用 $h(\mathbb{X})$ 来代表 $\mathbb{X}$ 的理想的聚类结果，来将 $\mathcal{F}(\mathbb{X}, \mathbf{C})$ 转变成仅包含 $\mathbb{X}$ 单变量的优化目标 f。新的优化目标定义为 $f(\mathbb{X}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}))$。不失一般性，我们有，

$$\max_{\mathbb{X}} f(\mathbb{X}) = \max_{\mathbb{X}} \mathcal{F}(\mathbb{X}, h(\mathbb{X})) = \max_{\mathbb{X}} \max_{\mathbf{C}|\mathbb{X}} \mathcal{F}(\mathbb{X}, \mathbf{C}) = \max_{\mathbb{X},\mathbb{C}} \mathcal{F}(\mathbb{X}, \mathbf{C}) \tag{3–5}$$

MM 算法框架的工作原理是找到一个替代函数 $g(\mathbb{X}|\mathbb{X}_0) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}_0))$，使目标函数 $f(\mathbb{X})$ 最小化。优化函数 g 将提高目标函数的值或使其保持不变。该过程具体步骤介绍如下：

- **构造步骤**：通过寻找现有匹配 $\mathbb{X}^{(t)}$ 上最优的聚类来构造替代函数 $g(\mathbb{X}|\mathbb{X}^{(t)}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}^{(t)}))$：

$$\begin{aligned} &\mathbf{C}^{(t)} = h(\mathbb{X}^{(t)}) = \arg\max_{\mathbf{C}} \frac{1}{\sum_{ij} c_{ij}} \cdot \sum_{ij} c_{ij} \cdot \mathrm{vec}(\mathbf{X}_{ij}^{(t)})^{\top} \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}_{ij}^{(t)}) \\ &\text{s.t. } c_{ik}c_{kj} \le c_{ij}, \forall i, j, k, \ \mathrm{SCC}(\mathbf{C}) = N_c. \end{aligned} \tag{3–6}$$

替代函数 $g(\mathbb{X}|\mathbb{X}^{(t)})$ 有：

$$g(\mathbb{X}|\mathbb{X}^{(t)}) \le f(\mathbb{X}) \text{ 对于所有} \mathbb{X}, \ g(\mathbb{X}|\mathbb{X}^{(t)}) = f(\mathbb{X}^t). \tag{3–7}$$

- **最大化步骤**：最大化 $g(\mathbb{X}|\mathbb{X}^{(t)})$:

$$\begin{aligned}\mathbb{X}^{(t+1)} &= \arg\max_{\mathbb{X}} g(\mathbb{X}|\mathbb{X}^{(t)}) = \arg\max_{\mathbb{X}} f(\mathbb{X}, h(\mathbb{X}^{(t)})) \\ &= \arg\max_{\mathbb{X}} \frac{1}{\sum_{ij} c_{ij}^{(t)}} \cdot \sum_{ij} c_{ij}^{(t)} \cdot \mathrm{vec}(\mathbf{X}_{ij})^{\top} \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}_{ij}) \\ s.t.\ & \mathbf{X}_{ij}\mathbf{1}_{n_j} \le \mathbf{1}_{n_i}, \mathbf{X}_{ij}^{\top}\mathbf{1}_{n_i} \le \mathbf{1}_{n_j}.\end{aligned} \tag{3–8}$$

以上迭代循环可以保证 $f(\mathbb{X})$ 收敛，也就是算法框架收敛，证明如下。

证明 (收敛性) 根据以上定义，我们令 $\mathbf{C} = h(\mathbb{X})$ 为 $\mathbb{X}$ 求解最优聚类指标，并定义 $f(\mathbb{X}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}))$ 将优化问题转化成单变量优化。我们进一步定义了基于现有最优聚类的替代函数 $g(\mathbb{X}|\mathbb{X}^{(t)}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}^{(t)}))$。该替代函数 $g(\mathbb{X}|\mathbb{X}^{(t)})$ 有两重性质。第一，因为 $f(\mathbb{X}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}))$ 利用了现有最优聚类结果，对于所有的 $\mathbb{X}$ 有：

$$g(\mathbb{X}|\mathbb{X}^{(t)}) = \mathcal{F}(\mathbb{X}, h(\mathbb{X}^{(t)})) \le \mathcal{F}(\mathbb{X}, h(\mathbb{X})) = f(\mathbb{X}) \tag{3–9}$$

第二，同样有

$$g(\mathbb{X}^{(t)}|\mathbb{X}^{(t)}) = \mathcal{F}(\mathbb{X}^{(t)}, h(\mathbb{X}^{(t)})) = f(\mathbb{X}^{(t)}) \tag{3–10}$$

接下来，我们最大化替代函数 $g(\mathbb{X}|\mathbb{X}^{(t)})$ 而不是原始函数 $f(\mathbb{X})$，优化公式如前文公式3–8所示，这一步可以通过现有的多图匹配求解器来进行。

我们交替地进行构造步骤和最大化步骤，对于每一次迭代，优化目标有：

$$f(\mathbb{X}^{(t+1)}) \ge g(\mathbb{X}^{(t+1)}|\mathbb{X}^{(t)}) \ge g(\mathbb{X}^{(t)}|\mathbb{X}^{(t)}) = f(\mathbb{X}^{(t)}). \tag{3–11}$$

故该优化目标不降。

根据图匹配问题的定义，优化目标显然有最优上界。单调有上界，故收敛。 □

由此可见，我们的框架可以应用不同的图聚类算法，例如谱聚类、多割算法、联合聚类算法[37, 42-43]来构造替代函数。同时，各类多图匹配算法[14, 16, 18-19]可以用于最大化步骤。

3.4 松弛聚类指标

前面介绍的框架有收敛性保证，但是构造步骤中的直接聚类产生的硬聚类指标会迅速收敛到次优解。我们通过以下命题说明这个问题。

命题 3.1 (硬聚类收敛速度) 如果每个类内的图数量 $\{N_{g_1}, N_{g_2}, \ldots, N_{g_{N_c}}\}$ 被固定，聚类指标一步收敛并且在之后的优化过程中固定：

$$\mathbf{C}^{(t)} = \mathbf{C}^{(t+1)},\ \text{if}\ \{N_{g_1}^{(t)}, N_{g_2}^{(t)}, \ldots, N_{g_{N_c}}^{(t)}\} = \{N_{g_1}^{(t+1)}, N_{g_2}^{(t+1)}, \ldots, N_{g_{N_c}}^{(t+1)}\}. \tag{3–12}$$

证明 (硬聚类收敛速度) 根据 MM 框架中的最大化步骤，如果我们忽略循环一致性，两张图之间的匹配结果是独立更新的。因此，$\mathbf{X}_{ij}$ 的提升并不会影响到其他匹配的更新结果。所以有：

$$\begin{aligned} J_{ij}^{(t+1)} &\ge J_{ij}^{(t)},\ \ \forall c_{ij}^{t} = 1 \\ J_{ij}^{(t+1)} &= J_{ij}^{(t)},\ \ \forall c_{ij}^{t} = 0 \end{aligned} \tag{3–13}$$

其中 $J_{ij} = \mathrm{vec}(\mathbf{X}_{ij})^{\top}\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij})$ 表示两张图的相似度数值。

我们使用反证法来证明。假设 $\mathbb{C}^{(t)} \neq \mathbb{C}^{(t+1)}$ 成立，根据优化框架，我们有：

$$f(\mathbb{X}^{(t+1)}) \geq g(\mathbb{X}^{(t+1)}|\mathbb{X}^{(t)}) \geq g(\mathbb{X}^{(t)}|\mathbb{X}^{(t)}), \tag{3–14}$$

也就是说

$$\mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t+1)}) \geq \mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t)}) \geq \mathcal{F}(\mathbb{X}^{(t)}, \mathbf{C}^{(t)}). \tag{3–15}$$

因此，

$$\begin{aligned} &\mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t+1)}) - \mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t)}) \\ =&\frac{1}{\sum c_{ij}^{(t+1)}} \sum_{ij} c_{ij}^{(t+1)} \cdot J_{ij}^{(t+1)} - \frac{1}{\sum c_{ij}^{(t)}} \sum_{ij} c_{ij}^{(t)} \cdot J_{ij}^{(t+1)} \\ \geq&0. \end{aligned} \tag{3–16}$$

因为 $\{N_{g_1}^{(t)}, N_{g_2}^{(t)}, \ldots, N_{g_{N_c}}^{(t)}\} = \{N_{g_1}^{(t+1)}, N_{g_2}^{(t+1)}, \ldots, N_{g_{N_c}}^{(t+1)}\}$，故有：

$$\sum c_{ij}^{(t+1)} = \sum c_{ij}^{(t)} \tag{3–17}$$

根据方程3–13，我们进一步得到

$$\begin{aligned} &\mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t+1)}) - \mathcal{F}(\mathbb{X}^{(t+1)}, \mathbf{C}^{(t)}) \\ =&\frac{1}{\sum c_{ij}^{(t)}} (\sum_{ij} c_{ij}^{(t+1)} \cdot J_{ij}^{(t+1)} - \sum_{ij} c_{ij}^{(t)} \cdot J_{ij}^{(t+1)}) \\ =&\frac{1}{\sum c_{ij}^{(t)}} (\sum_{c_{ij}^{(t+1)}=1, c_{ij}^{(t)}=0} J_{ij}^{(t+1)} - \sum_{c_{ij}^{(t+1)}=0, c_{ij}^{(t)}=1} J_{ij}^{(t+1)}) \\ =&\frac{1}{\sum c_{ij}^{(t)}} (\sum_{c_{ij}^{(t+1)}=1, c_{ij}^{(t)}=0} J_{ij}^{(t)} - \sum_{c_{ij}^{(t+1)}=0, c_{ij}^{(t)}=1} J_{ij}^{(t+1)}) \\ \geq&0 \end{aligned} \tag{3–18}$$

同时，根据 $J_{ij}^{(t+1)} \geq J_{ij}^{(t)}, \forall c_{ij}^{t} = 1$，我们有

$$\sum_{c_{ij}^{(t+1)}=1, c_{ij}^{(t)}=0} J_{ij}^{(t)} - \sum_{c_{ij}^{(t+1)}=0, c_{ij}^{(t)}=1} J_{ij}^{(t)} \geq \sum_{c_{ij}^{(t+1)}=1, c_{ij}^{(t)}=0} J_{ij}^{(t)} - \sum_{c_{ij}^{(t+1)}=0, c_{ij}^{(t)}=1} J_{ij}^{(t+1)}, \tag{3–19}$$

也就是

$$\frac{1}{\sum c_{ij}^{(t+1)}} \sum_{ij} c_{ij}^{(t+1)} \cdot J_{ij}^{(t)} - \frac{1}{\sum c_{ij}^{(t)}} \sum_{ij} c_{ij}^{(t)} \cdot J_{ij}^{(t)} \geq 0. \tag{3–20}$$

因此，

$$\mathcal{F}(\mathbb{X}^{(t)}, \mathbf{C}^{(t+1)}) \geq \mathcal{F}(\mathbb{X}^{(t)}, \mathbf{C}^{(t)}) \tag{3–21}$$

意味着 $\mathbf{C}^{(t)}$ 不是 $\mathbf{X}^{(t)}$ 的最优聚类解，与假设矛盾。所以我们有 $\mathbf{C}^{(t)} = \mathbf{C}^{(t+1)}$。 □

命题3.1表明，当给定目标聚类大小时，聚类指标是固定的。也就是说，MM 框架中的替代函数的构造只探索了每个类的大小，一旦类的大小被固定，聚类指标就会在一步之内收敛。此外，硬聚类指标限制了最大化的优化空间。在这种硬聚类框架下，匹配结果不能从根本上提高聚类质量，从而使其不能真正成为一个联合优化框架。因此，我们提出松弛对聚类指标的硬约束。

原始硬聚类指标的本质是找到一定数量的具有高平均相似度数值的图对（这是由聚类目标决定的）。基于类似的观察，我们为新的聚类指标 $\hat{\mathbf{C}}$ 提供了两个基于图对数量的松弛：

$$\begin{aligned} \sum_{ij} \hat{c}_{ij} &= r \cdot N^2 \quad \text{（全局约束，global constraint）;} \\ \sum_{j} \hat{c}_{ij} &= r \cdot N, \forall i \quad \text{（局部约束，local constraint）} \end{aligned} \tag{3–22}$$

其中 $r \in [0,1]$ 是一个用于调整所选图对的比例的超参数。全局约束限制了选择图对的总体数量，而局部约束则限制了每个图所连接的图对数量。我们把这个新的聚类指标 $\hat{\mathbf{C}}$ 称为**松弛聚类指标（relaxed cluster indicator）**，它有以下两个优点：

1) 松弛聚类指标忽略了聚类数量的硬性要求和聚类的传递关系 $c_{ik}c_{kj} \le c_{kj}$。相反，它对每个图进行单独判断，因此拥有一个更灵活的结构，可以探索更多的匹配信息。详细的实验结果会在第四章中给出。

2）它有利于无监督学习中的伪标签选择。它可以选择具有较高相似度数值的图对，从而提高训练中伪标签的质量。细节将在3.6.4中介绍。

3.5 基于松弛的算法：M3C

根据我们提出的松弛聚类指标 $\hat{\mathbf{C}}$，我们展示我们的算法 M3C（Minorize-Maximization Matching and Clustering）由初始化、构造、最大化三部分组成。M3C 的展示可见图3–1和算法3–1。

3.5.1 初始化

通过常见的二图匹配算法来获得初始二图匹配 $\mathbb{X}^0$。为了算法效率以及和前作[16, 19]的公平比对，我们选择 RRWM[10]作为初始的二图匹配求解器。

3.5.2 构造步骤

根据3.4中提出的算法，求解松弛聚类指标 $\hat{\mathbf{C}}$ 即为：

$$\begin{aligned} \hat{\mathbf{C}}^{(t)} =& \hat{h}(\mathbb{X}^{(t-1)}) = \arg\max_{\hat{\mathbf{C}}} \frac{1}{r \cdot N^2} \cdot \sum_{ij} c_{ij} \cdot \mathrm{vec}(\mathbf{X}_{ij}^{(t-1)})^\top \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}_{ij}^{(t-1)}) \\ \text{s.t.} \sum_{ij} c_{ij} =& r \cdot N^2 \text{ (global constraint) or } \sum_{j} c_{ij} = r \cdot N, \forall i \text{ (local constraint).} \end{aligned} \tag{3–23}$$

给定 $\mathbb{X}^{(t-1)}$ 和 r, 相似度数值和归一项是固定的。因此，求解方程3–23就是选择有更高相似度数值的图对。对于全局约束，我们根据相似度数值对所有图对进行排序，并取最高的 rN^2 个图对。对于局部约束，对于每张图，我们对其邻居的相似度数值进行排序，并取最高的 rN 个。此种方法类似 k 最近邻居（k-nearest-neighbor）。以上两种构造方法分别记作 **global-rank** 和 **local-rank**。

我们提出另一种融合了全局和局部约束的构造算法 **bi-rank**。首先定义一对图 $(\mathcal{G}_u, \mathcal{G}_v)$ 的局部排序为 $R_{uv} = i + j$，其中 $\mathcal{G}_u$ 是 $\mathcal{G}_v$ 第 i 近的邻居，$\mathcal{G}_v$ 是 $\mathcal{G}_u$ 第 j 近的邻居。此时我们可以全局地取排序 $\{R_{uv}\}$ 最小的一部分图。根据这种双向排序方法，我们得到的松弛聚类指标 $\hat{\mathbf{C}}$ 可以同时反应图与图之间全局和局部相似度关系。

算法 3–1 非学习算法：M3C

Input: 迭代次数 T, 相似度矩阵集合 $\mathbb{K}$, 构建策略.

通过二图匹配求解器 RRWM[10]得到初始二图匹配结果 $\mathbb{X}^{(0)}$;

for $t = 1 : T$ **do**

/* 构建步骤 */

利用三种构建算法：global-rank, local-rank, bi-rank 通过求解方程3–23中的 $\hat{\mathbf{C}}^t = \hat{h}(\mathbb{X}^{(t-1)})$ 构建替代函数 $g(\mathbb{X}|\mathbb{X}^{(t-1)})$;

/* 最大化步骤 */

令 $\mathbf{A}^{(t)} = \hat{\mathbf{C}}^{(t)}$ 构建超图。根据方程3–24最大化 $g(\mathbb{X}|\mathbb{X}^{(t-1)})$ 来优化超图 $\mathbf{A}^{(t)}$ 上的匹配链;

end

根据 $\mathbb{X}^{(t)}$ 和 $\mathbf{K}$，通过多割算法或者聚类算法[37, 42-43]求解聚类结果;

Output: 匹配结果 $\mathbb{X}^{(t)}$，聚类结果 $\mathbf{C}$

3.5.3 最大化步骤

硬聚类条件下的最大化步骤只寻求优化每个聚类中的匹配。因为聚类指标被松弛，我们现在展示如何将松弛后的指标应用于最大化步骤中。

正如我们在定义3.2中指出的，超图常常用于描述多图匹配中图和图之间的关系。通过松弛聚类指标，我们实际上可以构造一个非完全的超图：如果两张图属于同一类，那么就在它们之间连一条边。事实上，经过这种构建的超图的邻接矩阵 $\mathbf{A}$ 在数值上与我们构造的松弛聚类指标 $\hat{\mathbf{C}}$ 是完全一样的，所以我们令 $\mathbf{A} = \hat{\mathbf{C}}$。

令 $\mathbf{P}_{\mathbf{A}^{(t)}}$ 表示基于邻接矩阵 $\mathbf{A}$ 的所有路径（匹配组合优化空间）。遵循与多图匹配在完全超图上的路径优化规则相似的思路，我们将优化目标转变成：

$$\mathbb{X}^{(t)} = \underset{\mathbf{X}_{ij} \in \mathbf{P}_{\mathbf{A}^{(t)}}(i,j)}{\arg\max} \sum_{ij} \mathrm{vec}(\mathbf{X}_{ij})^{\top} \mathbf{K}_{ij} \mathrm{vec}(\mathbf{X}_{ij}) \tag{3–24}$$

可以看出，$\mathcal{G}_i$ 与 $\mathcal{G}_j$ 之间在超图上更多的路径能够产生更大的优化空间 $\mathbf{P}_{\mathbf{A}^{(t)}}(i, j)$，也就意味着两个属于同一类的图的匹配能够有更大的可能性被优化。这个性质和原本的硬聚类指标有着一致性。并且在对方程3–8给定松弛聚类指标 $\hat{\mathbf{C}}$ 时，其与方程3–24保持一致。

此外，当松弛聚类指标退化成原本的硬聚类指标时，超图会变成几个不相通的连通块：此时对匹配的优化只会发生在同一个连通块内。因此方程3–24是对方程3–8的良定义泛化。

根据以上算法，松弛聚类指标 $\hat{\mathbf{C}}$ 不再固定，也因此可以扩大最大化步骤中的图匹配组合优化空间。而最大化步骤中更好的优化空间能够提高匹配质量，进一步提高在构建步骤中松弛聚类指标的质量。两个步骤达到协同优化聚类和匹配结果。

3.6 非监督学习模型：UM3C

深度学习方法在图匹配中已经得到了广泛的应用[22-23, 25-26]，并证实了 CNN 和 GNN 作为特征提取器的强大作用。多图聚类匹配问题天然有着进行无监督学习的潜力[21]：通过非学习的求解器，我们能够从多张混合类别的图中求解出训练需要的伪标签来指导机器学习。因此，我们希望通过无监督学习来学习图像的点和结构特征，以进一步提高非学习求解器的表现。

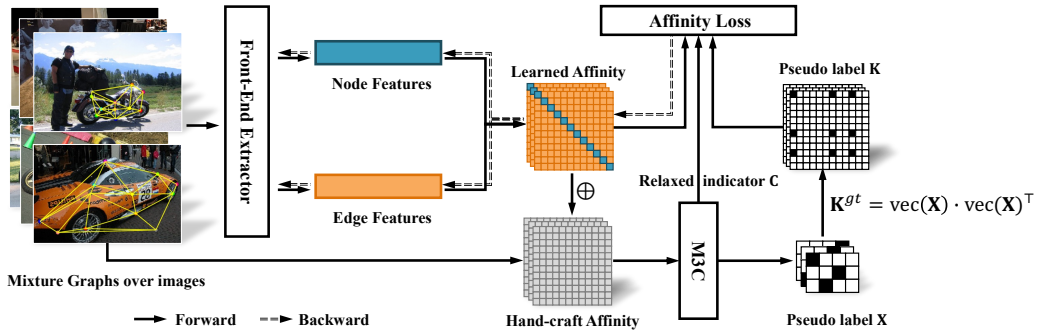


图 3–2 非监督学习模型 UM3C 流程概览。不同类型的多张图片和其中的特征点坐标作为输入给定到模型中。图结构由 Delaunay 三角剖分构造[51]。点和边特征分别由 VGG-16[52]提取并经过 SplineConv[47]优化。相似度矩阵 K 通过公式3–25将原始相似度矩阵 $\mathbf{K}^{raw}$ 和学习相似度矩阵 $\mathbf{K}^{learn}$ 相加构造。求解器 M3C 求解多图聚类匹配问题并得到伪匹配 $\mathbf{X}^{psd}$ 和伪相似度矩阵 $\mathbf{K}^{psd}$。相似度损失函数根据松弛聚类指标 $\hat{\mathbf{C}}$ 选择伪标签并被鼓励向伪相似度矩阵 $\mathbf{K}^{psd}$ 靠近。反向传播部分通过箭头展示。

Figure 3–2 Pipeline for unsupervised learning model: UM3C. Mixture of graphs of different classes are input where key point coordinates are given and graphs are built with Delaunay triangulation[51]. Node features and edge features are extracted by VGG-16[52] and SplineConv[47]. The final affinity K is obtained by adding learned affinity $\mathbf{K}^{learn}$ and hand-craft affinity $\mathbf{K}^{raw}$ (see Eq.3–25). M3C solve the MGMC on K and N_c to gain the pseudo matching $\mathbf{X}^{psd}$, as well as the pseudo affinity $\mathbf{K}^{psd}$. Affinity loss selects the pseudo pair based on relaxed indicator $\hat{\mathbf{C}}$ and encourages the learned affinity matrix to get close to $\mathbf{K}^{psd}$. Gradient back-propagation is shown by the arrows

3.6.1 模型总览

如图3–2所示，我们将 M3C 嵌入到非监督学习的框架中得到非监督学习模型 UM3C。输入的多张图会通过两条线路得到两种相似度矩阵：一条通过由 CNN 和 GNN 组成的学习模型得到相似度矩阵 $\mathbf{K}^{learn}$；另一条通过传统的构造方法，利用高斯核得到相似度矩阵 $\mathbf{K}^{raw}$。两个相似度矩阵会被组合起来然后经过我们的求解器 M3C 来得到学习模型训练需要的伪标签。同时伪标签会被用于计算相似度损失函数并指导学习模型构建更好的相似度矩阵 $\mathbf{K}^{learn}$。

3.6.2 基于边的相似度学习

GANN[21]使用了 Koopmans-Beckmann's QAP[53]形式的二次指派构造，而这种构造只着眼于点相似度和图结构的简单对齐。并且在训练初期，模型产生的点相似度不可靠时，会对求解器的结果产生很大的影响。

为了提高模型的鲁棒性和效果，UM3C 将问题构造拓展到更常规的 Lawler's QAP[12]问题。我们在学习模型中增加了基于边的相似度学习，这种调整帮助求解器可以使用二阶信息来对抗外点干扰。我们结合学习的和传统的相似度矩阵构造了基于边的相似度矩阵 $\mathbf{K}$，二者之间通过超参数 α 调整比例关系：

$$\mathbf{K} = \mathbf{K}^{learn} + \alpha \mathbf{K}^{raw}. \tag{3–25}$$

这里我们采用了前作的学习流程[26]来建立学习得到的相似度矩阵 $\mathbf{K}^{learn}$。点和边的特征通

过 VGG-16[52]提取后经 Spline CNN[47]优化。同时，基于传统方法的相似度矩阵 $\mathbf{K}^{raw}$ 同样仿照前作[16, 19]进行构建，其中包含了通过高斯核构建的几何信息。最后通过归一化统一两个相似度矩阵的尺度。

这种构造既保持了传统相似度矩阵稳定的特性，同时能够得到学习到的特征更具表达能力的特点。在最初几代训练中，传统的相似度矩阵 $\mathbf{K}^{raw}$ 在提高伪标签 $\mathbf{X}^{psd}$ 的质量中起到主导作用；而后期，学习到的相似度矩阵 $\mathbf{K}^{learn}$ 将优于传统相似度矩阵 $\mathbf{K}^{raw}$，最终，总体相似度矩阵可以得到进一步提高。

3.6.3 基于边的相似度损失函数

前作中，损失函数常用在预测的匹配 $\hat{\mathbf{X}}$ 和真实匹配 $\mathbf{X}^{gt}$（或者伪标签 $\mathbf{X}^{psd}$）之间。然而，预测的匹配 $\hat{\mathbf{X}}$ 会同时受到相似度矩阵构造和求解器的影响，当损失函数仅作用在匹配之间时，损失函数的导数会倾向于适配求解器。也就是说，损失函数会让学到的相似度矩阵适配于某个特定的求解器而非图匹配问题本身。这样最终会导致学习到的相似度矩阵只能在一个特定的求解器上有比较好的表现，而缺乏对图匹配整个问题的迁移能力。

因此，我们设计了一个可以跳过求解器求解的相似度损失函数，直接将反传给基于边的相似度矩阵：

$$\mathcal{L}(\mathbf{K}^{learn}, \mathbf{K}^{psd}) = \sum_{pq} k_{pq}^{psd} \log(k_{pq}^{learn}) + (1 - k_{pq}^{psd}) \log(1 - k_{pq}^{learn}) \tag{3–26}$$

其中伪相似度矩阵 $\mathbf{K}^{psd}$ 通过伪匹配标签 $\mathbf{X}^{psd}$ 构造：

$$\mathbf{K}^{psd} = \mathrm{vec}(\mathbf{X}^{psd}) \cdot \mathrm{vec}(\mathbf{X}^{psd})^{\top}. \tag{3–27}$$

注意这里 k_{pq} 代表 $\mathbf{K}$ 中第 p 行第 q 列的元素。这里的相似度损失函数排除了求解器的影响并且只专注于构造更适合真实结果的相似度矩阵。这种构造方法提高了相似度矩阵的鲁棒性，也能让其更加适配不同类型的求解器。

3.6.4 伪标签选择

此外，我们还在松弛聚类指标 $\hat{\mathbf{C}}$ 的指导下优化了损失函数和伪标签对的计算。

前作 GANN[21]在计算损失函数的时候将推断出所有属于同一类的图对的损失函数相加。这么做，导致一旦在聚类过程中出现一个错误，会有很多参与计算的伪标签对并非实际应该使用的伪标签对。这种一个小错误导致的大幅度偏差会影响 $\mathbf{X}^{psd}$ 的质量。同时，哪怕选择的是同一类中的图对，如果匹配准确度很低，伪标签的正确性也没有保证。

所以与其使用聚类中的所有图对，UM3C 结合了松弛聚类指标 $\hat{\mathbf{C}}$ 并选择有着更高相似度数值排名的图对作为伪标签。根据更高的相似度矩阵代表更好的匹配准确度这一观察，伪相似度矩阵 $\mathbf{K}^{psd}$ 的质量可以进一步得到提高。综上，总体的损失函数为：

$$\mathcal{L}_{all} = \sum_{ij} \hat{c}_{ij} \cdot \mathcal{L}(\mathbf{K}_{ij}^{learn}, \mathbf{K}_{ij}^{psd}). \tag{3–28}$$

它选择准确度较高的匹配对，使伪标签更接近于真实标签。

3.6.5 实现细节

网络结构和特征提取器 我们使用前作[26]中的提取器并对其进行一定的修改，具体模型流程如下：

表 3–1 UM3C 中使用的 vgg16_bn 的网络结构。预训练权重来自 PyTorch。加粗的行代表 `relu4_2`，`relu5_1`，和 `final` 层。他们的权重被用于计算点、边、全局特征。

Table 3–1 Network structure of vgg16_bn that we applied in UM3C. The pre-trained weight is download by PyTorch. The bold line denotes the `relu4_2`, `relu5_1`, and `final` layers whose outputs are applied as node features, edge features and global features.

	Layer	Channels	Kernel		Layer	Channels	Kernel
1	Conv2d	(3, 64)	(3, 3)	3	ReLU	(256, 256)	-
	BatchNorm2d	(64, 64)	-		MaxPool2d	-	2
	ReLU	(64, 64)	-	4	Conv2d	(256, 512)	(3, 3)
	Conv2d	(64, 64)	(3, 3)		BatchNorm2d	(512, 512)	-
	BatchNorm2d	(64, 64)	-		ReLU	(512, 512)	-
	ReLU	(64, 64)	-		**Conv2d**	**(512, 512)**	**(3, 3)**
	MaxPool2d	-	2		BatchNorm2d	(512, 512)	-
2	Conv2d	(64, 128)	(3, 3)		ReLU	(512, 512)	-
	BatchNorm2d	(128, 128)	-		Conv2d	(512, 512)	(3, 3)
	ReLU	(128, 128)	-		BatchNorm2d	(512, 512)	-
	Conv2d	(128, 128)	(3, 3)		ReLU	(512, 512)	-
	BatchNorm2d	(128, 128)	-		MaxPool2d	-	2
	ReLU	(128, 128)	-	5	**Conv2d**	**(512, 512)**	**(3, 3)**
	MaxPool2d	-	2		BatchNorm2d	(512, 512)	-
3	Conv2d	(128, 256)	(3, 3)		ReLU	(512, 512)	-
	BatchNorm2d	(256, 256)	-		Conv2d	(512, 512)	(3, 3)
	ReLU	(256, 256)	-		BatchNorm2d	(512, 512)	-
	Conv2d	(256, 256)	(3, 3)		ReLU	(512, 512)	-
	BatchNorm2d	(256, 256)	-		Conv2d	(512, 512)	(3, 3)
	ReLU	(256, 256)	-		BatchNorm2d	(512, 512)	-
	Conv2d	(256, 256)	(3, 3)		ReLU	(512, 512)	-
	BatchNorm2d	(256, 256)	-		**MaxPool2d**	**-**	**2**

- 详细的网络结构见表3–1首先计算在 ImageNet[54]上预训练的 VGG-16[52]网络中 `relu4_2`, `relu5_1` 层后的特征，分别作为 $\mathbf{F}_1$ 和 $\mathbf{F}_2$。将起连接得到 CNN 特征

$$\mathbf{F} = \texttt{CONCAT}(\mathbf{F}_1, \mathbf{F}_2) \tag{3–29}$$

- 将得到的特征和图邻接矩阵传递进行几何特征优化。其中，图邻接矩阵 $\mathcal{A}$ 由对定点位置进行 Delaunay 三角剖分[51]得到。之后通过 SplineConv[47]将更高阶的结构和几何信息嵌入到点特征 $\mathbf{F}^n$ 中。

$$\mathbf{F}^n = \texttt{SplineConv}(\mathbf{F}, \mathcal{A}) \tag{3–30}$$

SplineConv 的形式为：

$$\mathbf{F}_i^n = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{F}_j^n \cdot h_\Theta(\mathbf{E}_{i,j}) \tag{3–31}$$

其中 $\mathbf{F}_i^n$ 代表点 v_i 的点特征，$\mathcal{N}(i)$ 代表点 v_i 的邻居，$\mathbf{E}_{ij}$ 代表 v_i 和 v_j 之间一条边的特征，h_Θ 代表通过 B-样条卷积计算的核函数。

相似度矩阵 K 的构造 本文中，我们主要提及了两类相似度矩阵：传统的相似度矩阵 $\mathbf{K}^{raw}$ 和学习到的相似度矩阵 $\mathbf{K}^{learn}$。前者在非学习和学习方法中都有使用，而后者在 UM3C 中使用。

我们首先介绍 $\mathbf{K}^{raw}$ 的构造方法，主要遵循前作[15-16, 19-20, 55]。在这种传统的相似度矩阵 $\mathbf{K}^{raw}$ 中没有点相似度的参与，而边相似度由两部分组成：长度相似度和角度相似度。对于每条边 $e=((x_1,y_1),(x_2,y_2))$，它的长度特征记作 $d_e=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$，它的角度特征记作 $\theta_e=\tan^{-1}(\frac{y_1-y_2}{x_1-x_2+\epsilon})$。总体的边相似度记作：

$$k^{raw}_{e_1,e_2}=\exp\left(-\frac{1}{\sigma^2}\left(\beta\left|d_{e_1}-d_{e_2}\right|+(1-\beta)\left|\theta_{e_1}-\theta_{e_2}\right|\right)\right) \tag{3–32}$$

学习相似度 $\mathbf{K}^{learn}$ 从深度学习网络中提取得到[21, 26]。其中点的相似度 $\mathbf{F}^n_i$ 由 VGG-16 和 SplineConv 提取，边特征计算为：

$$\mathbf{F}^e_{ij}=\mathbf{F}^n_i-\mathbf{F}^n_j \tag{3–33}$$

学习的相似度矩阵 $\mathbf{K}^{learn}$ 为：

$$\begin{aligned}\mathbf{K}_u&=(\mathbf{F}^n)^\top\mathbf{\Lambda}\mathbf{F}^n\\ \mathbf{K}_q&=(\mathbf{F}^e)^\top\mathbf{\Lambda}\mathbf{F}^e\end{aligned} \tag{3–34}$$

其中 $\mathbf{K}_u$ 代表一阶相似度而 $\mathbf{K}_q$ 代表二阶相似度，为了训练稳定性，我们将 $\mathbf{\Lambda}$ 设成 $\mathbf{I}$。

最后，根据前文方程3–25构造 UM3C 中使用的相似度矩阵 $\mathbf{K}$。

3.7 本章小结

本章从基础符号定义出发，介绍了多图聚类匹配问题的定义与我们的观察。首先提出了一个最小化-最大化算法框架并证明其收敛性，为接下来的松弛提供收敛性保证。随后提出一种松弛聚类指标，将硬分类产生的硬聚类指标进行松弛，以提供更好的优化空间。我们提出一种基于收敛算法框架和松弛的非学习算法 M3C，并将其与多图匹配求解器和图聚类方法结合起来。最后，我们将提出的非学习算法 M3C 嵌入到非监督学习流程中，并提出基于边的相似度学习和相似度损失函数，以及基于松弛聚类指标的伪标签选择，来进一步完善学习模型，提高鲁棒性。

第四章　协同聚类匹配求解器与模型实验

本章实验中，所有非学习的求解器均在一台 2.30GHz 四核 CPU、内存 16GB 的个人笔记本电脑上运行，软件平台为 Matlab R2020a。所有机器学习方法在一个 CPU 为 Xeon-3175X@3.10GHz，GPU 为 RX8000，内存 128GB 的 Linux 工作台上运行。

4.1　数据集与对比方法

Willow-ObjectClass 数据集　该数据集[56]包含来自 Caltech-256[57]和 Pascal VOC 2007[58]数据集共 5 类 256 张图像：40 张汽车、40 张摩托车、50 张鸭子、66 张酒瓶和 109 张人脸。每张图片都有 10 个关键点的标签，我们也会在实验中加上生成的外点，除特殊说明外，外点均为随机产生。图结构由 Delaunay 三角剖分构造。对于基于学习的模型，我们遵循标准的流程，将图像裁剪到物体的边界框，并缩放到 256×256 像素。我们随机取样 N_c 个类和每个类中 N_g 张图并混合作为输入（以下简记为 $N_c\times N_g$）。除特殊说明外，我们报告重复 50 次测试后的平均结果。

对比方法　对于我们提出来的方法, 实验中主要使用 M3C-Hard, M3C, 和 UM3C：M3C-Hard 是我们提出的基于硬聚类的 MM 框架（见3.3），其中在构造步骤使用了谱聚类方法，在最大化步骤中使用了 MGM-Floyd 方法。M3C 是3.5中介绍的基于松弛聚类指标的算法。UM3C 是3.6 中介绍的无监督学习模型。后两种方法中，如无特殊说明，松弛聚类指标的构建算法均使用了 **bi-rank** 算法。

我们同时在非学习和学习的设置下评估我们提出方法的表现。对于非学习求解器，我们主要和 DPMC[20]以及 MGM-Floyd[19]对比，两者分别为多图聚类匹配和多图匹配中最先进的求解器。其他对比方法也包括[14, 16, 18]。

对于学习模型的比较，我们将 UM3C 与无监督的 GANN[21] 以及两个最先进的监督模型 BBGM[26]和 NGMv2[44]进行比较。

4.2　测试指标

我们根据前作[20-21]使用匹配准确性（Matching Accuracy, MA)[20]，聚类纯度（Clustering Purity, CP)[59]，随机指标指数（Rand Index, RI)[60]，聚类准确性（Clustering Accuracy, CA)[20]作为我们对匹配和聚类准确性的评测指标。

令一个聚类表示为 $C=\{\mathcal{G}_1\dots\mathcal{G}_n\}$，聚类真值为 C^{gt}，预测的聚类为 C，聚类指标真值为 $\mathbf{C}^{gt}(c_{ij}^{gt})$，预测的聚类指标为 $\mathbf{C}(c_{ij})$，以上指标的数学定义如下：

匹配准确性（Matching Accuracy, MA）　我们只考虑类内的匹配精度，因此通过对单一类的标准化，我们有

$$\mathbf{MA}=\frac{1}{\sum c_{ij}^{gt}}\sum_{ij}c_{ij}^{gt}\cdot\mathtt{ACC}(\mathbf{X}_{ij}),\tag{4–1}$$

其中 $\mathtt{ACC}(\mathbf{X}_{ij})$ 表示匹配 $\mathbf{X}_{ij}$ 的准确性。

算法 4–1 聚类算法流程

Input: 匹配结果 $\mathbb{X}$, 相似度矩阵 $\{\mathbf{K}_{ij}\}$.

1 对每一个图对，获取相似度数值：$J_{ij} = \mathrm{vec}(\mathbf{X}_{ij})^{\top}\mathbf{K}_{ij}\mathrm{vec}(\mathbf{X}_{ij})$;

2 对相似度数值矩阵稀疏化：$\left\{J_{ij}^{k}\right\} = \mathrm{KNN}(\{J_{ij}\}, k)$;

3 对相似度数值矩阵 $\left\{J_{ij}^{k}\right\}$ 应用聚类算法，例如谱聚类[37]、多割算法[43]等，以获得聚类指标 $\mathbf{C}$.

Output: 聚类结果 $\mathbf{C}$.

聚类纯度（Clustering Purity, CP） 定义为：

$$\mathbf{CP} = \frac{1}{N}\sum_{i=1}^{N_c}\max_{j\in\{1,\dots,N_c\}}\left|C_i \cap C_j^{gt}\right|, \tag{4–2}$$

其中 C_i' 是预测的第 i 个类别，C_j 是第 j 个类别的真值数。

随机指标指数（Rand Index, RI） 表示总体上聚类正确的图对数：

$$\mathrm{RI} = \frac{1}{N^2}\cdot\Big(\sum_{c_{ij}=1,c_{ij}^{gt}=1} 1 + \sum_{c_{ij}=0,c_{ij}^{gt}=0} 1\Big) \tag{4–3}$$

其中 $\sum_{c_{ij}=1,c_{ij}^{gt}=1} 1$ 表示同一类中预测标签一致的图数，$\sum_{c_{ij}=0,c_{ij}^{gt}=0} 1$ 表示不同类中预测标签不一致的图数。

聚类准确性（Clustering Accuracy, CA） 定义为：

$$\mathrm{CA} = 1 - \frac{1}{N_c}\left(\sum_{C_a}\sum_{C_a'\neq C_b'}\frac{\left|C_a'\cap C_a\right|\left|C_b'\cap C_a\right|}{\left|C_a\right|\left|C_a\right|} + \sum_{C_a'}\sum_{C_a\neq C_b}\frac{\left|C_a'\cap C_a\right|\left|C_a'\cap C_b\right|}{\left|C_a\right|\left|C_b\right|}\right) \tag{4–4}$$

其中 C_a, C_b 为聚类真值，C_a', C_b' 为预测的聚类。

4.3 测试规则与流程

我们总体遵循前作[20-21]的流程进行测试，以下介绍两个主要的处理步骤。

将二图匹配和多图匹配求解器运用于多图聚类匹配设置。 为了将二图匹配和多图匹配求解器运用于多图聚类匹配的问题设置下，我们首先需要从这些求解器中得到二图匹配结果 $\mathbb{X}$。之后，我们通过得到的匹配结果 $\mathbb{X}$ 和给定的相似度矩阵 $\{\mathbf{K}_{ij}\}$ 通过图聚类算法求解聚类指标 $\mathbf{C}$。在评估过程中，我们使用 $\mathbb{X}$ 和 $\mathbf{C}$ 对匹配、聚类结果进行评价。

聚类算法流程。 对所有的求解器（包括二图匹配、多图匹配、多图聚类匹配求解器），我们采用同一种聚类流程，具体如算法4–1所示。

第一步是对所有图对计算相似度数值 J_{ij}，随后需要对相似度数值矩阵进行稀疏化。此处我们遵循前作[20]中提出的流程来获得对聚类更高效的矩阵。特别的，对于一个图对，如果其中一张图是另一张图的 k 近邻，他们相应的相似度数值保留为 J_{ij}（和 J_{ji}），否则，其

表 4–1 UM3C 参数。
Table 4–1 Parameter of UM3C.

param	WillowObject 3×8	WillowObject 5×10	description
lr	10^{-3}	10^{-3}	learning rate
lr-steps	$\{100, 500\}$	$\{100, 500\}$	lr/=10 at these number of iterations
α	train: 1 test: 0	train: 1 test: 1	weight of $\mathbf{K}^{raw}$
β	0.9	0.9	weight parameter in $\mathbf{K}^{raw}$
σ^2	0.03	0.03	the scaling factor in $\mathbf{K}^{raw}$
T	2	2	max iterations of M3C

相似度数值会被设为 0。这里的参数 k 在本文所有实验中设为 10。稀疏化后的相似度数值记为 J_{ij}^k。

接下来我们对 $\left\{J_{ij}^k\right\}$ 应用聚类算法，包括谱聚类算法[37]和多割算法[43]。4.5.2我们会具体讨论不同聚类算法的影响。其余实验中，为了保持和前作[20-21]的公平对比，我们采用的都是和前作一致的谱聚类方法。

超参数设计。 对于非学习的求解器，我们虽然在算法设计中引入了超参数 r，在实验中，为了减小不同数据分布对不同超参数的要求产生的大量调参需求，我们使用另一种策略：在超图上按照排序加边的过程中，我们不断加边直到整张超图连通，产生的邻接矩阵 $\mathbf{A}$ 也就是我们需要的松弛聚类指标 $\hat{\mathbf{C}}$。

对于学习模型，UM3C 的参数选择参见表4–1。其他学习模型的参数均按照原始论文、代码中给定的参数进行训练和测试。

4.4 多图匹配聚类问题表现

4.4.1 多图聚类匹配问题实验

我们首先比较 M3C-Hard，M3C 和 UM3C 与求解器 DPMC[20], MGM-Floyd[19], CAO[16], MatchLift[14], MatchALS[18], RRWM[10]，非监督学习模型 GANN[21]，二图匹配监督学习模型 NGMv2[44], BBGM[26]的结果。我们按照前作[21]的设计，在 $N_c = 3, N_g = 8$ 的车、摩托车、鸭子类别上进行测试。

非学习方法比较结果 表4–2显示我们非学习算法 M3C 与其他非学习的算法相比达到了极具竞争力的结果，并且有着更低的时间复杂度。在没有外点的设置上，它有超过 1% 的匹配精度提升和极高的聚类准确性（与其他方法相比只损失 1-2%）。在有 2-4 个外点的设置下，它有着 3-5% 的聚类准确性提升和仅 3-4% 的匹配准确度损失。与基线 MM 框架算法 M3C-Hard 相比，M3C 达到超过 4-6% 的 MA 提升和 6-8% 的 CA 提升，这也说明了我们松弛聚类指标的有效性。总体上而言，M3C 有最平衡的表现，同时也显示了其对外点的鲁棒性。

我们的实验还表明，并不是所有的多图匹配求解器都可以直接应用于多图匹配聚类环境。MatchLift 和 MatchALS 两个求解器的性能明显较低，我们认为这是因为这些求解器对

表 4–2 WillowObject 数据集下不同算法在多图聚类匹配场景中的测试结果。其中 MA 代表匹配结果，CA,CP,RI 表示聚类结果，算法时间消耗也附在表格中。

Table 4–2 Evaluation of matching and clustering metric with inference time for the mixture graph matching and clustering on WillowObject. MA is the metric of matching performance, and CA, CP, and RI are used to demonstrate clustering performance.

$N_c \times N_g$		3 × 8, 0 outlier					3 × 8, 2 outliers					3 × 8, 4 outliers				
Metrics	Learning	MA ↑	CA ↑	CP ↑	RI ↑	time(s) ↓	MA ↑	CA ↑	CP ↑	RI ↑	time(s) ↓	MA↑	CA ↑	CP ↑	RI ↑	time(s) ↓
RRWM[10]	free	0.748	0.815	0.879	0.871	**0.4**	0.595	0.541	0.643	0.680	**0.4**	0.572	0.547	0.661	0.685	**0.6**
MatchLift[14]	free	0.764	0.769	0.843	0.839	7.8	0.530	0.612	0.726	0.730	10.6	0.512	0.582	0.701	0.709	11.5
MatchALS[18]	free	0.635	0.571	0.689	0.702	1.3	0.245	0.39	0.487	0.576	2.5	0.137	0.383	0.480	0.571	2.6
CAO-C[16]	free	0.875	0.860	0.908	0.903	3.3	**0.727**	0.574	0.678	0.704	3.7	**0.661**	0.562	0.674	0.695	4.9
MGM-Floyd[19]	free	0.879	**0.931**	**0.958**	**0.952**	2.0	0.716	0.564	0.667	0.696	2.3	0.653	0.580	0.690	0.708	2.9
DPMC[20]	free	0.872	0.890	0.931	0.923	1.2	0.672	0.617	0.724	0.733	1.4	0.630	0.600	0.707	0.722	2.3
M3C-Hard	free	0.838	0.855	0.907	0.899	0.4	0.620	0.576	0.684	0.705	0.6	0.596	0.587	0.694	0.713	0.7
M3C (ours)	free	**0.884**	0.911	0.941	0.938	0.5	0.687	**0.653**	**0.750**	**0.758**	0.6	0.635	**0.646**	**0.748**	**0.753**	1.0
NGMv2[44]	supervised	0.885	0.801	0.843	0.825	9.0	0.780	0.927	0.952	0.941	4.7	0.744	0.886	0.916	0.906	4.7
BBGM[26]	supervised	0.939	0.704	0.751	0.758	**1.6**	0.806	0.964	0.977	0.971	4.8	0.747	0.881	0.918	0.908	6.6
GANN[21]	unsup.	0.896	0.963	0.976	0.970	5.2	0.610	0.889	0.918	0.913	20.6	0.461	0.847	0.893	0.881	30.2
UM3C (ours)	unsup.	**0.955**	**0.983**	**0.988**	**0.988**	3.2	**0.858**	**0.984**	**0.989**	**0.986**	**3.3**	**0.815**	**0.981**	**0.987**	**0.986**	**3.6**

表 4–3 WillowObject 数据集下各算法在不同类数、图数设置中的表现结果。其中，MA 和 CA 分别表现匹配和聚类准确度。

Table 4–3 Evaluation of matching and clustering accuracy by varying the number of clusters, and number of graphs in each cluster on WillowObject. MA and CA are used as matching accuracy and clustering accuracy, respectively.

$N_c \times N_g$		3 × 20, 2 outliers		4 × 20, 2 outliers		5 × 20, 2 outliers		5 × 15, 2 outliers		5 × 10, 2 outliers		3 × [20, 10, 5], 2 outliers	
Metrics	Learning	MA ↑	CA ↑	MA ↑	CA ↑	MA ↑	CA ↑	MA ↑	CA ↑	MA ↑	CA ↑	MA ↑	CA ↑
RRWM[10]	free	0.658	0.932	0.642	0.858	0.665	0.790	0.648	0.693	0.664	0.679	0.633	0.681
CAO-C[16]	free	0.849	0.946	0.812	0.855	0.820	0.790	0.801	0.708	0.804	0.679	0.787	0.757
MGM-Floyd[19]	free	0.845	0.945	0.812	0.878	0.819	0.807	0.798	0.727	0.799	0.707	0.778	0.755
DPMC[20]	free	**0.867**	0.942	0.827	0.894	0.775	0.772	0.739	0.713	0.756	0.744	**0.795**	0.823
M3C-Hard	free	0.758	**0.966**	0.782	0.908	0.726	0.824	0.710	0.753	0.722	0.719	0.727	0.744
M3C (ours)	free	0.857	0.961	**0.851**	**0.933**	**0.835**	**0.900**	**0.812**	**0.805**	**0.809**	**0.780**	0.792	**0.881**
GANN[21]	unsup.	0.532	0.834	0.589	0.801	0.528	0.784	0.551	0.802	0.552	0.827	0.475	0.802
UM3C (ours)	unsup.	**0.874**	**0.992**	**0.897**	**0.981**	**0.879**	**0.972**	**0.876**	**0.975**	**0.878**	**0.975**	**0.872**	**0.984**

问题的构造方式不适配于多图匹配聚类的场景，所以我们在后续的实验中忽略了这些方法。

非监督学习模型比较结果 我们提出的非监督学习模型 UM3C 的表现较其他方法得到了显著提升：在匹配准确性上，它达到了与其他非学习求解器和非监督学习模型 GANN 相比 6-13% 的提升，以及在聚类准确性上 2-14% 的提升。

与监督学习模型比较结果 我们使用了 NGMv2[44] 和 BBGM[26]作为对比。两个学习模型都在二图匹配的场景下进行学习，对于不同的外点数目我们训练了不同的模型。聚类算法和指标按照与其他算法相同的流程进行。UM3C 的表现显著优于监督模型：在匹配准确性上提升 2-7%，在聚类准确性上提升 2-18%。这都表现了 UM3C 在特征提取和相似度矩阵构造上极好的能力。在此，我们得出两个启示：1）多图聚类匹配问题中，如果在成对的图上进行训练，监督模型缺乏利用其他图信息的机制，因此导致了匹配准确度上的差距。2）M3C 是一个基于聚类问题的求解器，其可以扩大不同类的图相似度之间的差距。BBGM 和 NGMv2

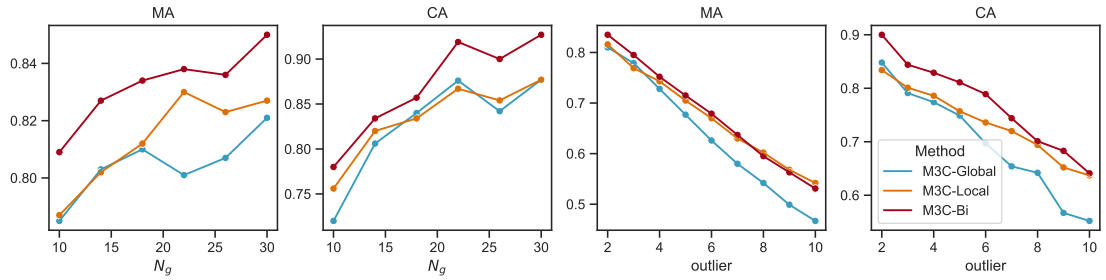


图 4–1 在不同类数、图数场景下对松弛聚类指标三种构建方式的比较。

Figure 4–1 Comparison of three ranking schemes (local, global, bi) by varying number of graphs in each cluster and number of outliers

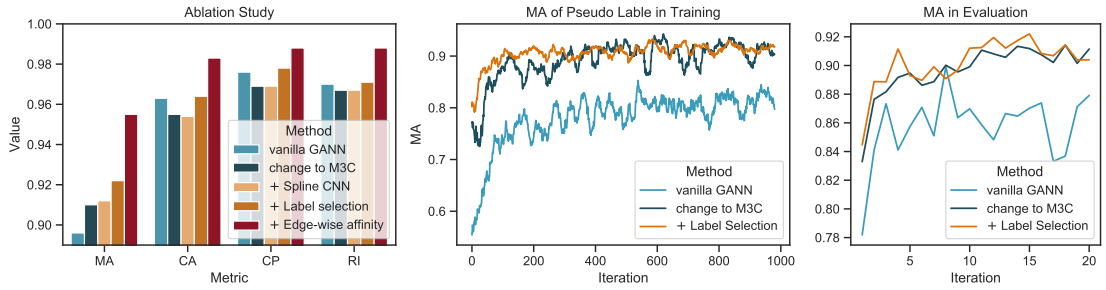


图 4–2 M3C 的消融实验。在 3×8 的场景下，我们依次添加各个组件。左：在匹配和聚类上的测试。右：伪标签在训练过程中按照迭代次数变化的准确性以及对应的测试结果。

Figure 4–2 Ablation study of M3Cby adding components under 3×8 without outlier setting. Left: Evaluation on matching and clustering performance. Right: Quality of pseudo labels while training by iteration and the respective evaluation performance

只着眼于最大化所有图的相似度，而因此会影响聚类准确性。

4.4.2 不同图片和类别数目下实验

这里我们测试算法对不同类数和图数场景下的泛化能力。对于 $N_c = 3$ 的场景，我们使用车、摩托车、酒瓶三类，对于 $N_c = 4$，我们使用车、摩托车、酒瓶、人脸四类。非平衡类的场景下，对于 $N_c = 3$，我们测试了 20 张车、10 张摩托车、5 张酒瓶的场景。GANN 和 UM3C 都在 $N_c = 5, N_g = 10$，没有外点的场景下进行训练。在所有测试中，我们随机产生 2 个外点。

表4–3验证了我们的方法在不同类数和图数上的鲁棒性和先进性。我们的非学习求解器 M3C 保持了与 DPMC 相比有竞争力的性能，在匹配精度上损失 1% 到提升 2%，在聚类精度上提升 2 – 9%。这些提升也反应了我们松弛聚类指标构造方法中的优越性。

UM3C 在所有的指标中均好于其他方法。它在所有设置中保持了 0.97 以上的聚类准确性和 0.87 以上的匹配准确性，与其他方法相比达到了 MA 上 1 – 7% 的提升和 CA 上 2 – 18% 的提升。由于 UM3C 的训练场景和测试场景不同，这也说明了其在图数、类数、外点数上都有很好的泛化能力。此外，即使 UM3C 是在一个较简单的环境下训练的，它依然可以被部署到一个更复杂的场景中，并取得令人满意的结果。

表 4–4 $3 \times 8, 2$ 外点下，谱聚类[37]和多割算法[43]对于非学习求解器的影响。带有 '-MC' 的算法表示使用多割算法作为聚类算法。

Table 4–4 Comparison of Spectral Clustering[37] and Multi-Cut[43] on learning-free solvers under $3 \times 8, 2$ outliers setting. Algorithms with '-MC' use Multi-Cut in clustering.

Method		MA ↑	CA ↑	CP ↑	RI ↑
With Spectral Clustering	MGM-Floyd	0.709	0.567	0.673	0.699
	M3C-Bi	0.687	0.653	0.750	0.758
With Multi-Cut	MGM-Floyd-MC	0.709	0.603	0.716	0.724
	M3C-Bi-MC	0.687	0.634	0.734	0.745

表 4–5 每一轮迭代下超图结构变化。实验在 $3 \times 8, 2$ 外点的场景下进行。对于 M3C，超图结构即为求解的聚类指标，对于 DPMC，超图的结构为求解的最大生成树。这里的数字代表与上一轮相比变化的边的数目。

Table 4–5 Changes in supergraph structure per iteration in $3 \times 8, 2$ outliers setting. For M3C, the structure is the cluster indicator, and for DPMC, the structure is the devised tree. The number is the edges changed per iteration.

Iteration	2	3	4	5	6	7	8	9	10
M3C-Hard	10.48	0.56	0	0	0	0	0	0	0
M3C-Bi	20.44	2.04	1.56	0.24	0.04	0.08	0	0	0
DPMC[20]	10.16	6.16	3.28	1.20	0.48	0.32	0.24	0.24	0.24

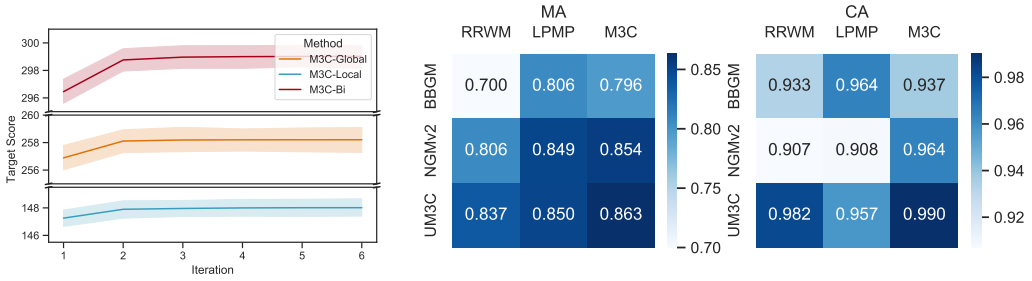
4.5 M3C 与 UM3C 方法分析

4.5.1 松弛聚类指标构造方法比较

我们在不同的场景下测试提出的三种松弛聚类指标构建方法的效果。M3C-Global、M3C-Local、M3C-Bi 分别指代由 global-rank、local-rank 和 bi-rank 构建的方法。我们的测试在固定类数 $N_c = 5$ 的情况下改变每一类中图的数目，固定 $N_c = 5, N_g = 20$ 的情况下改变外点的数目，这两个场景中进行。从图4–1中，我们可见 M3C-Bi 超过所有的方法，所以在前文的测试中，我们主要采用了 bi-rank 的构造方法。这些测试也说明了 global-rank 和 local-rank 也是对松弛聚类指标很好的近似。此外，这些实验也说明了我们的松弛聚类指标构建方式有着很强的鲁棒性和泛化能力。即使在有 10 个外点的情况下，他们依然能够达到 0.5 以上的匹配准确性和 0.65 以上的聚类准确性。同时，随着图数的增加，UM3C 在匹配和聚类精度的表现上都有提高。这也解释了虽然在表4–2（较简单的场景）中，M3C 没能超过其他的非学习求解器，但在表4–3（更复杂的场景）中，能够明显优于其他的非学习求解器。

4.5.2 消融实验

学习模块消融实验 我们在 $N_c = 3, N_g = 8$ 无外点的场景下测试 UM3C 每一个部分的有效性。基线方法通过用 M3C 代替 GANN 中的 GAGM 求解器构造。由于 GANN 没有使用 SplineConv 来优化特征，我们这里测试引入 SplineConv 对于我们结果的影响。我们通过依次加入组件来验证伪标签选择和基于边的相似度学习的有效性。



a) M3C 基于三种松弛聚类指标构建方法的收敛曲线。

b) 学习到的相似度矩阵的泛化性测试。x 轴表示测试求解器，y 轴表示相似度矩阵的学习模型。所有方法都基于 $3 \times 8, 2$ 外点的场景进行学习和测试。

图 4–3 M3C 的算法总体收敛性测试与学习模型的相似度矩阵泛化性测试。

Figure 4–3 Convergence curve of M3C and generalization test of the learned affinity matrix.

对于伪标签学习，图4–2右图展示了训练与测试时每一轮的伪标签的准确率。可见，在训练的早期（前 100 轮迭代），与基线方法相比，有伪标签选择的方法在准确度上有 5% 的提升，可见其能帮助选择离真实值更接近的伪标签。同时伪标签的学习也使得我们的学习过程更加稳定。

对于基于边的相似度学习，图4–2左图显示基于边的相似度学习对我们的学习模型产生了巨大的影响。这也显示了传统的相似度模型的缺陷以及学习基于边的相似度的必要性。SplineConv 对于提高匹配准确性亦有贡献。注意，我们的基线算法已经超过了 GANN，也说明我们的求解器 M3C 的有效性。

聚类方法消融实验 表4–4展示了谱聚类算法[37]和多割算法[43]在为 MGM-Floyd 和 M3C-Bi 聚类的区别。可见，虽然二者的差距略有缩小，但是不同的聚类方法并不会对两者聚类结果的大小关系产生根本上的影响。我们认为，好的聚类结果的根本是寻找好的匹配结果并为聚类产生区分度高的相似度数值矩阵。多割算法和谱聚类算法只是我们在聚类算法中的一类选择，其结果不应该对求解器的效果产生根本性的改变。因此，也为了和前作[20-21]进行更公平的比较，我们在其他的实验中均使用了谱聚类方法作为我们的聚类方法。

4.5.3 收敛性测试

表4–5展示 M3C-Hard，M3C-Bi 和 DPMC 每一轮后超图中结构的变化。在这里，M3C 的结构指代聚类指标（等价于超图的邻接矩阵），而对于 DPMC，其结构指代构造的最大生成树。表中的数值指的是与上一轮相比超图上变化的边的数目 $\sum |\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}|$，其中 $\mathbf{A}$ 为超图的邻接矩阵。

可见，DPMC 的结构来回变化并且最后不能够收敛，M3C-Hard 的结构快速收敛到了次优解。M3C-Bi 有更合理平衡的收敛速率，并且有着最平衡的表现。这些结果都证实了我们在3.4中提出的命题和证明的准确性。

图4–3a展现了算法的总体收敛性表现：可见我们的算法在三种松弛聚类指标构建方法上均可以有效收敛。这也与其超图结构的收敛性相匹配。注意到图中不同的目标函数值是由不同的 r 的选择导致的。

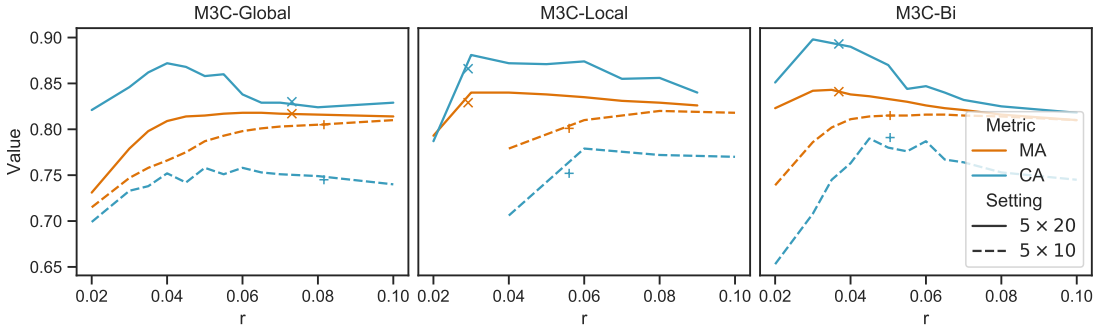


图 4–4 M3C 中超参数 r 在不同松弛聚类指标构建方法中的敏感性测试。实验在 5×20 和 $5 \times 10, 2$ 外点的场景下进行。标记为我们加边到超图连通情况下的 r 的均值。

Figure 4–4 Sensitivity study of the parameter r for three ranking schemes. Experiments are conducted under 5×20 and 5×10, both with 2 outliers, on WillowObject. The marker denotes the performance of our chosen r, which is to add edges until the supergraph is connected.

4.5.4 相似度矩阵泛化性测试

我们对学习到的相似度矩阵进行泛化性测试来说明我们采用的相似度学习的有效性。实验均在 $3 \times 8, 2$ 外点的场景进行学习和测试，求解器采用了 RRWM[10]（同为 M3C 中用来初始化的二图匹配求解器），LPMP[30]（学习模型 BBGM 中采用的求解器）和 M3C，学习模型采用了 BBGM，NGMv2 和 UM3C。对于 UM3C 只有 $\mathbf{K}^{learn}$ 参与了测试。图4–3b显示，UM3C 学习到的相似度在匹配和聚类精度上有着最好的泛化性。BBGM 的学习方法使其学习到的相似度矩阵只能适配于 LPMP 求解器，而损害其他求解器的表现。此外，UM3C 产生的相似度矩阵与 BBGM 相比甚至能够提高 LPMP 的表现。这些都证明了我们基于边的相似度学习的有效性，以及学习到的相似度矩阵的鲁棒性。

4.5.5 超参数敏感性测试

图4–4展示了在 5×20 和 $5 \times 10, 2$ 外点的场景下不同超参数 r 的表现。可见，M3C-Global 的表现在 $r > 0.04$ 和 $r > 0.06$ 下较稳定，而其聚类表现在 $r \geq 0.6$ 的情况下会显著下滑。这意味着我们设置的全局约束需要在一个合理的范围，一味增加边数无法提高我们的算法表现，同时，更少的边也会限制算法优化空间的大小。M3C-Local 和 M3C-Bi 的结果也符合这个规律。如图所示，对于 5×20 它们分别在 $r = 0.15$ 和 $r = 0.03$ 得到最好结果，对于 5×10，分别在 $r = 0.3$ 和 $r = 0.045$ 取到最优。

此外，对于不同的输入和不同的设置，最佳的 r 是不同的，为每个输入找到最佳的 r 过于耗费时间。因此，在实际使用的算法中，我们并不固定一个 r，而是按照约束排序加边，直到超图连通。图4–4中的标记 '×' 和 '+' 是这种方案在 5×20 和 5×10 场景中的平均 r 和各自的平均性能。实验结果也表明，这种方法得到的是最佳 r 的近似值，有助于算法在不进行大量计算的情况下达到接近最佳的性能。这也是我们前文提到的在传统求解器 M3C 和无监督方法 UM3C 中使用的方法。

4.6 时间复杂度分析

与 DPMC[20]和传统 MGM-Floyd[19]相比，M3C 和 UM3C 的速度提升来自于更低的复杂度下限。令 N 和 n 分别表示图和节点的数量，计算相似度数值矩阵的时间复杂度为 $O(N^2n^3)$，构造超图的时间复杂度为 $O(N^2\log N^2)$。在最坏的情况下，我们会在超图上添加 $\frac{N(N-1)}{2}$ 条边，此时 MGM-Floyd 的复杂度为 $O(N^3n^3)$。然而，添加的边数期望值要远小于最坏情况。事实上，在大多数情况下，它小于 $\frac{N_cN_g(N_g-1)}{2}$，其中 N_c 表示聚类数量，N_g 表示同一类最大图数。同时，在执行我们的算法时不需要计算循环一致性也降低了我们的时间成本。综上，我们的实际运行时间明显少于 DPMC 和传统 MGM-Floyd。这样的改进也导致了我们的学习模型只需要更少的训练和推论时间。

4.7 本章小结

本章主要介绍了我们的多图匹配聚类求解器 M3C 和非监督学习模型 UM3C 的实验结果。我们在 Willow-ObjectClass 上全面地测试了我们提出求解器和模型的性能：我们通过与其他求解器、非监督学习模型、监督学习模型在多图聚类匹配问题，不同图数、类数的场景下的各类实验，验证了我们方法的有效性。同时，我们还结合消融实验、松弛聚类指标构造方案比较、收敛性测试、相似度矩阵泛化性测试、超参数敏感性测试这五类实验，对我们的求解器和学习模型的各个部分进行了细致而综合的评价，并同时分析了各种算法优劣所在。这些实验展现了我们提出方法的收敛性、高鲁棒性、灵活优化结构的特点，证实了我们的算法和模型实现了前文提出的设计目标。

第五章　基于中间图的图匹配模型

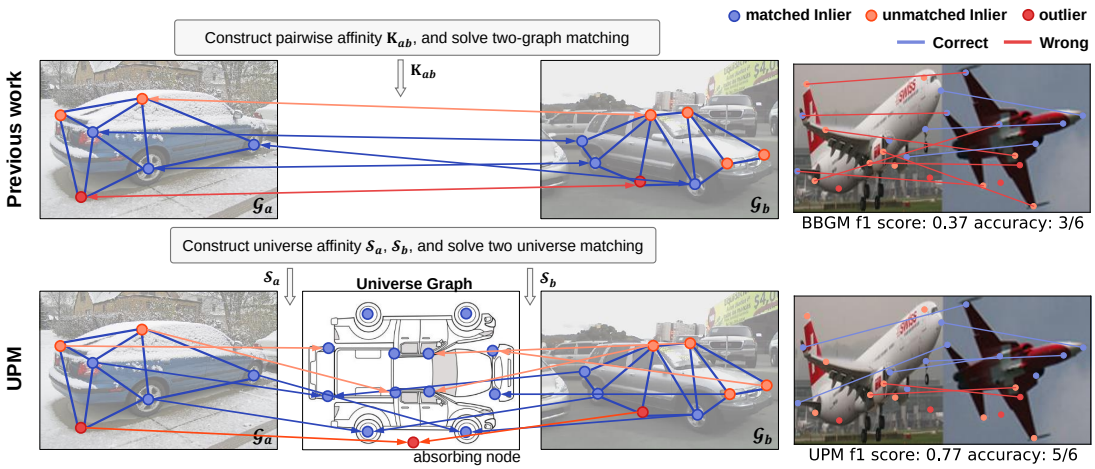


图 5–1 基于中间图的部分图匹配学习模型（Universe Partial Matching, UPM）预览以及与传统二图匹配学习模型 BBGM 的对比。和顶部传统的二图匹配模型相比，带有语义信息的中间图锚点与一个吸收锚点组成了整个中间图，并且作为桥梁为部分图匹配模型 UPM 服务。所有的图都首先与中间图进行匹配，然后通过中间图重建二图匹配结果。右侧展现的是二者分别对应的 BBGM 和 UPM 在实际表现中的区别。

Figure 5–1 Teaser for our method

Universe Partial Matching (UPM), a universe graph based partial matching solver, and comparison between matching instances of BBGM and UPM. Compared with the pairwise matching model on the top, a universe graph with all semantic inliers and an absorbing node is introduced as a bridge for matching in UPM. All the graphs are first matched to the universe graph and then pairwise matching is reconstructed via universe graph. We show the performance difference between BBGM and UPM on real-world data on the right.

5.1 中间图视角设计动机

中间图的概念在前作[18, 32]中有提出，其实际为一张虚拟的图，但所有输入图中的点都可以在中间图上找到对应的节点。在我们的设计中，增加了一个吸收节点用于与所有的外点进行匹配。我们首次将中间图引入深度学习图匹配模型，主要有以下两个原因：1）中间图可以汇总训练数据集中的所有图的信息，从而为点和点的区分提供一个全面的视角；2）中间图统一了二图匹配和多图匹配的流程，这使得我们可以为所有类型的图匹配问题引入一个统一的模型。

5.1.1 全局视角下的节点区分

我们首先回顾在基于二图匹配的学习模型[23, 26-27, 44]中遇到的节点类型。在一个二图匹配实例中，节点是否冗余是由一个二图对应关系（Pairwise graph correspondence, PGC）来决

表 5–1 部分图匹配中不同节点类型划分。

Table 5–1 Categorization of different nodes for matching.

	匹配内点（Matched Inlier）	未匹配内点（Unmatched Inlier）	外点（Outlier）
语义信息	✓	✓	×
二图匹配	✓	×	×

定的。PGC 代表一个节点是否在图对的另一张图中是否有匹配关系。给定一对图 $(\mathcal{G}_a, \mathcal{G}_b)$ 和其真值匹配 $\mathbf{X}_{ab}^{gt}$，点 $v_i \in \mathcal{G}_a$ 的 PGC 为：

$$\mathrm{PGC}(v_i|\mathcal{G}_a, \mathcal{G}_b, \mathbf{X}_{ab}^{gt}) = \begin{cases} 1, & \sum_{v_j \in \mathcal{G}_b} \mathbf{X}_{ab}^{gt}(v_i, v_j) = 1 \\ 0, & \sum_{v_j \in \mathcal{G}_b} \mathbf{X}_{ab}^{gt}(v_i, v_j) = 0 \end{cases} \tag{5–1}$$

然而，这种点类型的划分是粗糙的，并且忽略了冗余点中语义信息的不同，因此，我们需要从另一个视角来重新定义点的类型。

在前作[26-27]中，匹配的节点被聚类在一起而冗余节点会在特征空间中被排除。然而，我们发现，节点的类型会在不同的情况下发生变化。例如，在一个实例中两个匹配上的点有可能在另一个实例中变成冗余节点。因此，这两种情况下，学习流程中的梯度方向可能完全不同，甚至互相冲突。同时，外点总是冗余的，并且会永远被排除在学习流程之外。因此，训练结束后，他们会均匀地分散在特征空间中，并且很容易与其他点混淆。事实上，我们认为外点应该要被聚类在一起处理以得到更好的结果。总而言之，不同类型的节点在匹配时需要不同的策略，因此我们提出了更细粒度的对冗余节点的分类。

我们将节点分成三类：匹配内点（Matched Inlier），未匹配内点（Unmatched Inlier），和外点（Outlier），其中后两种均为我们所称的冗余节点。未匹配内点指代的是有语义信息，但是在二图匹配中没有对应点的节点，而外点指的是由随机因素导致的无意义的冗余节点。三者的分类见表5–1。在这里，我们根据内涵的语义信息对冗余节点分别划分成未匹配内点和外点。而我们对内涵语义信息的定义基于这样一个假设：具有特定语义信息的节点会重复出现，而通过随机错误得到的无意义的节点是唯一的。也就是说，只要我们的数据集足够大，具有内涵语义信息的节点就一定会与其他节点有对应关系，而无意义的外点则永远是独立存在的。因此我们提出了多图对应关系（Multiple graph correspondence, MGC）来定义个节点是否具有内涵语义信息：

$$\mathrm{MGC}(v_i|\mathcal{G}_a, \mathbb{G}, \mathbb{X}^{gt}) = \begin{cases} 1, & \sum_{\mathcal{G}_b \in \mathbb{G}} \sum_{v_j \in \mathcal{G}_b} \mathbf{X}_{ab}^{gt}(v_i, v_j) > 0 \\ 0, & \sum_{\mathcal{G}_b \in \mathbb{G}} \sum_{v_j \in \mathcal{G}_b} \mathbf{X}_{ab}^{gt}(v_i, v_j) = 0 \end{cases} \tag{5–2}$$

其中，$\mathbb{G}$ 代表训练集中所有的图，$\mathbb{X}^{gt}$ 代表所有真实匹配结果。

值得注意的是，MGC 是在多图匹配的视角下基于数据集中所有的图 $\mathbb{G}$ 和其真实匹配 $\mathbb{X}^{gt}$ 定义的。因此，二图匹配的学习模型并不能很好地区分两者。这也是为什么我们将中间图的概念引入到机器学习方法中，来聚合训练数据中所有图的全局信息。如图5–1中所示，不同的节点类型在匹配过程中有着不同的表现，由此可见，不同类型的节点通过中间图可以很好地被区分出来。此外，中间图为输入的图提供了一个模板，输入图中的每一个点都被鼓励和中间图中的锚点相匹配，也因此避免了前作中不同匹配实例之间梯度冲突的影响。

5.1.2 图匹配的统一模型

如前文介绍，图匹配中有许多类型的任务，包括二图匹配、多图匹配、多图匹配聚类、增量式匹配等。由于中间图的优点，所有的问题都可以通过我们提出的模型 **UPM** 解决。

二图匹配和多图匹配之间根本性的差别在于循环一致性[16, 18]，也就是说，根据不同图匹配关系得到的匹配应该是相同的。在我们的方法中，中间图作为一个中心图，所有输入的图都会和中间图首先进行匹配，然后再从与中间图的匹配中重建出二图匹配结果。因此，循环一致性在这一中心化的结构中被自动满足。也就是说，二图匹配和多图匹配在中间图的视角下得到统一。

在线多图匹配问题，也被称为增量式多图匹配问题，在前作[33]中提出。在实践中，参与匹配的图往往不是一次性获得的，而是随着时间的推移而手机的，例如，街景车拍摄的照片，监控摄像头拍摄的视频，新发现的蛋白质与现有的蛋白质。在这种情况下，将所有增加的图都与原来的图进行一遍匹配是低效的，如果需要循环一致性的参与，将所有的图每次进行后续处理更是极度耗费时间。而对于中间图而言，只要在中间图和新加入的图中进行一次匹配，我们就可以得到整体的匹配结果，这就将时间复杂性从线性级降到了常数级。同时，这样的方法还具有更低的空间复杂性。由于二图匹配结果可以通过中间图的匹配结果进行重建，我们只需要保留后者即可。这样，空间复杂度也由平方级降到了线性。

另一类图匹配问题是我们前文提到的多图匹配聚类问题[20]。大多数匹配问题假设所有参与匹配的图来自于同一类，并且拥有相同的内点，这种设定过于理想。多图匹配聚类问题提出了考虑不同类别的图作为输入的情况。而我们止痒将属于不同类别的图的节点之间不匹配，多图匹配聚类问题就可以转化为一种部分匹配问题。由于中间图可以区分节点类型，从而拒绝来自不同类的节点的配对，因此，中间图也能够应对多图匹配聚类问题。

总而言之，中间图视角下的匹配对于所有任务都是鲁棒而高效的。所有种类的匹配任务可以在中间图视角下被联系起来。我们提出的 **UPM** 模型可以被视为图匹配问题的统一模型。

5.2 中间图相似度学习

5.2.1 中间图相似度构造

根据前作[55]中提出的相似度矩阵分解方法，之前的工作[22, 25-26] 构建相似度 $\mathcal{S}$：

$$\mathcal{S}_{ab} = \mathbf{F}_a \mathbf{\Lambda} \mathbf{F}_b^\top \tag{5–3}$$

其中 $\mathbf{F}_a \in \mathbb{R}^{n_a \times d}, \mathbf{F}_b \in \mathbb{R}^{n_b \times d}$ 分别表示图 $\mathcal{G}_a, \mathcal{G}_b$ 的特征，$\mathbf{\Lambda}$ 表示相似度的可学习权重。这种相似度构造方法注重于一对图之间的相似性，也仅适合完全匹配的设置。对于部分图匹配，大量的外点会破坏图之间的几何相似性，从而导致特征提取之间的巨大差距。在完全匹配中被遵循的图的相似性会在部分图匹配中被大幅破坏，从而导致相似度矩阵质量的下降。

在本文中，我们提出一种新的相似度学习方法，称为中间图相似度（Universe affinity），来克服部分图匹配问题带来的挑战。具体来说，我们构造了一个带有中间图特征锚点 $\{\mathbf{F}_u^{(1)} \dots \mathbf{F}_u^{(n_u)}\}$ 的中间图 $\mathbf{U}$，并尝试学习单个图和中间图之间的相似度 S_a。

$$\mathcal{S}_a = \mathbf{F}_a \mathbf{\Lambda}_u \mathbf{F}_u^\top \tag{5–4}$$

其中 $\mathbf{\Lambda}_u$ 和 $\mathbf{F}_u$ 在所有匹配实例之间共享权重。令 $\mathbf{M} = \mathbf{\Lambda}_u \mathbf{F}_u^\top$ 为可学习权重，我们有：

$$\mathcal{S}_a = \mathbf{F}_a \mathbf{M} \tag{5–5}$$

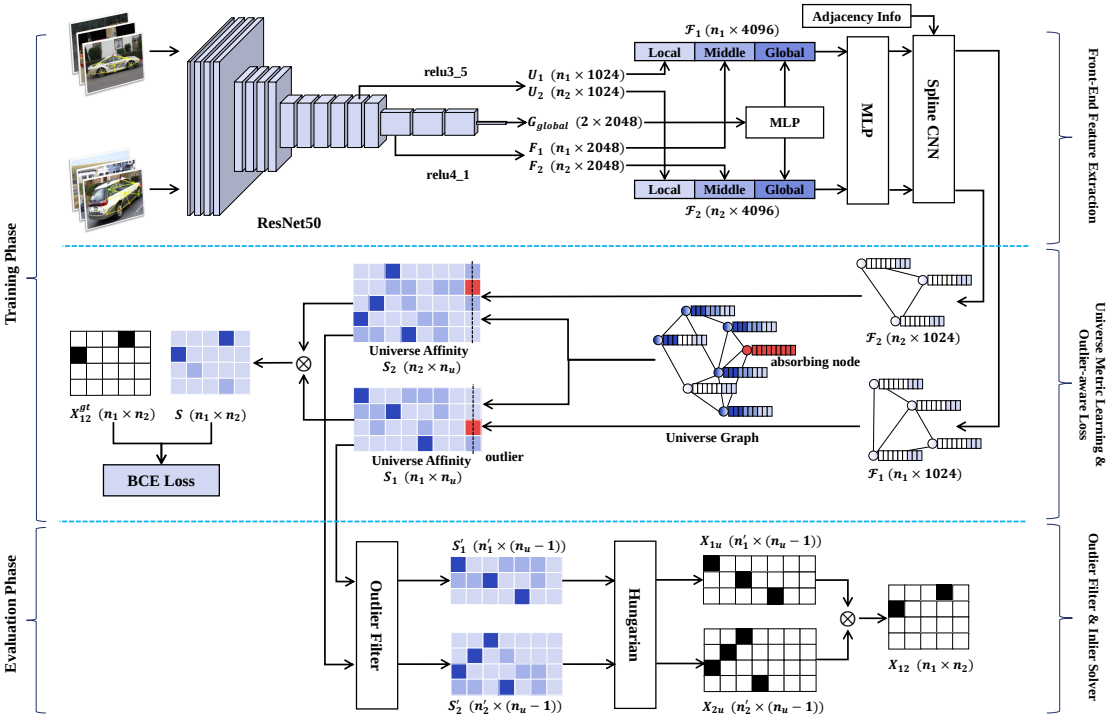


图 5–2 UPM 的总体学习流程。整个流程主要分成三个部分：前端特征提取、中间图相似度学习、中间图匹配求解器。在前端特征提取中，我们为每一个节点，使用 ResNet50[61]提取局部、中间、全局特征，并将他们输入 SplineConv[47]来获得最终的点特征。在中间图相似度学习中，我们构建输入图和中间图之间的中间图相似度，并使用外点感知的损失函数来训练中间图，具体方法在5.2和5.3中介绍。在测试阶段，我们使用外点过滤器和内点求解器来从连续解中求解离散解。外点过滤器会去除求得相似度矩阵的最后一列和对应外点所在行，内点求解器使用匈牙利算法[62]来计算输入图和中间图之间的匹配结果，最后从中间图匹配结果中重建二图匹配结果。

Figure 5–2 Overall pipeline for UPM. The pipeline consists of three parts: front-end feature extraction, universe metric learning, and universe matching solver. In front-end feature extraction, we extract local, middle, global features for each node with ResNet50[61] backbone and input them to SplineConv[47] to obtain node feature. In the universe metric Learning part, we build universe affinity between input graphs and universe graph respectively and train the universe graph with an outlier-aware loss, which will be discussed in detail in Sec. 5.2 and Sec. 5.3. The red nodes in universe is absorbing node and red element in matrix illustrates that it is an outlier. In the evaluation phase, outlier filter and inlier solver are adopted to obtain a discrete solution. Outlier filter removes the last column and the rows of outliers. Inlier solver applies the Hungarian algorithm[62] to obtain universe matching results. The pairwise matching is reconstructed from universe matching.

这等价于对特征进行一次线性变换。我们还增加了 `batchnorm` 和 `relu` 层来增加相似度学习中的非线性性，这样可以提取更高质量的中间图相似度。

与从单个源图中提取的节点特征相比，中间图上锚点特征 $\mathbf{F}_u$ 汇聚了训练数据中所有图的信息，因此对节点和结构噪声的鲁棒性更强。因此，由中间图锚点特征构造的中间图相似

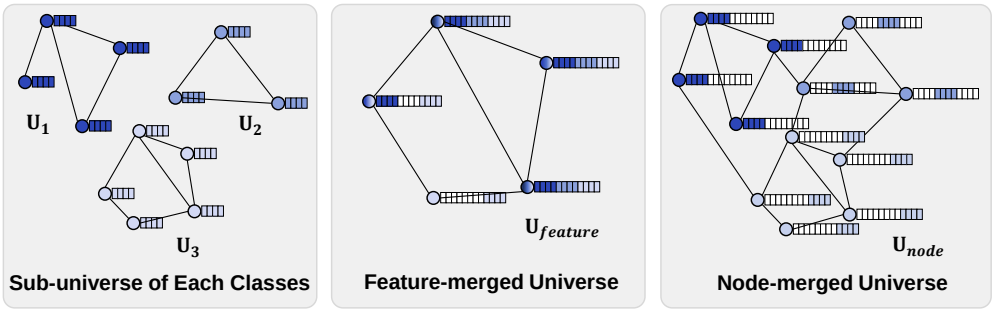


图 5–3 中间图的两种构造方法说明。如第一张图所示，假设有三个子中间图，其节点和特征由不同颜色表示。第二、三张图展示了中间图是如何将这些子中间图进行融合的。中间图特征中空白的矩形表示中间图锚点特征 $\mathbf{F}_i$ 中被浪费的通道。基于特征的中间图锚点中多种颜色表示子中间图中的几个节点被聚合在一起。我们意在展现点特征是如何合并在一起的，边和邻接关系在这里没有意义。

Figure 5–3 Illustration of two ways of universe graph construction. There are three sub-universe graphs with nodes and features of different colors. The second and the third graph show how universe graph merges these sub-universe graphs. The blank rectangles in universe graph represent the wasted channel of the universe anchor feature $\mathbf{F}_i$. The color of nodes in the feature-merged universe graph denotes that several nodes of sub-universe graphs are aggregated together. Note we aim to show the concatenation on node features. Edge features are ignored and the adjacency is meaningless.

度更加可靠。中间图上锚点特征将潜在的特征空间分成不同的部分，并鼓励同一部分的节点特征进行匹配。

同时，与二图的相似度相比，中间图相似度并不会缺少信息，因为二图的相似度可以很容易地通过中间图相似度重建：

$$\mathcal{S}_{ab} = \mathbf{F}_a \mathbf{\Lambda} \mathbf{F}_b^\top = \mathbf{F}_a \mathbf{M}\mathbf{M}^\top \mathbf{F}_b^\top = \mathcal{S}_a \mathcal{S}_b^\top \tag{5–6}$$

虽然中间图相似度可以同时应用于节点和边特征，我们在实际的学习过程中放弃了边特征的学习，主要基于以下三点考量：1）由于部分匹配中存在大量外点，两个图的结构相似性被严重破坏。基于边的相似度学习很难提升匹配准确性，甚至会损害匹配结果。2）由于使用了 SplineConv 对点特征进行优化，我们学习到的点特征中已经承载了几何信息。3）中间图上的边数量极其庞大，特别是对于基于节点的中间图（见5.2.2介绍）。基于边的相似度学习会非常不稳定并且难以收敛。综上，本文中，我们仅着眼于部分图匹配问题中的点相似度的学习。

5.2.2 中间图结构分析

图匹配问题意在探索不同事物之间的图相似性，因此我们希望同一个模型能够解决不同类别之间的匹配，例如，我们希望通过一个模型，就能同时解决对于汽车的匹配和对于船的匹配。但是，对于每一个不同的图类，它们都会产生一个属于自己的中间图和对应的锚点特征。因此，这要求我们进一步探索这些来自不同类别的图的中间图如何融合在一起。

为了便于讨论，我们把每个类的中间图称为子中间图 $\mathbf{U}_i$，其中间图锚点特征记作 $\{\mathbf{F}_i^{(1)}, \ldots, \mathbf{F}_i^{(n_i)}\}$，$\mathbf{F}_i^{(j)} \in \mathbb{R}^d$ 代表节点 $v_j \in \mathbf{U}_i$ 的特征，n_i 代表的锚点个数。此时，中间图

$\mathbf{U}$ 指所有子中间图的融合图，其表示方法为：$\{\mathbf{F}_u^{(1)}, \ldots, \mathbf{F}_u^{(n)}\}$。注意我们只关心中间图中的点特征，而在讨论中忽略其边特征和邻接关系。

我们针对中间图的结构提出两种假设：基于特征的中间图和基于节点的中间图，如图5–3中所示。

基于特征的中间图 令 n 表示最大的子中间图大小，$n = \max n_i$，其中中间图锚点特征 $\mathbf{F}_u^{(i)}$ 聚合了每个子中间图中第 i 个节点特征：

$$\mathbf{F}_u^{(i)} = \mathtt{CONCAT}(\mathbf{F}_1^{(i)}, \ldots, \mathbf{F}_m^{(i)}) \tag{5–7}$$

$\mathbf{U}$ 中的特征数 n 随着图类数的增加而保持不变。总特征的维度有一个固定的上限控制其最大表达能力（往往受到前端模型的限制）。因此，这类中间图构造方法在时间和空间上的效率更高，更适用于有很多图类或图类之间关系复杂的情况。

基于节点的中间图 令 n 表示所有子中间图的节点数之和 $n = \sum n_i$，中间图的整体 $\mathbf{U}$ 聚合了所有子中间图的特征：

$$\mathbf{U} = \bigcup_i \mathbf{U}_i = \bigcup_i \{\mathbf{F}_i^{(1)}, \ldots, \mathbf{F}_i^{(n_i)}\} \tag{5–8}$$

这种中间图的构造方法对不同的图类更为敏感。其可以拒绝来自不同类别的两张图之间的匹配，所以更适用于多图匹配聚类这一场景。其缺点是，它的中间图往往会随着图类的数目 m 的增大而增大。此外，由于所有的特征是从共享权重的前端模型得到，其特征的维度依然和输入的图特征保持一致，也就意味着有很多的通道被浪费了。如图5–3中右图所示，其特征维度依然与之前的相同（为 9），而实际被使用的只有 3 个通道，因此在基于节点的中间图中有 6 个通道的浪费。

在实践中，我们可以通过改变中间图的大小和相关的训练规则来改变这两种结构。对于基于特征的中间图，我们可以估计所有类别的最大特征维度，并对同一类别的训练对进行采样。对于基于节点的中间图，我们估计所有类别的特征数之和，并且将一半的训练数据设置成不同类型的图匹配，而对应的匹配真值设为全不匹配。特征通道的分配通过模型本身自适应地学习，我们只把特征维度设为常数。在实验章节，我们证明中间图的构造在两种情况下均符合我们的猜想。

5.3 外点聚类的损失函数设计

一个很自然的想法是，外点需要被聚类并且一起处理以得到更好的匹配结果。然而，我们碰到的主要的挑战在于，外点总是没有对应的匹配点，也因此缺乏有效的指导使他们聚集在一起。因此，在前作中，外点会在匹配的过程中被排斥，并且随机分散在特征空间，影响最终的匹配结果。

在 UPM 中，我们在中间图上提供了一个吸收节点作为外点的对应锚点，并且设计了一个外点感知的损失函数来鼓励外点与该锚点匹配。在这一节中，我们会先介绍基础的损失函数，然后将其改造成外点感知的损失函数。

算法 5–1 基于中间图的部分匹配模型 (Universe Partial Matching, UPM)

Input: 图对 $\{\mathcal{I}_1, \mathcal{I}_2\}$, 点集 $\{\mathcal{V}_1, \mathcal{V}_2\}$

从 ResNet50 中通过 `relu3_5`, `relu4_1` 和 `final` 层提取特征:
$\mathcal{U}_{local}, \mathbf{F}_{middle}, \mathbf{G}_{global} \leftarrow \texttt{ResNet}(\mathcal{I}_1, \mathcal{I}_2, \mathcal{V}_1, \mathcal{V}_2)$;

点特征提取: $\mathbf{F}_1, \mathbf{F}_2 = \texttt{MLP}(\mathcal{U}_{local}, \mathbf{F}_{middle}, \texttt{MLP}(\mathbf{G}_{global}))$;

通过 SplineConv 和图结构优化点特征:
$\mathbf{F}_1 = \texttt{SplineConv}(\mathbf{F}_1, \mathcal{A}_1), \mathbf{F}_2 = \texttt{SplineConv}(\mathbf{F}_2, \mathcal{A}_2)$;

对每张图计算中间图相似度:
$\mathcal{S}_1 \leftarrow \texttt{SoftMax}(\texttt{MLP}(\mathbf{F}_1)), \mathcal{S}_2 \leftarrow \texttt{SoftMax}(\texttt{MLP}(\mathbf{F}_2))$;

if 对比学习特征优化 **then**

  将学习特征投射到更低维空间: $\mathbf{F}'_1 = \texttt{Projector}(\mathbf{F}_1), \mathbf{F}'_2 = \texttt{Projector}(\mathbf{F}_2)$;

  计算对比损失函数: $\mathcal{L}_{con} = \frac{1}{2n}\sum l(i,j)$, 其中 $l(i,j)$ 由方程5–22求得

if 训练阶段 **then**

  去除最后一列以处理外点: $\mathcal{S}'_1 \leftarrow \mathcal{S}_1(:, :n-1), \mathcal{S}'_2 \leftarrow \mathcal{S}_2(:, :n-1)$;

  计算匹配损失函数: $\mathcal{L} = \texttt{BCELoss}(\mathcal{S}'_1(\mathcal{S}'_2)^\top, \mathbf{X}_{gt}) + \gamma\mathcal{L}_{con}$

else if 测试阶段 **then**

  去除最后一列以及对应外点行:
  $\mathcal{S}'_1 \leftarrow \texttt{OutlierFilter}(\mathcal{S}_1(:, :n-1))$,
  $\mathcal{S}'_2 \leftarrow \texttt{OutlierFilter}(\mathcal{S}_2(:, :n-1))$;

  计算与中间图之间匹配: $\mathbf{X}_1 \leftarrow \texttt{Hungarian}(\mathcal{S}'_1), \mathbf{X}_2 \leftarrow \texttt{Hungarian}(\mathcal{S}'_2)$;

  将外点行设成 0 并利用 $\mathbf{X}_1, \mathbf{X}_2$ 重建二图匹配结果: $\mathbf{X}_{12} \leftarrow \texttt{Fill}(\mathbf{X}_1)\texttt{Fill}(\mathbf{X}_2^\top)$

Output: 二图匹配 $\mathbf{X}_{12}$

5.3.1 基础损失函数设计

机器学习的图匹配问题可以看做最大化与真值匹配相似度的过程[27]。给定一批训练样本 $\{\mathbf{X}_{ab}^{(l)}, G_a^{(l)}, G_b^{(l)}\}, l = 1, 2, ..., N$，机器学习的图匹配问题目标是最大化相似度:

$$\arg\max_{\theta} \prod_{l} P_\theta(\hat{\mathbf{X}}_{ab} = \mathbf{X}_{ab}^{(l)} | \mathcal{G}_a^{(l)}, \mathcal{G}_b^{(l)}), \tag{5–9}$$

其中 $\hat{\mathbf{X}}_{ab}$ 表示模型预测的匹配。如果我们放松对 $\hat{\mathbf{X}}_{ab}$ 的置换矩阵约束，则可以将相似度分解到点层面:

$$P_\theta(\hat{\mathbf{X}}_{ab} = \mathbf{X}_{ab} | \mathcal{G}_a, \mathcal{G}_b) = \prod_{ij} P_\theta(\hat{x}_{ij} = x_{ij} | \mathcal{G}_a, \mathcal{G}_b). \tag{5–10}$$

$\hat{x}_{ij}$ 和 x_{ij} 分别表示 $\hat{\mathbf{X}}_{ab}$ 和 $\mathbf{X}_{ab}$ 的每一行，也就是点 v_i 和 v_j 的匹配相似度。注意，$\mathbf{X}_{ab}$ 是符合匹配要求的匹配真值，而预测的结果 $\hat{\mathbf{X}}_{ab}$ 会受到 $\mathbf{X}_{ab}$ 的指导，因此也满足匹配约束。

令 $p_{ij}^{(k)} = P_\theta(\hat{x}_{ij}^{(k)} = 1)$ 表示 v_i 和 v_j 的匹配概率，我们可以进一步改写优化目标:

$$\begin{aligned}
&\arg\max_{\theta} \prod_{k} P_\theta(\mathbf{X}^{(k)} | \mathcal{G}_a^{(k)}, \mathcal{G}_b^{(k)}) \\
=&\arg\max_{\theta} \sum_{k}\sum_{ij} \log_\theta(\hat{x}_{ij}^{(k)} = x_{ij}^{(k)}) \\
=&\arg\max_{\theta} \sum_{k}\sum_{ij} x_{ij}^{(k)} \log(p_{ij}^{(k)}) + (1 - x_{ij}^{(k)})(1 - \log(p_{ij}^{(k)})),
\end{aligned} \tag{5–11}$$

其形式与二元交叉熵（Binary cross entropy, BCE）一致，因此，我们仿照前作[23]将 BCE 损失函数引入机器学习的图匹配问题中：

$$\mathcal{L} = \sum_{ij} \mathcal{L}_{ij} = \sum_{ij} \mathrm{BCE}(p_{ij}, x_{ij}). \tag{5–12}$$

对于二图匹配对 $(\mathcal{G}_a, \mathcal{G}_b)$，我们通过中间图定义匹配概率 p_{ij}^{ab}。令 p_{ik}^{a} 代表图中点 $v_i \in \mathcal{G}_a$ 和中间图 $\mathbf{U}$ 上锚点 v_k 之间的中间图匹配概率，其可以计算为：

$$p_{ij}^{ab} = \sum_{v_k \in \mathbf{U}} p_{ik}^{a} p_{jk}^{b} = (p_i^a)^\top p_j^b, \tag{5–13}$$

其中 p_i^a (p_j^b) 是中间图和点 v_i (v_j) 之间的中间图匹配概率分布，由对中间图相似度 $\mathcal{S}^a, \mathcal{S}^b$ 进行归一化（`softmax`）操作后得到。

我们接下来计算反传的梯度，来对 `BCE` 损失函数对中间图相似度学习的影响：

$$\frac{\partial \mathcal{L}_{ij}}{\partial \mathcal{S}_{it}^a} = \begin{cases} \frac{p_{it}^a}{p_{ij}^{ab}} \cdot \sum_k p_{ik}^a (p_{jk}^b - p_{jt}^b), & x_{ij} = 1; \\ \frac{p_{it}^a}{p_{ij}^{ab}} \cdot \sum_k p_{ik}^a (p_{jt}^b - p_{jk}^b), & x_{ij} = 0. \end{cases} \tag{5–14}$$

此外 v_i 上的梯度之和应为 0：

$$\sum_t \frac{\partial \mathcal{L}_{ij}}{\partial \mathcal{S}_{it}^a} = \pm \sum_t p_{it}^a \mp \frac{1}{p_{ij}^{ab}} \sum_t p_{it}^a p_{jt}^b = 1 - 1 = 0. \tag{5–15}$$

梯度的符号由 $\sum_k p_{ik}^a (p_{jk}^b - p_{jt}^b)$ 决定。

因而：

- 当匹配真值 $x_{ij} = 1$，如果 p_{jt}^b 大，$\mathcal{S}_{it}^a$ 会增加，如果 p_{jt}^b 小，$\mathcal{S}_{it}^a$ 会减少。也就是说，损失函数 $\mathcal{L}_{ij}$ 鼓励匹配概率 p_i^a 和 p_j^b 越来越接近。
- 当匹配真值 $x_{ij} = 0$，所有的梯度都会相反，那么 p_i^a 和 p_j^b 会趋向于分离。

5.3.2 外点感知的损失函数设计

然而基础的 `BCE` 损失函数并不很很好地处理外点，因为外点 v_i 和 v_j 之间没有直接的匹配关系，中间图匹配相似度 p_i^a 和 p_j^b 之间也互相排除。在这种情况下，外点很难被聚类并匹配到中间图的吸收锚点上。为了解决这一问题，我们通过去除点匹配概率 p_{ij}^{ab} 的最后一列（外点列）来定义节点的部分匹配概率：

$$p'_{ij} = \sum_{k=1}^{n-1} p_{ik}^a p_{jk}^b, \tag{5–16}$$

此时损失函数变为：

$$\mathcal{L}' = \sum_{ij} \mathrm{BCE}(p'_{ij}, x_{ij}) = \sum_{ij} \mathrm{BCE}(\sum_{k=1}^{n-1} p_{ik}^a p_{jk}^b, x_{ij}) \tag{5–17}$$

在新的损失函数设计中，$t \neq n$ 的梯度 $\frac{\partial \mathcal{L}'_{ij}}{\partial \mathcal{S}_{it}^a}$ 不会有很大的变化：

$$\frac{\partial \mathcal{L}'_{ij}}{\partial \mathcal{S}_{it}^a} = \begin{cases} \frac{p_{it}^a}{p_{ij}^{ab}} \cdot [\sum_{k=1}^{n} p_{ik}^a (p_{jk}^b - p_{jt}^b) - p_{jn}^b p_{in}^a], & x_{ij} = 1 \\ \frac{p_{it}^a}{p_{ij}^{ab}} \cdot [\sum_{k=1}^{n} p_{ik}^a (p_{jt}^b - p_{jk}^b) + p_{jn}^b p_{in}^a], & x_{ij} = 0 \end{cases} \tag{5–18}$$

其符号由 $\sum_k p_{ik}^a(p_{jk}^b - p_{jt}^b) - p_{jn}^b p_{in}^a$ 定义，并且前文所有的结论依然成立。当 $t = n$，梯度变为：

$$\frac{\partial \mathcal{L}'_{ij}}{\partial \mathcal{S}_{in}^a} = \begin{cases} p_{in}^a p'_{ij}/p_{ij}^{ab}, & x_{ij} = 1 \\ -p_{in}^a p'_{ij}/p_{ij}^{ab}, & x_{ij} = 0 \end{cases} \tag{5–19}$$

对于外点，其永远不会有匹配上的对应点，因此 x_{ij} 始终为 0。它们的梯度总是负数，其与中间图上吸收锚点的相似度永远不会下降。

对于内点，其基于中间图的相似度分布 p_i 有两个峰值：中间图上正确的匹配锚点和吸收锚点，对于其他的锚点，值应当接近于 0。因此，$\mathcal{S}_{in}$ 的梯度可以拓展为：

$$-p_{in}^a p'_{ij}/p_{ij}^{ab} = -p_{in}^a \frac{\sum_{k=1}^{n-1} p_{ik}^a p_{jk}^b}{\sum_{k=1}^{n-1} p_{ik}^a p_{jk}^b} \tag{5–20}$$

当真值匹配 $x_{ij} = 0$ 时，只有 $p_{in}^a p_{jn}^b$ 相对较大，而其他项可以被忽略不计，并且梯度接近于零。另一方面，当 $x_{ij} = 1$，其与中间图上正确锚点和吸收锚点的相似度项都被激活，因此 p'_{ij}/p_{ij}^{ab} 会是一个常数（而不是零）。总而言之，正确匹配（$x_{ij} = 1$）的梯度总是大于负匹配（$x_{ij} = 0$），这也就一定程度上地避免了内点与中间图上吸收锚点的匹配。

综上，通过删除节点匹配相似度分布的最后一列（外点列），我们的损失函数可以区分内点和外点。因此外点可以被聚类到中间图上的吸收锚点，即使他们没有匹配真值的标签作为引导。这些设计均有效地帮助内点和外点的学习。总体算法流程见算法5–1。

5.4 对比学习视角的特征优化

基于中间图匹配的视角，图匹配问题中的正匹配和负匹配天然与对比学习暗含对应关系。事实上，从这种点匹配的视角来看，我们可以把正匹配（$x_{ij} = 1$）视作对比学习中的正样本，把负匹配（$x_{ij} = 0$）视作对比学习中的负样本，我们可以很容易地将图匹配问题中点特征的优化转换成对比学习问题。于是，从这种视角出发，我们借用 SimCLR[63]中的对比学习方法，来对我们学习过程中的点特征进行进一步的优化。

在经过特征提取器后，我们仿照前作[63-64]，使用一个投射器，将点特征投射到低维的特征空间：

$$\mathbf{F}'_a = \texttt{Projector}(\mathbf{F}_a) = \mathbf{W}_b(\sigma(\mathbf{W}_a\mathbf{F}_a)) \tag{5–21}$$

我们实际采用一层 `MLP` 作为此处的投影器。

根据前文的观察，我们把匹配的点对视作正样本。由于 SimCLR 对于数据批量大小的要求，在图对 $(\mathcal{G}_i, \mathcal{G}_j)$ 中对于一对匹配上的点 $(v_i, v_j), v_i \in \mathcal{V}_i, v_j \in \mathcal{V}_j$，对于 v_i，我们把 v_j 视作其正样本，把 $\mathcal{G}_j$ 中所有没与 v_i 匹配上的点，与 $\mathcal{G}_i$ 中除 v_i 外的所有点，视为 v_i 的负样本，对于 v_j 同理。此时，对于该点对，我们有其对比学习损失函数：

$$\begin{aligned} l(i,j) = -\Bigg(&\log \frac{\exp(s_{ij}/\tau)}{\sum_{k\neq j}\exp(s_{ik}/\tau) + \sum_{k\neq i}\exp(s_{ik}/\tau)} \\ &+ \log \frac{\exp(s_{ij}/\tau)}{\sum_{k\neq i}\exp(s_{jk}/\tau) + \sum_{k\neq j}\exp(s_{jk}/\tau)}\Bigg) \end{aligned} \tag{5–22}$$

其中 τ 是调节对比范围的参数，s_{ij} 表示点 v_i, v_j 之间的特征相似度，我们在实验中采用余弦相似度（Cosine similarity）作为这里的相似度度量函数。那么对于一对图，我们有总体的对比损失函数：

$$\mathcal{L}_{con} = \frac{1}{2n}\sum l(i,j) \tag{5–23}$$

其中 n 代表有匹配的点数。在最后计算总体损失函数时，我们使用参数 γ 作为对比损失函数与匹配函数的平衡参数。

注意到，这样的特征学习方法其实并不仅仅局限于我们提出的基于中间图的方法。事实上，其可以作为一种预训练方法，并将经过预训练后的特征，在其他的机器学习图匹配模型上进行进一步的微调学习，以帮助现有的其他机器学习图匹配模型得到更好的求解结果。

由于预训练方法不再受到学习时二图匹配的限制，我们可以进一步扩展对于负样本的定义。也就是说，对于一批图 $\mathcal{G}_1,\dots,\mathcal{G}_N, v_i \in \mathcal{G}_1$，其正样本包括 $\mathcal{G}_2,\dots,\mathcal{G}_N$ 中所有能与其匹配的点，而负样本则包括 $\mathcal{G}_2,\dots,\mathcal{G}_N$ 中所有未与其匹配的点和 $\mathcal{G}_1$ 中除其本身以外的点。此时，一个点对对比损失函数为：

$$\begin{aligned} l(i,j) = -\Bigg(&\log \frac{\exp(s_{ij}/\tau)}{\sum_{k\neq i}\exp(s_{ik}/\tau) + \sum_{\mathcal{G}_j\neq\mathcal{G}_i}\sum_{x_{ij}=1,k\neq j}\exp(s_{ik}/\tau)} \\ &+\log \frac{\exp(s_{ij}/\tau)}{\sum_{k\neq j}\exp(s_{jk}/\tau) + \sum_{\mathcal{G}_i\neq\mathcal{G}_j}\sum_{x_{ij}=1,k\neq i}\exp(s_{jk}/\tau)}\Bigg) \end{aligned} \tag{5–24}$$

在实际实验中，我们对 BBGM[26]采用特征值预训练的方法，对我们的 UPM 直接采用对比损失函数结合匹配损失函数进行训练，以达到对节点特征进一步优化的训练效果。注意到，与其他对比学习自监督模型相比，我们的训练在有监督的条件下进行，监督信号为匹配真值。

5.5 算法局限性与拓展空间讨论

在面对多图部分匹配这一长久且具有挑战性的问题时，UPM 也有一定的局限性。首先，多图对应关系（MGC）的定义依赖于数据集的真实标签，这些标签会被用来区分内点和外点。然而，这种判断的可信度取决于图对的标注程度。当数据集小而噪声大，或者标签极度稀少的时候，MGC 的可靠性会被打破，也从而导致其性能不佳。在实验的 ‘table’ 和 ‘sofa’ 这两个类中，由于类别的图片数目很少，标签稀疏，我们方法的性能有所下降。

另一个局限性在于，我们的算法的召回率与准确性相比较低。虽然 UPM 可以识别节点类型从而减少假性匹配，但他的预测仍然遗漏了很多正确的匹配。后文的实验结果也现实，虽然我们定义的错匹配（ill-matching）和过度匹配（over-matching）均有所下降，但是正确的匹配并没有大幅上升。也就是说，我们的算法 UPM 对于其预测结果是谨慎的，它会遗漏一些不明显的匹配。这主要归因于 UPM 会将一些不太明显的内点与中间图上的吸收锚点进行匹配。在某些情况下，当一些内点在整个数据集中很少出现的时候，负匹配带来的更新会超过正匹配的指导，因此一些内点会被视作外点。这个问题影响我们的算法 UPM，使其不能做出更好的我预测，这也是我们未来可以拓展改进的空间。

5.6 本章小结

本章中，我们首先介绍了将中间图这一概念引入机器学习的图匹配问题求解的动机，和我们提出的中间图视角下的部分图匹配问题定义。之后我们介绍了中间图相似度学习的设计模型，并对其进行了数学分析。我们对中间图相似度学习、中间图结构，以及外点聚类的损失函数设计进行了详细的讨论。我们还介绍了基于中间图视角的对比学习方法，以及以此引出的预训练方法。最后我们探讨了模型 **UPM** 的局限和可能的拓展。

第六章　基于中间图的图匹配模型实验

本章中，我们从三个方面展示我们的实验结果：部分图匹配下的对比，UPM 方法的探索与分析，和部分拓展应用场景。所有的实验在一个有 Nvidia 2080Ti GPU，Intel i7-7820X CPU @ 3.60GHz，128GB 内存的 Linux 工作台上运行。

6.1　数据集与对比方法

PascalVOC 数据集　改数据集[65]与 Berkeley 标记[66]包含带有边界框的 20 类物体。我们遵循 NGM 和 BBGM[25-26]中的标准准备流程，所有物体被裁剪至其边界框，并缩放到 256 × 256 像素。由此产生的图像按照类别包含最多 23 个关键点。我们遵循前作[26]提出的 'Keypoint inclusion' 的设置，不对未匹配的内点进行过滤。此外，我们会随机在图像坐标上进行抽样以增加外点。我们同样遵循 NGM 和 BBGM 的设置来分割训练和测试集，其中训练数据包括 7020 张图像，测试数据包括 1682 张图像。

Willow-ObjectClass 数据集　该数据集[56]包含来自 Caltech-256[57]和 Pascal VOC 2007[58]数据集共 5 类 256 张图像：40 张汽车、40 张摩托车、50 张鸭子、66 张酒瓶和 109 张人脸。每张图片都有 10 个关键点的标签，我们也会在实验中加上生成的外点，除特殊说明外，外点均为随机产生。图结构由 Delaunay 三角剖分构造。我们遵循标准的流程，将图像裁剪到物体的边界框，并缩放到 256×256 像素。我们从每个类中选择 20 张图片作为训练数据，其余图片均作为测试数据。

对比方法　我们将 UPM 和多种前作方法进行对比，主要包括两类：机器学习图匹配求解模型和非学习的求解器。在我们的实验中，后者的相似度矩阵由预训练模型产生，以达到尽量公平的对比。

对于学习方法，与前作采用 VGG[52]网络不同的是，我们的实验中，均使用 ResNet50[61]来提高模型的表达能力和成本效率。为了进行公平比较，我们用 ResNet50 替换了所有方法的特征提取器，并采用其论文或代码中发布的最佳超参数进行训练，在 `MLP` 层中，特征维度被减少到 1024 维，保持和我们方法的一致性。参与比较的方法包括：PCA / IPCA[23]，NGM[25]，BBGM[26]，DLGM[27]。注意其中 NGM 和 BBGM 是现有的最佳二图匹配机器学习求解模型，DLGM 的贡献与所有方法正交因此也可以被使用到我们的模型中。由于 DLGM 方法未公开代码，我们无法替换特征提取器进行实验，只能直接汇报其论文中的结果。

非学习求解器并不对前端提取器进行学习，只通过得到的相似度矩阵进行求解，因此我们采用由 ResNet50 训练得到的 BBGM[26]的相似度矩阵作为非学习方法的输入。非学习求解器的所有超参数均按照其论文中汇报的设置。我们对比的方法包括：GAGM[9]，RRWM[10]，BPF[67]，ZAC[68]，CAO[16]，DPMC[20]，GA-MAMC/GANN-MGMC[21]。其中前四个均为二图匹配求解器，CAO 为多图匹配求解器，DPMC 和 GA-MAMC/GANN-MGMC 为多图匹配聚类求解器。

表 6–1 三种错误匹配类型：误匹配、错匹配、过匹配。错误匹配类型由某个点对中两个点的类型来决定，第一行第一列分别表示两个点的类型与其对应的匹配类型。

Table 6–1 Wrong match types: mismatching, ill-matching, and over-matching. The wrong match types are defined by the node type of two nodes that belong to the specific node correspondence. The header row and header column represent their node types.

	匹配内点（MI）	未匹配内点（UI）	外点（O）
匹配内点（MI	mismatching	ill-matching	over-matching
未匹配内点（UI）	ill-matching	ill-matching	over-matching
外点（O）	over-matching	over-matching	over-matching

6.2 测试指标与流程

6.2.1 测试指标

F1 分数 我们把匹配结果看成二分类问题，因此 F1 数值可以用于作为评价指标，其首次被 BBGM[26]在 'Keypoint inclusion' 的设置中使用。F1 数值计算为：

$$\mathbf{F}_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{6–1}$$

其中 TP 表示真阳性，FP 表示假阳性，FN 表示假阴性。F1 分数是我们实验中最重要的指标。

准确度（Accuracy） 我们也采用准确率做为在线匹配的评价指标，其代表预测匹配中正确节点占据的比例：

$$\mathbf{acc} = 1 - \frac{\|\bar{\mathbf{X}} - \mathbf{X}_{gt}\|_F^2}{\|\bar{\mathbf{X}}\|_F^2} \tag{6–2}$$

其中 $\|\cdot\|_F$ 代表 Frobenius 范数。

匹配类型（Matching Type） 匹配类型用于确定节点类型识别能力。我们将所有的匹配预测氛围四种类型：正确匹配（correct matching），误匹配（mismatching），错匹配（ill-matching），过匹配（over-matching）。第一种表示正确的匹配，后面三种表示三种不同类型的错误。点对 (v_i, v_j) 的预测匹配类型由点 v_i, v_j 的类型所决定，其对应关系见表6–1。错匹配的数目可以表示模型区分未匹配内点的能力，过匹配的数目表示模型区分外点的能力。我们计算每种匹配类型占所有匹配的比例，以说明模型在区分节点类型上的能力。

聚类指标

- **聚类匹配准确性 / F1 分数（Matching Accuracy of Cluster / Matching F1 Score, MAC/F1C）**类内的匹配质量：

$$\begin{aligned}
\mathbf{MA} &= 1 - \frac{1}{k}\sum_{j=1}^{k}\frac{1}{|C_j|^2}\sum_{(\mathcal{G}_a,\mathcal{G}_b)\in C_j}\mathbf{acc}(\mathbf{X}_{ab}, \mathbf{X}_{ab}^{gt}) \\
\mathbf{F1C} &= 1 - \frac{1}{k}\sum_{j=1}^{k}\frac{1}{|C_j|^2}\sum_{(\mathcal{G}_a,\mathcal{G}_b)\in C_j}\mathbf{F1}(\mathbf{X}_{ab}, \mathbf{X}_{ab}^{gt})
\end{aligned} \tag{6–3}$$

- **聚类纯度（Clustering Purity, CP）**令 C_i 表示预测的第 i 个类，C_j^{gt} 表示第 j 个类的真值，k 代表聚类个数，m 代表总图数：

$$\mathbf{CP} = \frac{1}{m}\sum_{i=1}^{k}\max_{j\in\{1\dots k\}}|C_i \cap C_j^{gt}|. \tag{6–4}$$

- **随机指标指数（Rand Index, RI）**令 n_{11} 表示同一类中预测标签一致的图数，n_{00} 表示不同类中预测标签不一致的图，有：

$$\mathbf{RI} = \frac{n_{11}+n_{00}}{n}. \tag{6–5}$$

- **聚类准确性（Clustering Accuracy, CA）**通过计算错误的聚类数目得到准确性：

$$\begin{aligned}\mathbf{CA} = 1 - \frac{1}{k}\Big(&\sum_{C_i}\sum_{C_{j_1}^{gt}\neq C_{j_2}^{gt}}\frac{1}{|C_i||C_i|}\cdot|C_i\cap C_{j_1}^{gt}||C_i\cap C_{j_2}^{gt}| \\ &+\sum_{C_{i_1}\neq C_{i_2}}\sum_{C_j^{gt}}\frac{1}{|C_{i_1}||C_{i_2}|}\cdot|C_{i_1}\cap C_j^{gt}||C_{i_2}\cap C_j^{gt}|\Big)\end{aligned} \tag{6–6}$$

6.2.2 训练与测试流程

在我们的实验中主要包括以下三种测试与训练流程：部分图匹配设置，在线增量式匹配设置，多图匹配聚类问题。

- **部分图匹配设置** 是本文的基本设置。未匹配的内点和外点要么是从数据集本身获得的，要么由我们手动添加。在一些实验中，有可能没有未匹配内点或外点。测试图随机从同一类的图中抽样得到，并且我们每一次只利用两张图构造我们的匹配实例。测试步骤与训练步骤保持一致。
- **在线增量式匹配** 在[19, 33]中提出，指的是输入图并不是一次性给定而是在匹配过程中一个接一个输入的情况。本文中，我们在训练的时候保持与部分图匹配设置相同的设置。在测试阶段，我们随机从同一类图中取样 15 张图，并将他们依次送入模型。此外，在线匹配是在部分图匹配的基础上进行的，其中未匹配内点和外点也按照标准的部分图匹配场景设置。
- **多图匹配聚类** 在[20]中提出，指的是处理当输入为不同类的图的情况，并需要同时获得匹配和聚类的结果。同另外两类问题已知，我们也为数据集添加了外点。在训练阶段，我们保留一半的样本来自于同一类图，另外一半样本为不同类型的图组成的图对，这些图对之间的匹配真值全部设置为零。在评估阶段，我们按照[20-21]中的流程，随机选择几个类，并在选定的类中取样一部分图。聚类结果通过谱聚类[37]方法基于相似度数值得到。UPM 的相似度数值定义为：

$$\text{Affinity} = \hat{\mathbf{X}}_{ab}\circ(\mathcal{S}_a\mathcal{S}_b^{\top}) \tag{6–7}$$

其中 $\circ$ 代表 Hadamard 乘积。

6.3 部分图匹配表现

我们首先在 PascalVOC 数据集上进行了两个均带有未匹配点的不同设置的标准部分图匹配对比实验：

表 6–2 PascalVOC 上带有未匹配内点和 2 个外点的实验的 F1 分数。所有机器学习方法都使用 ResNet50 作为特征提取器。非学习方法的相似度矩阵通过 BBGM 学习到的一阶和二阶相似度构造而成。我们采用原论文、代码中提供的参数，或者对其进行微调，以得到为部分图匹配场景最合适的参数。

Table 6–2 F1 score comparison on PascalVOC with all unmatched inliers and two random outliers. All the learning-based methods apply the ResNet50 backbone. The affinity matrix of the learning-free solver is built with the unary and quadratic cost learned by BBGM. We either adopt the optimal hyperparameters reported in their paper or adjust them to fit the partial matching setting.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	avg
IPCA[23]	32.6	51.0	32.7	26.4	64.0	45.2	25.1	43.0	26.0	41.3	22.1	39.5	41.0	40.1	30.7	57.6	33.0	23.1	37.0	58.1	38.46
NGM[25]	35.3	57.4	42.3	31.9	68.5	47.0	30.7	49.4	33.4	48.3	38.5	48.5	46.7	49.5	39.9	70.8	36.8	**38.7**	45.0	58.1	45.84
BBGM[26]	36.2	63.2	54.1	36.0	75.1	60.8	20.9	65.1	33.7	61.2	36.0	60.1	57.4	57.2	39.4	81.3	53.5	27.1	51.3	78.5	52.41
BBGM[26]+CL	38.5	64.9	56.4	39.3	77.4	62.2	47.8	65.5	34.2	62.7	44.3	62.6	57.4	56.8	49.0	82.6	55.5	25.3	58.6	79.2	56.00
GAGM[9]	26.7	49.3	36.6	25.1	61.9	40.7	9.0	45.4	22.5	39.3	13.1	41.0	37.8	39.1	27.9	59.8	29.1	19.7	26.0	54.9	35.25
RRWM[10]	28.3	54.6	40.3	27.4	70.6	46.3	9.0	49.8	23.5	42.8	15.1	44.8	38.1	39.1	30.3	66.8	29.3	18.1	27.1	64.0	38.27
BPF[67]	25.8	55.2	38.8	26.0	70.7	45.7	9.5	49.4	22.9	40.2	14.4	44.1	36.9	40.0	29.0	68.6	28.6	18.8	28.1	63.8	37.83
ZAC[68]	37.2	62.3	50.8	34.0	76.0	55.9	23.8	63.7	33.2	58.8	37.7	59.4	55.8	55.4	39.1	76.8	49.3	29.0	49.1	68.9	50.81
UPM	**46.3**	**67.2**	**65.3**	**44.9**	**80.5**	**72.7**	**63.5**	**71.9**	**49.1**	**73.5**	**58.8**	**69.4**	**63.6**	**62.4**	**55.5**	**86.4**	**63.0**	28.4	**68.5**	**80.8**	**63.57**
UPM+CL	47.3	66.8	64.8	39.6	82.9	74.6	64.6	67.8	50.3	70.7	63.7	69.2	62.3	63.5	55.0	86.3	63.5	48.1	67.6	81.4	64.50

表 6–3 PascalVOC 上带有未匹配内点和无外点的实验的 F1 分数比较。所有机器学习方法除了 DLGM，均使用 ResNet50 作为特征提取器。非学习方法的相似度矩阵通过 BBGM 学习到的一阶和二阶相似度构造而成。我们采用原论文、代码中提供的参数，或者对其进行微调，以得到为部分图匹配场景最合适的参数。

Table 6–3 F1 score comparison on PascalVOC with all unmatched inliers and without any outlier. All the learning-based methods apply the ResNet50 backbone except DLGM. The affinity matrix of the learning-free solver is built with the unary and quadratic cost learned by BBGM. We either adopt the optimal hyperparameters reported in their paper or adjust them to fit the partial matching setting.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	avg
IPCA[23]	33.0	56.5	38.8	32.1	80.1	55.0	30.8	43.4	32.2	44.8	39.1	40.5	44.2	52.4	34.0	81.0	38.8	26.1	55.7	75.3	46.69
NGM[25]	42.4	69.3	51.2	40.5	87.0	62.6	43.5	57.9	45.1	57.3	53.4	55.0	55.9	62.5	46.6	92.5	44.7	41.8	66.6	73.6	57.46
DLGM*[27]	43.8	72.9	58.5	**47.4**	86.4	71.2	53.1	66.9	54.6	67.8	**64.9**	65.7	66.9	70.8	47.4	96.5	61.4	**48.4**	77.5	83.9	65.30
BBGM[26]	45.7	75.4	64.3	**47.4**	**88.2**	63.3	49.5	70.2	44.9	69.5	54.8	67.3	66.6	71.8	53.1	95.9	67.0	33.0	**80.5**	80.5	64.45
BBGM[26]+CL	50.0	73.0	66.9	51.3	88.9	64.6	64.7	71.7	44.2	72.5	51.8	70.9	69.5	71.6	56.4	95.8	66.4	33.2	77.9	83.4	66.22
GAGM[9]	37.3	60.3	47.7	39.1	75.0	53.7	36.1	59.2	34.8	56.2	40.2	54.4	51.9	54.3	39.0	83.8	48.8	19.0	58.3	68.7	50.89
RRWM[10]	39.4	63.7	52.1	39.7	76.5	57.9	28.5	64.3	37.9	60.3	43.0	57.9	54.2	54.0	43.3	85.0	50.4	22.0	66.1	69.9	53.30
BPF[67]	39.2	66.5	51.2	40.0	77.0	59.0	26.7	63.8	36.9	60.6	49.7	58.1	54.3	58.2	43.6	84.9	50.7	23.0	66.4	70.4	54.02
ZAC[68]	43.5	62.8	57.5	43.9	72.1	61.2	33.6	69.0	38.3	64.1	46.4	66.2	61.8	61.4	46.4	83.2	58.1	27.9	69.8	72.6	56.99
UPM	**50.5**	**74.1**	**67.3**	46.1	86.5	**74.9**	**75.6**	**73.0**	**59.4**	**73.0**	43.3	**72.4**	**67.6**	**72.2**	**55.2**	**97.5**	**68.3**	45.0	75.5	**85.1**	**68.11**
UPM+CL	48.9	76.5	67.8	48.5	89.4	78.6	55.6	72.2	64.2	72.6	64.3	74.6	69.9	73.5	54.2	97.2	65.2	53.4	83.0	81.0	69.53

- PascalVOC 上带有未匹配内点和 2 个外点的实验，
- PascalVOC 上带有未匹配内点和无外点的实验。

在训练阶段，我们总共训练 10 轮 UPM，在每轮中，我们随机取样 16000 对图参与训练。我们同样随机取样 8000 对图进行测试。我们使用大小为 30，特征维度为 1024 维的基于特征

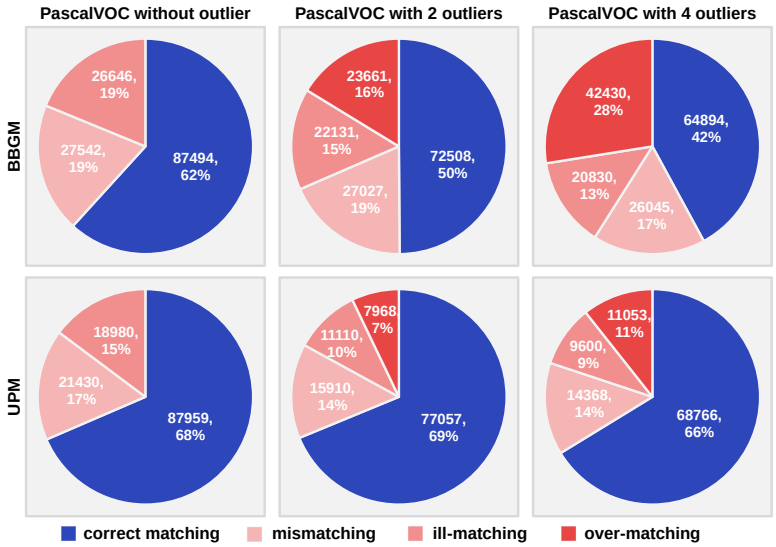


图 6–1 PascalVOC 上不同外点下匹配类型的分布。作为代表，我们测试了 BBGM 和 UPM 的表现对比。图上的数字代表不同匹配类型在各个类中的出现次数之和，我们也汇报了其占比。从图中可见，UPM 在区分节点类型上有出色的表现。

Figure 6–1 Distribution of matching type on PascalVOC with different numbers of outliers. For representativeness, both state-of-the-art BBGM and the proposed UPM are tested. The occurrence number of each matching type over all the categories are summed up. Both the occurrence number and proportion are reported for each matching type. It shows our method UPM has superiority on distinguishing node types.

的中间图结构。表6–2和表6–3中为实验结果和对应报告的 F1 分数。

可以看出，UPM 在这两种情况下都优于所有其他的方法。在复杂的环境中，UPM 比深度学习模型 BBGM 和非学习的求解器 ZAC 的平均水平高出 11.16% 和 12.76%，甚至在几乎所有的类别中都超过了同行方法. 在没有外点的场景中，外点感知损失函数和外点过滤器都不再起效。然而 UPM 和其他方法仍然体现了显著的优势：对于学习方法提高 2.81%，对非学习方法提高 11.12%。此外，UPM 在 ‘table’（表6–3）和 ‘sofa’（表6–2）中的表现不佳，这事实上支持了我们在5.5中的分析，也就是说，我们的方法会受到小而充满噪声的数据集的影响。

我们还结合对比学习特征优化方法在 UPM 和 BBGM 上进行训练。其中在 UPM 上，我们直接采用带有对比学习损失函数的损失函数进行学习，对于 BBGM，我们则根据对比学习损失函数对特征提取器进行了一轮预训练，预训练采用一组 4 张图，批大小为 8 的训练方法，之后再利用 BBGM 在预训练的特征提取器上进行图匹配问题的学习。二者的结果均在表6–2和表6–3中带有 ‘+CL’ 的行中展示。对于 UPM，对比学习产生了约 1% 的效果提升，对于 BBGM，对比学习产生了 1-3% 的效果提升，可见我们提出的基于中间图视角的对比学习不仅可以帮助提升 UPM 的特征提取器，更是能够作为一种预训练方法，为其他的机器学习模型提供帮助。注意到，这种预训练方法与所有的学习方法均正交，也就是说，其方法和效果是普适的。

图6–1展现了节点类型识别上两个模型的比较。我们在 0/2/4 个外点的场景下进行了三轮测试，并报告了 BBGM 和 UPM 的各个匹配类型在所有匹配中占用的比例（具体类型见

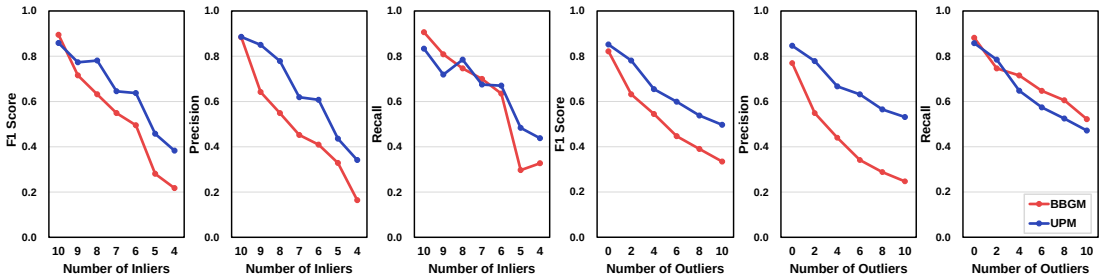


图 6–2 WillowObject 上不同部分图匹配场景下的 F1 分数、精度、召回率。左侧的图片用于展现 UPM 对于未匹配内点的鲁棒性，右侧的图片用于展现 UPM 对于外点的鲁棒性。

Figure 6–2 F1 score, precision and recall on Willow-Object with different extensions of partial matching cases. The figures on the left show the robustness of our method against unmatched inliers, while the figures on the right show the robustness of our method against outliers.

表6–1)。可见，在没有外点的情况下，UPM 的正确匹配率提高了 6%。与 BBGM 的差距在有外点的场景下更大，甚至在 4 个外点的情况下达到了 24%。同时，我们的错误匹配率从 0 到 4 个外点的情况下依然保持一致。也就是说，UPM 在处理外点的时候有着极强的鲁棒性，我们将在接下来用更严格的实验说明这一点。

我们在 WillowObject 这一数据集上对 UPM 针对未匹配内点和外点进行了压力测试：

- WillowObject 上保持 2 个外点，随机去除 0 ~ 6 个内点，
- WillowObject 上保持去除 5 个内点，随机增加 0 ~ 10 个外点。

我们对模型进行 2 轮训练，每一轮随机采样 400 张图用于训练，100 张图用于测试。我们将中间图大小设置成 20 和 25，保持特征维度为 1024，并采用基于特征的中间图结构。

如图6–2所示, UPM 超过 BBGM, 尤其是在极端测试数据中。在针对内点的测试中, UPM 在 F1 上超过 BBGM 达到 5% 到 10%。在对于外点的测试中，UPM 超过 BBGM 达 3% 到 15%。并且，测试场景越复杂，UPM 的表现越强大。同时，我们还表明，在预测精度上，UPM 有着很大的优势。在所有情况下，我们的召回率与 BBGM 相似，这也支持了我们在5.5中提到的我们的预测倾向于给出保守的预测结果。

6.4 UPM 方法的探索与分析

我们主要进行了四个实验来进一步探索与分析我们提出的 UPM 方法：中间图结构、中间图相似度与二图相似度对比、泛化性测试、超参数敏感性测试。

6.4.1 中间图结构

为了证实我们在5.2.2中提到的中间图结构的假说，我们在 WillowObject 上进行了一个实验来对基于特征和基于节点的中间图分布进行可视化，结果如图6–3。

正如我们之前提到的，基于特征的中间图的大小较小，为 UPM-20（20 为中间图中的点个数 n_u），而基于节点的中间图较大，为 UPM-90。我们用标准的部分图匹配模型训练 UPM-20，用多图匹配聚类的训练流程训练 UPM-90。两个模型都在有 2 个外点的设置下训练，并且没有去掉任何内点。UPM-20 和 UPM-90 都在 F1 分数上取得了超过 85 的结果，也就是说，我们的模型都在最理想的情况下得到训练。我们在图6–3中由此通过展现点匹配结果分布来表现两类中间图的结构分布。

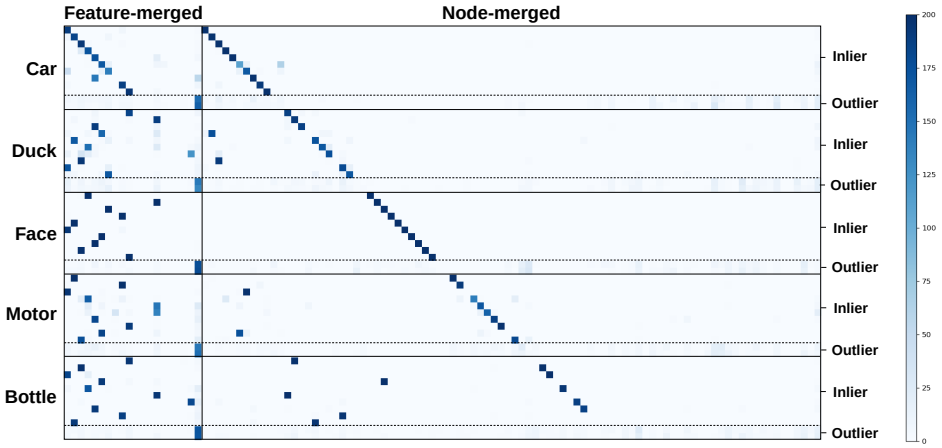


图 6–3 WillowObject 上带有 2 个外点的中间图匹配相似度分布。我们在每个类中随机取样 200 张图并计算每个点与对应中间图上锚点的匹配关系。X 轴代表基于特征的中间图和基于节点的中间图，其大小分别为 20 和 90。Y 轴代表不同类中有不同语义信息的内点和随机添加的外点，其用虚线隔开。中间图匹配点对出现的次数越多在图中的颜色越深。由于中间图上的锚点与其置换关系无关，我们在可视化的过程中调整了节点列的顺序。

Figure 6–3 Universe node matching likelihood distribution on WillowObject with two outliers. We randomly sample 200 graphs for each category and count the correspondence between input graphs and two types of universe graphs. The X-axis denotes the nodes in the feature-merged graph and node-merged universe graph, whose size is 20 and 90 respectively. The Y-axis denotes the nodes of different semantic meaning in all the categories, where inlier and outlier are divided by dotted lines. Darker color represents more appearances for the node correspondence pair. Since all the universe nodes are permutation invariant, we adjust the order of columns for better visualization.

首先，基于特征的中间图结构将来自不同类别的节点特征合并起来，以便将不同类别中的不同节点匹配到同一个中间图锚点上，而基于节点的中间图结构会将所有节点分开划分，这两个观察支持了我们对两种图结构的假设（图5–3）。此外，基于特征的中间图的外点分布也支持了我们关于外点过滤器和外点感知损失函数设计的动机。外点会被匹配到中间图上的吸收锚点。对于基于节点的中间图，可见外点会分散在第 61-90 个锚点之间，这主要是因为中间图大小远超准确的特征个数，因此为外点的额外匹配留下了空间。

6.4.2 中间图相似度与二图相似度

正如我们在方程5–6中提到的，通过中间图相似度（Universe Affinity, UA）构造二图相似度（Pairwise Affinity, PA）的过程并不会产生信息损失。因此我们通过实验来比较中间图相似度构建的二图相似度和传统的二图相似度之间的质量。由于我们只有节点相似度，图 $\mathcal{G}_a, \mathcal{G}_b$ 之间的相似度矩阵 $\mathbf{K}^u$ 构造为：

$$\begin{aligned} \mathbf{K}^u_{n_i,n_k} &= \mathcal{S}_{ab}(i,k)^2 \\ \mathbf{K}^u_{e_{ij},e_{kl}} &= \mathcal{S}_{ab}(i,k) \cdot \mathcal{S}_{ab}(j,l) \end{aligned} \tag{6–8}$$

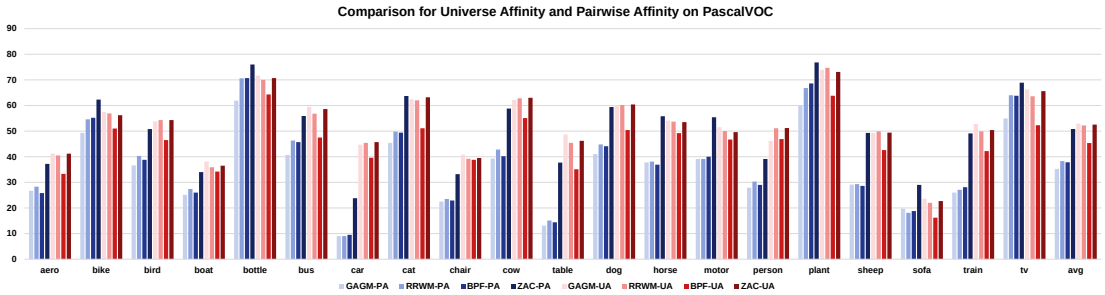


图 6–4 PascalVOC 上中间图相似度和二图相似度对比。中间图相似度 $\mathbf{K}^u$ 由 UPM 学习到的 $\mathcal{S}$ 按照方程6–8构造而成。二图相似度由 BBGM 使用 ResNet50 提取器学习得到。'-UA' 表示使用中间图相似度 $\mathbf{K}^u$ 的非学习求解器，'-PA' 表示使用二图相似度的非学习求解器。平均的 F1 分数在最后一列显示。

Figure 6–4 Comparison of F1 score for universe affinity and pairwise affinity on PascalVOC. The average F1 score is reported at last. Universe affinity matrix $\mathbf{K}^u$ is built with $\mathcal{S}$ learned by UPM as shown in Eq. 6–8. Pairwise affinity matrix is learned by BBGM with Resnet50 backbone. Methods with '-UA' denotes the learning-free solver with universe affinity $\mathbf{K}^u$, and '-PA' denotes for pairwise affinity matrix.

其中 $\mathcal{S}_{ab}$ 由公式5–6中的中间图相似度 $\mathcal{S}_a, \mathcal{S}_b$ 构造而成，且其是良定义的，因为在完全匹配和图结构为完全图的情况下，$\mathbf{K}^u$ 的 QAP 问题等价于 $\mathcal{S}_{ab}$ 上的线性指派问题。

$$\begin{aligned}
&\arg\max_{\mathbf{X}} \operatorname{vec}(\mathbf{X})^\top \mathbf{K}^u \operatorname{vec}(\mathbf{X}) \\
=&\arg\max_{\mathbf{X}} \sum_{\mathbf{X}_{ik}=1} \sum_{\mathbf{X}_{jl}=1} \mathcal{S}_{ab}(i,k) \cdot \mathcal{S}_{ab}(j,l) \\
=&\arg\max_{\mathbf{X}} \left(\sum_{\mathbf{X}_{ij}=1} \mathcal{S}_{ab}(i,j) \right)^2 \\
=&\arg\max_{\mathbf{X}} \sum_{\mathbf{X}_{ij}=1} \mathcal{S}_{ab}(i,j)
\end{aligned} \tag{6–9}$$

我们将中间图相似度 $\mathbf{K}^u$ 与由 BBGM 通过 ResNet50 训练得到的二图相似度 $\mathbf{K}$ 进行比较。他们以相同的形式送入非学习求解器 GAGM、RRWM、BPF、ZAC 中。我们遵循 PascalVOC 上标准的部分图匹配测试流程，随机增加两个外点并且不去除未匹配内点。两种相似度在各个类别上的结果如图6–4所示。

总体上可见，所有使用 $\mathbf{K}^u$ 的方法都比使用传统二图相似度 $\mathbf{K}$ 的方法更好。对于最新的非学习求解器 ZAC（注意 ZAC 有专门针对外点的优化），中间图相似度提供了 2% 的表现提升，而其对 GAGM 和 RRWM 的提升更为明显（17% 和 14%）。同时所有的方法在使用中间图相似度的时候会倾向于得到相似的匹配结果，GAGM、RRWM、ZAC 得到的结果几乎相同。这可能是因为基于点的中间图相似度能够简化基于边的相似度信息，从而降低了寻找最优解的难度。

6.4.3 泛化性测试

我们还在部分图匹配的设置上测试 UPM 的泛化性。这里的泛化性指的是当测试和训练环境不同（例如，不同的外点数目或者不同的未匹配内点数目）时，训练模型的表现变化。

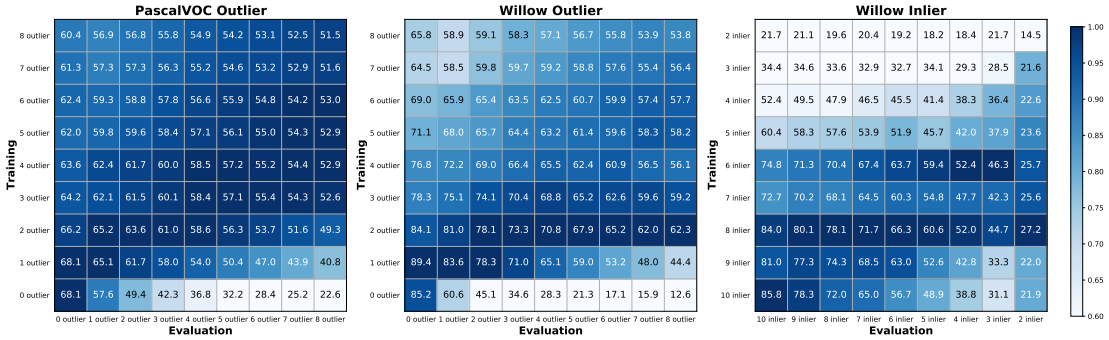


图 6–5 UPM 在 PascalVOC 和 WillowObject 上对于外点和未匹配内点的泛化性测试。模型在 Y 轴坐标的设置上进行训练，在 X 轴坐标的设置上进行测试。我们汇报了所有的 F1 分数。图中颜色根据 F1 分数关于其所在列的最大值进行归一化，更深的颜色代表同一个测试场景中更好的表现。一行的颜色越深代表训练模型的泛化能力越强。

Figure 6–5 Generalization study on outliers and unmatched inliers shown as confusion matrix of our proposed UPM on Pascal VOC and Willow Object. Models are trained on the setting of the y-axis and tested on the setting of the x-axis. F1 scores are reported in the entries of confusion matrices. The color map is determined by the F1 score value in the cell normalized by the highest value in its column. Darker color denotes relatively better performance in the same testing setting. The darker the row is, the better the generalization ability of the model.

对于 PascalVOC 数据集，我们保留所有的未匹配内点，并在 0 到 8 个外点的场景下测试泛化性。对于 WIllowObject 数据集，我们首先随机去除 2 个未匹配内点，并在 0 到 8 个外点上测试泛化性，之后我们随机增加 2 个外点，并在去除 0 到 8 个内点的场景下测试泛化性。它们的 F1 分数在图6–5中呈现。表格的颜色是由当前单元格所在列的最高值归一化后决定的，更深的颜色表示同一个测试场景下更好的表现。同一行的深度越深，表示模型的泛化性越好。

可见，针对不同数据的外点，UPM 表现了强大的泛化性。用更少外点训练出来的模型能够在有很多外点的测试场景下依然得到很好的 F1 分数。在有 4/5/6/7 个外点的情况下，模型在 3 个外点的场景中训练能够得到最好的结果。在大量外点中训练的模型也能在更简单的场景中得到比较好的结果。大部分模型都得到了超过最高 F1 分数 80% 的结果，包括 WillowObject 中未匹配内点和外点较 PascalVOC 上更严峻的场景。值得注意的是，只有在没有外点的情况下训练出来的模型不能很好地泛化到有大量外点的情况，这也表明在训练中引入外点可以显著提高泛化能力。

另一方面，用去掉部分内点的场景训练的模型在去掉更多内点的测试中也能得到最高的 F1 分数（比如在去掉 2 个内点场景下训练的模型）。然而，当训练场景过于复杂时，模型的能力也会急剧下降，比如当我们随机去除 10 个内点中的 8 个时，得到的训练数据很难存在有效的匹配对，因此在这样实例上进行训练的模型的结果很差。

此外，我们并没有对不同类别使用混淆矩阵进行研究。因为我们很难对一个未知的类构建有效的中间图，我们的模型 UPM 在这类泛化实验中注定难以得到很好的结果。

6.4.4 超参数敏感性测试

我们通过两种中间图构建方案对中间图大小 n_u 进行了超参数敏感性测试。对于基于特征的中间图，我们在 PascalVOC 上使用标准的部分图匹配流程，并在 2 个外点，保留内点的

表 6–4 基于特征的中间图（上）和基于节点的中间图（下）的超参数敏感性测试。基于特征的中间图在同一类的输入图上进行训练和测试，基于节点的中间图在训练时有一半的数据输入为来自不同类的图，这些输入的真值匹配设成零。UPM-24 和 UPM-244 是两类中间图的精确大小。我们在表中汇报 F1 分数。

Table 6–4 Hyper-parameter sensitivity exploration for feature-merged (top) and node-merged (bottom) universe graph respectively. Feature-merged universe graph based UPM are trained and evaluated on pairs with the same category, while node-merged universe graph based UPM are trained and evaluated on pairs where half of them are of different categories. The ground truth matching between different categories is set to zero matrix. UPM-24 and UPM-244 show the performance of our method with the exactly universe size (inliers and an absorbing node) under feature-merged and the node-merged universe.

PascalVOC	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	avg
UPM-15	44.2	65.7	63.5	42.8	80.3	69.4	43.3	58.1	50.1	60.0	48.0	57.7	57.1	61.7	28.6	**86.7**	55.5	37.3	64.4	78.1	57.63
UPM-20	46.0	66.6	62.6	38.7	**82.3**	71.8	50.0	70.1	**54.7**	69.3	42.3	69.2	**66.9**	61.8	44.5	**86.7**	61.2	45.6	**68.8**	77.4	61.83
UPM-24	**47.1**	64.7	63.9	43.6	78.9	70.6	**67.1**	68.5	51.7	67.9	35.7	69.3	63.5	59.1	54.7	86.4	**62.6**	36.4	65.3	78.8	61.78
UPM-25	46.6	65.8	64.7	39.8	80.6	70.0	52.7	69.3	54.0	69.5	55.9	69.0	63.9	62.0	51.3	85.9	59.2	**48.1**	66.6	80.4	62.75
UPM-30	46.3	**67.2**	**65.3**	**44.9**	80.5	**72.7**	63.5	**71.9**	49.1	**73.5**	**58.8**	**69.4**	63.6	62.4	**55.5**	86.4	63.0	28.4	68.5	**80.8**	**63.59**
UPM-35	46.1	66.6	64.3	38.4	80.0	71.1	50	70.6	50.4	69.7	31.4	68.8	63.1	**63.7**	52.0	87.0	62.2	40.4	66.9	78.3	61.04
UPM-200	43.4	63.3	56.8	40.4	78.7	64.4	47.5	64.9	49.1	64.2	42.6	66.3	56.4	58.9	36.9	78.5	58.2	36.9	55.4	70.3	56.65
UPM-244	43.4	57.1	54.8	35.9	78.3	69.1	53	66.1	42.3	62.1	40.2	65.9	56.6	56.5	34.3	**84.3**	**61.0**	40.9	**64.0**	**77.3**	57.14
UPM-300	47.6	**66.7**	**61.6**	**46.1**	80.6	**69.5**	**59.2**	**67.2**	**55.9**	**70.4**	31.2	**70.1**	64.0	63.3	**53.3**	81.9	**61.0**	39.6	60.1	74.9	**61.2**
UPM-400	**47.7**	65.8	59.9	41.8	79.3	70.0	57.0	65.9	49.5	67.5	**47.1**	68.1	**65.5**	**63.4**	45.2	79.5	60.3	**44.3**	61.5	71.7	60.55
UPM-500	45.4	62.5	55.5	43.4	**80.8**	67.4	51.7	63.0	51.9	61.6	37.0	65.5	60.8	58.7	29.2	77.1	55.9	38.6	58.7	72.74	56.9

场景中进行训练和测试。由于 n_u 定义为子中间图的点特征数目 $n_u = \max n_i$，我们观察 n_u 取为 15 到 35 对应的性能变化。对于基于节点的中间图，我们采取前文所述的聚类匹配场景，我们报告 UPM 中间图大小 $n_u = 200, 300, 400, 500$ 的结果。注意 'person' 这个类有 23 种标签，所有类别加起来的标签数为 243，故我们特别测试了 **UPM-24** 和 **UPM-244** 的情况（都包含一个吸收锚点）。

如表6–4所示，UPM 对不同的中间图大小鲁棒。对于基于特征的中间图，即使高估或低估 40% 的节点数目也只下降 6% 。另一方面，在基于节点的中间图场景下，哪怕给定的 n_u 翻倍，在 F1 分数上也只下降 4.3% 。此外，我们还发现，低估中间图大小会比高估其大小产生更严重的影响，UPM 甚至鼓励稍微高估中间图的大小来达到最好的表现。如表6–4中所示，**UPM-30** 比 **UPM-24** 还高出 1.82%，即使后者对中间图的准确大小估计得更好。

6.5 在线匹配与多图匹配聚类拓展

我们在两个拓展问题上进一步测试 UPM 的表现：在线多图匹配（增量式多图匹配）[33, 69]与多图匹配聚类[20]。其中，前者涉及到输入的图依次进入的情况，而后者涉及给定的图包含多个类的情况。这两类情况都与传统的图匹配问题有所区别，前者要求能够更好地利用现有的匹配结果以提高匹配效率，后者要求能够在匹配的同时进行聚类。我们的实验说明 UPM 在这两种拓展场景中也能有很好的应用。

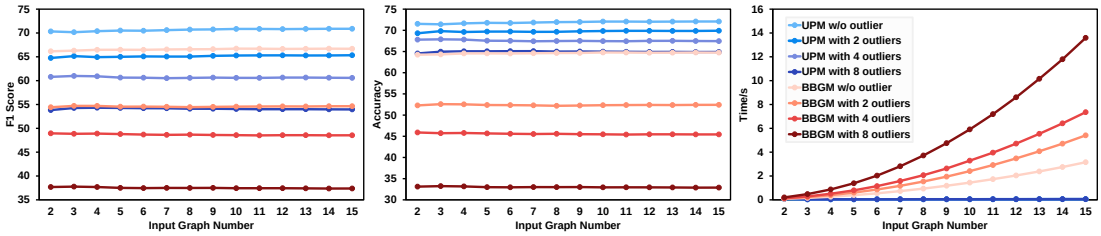


图 6–6 在线部分图匹配的 F1 分数、匹配准确率和时间开销比较图。我们在 PascalVOC 带有 0/2/4/8 个外点的设置下分别测试 UPM 和 BBGM 的结果。

Figure 6–6 F1 score, accuracy and time cost on online partial matching. We compare UPM with BBGM on PascalVOC with 0/2/4/8 outliers.

表 6–5 PascalVOC 上多图匹配聚类测试结果。我们保留所有未匹配内点并随机添加 2 个外点急性测试。表上的场景设置 3×8 表示随机取样 3 个类每个类取样 8 张图的情况。我们汇报 F1C 结果由于其相较于 MAC 指标更适用于部分图匹配的场景。

Table 6–5 Mixture graph matching and clustering on PascalVOC with inference time overhead in seconds. All the unmatched inliers are reserved and two outliers are randomly added for each image.'3×8' denotes that we randomly pick three categories in PascalVOC and for each category, we randomly sample 8 graphs to build the evaluation instance. Instead of MAC, we report the F1C score, which is more suitable for partial matching.

PascalVOC	cluster # × cluster size: 3×8					cluster # × cluster size: 4×6					cluster # × cluster size: 6×4				
	F1C↑	CP↑	RI↑	CA↑	time↓	F1C↑	CP↑	RI↑	CA↑	time↓	F1C↑	CP↑	RI↑	CA↑	time↓
BBGM[26]	52.68	83.89	85.51	81.75	2.36	52.37	84.24	86.46	82.06	2.36	53.25	**80.70**	89.42	**79.08**	3.48
UPM-30	50.16	81.15	81.59	77.01	**0.32**	47.33	78.02	83.72	72.74	**0.33**	48.69	71.62	84.86	67.90	**0.43**
UPM-300	**64.79**	**89.69**	**89.94**	**87.09**	0.34	**63.78**	**87.67**	**90.91**	**84.62**	**0.33**	**64.12**	80.33	**90.45**	77.00	**0.43**

6.5.1 在线多图匹配

我们随机选取 PascalVOC 中 15 张同一类的图，并为每张图随机加上 0/2/4/8 个外点，不去除内点。这些图一张一张地输入到我们的模型中。对于 BBGM，我们将新来的图与之前的所有图一一进行匹配。对于 UPM，我们直接将新来的图与中间图进行匹配，并且通过中间图计算新来的图与之前的图的二图匹配结果：$\hat{\mathbf{X}}_{ij} = \hat{\mathbf{X}}_{iu}\hat{\mathbf{X}}_{ju}^{\top}$。可以看出，此时，每张新来的图都只用与中间图匹配一次，其时间复杂度为常数。我们评估了随着图片数目增加得到的 F1 分数、准确度、时间开销，所有类的平均结果呈现在图6–6中。

所有方法的 F1 分数和准确度在每次新来一张图之后基本保持不变。同时，UPM 的运行速度显著快于 BBGM，且 BBGM 的时间开销几乎是线性增长，而 UPM 的时间开销几乎保持常数。这也展示了我们的中间图匹配机制在加速求解上的优势。

6.5.2 多图匹配聚类

在多图匹配聚类问题中，我们在 PascalVOC 和 WillowObject 上分别进行测试。所有学习模型都在 PascalVOC 和 WillowObject 上进行训练和测试，其一半的训练数据为不同类的图片，其匹配真值被设为零。在测试阶段，聚类结果通过在相似度数值上进行谱聚类[37]得到，与 DPMC 和 GA-MGMC 的测试标准相匹配。

表 6–6 WillowObject 上多图匹配聚类问题的测试结果。我们遵循前作[21] 中的测试设置，不添加额外的外点也不去除内点，并随机选用 8 张汽车、8 张鸭子、8 张摩托车作为测试数据。

Table 6–6 Mixture graph matching and clustering on WillowObject with inference time overhead in seconds. We follow[21] by randomly picking 8 cars, 8 ducks and 8 motorbike without outliers and unmatched inliers.

Willow	CP↑	RI↑	CA↑	MAC↑	time↓
RRWM[10]	87.9	87.1	81.5	74.8	**0.4**
CAO-c[16]	90.8	90.3	86.0	87.8	3.3
DPMC[20]	93.1	92.3	89.0	87.2	1.2
GA-MGMC[21]	92.1	90.5	89.3	65.3	10.6
GANN-MGMC[21]	97.6	97.0	96.3	89.6	5.2
UPM-90	**98.92**	**98.59**	**98.18**	**94.24**	0.52

我们首先将 UPM 与 WillowObject 上的非学习求解器进行比较。实验在 8 张汽车，8 张鸭子，8 张摩托车的场景下进行。结果如表6–6所示。可见 UPM 在所有测试指标上的表现均超过非学习方法，并且比传统求解器快 2× ~ 20×，同时有着良好的性能。与 RRWM 相比，在相近的时间成本下，我们在所有的匹配聚类评价指标上表现提高了超过 10%。

然而，几乎所有机器学习方法在 WillowObject 上的表现都很好，因为 WillowObject 是一个挑战性较小的数据集。在 PascalVOC 上，由于存在旋转、裁剪，其场景更为复杂。我们将 **UPM-30**（基于特征的中间图）和 **UPM-300**（基于节点的中间图）与 BBGM 相比较。在测试过程中，我们对类别和图均进行抽样，其按照 $N_c \times N_g$ 表示，其中 N_c 表示聚类数目，N_g 代表每个类中图片数目。

如表6–5所示，**UPM-300** 在聚类和匹配上的表现超过其他两种方法。在 $3\times 8, 4\times 6$ 的场景中，**UPM-300** 在 CP，RI，CA 这三个指标上达到 3-6% 的提升，同时匹配结果 F1C 和速度也有显著提高。随着图片类别的增加，CA、RI 和 CP 等聚类指标的差距也在缩小，但是 **UPM-300** 在 F1 分数上的优势还是非常明显。另一方面，**UPM-30** 不能很好地应对多图匹配聚类的场景，因为基于特征的中间图会将不同子中间图中的点特征混在一起，故无法拒绝跨类的匹配。这也支持了我们对于基于特征的中间图和基于节点的特征图特点的猜想。

6.6 本章小结

本章主要介绍了我们的机器学习模型 UPM 在各种实验场景下的实验结果。我们首先对 UPM 在经典的部分图匹配场景下进行测试，验证了 UPM 在部分图匹配问题下优秀的性能。我们随后通过大量的实验，对 UPM 中的各个组件进行了分析，包括对中间图结构猜想的分析，中间图相似度与二图相似度的区别分析，以及通过泛化性实验和超参数敏感度测试以证明我们方法的泛化能力、鲁棒性。最后我们还在两个更为复杂的拓展场景，在线多图匹配和多图匹配聚类问题下进行测试。UPM 在以上各个场景中均证实了其有效性。此外，据我们所知，UPM 是第一个可以同时处理二图匹配、多图匹配、多图匹配聚类、增量式匹配问题的机器学习算法，其不仅较先前方法有更高的准确率，更显著加速匹配速度。

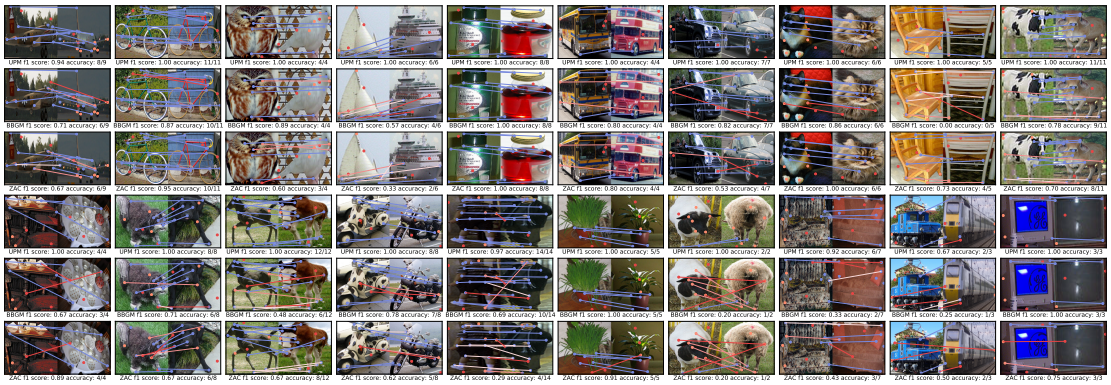


图 6–7 PascalVOC 上三种算法的可视化比较。图例与图5–1保持一致：蓝、橙、红色的点分别表示匹配内点、未匹配内点、外点。蓝线表示正确匹配，三种红线表示三种错误匹配。我们同时比较 UPM、BBGM、ZAC 的匹配结果，并汇报其 F1 分数和准确率。

Figure 6–7 Visualization of matching results on 20 PascalVOC Keypoint categories. The legend follows Fig. 5–1. Blue, orange, and red nodes denote matched inlier, unmatched inlier, and outlier. Blue lines represent correct and three types of red lines represent mismatching, ill-matching, and over-matching respectively. UPM, BBGM, and ZAC are eval at the same matching pair, where F1 score and accuracy are reported at the bottom.

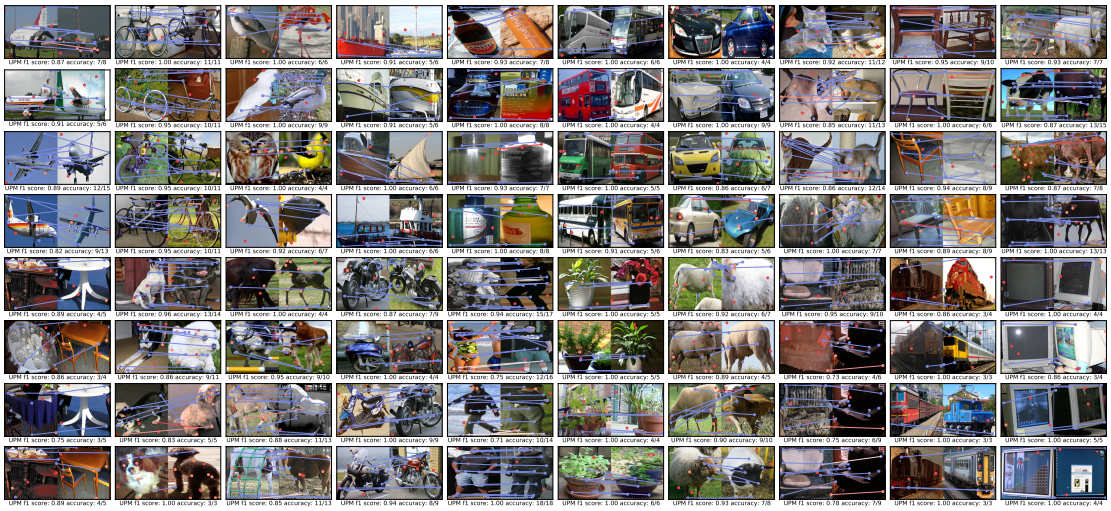


图 6–8 PascalVOC 上 UPM 在不同类型图上的可视化。图例与图5–1保持一致：蓝、橙、红色的点分别表示匹配内点、未匹配内点、外点。蓝线表示正确匹配，三种红线表示三种错误匹配。我们展现 UPM 对抗旋转、遮挡、大量外点的能力。

Figure 6–8 Visualization of UPM matching results on 20 PascalVOC Keypoint categories. The legend follows Fig. 5–1. Blue, orange, and red nodes denote matched inlier, unmatched inlier, and outlier. Blue lines represent correct and three types of red lines represent mismatching, ill-matching, and over-matching respectively. We show the success of UPM against rotation, occlusion, and massive outliers.

全文总结

本文主要研究了针对多种图匹配问题的机器学习算法。针对多图匹配聚类问题，本文基于最小化-最大化的条理化算法 M3C，协同解决图匹配与聚类问题。M3C 结合我们提出的松弛聚类指标，扩展了本问题的优化空间，并拥有理论上收敛性保证。同时，我们将 M3C 嵌入至非监督学习框架，提出非监督学习模型 UM3C，并设计基于边的相似度学习和伪标签选择策略。实验结果表明，非学习算法 M3C 达到和现有最优求解器 DPMC[20]一致甚至更好的水平，非监督学习模型 UM3C 大幅超过最优模型 GANN[21]，甚至在 WillowObject 中超过监督学习模型 NGMv2[25]和 BBGM[26]。同时我们还通过消融实验、收敛性测试、泛化性测试、超参数敏感性测试来检验 M3C 和 UM3C 各个设计部分的有效性和鲁棒性，证实了我们设计的算法在现实场景中的应用能力。

本文中，我们还将中间图这一概念首次引入机器学习图匹配模型。我们研究了图匹配问题中各节点类型的定义，并利用我们提出的中间图作为对节点进行类别区分的模板，提出了基于中间图的部分图匹配学习算法 UPM。受益于对中间图结构的设计，UPM 可以作为图匹配中所有任务的统一模型，并对所有匹配节点进行细粒度上的区分。在部分图匹配中，UPM 在所有任务下都取得了最优结果，其表现超过现有最优学习模型和非学习求解器达 3 – 10%，速度提升 8×。基于中间图视角提出的对比学习方法也能作为与前作正交的优化方法，帮助所有学习模型得到表现提升。我们还进一步通过实验说明了 UPM 对不同数目的未匹配内点和外点的鲁棒性，验证了对中间图构造设计的猜想，证明了中间图构造相似度超过二图相似度的性能，以及模型优秀的泛化性能和对超参数的容忍性。

如前文所言，我们的方法依然具有一定的局限性。我们提出针对多图匹配聚类问题的松弛聚类指标依然是离散的，我们希望能将其进一步松弛到连续空间，以拓展优化空间。现有的非监督学习方法依然受到传统原始相似度的影响，其在更有挑战性的 PascalVOC 上的表现不够理想。基于中间图的方法依然需要大量标注数据的参与，而这些数据的获取需要巨大的成本。UPM 模型取得了很高的匹配精度，但在召回率上依然有提升空间。以上几点，我们希望能够作为未来图匹配问题机器学习求解模型的设计方向，设计出高鲁棒性、低训练数据需求的机器学习图匹配算法。

机器学习模型正在向少数据、少监督、高鲁棒性方向前进，而现有图匹配问题的机器学习方法在这几方面仍在起步阶段。本文通过对图匹配问题在这几个维度的探索，提出了基于超图思想的松弛聚类指标和伪标签选择，基于边的相似度学习，中间图视角下的相似度学习，对比学习等方法。我们希望这些工作都能够为未来图匹配问题的机器学习求解方法提供新的视角和启发，并在下游任务中得到更广泛的应用。

参考文献

[1] SHEN D, DAVATZIKOS C. HAMMER: hierarchical attribute matching mechanism for elastic registration[J]. IEEE transactions on medical imaging, 2002, 21(11): 1421-1439.

[2] DUAN K, PARIKH D, CRANDALL D, et al. Discovering localized attributes for fine-grained recognition[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition. [S.l. : s.n.], 2012: 3474-3481.

[3] DEMIRCI M F, SHOKOUFANDEH A, KESELMAN Y, et al. Object recognition as many-to-many feature matching[J]. International Journal of Computer Vision, 2006, 69(2): 203-222.

[4] BERG A C, BERG T L, MALIK J. Shape matching and object recognition using low distortion correspondences[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05): vol. 1. [S.l. : s.n.], 2005: 26-33.

[5] PETTERSON J, YU J, MCAULEY J J, et al. Exponential family graph matching and ranking[C]//Advances in Neural Information Processing Systems. [S.l. : s.n.], 2009: 1455-1463.

[6] NAM H, HAN B. Learning multi-domain convolutional neural networks for visual tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2016: 4293-4302.

[7] IQBAL U, MILAN A, GALL J. Posetrack: Joint multi-person pose estimation and tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l. : s.n.], 2017: 2011-2020.

[8] SIMON I, SNAVELY N, SEITZ S M. Scene summarization for online image collections[C]//2007 IEEE 11th International Conference on Computer Vision. [S.l. : s.n.], 2007: 1-8.

[9] GOLD S, RANGARAJAN A. A Graduated Assignment Algorithm for Graph Matching[J]. IEEE Trans. Pattern Anal. Mach. Intell., 1996, 18: 377-388.

[10] CHO M, LEE J, LEE K M. Reweighted random walks for graph matching[C]//European conference on Computer vision. [S.l. : s.n.], 2010: 492-505.

[11] LEE J, CHO M, LEE K M. Hyper-graph matching via reweighted random walks[C]//CVPR 2011. [S.l. : s.n.], 2011: 1633-1640.

[12] LOIOLA E M, de ABREU N M M, BOAVENTURA-NETTO P O, et al. A survey for the quadratic assignment problem[J]. EJOR, 2007: 657-90.

[13] LAWLER E L. The Quadratic Assignment Problem[J]. Management Science, 1963, 9: 586-599.

[14] CHEN Y, GUIBAS L, HUANG Q. Near-Optimal Joint Object Matching via Convex Relaxation[C]//XING E P, JEBARA T. Proceedings of Machine Learning Research: Proceedings of the 31st International Conference on Machine Learning: vol. 32: 2. Bejing, China: PMLR, 2014: 100-108.

[15] YAN J, WANG J, ZHA H, et al. Consistency-driven alternating optimization for multigraph matching: A unified approach[J]. TIP, 2015, 24(3): 994-1009.

[16] 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, 2015, 38(6): 1228-1242.

[17] HUANG Q X, GUIBAS L. Consistent shape maps via semidefinite programming[C]//Computer Graphics Forum: vol. 32: 5. [S.l. : s.n.], 2013: 177-186.

[18] ZHOU X, ZHU M, DANIILIDIS K. Multi-image matching via fast alternating minimization[C]//Proceedings of the IEEE International Conference on Computer Vision. [S.l. : s.n.], 2015: 4032-4040.

[19] JIANG Z, WANG T, YAN J. Unifying Offline and Online Multi-graph Matching via Finding Shortest Paths on Supergraph[J]. TPAMI, 2021, 43(10): 3648-3663.

[20] WANG T, JIANG Z, YAN J. Clustering-aware Multiple Graph Matching via Decayed Pairwise Matching Composition[J]. AAAI, 2020.

[21] WANG R, YAN J, YANG X. Graduated Assignment for Joint Multi-Graph Matching and Clustering with Application to Unsupervised Graph Matching Network Learning.[C]//NeurIPS. [S.l. : s.n.], 2020.

[22] ZANFIR A, SMINCHISESCU C. Deep learning of graph matching[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2018: 2684-2693.

[23] WANG R, YAN J, YANG X. Learning combinatorial embedding networks for deep graph matching[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. [S.l. : s.n.], 2019: 3056-3065.

[24] WANG T, LIU H, LI Y, et al. Learning Combinatorial Solver for Graph Matching[C]//CVPR. [S.l. : s.n.], 2020: 7568-7577.

[25] WANG R, YAN J, YANG X. Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[26] ROLÍNEK M, SWOBODA P, ZIETLOW D, et al. Deep graph matching via blackbox differentiation of combinatorial solvers[C]//European Conference on Computer Vision. [S.l. : s.n.], 2020: 407-424.

[27] YU T, WANG R, YAN J, et al. Deep Latent Graph Matching[C]//International Conference on Machine Learning. [S.l. : s.n.], 2021: 12187-12197.

[28] DYM N, MARON H, LIPMAN Y. DS++ a flexible, scalable and provably tight relaxation for matching problems[J]. ACM Transactions on Graphics (TOG), 2017, 36(6): 1-14.

[29] BERNARD F, THEOBALT C, MOELLER M. Ds*: Tighter lifting-free convex relaxations for quadratic matching problems[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2018: 4310-4319.

[30] SWOBODA P, ROTHER C, ABU ALHAIJA H, et al. A study of lagrangean decompositions and dual ascent solvers for graph matching[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2017: 1607-1616.

[31] WANG Q, ZHOU X, DANIILIDIS K. Multi-image semantic matching by mining consistent features[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2018: 685-694.

[32] PACHAURI D, KONDOR R, SINGH V. Solving the multi-way matching problem by permutation synchronization[C]//Advances in neural information processing systems. [S.l. : s.n.], 2013: 1860-1868.

[33] YU T, YAN J, LIU W, et al. Incremental multi-graph matching via diversity and randomness based graph clustering[C]//Proceedings of the European Conference on Computer Vision (ECCV). [S.l. : s.n.], 2018: 139-154.

[34] YAN J, YIN X C, LIN W, et al. A Short Survey of Recent Advances in Graph Matching[C]//ICMR. [S.l. : s.n.], 2016.

[35] YAN J, YANG S, HANCOCK E. Learning Graph Matching and Related Combinatorial Optimization Problems[C]//IJCAI. [S.l. : s.n.], 2020.

[36] KRAHN M, BERNARD F, GOLYANIK V. Convex Joint Graph Matching and Clustering via Semidefinite Relaxations[C]//2021 International Conference on 3D Vision (3DV). [S.l. : s.n.], 2021: 1216-1226.

[37] NG A Y, JORDAN M I, WEISS Y. On spectral clustering: Analysis and an algorithm[C]// Advances in neural information processing systems. [S.l. : s.n.], 2002: 849-856.

[38] HARTMANIS J. Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson)[J]. Siam Review, 1982, 24(1): 90.

[39] POLJAK S, RENDL F. Solving the max-cut problem using eigenvalues[J]. Discrete Applied Mathematics, 1995, 62(1-3): 249-278.

[40] TREVISAN L. Max cut and the smallest eigenvalue[J]. SIAM Journal on Computing, 2012, 41(6): 1769-1786.

[41] GOEMANS M X, WILLIAMSON D P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming[J]. Journal of the ACM (JACM), 1995, 42(6): 1115-1145.

[42] KAPPES J H, SWOBODA P, SAVCHYNSKYY B, et al. Multicuts and perturb & map for probabilistic graph clustering[J]. Journal of Mathematical Imaging and Vision, 2016, 56(2): 221-237.

[43] SWOBODA P, ANDRES B. A message passing algorithm for the minimum cost multicut problem[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l. : s.n.], 2017: 1617-1626.

[44] WANG R, YAN J, YANG X. Combinatorial Learning of Robust Deep Graph Matching: an Embedding based Approach[J]. IEEE TPAMI, 2020.

[45] ZHANG Z, LEE W S. Deep Graphical Feature Learning for the Feature Matching Problem[C]//ICCV. [S.l. : s.n.], 2019: 5087-5096.

[46] YU T, WANG R, YAN J, et al. Learning deep graph matching with channel-independent embedding and Hungarian attention[C]//Int. Conf. Learn. Represent. [S.l. : s.n.], 2020.

[47] FEY M, ERIC LENSSEN J, WEICHERT F, et al. Splinecnn: Fast geometric deep learning with continuous b-spline kernels[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l. : s.n.], 2018: 869-877.

[48] POGANČIĆ M V, PAULUS A, MUSIL V, et al. Differentiation of blackbox combinatorial solvers[C]//International Conference on Learning Representations. [S.l. : s.n.], 2019.

[49] YING Z, WANG A, YOU J, et al. Neural subgraph matching[J]. ArXiv preprint arXiv:2007.03092, 2020.

[50] BAI Y, XU D, SUN Y, et al. GLSearch: Maximum Common Subgraph Detection via Learning to Search[C]//International Conference on Machine Learning. [S.l. : s.n.], 2021: 588-598.

[51] DELAUNAY B, et al. Sur la sphere vide[J]. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, 1934, 7(793-800): 1-2.

[52] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//ICLR. [S.l. : s.n.], 2014.

[53] KOOPMANS T C, BECKMANN M. Assignment problems and the location of economic activities[J]. Econometrica, 1957: 53-76.

[54] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[55] ZHOU F, TORRE F. Factorized Graph Matching[J]. IEEE PAMI, 2016.

[56] CHO M, ALAHARI K, PONCE J. Learning Graphs to Match[C]//ICCV. [S.l. : s.n.], 2013.

[57] GRIFFIN G, HOLUB A, PERONA P. Caltech-256 Object Category Dataset[R]. [S.l.]: California Institute of Technology, 2007.

[58] EVERINGHAM M, VAN GOOL L, WILLIAMS C K, et al. The PASCAL visual object classes challenge 2007 (VOC2007) results[J]., 2007.

[59] MANNING C D, SCHÜTZE H, RAGHAVAN P. Introduction to information retrieval[M]. [S.l.]: Cambridge University Press, 2008.

[60] RAND W M. Objective criteria for the evaluation of clustering methods[J]. Journal of the American Statistical association, 1971, 66(336): 846-850.

[61] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 2016: 770-778.

[62] KUHN H W. The Hungarian method for the assignment problem[C]//Export. Naval Research Logistics Quarterly. [S.l. : s.n.], 1955: 83-97.

[63] CHEN T, KORNBLITH S, NOROUZI M, et al. A Simple Framework for Contrastive Learning of Visual Representations[J]. ArXiv preprint arXiv:2002.05709, 2020.

[64] XIE Z, LIN Y, ZHANG Z, et al. Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning[Z]. 2021. arXiv: 2011.10043 [cs.CV].

[65] EVERINGHAM M, GOOL L, WILLIAMS C K, et al. The Pascal Visual Object Classes (VOC) Challenge[J]. IJCV, 2010, 88(2): 303-338.

[66] BOURDEV L D, MALIK J. Poselets: Body part detectors trained using 3D human pose annotations[J]. 2009 IEEE 12th International Conference on Computer Vision, 2009: 1365-1372.

[67] WANG T, LING H, LANG C, et al. Graph matching with adaptive and branching path following[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2017.

[68] WANG F, XUE N, YU J G, et al. Zero-assignment constraint for graph matching with outliers[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. [S.l. : s.n.], 2020: 3033-3042.

[69] CHEN L, LIN G, WANG S, et al. Graph Edit Distance Reward: Learning to Edit Scene Graph[J]. European Conference on Computer Vision, 2020.

致　谢

岁月不居，时节如流，大学四年，转眼已至尾声。因为疫情，我无法好好和自己的本科作别，只能在漂泊中写下此文，心中感慨万千。

感谢 ACM 班，感谢 ACM 班的同学、老师、学长学姐们。我很少将自己联结于一个抽象的集体，但在 ACM 班的这四年，却让我感到了无比的充实和幸福。我能拥有如此多优秀的同学，我与他们共享课堂，与他们在交流问题时享受思维的碰撞。我和他们组队大作业，也曾在夜深时回味他们的小讲堂。ACM 班的每个人，你们的热情和才华，照亮了我四年的大学生活，我希望能永远做大家的“陆姐姐”。也要感谢 ACM 班的学长学姐们，你们不求回报的指点和鼓励，让我也希望能做一个尽力帮助学弟学妹的人。

感谢我的导师们，教我待人处世，也教我审慎思辨。你们严肃而不失亲切的指点，将让我终生受益。

俞勇老师，您从海量的申请者中将我选入 ACM 班。四年来，每一次与您的谈话，我都铭记在心。您鼓励我，肯定我，教会我坚持，亦教育我如何成大事。您说我有“一种巨大的能量”，因为您相信我，我也愿意相信自己，能在未来的每一天，都比过去更好一些。

感谢翁惠玉老师，作为我们班主任的您记忆中一直是微笑的。无论是在您的办公室，还是做程设助教时的机考夜，我总是能毫无保留地向您倾诉我的快乐和悲伤。您也总是愿意耐心聆听，为我们的学习与生活提出建议。您的热情，就像黑暗中的炬火。即使我离开了大学校园，也依然会感念您给予我的光亮。

感谢严骏驰老师，您是我的第一位科研导师。您让我安全而自由地成长，带领我体会到了科研探索最初的乐趣，也在我迷茫时给予了可靠的帮助。您不遗余力地指导我科研的方方面面，大到探索的门径，小到论文的措辞。当我在跌跌撞撞地探索科研之路时，您就是天顶的北极星。我只要一仰头，就能望见您为我指明的方向。您将我推荐给黄其兴老师，最终也成为我博士生涯的去处。成为您的学生，我何其有幸。

感谢张宁老师在我毕业设计期间给予的帮助与指导。

感谢我的合作者，蒋泽天和王天哲学长。我们不仅成为科研上的合作者，更是生活里的好朋友。我们并肩走过了漫漫长路，不断地讨论问题、检查代码，甚至推倒重来，我不会畏惧，因为我知道有你们可依靠。感谢 ThinkLab 的朋友们。会议截稿后的深夜，回寝室的路上空无一人，我们嬉笑打闹的声音，能惊动头顶的星星。愿我们的论文早日发表，也愿我们的友谊之树长青。

感谢我的家人们。我的母亲，她看起来普通，但其实极其聪慧。虽然不懂我的方向，她总是无条件地支持我的选择、尊重我的意见，给我提供最舒适的生活环境。我的父亲，他是我在编程上的启蒙者。虽然我早早超出了他的水平，依然感谢他带我将双手放上键盘的那一刻。父母总是把我当成孩，但现在，我很幸运，能让他们看到孩子要飞向更高更远的地方。愿我们平安、幸福、健康，拥有快乐和自由。

感谢共同走过四年的女孩子们，傅凌玥、李云卿、王一、吴怀瑾。我们或许不是所谓的“大多数”，但你们赋予我的情感，就像一道铠甲，让我有披荆斩棘的勇气。女孩子们的感情是世界上最美妙的情感，你们是多么的鲜活明快。在此之前，我总是不免感到孤独。直到遇见你们，和你们并肩同行、谈天说地，我才感到灵魂有了居处。或许未来我们会走上不同的道路，但这里绝不是终点，期待我们的远方还能有彼此的身影。祝福你们走出自己的精彩，

我们一起去打破那些无形的桎梏，去大声地说，大胆地做。

感谢陪伴我的朋友们，在我十六年的求学生涯中，你们帮助我走到今天，我们共享欢笑与泪水，成功与失败。特别是厉千珞，我十六年来最好的朋友，我们一路走来，曾经为自己的梦想挣扎，而如今都已走回自己当初认为的正轨，祝你我都能在我们热爱的事业里勇敢地向上走。虽无法列出其他所有人的名字，我也祝福所有的朋友们，我爱你们，愿你们也心有所爱。

感谢电影，感谢艺术，感谢美。偶尔，我的生活会落入灰暗，但是电影与艺术，让我保持对美好的感知。“电影被发明后，人类的生命至少延长了三倍。”我在其中感受喜怒哀乐，电影让我看到人间百态，让我知道路走向何方。感谢 800 号电影社，让我认识一群分享爱好的朋友们，我会永远怀念那些夜宵与酒。美是我永远的追求，我爱美好的事，美好的物，美好的人。感谢泰勒的歌，蒂莫西的表演，达米恩与瓦尔达的影像，我无法与你们相见相识，但拥有这些美好，你们如在身边。

最后，我想感谢我自己。从写下第一行 ‘hello world’ 已过去十几年，这一路上的故事大多已经记忆不清，但此程绝非坦途。经历过看不透的知识点，经历过滑铁卢，更是经历过“女孩子太辛苦了”的拷问。我不知道如今所成是否能够得上反击这些质疑，但自觉已足够精彩。我感谢那个从未放弃的自己，感谢那个想着“不如试一试”的自己，感谢那个在自卑与自傲间走钢丝的自己，感谢那个倔强的不服输的自己。我有无数次可能滑向深渊，太多大风大浪，摇摇摆摆，终究到达彼岸。我如同爱乐之城里最后的姑娘，与那个挣扎的流泪的迷茫的自己相视一笑。

如今的我一路游荡，最终在 ACM 班这个集体中找到了根。接下来又将再度启程，走向更广阔的远方。他们问我远方会怎样——

远方如日光般熠熠生辉。

MACHINE LEARNING METHODS ON GRAPH MATCHING

Graph matching (GM) has been a standing problem in pattern recognition and computer vision and found wide applications in image registration[1], recognition[2-3], 3D shape matching[4-5], and structure from motion[8]. In general, it refers to finding consistent node and edge correspondences between two or multiple graphs by maximizing the resulting affinity scores. The standard formulation of graph matching problem is the quadratic assignment problem (QAP), whose affinity matrix encodes second-order[9-10] or even higher-order[11] information of the graph-structured data. A popular extension for classic two-graph matching (2GM) is multiple graph matching (MGM)[14-19] where cycle consistency[15, 17] is often enforced. GM and MGM are all NP-hard and approximate algorithms are proposed over the decades[34-35].

The above two settings commonly assume the input graphs are of the same category, while in the real-world application, acquiring graph labels is time-consuming and some cases are naturally of mixture modes. Mixture graph matching and clustering (MGMC) is therefore proposed to align graph-structured data and divide them into different clusters at the same time. Two recent works[20-21] have explored this setting, but they suffer from some drawbacks. 1) Convergence. DPMC cannot converge stably while GANN does graph matching and graph clustering only once. There are no theoretical convergence guarantees of their joint optimization. 2) Rigid Structure. DPMC and GANN heavily rely on the tree structure and cluster algorithm, respectively. 3) Robustness. GANN is vulnerable to noise. The model hardly works when outliers increase. All these hinder them from achieving better performance and real-world application.

Machine learning methods have shown their potential in pushing the frontier of GM for real-world image data[22-27]. Benefiting from the power of convolutional neural networks (CNNs) and graph neural networks (GNNs), expressive visual and structural features are extracted from the source data. These methods assume that all nodes of one graph can find their correspondences in the other graph, which is considered a bijective mapping. However, data collected from real-world often break such assumptions because of rotation, occlusion, and viewpoint difference, as well as the noisy hand-annotated (or algorithm-generated) labels. It, therefore, forms a partial matching which is a more challenging case and has not been fully explored. Moreover, current methods in machine learning of graph matching mostly focus on 2GM and supervised learning. They can hardly yield the power of multiple graphs and underlying cycle consistency. Neither can they deny matching two graphs of different categories.

In this paper, we propose to address the above issues in machine learning methods of graph matching problems in the following directions:

1) Propose algorithms with convergence guarantee, flexible structure, and high robustness in mixture graph matching and clustering problems.

2) Design a QAP-based unsupervised learning pipeline that can train on unlabeled data to handle the MGMC problem.

3) Develop a machine learning mechanism for the partial matching problem.

4) Unify 2GM, MGM, MGMC, and online matching problems under the same machine learning model.

To tackle the first two challenges, we first propose a Minorize-maximization based framework algorithm with construction-step and maximization-step to iteratively tackle the clustering and matching problem. Its convergence is proven under our problem formulation. We then point out the initial hard clustering indicator of this framework is too rigid and converges quickly to a sub-optimal. It is fixed when given the target cluster size. That is, the construction-step only explores the size of each cluster, and once the cluster sizes are fixed, the cluster indicator converges within one step. The fixed indicator limits the optimization space of the maximization-step and the matching results can not fundamentally improve the clustering quality. Therefore, we propose to relax the hard constraints on the original cluster indicator. We observe that the essence of the original hard constraint is to find a certain amount of graph pairs (which is limited by strict cluster division) with a high average affinity score. Based on this observation, we provide two relaxed constraints to replace the transactive relationship in hard clustering and induce the relaxed cluster indicator. The above framework together with our relaxed indicator forms the algorithm M3C. We show how this algorithm works with initialization, construction-step to construct relaxed indicator, and matching-step to utilize supergraph-based optimization. Specifically, three ranking schemes, global-rank, local-rank, and bi-rank are used to construct relaxed cluster indicator. MGM-Floyd is modified to fit into the maximization-step for its cost-efficiency.

Based on M3C, we propose UM3C as an unsupervised learning pipeline to learn image features and graph structure to solve the second challenge. In the previous work, GANN[21] applies the formation of Koopmans-Beckmann's QAP[53], which only focuses on node affinity and graph structure alignment. UM3C expands the formulation into a more general setting, Lawler's QAP[12]. We add edge-wise affinity learning and edge-wise affinity loss to directly learn the affinity matrix $\mathbf{K}$. Compared to other learning pipelines where the loss function is often applied between the prediction matching and the ground truth, our affinity loss excludes the effect of the solver and only focuses on the quality of affinity, making the affinity robust and suitable for different kinds of solvers. Moreover, we also optimize the pseudo label pair selection for loss calculation under the guidance of the relaxed cluster indicator. Instead of using all the pairs in the cluster, UM3C follows the relaxed cluster indicator and selects the graph pairs with higher affinity rank as pseudo labels. This scheme brings the pseudo affinity label closer to the ground truth.

We test the performance of M3C and UM3C on WillowObject in the experiment chapter. M3C achieves competitive results as the previous learning-free solvers and has the most balanced performance with outliers. In the more complex settings, it notably outperforms peer methods in both matching and clustering accuracy. We also verify the superiority of our proposed ranking schemes by comparing M3C with the baseline hard clustering method M3C-Hard. UM3C significantly outperforms GANN in all experiments and maintains a good matching and clustering accuracy when increasing cluster size, graph, and outlier numbers. Though trained under a simpler setting, it can also be deployed into a more complicated scenario and achieve satisfying performance. We also conduct several ablation studies, convergence tests, generalization tests, and hyperparameters sensitivity studies to demonstrate how each proposed component works upon each other. All these promising results show that our methods achieve our goal of convergence, robustness, and flexible structure.

We address the next two challenges by introducing a 'universe' perspective into machine learning methods for graph matching problems. It first match the node in a certain instance to a universe graph (instead of each other), then reconstruct the pair-wise matching. Under the universe perspective, we define three node types on a fine-grained level: matched inlier, unmatched inlier, and outlier. We point out that the previous pairwise graph matching method can be easily confused by matched inlier and outlier, and can't learn the underlying semantic meaning of these nodes. We, therefore, introduce UPM, a learning pipeline with two core components: universe metric learning and outlier-aware loss. Universe metric learning helps to learn the universe affinity and construct the virtual universe graph, which serves as a bridge for inliers and helps to avoid ill-matching. We also discuss two types of universe graph construction to make it adapt to different application settings. Outlier-aware loss serves to cluster out all the outliers and increase the robustness of our method. Besides these improvements, we propose a contrastive learning scheme based on this universe perspective to refine the feature extractor. We show that it is orthogonal to all the previous work and thus can be easily applied.

Moreover, we further analyze UPM's transferability to more graph matching tasks. Since the cycle consistency is automatically satisfied with the universe graph, we unify both 2GM and MGM under this framework. The structure of the universe graph also helps to reject matching between nodes from different types of graphs. The time and space efficiency also make UPM an optimal choice for online matching. These merits all make UPM a unified model for graph matching.

In the experiment chapter, we test UPM's ability in the face of massive unmatched inliers and outliers on PascalVOC and WillowObject. It significantly outperforms previous work by 3-10% in our main test and the gap grows even bigger when the setting becomes more complex. The pre-trained feature extractor using contrastive learning also helps BBGM achieve a better performance, which means our universe perspective is also illuminating for other works. In our test, UPM can distinguish node types better than BBGM, which helps to achieve higher precision in the matching. We detailed analyze the universe structure and universe affinity generated by UPM and conform our conjectures on their characters. Generalization tests and hyperparameter sensitivity tests also prove the robustness of our design. We further compare UPM on MGMC and the online matching settings. Results demonstrate the great performance of UPM in both accuracy and efficiency. Detailed visualization is also presented in this paper.

In conclusion, this paper successively designed learning-free algorithm M3C, unsupervised learning model UM3C, and universe-based pipeline UPM for four challenges listed above. They all achieve state-of-the-art performance in our tests with high robustness, efficiency, and flexible design structure. M3C and UM3C push the frontier of MGMC and allow for further exploration in the unsupervised learning model of graph matching problems. UPM serves as a unified model for all the tasks in graph matching and can distinguish node types at a fine-grained level. It significantly improves the machine learning methods in the partial matching setting. We hope these works could provide new perspectives and inspirations for machine learning methods in graph matching problems, especially for robustness, fewer labeled data, and more general settings, and provide new opportunities for downstream applications.