



東南大學
SOUTHEAST UNIVERSITY

复杂系统场景数据生成理论与技术

康明与^{*}

kangmingyu.china@gmail.com

东南大学网络空间安全学院

2024年12月28日

目录

摘要	I
Abstract	II
符号约定	III
第一章 绪论	1
1.1 研究目的与意义	1
1.2 问题描述及研究现状	3
1.3 研究内容与创新点	6
1.4 本文组织结构	9
第二章 基于系统辨识的场景数据生成	11
2.1 数据驱动系统辨识	11
2.1.1 稀疏辨识	11
2.1.2 系统基函数的选择	15
2.1.3 人工神经网络	16
2.1.4 神经微分方程	18
2.1.5 物理信息神经网络	20
2.2 复杂系统的网络表示	22
2.2.1 从人工神经网络到复杂网络	22
2.2.2 复杂网络协同性和可控性	24
2.2.3 复杂网络弹性和低秩性	26
2.3 复杂网络动力学系统辨识	28
2.3.1 ASIND 算法原理	28
2.3.2 实验设置	30

2.3.3 实验结果	31
第三章 基于生成模型的场景数据生成	33
3.1 生成模型	33
3.1.1 双样本问题.....	33
3.1.2 变分自编码器	35
3.1.3 生成对抗网络	36
3.1.4 生成流模型.....	38
3.1.5 生成扩散模型	39
3.1.6 得分生成模型	42
3.2 多模态场景数据生成.....	43
3.2.1 条件场景数据生成	43
3.2.2 多模态融合表示和跨模态表示转换	46
3.2.3 专用多模态特征提取.....	48
3.2.4 通用多模态特征提取.....	52
3.2.5 基于生成式大模型的多模态场景数据生成	54
3.3 基于跨时空模态生成对抗网络的场景数据生成框架	56
3.3.1 跨时空模态生成对抗网络原理.....	57
3.3.2 时空 Transformer 原理	57
3.3.3 跨时空模态场景数据生成算法	59
3.3.4 实验设置	59
3.3.5 实验结果	61
3.4 基于跨时空模态生成对抗网络的完全数据缺失插补技术	63
3.4.1 跨时空模态生成对抗网络与完全数据缺失问题的联系	64
3.4.2 时空模态融合表示	65
3.4.3 跨时空模态完全数据缺失插补算法	66
3.4.4 实验设置	67
3.4.5 实验结果	68

第四章 基于因果推断的场景数据生成	71
4.1 反事实推理	71
4.1.1 潜在结果模型	71
4.1.2 潜在结果模型的基础假设	73
4.1.3 结构因果模型	74
4.1.4 贝叶斯网络与因果网络	76
4.1.5 贝叶斯网络的基础假设	77
4.2 潜在结果模型和结构因果模型的根本分歧	78
4.2.1 可忽略性和后门准则	78
4.2.2 反事实可证伪性	80
4.2.3 直接因果和间接因果	81
4.3 复杂系统时空因果推断	83
4.3.1 贝叶斯网络中的马尔可夫等价关系	83
4.3.2 时空贝叶斯网络	84
4.3.3 时空因果等价性	85
4.3.4 时空因果唯一性	88
4.4 基于高阶因果熵的唯一因果网络发现	91
4.4.1 高阶因果熵算法原理	91
4.4.2 实验设置	92
4.4.3 实验结果	93
4.4.4 神经高维高阶因果熵估计	94
第五章 总结与展望	96
5.1 本文总结	96
5.2 未来研究展望	97
参考文献	98
致谢	118

摘要

本文旨在促进人工智能基础理论与安全技术的发展。本文针对复杂系统场景数据生成问题展开研究，研究基于系统辨识、生成模型和因果推断等三类复杂系统场景数据生成方法。本文从数据分布偏移（一致）这一问题，沿着系统辨识、生成模型和因果推断这一主线展开论述，形成一整套复杂系统场景数据生成理论框架和技术方法论，并最终论证从观测场景数据中外推数据的可行性。

关键词： 场景数据生成，复杂系统，系统辨识，生成模型，因果推断。

ABSTRACT

The goal of this work is to advance the field of artificial intelligence and cyber security. This work conducts research on scenarios generation in complex system, and addresses the issue of data distribution shift with the tools of system identification, generative model and causal inference. This work proposes a more generalized framework to encompasses the fields of system identification, generative model and causal inference, and finally demonstrates the feasibility of extrapolating data from the observed scenarios data.

Keywords: Scenarios generation, complex system, system identification, generative model, causal inference.

符号约定

\emptyset	空集
$A \setminus B$	差集
\cup	并集
\cap	交集
\mathbb{R}	实数
\mathbb{N}	自然数
\mathbb{R}^+	正实数
\mathbb{Z}	整数
\mathbb{Z}^+	正整数
$[a, b]$	闭区间
(a, b)	开区间
$[\mathbf{a}, \mathbf{b}]$	矢量或矩阵拼接
a.e.	almost everywhere
a.s.	almost surely
s.t.	subject to
$ \cdot $	绝对值或集合的元素个数
$\ \cdot\ $	L_2 范数
$\ \cdot\ _p$	L_p 范数
\sup	上确界
\inf	下确界
\odot	哈达玛积
∇	梯度算子
$\text{diag}(\mathbf{a})$	对矢量 \mathbf{a} 对角化成矩阵
$\det(A)$	矩阵 A 的行列式值
$\text{tr}(A)$	矩阵 A 的迹

第一章 绪论

1.1 研究目的与意义

本文旨在促进人工智能基础理论与安全技术的发展。自 ChatGPT 于 2022 年发布以来，生成式大模型技术走向商用并迅速发展。至今，同类的国产生成式大模型（如百度文心一言和讯飞星火等）也迅速跟进，并获得了极大的发展。以数据、算法和算力为核心的新一代人工智能技术助力人类社会工业生产自动化，并逐步解放和发展新质生产力，实现社会生产力的整体跃升。但是，这同时也带来了新的时代问题，包括对高质量无污染数据的需求，对可解释可扩展且安全可靠的算法的需求，以及对强大且可持续的底层算力支持的需求。在《国家人工智能产业综合标准化体系建设指南（2024 版）》^[1] 中，人工智能产业链主要划分为基础层、框架层、模型层、应用层等四个层级。如图 1.1 所示，越往高层则越是面向具体的应用场景，而越往低层则越是面向技术基础。因而如果想要继续拓宽应用层面的视野，以及激发对下一代人工智能技术的想象力，则需要继续在数据、算法和算力等基础技术研究上有所突破。

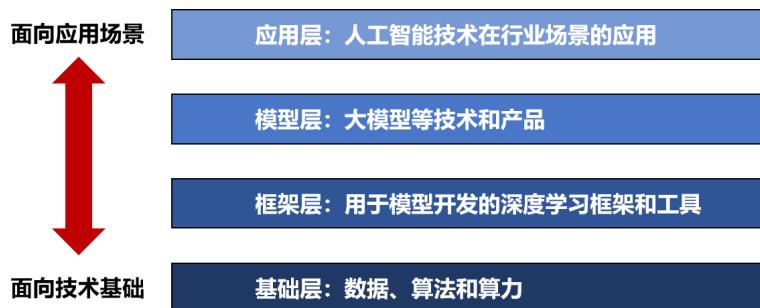


图 1.1: 人工智能产业链层级示意图

回顾人工智能的发展史，自 A. Turing^[2] 提出了“图灵测试”，并以此作为判断机器是否具备智能的标准，人工智能的概念被首次提出。此后，达特茅斯会议^[3] 确立了七个人工智能议题，也即自动计算机、如何对计算机进行编程以使用语言、神经网络、计算规模理论、自我改进、抽象、随机性与创造性，人工智能被确立为一个独立的研究领域。

再之后的人工智能技术发展大致可以分为两个阶段。第一个阶段是以符号主义观点为核心的人工智能的发展，也即认为人工智能依赖于知识，而知识的基本元素是符号及其操作规则（物理符号系统假设^[4-6]），因此，符号人工智能也被称为基于知识驱动的人工智能。它通常需要一组符号，以及操作符号的规则，研究方法则是通过将符号和操作规则设计并组合成物理逻辑系统，或编入计算机程序，从而实现对人脑功能的模拟。但是，组成符号系统的知识通常来自于特定领域，因而难以在差异巨大的不同领域之间泛化。并且，符号和操作规则在一开始就依据领域知识固定下来，如果出现超出知识的情况，则该系统通常难以正确应对，即便后来引入了概率推理以解决不确定性问题^[7]，该泛化问题依然难以从根本上解决。

第二个阶段是以连接主义观点为核心的人工智能的发展，也即利用人工神经网络构建起“激励 - 响应”的连接，而这些连接促成了智能的形成。基于此观点，W. S. McCulloch 和 W. Pitts^[8] 提出了基于阈值逻辑的神经元数学模型（M-P 神经元模型）。D. O. Hebb^[9] 提出了 Hebb 学习率，也即同时激活 (fire) 的神经元

其连接会得到加强。基于此二者，F. Rosenblatt^[10] 提出了感知机模型，它是人工神经网络的雏形。此后，S. Linnaismmaa 首先提出反向传播算法^[11]，后续又由 D. E. Rumelhart, G. E. Hinton 和 R. J. Williams^[12] 做了系统性地分析和验证，从而将感知机模型推广到多层。再由 K. He 等人^[13,14] 提出残差网络模型，系统分析了残差结构对于解决深层人工神经网络退化问题的作用，才真正将多层感知机模型推广到深层，这时候人工神经网络正式开启了深度学习时代，并持续至今。值得注意的是，深度学习本身不依赖于知识（符号和操作规则），而是依赖于多模态的感官信息，如视觉、听觉、嗅觉、味觉、触觉等感官接收到感官信息。这些感官信息在计算机中以多模态数据的形式呈现，因此相对于符号人工智能，它也被称为基于数据驱动的人工智能。

然而，深度学习要具备泛化性必须满足以下假设：(1) 多样性假设：人工神经网络所能表示的函数族足够多样以涵盖所要表示的函数；(2) 充分性假设：用于训练人工神经网络参数的样本数据能够充分地捕捉总体的结构和特征。多样性假设要求人工神经网络的参数量要足够大，无论是深度的，或是浅层但宽的，从而使得人工神经网络具备相应的表示能力 (expressive power) 以逼近任意的函数^[15-19]。充分性假设要求用于训练的样本数据要足够充分，从而使得人工神经网络能够正确捕捉总体的结构和特征，而为满足这一条件，则通常要求样本数据量趋近于无限^[20]。这种无限样本的要求在实际应用中是无法满足的。尤其是在复杂系统中，由于不适定问题^[21,22] 的存在，求解的人工神经网络参数往往对样本数据的微小变化十分敏感，从而导致求解的结果差异巨大。

因此，针对下一代人工智能应具备怎样的特性，主要存在着如下不同的观点。其一是将知识和数据结合，采用“大数据、大模型、大算力”模式解决人工智能算法在复杂场景下的泛化性问题^[23-25]。由于近年来，国内外互联网技术和数据存储技术的飞速发展，使得来自于不同领域的大量数据快速积累，从而使现有的深度学习应用一定程度上满足了对训练数据的充分性假设。同时，由于计算机并行计算技术（主要是指 NVIDIA 为代表的 GPU 技术和华为海思为代表的 NPU 技术）也同步发展，因而使得深度学习应用可以支持拥有巨大参数量的人工神经网络的参数训练，从而使其满足了对函数表示的多样性假设。基于此，诞生了以 ChatGPT 为代表的大语言模型，或者更笼统地称为大人工智能模型 (Foundation Models^[25])。然而，此类模型并没有真正克服深度学习的固有缺陷。当此前积累的领域知识、训练数据和算力达到可利用的极限，其很快就面临如何可持续地获得领域知识、训练数据和算力的新问题。

其二是引入协同交互机制，采用“人、机、环境协同交互”模式解决人工智能在动态开放环境中所面临的数据分布偏移问题^[26-30]。该观点暗含了样本数据的充分性假设不可能满足，并指出通过为人工智能增加反馈机制，使其可以不断与人和环境交互，从而一方面适应开放环境的动态变化，另一方面适应人对人工智能指示的变化，其适应过程采用单/多智能体自适应的方式实现。基于此，世界模型 (World Models)^[26,27] 被提出，其主要采用自监督学习和强化学习以实现多智能体的协同交互机制，同时采用层级化表示技术以实现智能体间知识和数据的共享。同类技术还应用在以百度 Apollo 为代表的自动驾驶解决方案，其提供的萝卜快跑自动驾驶出行服务目前已得到实际的技术落地。但是，该技术仍然有其局限性。该技术要求智能体能够在不断地交互中，通过反馈机制更新模型参数。但是在一些具有高安全性和高可靠性要求的应用场景（如电力网络和通信网络）中，则不允许如此地反复“试错”。因而这就要求智能体在一开始就具备足够的智能，但这种智能又依赖于知识和数据，所以这又回到了第一个观点所面临的问题。

其三是引入因果机制^[31]。不同于以上，前二观点最终会落脚在知识和数据层面，其中的“数据”来自于感官信息，或者称为观测数据。而因果观点不仅关注观测数据，而且关注未观测的反事实，其对于反事实的定义可以追溯到统计学中的潜在结果模型^[32]。因果观点主要包含两个主张：(1) 推断出的因果关系可以作为知识指导模型自主设计，从而实现模型泛化；(2) 反事实思考可以是封闭的，不需要与开放环境交互。因此，因果观点既主张人工智能能够自主地学习知识，从而解决观点一中对外部获取知识的依赖。同时又主张通过人工智能的反事实思考回避与开放环境的交互，从而解决观点二中的“试错”成本的问题。尽管如此，因果观点在具体实践上其实并没有形成一个统一的方法论，就目前两个主要框架（潜

在结果模型^[32,33] 和结构因果模型^[34,35]) 来看，其所依赖的基础假设并不完全相同，因而其推理结果往往具有不同的解释。

因此，本文撰写时正处在人工智能技术飞速发展，且亟需思考其理论基础和未来技术走向的时间点。本文将聚焦复杂系统场景，沿着从复杂系统辨识到生成模型，再到因果推断的主线，集中论述如何利用更少的外部知识实现复杂系统场景数据生成等一系列问题。

1.2 问题描述及研究现状

在讨论场景数据生成问题之前，本文首先需要明确一些概念。首先，场景数据是（计算机）人工智能对真实世界的描述，其获得的方式可以来自于对真实世界的多模态感知，也可以来自于多模态的场景数据生成过程。因此，如图 1.2 所示，人工智能通过构造生成器 g ，实现从场景数据 x 到场景数据 y 的生成。生成过程中可以融合外部知识 g' ，外部知识是已经经过构造和参数训练的生成器，“外部”是相对该次生成过程而言。外部知识的融合过程可以是该生成器 g 的结构设计过程，也可以是一些额外的信息编码过程。比如，在结构反事实推理技术^[34,35] 中，往往需要假设变量间服从某种因果关系，也即一个因果网络，然后才能进行反事实推理。类似的，网络动力学辨识技术^[36,37] 通常需要对参数模型作出假设，也即假设该系统服从网络动力学，然后才可以辨识网络动力学模型参数。这种假设来自于领域知识，领域知识显然并不是无源之水，而是来自于人脑中类似的“场景数据生成”过程。因此在图 1.2 中，这部分隐藏的场景数据也用虚线框表示了出来。当生成器 g 得以构造，则可以进而利用其生成场景数据 y ，并进而重构出认知世界。当场景数据 x 和 y 满足某种一致性时，则可以认为认知世界与真实世界一致。基于此，本文定义场景是一族场景数据的综合，用于描述人工智能所感知和认知的真实世界。

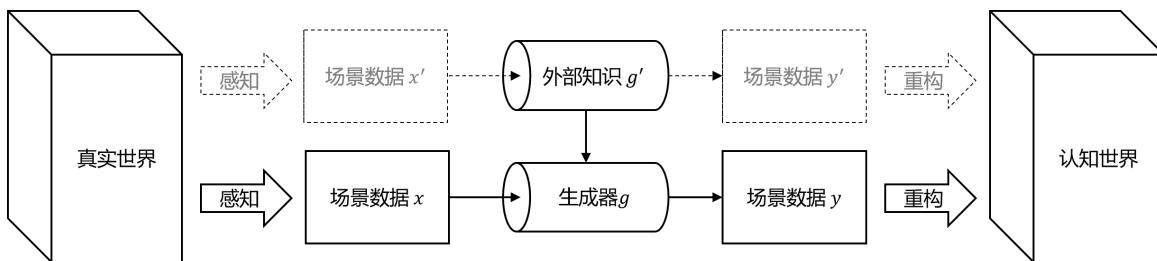


图 1.2: 场景数据生成示意图

然后本文定义数据分布一致和偏移，如下：

定义 1(数据分布一致和偏移). 在图 1.2 中，如果 $P(x) \neq P(y)$ ，则产生数据分布偏移，否则没有产生数据分布偏移，又称为数据分布一致。

这里 $P(\cdot)$ 在连续情况下表示概率密度函数，在离散情况下表示概率质量函数，以下统称数据分布。数据分布偏移常出现于时间序列领域，比如在插值任务中，已知一段时间的数据点值，从而推断和填补这段时间内的未知数据点值，因此通常会假设该时间序列是平稳的，也即假设其数据分布的统计特性在该段时间内不变。还比如在预测任务中，已知一段时间的数据点值，然后预测未来某时间点或一段时间的未知数据点值。由于未来的数据点值无法在预测时间点前观测，因而其数据点值存在超出观测数据分布的可能性，但这种超出观测数据分布的可能性是无法提前估计的。因此为了简化问题，一些预测任务仍然会做平稳性假设，也即假设观测数据和待预测的未知数据保持数据分布一致，从而使得在观测数据上训练的模型可以泛化到待预测的未知数据上。因此实际上，数据分布偏移的可能性使得预测任务通常要比插值任务更难处理，且该问题在复杂系统中尤为常见，而平稳性假设就是数据分布偏移（一致）在时间序列研究领域中的一个具体含义。

定义 1 还可以扩展到场景数据生成问题中。如果生成方法约束 $P(x) = P(y)$ ，那么该方法是数据分布一致的，其生成的场景数据不会超出观测数据分布。比如在系统辨识方法中，通常采用最小二乘法或最

大似然估计等方法，使得辨识模型采样结果逼近观测数据点^[38]。比如，在生成模型研究^[39-42]中，均采用最大似然估计等方法求解 $P(x)$ ，实际等价于约束 $P(x) = P(y)$ 。比如，在潜在结果模型中，通常会做出一致性假设^[43,44]，也即求解的反事实结果不会超出观测数据分布。

相对的，如果生成方法是数据分布偏移的，那么其生成的场景数据可能会超出观测数据，若如此，则称其为外推的，显然外推的数据一定会导致数据分布偏移。比如，在结构因果模型^[34,35]中，通常不会假设求解的反事实结果在观测数据分布内，但同时，求解的反事实结果也无法通过观测数据验证其准确性，这也是本文关注的重点问题之一。类似概念还出现在 OOD (out-of-distribution) 问题^[45]中，该问题明确定义了 OOD 样本是不在训练数据集中的样本。此外，在领域泛化问题^[46,47]中，模型会在一个或多个源数据域上训练，然后试图泛化到新的未知数据域。这里数据域可以等价于数据集，且新的数据域中可能会有训练数据集中未出现过的样本。同类研究还有迁移学习^[48]，模型会在源数据域上进行预训练，并试图在目标数据域上微调从而实现泛化，目标数据域和源数据域的数据分布通常不同。其和领域泛化的主要区别在于，迁移学习过程中，模型可能会在训练过程中访问目标数据域，而领域方法则禁止访问。其他还有很多类似的研究问题，本文不在此一一列举，只是强调此类问题无论如何定义，其本质仍然是在解决数据分布偏移问题，只是其为了解决问题，可能做了不同的假设。值得注意的是，这些假设往往针对特定的应用场景，并不一定通用。

基于此，本文定义场景数据生成问题为已知场景数据 x ，利用外部知识 g' ，构建生成器 g ，实现生成场景数据 y ，如图 1.2 所示。在基于系统辨识的场景数据生成问题中，本文约束 $P(x) = P(y)$ ，且允许利用外部知识帮助设计系统模型，从而实现模型参数辨识。在基于生成模型的场景数据生成问题研究中，本文仍然约束 $P(x) = P(y)$ ，但不利用外部知识。本文将该问题归约为一个双样本问题^[49,50]，也即检测两组样本数据是否服从相同的数据分布，从而利用最大似然估计方法训练模型参数。在基于因果推断的场景数据生成问题中，本文不约束 $P(x) = P(y)$ ，且论述如何通过结构因果模型在无外部知识的情况下实现数据外推。因此，沿着系统辨识、生成模型、因果推断这条主线，其对外部知识的需求越来越少，且逐渐面向数据分布偏移问题的解决，这是更加接近于图 1.2 中所描述的人工智能对真实世界的认知过程。

如前所述，人工智能为真实世界建立系统模型并辨识它，其最终目的往往是为了利用这个模型理解并预测真实世界的未来变化。但是，真实世界的许多现象均具备复杂系统的特征^[51-54]，这会使得辨识系统模型的问题变得不适用，因而使其无法通过有限样本辨识出一个存在且唯一的形式化解。辨识一个复杂系统模型往往是困难的，但如果仅仅是学习一个预测的规则，则问题会变得简单许多^[55,56]。本文接下来讨论复杂系统中场景数据生成的瓶颈和难点。

目前，复杂系统并没有一个统一的定义，其复杂主要是相对于简单而言，一个简单的系统往往可以通过单一规则、线性关系或少数几个变量进行描述，而复杂系统则不能。比较有代表性的定义是：

定义 2 (复杂系统^[51])。 如果一个系统满足以下条件之一：

1. 是动态非线性系统，
2. 是混沌系统，
3. 是包含多个行为主体且具有层次结构的系统，
4. 是具有自适应能力的演化系统，

则称该系统为复杂系统。

因此，本文通过一阶微分方程建模非线性系统动力学。已知状态变量 $X_1, \dots, X_N \in \mathbb{R}$ ，系统建模如

下:

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1(t), x_2(t), \dots, x_N(t)), \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t), \dots, x_N(t)), \\ &\vdots \quad \vdots \\ \dot{x}_N(t) &= f_N(x_1(t), x_2(t), \dots, x_N(t)).\end{aligned}\tag{1.1}$$

这里对于 $i = 1, \dots, N$, $x_i(t)$ 是状态变量 X_i 在时刻 t 的状态值, 且 $\dot{x}_i(t) = dx_i(t)/dt$ 是 $x_i(t)$ 关于时间 t 的导数。为简化表示, 公式 (1.1) 也可以写为

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)),\tag{1.2}$$

其中 $\mathbf{x}(t) = [x_1(t), \dots, x_N(t)]$, $\dot{\mathbf{x}}$ 是其导数。

如果公式 (1.2) 同时又是混沌系统^[21], 则该系统对初始条件具有敏感依赖性, 也即 $\forall \mathbf{x}(0), \exists \delta > 0$, 使得 $\lambda = \lim_{t \rightarrow \infty} \lim_{\delta \mathbf{x}(0) \rightarrow 0} \frac{1}{t} \ln \left| \frac{\delta \mathbf{x}(t)}{\delta \mathbf{x}(0)} \right| > 0$, 这里 λ 是 Lyapunov 指数。如果一个系统是混沌系统, 那么辨识 \mathbf{F} 并做长期预测就变得不适当, 但短期预测可能是适当的。本文这里给出适当性问题的定义, 如下

定义 3 (适当性问题^[22,57]). 一个问题被称为适当的, 如果它满足以下三个条件:

1. 解的存在性: 对于给定的输入, 问题应该至少有一个解;
2. 解的唯一性: 解应该是唯一的;
3. 解对初始条件的连续依赖性: 解对输入数据的微小扰动是连续的, 即小的输入变化不会引起解的剧烈变化。

否则就称该问题是不适当的。这里给出本文能解决的问题的边界, 本文尝试解决一些适当的或者相对更适当的问题。

因此, 本文引出遍历性 (ergodicity) 假设。基于遍历性假设, 可以得到 Birkhoff 遍历定理。在此之前, 本文首先定义 σ 代数, 如下:

定义 4 (σ 代数). 设由 \mathbf{X} 的一些子集构成集合 \mathcal{B} , 则称 \mathcal{B} 是 σ 代数, 如果它满足:

1. $\mathbf{X} \in \mathcal{B}$,
2. $B \in \mathcal{B}$ 蕴含 $\mathbf{X} \setminus B \in \mathcal{B}$,
3. $B_n \in \mathcal{B}, n \in \mathbb{N}$ 蕴含 $\bigcup_{n=1}^{\infty} B_n \in \mathcal{B}$ 。

基于 σ 代数, 本文定义概率测度, 如下:

定义 5 (概率测度). $P : \mathcal{B} \rightarrow [0, 1]$ 称为 $(\mathbf{X}, \mathcal{B})$ 上的概率测度, 如果它满足:

1. $P(\emptyset) = 0$,
2. $P(\mathbf{X}) = 1$,
3. 若 $\{B_1, \dots, B_\infty\} \subset \mathcal{B}$ 为互不相交的子集序列, 则 $P\left(\bigcup_{n=1}^{\infty} B_n\right) = \sum_{n=1}^{\infty} P(B_n)$ 。

基于此, 本文称 $(\mathbf{X}, \mathcal{B}, P)$ 为概率测度空间。进而定义保测映射, 如下:

定义 6 (保测映射). 概率测度空间 $(\mathbf{X}, \mathcal{B}, P)$ 上的映射 $\mu : \mathbf{X} \rightarrow \mathbf{X}$ 为保测的, 如果 μ 满足 $\mu^{-1}(\mathcal{B}) \subset \mathcal{B}$, 且对 $\forall B \in \mathcal{B}, P(\mu^{-1}B) = P(B)$ 。

于是引出遍历性假设, 如下:

假设 1 (遍历性). 概率测度空间 $(\mathbf{X}, \mathcal{B}, P)$ 上的保测映射 $\mu : \mathbf{X} \rightarrow \mathbf{X}$ 是遍历的, 如果对 $B \in \mathcal{B}, \mu^{-1}(B) = B$, 蕴含 $P(B) = 0$ 或 $P(B) = 1$ 。

于是对于公式 (1.1), 可得 Birkhoff 遍历定理, 如下:

定理 1.2.1 (Birkhoff 遍历定理^[58]). 设概率测度空间 $(\mathbf{X}, \mathcal{B}, P)$ 上的保测映射 $\mu : \mathbf{X} \rightarrow \mathbf{X}$ 是遍历的, 则对 $i = 1, \dots, N, f_i \in L^1(P) = \{f : \mathbf{X} \rightarrow \mathbb{R} | \int |f| dP < \infty\}$, 有

$$\lim_{J \rightarrow \infty} \frac{1}{J} \sum_{j=0}^{J-1} f_i(\mu^j \mathbf{x}) = \int f_i dP, \text{ a.e. } \mathbf{x} \in \mathbf{X}. \quad (1.3)$$

因此, 满足遍历性假设的系统称为遍历系统, 而该定理实际等价于, 对于公式 (1.2), 给定系统的任意初值 $\mathbf{x}(0)$, 当时间 $t \rightarrow \infty$, 则 $\mathbf{x}(t)$ 会访问系统状态空间 $\mathbf{X} = [X_1, \dots, X_N] \in \mathbb{R}^N$ 中的所有区域。该定理意味着, 当时间 $t \rightarrow \infty$, 一个遍历系统的观测时序轨迹的平均值等价于其状态空间平均值。遍历性使得可以通过一条系统的时序轨迹辨识出系统整体的性质。

1.3 研究内容与创新点

本文针对复杂系统场景数据生成问题展开研究, 研究基于系统辨识、生成模型和因果推断等三类复杂系统场景数据生成方法。本文从数据分布偏移(一致)这一问题, 沿着系统辨识、生成模型和因果推断这一主线展开论述, 形成一整套复杂系统场景数据生成理论框架和技术方法论, 并最终论证了构造生成器 g 并实现 $P(x) \neq P(y)$ 的可行性。不过仍必须声明的是, 由于系统辨识、生成模型和因果推断这三个领域各自的知识量恢弘庞大, 本文除了在此三者的具体研究中做出了相应的前沿研究成果, 还要在其上整合三者并为复杂系统场景数据生成理论与技术研究服务。因此, 本文研究价值重在整合, 并尝试引出一些新的值得探索的科学问题。

本文在基于系统辨识、生成模型和因果推断的场景数据生成研究方面均进行了具体的研究, 主要贡献如下:

1. 本文提出了 ASIND 算法, 用于辨识复杂网络动力学系统, 见本文章节 2.3。该算法在文献^[36,37]的研究基础上更进一步, 不依靠对网络动力学的任何先验假设, 在自驱动函数、交互函数和交互网络均未知的情况下, 仍然实现了网络动力学系统模型的准确辨识。
2. 本文提出了基于跨时空模态生成对抗网络的场景数据生成框架, 用于实现新能源电网中的场景数据生成, 见本文章节 3.3。受自然环境影响, 新能源电网中的不确定性非常高, 尤其是风能和太阳能, 这部分极高的不确定性会传导至传统电网中, 从而使得整个电网变得脆弱^[59,60]。而生成的场景数据可以用于估计新能源电网中的不确定性, 为新能源电网的安全控制、优化与经济调度提供决策边界。该研究将跨时空模态生成对抗网络模型及其算法在风能和太阳能数据集上进行了测试, 实验证明了该模型和算法的性能达到了 SOTA (state-of-the-art) 水平。该研究还设计了一个随机日前经济调度技术应用场景, 验证了该生成对抗网络在具体的电网安全应用场景中的有效性。
3. 本文提出了基于跨时空模态生成对抗网络的完全数据缺失插补技术, 用于解决新能源电网中部分节点受通讯故障^[61,62]或外部的网络攻击^[63,64]而产生的完全数据缺失问题, 见本文章节 3.4。该研究利

用基于 Node2vec 算法的图矢量化嵌入技术、时序标签嵌入技术和光伏电站容量标签嵌入技术，将对应模态的场景数据均嵌入成矢量表示，从而实现时空模态融合表示，进而输入跨时空模态生成对抗网络，最终实现从图结构数据到时序数据的跨模态场景数据生成和缺失值插补。该研究在光伏数据集上进行了数值仿真测试，实验结果验证了该跨时空模态生成对抗网络能够解决完全数据缺失问题，且性能达到 SOTA。

4. 本文提出了时空贝叶斯网络模型，用于建模复杂系统中的时空因果，见本文章节 4.3.2。时空因果建模为复杂系统根因分析提供了基础模型，使得针对复杂系统的根因分析成为可能。该研究证明了时空贝叶斯网络具有时空因果等价性、可分解性和唯一性等性质，见本文章节 4.3.3 和本文章节 4.3.4。其中时空因果唯一性表明时空贝叶斯网络中不存在等价的网络结构（独立性相同但网络结构不同），其使得基于时空贝叶斯网络的反事实推理具有唯一的结果，这相对于传统贝叶斯网络是一个十分重要的优势。
5. 本文提出了前向反事实推理算法，用于递归地实现多节点干预后分布校正，该算法具有时间复杂度 $\mathcal{O}(|\mathbf{V}|^2 + |\mathbf{V}||\mathbf{E}|)$ ，这里 $|\mathbf{V}|$ 是有向无环图的节点数， $|\mathbf{E}|$ 是有向无环图的连边数，从而使得基于结构因果模型的反事实推理能够在多项式时间内完成，研究内容见本文章节 4.2.3 中算法 3。值得一提的是，该研究启发于文献^[65] 的一份审稿意见，其中审稿人认为无法在文献^[65] 涉及的时空贝叶斯网络上进行反事实推理，于是本文作者在文献^[66] 中回应，只要给出的时空贝叶斯网络是有向无环图（实际上必然是有向无环的），那么就可以通过前向反事实推理算法在多项式时间内完成推理。然而，当前向反事实推理算法提出后，本文作者却又在文献^[67] 中发现了类似的递归算法，尽管二者都是基于递归思想，但前向反事实推理算法对递归过程的描述可能更加清楚，因此二者可以互为补充。
6. 本文提出了基于高阶因果熵的唯一因果网络发现算法，能够在时间复杂度 $\mathcal{O}(N^3 \tau_{\max})$ 下唯一辨识时空贝叶斯网络，从而进行复杂系统时空因果推断，这里 N 是系统中变量数， τ_{\max} 是最大因果时延，见本文章节 4.4。该研究还将高阶因果熵算法在仿真数据集上进行了测试，实验结果表明该算法的性能达到 SOTA。本文还将高阶因果熵算法扩展到高维场景，也即系统变量的属性特征是高维的，针对这种特殊情况，本文利用人工神经网络进行特征提取，从而解决了高维场景下的时空因果发现。

本文整合了以上研究内容，并形成一整套复杂系统场景数据生成理论框架和技术方法论。本文首先讨论了基于系统辨识的场景数据生成方法，见本文章节 2.1.1。本文将稀疏辨识和人工神经网络均视为数据驱动系统辨识的具体模型和算法，从而揭示其二者在系统辨识准确性和可解释性上的平衡关系，也即稀疏辨识可解释性高，但准确性相对低，而人工神经网络准确性高，但可解释性低。从这一思路出发，本文认为物理信息神经网络在二者间折中，在准确性和可解释性上均有优势。进而本文在章节 2.2，由于人工神经网络具有极强的函数近似能力，能够近似任意的复杂连续函数，于是本文提出“网络即系统”这一研究思路，也即认为任何复杂系统均可以由一个人工神经网络进行近似表示，从而引出针对复杂系统的网络表示研究，进而延伸到对复杂网络的研究，从而揭示网络表示的复杂系统所应具备的协同性、可控性、弹性和低秩性。为了在无任何先验知识的情况下实现任意复杂网络动力学的系统辨识，本文在章节 2.3 中提出 ASIND 算法，其能够从观测的时序数据中准确辨识出一个复杂网络系统模型，为网络表示的复杂系统研究提供建模基础。

本文进而转向对基于生成模型的场景数据生成方法的讨论，见本文第三章。本文揭示了五大类基础的生成模型（变分自编码器、生成对抗网络、生成流模型、生成扩散模型和得分生成模型）在双样本问题上的研究共性，本文认为所有的生成模型均需要约束观测场景数据和生成场景数据分布一致，见本文章节 3.1。此外，所有的生成模型均具备 “ $f - g$ ” 构造，这里 $f : x \rightarrow z$ 实现从观测数据 x 到特征 z 的特征提取， $g : z \rightarrow x$ 实现从特征 z 到场景数据的生成。尽管 f 和 g 在不同的生成模型中有不同的称呼，但这样的 “ $f - g$ ” 结构总是具备。基于基础的生成模型，本文讨论延伸至多模态场景数据生成，见本文章节 3.2。本文在生成模型的基础上扩展出条件场景数据生成，采用的思路同样是双样本问题和 “ $f - g$ ” 结构，只是

增加了条件标签 c , 见本文章节 3.2.1。本文进而认为, 所有的多模态场景数据生成方法只有两种形式, 其一是将基础生成模型中的观测场景数据 x 直接视作多模态场景数据, 其二是将条件场景数据生成模型中的观测场景数据 x 和条件标签 c 视作不同模态, 见本文章节 3.2.2。本文由此展开对多模态特征提取的方法的讨论, 其主要分为专用多模态特征提取和通用多模态特征提取, 前者更多考虑模态的结构特征而设计隐式的特征提取方法, 而后者并不特别考虑模态的结构特征而设计显式的特征提取方法, 分别见本文章节 3.2.3 和本文章节 3.2.4。本文进而讨论基于生成式大模型的多模态场景数据生成, 集中讨论了构造生成式大模型的几项重要技术, 包括预训练和微调技术、提示技术和混合专家模型等, 见本文章节 3.2.5。当然, 本文并没有完整描述生成式大模型的实现方法, 其主要原因是因为 (1) 实现生成式大模型无法避免地需要讨论特定的领域知识, 而这些与本文研究内容无关, (2) 生成式大模型的实现难度往往不在技术层面, 也即其需要考虑的更多是, 如何让生成式大模型的商用收益高于模型训练的成本和数据获取的成本, 只有如此, 才能使得一个商用的生成式大模型随着用户数据的不断输入, 而不断地迭代更新模型参数, 从而获得更好的性能, 以获得更大的收益, 这是本文的一个重要观点。此外, 本文于章节 3.3 和章节 3.4 提供了两个基于跨时空模态生成对抗网络的场景数据生成数值案例用于佐证本文观点, 也即基于生成模型的多模态场景数据生成是可以实现的。

本文进而转向对基于因果推断的场景数据生成方法的讨论, 见本文第四章。这部分研究起源于对突破数据分布一致约束 ($P(x) = P(y)$, 见定义 1) 的尝试, 如前所述, 无论是系统辨识或是生成模型, 其总是要求场景数据生成过程满足数据分布一致, 只有这样才能保证生成的场景数据不会越出对该系统的所有观测。但在真实世界中, 人脑却经常能够对未知的事物做出种种推测, 并且这些推测的结论虽起源于人类经验, 却往往与人类经验大相径庭。为了探索这种对未知事物的推测机制, 本文作者找到了与之理念相契合的研究领域, 也即反事实推理, 反事实推理是因果推断研究领域中十分重要研究内容, 同时也是其研究目的之一。因此, 本文先讨论了反事实推理的两大重要研究框架, 也即潜在结果模型和结构因果模型, 并且论述了其二者的使用方法及其基础假设, 见本文章节 4.1。本文进而讨论了潜在结果模型和结构因果模型的根本分歧, 见本文章节 4.2, 其二者的一个重要分歧在于, 潜在结果模型做出了一致性假设, 从而约束其反事实推理的结果不会越出观测数据, 而结构因果模型则不做此假设, 因而其更注重思考和推理的过程, 而不关注反事实推理的结果与观测数据是否一致, 从而为突破数据分布一致约束提供了可能性。本文沿着这个思路, 将结构因果模型的研究延伸至复杂系统领域, 并进行了复杂系统时空因果推断的相关研究, 见本文章节 4.3。本文提出了时空贝叶斯网络模型, 用于建模复杂系统中的时空因果, 并证明了时空贝叶斯网络具有时空因果等价性、可分解性和唯一性等重要性质。本文进而论述了利用前向反事实推理算法 (见本文算法 3) 在时空贝叶斯网络上进行反事实推理的方法。本文还论述了如何利用高阶因果熵算法进行唯一因果网络发现, 从而使得其反事实推理结果唯一, 见本文章节 4.4。

综上所述, 本文沿着系统辨识、生成模型和因果推断这条研究主线, 最终形成了一整套复杂系统场景数据生成理论框架和技术方法论。本文重点探讨了如何进行复杂系统场景数据生成才能突破数据分布一致的难题。

本文研究创新性集中在研究视角创新、理论创新和技术创新等方面。其中最重要的就是研究视角创新, 本文提出了复杂系统场景数据生成的基本框架, 并通过数据分布偏移 (一致) 这一问题串联了系统辨识、生成模型和因果推断等三个重要研究领域, 并在其上整合出复杂系统场景数据生成的理论与技术。本文为后续研究提供了一个新的研究视角, 使得后续研究可以继续沿着这条主线展开, 最终赋予人工智能更接近人类水平的认知能力。

为了实现这一目标, 本文展开了针对复杂系统时空因果推断的研究, 并提出了时空贝叶斯网络模型, 用于建模复杂系统中的时空因果。本文证明了时空贝叶斯网络具有时空因果等价性、可分解性和唯一性等重要性质, 为复杂系统时空因果推断提供了理论基础。本文集中论述了通过基于时空贝叶斯网络的结构因果模型实现数据外推的可行性, 见本文章节 4.2 和本文章节 4.3, 该论述是本文最重要的理论创新之一, 其明确地给出了人工智能实现数据外推所应做出的假设条件。

此外，本文在具体的技术研究上也作出了相应的技术创新，具体体现在诸多算法的提出，包括 ASIND 算法（见本文章节 2.3）、跨时空模态场景数据生成算法（见本文算法 1）、跨时空模态完全数据缺失插补算法（见本文算法 2）、前向反事实推理算法（见本文算法 3）和高阶因果熵算法（见本文算法 4），这些算法共同使得复杂系统场景数据生成得以实现。

1.4 本文组织结构

本文组织结构如图 1.3 所示，该图中标记了本文所有章节及其对应的编号。可以清楚地看到，本文起始于绪论，落脚于总结与展望，中间基于系统辨识、生成模型和因果推断的场景数据生成分别对应第二、三和四章，且彼此相对独立，但又在场景数据生成问题上保持联系。本文围绕这三种场景数据生成方法展开讨论，并逐步延伸至具体的研究内容，最终汇总至第五章总结与展望。

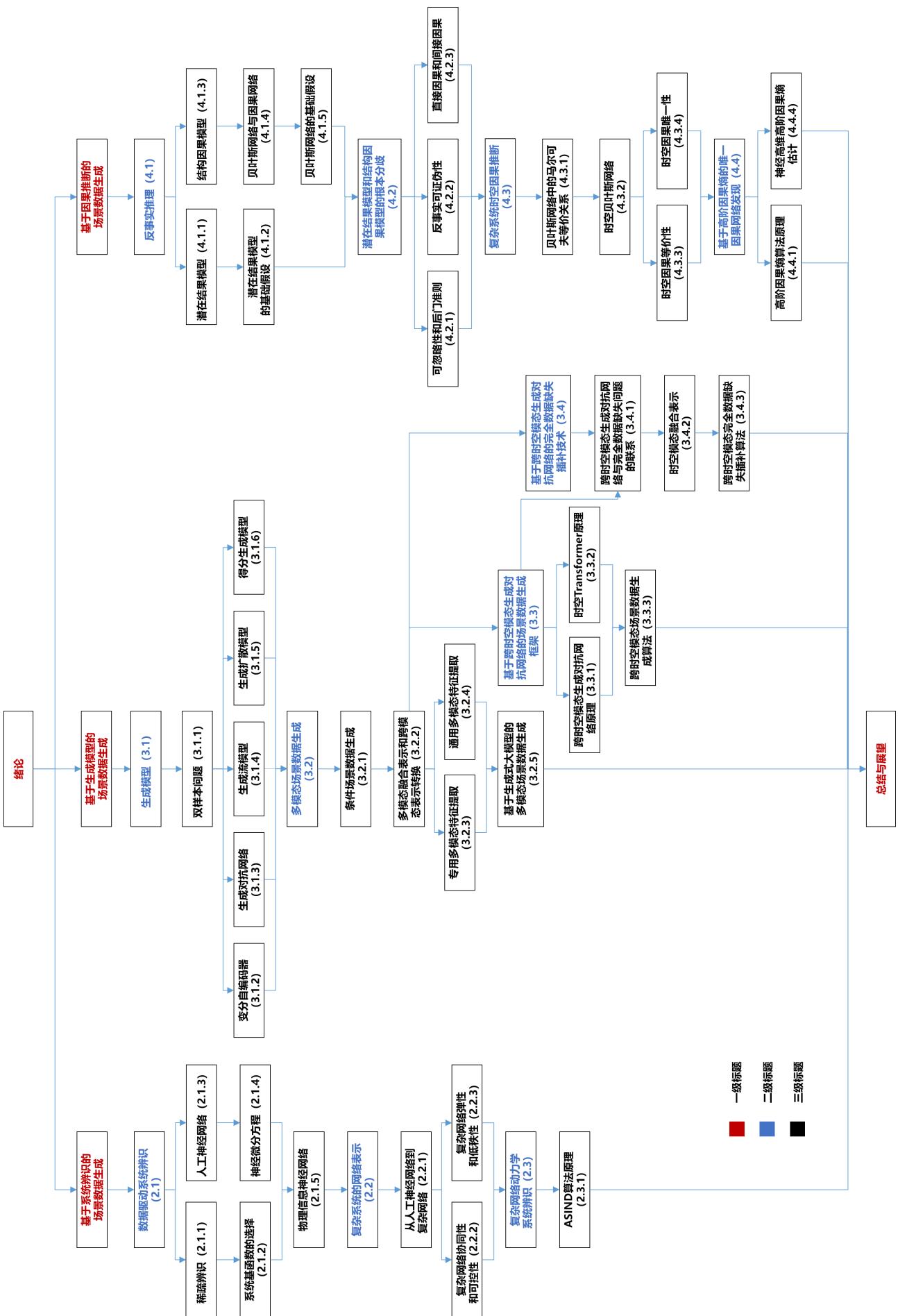


图 1.3: 本文组织结构图。

第二章 基于系统辨识的场景数据生成

基于大数据技术，数据驱动的系统辨识成为可能，其主要的关注点在于如何利用大量数据辨识系统模型（见公式(1.2)）。当然，系统辨识技术本就建立在适应观测数据的基础上，而由于有大量的观测数据可供使用，从而使得数据驱动的系统辨识技术可以对系统模型采用更加宽松的假设。尤其是人工神经网络的使用，使得建模过程变得更加“粗糙”，建模时无需过多考虑专业的领域知识，反而还能更好的拟合观测数据，但同时模型的可解释性也逐渐下降。本文后续将按照这一思路进行论述。

2.1 数据驱动系统辨识

2.1.1 稀疏辨识

稀疏辨识最早来自于压缩感知技术^[68-70]的应用。压缩感知的核心概念是：大部分信号都可以通过一些合适的基进行稀疏重构，这使得可以以低于传统奈奎斯特采样定理要求的采样率进行信号采样，降低了采样成本。稀疏辨识把压缩感知的思路沿用到了对系统函数的辨识上，进而假设系统函数可以由一些简单的基函数重构表示，这些可供选择的基函数可能有无穷多个，但实际被辨识出的只是很少的一部分，因而称为稀疏辨识。

为介绍稀疏辨识方法，本文首先引出最小二乘法作为该方法的基础。已知系统模型，也即公式(1.2)，且已知该系统的观测时序轨迹数据 $\mathbf{x} \in \mathbb{R}^{T \times N}$ ，如下

$$\mathbf{x} = \begin{bmatrix} x_1(t_0) & x_2(t_0) & \dots & x_N(t_0) \\ x_1(t_1) & x_2(t_1) & \dots & x_N(t_1) \\ \vdots & \vdots & \dots & \vdots \\ x_1(t_{T-1}) & x_2(t_{T-1}) & \dots & x_N(t_{T-1}) \end{bmatrix} \text{time } \downarrow \quad (2.1)$$

于是可以通过一阶差分法获得 $\dot{\mathbf{x}} = [\dot{\mathbf{x}}_1, \dots, \dot{\mathbf{x}}_N] \in \mathbb{R}^{(T-1) \times N}$ ，也即 $\dot{x}_i(t_j) = \frac{x_i(t_{j+1}-t_j)-x_i(t_j)}{t_{j+1}-t_j}, j = 0, \dots, T-2$ 。注意，这里一阶差分法会导致一个样本的损失，实际执行时可以去掉 \mathbf{x} 中相应的样本来对齐。于是参数化模型（见公式(1.1)）得 $f_i(\mathbf{x}, \boldsymbol{\theta}) \in \mathbb{R}^{T-1}, i = 1, \dots, N$ ，表示 $f_i(\cdot, \boldsymbol{\theta})$ 作用在 \mathbf{x} 上。则可以采用最小二乘法获得参数，如下：

$$\min_{\boldsymbol{\theta}} \|\dot{\mathbf{x}}_i - f_i(\mathbf{x}, \boldsymbol{\theta})\|_2^2. \quad (2.2)$$

如果 f_i 是线性的，也即 $f_i(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{x}\boldsymbol{\theta}$ ，其中 $\boldsymbol{\theta} \in \mathbb{R}^N$ ，则通过求驻点可得公式(2.2)的解为 $\boldsymbol{\theta} = (\mathbf{x}^\top \mathbf{x})^{-1} \mathbf{x}^\top \dot{\mathbf{x}}_i$ 。

但如果 f_i 是非线性的，则可以将其表示为一组基函数的线性加权，如下

$$f_i(\mathbf{x}, \boldsymbol{\theta}) = \sum_{m=1}^M g_m(\mathbf{x})\theta_m = \mathbf{g}(\mathbf{x})\boldsymbol{\theta}, \quad (2.3)$$

其中 g_m 为基函数， $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{(T-1) \times M}$ ，表示字典矩阵 $[g_1(\cdot), \dots, g_M(\cdot)]$ 作用在 \mathbf{x} 上。注意这里基函数是固定的，没有参数，参数是需要辨识的权重 $\boldsymbol{\theta} \in \mathbb{R}^M$ 。为了使得拟合效果更好，理论上可以通过增加 g_m 以增强对 f_i 的表示能力，当 $M = T - 1$ ，称字典矩阵为完备的，当 $M > T - 1$ ，称字典矩阵为过完备的，

当 $M < T - 1$ 时，称字典矩阵为欠完备的。但增加 g_m 的同时也会增加计算开销，而且增加 g_m 还会导致过拟合，因为可能有一些不应当被辨识出来的基函数被赋予了非零且微小的权重，由于非线性系统常有的初始条件敏感依赖性，这些微小的影响可能会使系统的长期预测变得困难。所以，为了在二者中折中，则在公式 (2.2) 中增加 L_0 正则项，以尽可能减少非零参数的个数。整理上述，得到稀疏辨识问题，如下：

$$\begin{aligned} \min_{\theta} \|\theta\|_0 \\ \text{s.t. } \dot{\mathbf{x}}_i = \mathbf{g}(\mathbf{x})\theta. \end{aligned} \quad (2.4)$$

其中 $\|\theta\|_0$ 为 L_0 范数，其被定义为参数矢量 θ 中非零元素的个数。本文通过网络结构可视化了稀疏辨识问题，如图 2.1 所示，稀疏辨识问题分为两部分，其一是从输入到字典矩阵的构建过程，这里网络节点间可以是全连接的，参数是固定的。其二是从字典矩阵到输出过程，也即求解公式 (2.4)。直观地看，稀疏辨识同样具备人工神经网络的特征，本文后续还会将其和人工神经网络的表示联系起来。

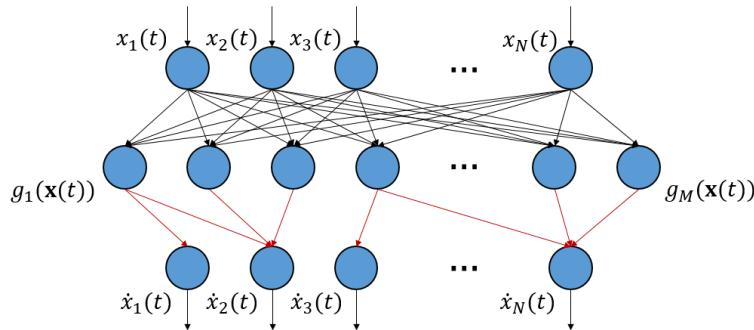


图 2.1: 稀疏辨识示意图。红色连边表示辨识的参数 θ 。

公式 (2.4) 的问题是 NP-hard 问题，也即没有算法能够在多项式时间求解该问题并验证解的准确性。因而求解该问题主要有两种思路：(1) 通过一些启发式的近似算法逼近该问题的解，但其解无法验证准确性。(2) 通过一些凸松弛方法将公式 (2.4) 转换为其他等价的且可以精确求解的问题，但需要做出更多假设，如果不满足假设条件，则求出的解不一定是公式 (2.4) 的解。以下主要针对这两种思路展开论述。

首先是近似算法，近似算法主要采用贪婪策略，依据规则选择最合适的一组基函数，最终使得重构误差低于阈值。基于此，S.G. Mallat 等人^[7] 提出了匹配追踪 (Matching Pursuit) 算法，其解决思路可以分为五个步骤：

1. 标准化 $\mathbf{g}(\mathbf{x})$ 为 $[\tilde{\mathbf{g}}_1, \dots, \tilde{\mathbf{g}}_M]$ ，初始化迭代步数 $j = 0$ ，令 $\mathbf{r}^{(j)} = \dot{\mathbf{x}}_i$ 。

2. 找到与残差矢量最相似的未被选择过的基函数：

$$m = \arg \max_m \tilde{\mathbf{g}}_m^\top \mathbf{r}^{(j)}. \quad (2.5)$$

3. 根据选择的基函数求解对应权重：

$$\theta_m = \tilde{\mathbf{g}}_m^\top \mathbf{r}^{(j)}. \quad (2.6)$$

4. 更新残差矢量：

$$\mathbf{r}^{(j+1)} = \mathbf{r}^{(j)} - \tilde{\mathbf{g}}_m \theta_m. \quad (2.7)$$

5. 通过阈值法判断 $\|\mathbf{r}^{(j+1)}\|_2$ 是否小于阈值。如果小于则算法收敛并返回权重，未被选择的基函数权重为 0。否则，回到步骤 2。

匹配追踪算法将 $\dot{\mathbf{x}}_i$ 分解为已被匹配的部分和残差，每次选择与残差方向最接近的基函数矢量，通过投影法求解其对应权重，如图 2.2 所示，然后更新残差，重复过程直到收敛。基于此类正交思想，Y.C.

Pati 等人^[72]提出了正交匹配追踪 (Orthogonal Matching Pursuit) 算法, 其在统计学中又称为前向逐步回归 (Forward Stepwise Regression) 算法^[73], 以及在文献^[74]中称为最小二乘正交匹配追踪算法, 但实际上是同一种算法。其主要改进是在匹配追踪算法步骤 3 中, 利用全部已经选择的基函数矢量通过最小二乘法求解所有矢量的权重, 也即 $\boldsymbol{\theta}|_{\mathbf{M}} = \arg \min_{\boldsymbol{\theta}|_{\mathbf{M}}} \|\dot{\mathbf{x}} - \mathbf{g}(\mathbf{x})|_{\mathbf{M}}^{\top} \boldsymbol{\theta}|_{\mathbf{M}}\|_2^2$ 。这里, \mathbf{M} 是已选择的基函数的索引, $\mathbf{g}(\mathbf{x})|_{\mathbf{M}}$ 和 $\boldsymbol{\theta}|_{\mathbf{M}}$ 分别表示被选择的基函数矢量和对应的参数。因此, 实际上正交匹配追踪算法和基本的匹配追踪算法最主要的区别在于求解参数的不同, 后者是一次更新一个参数, 前者是每次更新所有参数。这同时会使得其残差的计算结果不同, 而且正交匹配追踪算法的计算开销更高。此外还值得注意的是, 基本的匹配追踪算法中使用了标准化, 对求解的参数需要逆标准化到对应尺度。

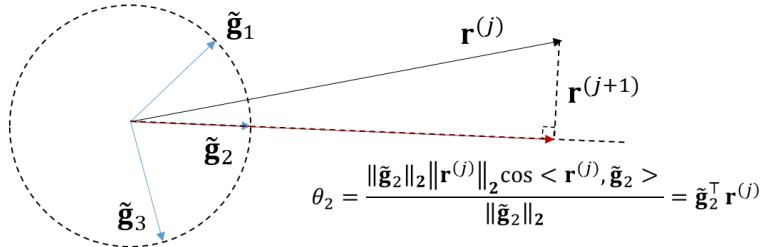


图 2.2: 匹配追踪算法中公式 (2.5) 和公式 (2.6) 示意图

除了近似算法, 还可以通过一些凸松弛方法将公式 (2.4) 转换为其他更易求解的问题, 比如转换成 L_1 正则化问题 (“Lasso 问题”^[75]), 如下:

$$\begin{aligned} & \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 \\ \text{s.t. } & \dot{\mathbf{x}}_i = \mathbf{g}(\mathbf{x})\boldsymbol{\theta}. \end{aligned} \quad (2.8)$$

于是有如下定理:

定理 2.1.1 (正交贪婪算法等价性^[74, 76]). 公式 (2.4) 和公式 (2.8) 等价, 如果满足:

$$\|\boldsymbol{\theta}\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{g}(\mathbf{x}))} \right), \quad (2.9)$$

这里 $\mu(\mathbf{g}(\mathbf{x})) = \max_{m_1 \neq m_2} \frac{|g_{m_1}^{\top}(\mathbf{x})g_{m_2}(\mathbf{x})|}{\|g_{m_1}(\mathbf{x})\|_2 \|g_{m_2}(\mathbf{x})\|_2}$ 。

由于 L_1 范数具有凸性, 因此如果公式 (2.8) 的最优解存在, 则该解唯一。当满足定理 2.1.1 中假设条件时, 公式 (2.8) 的解等价于公式 (2.4) 的解。具体证明过程见文献^[74] 中 Theorem 4.3。其主要思想是在矩阵 $\mathbf{g}(\mathbf{x})$ 中找到 $\|\boldsymbol{\theta}\|_0$ 个互相最不相关的列。因为在匹配追踪等贪婪算法中, 如图 2.2 所示, 当找到一个与残差矢量最相似的列矢量 (基函数矢量) 后, 计算出的新残差与该列矢量垂直, 也即不相关。因而在新一轮迭代中, 找到的下一个列矢量与该列矢量也应尽可能接近垂直, 所以由此找到的所有列矢量应当尽可能互不相关, 因而具有有限等距性质 (见文献^[76] 中 Definition 1.1)。

求解公式 (2.8) 的精确算法有很多。其一是将其转换为线性规划问题, 用求解线性规划的算法等价求解^[77]。简言之, 引入非负参数矢量 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^M$, 其中 $\mathbf{u} = \max(0, \boldsymbol{\theta})$, $\mathbf{v} = -\min(0, \boldsymbol{\theta})$, 因此 $\boldsymbol{\theta} = \mathbf{u} - \mathbf{v}$ 。构建新参数矢量 $\mathbf{w} = [\mathbf{u}^{\top}, \mathbf{v}^{\top}]^{\top} \in \mathbb{R}^{2M}$, 从而使得公式 (2.8) 中的 $\|\boldsymbol{\theta}\|_1 = \sum_{m=1}^{2M} w_m$, 这里 w_m 为 \mathbf{w} 的第 m 个参数。同时, $\mathbf{g}(\mathbf{x})\boldsymbol{\theta} = [\mathbf{g}(\mathbf{x}), -\mathbf{g}(\mathbf{x})]\mathbf{w}$ 。整理得线性规划问题:

$$\begin{aligned} & \min_{\mathbf{w}} \sum_{m=1}^{2M} w_m \\ \text{s.t. } & \dot{\mathbf{x}}_i = [\mathbf{g}(\mathbf{x}), -\mathbf{g}(\mathbf{x})]\mathbf{w}, \\ & \mathbf{w} \geq 0. \end{aligned} \quad (2.10)$$

当公式(2.10)求解完成，则可得最优解 \mathbf{u}^* 和 \mathbf{v}^* ，则 $\boldsymbol{\theta}^* = \mathbf{u}^* - \mathbf{v}^*$ 。值得注意的是，此类转换方法虽然能够准确地等价求解公式(2.8)，但是同时参数的维度也变为 $2M$ ，一方面计算开销变为两倍，另一方面在高维情况下会极大增加求解难度。

其二是使用次梯度方法，并通过梯度下降法迭代求解最优参数。简言之，首先构建等式 $\boldsymbol{\theta} = \mathbf{w}$ ，于是可以对公式(2.4)做变换如下：

$$\begin{aligned} & \min_{\boldsymbol{\theta}} \|\mathbf{w}\|_1 \\ \text{s.t. } & \dot{\mathbf{x}}_i = \mathbf{g}(\mathbf{x})\boldsymbol{\theta} \\ & \boldsymbol{\theta} = \mathbf{w}. \end{aligned} \quad (2.11)$$

于是通过增广拉格朗日(Augmented Lagrangian)方法将上述公式转换为单目标无约束优化问题：

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\dot{\mathbf{x}}_i - \mathbf{g}(\mathbf{x})\boldsymbol{\theta}\|_2^2 + \lambda \|\mathbf{w}\|_1 + \frac{1}{\rho} \mathbf{v}^\top (\boldsymbol{\theta} - \mathbf{w}) + \frac{1}{2\rho} \|\boldsymbol{\theta} - \mathbf{w}\|_2^2, \quad (2.12)$$

其中 $\mathbf{v} \in \mathbb{R}^M$ 是拉格朗日乘子， λ 和 ρ 是惩罚因子。于是可以将公式(2.12)拆成三个子问题交替求解^[78]：

$$\boldsymbol{\theta}^{(j+1)} = \arg \min_{\boldsymbol{\theta}} \frac{1}{2} \|\dot{\mathbf{x}}_i - \mathbf{g}(\mathbf{x})\boldsymbol{\theta}\|_2^2 + \frac{1}{\rho} \mathbf{v}^\top (\boldsymbol{\theta} - \mathbf{w}^{(j)}) + \frac{1}{2\rho} \|\boldsymbol{\theta} - \mathbf{w}^{(j)}\|_2^2 \quad (2.13)$$

$$\begin{aligned} &= \left[\mathbf{g}^\top(\mathbf{x})\mathbf{g}(\mathbf{x}) + \frac{1}{\rho} \mathbf{I} \right]^{-1} \left[\mathbf{g}^\top(\mathbf{x})\dot{\mathbf{x}}_i + \frac{1}{\rho} (\mathbf{v}^{(j)} - \mathbf{w}^{(j)}) \right] \\ \mathbf{w}^{(j+1)} &= \arg \min_{\mathbf{w}} \lambda \|\mathbf{w}\|_1 + \frac{1}{\rho} \mathbf{v}^\top (\boldsymbol{\theta}^{(j+1)} - \mathbf{w}) + \frac{1}{2\rho} \|\boldsymbol{\theta}^{(j+1)} - \mathbf{w}\|_2^2, \end{aligned} \quad (2.14)$$

$$\begin{aligned} &= \arg \min_{\mathbf{w}} \lambda \rho \|\mathbf{w}\|_1 + \frac{1}{2} \|\mathbf{v}^{(j)} + \boldsymbol{\theta}^{(j+1)} - \mathbf{w}\|_2^2, \\ &= S_{\lambda\rho}(\mathbf{v}^{(j)} + \boldsymbol{\theta}^{(j+1)}) \end{aligned} \quad (2.14)$$

$$\mathbf{v}^{(j+1)} = \mathbf{v}^{(j)} + \frac{1}{\rho} (\boldsymbol{\theta}^{(j+1)} - \mathbf{w}^{(j+1)}) \quad (2.15)$$

这里 $S_{\lambda\rho}(\mathbf{v}^{(j)} + \boldsymbol{\theta}^{(j+1)})$ 为软阈值算子。由于 $\|\mathbf{w}\|_1$ 在原点处不可微，因而这里求解公式(2.14)时对 $w_m, m = 1, \dots, M$ 使用次微分，也即：

$$\partial \|\mathbf{w}\|_1 = \begin{cases} 1, & \text{if } w_m > 0, \\ [-1, 1], & \text{if } w_m = 0, \\ -1, & \text{if } w_m < 0. \end{cases} \quad (2.16)$$

次微分是所有次梯度的集合，也即 $\partial \|\mathbf{w}\|_1 = \{\mathbf{u} \in \mathbb{R}^M | \|\mathbf{w}'\|_1 - \|\mathbf{w}\|_1 \geq \mathbf{u}^\top (\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}', \mathbf{w} \in \mathbb{R}^M\}$ 。从文献^[79]可知，当且仅当 $\mathbf{0} \in \partial \|\mathbf{w}^*\|_1$ 时， \mathbf{w}^* 为全局最小值。因此，通过求驻点可得软阈值算子，定义如下：

$$S_{\lambda\rho}(\mathbf{v}^{(j)} + \boldsymbol{\theta}^{(j+1)}) = \begin{cases} v_m^{(j)} + \theta_m^{(j+1)} - \lambda\rho, & \text{if } v_m^{(j)} + \theta_m^{(j+1)} \geq \lambda\rho, \\ 0, & \text{if } |v_m^{(j)} + \theta_m^{(j+1)}| < \lambda\rho, \\ v_m^{(j)} + \theta_m^{(j+1)} + \lambda\rho, & \text{if } v_m^{(j)} + \theta_m^{(j+1)} \leq -\lambda\rho. \end{cases} \quad (2.17)$$

公式(2.13)-(2.15)又称为交替方向乘子法(Alternating Direction Method of Multipliers, ADMM)^[78]。由于公式(2.12)是凸问题，因此该算法有解且唯一。

除了转换成 L_1 正则化问题，徐宗本等人^[80,81]还提出了 $L_{1/2}$ 正则化问题，如下：

$$\begin{aligned} & \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_{1/2}^{1/2} \\ \text{s.t. } & \dot{\mathbf{x}}_i = \mathbf{g}(\mathbf{x})\boldsymbol{\theta}. \end{aligned} \quad (2.18)$$

实验表明，该问题可以产生比 L_1 正则更加稀疏的解。但该问题是非凸的，解不一定唯一。

2.1.2 系统基函数的选择

除了辨识算法，稀疏辨识中的一个重要部分就是基函数的选择，基函数的选择主要有两类范式，一类是令所有的字典矩阵元素都基于某种特定的基函数生成，另一类是任意地根据假设生成字典矩阵元素，这通常会生成耦合项。所以回到公式(2.1)，根据这样的观测数据 \mathbf{x} 获得 $\dot{\mathbf{x}}$ ，参考公式(2.2)和公式(2.3)可以得到对 $\dot{\mathbf{x}}$ 的重构，也即 $\dot{\mathbf{x}}_i = \sum_{m=1}^M g_m(\mathbf{x})\theta_m = \mathbf{g}(\mathbf{x})\boldsymbol{\theta}, i = 1, \dots, N$ 。这里 $\mathbf{g}(\mathbf{x})$ 是字典矩阵。

其一，如果任意地生成字典矩阵，则可得 SINDy 算法^[82,83]，其字典矩阵结构如下：

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 1 & \text{Poly}^1(\mathbf{x}(t_0)) & \text{Poly}^2(\mathbf{x}(t_0)) & \dots & \sin(\mathbf{x}(t_0)) & \cos(\mathbf{x}(t_0)) & e^{\mathbf{x}(t_0)} \\ 1 & \text{Poly}^1(\mathbf{x}(t_1)) & \text{Poly}^2(\mathbf{x}(t_1)) & \dots & \sin(\mathbf{x}(t_1)) & \cos(\mathbf{x}(t_1)) & e^{\mathbf{x}(t_1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & \text{Poly}^1(\mathbf{x}(t_{T-2})) & \text{Poly}^2(\mathbf{x}(t_{T-2})) & \dots & \sin(\mathbf{x}(t_{T-2})) & \cos(\mathbf{x}(t_{T-2})) & e^{\mathbf{x}(t_{T-2})} \end{bmatrix}. \quad (2.19)$$

注意这里少一条数据，因为差分计算 $\dot{\mathbf{x}}$ 时会损失一条数据。其中， $\text{Poly}(\cdot)$ 是多项式基函数，比如：

$$\begin{aligned} \text{Poly}^1(\mathbf{x}(t)) &= [x_1(t) \ \dots \ x_N(t)] \\ \text{Poly}^2(\mathbf{x}(t)) &= [x_1^2(t) \ x_1(t)x_2(t) \ x_1(t)x_3(t) \ \dots \ x_2^2(t) \ \dots \ x_N^2(t)]. \end{aligned} \quad (2.20)$$

其他更高阶的多项式可以此类推。本文这里只是举例，实际上还可以构建任意复杂的基函数，当然如前所述，这样做会增加算法复杂度，从而降低算法计算效率和准确率。其他采用这种构建方法的算法还有基于贝叶斯模型的稀疏贝叶斯辨识算法^[84,85]，稀疏贝叶斯算法对参数 $\boldsymbol{\theta}$ 做了高斯分布的假设，从而降低其求解难度，但其本质是在算法层面减少这种“任意性”带来的影响。

其二，如果采用统一的某种特定基函数生成，则通常采用一些信号重构算法。与其一构建方法相比，这种字典矩阵构建方法的可解释性会低一些，但好处是通常不需要构建庞大的过完备字典矩阵就可以实现近似。本文列出一些常见的此类基函数，见表 2.1。首先是傅里叶基函数，其来源于傅里叶变换，如下：

$$\begin{cases} x(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt, \\ x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} x(\omega)e^{j\omega t} d\omega. \end{cases} \quad (2.21)$$

简言之，第一个公式将 $x(t)$ 从时域转换至频域 $x(\omega)$ ，这里 ω 是频率。因此第二个公式将 $x(\omega)$ 转换至 $x(t)$ ，又称为傅里叶逆变换。傅里叶基函数重构方法将不同的 $\frac{1}{2\pi}x(\omega)$ 当作未知参数 θ_m ，将 $e^{j\omega t}$ 用来设计基函数 $g_m(t)$ ，重构的目标信号设为 $\dot{x}_i(t)$ 即可。由此可得傅里叶基函数的离散表示形式，见表 2.1。

然后是小波基函数，其发展于傅里叶变换，简言之，就是用不同尺度的小波基函数组合重构目标信号。比如哈尔小波^[86]，定义如下：

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t < \frac{1}{2}, \\ -1, & \text{if } \frac{1}{2} \leq t < 1, \\ 0 & \text{if otherwise.} \end{cases} \quad (2.22)$$

从而可定义不同尺度的小波基函数，见表 2.1。从而可得重构公式如下：

$$\dot{x}_i(t) = \sum_{a,b} g_{(a,b)}(t)\theta_{(a,b)} = \sum_{a,b} \frac{1}{\sqrt{|a|}} \phi\left(\frac{t-b}{a}\right) \theta_{(a,b)}. \quad (2.23)$$

这里令 $m = (a, b)$ ，不同的 a 和 b 将小波基函数缩放到不同尺度，一个基函数组成字典矩阵中的一列数值。

其实小波基函数已经具备高斯分布的基本形式，只要将 a 视为 $2\sigma^2$ ，将 b 视为均值 μ ，就可以得到高斯基函数，见表 2.1。不过其物理意义可能不同，高斯基函数的 μ 和 σ^2 用于描述观测数据的均值和方差，不同的 μ 和 σ^2 意味着观测数据可能是多簇中心的。类似的建模方法还可以参考高斯混合模型（见文献^[87]第九章）。因此，如果均值来自于对观测数据的聚类，从而均值参数变为有限个簇中心 c ，则得到径向基函数^[88]，见表 2.1。

表 2.1: 基函数对比

基函数	函数形式	符号解释
傅里叶基函数	$g_m(t) = e^{j\frac{2\pi(m-1)t}{M}}$	这里 $t = t_0, \dots, t_{T-1}$, 离散形式下取 $t = 0, 1, \dots, T-1$ 。根据欧拉公式, $e^{j\omega} = \cos \omega + j \sin \omega$, 三角函数服从 2π 周期, 因此 $\omega \in [0, 2\pi]$ 均匀取 M 个, 也即 $\omega = \frac{2\pi(m-1)}{M}, m = 1, \dots, M$ 。这里取 M 个采样点, 且 $M = T$, 是因为奈奎斯特采样定理 ^[89] 要求频域采样点应与时域采样点数量匹配。
小波基函数	$g_{(a,b)}(t) = \frac{1}{\sqrt{ a }} \phi\left(\frac{t-b}{a}\right)$	这里 $a, b \in \mathbb{R}, a \neq 0$, 二者将基础的小波基函数 $\phi(\cdot)$ 缩放到任意尺度。常见的小波基函数有哈尔小波 ^[86] 、莫莱特小波 ^[90, 91] 等。
高斯基函数	$g_{(\mu,\sigma)}(x_i(t)) = e^{-\frac{(x_i(t)-\mu)^2}{2\sigma^2}}$	这里 $\mu \in \mathbb{R}$ 和 $\sigma \in \mathbb{R}$ 分别为均值和标准差。其系统模型类似于高斯混合模型（见文献 ^[87] 第九章），是一种概率建模。
径向基函数	$g_{(c,\sigma)}(x_i(t)) = e^{-\frac{(x_i(t)-c)^2}{2\sigma^2}}$	这里 $c \in \mathbb{R}$ 和 $\sigma \in \mathbb{R}$ 分别为均值和标准差, 与高斯基函数的不同之处在于均值 c 来自于对观测数据的聚类, 因而只有有限个簇中心。

2.1.3 人工神经网络

在稀疏辨识方法中, 理论上需要选择足够多样的基函数并构建字典矩阵, 从而使得其复合函数能够涵盖需要辨识的目标系统函数, 如图 2.1 所示。但如果基函数的数量 M 远大于采样步数 T , 则使得其辨识难度极大增加, 因而通常需要根据领域知识或一些先验假设以缩小基函数的选择范围。而神经微分方程方法则提供了另一种思路, 也即使用人工神经网络建模目标系统函数。人工神经网络使用任意的基函数复合表示目标系统函数, 从而无需任何领域知识或先验假设。以下针对该方法及其各种变种展开论述。

在论述神经微分方程方法之前, 本文首先论述人工神经网络的建模过程。在 1943 年, W. S. McCulloch 和 W. Pitts^[8] 提出了基于阈值逻辑的神经元数学模型, 也即 M-P 神经元模型, 如图 2.3(a)所示。该神经元模型模拟真实的神经元脉冲放电 (fire) 过程, 其对所有输入进行加权求和, 并根据计算出的值经过函数 $g(\cdot)$ 决定是否放电, 如果放电则输出 1, 如果抑制则输出 0, 因而在图 2.3(a)中, 函数 $g(\cdot)$ 也称为激活函数。然而, 在 M-P 神经元模型中, 其参数是固定的。因此 F. Rosenblatt^[10] 在 M-P 神经元的基础上又提出了感知机模型, 感知机模型和 M-P 神经元模型在结构上是相同的, 区别主要在于感知机模型的参数是可学习的, 其提供了一种学习算法用于更新模型参数 $\theta_i, i = 1, \dots, N$ 。由此, 将感知机模型堆叠起来, 就可以获得多层次感知机模型, 也即前馈神经网络 (Feedforward Neural Network, FNN) 模型, 如图 2.3(b)所示。将其与图 2.1 直观对比, 可以发现二者十分相似, 区别在于图 2.1 中 $g_m, m = 1, \dots, M$ 是预设的基函数, 而图 2.3(b)中 g 可以是任意的非线性激活函数。理论上, 前馈神经网络模型可以近似表示任意的实连续函数^[15, 16]。其深度 (神经网络层数) 和宽度 (每层神经元数) 会影响其近似效果, 当神经元数量固定, 其深度增加会导致其表示能力增加, 且存在指数上界和多项式下界, 这导致了前馈神经网络向深度神经网络演变^[17-19]。

然而, 这种表示能力的增加并不是无止境的。前馈神经网络的参数更新需要经过反向传播 (Backpropagation) 算法^[11, 12], 该算法通过链式求导法则求解每一层神经网络参数的梯度, 然后根据梯度下降原理更新参数, 因而求解的误差会随着层数的增加而累积。参考图 2.3(b), 如果用 $l = 0, \dots, L$ 依次标记输入层、多个隐藏层、输出层, 则前馈神经网络中每个神经元的计算过程可以重写为:

$$\begin{aligned} a_{m_{L-l}}^{(l)} &= b_{m_{L-l}}^{(l)} + \sum_{m_{L-(l-1)}=1}^{M^{(l-1)}} \theta_{m_{L-l} m_{L-(l-1)}}^{(l)} g_{m_{L-(l-1)}}^{(l-1)}, \\ g_{m_{L-l}}^{(l)} &= g(a_{m_{L-l}}^{(l)}), \end{aligned} \quad (2.24)$$

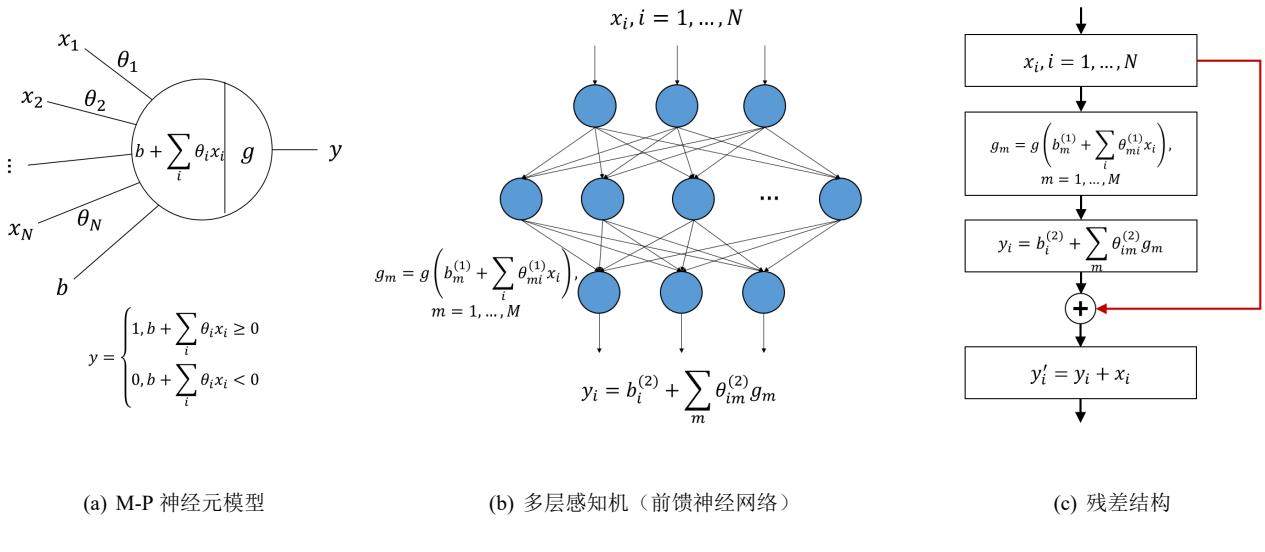


图 2.3: 人工神经网络示意图

其中, $\theta_{m_{L-l} m_{L-(l-1)}}^{(l)}$ 为输入 $g_{m_{L-(l-1)}}^{(l-1)}$ 连接当前神经元输出 $g_{m_{L-l}}^{(l)}$ 的对应参数, $b_{m_{L-l}}^{(l)}$ 为对应偏置参数。对每一层, 上一层的输出是当前层的输入。则反向传播算法原理可描述如下:

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_0 m_1}^{(L)}} &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g_{m_0}^{(L)}} \times \frac{\partial g_{m_0}^{(L)}}{\partial a_{m_0}^{(L)}} \times \frac{\partial a_{m_0}^{(L)}}{\partial \theta_{m_0 m_1}^{(L)}}, \\ \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_1 m_2}^{(L-1)}} &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g_{m_0}^{(L)}} \times \frac{\partial g_{m_0}^{(L)}}{\partial a_{m_0}^{(L)}} \times \left(\sum_{m_1=1}^{M^{(L-1)}} \frac{\partial a_{m_0}^{(L)}}{\partial g_{m_1}^{(L-1)}} \times \frac{\partial g_{m_1}^{(L-1)}}{\partial a_{m_1}^{(L-1)}} \times \frac{\partial a_{m_1}^{(L-1)}}{\partial \theta_{m_1 m_2}^{(L-1)}} \right), \\ \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_2 m_3}^{(L-2)}} &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g_{m_0}^{(L)}} \times \frac{\partial g_{m_0}^{(L)}}{\partial a_{m_0}^{(L)}} \times \left(\sum_{m_1=1}^{M^{(L-1)}} \frac{\partial a_{m_0}^{(L)}}{\partial g_{m_1}^{(L-1)}} \times \frac{\partial g_{m_1}^{(L-1)}}{\partial a_{m_1}^{(L-1)}} \times \left(\sum_{m_2=1}^{M^{(L-2)}} \frac{\partial a_{m_1}^{(L-1)}}{\partial g_{m_2}^{(L-2)}} \times \frac{\partial g_{m_2}^{(L-2)}}{\partial a_{m_2}^{(L-2)}} \times \frac{\partial a_{m_2}^{(L-2)}}{\partial \theta_{m_2 m_3}^{(L-2)}} \right) \right) \\ &\dots \end{aligned} \quad (2.25)$$

其中, $\mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})$ 为作用在目标标签 \mathbf{y} 和第 L 层输出 $\mathbf{g}^{(L)}$ 上的损失函数。同理易得对偏置参数的梯度反向传播过程。易见其每一层参数的梯度求解过程中有大量重合, 因而其可以从输出层向输入层递归求解。

其中, 激活函数 $g(\cdot)$ 早期统一选择为 $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$, 或者双曲正切函数 $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 。 sigmoid 函数的导数取值范围为 $(0, \frac{1}{4}]$, 双曲正切函数的导数取值范围为 $(0, 1]$ 。因而 $l = 1, \dots, L$ 越靠近浅层, 公式 (2.25) 中 $\frac{\partial g_{m_{L-l}}^{(l)}}{\partial a_{m_{L-l}}^{(l)}}$ 等项经过逐层累乘之后, 会使得该层参数的梯度 $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_{L-l} m_{L-(l-1)}}^{(l)}}$ 越趋近于零, 从而参数停止更新。当然也可以设置为其他激活函数, 但如果其导数绝对值可能大于 1, 则经过上述累乘过程, 其梯度也可能变得很大, 从而使得参数更新过程发生震荡。因此, 改用线性整流单元 (Rectified Linear Unit, ReLU) 函数, 也即 $\text{ReLU}(x) = \max(0, x)$, 可以缓解这个问题。ReLU 函数在 $x < 0$ 时导数为 0, 在 $x > 0$ 时导数为 1, 因而经过累乘不会产生梯度“消失”或“爆炸”等问题。但其在 $x = 0$ 时不可导, 因而求解时通常采用次梯度方法, 其处理方法类似公式 (2.16)。也正因为采用了次梯度, 所以反向传播算法求解的误差会随着层数的增加而累积, 从而网络退化, 性能不升反降。

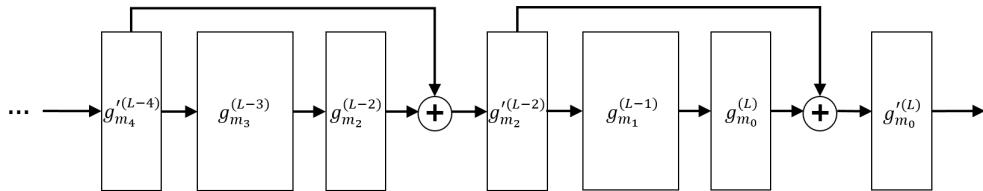


图 2.4: 残差网络示意图

为了缓解这一问题, K. He 等人^[13, 14] 提出了残差网络 (ResNet) 模型, 系统分析了残差结构对于解决深

层前馈神经网络退化问题的作用。K. He 等人认为，如果给定一个深层前馈神经网络，在该网络的基础上增加一层，新网络具有更强的表示能力，对目标函数的近似效果应该更好，然而实验结果却与预期相反。理论上，新增的神经网络层至少可以近似恒等函数，从而使得新网络的效果至少不会低于新增前。因此，说明深层前馈神经网络对恒等函数的近似效果并不好，残差网络模型则基于这一观点，将隐藏层计算结果与输入求和后再输出，从而形成残差结构，如图 2.3(c) 和图 2.4 所示。本文从反向传播算法给出另一种解释，对于图 2.4 中残差网络，可得示例如下：

$$\begin{aligned}
 & \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_3 m_4}^{(L-3)}} \\
 &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g'^{(L)}_{m_0}} \times \frac{\partial g'^{(L)}_{m_0}}{\partial \theta_{m_3 m_4}^{(L-3)}} \\
 &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g'^{(L)}_{m_0}} \times \left(\sum_{m_0=1}^{M^{(L)}} \frac{\partial g'^{(L)}_{m_0}}{\partial g^{(L)}_{m_0}} \times \frac{\partial g^{(L)}_{m_0}}{\partial \theta_{m_3 m_4}^{(L-3)}} + \sum_{m_2=1}^{M^{(L-2)}} \frac{\partial g'^{(L)}_{m_0}}{\partial g'^{(L-2)}_{m_2}} \times \frac{\partial g'^{(L-2)}_{m_2}}{\partial \theta_{m_3 m_4}^{(L-3)}} \right) \\
 &= \sum_{m_0=1}^{M^{(L)}} \frac{\partial \mathcal{L}}{\partial g'^{(L)}_{m_0}} \times \left(\sum_{m_0=1}^{M^{(L)}} \frac{\partial g'^{(L)}_{m_0}}{\partial g^{(L)}_{m_0}} \times \frac{\partial g^{(L)}_{m_0}}{\partial a^{(L)}_{m_0}} \times \left(\sum_{m_1=1}^{M^{(L-1)}} \frac{\partial a^{(L)}_{m_0}}{\partial g^{(L-1)}_{m_1}} \times \frac{\partial g^{(L-1)}_{m_1}}{\partial a^{(L-1)}_{m_1}} \times \left(\sum_{m_2=1}^{M^{(L-2)}} \frac{\partial a^{(L-1)}_{m_1}}{\partial g'^{(L-2)}_{m_2}} \times \frac{\partial g'^{(L-2)}_{m_2}}{\partial \theta_{m_3 m_4}^{(L-3)}} \right) \right) \right. \\
 &\quad \left. + \sum_{m_2=1}^{M^{(L-2)}} \frac{\partial g'^{(L)}_{m_0}}{\partial g'^{(L-2)}_{m_2}} \times \frac{\partial g'^{(L-2)}_{m_2}}{\partial \theta_{m_3 m_4}^{(L-3)}} \right)
 \end{aligned} \tag{2.26}$$

注意，本文这里只推导至第 $L - 2$ 层。但仍可以直观看出， $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{g}^{(L)})}{\partial \theta_{m_3 m_4}^{(L-3)}}$ 可以通过残差结构以更少的乘积项展开至 $\frac{\partial g'^{(L-2)}_{m_2}}{\partial \theta_{m_3 m_4}^{(L-3)}}$ ，从而减少累积的计算误差，以及缓解因激活函数导数的特性导致的梯度“消失”或“爆炸”问题。

因此，以上通过前馈结构和残差结构连接起神经元的模型统称为人工神经网络模型。相比于稀疏辨识方法，人工神经网络在构建时不需要领域知识或先验假设以帮助选择基函数。其激活函数的构造往往是极其简单的，但通过多个此类简单函数的复合却可以近似表示任意的复杂连续函数。因而人工神经网络也被称为“黑盒”模型 (“Black Box” Model)，因为通常其参数经过训练后，无法通过复合这些简单函数重构系统函数，所以其可解释性往往不足，这是此类模型的缺陷之一。

2.1.4 神经微分方程

相比于稀疏辨识方法，人工神经网络在构建时不需要领域知识或先验假设以帮助其选择基函数，因此 T. Chen 等人^[92] 提出基于人工神经网络的系统辨识方法，用人工神经网络建模公式 (1.1)、公式 (1.2) 和公式 (2.1)，因而称其为神经常微分方程 (Neural Ordinary Differential Equation, NODE) 方法。本文将神经常微分方程方法及其建立在各种微分方程上的变种方法统称为神经微分方程方法。

接上文，本文这里标记人工神经网络为 $\mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}) \in \mathbb{R}^N$ ，其输入为 t 时刻样本 $\mathbf{x}(t) \in \mathbb{R}^N$ ，参数为 $\boldsymbol{\theta}$ 。因而由残差结构 (见图 2.3(c)) 作用在样本 $\mathbf{x}(t_j)$, $j = 0, \dots, T - 1$ 上可以得到：

$$\begin{aligned}
 \mathbf{x}(t_{T-1}) &= \mathbf{F}(\mathbf{x}(t_{T-2}), \boldsymbol{\theta})(t_{T-1} - t_{T-2}) + \mathbf{x}(t_{T-2}), \\
 &\vdots \\
 \mathbf{x}(t_2) &= \mathbf{F}(\mathbf{x}(t_1), \boldsymbol{\theta})(t_2 - t_1) + \mathbf{x}(t_1), \\
 \mathbf{x}(t_1) &= \mathbf{F}(\mathbf{x}(t_0), \boldsymbol{\theta})(t_1 - t_0) + \mathbf{x}(t_0), \\
 \mathbf{x}(t_0) &=
 \end{aligned} \tag{2.27}$$

从而从 $\mathbf{x}(t_0)$ 开始连续累加部分项，可以得到：

$$\mathbf{x}(t) = \sum_{t_{j-1} < t_j \leq t} \mathbf{F}(\mathbf{x}(t_{j-1}), \boldsymbol{\theta})(t_j - t_{j-1}) + \mathbf{x}(t_0). \tag{2.28}$$

这里 $t \in \{t_0, t_1, \dots, t_{T-1}\}$ 。因此，如果对 $j = 1, \dots, T-1$ ，有 $t_j - t_{j-1} = 1$ ，则得到标准残差网络模型（见图 2.4）。如果 $\max\{t_j - t_{j-1} | j = 1, \dots, T-1\} \rightarrow 0$ ，则得到残差网络的连续形式：

$$\mathbf{x}(t) = \int_{t_0}^t \mathbf{F}(\mathbf{x}(s), \boldsymbol{\theta}) ds + \mathbf{x}(t_0). \quad (2.29)$$

这里 $t \in [t_0, t_{T-1}]$ ，从而等号两边对 t 求导，可以得到神经常微分方程：

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}). \quad (2.30)$$

求解公式 (2.30) 中参数的梯度主要有两种算法，一种是时序反向传播 (Backpropagation Through Time, BPTT) 算法，另一种是伴随灵敏度 (Adjoint Sensitivity) 算法。时序反向传播算法是反向传播算法（见公式 (2.25)）在时间序列应用中的一个特例，其广泛应用于各种具有循环结构^[93, 94] 的神经网络中，比如长短时记忆 (Long Short-Term Memory, LSTM) 网络^[95, 96]、门控循环单元 (Gated Recurrent Unit, GRU) 网络^[97] 等等。本文利用时序反向传播算法求解公式 (2.30) 中参数，如下：

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \boldsymbol{\theta}} &= \sum_{j=1}^{T-1} \frac{\partial \mathcal{L}(\mathbf{y}(t_j), \mathbf{x}(t_j))}{\partial \boldsymbol{\theta}} \\ &= \sum_{j=1}^{T-1} \frac{\partial \mathcal{L}(\mathbf{y}(t_j), \mathbf{x}(t_j))}{\partial \mathbf{x}(t_j)} \times \int_{t_0}^{t_j} \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} dt. \end{aligned} \quad (2.31)$$

这里 \mathbf{y} 是目标标签，也即 $t = t_1, \dots, t_{T-1}$ 等时刻的 $\mathbf{x}(t)$ 的观测状态值。注意 t_0 是初值，不算在损失函数 \mathcal{L} 中。

另一种解法是伴随灵敏度算法，该算法的空间复杂度比时序反向传播算法的更低。由于 $\mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta})$ 的参数不随时间改变，所以令 $\boldsymbol{\theta}(t) \equiv \boldsymbol{\theta} \in \mathbb{R}^M$ ，且 $\frac{d\boldsymbol{\theta}(t)}{dt} = \mathbf{0}$ ，从而可以构造

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\theta}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t)) \\ \mathbf{0} \end{bmatrix} = f_{\text{aug}}(\mathbf{x}(t), \boldsymbol{\theta}(t)). \quad (2.32)$$

然后定义伴随状态变量，如下：

$$\mathbf{a}_{\text{aug}}(t) = \begin{bmatrix} \mathbf{a}_x(t) & \mathbf{a}_{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x}(t)} & \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \boldsymbol{\theta}(t)} \end{bmatrix} = \frac{d\mathcal{L}(\mathbf{y}, \mathbf{x})}{d[\mathbf{x}(t), \boldsymbol{\theta}(t)]} \quad (2.33)$$

从而令 $\mathbf{z}(t) = [\mathbf{x}(t), \boldsymbol{\theta}(t)]$ ，可得：

$$\begin{aligned} \mathbf{a}_{\text{aug}}(t) &= \frac{d\mathcal{L}(\mathbf{y}, \mathbf{x})}{d\mathbf{z}(t + \Delta t)} \times \frac{d\mathbf{z}(t + \Delta t)}{d\mathbf{z}(t)} \\ &= \frac{d\mathcal{L}(\mathbf{y}, \mathbf{x})}{d\mathbf{z}(t + \Delta t)} \times \frac{d[\mathbf{z}(t) + f_{\text{aug}}(\mathbf{z}(t))\Delta t + o(\Delta t)]}{d\mathbf{z}(t)} \\ &= \frac{d\mathcal{L}(\mathbf{y}, \mathbf{x})}{d\mathbf{z}(t + \Delta t)} \times \left[\mathbf{1} + \frac{df_{\text{aug}}(\mathbf{z}(t))}{d\mathbf{z}(t)} \Delta t + o(\Delta t) \right] \\ &= \mathbf{a}_{\text{aug}}(t + \Delta t) + \mathbf{a}_{\text{aug}}(t + \Delta t) \frac{df_{\text{aug}}(\mathbf{z}(t))}{d\mathbf{z}(t)} \Delta t + o(\Delta t). \end{aligned} \quad (2.34)$$

进而移项并对 Δt 求极限得 $\mathbf{a}_{\text{aug}}(t)$ 导数，如下：

$$\begin{aligned} \frac{d\mathbf{a}_{\text{aug}}(t)}{dt} &= \lim_{\Delta t \rightarrow 0} \frac{\mathbf{a}_{\text{aug}}(t + \Delta t) - \mathbf{a}_{\text{aug}}(t)}{\Delta t} \\ &= -\mathbf{a}_{\text{aug}}(t) \frac{df_{\text{aug}}(\mathbf{z}(t))}{d\mathbf{z}(t)} \\ &= - \begin{bmatrix} \mathbf{a}_x(t) & \mathbf{a}_{\theta}(t) \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \mathbf{x}(t)} & \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{a}_x(t) \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \mathbf{x}(t)} & -\mathbf{a}_x(t) \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{d\mathbf{a}_x(t)}{dt} & \frac{d\mathbf{a}_{\theta}(t)}{dt} \end{bmatrix}. \end{aligned} \quad (2.35)$$

从而对于 $t \in [t_{j-1}, t_j]$, 可得伴随状态变量:

$$\mathbf{a}_x(t) = \int_{t_j}^t \frac{d\mathbf{a}_x(t)}{dt} dt + \mathbf{a}_x(t_j) = - \int_{t_j}^t \mathbf{a}_x(t) \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \mathbf{x}(t)} dt + \mathbf{a}_x(t_j). \quad (2.36)$$

进而可得参数梯度:

$$\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \boldsymbol{\theta}(t)} = \int_{t_0}^{t_{T-1}} \frac{d\mathbf{a}_\theta(t)}{dt} dt = \int_{t_{T-1}}^{t_0} \mathbf{a}_x(t) \frac{\partial \mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}(t)} dt. \quad (2.37)$$

因此, 伴随灵敏度算法实际上先将 $\mathbf{x}(t)$ 从时刻 t_0 迭代求解至时刻 t_{T-1} , 其间会计算 $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x}(t_j)}$, $j = 0, \dots, T-1$, 再沿反方向从 t_{T-1} 累加伴随状态变量值至时刻 t_0 , 如图 2.5 所示。注意在 $t_j, j = 0, \dots, T-1$ 时刻, 伴随状态变量需要加上 $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x}(t_j)}$ 以校正伴随状态变量在反向累加过程中积累的误差 (参考公式 (2.36))。其中, $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x}(t_j)}$ 可由一些深度学习框架 (如 Pytorch、TensorFlow) 通过自动微分技术获得。

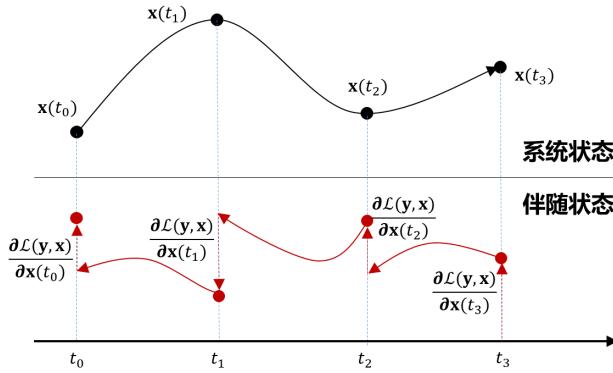


图 2.5: 伴随灵敏度算法示意图^[92]

此外, 对于伴随灵敏度算法还值得注意的是, 文献^[92] 中强调该算法和时序反向传播算法 (见公式 (2.31)) 的时间复杂度均为 $\mathcal{O}(\tilde{T})$, 这里 \tilde{T} 为微分方程求解器的迭代步数。但是, 文献^[92] 还认为该算法的空间复杂度仅为 $\mathcal{O}(1)$, 理由是伴随状态变量在任意时刻都只占用相同的存储。而时序反向传播算法的空间复杂度为 $\mathcal{O}(\tilde{T})$, 因为反向传播算法需要保存每个时刻的梯度, 见公式 (2.31)。然而实际上, 伴随灵敏度算法同样需要保存每个时刻的 $\frac{\partial \mathcal{L}(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x}(t_j)}$ 用于校正, 才能反向累加伴随状态得到正确的参数梯度, 因而其空间复杂度其实同样为 $\mathcal{O}(\tilde{T})$ 。

2.1.5 物理信息神经网络

要使得系统辨识具有足够的泛化性, 稀疏辨识方法通过引入知识筛选基函数, 从而提高函数近似的准确性。而神经微分方程方法则不需要知识, 其仅仅依靠简单函数的复合就可以近似微分方程中的目标函数, 但同时由于其“黑盒”特性, 从而可解释性较低, 这对于实际应用往往是不可接受的。因此, 物理信息神经网络 (Physics-Informed Neural Network, PINN) 方法^[98, 99] 折中了二者, 一方面利用人工神经网络的函数近似原理逼近任意的实连续函数, 另一方面为人工神经网络嵌入知识, 解决其可解释性不足的问题。这里“知识”延续了物理符号系统^[4-6] 中对知识的定义, 也即一组符号及作用在符号上的规则, 其规则在物理信息神经网络中通常为偏微分方程。因此, 一种基于物理信息神经网络的系统辨识范式可被定义如下:

1. (物理) 知识和数据;
2. 人工神经网络。

为方便描述, 这里在系统的观测时序轨迹数据 $\mathbf{x} \in \mathbb{R}^{T \times N}$ (见公式 (2.1)) 的基础上额外引入一个目标标签变量 Y , 定义其观测数据为 $\mathbf{y} = [y(t_0), \dots, y(t_{T-1})]^\top$ 。其也可以视为从原观测数据中取出一列作为目

标标签变量的观测数据，因而不失其一般性。进而给出知识，也即一个偏微分方程，定义如下：

$$f_{PDE}(X_1, \dots, X_N, Y, \frac{\partial Y}{\partial X_1}, \dots, \frac{\partial Y}{\partial X_N}, \frac{\partial^2 Y}{\partial X_1^2}, \dots, \frac{\partial^2 Y}{\partial X_1 \partial X_N}, \frac{\partial^2 Y}{\partial X_2 \partial X_1}, \dots, \frac{\partial^2 Y}{\partial X_N^2}, \dots) = 0, \quad (2.38)$$

从而构建人工神经网络为 $\mathbf{F}(\mathbf{x}(t), \boldsymbol{\theta}) \in \mathbb{R}^D$ ，其输入为 t 时刻样本 $\mathbf{x}(t) \in \mathbb{R}^N$ ，参数为 $\boldsymbol{\theta}$ ，输出为 $y(t)$ 和其他 $f_{PDE}(\dots)$ 中存在的元素（除了 X_1, \dots, X_N ）的积分的值， D 为输出元素个数。举个例子，如图 2.6 所示，如果要求 $\frac{\partial^2 Y}{\partial X_2 \partial X_3}$ ，则需要定义输出为 $\frac{\partial Y}{\partial X_2}$ 。

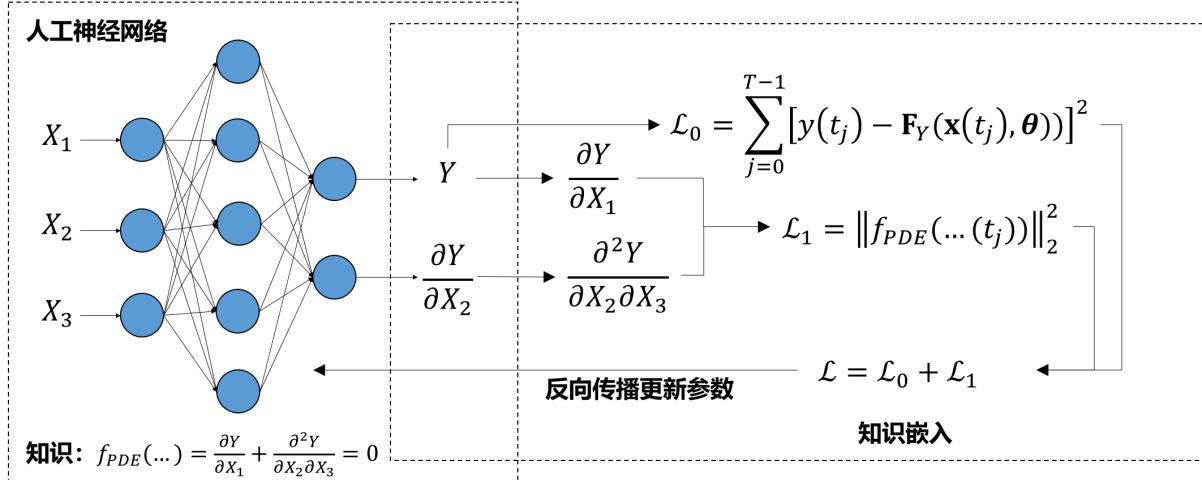


图 2.6：物理信息神经网络示意图。

从而可以定义两个基于重构误差的损失函数，首先是对变量 Y 的观测数据的重构，如下：

$$\mathcal{L}_0 = \sum_{j=0}^{T-1} [y(t_j) - \mathbf{F}_Y(\mathbf{x}(t_j), \boldsymbol{\theta})]^2, \quad (2.39)$$

其中， $\mathbf{F}_Y(\mathbf{x}(t_j), \boldsymbol{\theta}) \in \mathbb{R}$ 表示人工神经网络关于变量 Y 的输出值。然后是对知识的重构，如下：

$$\mathcal{L}_1 = \sum_{j=0}^{T-1} [f_{PDE}(\dots(t_j))]^2. \quad (2.40)$$

其中， $f_{PDE}(\dots(t_j)) \in \mathbb{R}$ 表示 $f_{PDE}(\dots)$ 作用在其中元素于 t_j 时刻的值上，计算结果来自于人工神经网络的输出。从而得到联合损失函数，如下：

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_1, \quad (2.41)$$

这里 λ 是惩罚因子。由于公式 (2.38)-(2.41) 具有一般性，因此理论上任意的以偏微分方程形式描述的知识都可以嵌入人工神经网络中。

求解物理信息神经网络参数的关键在于利用反向模式的自动微分技术^[100] 求解偏导数，其原理基于反向传播算法中建立在人工神经网络上的链式求导法则^[11, 12]，可参考公式 (2.25)。由于在公式 (2.38)-(2.41) 中已经构建了从输入 X_1, \dots, X_N 到输出 $f_{PDE}(\dots)$ 中部分元素的端到端架构，因而可以通过反向传播算法倒推网络中各个节点偏导数，包括输入节点。比如在图 2.6 中，可得：

$$\frac{\partial^2 Y}{\partial X_2 \partial X_3} = \sum_{m=1}^M \frac{\partial}{\partial g_m} \left(\frac{\partial Y}{\partial X_2} \right) \times \frac{\partial g_m}{\partial a_m} \times \frac{\partial a_m}{\partial X_3}. \quad (2.42)$$

其中， M 是该隐藏层的神经元数， $g_m = g(a_m)$ 为该神经元输出， $g(\cdot)$ 是激活函数，更多原理细节可参考公式 (2.25)。基于反向模式自动微分技术，理论上可以求解 $f_{PDE}(\dots)$ 中任意元素。当人工神经网络参数收敛，则该模型蕴含了关于系统的（物理）知识。但仍需注意的是，使用了人工神经网络后，尤其是多层网络，系统函数无法明确描述，因此稀疏辨识等方法仍然具有其在可解释性上的优势。

2.2 复杂系统的网络表示

系统并不客观存在，而是人们认识和改造世界的一种方式。基于数据驱动的系统辨识技术可以辨识一个形如公式(1.1)的复杂系统，并且通过人工神经网络理论上可以近似表示任意的实连续系统函数，因此，人工神经网络也可以视为对复杂系统的一种网络建模，也即一个人工神经网络对应一个系统。简言之，“网络即系统”。那么，能否通过研究一个网络从而研究一个系统？本文针对该问题进行讨论。

2.2.1 从人工神经网络到复杂网络

人脑通过各种感觉器官获取感官信息，进而通过脑网络(brain network)或者说神经回路(neural circuits)实现对真实世界的建模和辨识，从而能理解和预测真实世界的变化规律，最终指导并付诸实践。本文将脑网络、神经回路等名称统称为神经网络，以区分人工神经网络。然而，真实的神经网络却并不像公式(2.30)一样通用，它有更加具体的表示形式，本文列举了一些具体的动力学形式，见表2.2。

与公式(2.30)相比，真实的神经网络动力学往往具有如下特点：(1)激活函数不一定连续。比如在M-P神经元模型^[8]中，激活函数采用了阶跃函数。还比如在表2.2中，Integrate-and-Fire神经网络采用了狄拉克Delta函数模拟神经元尖峰放电。(2)神经元器件的模型可能是高维的，也即神经元并不是构成神经网络的最小单元，其可以继续再分。比如在表2.2中，Hodgkin-Huxley模型将神经元建模成由钠、钾离子电流值和泄露电流值控制的膜电位变化关系。Rulkov Map神经动力学将神经元建模为了膜电位和离子浓度两部分，当离子浓度到达峰值，则膜电位开始变化，神经元内部会反复这个过程。(3)神经元及神经元之间的连接往往是稀疏的。就如稀疏辨识一样，少量神经元及其连接对模型产生关键影响^[111,112]。(4)神经元之间信号传递由于传递介质不同，也会有不同的延迟^[111,112]。比如神经元之间通过电信号实现快速通信，而化学信号则要慢很多。神经元之间通信是异步的，而不是像公式(2.30)中，每一层神经元同步接

表2.2: 神经网络动力学对比

网络模型	动力学	符号解释
连续时间循环 神经网络 ^[101-103]	$\frac{dx(t)}{dt} = -\frac{x(t)}{\gamma} + \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t))$ $\mathbf{y}(t) = o(\mathbf{x}(t))$	从公式(2.1)中划分出状态变量 \mathbf{x} 、输入变量 \mathbf{u} 和输出变量 \mathbf{y} ，如图2.7(a)所示。 $o(\cdot)$ 用于输出 \mathbf{y} ， γ 是时间常数。
Integrate-and-Fire 神经网络 ^[104,105]	$\frac{dx_i(t)}{dt} = -\frac{x_i(t)}{\gamma} + \sum_j A_{ij} \sum_k \delta(t - t_j^k - \tau)$	$\delta(\cdot)$ 是狄拉克 δ 函数， t_j^k 是神经元 j 的第 k 次放电时间。 γ 是时间常数， τ 是轴突传递时延。 A_{ij} 是神经元邻接矩阵的 i 行 j 列值
Hodgkin-Huxley 神经回路 ^[106,107]	$C_M \frac{dE_i}{dt} = -I_{ion} + \sum_j I_{ext,j}$ $I_{ion} = \frac{E_i - E_{Na}}{R_{Na}} + \frac{E_i - E_K}{R_K} + \frac{E_i - E_L}{R_L}$ $I_{ext,j} = A_{ij}(V_j - V_i)$	如图2.7(b)所示， E 是膜电位， R 是电阻， I 是电流， C_M 是电容。 Na 是钠离子， K 是钾离子， L 是泄露值。 A_{ij} 是神经元邻接矩阵的 i 行 j 列值，其物理意义是两个神经元之间电阻值的倒数。 I_{ion} 提供了相反的电流用来平衡外部电流 $\sum_j I_{ext,j}$ 的冲击。
Rulkov Map 神经动力学 ^[108-110]	$x_i(n+1) = \frac{\alpha}{1+x_i(n)^2} + x'_i(n) - \sum_j L_{ij}x_j(n)$ $x'_i(n+1) = x'_i(n) - \beta x_i(n) - \sigma$	每个神经元由两个变量 x 和 x' 以离散时间表示， x 表示膜电位， x' 表示离子浓度变化， n 是步数。 L_{ij} 是拉普拉斯矩阵的 i 行 j 列值。 α 、 β 、 σ 均为超参数，控制它们引发神经元连续放电和混沌现象，如图2.7(c)所示。

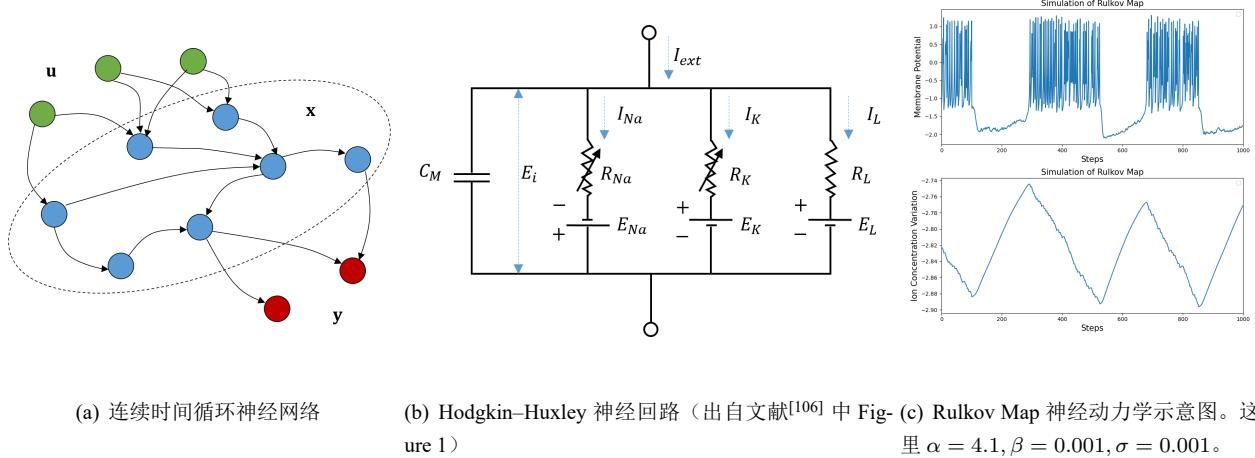


图 2.7: 神经网络示意图

收上一层的输出作为本层输入，并同步激活输入到下一层。

然而，即便如公式 (2.30) 一般的人工神经网络模型与真实的神经网络有诸多不同，现有的人工神经网络依然表现出了一定程度的智能。因此，可以认为其二者共同之处才是形成智能的关键。正如连接主义观点认为，构成“激励 - 响应”的连接关系是形成智能的关键，如图 2.8(a) 所示，而智能则是从复杂的网络结构中涌现出的结果^[113, 114]。因此，将表 2.2 建模成复杂网络动力学方程，如下：

$$\frac{dx_i(t)}{dt} = F(x_i(t)) + \sum_{j=1}^N A_{ij} G(x_i(t), x_j(t)), \quad (2.43)$$

其中， $x_i(t), i = 1, \dots, N$ 是复杂网络节点在时刻 t 的状态值， $F(\cdot)$ 是自驱动函数， G 是交互函数， A_{ij} 是网络邻接矩阵的 i 行 j 列值。由于其主要刻画网络节点之间的交互特性，因而其又称为交互网络 (interactive network) 或者耦合网络 (coupling network)。

公式 (2.43) 中的网络结构主要有三类。其一是 Erdős-Rényi (ER) 随机图^[115]，其还由 E. N. Gilbert^[116]同年提出。简言之，就是二值网络中的连边随机出现，出现概率均服从二项分布，也即以概率 p 出现连边且值为 1，以概率 $1 - p$ 不出现连边且值为 0。由此，定义一个图 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ，其中 $\mathbf{V} = \{X_1, \dots, X_N\}$ ， $\mathbf{E} = \{e_{ij} | A_{ij} = 1\}$ 。因此，ER 随机图的度分布 $P(k) = C_M^k p^k (1-p)^{M-k}$ ，这里 $M = \frac{N(N-1)}{2}$ ，也即从所有可能的 M 个连边中随机选择出现 k 个连边，所以其度分布服从二项分布。

其二是小世界网络 (small-world network)^[117]，简言之，就是随着网络连边出现的随机性增大，其平均路径长度和聚类系数都会逐渐降低并趋近于一个固定值。小世界网络的构造方法主要有两种，(1) 如图 2.8(b) 所示，首先依平均度 $\langle k \rangle$ 构造一个规则网络，网络中每个节点都连接其他 $\langle k \rangle$ 个节点。然后对网络中的每一条连边依概率 p 重新连接至其他节点。若如此做，则当 $p \rightarrow 1$ ，平均路径长度趋近于 $\ln N / \ln \langle k \rangle$ ，聚类系数趋近于 $\langle k \rangle / N$ 。其中聚类系数定义为 $\frac{1}{N} \sum_{i=1}^N \frac{2|\{e_{u,v} | u, v \in \mathcal{N}(i)\}|}{|\mathcal{N}(i)|(|\mathcal{N}(i)|-1)}$ ，这里 $\mathcal{N}(i) = \{j | e_{ij} \in \mathbf{E}\}$ 表示节点 i 的邻居集合。(2) 第二种方法^[118] 是在方法 (1) 的规则网络中依概率 p 随机添加新连边，且不删除原有规则网络的连边。因此既保持了规则网络的结构，又添加了新的远程连接。通常方法 (2) 比方法 (1) 拥有更高的聚类系数。

其三是无标度网络 (scale-free network)^[119]。简言之，该网络的度分布呈现幂律分布，也即 $P(k) = k^{-\gamma}$ ，如图 2.8(c) 所示。无标度网络的构造方法是，首先初始化一个包含少量节点的网络，每个节点之间随机连接，然后逐次添加新节点，每个新节点与网络中已存在的节点连接。每次连接的节点根据其度数选择，度数越高，越容易被选中。因此，当添加节点数足够多时，图中会有少量节点拥有较高的度。

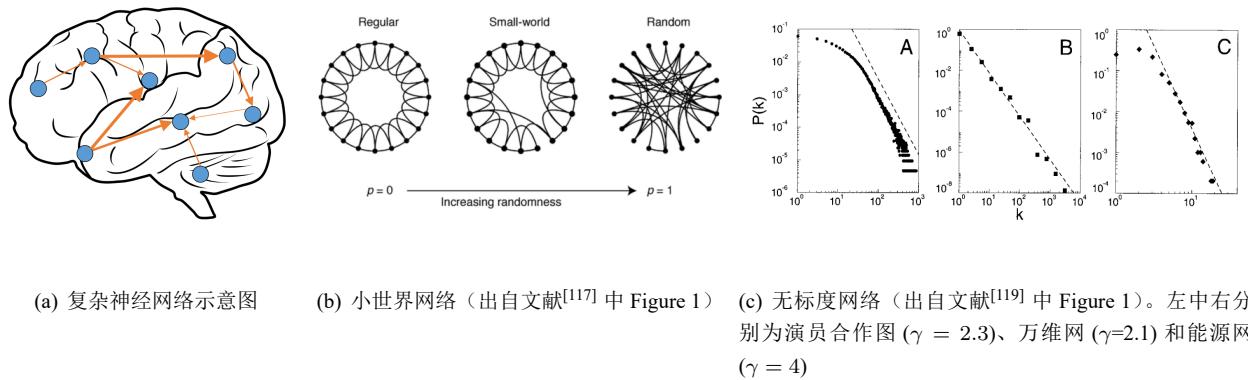


图 2.8: 复杂网络示意图

2.2.2 复杂网络协同性和可控性

可控即安全，可控性是网络系统安全的核心保障。当满足一定条件，形如公式 (2.43) 网络动力学就会具有协同性^[120]，也即网络系统节点的状态逐渐趋同，这在自然界中也是广泛存在的涌现现象。比如经典的 Kuramuro 动力学模型^[121, 122]：

$$\frac{dx_i(t)}{dt} = \omega_i + \frac{c}{N} \sum_{j=1}^N A_{ij} \sin(x_j(t) - x_i(t)), \quad (2.44)$$

其中 $\omega_i \in \mathbb{R}$ 是振子 $i = 1, \dots, N$ 的自然频率， $c \in \mathbb{R}$ 是耦合系数。其协同现象如图 2.9(a) 所示，所有节点的状态均趋于稳定，也即速度 $\frac{dx_i(t)}{dt}$ 趋于零。

当然，系统节点状态协同在自然界中只是一种偶发现象，为了能够使系统稳定地协同，通常需要引入外在干预，由此引出网络系统可控性的问题。该问题最早可以追溯到线性系统可控性问题，线性控制系统定义如下：

$$\frac{d\mathbf{x}(t)}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t), \quad (2.45)$$

这里 $\mathbf{x}(t) = [x_1(1), \dots, x_N(t)]$ 是系统状态变量， $\mathbf{u}(t) = [u_1(t), \dots, u_M(t)]$ 是控制输入变量， $A \in \mathbb{R}^{N \times N}$ 和 $B \in \mathbb{R}^{N \times M}$ 是连接矩阵 (wiring diagram)，描述了 \mathbf{x} 和 \mathbf{u} 之间的连接方式和连接强度。因此，公式 (2.45) 可以表示成网络系统，如图 2.9(b) 所示。由此可得可控性矩阵，如下

$$C_{\text{control}} = [B, AB, A^2B, \dots, A^{N-1}B]. \quad (2.46)$$

可以定义系统可控如下，

定义 7 (可控性). 如果对任意系统初始状态 $\mathbf{x}(t_0)$ ，可以控制 \mathbf{u} ，使得 $\mathbf{x}(t), t > t_0$ ，转移到任意系统状态。

因此，可得线性系统可控性定理：

定理 2.2.1 (线性系统可控性^[123]). 如果线性控制系统 (公式 2.45) 满足

$$\text{rank}(C_{\text{control}}) = N, \quad (2.47)$$

则该系统是可控的。这里 $\text{rank}(\cdot)$ 是矩阵秩。

证明. 易得公式 (2.45) 的解为

$$\mathbf{x}(t) = e^{At} \left[\int_{t_0}^t e^{-At} B\mathbf{u}(t) dt + e^{-At_0} \mathbf{x}(t_0) \right], \quad (2.48)$$

由哈密顿-凯莱定理可得

$$e^{-At} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \cdots + \alpha_{N-1} A^{N-1}, \alpha_0, \dots, \alpha_{N-1} \in \mathbb{R}. \quad (2.49)$$

代入公式(2.48)可得

$$\begin{aligned} e^{-At}\mathbf{x}(t) - e^{-At_0}\mathbf{x}(t_0) &= \int_{t_0}^t \left(\sum_{j=0}^{N-1} \alpha_j A^j \right) B\mathbf{u}(t) dt \\ &= \int_{t_0}^t [B, AB, A^2B, \dots, A^{N-1}B] [\alpha_0 \mathbf{u}^\top(t), \dots, \alpha_{N-1} \mathbf{u}^\top(t)]^\top dt. \end{aligned} \quad (2.50)$$

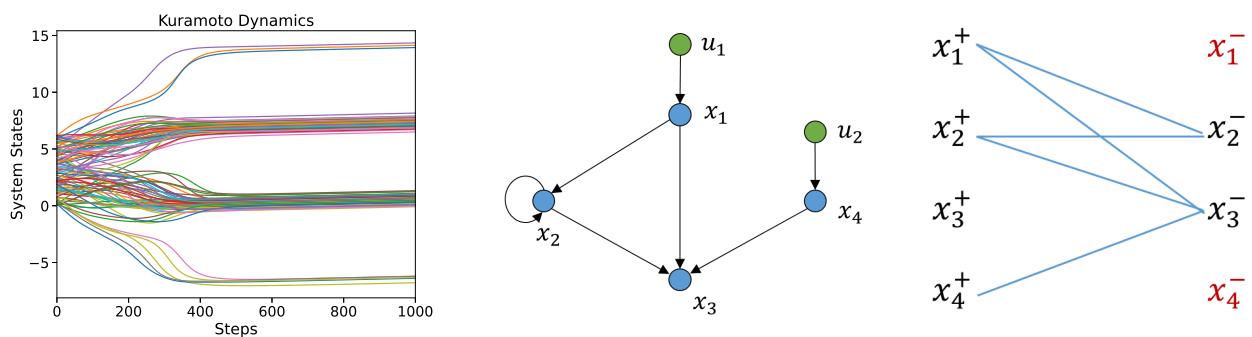
依定义7可知, 对任意的 $\mathbf{x}(t_0)$ 和 $\mathbf{x}(t)$, 上式均有解, 因此可得 $\text{rank}(C_{\text{control}})$ 满秩。 \square

尽管定理2.2.1给出了线性系统可控的充要条件, 但通过该条件判定可控性并不容易做到, 因为很多情况下矩阵 A 和 B 是未知的。因此, C. Lin^[124] 提出了结构可控性条件, 其认为即便 A 的具体参数未知, 但如果能知道其网络的连接方式, 依然能够判断其是否可控。而一个网络系统是否可控, 主要取决于(1)该网络中是否存在节点不可到达, 对不可到达的节点需要增设控制节点; (2)是否存在扩张, 如图2.9(b)所示, 节点 x_1 分叉到节点 x_2 和节点 x_3 。如果存在扩张, 则意味着上游节点的输出信号可能无法覆盖足够多的下游节点。由此, Y. Liu^[125] 等人提出结构可控性的度量方法, 并提出通过二分图最大匹配算法找到使得系统可控的最小数量的控制节点(driver nodes)。如图2.9(c)所示, 将图2.9(b)中节点按“出(+)”-入(-)”分成左右两边并连接起来, 左边每个节点只选择一次, 其对应一条连接的右边节点会被选中, 尽可能不重复, 从而会匹配到最大的数量的被控节点, 而没有被匹配到的节点就是最小数量的控制节点。因此, 其可控性可以量化为:

$$n_D = \frac{\max\{1, N - |\mathbf{E}^*|\}}{N}, \quad (2.51)$$

这里 $|\mathbf{E}^*|$ 是最大匹配的连边数, 也是最小数量的控制节点数。因此, 根据算法原理可以得到两个结论: (1)稀疏图可控性较低, 因为可供匹配的连边较少。(2)可控性高的网络系统会尽可能避免高出度节点, 因为会产生扩张, 使得可以匹配的连边数变少。此外, 文献^[125]主要针对有向网络, Z. Yuan^[126]等人针对无向网络依然得出类似的可控性结论, 也即对于一个无向网络 A , 其可控性为

$$n_D = \frac{\max\{1, N - \text{rank}(A)\}}{N}. \quad (2.52)$$



(a) Kuramoto 动力学。这里一共 100 个振子, 采用 ER 随机图^[115,116]连接, 连接概率 $p = 0.2$ 。 $K = 3$, ω_i 由均匀分布 $U(0, 1)$ 随机选取。

(b) 网络可控性示意图^[125]。

(c) 二分图最大匹配算法示意图^[125]。

图 2.9: 复杂网络协同性和可控性示意图

进而, X. Wang 和 G. Chen^[127,128] 将公式 (2.45) 推广到非线性形式, 如下:

$$\begin{aligned}\frac{d\mathbf{x}_i(t)}{dt} &= F(\mathbf{x}_i(t)) + c \sum_{j=1}^N A_{ij} \Gamma \mathbf{x}_j(t) - cd\Gamma(\mathbf{x}_i - \bar{\mathbf{x}}), i = 1, \dots, M, \\ \frac{d\mathbf{x}_i(t)}{dt} &= F(\mathbf{x}_i(t)) + c \sum_{j=1}^N A_{ij} \Gamma \mathbf{x}_j(t), i = M+1, \dots, N\end{aligned}\quad (2.53)$$

这里 $c > 0$ 表示耦合强度, $d > 0$ 表示反馈增益, $\bar{\mathbf{x}} \in \mathbb{R}^{N_{\text{node}}}$ 是协同的目标。值得注意的是, 这里用 $\mathbf{x}_i(t) \in \mathbb{R}^{N_{\text{node}}}$ 描述一个节点的状态值, 因而 $\Gamma \in \mathbb{R}^{N_{\text{node}} \times N_{\text{node}}}$ 是 0-1 矩阵, 用于描述变量内部状态间的耦合。尽管如此, 实际上公式 (2.53) 仍然可以转换为标准的复杂网络动力学模型 (见公式 (2.43))。文献^[127,128] 分别证明了, 当网络为无标度网络或小世界网络, 则只需要控制部分节点, 就可以实现全局的协同一致。W. Yu 等人^[129] 将其推广到任意复杂网络情况。W. Yu 等人^[130] 还将其推广到二阶非线性系统的协同。此外, 文献^[131,132] 研究了动态网络下的协同一致控制, 其观点认为动态切换网络比静态网络具有更高可控性。

进而, M. T. Angulo 等人^[133] 研究了具备

$$\frac{d\mathbf{x}(t)}{dt} = F(\mathbf{x}(t)) + G(\mathbf{x}(t))\mathbf{u}(t) \quad (2.54)$$

形式的网络系统的结构可控性条件, 这里 $\mathbf{x}(t) \in \mathbb{R}^N$, $\mathbf{u}(t) \in \mathbb{R}^M$, 函数 $F(\cdot)$ 和 $G(\cdot)$ 描述了网络连接关系。M. T. Angulo 等人^[134] 还研究了具备

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.55)$$

形式的一般性非线性系统的结构可控性, 该系统同样服从一个网络的连接关系, 但在公式中没有显式表示。R. M. D'Souza 等人^[135] 认为公式 (2.54) 和公式 (2.55) 在一些微弱的条件下, 其结构可控性是等价的。

2.2.3 复杂网络弹性和低秩性

复杂即是简单, 复杂的现象往往从其内部的简单事物中衍生出来, 而复杂网络弹性 (resilience) 则具备这样更基本的性质。复杂网络弹性有许多同类定义, 比如指网络系统对外部攻击的忍耐度 (tolerance)^[136], 当外部攻击超过忍耐度, 则网络系统崩溃。还比如网络系统对于内部误差和外部扰动的抵御能力, 也即鲁棒性 (robustness)^[137]。还比如, 随着网络参数变化, 系统的轨迹发生改变, 从而产生多个分支 (bifurcation)^[138], 该现象又称为相变 (phase transition)。举个例子, 对于一维系统 (参考公式 (1.1))

$$\dot{x} = f(x, \beta), \quad (2.56)$$

这里 β 是系统参数, 且忽略时间参数。则其相变点 β^c 需要满足:

$$\begin{cases} f(x^c, \beta^c) = 0, \\ \frac{\partial f(x, \beta^c)}{\partial x} \Big|_{x=x^c} = 0. \end{cases} \quad (2.57)$$

因此, 对于如图 2.10(a)所示的示例, 易得其相变点 $\beta^c = 0$ 。当 $\beta < \beta^c$ 时, 系统存在一个稳定点, 但当 $\beta \geq \beta^c$ 时, 系统不存在稳定点, 因而崩溃。总而言之, 本文称上述现象或性质均为复杂网络弹性。

然而, 复杂网络系统 (见公式 (2.43)) 通常是高维的 (多变量的), 直接对其分析弹性十分困难。J. Gao 等人^[139] 提出了一种针对网络系统的降维方法, 可以将其降至一维, 从而利用公式 (2.57) 中的判据分析复杂网络系统的弹性。忽略时间参数而不失一般性, 可以定义:

$$x_{\text{eff}} = \frac{\mathbf{1}^\top A \mathbf{x}}{\mathbf{1}^\top A \mathbf{1}} = \frac{\sum_{n=1}^N k_n^{\text{out}} x_n}{\sum_{n=1}^N k_n^{\text{out}}}. \quad (2.58)$$

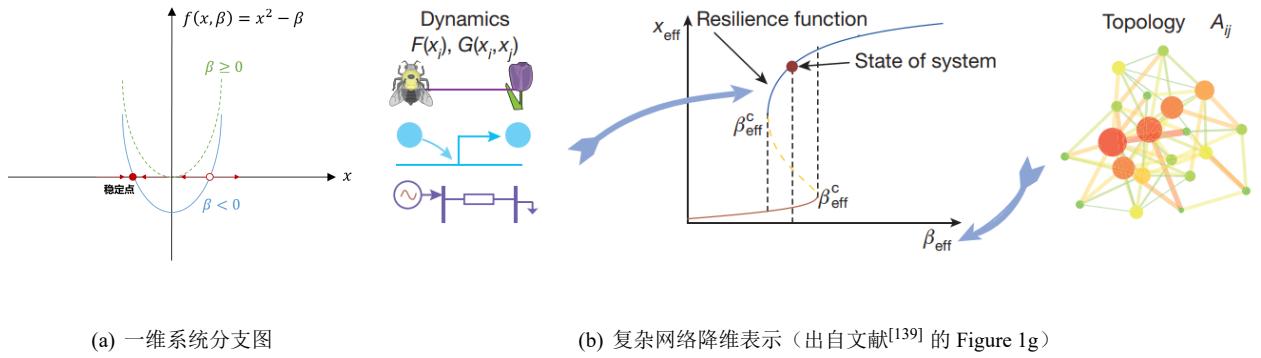


图 2.10: 复杂网络弹性示意图

这里 $\mathbf{x} \in \mathbb{R}^N$, $A \in \mathbb{R}^{+N \times N}$ 是邻接矩阵, 于是 $k_n^{\text{out}} = \sum_{i=1}^N A_{in}$, 也即节点 n 的带权出度。同样地, 定义节点 n 的带权入度为 $k_n^{\text{in}} = \sum_{j=1}^N A_{nj}$, 于是类似可得

$$\beta_{\text{eff}} = \frac{\mathbf{1}^\top A \mathbf{k}^{\text{in}}}{\mathbf{1}^\top A \mathbf{1}} = \frac{\sum_{n=1}^N k_n^{\text{out}} k_n^{\text{in}}}{\sum_{n=1}^N k_n^{\text{out}}}, \quad (2.59)$$

这里 $\mathbf{k}^{\text{in}} = [k_1^{\text{in}}, \dots, k_N^{\text{in}}]^\top$ 。于是可以将公式 (2.43) 降低至一维, 如下:

$$\dot{x}_{\text{eff}} = F(x_{\text{eff}}) + \beta_{\text{eff}} G(x_{\text{eff}}, x_{\text{eff}}). \quad (2.60)$$

进而可以通过公式 (2.57) 求解相变点, 该过程如图 2.10(b) 所示。J. Gao 等人^[139] 进一步提出, 网络系统弹性取决于其网络结构。将公式 (2.59) 转换为:

$$\beta_{\text{eff}} = \langle k \rangle + \mathcal{S} \mathcal{H}, \quad (2.61)$$

由于网络中出度之和等于入度之和, 这里令 $\langle k^{\text{out}} \rangle = \frac{1}{N} \sum_{n=1}^N k_n^{\text{out}} = \langle k \rangle = \frac{1}{N} \sum_{n=1}^N k_n^{\text{in}} = \langle k^{\text{in}} \rangle$, $\langle k \rangle$ 是平均度, 决定了网络连边的稀疏程度。于是, 相关性系数 $\mathcal{S} = \langle k^{\text{in}} k^{\text{out}} \rangle - \langle k^{\text{in}} \rangle \langle k^{\text{out}} \rangle / \sigma^{\text{in}} \sigma^{\text{out}}$ 。这里 $\langle k^{\text{in}} k^{\text{out}} \rangle = \frac{1}{N} \sum_{n=1}^N k_n^{\text{out}} k_n^{\text{in}}$, σ^{in} 和 σ^{out} 分别是各节点入度分布和出度分布的标准差。 \mathcal{S} 描述了两个节点之间互连视角下的相关性, 其趋近于一, 则两个节点倾向于互联, 否则越趋近于负一, 则越倾向于单向连接。因此易知无向图中 $\mathcal{S} = 1$ 。此外, $\mathcal{H} = \sigma^{\text{in}} \sigma^{\text{out}} / \langle k \rangle$, 描述了节点度的同质化程度, 当 $\langle k \rangle$ 固定, \mathcal{H} 越大, 网络节点的度分布方差越小, 网络节点越具有相同的度, 则网络连边出现倾向于规则, 反之则倾向于随机。由于 β_{eff}^c 可以由公式 (2.57) 和公式 (2.60) 求出, 因而可以通过提升 $\langle k \rangle$ 、 \mathcal{S} 和 \mathcal{H} 等从而使得 $\beta_{\text{eff}} > \beta_{\text{eff}}^c$ 以避免系统崩溃。

当然也有不同观点, 比如 C. Tu 等人^[140] 提出了一个更加泛化的弹性边界条件, 其认为 $N \gg \frac{\sigma}{|\mu|}$ 时, 网络系统是具有弹性的, 这里 μ 和 σ 分别是网络连边出现概率分布的均值和标准差。J. Jiang 等人^[141] 提出了一个将系统降低至二维以分析弹性边界条件的方法, 不过其降维方法仍然基于公式 (2.58) 和公式 (2.59)。E. Laurence 等人^[142] 认为, 的确对于部分网络系统降低至二维更好, 但具体降低至几维应当取决于网络的特征值。其首先论证文献^[139] 中降维方法等效于对邻接矩阵的基于谱分解的降维, 也即:

$$x_{\text{eff}} = \frac{\sum_{n=1}^N k_n^{\text{out}} x_n}{\sum_{n=1}^N k_n^{\text{out}}} = \frac{\mathbf{v}_D^\top \mathbf{x}}{\mathbf{1}^\top \mathbf{v}_D}, \quad (2.62)$$

如果满足 $\frac{\langle (k^{\text{out}})^2 \rangle}{\langle k^{\text{out}} \rangle} > \sqrt{\max(k_i^{\text{out}})} \ln N$, 这里 $i \in \{1, \dots, N\}$, 且条件容易满足。此外, \mathbf{v}_D 是邻接矩阵最大特征值对应的特征矢量。因此, 降低至几维取决于最大特征值, 当邻接矩阵特征值为:

$$|\lambda_1| \approx |\lambda_2| \approx \dots \approx |\lambda_l| \gg |\lambda_{l+1}| \geq \dots \geq |\lambda_N|, \quad (2.63)$$

可知应当降低至 l 维。因此, 可以认为复杂网络弹性蕴含于更基本的性质, 也即复杂网络的低秩性之中^[143-145]。不过, 复杂网络邻接矩阵的低秩分解在数学上已经有许多成熟的算法, 但如何针对大规模的矩阵进行快速的低秩分解仍是需要研究的重点。以及如 J. Gao^[143] 所言, 通过低秩分解视角揭开复杂网络动力学在高阶网络上的涌现规律同样值得研究。

2.3 复杂网络动力学系统辨识

2.3.1 ASIND 算法原理

尽管已经可以利用前文所述的一些理论工具分析复杂网络系统, 但这一切的前提是有一个可以研究的系统模型, 因此首先需要确定一个真实世界的系统是复杂网络系统, 然后能够准确地辨识它。所以需要回到一开始的系统辨识问题, 如何才能利用观测数据构建一个复杂网络系统并辨识它? 针对这个问题, T. Gao 等人^[36] 提出了一种两相推断法 (Two-Phase Inference, TPI)。简言之, 对于公式 (2.43), 其假设邻接矩阵 A 已知, 而自驱动函数 $F(\cdot)$ 和交互函数 $G(\cdot, \cdot)$ 未知, 然后分别对这两个函数构建基函数矩阵, 从而将其转换成“Lasso 问题”(见公式 (2.8)) 并求解。与之相对的, B. Prasse 等人^[37] 提出了一种两步预测法 (Two-Step Prediction, TSP)。该方法假设邻接矩阵 A 未知, 但是自驱动函数 $F(\cdot)$ 和交互函数 $G(\cdot, \cdot)$ 已知, 从而同样将其转换成“Lasso 问题”求解。由此可见, TPI 算法和 TSP 算法所解决的问题是对偶的。此外, I. Topal 和 D. Eroglu^[110] 同样假设 F 和 G 已知, 从而利用 SINDy 算法 (见本文章节 2.1.2) 求解 A 。该算法在 Rulkov Map 动力学 (见表 2.2) 上进行测试, 从而可以解决高维的复杂网络动力学辨识。值得注意的是, 这里“高维”是指一个网络节点用多个属性变量进行描述。TPI 算法同样声称可以解决高维问题, 但其与文献^[110] 不同的是, TPI 算法隐含地假设了不同维度之间相互独立, 从而针对每个维度单独进行辨识, 而文献^[110] 则没有假设其独立, 从而解决了维度之间存在耦合的问题。J. Koch 等人^[146] 提出了针对复杂网络动力学的神经微分方程算法 (见本文章节 2.1.4)。其更进一步地解决 F 、 G 和 A 均未知情况下的辨识问题, 但也正如前文所述, 该算法由于使用了神经网络模型建模 F 和 G , 因而可解释性不高。

因此, 为了实现可解释的复杂网络动力学辨识问题, 本文提出了一种交替稀疏网络动力学辨识算法 (Alternating Sparse Identification of Network Dynamics, ASIND)。该算法利用稀疏辨识技术, 将 F 、 G 、 A 均视为未知参数, 从而将其转换成“Lasso 问题”求解。因此, ASIND 算法相比以往算法的优势在于: (1) 对知识的需要更少; (2) 可解释性更高。首先, 假设给定观测数据 \mathbf{x} , 同时通过差分法得到其导数 $\dot{\mathbf{x}}$, 参考公式 (2.1)。从而对于节点 $i = 1, \dots, N$, 定义复杂网络动力学辨识的“Lasso 问题”, 如下:

$$\begin{aligned} & \min_{w, A} \sum_{i=1}^N \sum_{m=1}^{M_1+M_2} \|w_{im}\|_1 + \sum_{i=1}^N \sum_{j=1}^N \|A_{ij}\|_1 \\ & \text{s.t. } \dot{\mathbf{x}}_i = \sum_{m=1}^{M_1} w_{im} F_m(\mathbf{x}_i) + \sum_{m=1}^{M_2} w_{im} \sum_{j=1}^N A_{ij} G_m(\mathbf{x}_i, \mathbf{x}_j), \\ & \quad A_{ij} \geq 0, i = 1, \dots, N, j = 1, \dots, N. \end{aligned} \quad (2.64)$$

这里 $F_m(\cdot), m = 1, \dots, M_1$ 和 $G_m(\cdot, \cdot), m = 1, \dots, M_2$, 分别是基函数, 用于重构 F 和 G , 而 $w_{im}, m = 1, \dots, M_1 + M_2$ 则是对应的基函数系数。进而通过增广拉格朗日方法 (见公式 (2.12)), 将其转换为:

$$\begin{aligned} & \min_{\mathbf{w}, A, \boldsymbol{\lambda}} \mathcal{L}(\mathbf{w}, A, \boldsymbol{\lambda}) \\ & \text{s.t. } A_{ij} \geq 0, i = 1, \dots, N, j = 1, \dots, N. \end{aligned} \quad (2.65)$$

其中,

$$\begin{aligned}\mathcal{L}(\mathbf{w}, A, \boldsymbol{\lambda}) = & \sum_{i=1}^N \sum_{m=1}^{M_1+M_2} \|w_{im}\|_1 + \sum_{i=1}^N \sum_{j=1}^N \|A_{ij}\|_1 \\ & + \sum_{i=1}^N \lambda_i \left(\dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im} \sum_{j=1}^N A_{ij} G_m(\mathbf{x}_i, \mathbf{x}_j) \right) \\ & + \frac{\rho}{2} \sum_{i=1}^N \left\| \dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im} \sum_{j=1}^N A_{ij} G_m(\mathbf{x}_i, \mathbf{x}_j) \right\|_2^2.\end{aligned}\quad (2.66)$$

这里 $\boldsymbol{\lambda} \in \mathbb{R}^N$ 是拉格朗日乘子, 与节点数量对应。 $\rho \in \mathbb{R}$ 是惩罚因子。从而公式 (2.65) 可以依如下步骤交替求解。首先更新 A ,

$$\begin{aligned}A^{(k+1)} &= \arg \min_A \mathcal{L}(\mathbf{w}^{(k)}, A, \boldsymbol{\lambda}^{(k)}), \\ \text{s.t. } A_{ij} &\geq 0, i = 1, \dots, N, j = 1, \dots, N.\end{aligned}\quad (2.67)$$

然后更新 \mathbf{w} ,

$$\mathbf{w}^{(k+1)} = \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, A^{(k+1)}, \boldsymbol{\lambda}^{(k)}). \quad (2.68)$$

然后更新 $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \alpha \times \partial \mathcal{L}(\mathbf{w}^{(k+1)}, A^{(k+1)}, \boldsymbol{\lambda}) / \partial \boldsymbol{\lambda}, \quad (2.69)$$

这里 $\alpha \in \mathbb{R}$ 用于控制 $\boldsymbol{\lambda}$ 更新的步长。

首先求解公式 (2.67), 其可以独立地对每个节点求解 A_i , A_i 是 A 的第 i 行, 从而 $A_i^{(k+1)} = [A_{i1}^{(k+1)}, \dots, A_{iN}^{(k+1)}]$, $i = 1, \dots, N$ 。于是得到:

$$\begin{aligned}& \arg \min_{A_i} \mathcal{L}(\mathbf{w}^{(k)}, A_i, \boldsymbol{\lambda}^{(k)}) \\ &= \arg \min_{A_i} \sum_{j=1}^N A_{ij} + \lambda_i^{(k)} \left(\dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im}^{(k)} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im}^{(k)} \sum_{j=1}^N A_{ij} G_m(\mathbf{x}_i, \mathbf{x}_j) \right) \\ & \quad + \frac{\rho}{2} \left\| \dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im}^{(k)} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im}^{(k)} \sum_{j=1}^N A_{ij} G_m(\mathbf{x}_i, \mathbf{x}_j) \right\|_2^2 \\ \text{s.t. } A_{ij} &\geq 0, j = 1, \dots, N.\end{aligned}\quad (2.70)$$

然后求解公式 (2.68), 可以对每个节点独立求解 $\mathbf{w}_i^{(k+1)} = [w_{i1}^{(k+1)}, \dots, w_{i(M_1+M_2)}^{(k+1)}]$, $i = 1, \dots, N$, 如下:

$$\begin{aligned}& \arg \min_{\mathbf{w}_i} \mathcal{L}(\mathbf{w}_i, A^{(k+1)}, \boldsymbol{\lambda}^{(k)}), \\ &= \arg \min_{\mathbf{w}_i} \sum_{m=1}^{M_1+M_2} \|w_{im}\|_1 + \lambda_i^{(k)} \left(\dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im} \sum_{j=1}^N A_{ij}^{(k+1)} G_m(\mathbf{x}_i, \mathbf{x}_j) \right) \\ & \quad + \frac{\rho}{2} \left\| \dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im} \sum_{j=1}^N A_{ij}^{(k+1)} G_m(\mathbf{x}_i, \mathbf{x}_j) \right\|_2^2.\end{aligned}\quad (2.71)$$

然后参考公式 (2.10) 将该问题转换为二次规划问题。也即, 令 $\mathbf{u}_i = [u_{i1}, \dots, u_{i(M_1+M_2)}]$, 且 $u_{im} = \max(0, w_{im})$, $m = 1, \dots, M_1 + M_2$ 。同样地, 令 $\mathbf{v}_i = [v_{i1}, \dots, v_{i(M_1+M_2)}]$, 且 $-v_{im} = \min(0, w_{im})$, $m = 1, \dots, M_1 + M_2$ 。因此, $u_{im} \geq 0, v_{im} \geq 0$, 且 $w_{im} = \max(0, w_{im}) + \min(0, w_{im}) = u_{im} - v_{im}$ 。于是重写公

式(2.71)得到:

$$\begin{aligned}
 &= \arg \min_{\mathbf{u}_i, \mathbf{v}_i} \sum_{m=1}^{M_1+M_2} u_{im} + \sum_{m=1}^{M_1+M_2} v_{im} \\
 &\quad + \lambda_i^{(k)} \left[\dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} (u_{im} - v_{im}) F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} (u_{im} - v_{im}) \sum_{j=1}^N A_{ij}^{(k+1)} G_m(\mathbf{x}_i, \mathbf{x}_j) \right] \\
 &\quad + \frac{\rho}{2} \left\| \dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} (u_{im} - v_{im}) F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} (u_{im} - v_{im}) \sum_{j=1}^N A_{ij}^{(k+1)} G_m(\mathbf{x}_i, \mathbf{x}_j) \right\|_2^2, \\
 \text{s.t. } & u_{im} \geq 0, v_{im} \geq 0, m = 1, \dots, M_1 + M_2.
 \end{aligned} \tag{2.72}$$

容易发现, 公式(2.70)和公式(2.72)具有相同的二次规划问题形式, 因此可以通过同一种二次规划求解器求解, 本文倾向于使用 Gurobi 软件^[147]求解, 不过换别的求解器并不影响本文理论的一般性。

最后求解公式(2.69), 于是得到:

$$\partial \mathcal{L}(\mathbf{w}^{(k+1)}, A^{(k+1)}, \boldsymbol{\lambda}) / \partial \boldsymbol{\lambda} = \dot{\mathbf{x}}_i - \sum_{m=1}^{M_1} w_{im}^{(k+1)} F_m(\mathbf{x}_i) - \sum_{m=1}^{M_2} w_{im}^{(k+1)} \sum_{j=1}^N A_{ij}^{(k+1)} G_m(\mathbf{x}_i, \mathbf{x}_j). \tag{2.73}$$

从而可以利用其替换公式(2.69)中的对应项。由于公式(2.65)是凸问题, 且公式(2.70)和公式(2.72)也是凸问题, 因而易知其有解且唯一, 具体分析见本文章节 2.1.1。

2.3.2 实验设置

为了验证 ASIND 算法有效性, 本文设计预测实验以测试 ASIND 算法的数据外推能力(参考定义 1)。简言之, 给定时序观测轨迹 $\mathbf{x} \in \mathbb{R}^{T \times N}$, 在其上辨识网络动力学, 然后向后继续预测 T' 步, 从而比较预测的 T' 步轨迹与真实轨迹之间的差异。本文这里 $T = 900$, $T' = 100$, 也即向后预测 100 步。

本文采用仿真数据进行测试, 为此首先需要构造一个形如公式(2.43)的网络动力学系统, 其主要包含三个部分, 也即交互网络结构 A 、自驱动函数 F 和交互函数 G 。针对交互网络, 本文采用了三种网络结构, 分别是 ER 随机图、小世界网络和无标度网络, 其原理可参考本文章节 2.2.1, 参数设置如下:

1. ER 随机图: 连边出现概率 $p = 0.1$ 。
2. 小世界网络: 平均度 $\langle k \rangle = 4$, 重连概率 $p = 0.1$ 。
3. 无标度网络: $\alpha = 0.41$ 控制新节点连接到现有节点的概率, 越大表示新节点更倾向于连接到具有高出度的节点。 $\beta = 0.54$ 控制现有节点的入度对新节点连接的影响, 越大表示新节点更倾向于连接到已有的高入度节点。随机重连概率 $\gamma = 0.05$, 越大表示连接的随机性越强。

全部共 100 个网络节点, 也即 100 个变量, 因此 $N = 100$ 。此外, 本文选择了四种自驱动函数 F 和交互函数 G 配置, 从而构成四种常见的网络动力学, 分别是 Kuramoto 模型^[121, 122]、Susceptible-Infected-Susceptible (SIS) 模型^[148, 149]、Lotka-Volterra (LV) 模型^[150] 和 Michaelis-Menten (MM) 模型^[139, 151], 其参数设置如表 2.3 所示。

此外, 为了测试 ASIND 算法的预测性能, 本文引入了平均绝对误差 (Mean Absolute Error, MAE)、均方根误差 (Root Mean Squared Error, RMSE) 和平均绝对百分比误差 (Mean Absolute Percentage Error, MAPE)

表 2.3: 网络动力学特例

网络动力学	$F(\mathbf{x}_i)$	$G(\mathbf{x}_i, \mathbf{x}_j)$	符号解释
Kuramoto 模型	ω_i	$\frac{c}{N} \sin(\mathbf{x}_j - \mathbf{x}_i)$	Kuramoto 模型描述了 N 个相互耦合的振子通过相位耦合实现同步的过程。在该模型中，每个振子具有自身的自然频率，耦合项促进了振子之间的相互影响，从而导致集群行为。因此这里 ω_i 是振子 i 的自然频率， $c > 0$ 表示耦合系数。
SIS 模型	$-\delta_i \mathbf{x}_i$	$\gamma_i(1 - \mathbf{x}_i)\mathbf{x}_j$	SIS 模型描述了 N 个个体在传染病传播过程中，在易感和感染状态之间的循环。感染的个体在一定时间后恢复为易感状态，从而可以再次被感染，该模型被用于研究传染病在群体中的动态传播特性，因此这里 $\delta_i > 0$ 表示治愈率， $\gamma_i \geq 0$ 表示感染率。
LV 模型	$\mathbf{x}_i(\alpha_i - \theta_i \mathbf{x}_i)$	$-\gamma_i \mathbf{x}_i \mathbf{x}_j$	LV 模型描述了两个或多个物种在同一生态系统中竞争资源的动态变化，其中物种的增长受到自身种群密度和其他竞争物种密度的影响，从而揭示了物种共存和竞争排斥等生态现象。因此这里增长率参数 $\alpha_i > 0, \theta_i > 0$ ，此外， γ_i 是竞争率参数。
MM 模型	$-\mathbf{x}_i$	$\mathbf{x}_j^h (1 + \mathbf{x}_j^h)^{-1}$	MM 模型用于描述生物网络中酶和底物之间的相互作用，帮助理解代谢过程和信号传导。通过模拟这些非线性反应，可以揭示生物网络系统中的稳态和响应特性。因此这里参数 h 表示希尔系数 (Hill coefficient)，用于描述酶与其底物之间的结合强度。

等指标，定义如下：

$$\begin{aligned}
 \text{MAE} &= \frac{1}{N \times T'} \sum_{i=1}^N \sum_{t=1}^{T'} |x_i(t) - \hat{x}_i(t)|, \\
 \text{RMSE} &= \sqrt{\frac{1}{N \times T'} \sum_{i=1}^N \sum_{t=1}^{T'} (x_i(t) - \hat{x}_i(t))^2}, \\
 \text{MAPE} &= \frac{1}{N \times T'} \sum_{i=1}^N \sum_{t=1}^{T'} \left| \frac{x_i(t) - \hat{x}_i(t)}{x_i(t)} \right| \times 100\%,
 \end{aligned} \tag{2.74}$$

这里 $\hat{x}_i(t)$ 是观测场景数据 $x_i(t)$ 的预测值。一共 N 个节点，向后预测 T' 步。

2.3.3 实验结果

预测实验结果如表 2.4 所示，可以看出，(1) ASIND 算法在多个网络动力学模型上辨识准确率较高，从而可以预测 100 步之多。(2) 横向对比不同的网络结构，同等模型参数条件下，基于 ER 随机图和无标度网络的网络动力学能够实现准确的预测，而基于小世界网络的网络动力学则较难预测。(3) 纵向对比不同的网络动力学模型，Kuramoto 模型相对较难准确预测，而其它三类网络动力学模型则能够准确预测。

此外，本文尝试比较真实的网络结构 A 和 ASIND 算法重构的网络结构 \hat{A} 之间的差异，举例基于 ER 随机图的 Kuramoto 模型，结果如图 2.11 所示，可以直观地发现，两个 ER 随机图并不相似，其交并比只有 16.1%，但是 ASIND 算法的重构却是比较准确的。因此，自驱动函数 F 和交互函数 G 相同的情况下，交互网络结构不同却可能采样出相似的系统轨迹，于是这些导致相同的系统轨迹的交互网络结构之间是

否存在等价性，这仍然是一个值得探索的问题。

表 2.4: ASIND 算法预测性能

网络动力学	ER 随机图			小世界网络			无标度网络		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Kuramoto 模型	0.0005	0.0007	0.03%	0.0121	0.0234	0.57%	0.0087	0.0235	0.20%
SIS 模型	0.0002	0.0003	0.04%	0.0006	0.0010	0.32%	0.0005	0.0011	0.40%
LV 模型	0.0002	0.0005	0.08%	0.0003	0.0006	0.11%	0.0005	0.0010	0.09%
MM 模型	0.0004	0.0007	0.17%	0.0004	0.0009	0.29%	0.0003	0.0008	0.21%

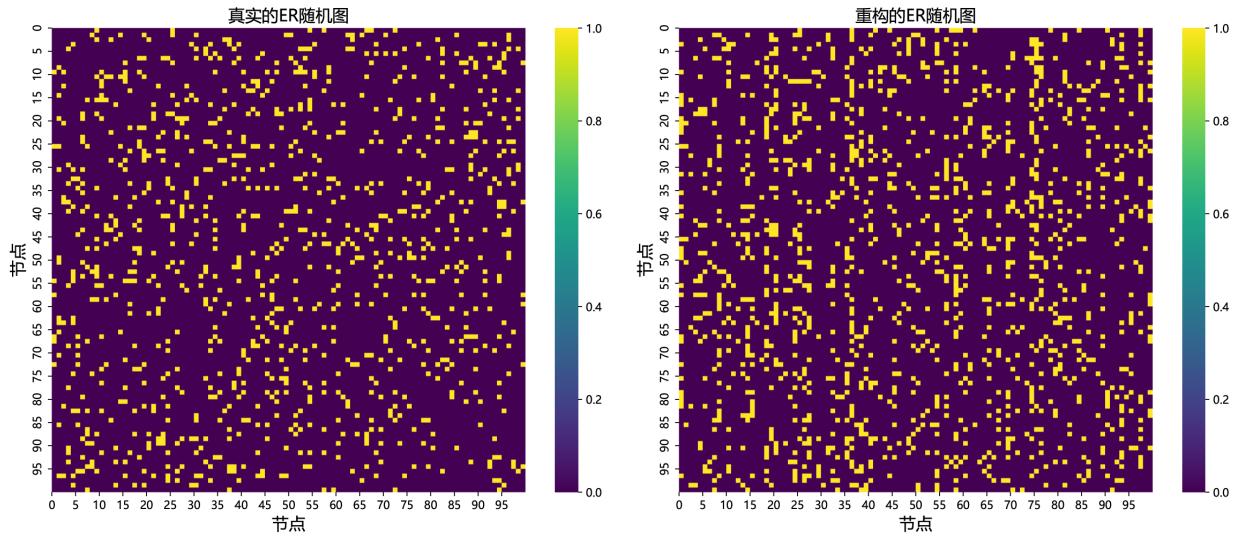


图 2.11: 真实的 ER 随机图和重构的 ER 随机图对比，此二图邻接矩阵的交并比为 16.1% ($\frac{|A \cap \hat{A}|}{|A \cup \hat{A}|} \times 100\%$)。

第三章 基于生成模型的场景数据生成

真实世界的观测数据中往往同时存在着确定的部分和不确定的部分。基于系统辨识的场景数据生成方法需要对真实世界先构建系统模型，然后才能辨识系统模型。即便是通过人工神经网络建模，其也只是在构建系统模型时需要的知识更少或没有，但其构建系统模型的前置需要不会改变。因此，如果称基于系统辨识的场景数据生成方法是对真实世界确定性的建模，那么相反地，基于生成模型的场景数据生成方法则是对真实世界的不确定性进行建模，因而其核心是利用概率描述真实世界中的各种不确定性，或者说各种可能出现的状况。也即，已知场景数据 x ，利用外部知识 g' ，构建生成器 g ，实现生成场景数据 y ，并且约束 $P(x) = P(y)$ ，如图 1.2 所示。因此从这里开始，本文用 x 或 y 表示场景数据或一个场景样本，不限制样本表示的具体结构，本文后续将按照这一思路进行论述。

3.1 生成模型

3.1.1 双样本问题

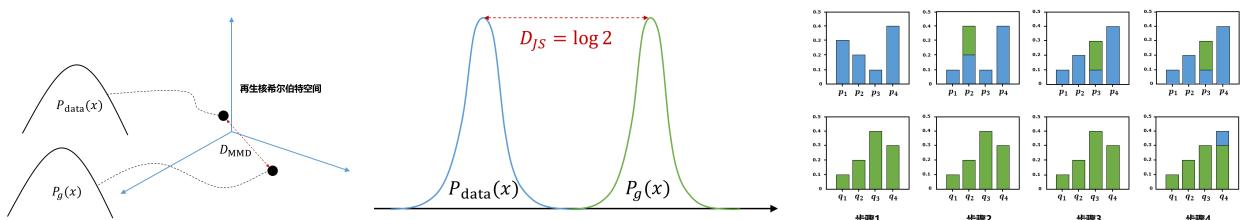
由于约束 $P(x) = P(y)$ ，因此生成模型将该问题归约为一个双样本问题^[49,50]，也即令 $P_{\text{data}}(x)$ 为观测样本的数据分布，令 $P_g(x)$ 为生成器生成样本的数据分布，进而检测两组样本数据是否服从相同的数据分布。从而定义问题：

$$\min_g D(P_{\text{data}}(x) \parallel P_g(x)). \quad (3.1)$$

这里 $D(\cdot \parallel \cdot)$ 是概率距离度量。文献^[49,50]中定义最大均值差异 (Maximum Mean Discrepancy, MMD)，如下：

$$D_{\text{MMD}}(P_{\text{data}}(x) \parallel P_g(x)) = \mathbb{E}_{x,y \sim P_{\text{data}}} [K(x,y)] + \mathbb{E}_{x,y \sim P_g} [K(x,y)] - 2\mathbb{E}_{x \sim P_{\text{data}}, y \sim P_g} [K(x,y)]. \quad (3.2)$$

其中 $K(\cdot, \cdot)$ 是再生核函数^[152]，简言之，其会将样本 x 和 y 映射到再生核希尔伯特空间后求内积，由此避免计算数据分布，从而直接计算两个分布在再生核希尔伯特空间中的距离，如图 3.1(a) 所示。MMD 距离满足 $D_{\text{MMD}} = 0$ 当且仅当 $P_{\text{data}}(x) = P_g(x)$ 。



(a) D_{MMD} 示意图（出自文献^[153]中 (b) D_{JS} 示意图。当两个分布不交叠，则 (c) D_{WD} 示意图。这里 $D_{\text{WD}}(p||q) = |p_1 - q_1| \times 2 + |p_4 - q_4| = 0.5$ 。Figure 3)。
 $D_{\text{JS}} \equiv \log 2$ 。

图 3.1: 概率距离度量示意图

还可以定义 Kullback–Leibler (KL) 散度^[154]，如下：

$$D_{\text{KL}}(P_{\text{data}}(x) \parallel P_g(x)) = \int P_{\text{data}}(x) \log \frac{P_{\text{data}}(x)}{P_g(x)} dx. \quad (3.3)$$

KL 散度满足 $D_{\text{KL}} = 0$ 当且仅当 $P_{\text{data}}(x) = P_g(x)$ 。从而得到

$$\begin{aligned} \min_g D_{\text{KL}}(P_{\text{data}}(x) \| P_g(x)) &= \min_g \int P_{\text{data}}(x) \log \frac{P_{\text{data}}(x)}{P_g(x)} dx \\ &= \max_g \int P_{\text{data}}(x) \log \frac{P_g(x)}{P_{\text{data}}(x)} dx + \int P_{\text{data}}(x) \log P_{\text{data}}(x) dx \\ &= \max_g \int P_{\text{data}}(x) \log \left[P_g(x) \times \frac{P_{\text{data}}(x)}{P_{\text{data}}(x)} \right] dx \\ &= \max_g \mathbb{E}_{P_{\text{data}}(x)} [\log P_g(x)]. \end{aligned} \quad (3.4)$$

注意这里 $\int P_{\text{data}}(x) \log P_{\text{data}}(x) dx$ 略去，是因为 $P_{\text{data}}(x)$ 是已知的观测数据分布，所以可以视为常数。更重要的是，这里 $P_g(x)$ 也称为似然概率，当调整 g 从而最大化似然概率，则 $P_g(x) = P_{\text{data}}(x)$ 。

此外，KL 散度是不对称的，也即 $D_{\text{KL}}(P_{\text{data}}(x) \| P_g(x)) \neq D_{\text{KL}}(P_g(x) \| P_{\text{data}}(x))$ 。但是利用两个 KL 散度的和可以得到对称的 Jensen-Shannon (JS) 散度^[155]，定义如下：

$$D_{\text{JS}}(P_{\text{data}}(x) \| P_g(x)) = \frac{1}{2} D_{\text{KL}}\left(P_{\text{data}}(x) \| \frac{P_{\text{data}}(x) + P_g(x)}{2}\right) + \frac{1}{2} D_{\text{KL}}\left(P_g(x) \| \frac{P_{\text{data}}(x) + P_g(x)}{2}\right) \quad (3.5)$$

$$= \int \left[\frac{1}{2} P_{\text{data}}(x) \log \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)} + \frac{1}{2} P_g(x) \log \frac{P_g(x)}{P_{\text{data}}(x) + P_g(x)} \right] dx + \log 2. \quad (3.6)$$

可知 $D_{\text{JS}} = 0$ 当且仅当 $P_{\text{data}}(x) = P_g(x)$ ，因为只有此种情况下，两个 D_{KL} 均为零。尽管如此，但如果将 D_{JS} 代入公式 (3.1)，则实际上很难求解，尤其是在高维情况下（指样本 x 用高维特征矢量表示）。因为在高维情况下， $P_{\text{data}}(x)$ 和 $P_g(x)$ 容易没有交叠，因而无论如何调整 g ，公式 (3.6) 的积分内部始终为零，从而 $D_{\text{JS}} \equiv \log 2$ ，也就失去了度量距离的能力，如图 3.1(b) 所示。

因此，还可以采用 Wasserstein Distance (WD) 距离度量^[156, 157]，其一阶形式定义如下：

$$D_{\text{WD}}(P_g(x) \| P_{\text{data}}(x)) = \inf_{\pi \in \Pi(P_{\text{data}}, P_g)} \mathbb{E}_{(x,y) \sim \pi} [\|x - y\|]. \quad (3.7)$$

这里 $\Pi(P_{\text{data}}, P_g)$ 是二者的联合分布，描述对所有 x ，“搬运” $P_{\text{data}}(x)$ 到 $P_g(x)$ 所需要步骤，因此 WD 距离描述了这个过程的最小计算开销，如图 3.1(c) 所示。可知 $D_{\text{WD}} = 0$ 当且仅当 $P_{\text{data}}(x) = P_g(x)$ ，因为只有当两分布相等时无需任何“搬运”。本文列出这几种概率距离度量的对比，见表 3.1。

表 3.1: 概率距离度量对比

概率距离度量	优点	缺点
KL 散度	计算简单，有坚实的信息论理论支持。	非对称，因而实际上不能视作距离度量。
JS 散度	具有对称性，可以作为距离度量。	高维情况下两分布容易不交叠，从而使 $D_{\text{JS}} \equiv \log 2$ ，会失去度量能力。
WD 距离	具有对称性，可以作为距离度量。由其定义决定了，即便在高维情况下，其功能仍然良好。	由其定义决定了其不易求梯度。如若需要求梯度，则往往需要在满足一些约束条件，才能完成极其复杂的形式转换。
MMD 距离	具有对称性，可以作为距离度量。高维情况下仍然功能良好。	核函数决定了 MMD 距离的形式，因而核函数会影响其计算结果。经验上来说，不合理的核函数选择可能会使得 MMD 距离值整体上过大或过小，过大则会使得公式 (3.1) 在求解梯度时震荡，过小则梯度消失。因而 MMD 距离的技巧性较强。

3.1.2 变分自编码器

D. P. Kingma 和 M. Welling^[39] 提出了变分自编码器 (Variational Autoencoder, VAE) 模型用于场景数据生成。如图 3.2 所示, VAE 模型构造了“编码器 - 解码器”结构, 并利用神经网络建模编码器 f 和解码器 g , 从而对样本 x 依据 $x \rightarrow z \rightarrow x$ 顺序计算。当特征 z 已知, 则通过采样不同的 z 实现对 x 的生成, 因此解码器 g 就是生成器。

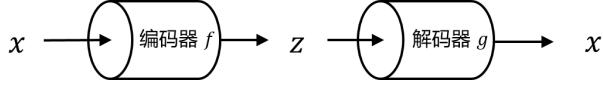


图 3.2: 变分自编码器示意图

因此这个问题的第一个关键之处在于, VAE 首先将观测数据 $x \sim P_{\text{data}}(x)$ 通过编码器 f 进行编码从而得到 $z \sim P_f(z|x)$, 再通过解码器 g 进行解码从而得到生成的场景数据 $x \sim P_g(x|z)$, 并要求 $P_{\text{data}}(x) = P_g(x)$ 。对于 f 和 g , 其所面向的是同一个 z , 因此对齐 z 得到

$$\begin{aligned}
 D_{\text{KL}}(P_f(z|x) || P_g(z|x)) &= \int P_f(z|x) \log \frac{P_f(z|x)}{P_g(z|x)} dz \\
 &= \int P_f(z|x) \log \frac{P_f(z|x)P_g(x)}{P_g(x|z)P_g(z)} dz \\
 &= D_{\text{KL}}(P_f(z|x) || P_g(z)) - \mathbb{E}_{P_f(z|x)} [\log P_g(x|z)] + \log P_g(x) \\
 &\geq 0.
 \end{aligned} \tag{3.8}$$

从而整理得到

$$\log P_g(x) \geq \mathcal{L}(x, f, g) = -D_{\text{KL}}(P_f(z|x) || P_g(z)) + \mathbb{E}_{P_f(z|x)} [\log P_g(x|z)]. \tag{3.9}$$

从而可以代入公式 (3.4) 通过最大似然估计求解。这个转换的巧妙之处在于将一个复杂的问题转换成一个相对简单的问题, 因为 VAE 模型的计算顺序是 $x \rightarrow z \rightarrow x$, 所以求解 $P_g(z|x)$ 是困难的, 但公式 (3.9) 中的各个元素则易于求解。由于公式 (3.9) 的 $\log P_g(x)$ 始终大于或等于 $\mathcal{L}(x, f, g)$, 因而可以通过最大化 $\mathcal{L}(x, f, g)$ 以最大化 $\log P_g(x)$, 因此 $\mathcal{L}(x, f, g)$ 又称为证据下界 (Evidence Lower Bound, ELBO)。

这个问题的第二个关键之处在于对特征 z 的重参数化。要生成丰富多样的场景 $x = g(z)$, 则要求输入的 z 也是丰富多样的。因而不失一般性, 假设特征 $z \in \mathbb{R}^M$ 服从高斯分布, 则 $P_g(z) \sim N(\mathbf{0}, \mathbf{I})$, 而 $P_f(z|x) \sim N(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$, 这里 $\boldsymbol{\mu}, \boldsymbol{\sigma}^2 \in \mathbb{R}^M$ 分别是均值和方差, $\text{diag}(\cdot)$ 是对角化函数。如公式 (3.8) 所言, 需要对齐两个正态分布。然而, 由于编码器 f 和解码器 g 都是人工神经网络, 其参数都需要用反向传播算法更新 (见公式 (2.25)), 而正态分布没法直接求梯度, 因此引出重参数技巧。令 $z \sim P_f(z|x)$ 且 $z = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, 这里 $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I})$ 。相当于根据参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 对标准正太分布进行缩放, 从而将 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$ 重参数化, 并通过反向传播算法求解梯度并优化。代入公式 (3.9) 得到新的证据下界, 如下

$$\mathcal{L}(x, f, g) = \frac{1}{2} \sum_{m=1}^M (1 + \log \boldsymbol{\sigma}_m^2 - \boldsymbol{\mu}_m^2 - \boldsymbol{\sigma}_m^2) + \mathbb{E}_{P_f(z|x)} [\log P_g(x|z)] \tag{3.10}$$

这里 $\boldsymbol{\mu}_m$ 和 $\boldsymbol{\sigma}_m^2$ 是参数矢量的第 m 个参数, 具体推导过程可见文献^[39] 的附录 B。因此在执行过程中, 编码器 f 实际上输出的是参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}^2$, 再从 $N(\mathbf{0}, \mathbf{I})$ 中采样 $\boldsymbol{\epsilon}$ 得到 z , 最后输入解码器 g 生成场景。

基于 VAE 模型, D. P. Kingma 等人^[158] 提出 Conditional VAE (CVAE) 模型, 该模型在 VAE 的基础上引入了条件标签用作条件生成, 使得 CVAE 可以根据需要生成多模态的场景数据。这里条件标签实际上也是一种场景数据, 收集于对真实世界的观测, 因此这里同样说明场景数据就是 (计算机) 人工智能对真实世界的各种描述, 体现了生成模型的灵活性。此外, T. N. Kipf 等人^[159] 提出变分图自编码器 (Variational Graph Autoencoder, VGAE) 模型, 该模型结合了图卷积神经网络 (Graph Convolution Network, GCN) 模型^[160] 和 VAE 模型, 从而使得解码器 g 可以被用来生成图结构数据。本文后续还会再次分类讨论这些变种模型。

3.1.3 生成对抗网络

I. Goodfellow 等人^[40]提出了生成对抗网络 (Generative Adversarial Network, GAN) 用于场景数据生成。GAN 模型构造了“生成器 - 判别器”结构，并利用神经网络 g 和 f 分别建模生成器和判别器，如图 3.3 所示。首先采样特征 z ，输入生成器 g 从而生成样本 x ，将生成样本和观测样本一同输入判别器 f ，判别器会输出标签 y ，如果 $y = 1$ 则表示判别为观测样本，如果 $y = 0$ 则表示判别为生成样本。因此在这个过程中，判别器试图分辨生成样本和观测样本，而生成器则试图生成样本以“欺骗”判别器，从而构成 min-max 博弈，如下：

$$\min_g \max_f \mathbb{E}_{x \sim P_{\text{data}}(x)} [\log f(x)] + \mathbb{E}_{x \sim P_g(x)} [\log(1 - f(x))]. \quad (3.11)$$

注意这里 $P_g(x) = P(g(z))$ ，通过向 g 输入不同的 z 从而生成不同的场景样本 x ，一般假设 $z \sim N(\mathbf{0}, \mathbf{I})$ 。

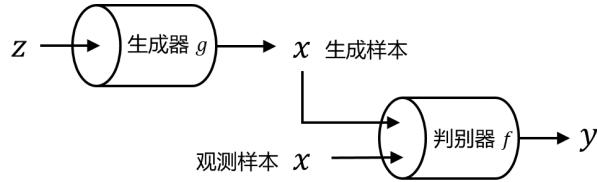


图 3.3: 生成对抗网络示意图

为了说明这个问题，本文从生成器和判别器的视角分别构造两个分布，如下：

$$P(x, y) = \begin{cases} P_{\text{data}}(x)P_1, & \text{if } y = 1, \\ P_g(x)P_0, & \text{if } y = 0. \end{cases} \quad Q(x, y) = \begin{cases} P_{\text{data}}(x)Q_f(y = 1|x), & \text{if } y = 1, \\ P_g(x)Q_f(y = 0|x), & \text{if } y = 0. \end{cases} \quad (3.12)$$

这里假设 $P_0 = P_1 = \frac{1}{2}$ ，表示生成器生成了同观测样本一样多的样本，因此各占一半。 $Q_f(y = 1|x) + Q_f(y = 0|x) = 1$ ，分别表示判别器判断一个样本是观测样本和生成样本的概率。由此得到 $P(x, y)$ 和 $Q(x, y)$ 之间的差异，如下：

$$\begin{aligned} D_{\text{KL}}(P(x, y) || Q(x, y)) &= \int P(x, y = 1) \log \frac{P(x, y = 1)}{Q(x, y = 1)} dx + \int P(x, y = 0) \log \frac{P(x, y = 0)}{Q(x, y = 0)} dx \\ &= \int P_{\text{data}}(x)P_1 \log \frac{P_{\text{data}}(x)P_1}{Q_f(y = 1|x)P_{\text{data}}(x)} dx + \int P_g(x)P_0 \log \frac{P_g(x)P_0}{Q_f(y = 0|x)P_{\text{data}}(x)} dx \\ &\propto \int P_{\text{data}}(x) \log \frac{1}{Q_f(y = 1|x)} dx + \int P_g(x) \log \frac{P_g(x)}{Q_f(y = 0|x)P_{\text{data}}(x)} dx. \end{aligned} \quad (3.13)$$

进而对齐分布 $P(x, y)$ 和 $Q(x, y)$ ，调整判别器 f 以最小化分布 $P(x, y)$ 和 $Q(x, y)$ 的差异，由此从公式 (3.13) 得到

$$\min_f D_{\text{KL}}(P(x, y) || Q(x, y)) = \max_f \mathbb{E}_{x \sim P_{\text{data}}(x)} [\log f(x)] + \mathbb{E}_{x \sim P_g(x)} [\log(1 - f(x))]. \quad (3.14)$$

易得解 $f^*(x) = Q_{f^*}(y = 1|x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)}$ ($a \log x + b \log(1 - x)$ 的极值点为 $x = \frac{a}{a+b}$)。由于生成器需要“欺骗”判别器，因此固定判别器 $f = f^*$ 后做相反动作，也即最小化公式 (3.14) 等式右边，从而得到

$$\begin{aligned} &\min_g \mathbb{E}_{x \sim P_{\text{data}}(x)} [\log f^*(x)] + \mathbb{E}_{x \sim P_g(x)} [\log(1 - f^*(x))] \\ &= \min_g \int P_{\text{data}}(x) \log \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)} dx + \int P_g(x) \log \left(1 - \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_g(x)}\right) dx \\ &= \min_g \int P_{\text{data}}(x) \log \frac{P_{\text{data}}(x)}{\frac{P_{\text{data}}(x) + P_g(x)}{2} \times 2} dx + \int P_g(x) \frac{P_g(x)}{\frac{P_{\text{data}}(x) + P_g(x)}{2} \times 2} dx \\ &= \min_g 2D_{\text{JS}}(P_{\text{data}}(x) || P_g(x)) - \log 4. \end{aligned} \quad (3.15)$$

所以回看公式(3.11)，实际上生成器试图最小化 $P_{\text{data}}(x)$ 和 $P_g(x)$ 之间的JS散度，而判别器试图对齐分布 $P(x,y)$ 和 $Q(x,y)$ ，因为 $P(x,y)$ 是生成器视角下的真实的混合分布，生成器有该混合分布的具体信息，而判别器则试图获取这些信息。当博弈达到均衡，则 $P_{\text{data}}(x) = P_g(x)$ ，该解同时满足公式(3.14)和公式(3.15)，因此虽然看起来判别器和生成器在对抗，但实际上殊途同归。

此外，尽管公式(3.15)可以对齐 $P_{\text{data}}(x)$ 和 $P_g(x)$ ，但实际上并不常用。因为在一开始，生成样本的质量比较差，从而 $P_{\text{data}}(x)$ 和 $P_g(x)$ 之间几无交叠，从而由JS散度导出的梯度并不稳定（见图3.1(b)）。因此实际上比较常用解决方法是，先固定 $g = g_0$ ，再令 $f = f^*$ ，则生成器仍然试图最小化 $P(x,y)$ 和 $Q(x,y)$ 的差异，由此从公式(3.13)得到

$$\begin{aligned} \min_g D_{\text{KL}}(Q(x,y) || P(x,y)) &= \min_g \int P_g(x) \log \frac{P_g(x)}{P_{f^*}(y=0|x)P_{\text{data}}(x)} dx \\ &= \min_g -\mathbb{E}_{x \sim P_g(x)} [\log f^*(x)] + D_{\text{KL}}(P_g(x) || P_{g_0}(x)) \end{aligned} \quad (3.16)$$

这里 $D_{\text{KL}}(P_g(x) || P_{g_0}(x))$ 表示一次交替优化过程中两个生成器之间的差异，假设每次更新 g 仅仅做微调，或者 g 参数收敛，则实际上该部分几乎可以忽略。从而实际上主要是调整 g 从而最小化 $-\mathbb{E}_{x \sim P_g(x)} [\log f^*(x)]$ ，将该目标函数作为公式(3.15)的替代，从而使得GAN在训练早期可以获取稳定的梯度。但还需注意的是，公式(3.16)不能保证 $P_{\text{data}}(x) = P_g(x)$ ，其数学意义仅仅是尽可能使判别器对所有样本均判别为观测样本。

另一种解决方法是换掉公式(3.15)的JS散度。比如，如果换成WD距离（见公式(3.7)），则可以得到Wasserstein GAN(WGAN)^[161]，如下：

$$\min_g \max_{f: \|f\|_L \leq 1} \mathbb{E}_{x \sim P_{\text{data}}(x)} [f(x)] - \mathbb{E}_{x \sim P_g(x)} [f(x)]. \quad (3.17)$$

值得注意的是，这里 $f(x) \in \mathbb{R}$ ，因此越高表示越倾向于判定为观测样本，反之则倾向于判定为生成样本。此外， $\|f\|_L \leq 1$ 表示判别器服从1-Lipschitz约束，也即 $\forall a, b, \|f(a) - f(b)\| \leq \|a - b\|$ 。文献^[161]使用clipping技巧来约束 f 的参数，也即通过 $\min(\max(f, -c), c)$, $c \in \mathbb{R}$ 将 f 的参数值约束在 $-c$ 到 c 之间，但在实际操作时，这会导致 f 的参数值集中在 $-c$ 和 c 上。因此，I. Gulrajani等人^[162]提出了梯度惩罚(Gradient Penalty, GP)以替换clipping技巧，从而得到改进的WGAN-GP，其通过如下公式优化判别器：

$$\max_f \mathbb{E}_{x \sim P_{\text{data}}(x)} [f(x)] - \mathbb{E}_{x \sim P_g(x)} [f(x)] + \lambda (\|\nabla_{\hat{x}} f(\hat{x})\| - 1)^2. \quad (3.18)$$

这里 λ 是惩罚因子。其巧妙之处在于，要满足1-Lipschitz约束，至少需要两个样本用于判定，而生成样本和观测样本正好满足。因此随机采样 $\epsilon \sim N(0, 1)$, $a \sim P_g(x)$, $b \sim P_{\text{data}}(x)$ ，则 $\hat{x} = \epsilon a + (1 - \epsilon)b$ ，则 \hat{x} 应满足 $\|\nabla_{\hat{x}} f(\hat{x})\| \leq 1$ ，从而得到梯度惩罚项。

还比如，如果换成MMD距离（见公式(3.2)），则可以得到MMD-GAN^[163]，如下：

$$\min_g \max_{f: \|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{x \sim P_{\text{data}}(x)} [f(x)] - \mathbb{E}_{x \sim P_g(x)} [f(x)]. \quad (3.19)$$

这里 $\|f\|_{\mathcal{H}} = K(f, f) \leq 1$, $K(\cdot, \cdot)$ 是再生核函数（可参考文献^[152]）。因此该约束表示判别器 f 的参数矢量的模长在再生核希尔伯特空间 \mathcal{H} 中不超过一，或者说参数矢量落在了再生核希尔伯特空间 \mathcal{H} 的一个单位球内。此外，如前所述（见公式(3.2)），再生核函数的选择会影响其MMD距离的值，因而MMD-GAN在使用时技巧性较强。

总之，不论如何改变概率距离度量，其通过生成器和判别器之间对抗的形式各自求解参数的核心思想不会改变。此外，M. Mirza等人^[164]提出了一种条件生成对抗网络(Conditional GAN, CGAN)模型，该模型引入了条件标签，使得GAN模型可以按需要生成特定模式的场景数据。J. Yoon等人^[165]提出Time-series GAN模型用于生成时间序列数据。T. Karras等人^[166]提出StyleGAN3模型用于生成多风格的高分辨率图像数据。

3.1.4 生成流模型

VAE 构造“编码器 - 解码器”，并通过对齐编码后和解码前的特征 z 以得到 $P_{\text{data}}(x) = P_g(x)$ 。GAN 构造“生成器 - 判别器”，并通过对齐生成样本、观测样本和标签的联合分布，从而得到 $P_{\text{data}}(x) = P_g(x)$ 。相比于前二者，生成流 (Flow) 模型更为直接，其假设 g 可逆，从而由函数关系 $x = g(z)$ 直接得出：

$$\log P_g(x) = \log P(g^{-1}(x)) + \log \left| \det \left(\frac{dg^{-1}(x)}{dx} \right) \right|. \quad (3.20)$$

该公式就是概率意义下的变量变换公式 (Change of Variables Formula)，可以参考文献^[167] 中的 Theorem 2.1.8。这里 g^{-1} 是 g 的逆，从而直接将 $\log P_g(x)$ 代入公式 (3.4) 即可通过最大似然估计求解 g 。这样做有很多好处，其一是对 z 的假设更为宽松，VAE 和 GAN 中均假设特征 z 服从标准正态分布，而 Flow 模型则不做此约束。其二是一旦确定 z ，则 $P_g(x)$ 可以得到显式表示，也即公式 (3.20)。其三是 Flow 模型结构更为简洁。类似 VAE，其同样可以视为构造了“编码器 - 解码器”结构，只是 Flow 模型中的编码器 g^{-1} 和解码器 g 是同一个参数模型，如图 3.4 所示，因而其结构更为简洁，但缺点是约束了样本 x 和特征 z 具有相同的形状，因此当观测样本 x 维度过高时容易造成维度浪费问题。

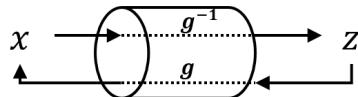


图 3.4: 生成流模型示意图

Flow 模型有很多优势，而这样做的困难点主要在于构造可逆的生成器 g 。为了使 g 满足可逆性，NICE (Nonlinear Independent Components Estimation) 模型^[41] 构造了一个可逆的人工神经网络模型 $g = g_1 \circ g_2 \circ \dots \circ g_T$ 。对于 $t = 1, \dots, T$ ， g_t 是耦合层，则映射 $g_t^{-1} : x \rightarrow z$ 满足：

$$\begin{aligned} z_1 &= x_1, \\ z_2 &= x_2 + f(x_1). \end{aligned} \quad (3.21)$$

这里 $x_1 \in \mathbb{R}^d$ 和 $x_2 \in \mathbb{R}^{N-d}$ 是对输入样本 $x \in \mathbb{R}^N$ 的一个划分，输出 $z \in \mathbb{R}^N$ 同样由 $z_1 \in \mathbb{R}^d$ 和 $z_2 \in \mathbb{R}^{N-d}$ 拼接而成。 $f(\cdot)$ 是一个由人工神经网络参数化的函数，将 x_1 从 d 维嵌入到 $N - d$ 维度。则映射 $g_t : z \rightarrow x$ 可以表示为：

$$\begin{aligned} x_1 &= z_1, \\ x_2 &= z_2 - f(z_1). \end{aligned} \quad (3.22)$$

因此易知 $g_t^{-1} \circ g_t$ 是恒等映射，而生成器 g 由多个这样的可逆的耦合层 $g_t, t = 1, \dots, T$ 复合而成，因此 g 也可逆。由于函数 $f(\cdot)$ 是非线性的，因而 g 在多次复合之后，其非线性表示能力更强。而输入样本 x 经过 $g_1^{-1}, \dots, g_T^{-1}$ 层层特征提取后得到特征 z ，再反过来经过 g_T, \dots, g_1 层层生成，最终得到场景样本 x ，因而 NICE 模型被称为生成流模型。

然而，由公式 (3.21) 解得

$$\frac{dg_t^{-1}(x)}{dx} = \begin{bmatrix} \mathbf{I}_{d \times d} & \mathbf{0} \\ \frac{df(x_1)}{dx_1} & \mathbf{I}_{N-d \times N-d} \end{bmatrix}. \quad (3.23)$$

因此， $\left| \det \left(\frac{dg_t^{-1}(x)}{dx} \right) \right| = 1$ ，由链式求导法则可知公式 (3.20) 中 $\left| \det \left(\frac{dg^{-1}(x)}{dx} \right) \right|$ 始终不变，从而梯度为零。

于是，L. Dinh 等人^[168] 提出 RealNVP (Real-valued Non-Volume Preserving) 模型用以改进 NICE 模型中可逆人工神经网络的可逆机制，从而优化可逆人工神经网络的梯度流。在 RealNVP 模型中，映射 $g_t^{-1} : x \rightarrow z$ 重新定义为：

$$\begin{aligned} z_1 &= x_1, \\ z_2 &= x_2 \odot e^{f_s(x_1)} + f_t(x_1), \end{aligned} \quad (3.24)$$

其中 $f_s(\cdot)$ 和 $f_t(\cdot)$ 均为由人工神经网络参数化的非线性函数， \odot 表示哈达玛积。因此 $g_t: z \rightarrow x$ 可以定义为：

$$\begin{aligned} x_1 &= z_1, \\ x_2 &= [z_2 - f_t(z_1)] \odot e^{-f_s(z_1)}. \end{aligned} \quad (3.25)$$

易知 $g_t^{-1} \circ g_t$ 是恒等映射，因此 g_t 可逆，则复合的 g 可逆。RealNVP 由此获得了比 NICE 更强的非线性表示能力和更易优化的梯度流，如下：

$$\frac{dg_t^{-1}(x)}{dx} = \begin{bmatrix} \mathbf{I}_{d \times d} & \mathbf{0} \\ \frac{\partial z_2}{\partial x_1} & \text{diag}(e^{f_s(x_1)}) \end{bmatrix}. \quad (3.26)$$

注意左下角 $\frac{\partial z_2}{\partial x_1}$ 虽然复杂，但不影响行列式计算结果。对比公式 (3.23) 和公式 (3.26)，可知后者使得公式 (3.20) 中 $\left| \det \left(\frac{dg_t^{-1}(x)}{dx} \right) \right|$ 产生变化，因而更易通过梯度优化 g 。

此外，基于 RealNVP 模型，D. P. Kingma 等人^[169] 提出 Generative Flow (Glow) 模型，该模型引入 1×1 卷积，从而构造了仿射耦合层，用于进一步优化 RealNVP 模型的梯度流。A. Pumarola 等人^[170] 提出了 Conditional Generative Flow (C-Flow) 模型，通过引入条件标签，实现多模式的条件生成。尽管 Flow 模型显式解出了 $\log P_g(x)$ ，但为了构造可逆人工神经网络，从而不得不约束特征 z 与观测样本 x 的形状始终保持一致，以实现多个耦合层的映射复合。由于 Flow 模型本身并没有设计额外的低维嵌入层，所以 Flow 模型存在着明显的维度浪费。比如对于图像数据，很多输入图片的背景像素点是几乎不变的，因而对这部分数据的学习毫无必要。因此，对于稍大尺寸的数据，比如高分辨率图片，Flow 模型则因为其参数量过于庞大而变得十分臃肿，极其限制其应用范围。

不过，Flow 模型更有价值的地方在于其利用概率化的正则流^[171, 172] 对生成过程的建模。其和传统系统建模方法的区别在于，系统建模方法会区分观测数据中的确定性和不确定性部分，确定性的部分是系统状态，不确定性的部分是噪声或扰动，从而可以构建一些随机微分方程模型或随机过程模型。而正则流则没有区分二者，其直接通过概率的方式描述数据分布在生成过程中的变化。正如本章节一开始所言，生成模型主要是面向系统中的不确定性进行建模，因而不管是 VAE、GAN 或是 Flow，其求解的 $P_g(x)$ 都没有区分系统状态、噪声和扰动等，而是将其“一概而论”地直接建模数据分布。

3.1.5 生成扩散模型

J. Sohl-Dickstein 等人^[42] 提出了一种基于非平衡热力学原理的生成方法，也叫生成扩散 (Diffusion) 模型。Diffusion 模型可以视为 VAE 模型的一个特例，同时又借鉴了 Flow 模型的正则流思想。其将 VAE 中的编码器 f 和解码器 g 建模成了正则流，如图 3.5 所示，令 $x_0 = x$ ，经过编码器 f 的层层编码后得到特征 $x_T = z$ ，再经过 g 的层层解码后得到生成的 $x_0 = x$ 。因此除了 x_0 是观测样本，其余的 $x_{1:T}$ 均可以视为提取的特征。

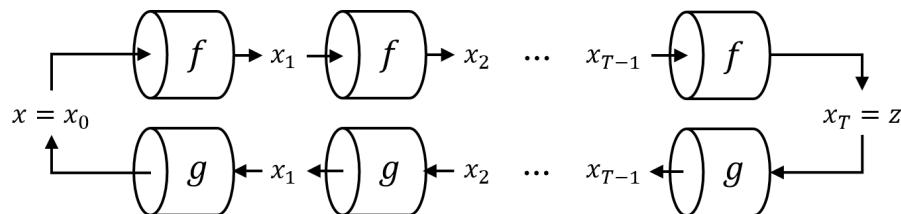


图 3.5: 生成扩散模型示意图

于是, 用 $x_{1:T}$ 替换公式 (3.8) 中的 z , 从而对齐 $x_{1:T}$ 得到:

$$\begin{aligned}
& D_{\text{KL}}(P_f(x_{1:T}|x_0)||P_g(x_{1:T}|x_0)) \\
&= \int \left[P_f(x_{1:T}|x_0) \log \frac{P_f(x_{1:T}|x_0)}{P_g(x_{1:T}|x_0)} \right] dx_{1:T} \\
&= \int \left[\prod_{t=2}^T P_f(x_{t-1}|x_t, x_0) P_f(x_T|x_0) \log \frac{P_g(x_0) \prod_{t=2}^T P_f(x_{t-1}|x_t, x_0) P_f(x_T|x_0)}{P_g(x_T) \prod_{t=1}^T P_g(x_{t-1}|x_t)} \right] dx_{1:T} \\
&= \log P_g(x_0) + D_{\text{KL}}(P_f(x_T|x_0)||P_g(x_T)) + \sum_{t=2}^T D_{\text{KL}}(P_f(x_{t-1}|x_t, x_0)||P_g(x_{t-1}|x_t)) - \mathbb{E}_{P_f(x_1|x_0)} \log P_g(x_0|x_1) \\
&\geq 0.
\end{aligned} \tag{3.27}$$

进而可得证据下界, 如下:

$$\log P_g(x_0) \geq -D_{\text{KL}}(P_f(x_T|x_0)||P_g(x_T)) - \sum_{t=2}^T D_{\text{KL}}(P_f(x_{t-1}|x_t, x_0)||P_g(x_{t-1}|x_t)) + \mathbb{E}_{P_f(x_1|x_0)} \log P_g(x_0|x_1). \tag{3.28}$$

然后将其代入公式 (3.4) 通过最大似然估计求解即可。

然而真正的难题在于如何构造 $x_{1:T-1}$ 的分布, 正如 VAE 中的重参数技巧, 实际上只假设了 $x_T \sim N(\mathbf{0}, \mathbf{I})$, 而 $x_{1:T-1}$ 等中间特征的分布并未假设。因此, J. Ho 等人^[173] 提出了一种去噪扩散概率模型 (Denoising Diffusion Probabilistic Models, DDPM), 其令 $P_f(x_t|x_{t-1}) \sim N(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$, 这里 β_t 是 t 时刻的扩散系数 (可参考文献^[174]), 是超参数。从而得到

$$\begin{aligned}
x_1 &= \sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon, \\
x_2 &= \sqrt{1-\beta_2}x_1 + \sqrt{\beta_2}\epsilon \\
&= \sqrt{1-\beta_2} \left[\sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon \right] + \sqrt{\beta_2}\epsilon \\
&= \sqrt{(1-\beta_2)(1-\beta_1)}x_0 + \sqrt{1-(1-\beta_2)(1-\beta_1)}\epsilon, \\
x_3 &= \sqrt{1-\beta_3}x_2 + \sqrt{\beta_3}\epsilon \\
&= \sqrt{1-\beta_3} \left[\sqrt{(1-\beta_2)(1-\beta_1)}x_0 + \sqrt{1-(1-\beta_2)(1-\beta_1)}\epsilon \right] + \sqrt{\beta_3}\epsilon \\
&= \sqrt{(1-\beta_3)(1-\beta_2)(1-\beta_1)}x_0 + \sqrt{1-(1-\beta_3)(1-\beta_2)(1-\beta_1)}\epsilon, \\
&\vdots \\
x_t &= \sqrt{\prod_{s=1}^t (1-\beta_s)}x_0 + \sqrt{1-\prod_{s=1}^t (1-\beta_s)}\epsilon.
\end{aligned} \tag{3.29}$$

这里 $\epsilon \sim N(\mathbf{0}, \mathbf{I})$, 从而得到 $P_f(x_t|x_0) \sim N(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$, 这里 $\bar{\alpha}_t = \prod_{s=1}^t (1-\beta_s)$ 。进而通过配方法得到

$$\begin{aligned}
P_f(x_{t-1}|x_t, x_0) &= \frac{P_f(x_t|x_{t-1}, x_0)P_f(x_{t-1}|x_0)}{P_f(x_t|x_0)} \\
&\propto e^{-\frac{1}{2} \left[\frac{(x_t - \sqrt{1-\beta_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\alpha_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\alpha_t} \right]} \\
&= C(x_t, x_0)e^{-\frac{1}{2} \left[x_{t-1} - \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \right) \right]^2 / \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t} \\
&\sim N \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I} \right).
\end{aligned} \tag{3.30}$$

这里 $C(x_t, x_0)$ 是配方余出来的关于 x_t 和 x_0 的项，由于变量仅为 x_{t-1} ，因而其可以视为常数项。值得再次强调的是，扩散系数 β_t 是超参数，且 $x_0 = x$ 是观测样本， $x_T = z \sim N(\mathbf{0}, \mathbf{I})$ 由假设可知服从标准正态分布，因而实际上公式 (3.30) 是 f 视角下 $x_t, t = 2, \dots, T$ 的显式解，而真正要做的是对齐 P_f 和 P_g 。

于是，令

$$P_g(x_{t-1}|x_t) \sim N(\boldsymbol{\mu}_g(x_t), \sigma_t^2 \mathbf{I}). \quad (3.31)$$

这里 $\boldsymbol{\mu}_g(x_t)$ 是由 g 输出的均值，取决于 x_t 。而 $\sigma_t^2 = \beta_t$ 或 $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$ 是个预设的常数，文献^[173] 认为这两种取值效果相同。公式 (3.31) 对 $t = 2, \dots, T$ 都有效，但对于 $t = 1$ 时，代入公式 (3.28) 中求解末项期望值则比较难解。因此，文献^[42] 中做了简化，其令

$$P_g(x_0|x_1) = P_f(x_0|x_1) = \frac{P_f(x_1|x_0)P_f(x_0)}{\int P_f(x_1|x_0)P_f(x_0)dx_0} = \text{const}, \quad (3.32)$$

这里由先前的假设易知 $P_f(x_1|x_0) \sim N(\sqrt{1-\beta_1}x_0, \beta_1 \mathbf{I})$ ，且 $P_f(x_0)$ 是观测样本的分布，从而该末项期望值可以视为常数。而文献^[173] 则设计了一个额外的解码器用于输出 $P_g(x_0|x_1)$ ，但该方法技巧性较强，主要针对图像数据，因此本文这里主要采用公式 (3.32)。此外还需注意的是， $x_0 = x$ 是观测样本， $x_T = z \sim N(\mathbf{0}, \mathbf{I})$ ，二者均已知，所以

$$D_{\text{KL}}(P_f(x_T|x_0)||P_g(x_T)) = \text{const} \quad (3.33)$$

也可以视为常数。

至此，公式 (3.28) 中所有项已经全部解出。从而将公式 (3.30)-(3.33) 代入公式 (3.28) 中，得到新的证据下界，如下：

$$\begin{aligned} \mathcal{L}(x, f, g) &= - \sum_{t=2}^T D_{\text{KL}}(P_f(x_{t-1}|x_t, x_0)||P_g(x_{t-1}|x_t)) + \text{const} \\ &= - \sum_{t=2}^T \frac{1}{2} \left[(\tilde{\boldsymbol{\mu}}(x_t, x_0) - \boldsymbol{\mu}_g(x_t))^{\top} \Sigma_2^{-1} (\tilde{\boldsymbol{\mu}}(x_t, x_0) - \boldsymbol{\mu}_g(x_t)) - N - \log \det(\Sigma_2^{-1} \Sigma_1) + \text{tr}(\Sigma_2^{-1} \Sigma_1) \right] \\ &\quad + \text{const} \\ &\propto - \sum_{t=2}^T \frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}(x_t, x_0) - \boldsymbol{\mu}_g(x_t)\|_2^2 \end{aligned} \quad (3.34)$$

这里 $\Sigma_1 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \mathbf{I}$, $\Sigma_2 = \sigma_t^2 \mathbf{I}$, N 是样本 x 的维度，也是常数，从而可以对证据下界近似。此外， $\tilde{\boldsymbol{\mu}}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$ 。所以实际上，Diffusion 模型的目标函数主要是对齐 f 和 g 在对应 $x_{1:T-1}$ 处的均值。于是更进一步，由公式 (3.29) 可知 $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ ，从而得到 $x_0 = (x_t - \sqrt{1-\bar{\alpha}_t}\epsilon) / \sqrt{\bar{\alpha}_t}$ 。代入 $\tilde{\boldsymbol{\mu}}_t(x_t, x_0)$ 可得

$$\begin{aligned} \tilde{\boldsymbol{\mu}}(x_t, x_0(x_t, \epsilon)) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon)}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}} + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t \\ &= \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right), \end{aligned} \quad (3.35)$$

因此，同理可得

$$\boldsymbol{\mu}_g(x_t) = \tilde{\boldsymbol{\mu}}(x_t, x_0(x_t, \epsilon_g(x_t))) = \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_g(x_t) \right) \quad (3.36)$$

这里 $\epsilon_g(x_t)$ 表示将 x_t 输入生成器 g 后输出的噪声值。从而将公式 (3.35) 和公式 (3.36) 代入公式 (3.34)，可进一步简化证据下界：

$$\begin{aligned} \mathcal{L}(x, f, g) &= - \sum_{t=2}^T \frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}(x_t, x_0) - \boldsymbol{\mu}_g(x_t)\|_2^2 \\ &\propto - \sum_{t=2}^T \frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)} \|\epsilon - \epsilon_g(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)\|_2^2. \end{aligned} \quad (3.37)$$

从而将公式(3.37)代入公式(3.4)通过最大证据下界以最大化似然概率求解 g 。

基于DDPM模型,J. Song等人^[175]提出了DDIM(Denoising Diffusion Implicit Models)模型,其发现这里 $\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ 可一次直接求出,因而实际上从 $x_0 \rightarrow x_T$ 的“编码”过程中不需要逐步加噪声采样,这是DDIM模型的一个主要观点。DDIM模型的另一个主要观点是,可以直接令 $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$,且实际上 $\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}$ 。从而一方面,公式(3.37)中的 β_t 可以被消去,于是避免了对 β_t 的选择,而证据下界只取决于 $\bar{\alpha}_t$ 。另一方面,公式(3.30)和公式(3.31)的方差均为 σ_t^2 ,由于 σ_t^2 可以任意选择而由此增加了生成样本的丰富性。尽管如此,DDPM模型和DDIM模型是殊途同归的,前者是后者的一个特例。

实际上,DDPM模型原文^[173]还提出了一个隐含观点,也即可以直接令证据下界为

$$\mathcal{L}(x, f, g) = -\|\epsilon - \epsilon_g(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)\|_2^2. \quad (3.38)$$

其中 t 以 $1/T$ 的概率在 1 到 T 中随机选择。DDPM模型原文^[173]声称该目标函数实验效果更好。其中一个合理的解释是,因为扩散过程具有马尔可夫性,因此每一步的目标函数实际上是相互独立的,从而公式(3.37)前面的系数就可以被近似,并且不会影响实际效果,而且可以简化计算过程。

此外,A. Q. Nichol等人^[176]针对DDPM模型的对数似然概率目标函数做了诸多改进,将采样步数 T 从1000步提升至4000步。J. Austin等人^[177]提出了一种离散的Diffusion模型,主要针对离散状态的场景数据生成。B. Kawar等人^[178]提出了基于预训练的Diffusion模型,五倍提升模型的训练速度。J. Liu等人^[179]提出了针对残差变量的Diffusion模型,简言之,就是为特征 $x_{1:T}$ 增加了残差项。

3.1.6 得分生成模型

A. Hyvärinen^[180]提出得分匹配方法,定义如下:

$$\min_g \frac{1}{2} \mathbb{E}_{P_{\text{data}}(x)} \left[\|\nabla_x \log P_g(x) - \nabla_x \log P_{\text{data}}(x)\|_2^2 \right]. \quad (3.39)$$

文献^[180]中Theorem 2证明,公式(3.39)中目标函数值为零,当且仅当 $P_g(x) = P_{\text{data}}(x)$ 。P. Vincent^[181]将其推广到条件得分匹配,其后由Y. Song和S. Ermon^[182]将条件得分匹配方法用于场景数据生成,定义如下:

$$\min_g \frac{1}{2} \mathbb{E}_{P(\tilde{x}|x)P_{\text{data}}(x)} \left[\|\nabla_{\tilde{x}} \log P_g(\tilde{x}|x) - \nabla_{\tilde{x}} \log P(\tilde{x}|x)\|_2^2 \right]. \quad (3.40)$$

这里 $P(\tilde{x}) = \int P(\tilde{x}|x)P_{\text{data}}(x)dx$,也即对观测样本 x 施加噪声后得到的 \tilde{x} 的分布。如果噪声足够强,则 \tilde{x} 中关于 x 的部分被完全破坏。因此,Y. Song等人^[183]又将公式(3.40)转换为随机微分方程形式,类似生成扩散模型, $x_0 \sim P_{\text{data}}(x)$ 逐步扩散至 $x_T \sim N(\mathbf{0}, \mathbf{I})$,再反向逐步生成 $x_0 \sim P_g(x)$ 。其思路在于将生成扩散模型视为一个连续的随机微分方程,从而得到一个更加广义的条件得分目标函数。

首先,由文献^[174]可得前向和反向的连续扩散随机微分方程,如下:

$$\begin{cases} \text{前向扩散: } dx_t = \mathbf{F}(x_t)dt + \beta_t d\mathbf{w}, \\ \text{反向生成: } dx_t = [\mathbf{F}(x_t) - \beta_t^2 \nabla_{x_t} \log P_f(x_t|x_0)] dt + \beta_t d\mathbf{w}. \end{cases} \quad (3.41)$$

这里, $\mathbf{F}(x_t)$ 是随机微分方程中确定性的部分,又称为漂移项。 β_t 是扩散系数, $d\mathbf{w}$ 是维纳过程的增量,因此 $\beta_t d\mathbf{w}$ 又称为扩散项。具体推导过程还可参考文献^[184]。 $P_f(x_t|x_0)$ 是前向过程中 x_t 对应时刻的条件分布,由此可见,如果可知 $\log P_f(x_t|x_0)$,则可以通过反向过程最终生成 $x_0 \sim P_g(x)$ 。因此,由公式(3.40)(或参考文献^[181]中公式(4.3)),可直接构造条件得分目标函数如下:

$$\min_g \mathbb{E}_{P_f(x_t, x_0)} \left[\|s_g(x_t) - \nabla_{x_t} \log P_f(x_t|x_0)\|_2^2 \right]. \quad (3.42)$$

这里 t 以 $1/T$ 的概率在 1 到 T 中随机选择。 $s_g(x_t)$ 是生成器的值,输入 x_t ,输出对得分 $\nabla_{x_t} \log P_f(x_t|x_0)$ 的估计。由此可以发现,公式(3.42)和公式(3.38)在形式上十分相近,而前者可以视为后者的连续版本。

3.2 多模态场景数据生成

3.2.1 条件场景数据生成

条件场景数据生成是一类特殊的问题，简言之，就是在场景数据生成时增加条件标签 c 的约束，从而得到条件生成分布 $P_g(x|c)$ ，也即根据不同的标签 c 生成对应的场景数据。其中条件标签 c 同样本 x 一样都是观测数据，在章节 3.1 中，实际上是将二者一概而论为样本 x 。而在这里则将其区分开，因此 x 和 c 实际上并不要求是同一种数据形式，且通常是不一样的，由此产生了多模态场景数据生成的问题。此外，条件生成和一般的生成模型之间还有一些不同之处，最重要的区别在于条件标签 c 不由任何因素决定，因而可以得到一些等价的概率形式。比如，由公式 (3.4) 类似可得

$$\begin{aligned} \min_g D_{\text{KL}}(P_{\text{data}}(x, c) || P_g(x, c)) &= \min_g \iint P_{\text{data}}(x, c) \log \frac{P_{\text{data}}(x, c)}{P_g(x, c)} dx dc \\ &= \min_g \iint P_{\text{data}}(x, c) \log \frac{P_{\text{data}}(x|c)P_{\text{data}}(c)}{P_g(x|c)P_{\text{data}}(c)} dx dc \\ &= \max_g \mathbb{E}_{P_{\text{data}}(x, c)} [P_g(x|c)]. \end{aligned} \quad (3.43)$$

这里实际上假设了 $P_g(c) = P_{\text{data}}(c)$ ，因为条件样本 c 是观测样本，不受生成器 g 影响。

因此，把如上条件生成范式依次代入各个生成模型，可以得到对应的条件生成模型。首先是条件变分自编码器 (Conditional VAE, CVAE)^[158]，如图 3.6 所示，参考公式 (3.8) 可以得到

$$\begin{aligned} D_{\text{KL}}(P_f(z|x, c) || P_g(z|x, c)) &= \int P_f(z|x, c) \log \frac{P_f(z|x, c)}{P_g(z|x, c)} dz \\ &= \int P_f(z|x, c) \log \frac{P_f(z|x, c)P_g(x|c)P_g(c)}{P_g(x|z, c)P_g(c|z)P(z)} dz \\ &= D_{\text{KL}}(P_f(z|x, c) || P_g(z)) - \mathbb{E}_{P_f(z|x, c)} [\log P_g(x|z, c)] + P_g(x|c) \\ &\geq 0. \end{aligned} \quad (3.44)$$

这里假设了 $P_g(c) = P_g(c|z)$ ，因为条件样本 c 是观测样本，其不受 z 影响，如图 3.6 所示。进而得到证据下界，如下：

$$\log P_g(x|c) \geq -D_{\text{KL}}(P_f(z|x, c) || P_g(z)) + \mathbb{E}_{P_f(z|x, c)} [\log P_g(x|z, c)]. \quad (3.45)$$

代入公式 (3.43) 即可求解。

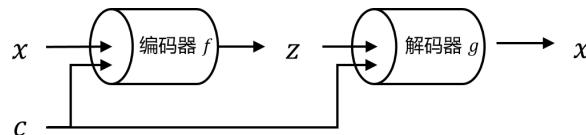


图 3.6: CVAE 示意图

类似还可得到条件生成对抗网络 (Conditional GAN, CGAN)^[164]，通过分别向生成器和判别器输入条件 c 以实现条件生成，如图 3.7 所示。参考公式 (3.12) 可以构造：

$$P(x, c, y) = \begin{cases} P_{\text{data}}(x, c)P_1, & \text{if } y = 1, \\ P_g(x|c)P_{\text{data}}(c)P_0, & \text{if } y = 0. \end{cases} \quad Q(x, c, y) = \begin{cases} P_{\text{data}}(x, c)Q_f(y = 1|x, c), & \text{if } y = 1, \\ P_{\text{data}}(x, c)Q_f(y = 0|x, c), & \text{if } y = 0. \end{cases} \quad (3.46)$$

这里 $P_0 = P_1 = \frac{1}{2}$, $Q_f(y=1|x, c) + Q_f(y=0|x, c) = 1$ 。从而得到:

$$\begin{aligned}
& D_{\text{KL}}(P(x, c, y) || Q(x, c, y)) \\
&= \iint \left[P(x, c, y=1) \log \frac{P(x, c, y=1)}{Q_f(y=1|x, c)} + P(x, c, y=0) \log \frac{P(x, c, y=0)}{Q_f(y=0|x, c)} \right] dx dc \\
&= \iint \left[P_{\text{data}}(x, c) P_1 \log \frac{P_{\text{data}}(x, c) P_1}{Q_f(y=1|x, c) P_{\text{data}}(x, c)} + P_g(x|c) P_{\text{data}}(c) P_0 \log \frac{P_g(x|c) P_{\text{data}}(c) P_0}{Q_f(y=0|x, c) P_{\text{data}}(x, c)} \right] dx dc \\
&\propto \iint \left[P_{\text{data}}(x, c) \log \frac{1}{Q_f(y=1|x, c)} + P_g(x|c) P_{\text{data}}(c) \log \frac{P_g(x|c) P_{\text{data}}(c)}{Q_f(y=0|x, c) P_{\text{data}}(x, c)} \right] dx dc.
\end{aligned} \tag{3.47}$$

进而类似公式 (3.14) 和公式 (3.15) 的思路, 构造生成器 g 和判别器 f 的博弈, 最终得到

$$\min_g \max_f \mathbb{E}_{(x, c) \sim P_{\text{data}}(x, c)} [\log f(x, c)] + \mathbb{E}_{x \sim P_g(x|c), c \sim P_{\text{data}}(c)} [\log(1 - f(x, c))]. \tag{3.48}$$

这里判别器 f 以 x 和 c 为输入, 并输出判别标签 y 。而生成器 g 也以 c 作为输入从而实现条件生成, 也即 $x = g(z, c)$, 这里 $z \sim N(\mathbf{0}, \mathbf{I})$ 。

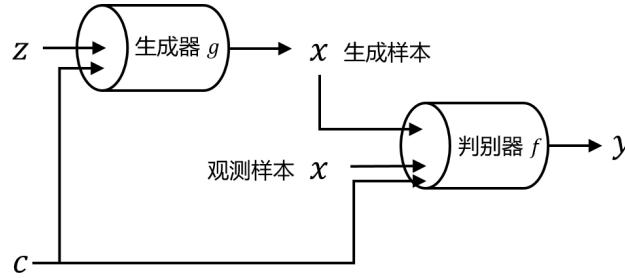


图 3.7: CGAN 示意图

类似还可得条件生成流 (Conditional Flow, CFlow) 模型, 简言之, 如果有函数关系 $x = g(z, c)$ 和 $z = g^{-1}(x, c)$, 如图 3.8 所示。于是参考公式 (3.20) 可得

$$\log P_g(x|c) = \log P(g^{-1}(x, c)) + \log \left| \det \left(\frac{\partial g^{-1}(x, c)}{\partial x} \right) \right| \tag{3.49}$$

代入公式 (3.43) 求解即可。

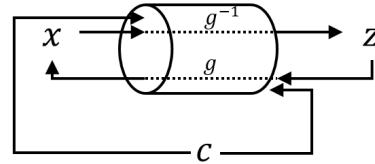


图 3.8: CFlow 示意图

当然, 问题的关键仍然在于构造可逆的 g 。本文参考 RealNVP 模型^[168] 构造 $g_t^{-1} : (x, c) \rightarrow z$, 如下

$$\begin{aligned}
z_1 &= x_1, \\
z_2 &= x_2 \odot e^{f_s(x_1, c)} + f_t(x_1, c)
\end{aligned} \tag{3.50}$$

从而得到 $g_t : (z, c) \rightarrow x$, 如下:

$$\begin{aligned}
x_1 &= z_1, \\
x_2 &= [z_2 - f_t(z_1, c)] \odot e^{-f_s(z_1, c)}
\end{aligned} \tag{3.51}$$

这里 $x_1 \in \mathbb{R}^d$ 和 $x_2 \in \mathbb{R}^{N-d}$ 是对输入样本 $x \in \mathbb{R}^N$ 的一个划分, 输出 $z \in \mathbb{R}^N$ 同样由 $z_1 \in \mathbb{R}^d$ 和 $z_2 \in \mathbb{R}^{N-d}$ 拼接而成。从而复合 $g_1 \circ \dots \circ g_T = g$, 于是可得

$$\frac{\partial g_t^{-1}(x, c)}{\partial x} = \begin{bmatrix} \mathbf{I}_{d \times d} & \mathbf{0} \\ \frac{\partial z_2}{\partial x_1} & \text{diag}(e^{f_s(x_1, c)}) \end{bmatrix} \tag{3.52}$$

代入公式(3.49)即可求解。此外, A. Pumarola 等人^[170]还提供了另一种构造方法, 其除了类似公式(3.50)和公式(3.51)对 x 和 c 进行转换, 还构造了另一个 RealNVP 模型对 c 进行转换。但本文认为转换 c 的 RealNVP 模型是没有必要的。一方面, 如果转换 c , 则意味着反向生成 x 的过程中输入的条件标签 c 是提取后的特征, 而不是 c 本身, 这与 CVAE (见图 3.6) 和 CGAN (见图 3.7) 的条件生成形式有本质差别。另一方面, 这种构造方法会使得模型参数量翻倍, 于是维度浪费的严重程度也会加倍。不过, 文献^[170]在多模态场景下会有一些参考价值。

类似还可得到条件生成扩散 (Conditional Diffusion, CDiffusion) 模型, 简言之, 就是在 x_0, \dots, x_T 之间前向扩散和反向生成的每一步中都加入条件标签 c , 如图 3.9 所示。在这个过程中, 条件标签 c 的加入会引起 x_t 均值的偏移, Z. Zhang 等人^[185]提出了 ShiftDDPMs 模型以建模这个均值偏移的过程。

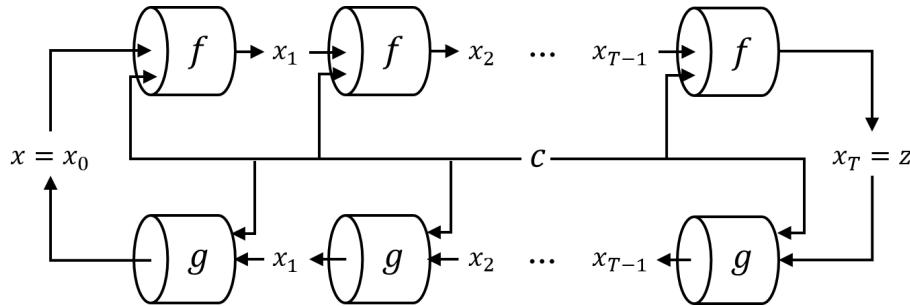


图 3.9: CDiffusion 模型示意图

首先直接定义 $P_f(x_t|x_0, c) = N(\sqrt{\bar{\alpha}_t}x_0 + c_t, (1 - \bar{\alpha}_t)\mathbf{I})$, 这里 $\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$, β_t 是扩散系数。此外, $c_t = h_t(c)$ 是偏移量, $h_t(\cdot)$ 是关于 c 的函数。从而依据公式(3.29)倒推,

$$\begin{aligned} x_t &= \sqrt{\bar{\alpha}_t}x_0 + c_t + \sqrt{(1 - \bar{\alpha}_t)}\epsilon, \\ &= \sqrt{1 - \beta_t} \left[\sqrt{\bar{\alpha}_{t-1}}x_0 + c_{t-1} + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon \right] - \sqrt{1 - \beta_t}c_{t-1} + c_t - \sqrt{1 - \beta_t}\sqrt{1 - \bar{\alpha}_{t-1}}\epsilon + \sqrt{(1 - \bar{\alpha}_t)}\epsilon \\ &= \sqrt{1 - \beta_t}x_{t-1} - \sqrt{1 - \beta_t}c_{t-1} + c_t + \sqrt{\beta_t}\epsilon, \end{aligned} \quad (3.53)$$

从而可得 $P_f(x_t|x_{t-1}, c) = N(\sqrt{1 - \beta_t}x_{t-1} - \sqrt{1 - \beta_t}c_{t-1} + c_t, \sqrt{\beta_t}\mathbf{I})$ 。从而参考公式(3.30)配方可得

$$\begin{aligned} &P_f(x_{t-1}|x_t, x_0, c) \\ &= \frac{P_f(x_t|x_{t-1}, x_0, c)P_f(x_{t-1}|x_0, c)}{P_f(x_t|x_0, c)} \\ &\propto e^{-\frac{1}{2} \left\{ \frac{[x_t - (\sqrt{1 - \beta_t}x_{t-1} - \sqrt{1 - \beta_t}c_{t-1} + c_t)]^2}{\beta_t} + \frac{[x_{t-1} - (\sqrt{\bar{\alpha}_{t-1}}x_0 + c_{t-1})]^2}{1 - \bar{\alpha}_{t-1}} - \frac{[x_t - (\sqrt{\bar{\alpha}_t}x_0 + c_t)]^2}{1 - \bar{\alpha}_t} \right\}} \\ &= C(x_t, x_0, c)e^{-\frac{1}{2} \left[x_{t-1} - \left(\frac{\beta_t \sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + c_{t-1} - \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} c_t \right) \right]^2 / \frac{\beta_t(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}} \\ &\sim N \left(\frac{\beta_t \sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + c_{t-1} - \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} c_t, \frac{\beta_t(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{I} \right). \end{aligned} \quad (3.54)$$

这里 $C(x_t, x_0, c)$ 是配方余出来的关于 x_t 、 x_0 和 c 的项, 由于变量仅为 x_{t-1} , 因而其可以视为常数项。又由于 $P_f(x_t|x_0, c) = N(\sqrt{\bar{\alpha}_t}x_0 + c_t, (1 - \bar{\alpha}_t)\mathbf{I})$, 所以 $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - c_t - \sqrt{1 - \bar{\alpha}_t}\epsilon)$ 。从而令 $\tilde{\mu}(x_t, x_0, c)$ 为公式(3.54)中 $P_f(x_{t-1}|x_t, x_0, c)$ 的均值, 参考公式(3.35), 可得

$$\begin{aligned} &\tilde{\mu}(x_t, x_0(x_t, \epsilon, c), c) \\ &= \frac{\beta_t \sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_t} \frac{1}{\sqrt{\bar{\alpha}_t}} (x_t - c_t - \sqrt{1 - \bar{\alpha}_t}\epsilon) + \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + c_{t-1} - \frac{\sqrt{1 - \beta_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} c_t \\ &= \frac{1}{\sqrt{1 - \beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) - \frac{1}{\sqrt{1 - \beta_t}} c_t + c_{t-1}. \end{aligned} \quad (3.55)$$

从而可以定义反向生成过程，如下：

$$P_g(x_{t-1}|x_t, c) \sim N(\boldsymbol{\mu}_g(x_t, c), \sigma_t^2 \mathbf{I}). \quad (3.56)$$

这里 σ_t^2 是场景数据生成时预设的方差，并且

$$\boldsymbol{\mu}_g(x_t, c) = \tilde{\boldsymbol{\mu}}(x_t, x_0(x_t, \epsilon_g(x_t, c), c), c) = \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_g(x_t, c) \right) - \frac{1}{\sqrt{1-\beta_t}} c_t + c_{t-1} \quad (3.57)$$

最后参考公式 (3.34) 和公式 (3.37)，可以得到最终的证据下界，如下：

$$\begin{aligned} \mathcal{L}(x, f, g) &= - \sum_{t=2}^T D_{\text{KL}}(P_f(x_{t-1}|x_t, x_0, c) || P_g(x_{t-1}|x_t, c)) \\ &= - \sum_{t=2}^T \frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}(x_t, x_0, c) - \boldsymbol{\mu}_g(x_t, c)\|_2^2 \\ &= - \sum_{t=2}^T \frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)} \|\boldsymbol{\epsilon} - \epsilon_g(\sqrt{\bar{\alpha}_t}x_0 + c_t + \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon})\|_2^2 \end{aligned} \quad (3.58)$$

当然也可以参考公式 (3.38) 得到更简化的证据下界，思路是一样的，只是多了个由条件 c 引出的偏移项。

3.2.2 多模态融合表示和跨模态表示转换

至此，本文已经讨论了最基本的生成模型，以及有条件标签的条件生成模型。本文进而讨论其二者与多模态场景数据生成的关联，简言之，多模态场景数据生成在模型层面仅有这两种形式，一种是最基本的生成模型，但是需要将场景样本联合表示，另一种是在条件生成模型中，将条件标签、观测样本和生成样本均视为不同模态的表示，从而实现多模态场景数据生成。因此，模型层面不再过多论述，本章节将多模态场景数据生成作为任务进行论述，主要包括多模态融合表示和跨模态表示转换等。

首先定义多模态场景数据生成针对多模态感官信息（如视觉、听觉、嗅觉、味觉、触觉等）进行研究，因而其数据表现形式多种多样，比如有时间序列数据、图结构数据、文本数据、图像数据、语音数据、表格数据等^[186]，如图 3.10(a)所示。这些不同形式的数据之间往往具有关联，比如，如果已有一段语音和一段文本，本来是通过两种模态数据进行记录，但也可以将其融为一体，仅通过另一种模态进行表示，这种提取多种模态间的联合表示信息就是一种多模态的融合表示任务，如图 3.10(b)所示。再比如一个人说出一句话，可以记录为语音，也可以记录为文本，但无论形式如何，其内容是一样的，因此如果语音数据有缺失，那么就可以用具有同样内容文本数据进行恢复，这种从文本到语音的转换就是一种跨模态的表示转换任务，如图 3.10(c)所示。这种定义可以扩展到更多模态且更加复杂的任务场景，这里将样本 x 拆分成 $x^{(1)}, \dots, x^{(N)}$ 等多模态形式，将联合表示特征标记为 z 。

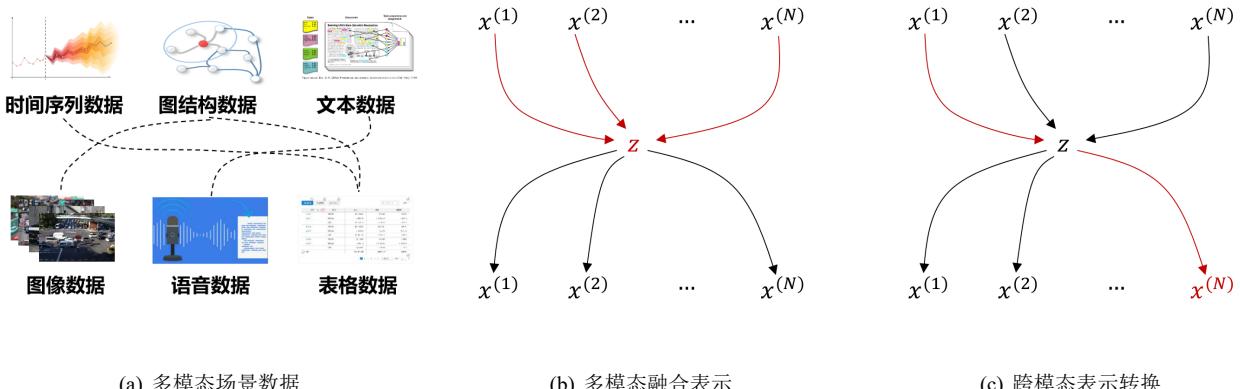


图 3.10: 多模态场景数据生成示意图

此外值得一提的是，多模态场景数据生成还有另一种称呼，也即“多视角”场景数据生成，在该研究出现早期，其主要是指对同一个观测目标从多个视角进行描述，从而可以利用不同视角下的观测数据之间的共有信息相互插值补全^[187, 188]。但实际上，经过多年的研究和发展，多视角已经逐步扩展到多模态领域，并且其二者无论是在面向的问题，还是在使用的解决方法上，都几无区别。本文作者在研究多模态场景数据生成时，常常遇到审稿人询问其二者的区别，可见该问题十分重要，需要在此澄清，多模态场景数据生成侧重于多模态场景数据的研究，而多视角场景数据生成则侧重利用不同视角下的观测数据实现数据插值补全任务，但现如今，其二者实际上已经没有任何区别，因此本文仅以多模态场景数据生成言之。

回归正题，多模态场景数据生成自然而然会面临的两个问题，其一是要生成场景数据，对于不同的模态，都需要对应的特征 z 以提供不确定性。然而由于有 $x^{(1)}, \dots, x^{(N)}$ 等 N 个模态，因此理论上需要不同的特征 z ，但在多模态场景数据生成中，特征 z 可以联合表示。此时的特征 z 是多个模态的联合特征，该联合特征需要在同一个特征空间中对齐。因此，比较直接的解决方法是将联合特征表示成统一的特征矢量，该方法又称为矢量化嵌入技术。矢量化嵌入技术有两个好处：（1）矢量化嵌入表示的特征容易对齐，且可解释性较高，通过欧氏距离或内积等度量可以直接计算不同特征之间的差异^[189, 190]。（2）矢量化嵌入表示可以节省表示的开销。比如在词矢量化嵌入问题中，早期的词矢量化技术主要是采用 one-hot 编码，这是一种离散的编码表示，一个词对应一个 0-1 矢量，且矢量之间两两相交，当词量巨大的时候，这种表示方法十分消耗计算资源。而 word2vec 技术^[191-193] 则将词嵌入到统一的特征空间中，从而使其可以连续表示，因而特征空间的维度可以更低，特征表示的开销也就更低。

且由此引出第二个问题，如何将不同的模态在同一个特征空间中对齐，并且又重新生成到同一个或另一个模态。由于不同模态之间的数据表示形式可能不同，所以一个最直接的解决方法就是采用“多编码器 - 多解码器”结构，如图 3.11 所示。简言之，就是一个模态对应一个编码器用于矢量嵌入，且对应一个解码器用于场景数据生成。该思想最早出现于 I. Sutskever 等人^[194] 提出的“序列 - 序列 (Sequence to Sequence)”结构，其同时也参考了变分自编码器的“编码器 - 解码器”结构，见图 3.2。

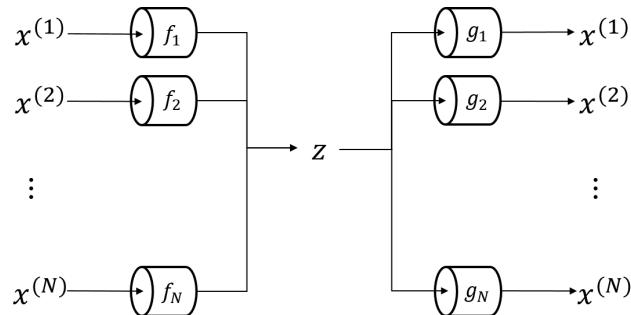


图 3.11: “多编码器 - 多解码器” 结构示意图

于是不同的编码器和解码器可由多个人工神经网络参数化，从而构成一个整体的学习系统，但这又会产生新的问题。其一是协同学习的问题^[195]。因为在多模态场景数据生成中，其数据往往异步获取，且不同模态数据的处理速度也不同，受限于数据量、模型参数量和算力，这意味着不同的人工神经网络需要协同完成场景数据生成。其二是数据安全问题。多模态数据往往是多源异构的，这意味着不仅是数据结构不同，其数据来源可能也不同。但为了保证数据安全，则不可能维持大量的数据在不同的来源地之间传输。一个比较好的解决方法是采用联邦学习方法^[196]，该方法在去中心化数据存储的基础上采用中性的学习机制更新模型参数，从而避免了由数据传输而引发的安全问题。群智学习方法^[197] 比联邦学习方法更进一步，其在学习机制上采用进化算法从而实现了去中心化，从而避免了中性化学习时模型参数传递而引发的安全问题。其三是模块化和稀疏性的问题^[198]。当模态数量上升，随之由多个人工神经网络组成的学习系统也会极其庞大和复杂，此时就需要对学习系统采用模块化设计。通过解耦局部功能并将其封装成功能模块，并在模块间保证通信，从而使其完成更加复杂的任务。这样做有两个好处，（1）学习系统具有弹性，可参考章节 2.2.3，部分模块被破坏不会影响其整体功能，且新模块能更加容易的接入

系统。(2) 功能输出具有稀疏性, 可以仅由少数几个模块决定, 而不用耗费整个系统的计算资源。

3.2.3 专用多模态特征提取

在多模态场景数据生成问题中, 需要解决从一种模态场景数据 $x^{(i)}$ 中提取特征 z , 以及再从特征 z 生成另一种模态场景数据 $x^{(j)}$ 的问题。理论上, 利用前馈神经网络(见章节 2.1.3)可以实现任意模态场景数据的特征提取, 因为不论是什么模态, 其场景数据总是能“展平”成一个矢量, 从而符合前馈神经网络的输入规则。然而, 该做法有一些缺点: (1) 特征提取计算效率低。因为前馈神经网络的每一层都会在所有输入节点和输出节点之间形成连边, 但这些连边往往是稀疏的, 也即大量的连边权重是零值, 所以对这些连边, 无论是在计算机中存储其权重还是在反向传播算法中更新其参数, 都十分消耗计算资源。(2) 对输入场景数据的重新排列可能会使其丢失关键信息。比如, 多变量的时间序列数据往往表示成形状为 $T \times N$ 的矩阵, 这里 T 是序列长度, N 是变量数。比如, 图像数据往往表示成形状为 $L \times W \times C$ 的张量, 这里 L, W, C 分别表示为图像的横向和纵向的像素点个数, 以及色彩空间的维度。因此如果直接将其“展平”成一个矢量, 那么时间序列会丢失变量间交互的空间信息, 图像数据则会丢失像素点之间固有的位置等关键信息。因此, 针对多模态场景数据, 往往需要针对其模态特性设计专用的人工神经网络用于特征提取。

于是, 首先是具备序列特性的场景数据, 比如文本词句、音频和其他时间序列数据等。这里用 x_1, \dots, x_T 表示序列样本, 因此其可以分为两个主要部分, 其一是序列样本的数值 $x_{1:T}$, 其二是与之对应的序列标签 $[1, \dots, T]$ 。根据对序列标签的特征提取方式, 发展出两种主要的序列特征提取方法, 也即显示和隐式序列特征提取方法。显式序列特征提取是指把序列标签与序列样本一同输入前馈神经网络进行编码, 从而得到特征 z , 如图 3.12(a)所示。而隐式序列特征提取则不会另外输入序列标签, 而是将序列样本按序列顺序依次输入 Cell 中, 从而将序列标签隐式编码进特征 z 中, 如图 3.12(b)所示。这里 Cell 是一个前馈神经网络的具体实现, 其编码原理可以参考公式(2.27)-(2.31)。

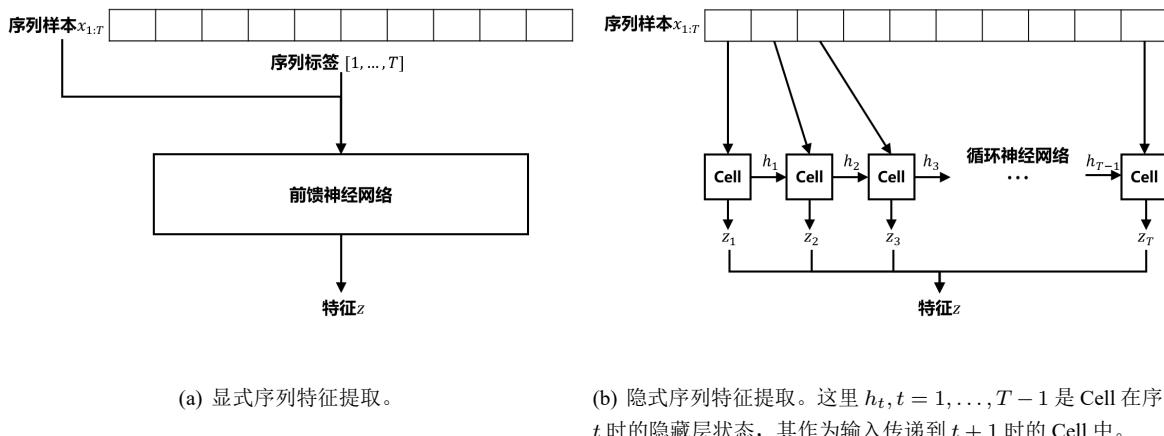


图 3.12: 序列特征提取示意图

实际上, 在序列特征提取研究的早期主要采用隐式编码的方式, 其存在一条研究主线, 从具备循环结构的循环神经网络^[93,94]开始。然而, 在基于循环神经网络的序列特征提取问题中, 由于其采用顺序依次提取特征的方式, 如图 3.12(b)所示, 因此 Cell 主要依靠其隐藏层状态保留一部分先提取的输入数据的特征, 但这部分特征会被后续输入的数据挤压其存储空间, 于是随着序列 t 的推进, 先提取的特征渐渐失真, 从而后提取的特征中无法保有先提取的特征的信息, 从而使得长序列的特征提取失效。因此, M. Schuster 和 K. K. Paliwal^[199]提出了双向循环神经网络, 其主要思想是顺序可以提取一遍特征, 倒序还可以再提取一遍特征, 从而缓解长序列特征信息失真的问题。但该方法并不能从根本上解决该问题, 因为当序列长度足够长时, 中间段的序列特征信息仍然会失真。基于此, 长短期记忆网络^[95,96]额外设计了一个贯穿全部序列特征提取的存储机制, 从而使得任何序列位置的 Cell 都能够访问此前提取的特征信息。

长短期记忆网络为此设计了输入门、输出门和遗忘门来控制这部分特征的存储和传递，从而解决了循环神经网络中长序列特征提取困难的问题。门控循环单元网络^[97]则在长短期记忆网络的基础上，将三个门合并为更新门和重置门，从而使用更少参数就可以实现长短期记忆网络同等的长序列特征提取性能。但实验上证明，门控循环单元网络与长短期记忆网络相比，通常其超参数（循环神经网络层数、隐藏层维度等）设置相同时，门控循环单元网络的性能一般优于长短期记忆网络。再然后，就是考虑连续型的循环神经网络^[101-103]，连续型循环神经网络是神经微分方程的一个特例，因此这里直接参考章节 2.1.4 即可。

然而，循环神经网络存在其根本性的问题，其一是此前提及的长序列特征提取难题，其二是特征提取效率的问题。由于循环神经网络必须依次将序列样本输入 Cell 中，因此其特征提取过程必须遍历每一个序列样本，则时间复杂度为 $\mathcal{O}(T)$ 。而相对地，显式序列特征提取则避开了这两个问题。由于其利用前馈神经网络直接编码序列样本数值和序列标签，因此其无需遍历每一个序列样本，而是并行特征提取，则一次特征提取的时间复杂度为 $\mathcal{O}(1)$ 。此外，由于前馈神经网络在每一个输入序列样本和输出特征之间构建起连接，因而其特征提取是全局的，从而不存在长序列特征提取的难题。但同时如前所述，如此做需要耗费巨大的计算资源以维持相应的人工神经网络规模。显式序列特征提取的一个具体实现就是注意力网络 (Attention Network, Transformer)^[200]，基于其会引出另一条研究主线，本文后续还会提及并讨论。

除了序列数据，还有具备线性变换（如平移、旋转、缩放等）作用下几何不变性的场景数据，又称为欧氏结构数据，比如图像、音频和文本词句等。此类数据的特性是其样本数值之间存在固定的位置关联，且这种关联不因为线性变换而发生几何性质上的改变，因此显然音频、文本词句等序列数据实际上也属于此类。因此欧氏结构数据也可以分为显式和隐式两种特征提取方法，显式特征提取则是将位置标签（序列标签是其一种）和样本数值一同输入前馈神经网络，其一个具体实现就是注意力网络^[200]，本文留到后续讨论。而隐式特征提取则在特征提取的过程中就将位置标签编码进特征中。

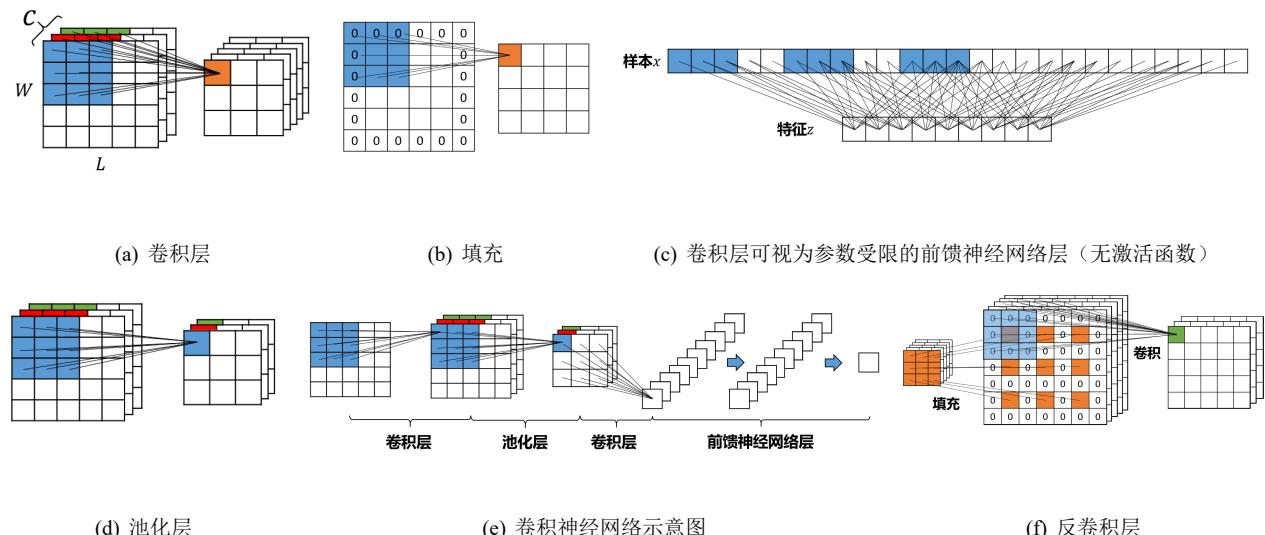


图 3.13: 卷积神经网络及其各组成部分示意图

为了实现隐式的欧氏结构数据特征提取，Y. LeCun^[201]提出了卷积神经网络用于提取图像数据的结构特征，因此这里以形状为 $L \times W \times C$ 的图像数据举例。卷积神经网络可解耦成两种模块，其一是卷积层，如图 3.13(a)所示，卷积层利用一个 $L' \times W' \times C$ 的卷积模板，从左上角第一个像素点开始，按照固定的步长依次从左向右滑动，当滑动到最右端，则换行继续按照从左至右提取特征，直至最后一个像素点。卷积模板中的每一个像素点都连接一个连边，像素点的数值同连边的权值做加权求和运算，从而得到右侧的橙色像素点。如果有多个卷积模板，则会卷积多次，于是在相同位置得到多个像素点，不同卷积模板计算得到的特征在色彩空间维度上堆叠，最终得到特征 z 。注意，不同卷积模板的参数不同，因此提取的特征也不同。此外，经过卷积得到的图像特征尺寸会比原图像尺寸更小，为了保证图像特征与原图像尺寸相

同，需要在特征提取前对原图像采取填充操作，如图 3.13(b)所示，填充零值的好处在于零值不会影响加权求和的结果。经过适当的填充操作，可以得到形状为 $L \times W \times C'$ 的图像特征，这里 C' 为卷积模板的数量，也是特征所属的色彩空间的维度。此外，卷积神经网络的参数由卷积模板决定，因此尽管卷积模板在原图像的不同像素点间滑动，但这些像素点共享了卷积模板的参数。因此实际上，卷积层可以视为一个参数受限的前馈神经网络层（无激活函数），如图 3.13(c)所示，将原图像“展平”成矢量，则蓝色像素点为一个卷积模板覆盖原图像像素点，卷积模板按照上述规律在矢量上的不同像素点间滑动，从而提取特征。其关键点在于，(1) 像素点共享卷积模板参数，因而对于图 3.13(c)中的前馈神经网络，其连边均采用卷积模板的参数，因而实际上只需要存储和更新卷积模板参数即可，计算效率更高。(2) 卷积模板只覆盖有限的局部像素点，因而更有利于提取局部的图像特征，避免了全局特征提取带来的误差累积。

其二是池化层，主要功能在于从输入图像的特征中仅保留关键特征，从而缩小图像特征的尺寸，因而又称为下采样层，如图 3.13(d)所示。其同样利用一个形状为 $L' \times W'$ 的模板在不同的像素点间和色彩空间维度上滑动，其会将模板覆盖到的像素点按照一定规则采样出来。其主要有两种采样方法，一种是平均池化，也即采样输出所有覆盖到的像素点的均值，其物理意义在于对输入图像做平滑处理。另一种是最大池化，也即采样输出所有覆盖到的像素点的最大值，其物理意义在于通常图像中的关键特征会比较亮，因而样本数值较高。与卷积层不同的一点是，池化层的模板没有需要更新的参数。由卷积层和池化层可以组成完整的卷积神经网络，如图 3.13(e)所示，其中前馈神经网络层常用于输出最后的特征。除了卷积层，还有反卷积层^[202]，如图 3.13(f)所示，其由填充操作和池化层组成，主要功能是提取比原图像尺寸更大图像特征，但实际上由于其仍然是采用卷积层以实现其功能，因此仍然会有信息损失，所以不能完全视为卷积层的逆操作。

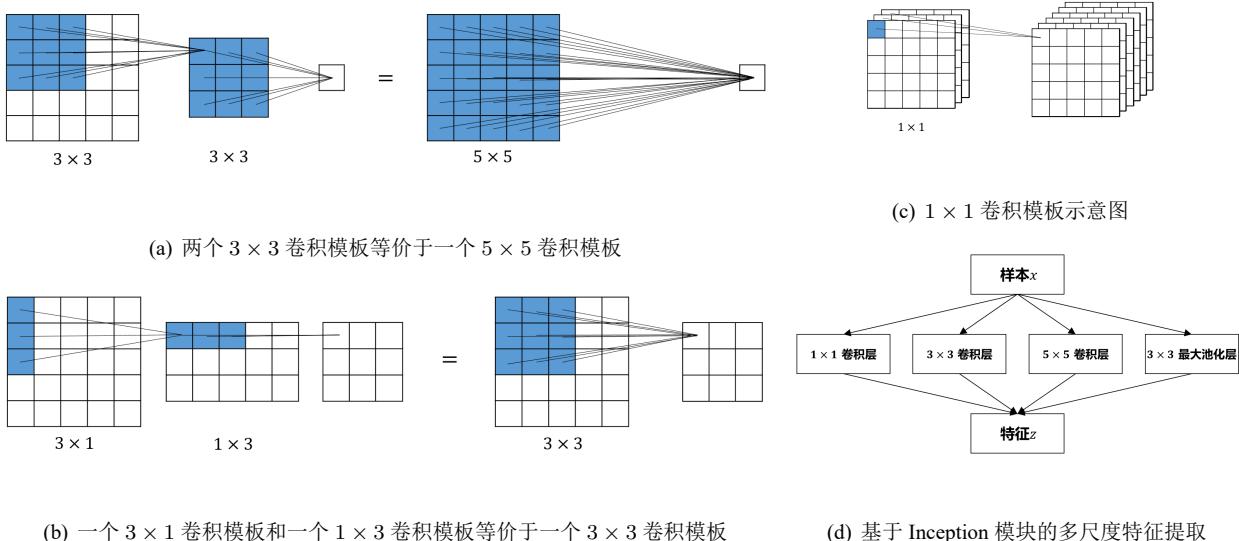


图 3.14: 不同尺寸的卷积模板等价性示意图

卷积神经网络的其中一条研究主线是关于卷积模板覆盖范围的研究，也称为关于感受野 (Receptive Field) 的研究。在文献^[201] 中，卷积神经网络主要采用 5×5 尺寸的卷积模板，而 K. Simonyan 和 A. Zisserman^[203] 则认为 5×5 卷积模板可以等价于两个 3×3 卷积模板，如图 3.14(a)所示，两个 3×3 卷积模板的总覆盖范围与 5×5 卷积模板相同。注意这里每一个卷积层最后均经过激活函数激活，因此卷积层可以理解为一个非线性的映射，所以在卷积模板总覆盖范围相同的情况下，两个 3×3 卷积模板所形成的非线性映射的复合与 5×5 卷积模板所形成的非线性映射相同。基于这一思路，C. Szegedy 等人^[204] 进一步提出用一个 3×1 卷积模板和一个 1×3 卷积模板替代一个 3×3 卷积模板，如图 3.14(b)所示。这样做主要有两个好处，(1) 卷积神经网络参数量进一步精简，比如在图 3.14(a)中，两个 3×3 卷积模板的参数量是 $3 \times 3 + 3 \times 3 = 18$ ，而一个 5×5 卷积模板的参数量是 25，因此前者的参数量少于后者。图 3.14(b)同理， $3 \times 1 + 1 \times 3 = 6 < 9 = 3 \times 3$ ，因此这种拆分会精简模型参数，从而提高计算效率。(2) 加深人工神经网

络层数，从而提高人工神经网络的表示能力。如前所述，前馈神经网络的深度（神经网络层数）和宽度（每层神经元数）会影响其近似效果，当神经元数量固定，其深度增加会导致其表示能力增加，且存在指数上界和多项式下界^[17-19]。不过，这种由卷积模板拆分导致的层数增加会引起网络退化，其解决方法是设计具有残差结构的卷积神经网络^[13,14]，残差网络可参考图 2.3(c)和图 2.4。G. Huang 等人^[205] 提出基于密集连接的卷积神经网络，该网络模型在图像分类实验效果上优于残差网络，但可解释性不足。

卷积神经网络的另一条研究主线是关于多尺度特征提取的研究。C. Szegedy 等人^[206] 提出了 1×1 卷积模板用于图像特征色彩空间维度的缩放，如图 3.14(c)所示。 1×1 卷积模板不会改变图像特征的尺寸，但在一个卷积层设置多个卷积模板，可以改变色彩空间的维度。此外，C. Szegedy 等人^[206] 还提出 Inception 模块，如图 3.14(d)所示，一个 Inception 模块由 1×1 、 3×3 、 5×5 等尺寸的卷积模板和一个 3×3 尺寸的池化层组成，从而可以提取多尺度的欧氏结构特征，文献^[204,207] 均利用卷积模板的等价性原理将 Inception 模块中各尺寸卷积模板拆分，从而实现更好的实验效果，但其特征提取原理仍然是多尺度的。此外还值得一提的是，Inception 模块的另一个重要创新在于其对于卷积神经网络的模块化设计，这种模块化设计更好的解耦了层与层之间的功能，如前所述，这有利于多模态特征提取。进而，G. Huang 等人^[208] 结合了密集连接和多尺度特征提取等重要思想，提出了多尺度密集网络模型，用于图像分类。

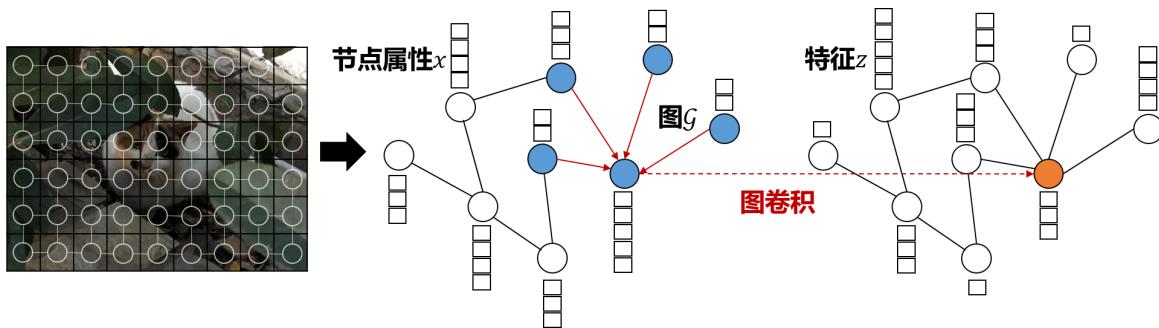


图 3.15: 图卷积神经网络示意图

除了欧氏结构数据，就是非欧氏结构数据，其通常不具备线性变换作用下的几何不变性，其典型代表就是图结构数据。如图 3.15 所示，图像数据可以视为图结构数据的一个特例，图结构数据主要包括两个部分，一个图 G ，以及图中节点的属性 x 。为了实现图结构数据的特征提取，F. Scarselli 等人^[209] 提出了一种基于空间域的图卷积神经网络模型，该模型类似卷积神经网络，其对图中每个目标节点，将图 G 中该节点的邻居信息“聚集”到该目标节点上，从而提取节点特征 z ，如图 3.15 所示，因此该特征 z 既包含了节点的属性 x ，也包含了图 G 的结构。其后，J. Bruna 和 Y. LeCun 等人^[210,211] 提出了基于谱域的图卷积神经网络模型，其利用图傅里叶变换，将图结构数据变换到谱域后进行滤波，然后再将滤波后的谱域图特征变换回空间域。其二者各有优劣，(1) 空间域图卷积神经网络的计算效率更高，也更直观，但其本质上是卷积神经网络在图结构数据上的扩展，而谱域图卷积神经网络多了图傅里叶变换的操作，因此前者的可解释性比后者差。(2) 谱域图卷积神经网络由于需要对图 G 做特征分解，因此计算效率非常低，尤其是在图 G 的规模非常大时。此后，针对图结构数据特征提取应该采用哪种图卷积神经网络，由此延伸出两条研究主线，至今仍在争论。

其一针对空间域图卷积神经网络，其研究的主要问题是如何有效提取图 G 的结构特征，由此发展出许多图结构采样方法。比如，M. Niepert 等人^[212] 提出了一种尺寸固定的图卷积模板，主要解决节点邻居数量不固定情况下如何有效提取固定数量的邻居信息的问题。Z. Liu 等人^[213] 提出了一种自适应图卷积模板，其会自适应变化的邻居节点数量。P. Veličković 等人^[214] 等人提出了一种基于图注意力机制的图卷积神经网络，简言之，就是通过人工神经网络学习一个图 G ，然后进行图卷积计算，所以本质上还是基于空间域的图卷积神经网络。当然这个图 G 也可以用先验图结构知识初始化，若如此做，则该模型退化为基本的空间域图卷积神经网络。值得一提的是，该注意力机制启发了注意力网络^[200] 的设计。

其二针对谱域图卷积神经网络，其研究的主要问题是如何近似图傅里叶变换，从而降低图傅里叶变换的计算复杂度。M. Defferrard 等人^[215] 利用切比雪夫多项式近似图傅里叶滤波器，从而实现更快的滤波。T. N. Kipf 和 M. Welling^[216] 则更进一步地只选用了切比雪夫多项式的前两阶项，从而极大提高了算法效率，但缺点是只能构造两层图卷积层，如果层数过多，则会产生过平滑问题，也即只保留了图结构信息，而滤掉了节点属性信息。为了解决过平滑问题，M. Chen 等人^[217] 设计了具有残差结构的图卷积神经网络，从而将谱域图卷积神经网络推广至深层。此外还值得一提的是，该工作论证了空间域和谱域图卷积神经网络在过平滑问题上具有相似性，因而从这里开始，两条研究主线在过平滑问题上汇到一起。进而，E. Chien 等人^[218] 利用单项式基 $(1, x, x^2, \dots)$ 近似图傅里叶变换，M. He 等人^[219] 利用 Bernstein 多项式近似图傅里叶变换，二者均可缓解过平滑问题。总之，图卷积神经网络至今仍然是一个十分活跃的研究领域，其还有很多问题亟需解决，但这并不影响本文结论，也即利用图卷积神经网络可以实现图结构数据的特征提取。

3.2.4 通用多模态特征提取

正如前文所言，“复杂即简单”，复杂的事物往往由简单的事物组成。针对纷繁复杂的模态，设计专用的人工神经网络从而提取多模态数据中的结构特征，其固然是一种高效的解决思路，但其在模型的结构设计上却不具备灵活性。但如果直接采用前馈神经网络又会导致计算效率低下的问题，因此，能否在特征提取效率和模型结构设计的灵活性之间找到一个折中？为了实现这个目的，A. Vaswani 等人^[200] 提出了注意力网络 (Attention Network, Transformer) 模型。其思路是，专用多模态特征提取之所以灵活性差，是因为其采用的专用人工神经网络大多采用了隐式的特征提取方法，因此要提高灵活性，则不能采用隐式的特征提取。注意力网络基于自注意力 (Self-Attention) 机制，定义如下：

$$\begin{aligned} z &= \text{softmax} \left(\begin{bmatrix} \mathbf{q}_1 \mathbf{k}_1^\top & \mathbf{q}_1 \mathbf{k}_2^\top & \cdots & \mathbf{q}_1 \mathbf{k}_N^\top \\ \mathbf{q}_2 \mathbf{k}_1^\top & \mathbf{q}_2 \mathbf{k}_2^\top & \cdots & \mathbf{q}_2 \mathbf{k}_N^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_N \mathbf{k}_1^\top & \mathbf{q}_N \mathbf{k}_2^\top & \cdots & \mathbf{q}_N \mathbf{k}_N^\top \end{bmatrix} / \sqrt{d} \right) \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} \\ &= \text{softmax} (\mathbf{q} \mathbf{k}^\top / \sqrt{d}) \mathbf{v} \\ &= \text{softmax} ((\mathbf{x} \mathbf{w}_q)(\mathbf{x} \mathbf{w}_k)^\top / \sqrt{d}) (\mathbf{x} \mathbf{w}_v) \end{aligned} \quad (3.59)$$

这里观测样本 $\mathbf{x} \in \mathbb{R}^{N \times T}$, T 在这里表示每个变量的属性维度，在序列数据中， T 常用于表示序列长度，因此可以视其为变量属性维度而不失一般性。 $\mathbf{w}_q, \mathbf{w}_k, \mathbf{w}_v \in \mathbb{R}^{T \times T'}$ 分别是三个矩阵，将观测样本 \mathbf{x} 从属性维度 T 嵌入到属性维度 T' ，由此可得 $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{N \times T'}$, $\mathbf{q}_i \in \mathbb{R}^{1 \times T'}, i = 1, \dots, N$ 是 \mathbf{q} 的第 i 个变量的嵌入特征，此对 \mathbf{k} 和 \mathbf{v} 同理。此外，这里 d 表示缩放因子，一般 $d = T'$ 。 $\text{softmax}(\cdot)$ 函数会将矩阵 $\mathbf{q} \mathbf{k}^\top / \sqrt{d}$ 归一化，从而每一行元素之和值为一，也即得到矩阵 A ，且 $A_{ij} = e^{\mathbf{q}_i \mathbf{k}_j^\top / \sqrt{d}} / \sum_{j'=1}^N e^{\mathbf{q}_i \mathbf{k}_{j'}^\top / \sqrt{d}}$ 。

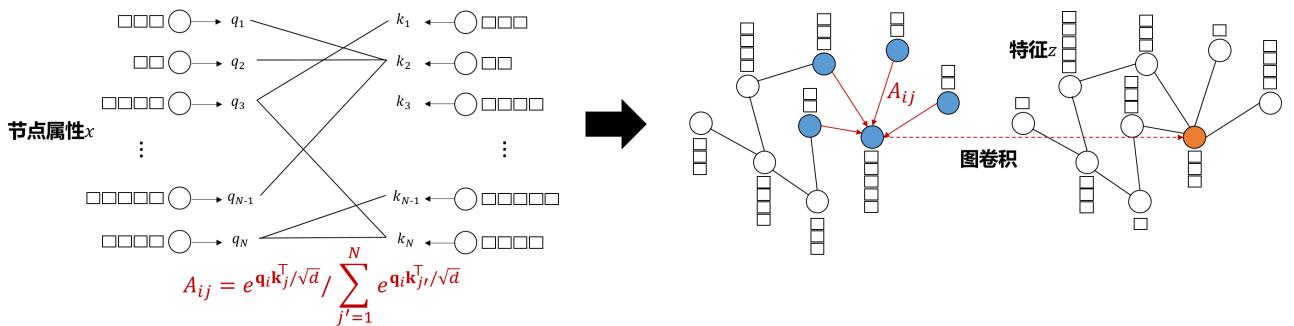


图 3.16: 自注意力机制示意图

因此实际上，注意力网络是从观测样本 x 自身先提取一个图 G 特征，然后再进行图卷积计算，其原

理等价于没有先验图 \mathcal{G} 的图注意力卷积神经网络^[214], 如图 3.16 所示。因此, 自注意力机制的“自”也有自适应(adaptive)的含义, 指的是图 \mathcal{G} 可自适应地得到, 图 \mathcal{G} 的邻接矩阵值 A_{ij} 由两个变量的嵌入特征的相关性得到, 这种相关性被定义为两个变量的属性在嵌入的特征空间中的内积, 此和再生核希尔伯特空间^[152]的原理十分相似(见本文章节 3.1.1)。此外, 文献^[200]提出的模型有一个使用更加广泛的名字, 也即 Transformer 模型。Transformer 模型会利用公式 (3.59) 提取多个特征 $z^{(1)}, \dots, z^{(M)} \in \mathbb{R}^{N \times T'}$, 之后将其拼接起来, 再接一个前馈神经网络层以加强其非线性表示的能力, 从而得到基于多头自注意力机制提取到的特征 z_{heads} , 也即

$$z_{\text{heads}} = \text{ReLU}([z^{(1)}, \dots, z^{(M)}] \mathbf{w}_0) \mathbf{w}_1 \quad (3.60)$$

这里 $\text{ReLU}(x) = \max(0, x)$, $\mathbf{w}_0 \in \mathbb{R}^{MT' \times T''}$, $\mathbf{w}_1 \in \mathbb{R}^{T'' \times T'''}$, T'' 和 T''' 是前馈神经网络层的神经元数。因此 Transformer 是多头注意力网络, 与基本的注意力网络略有区别, 但其仍然是以注意力网络为基础的。此外值得一提的是, 设计多头自注意力机制的本意是模仿卷积神经网络, 其利用多个卷积模板对同一个输入提取多样的特征, 而多头自注意力机制也利用多个“头”提取多样的特征。然而也有研究表明, 尽管采用了多头自注意力机制, 但多“头”学习到的图 \mathcal{G} 连边都十分密集, 且各层学习到的图 \mathcal{G} 区别不大^[219, 220], 因此是否有必要设置多头自注意力机制仍有争议。

于是, 针对其通用性的问题, J. Devlin 等人^[221]提出 BERT (Bidirectional Encoder Representations from Transformers) 模型用于处理自然语言。A. Dosovitskiy 等人^[222]在 BERT 的基础上提出 ViT (Vision Transformer) 模型用于处理视觉图像, 其将高分辨率图像切分成多个 16×16 的图像块(patch), 然后连同各个块之间的位置信息一同输入进 BERT 的变种模型, 从而将 BERT 模型泛用到计算机图像领域。ViT 模型的提出首次使得 Transformer 模型在图像识别任务上超过传统的卷积神经网络模型。其后, Z. Liu 等人^[223]进一步提出具有可变尺寸图像块的 ViT 变种模型。但值得一提的是, 图像块的切分方法十分影响 Transformer 模型的性能, 因此很难认为 ViT 模型的高性能完全是因为 Transformer, 而不是来自于一些更加技术性的图像块切分操作。更有甚者, A. Trockman 和 J. Z. Kolter^[224]认为完全依靠图像块划分和前馈神经网络就可以实现高性能的通用图像特征提取。

类似观点还存在于时间序列特征提取问题中, H. Zhou 等人^[225]提出基于稀疏自注意力机制的 Informer 模型用于处理时间序列特征提取和预测长时间序列。H. Wu 等人^[226]等人提出了基于自相关性机制的 Autoformer 模型, 简言之, 其将时间序列中的趋势性和季节性特征分解出来, 然后同序列的位置信息一同嵌入进 Transformer 模型, 因此也是 Transformer 模型一个变种。T. Zhou 等人^[227]提出 FEDformer 模型, 其将时间序列转换到频域, 然后用 Transformer 模型提取时间序列的频域特征。Y. Zhang 和 J. Yan^[228]提出 crossformer 模型用于提取多变量时间序列的变量间交互特征。Y. Liu 等人^[229]提出 iTransformer 模型, 基本的 Transformer 模型会处理多变量在同时刻的特征, 而 iTransformer 模型实际上只处理单变量的数据, 从而就可以获得更好的预测性能。同类观点还有, A. Zeng 等人^[230]提出 DLinear 模型, 其仅用单层的前馈神经网络, 结合时间序列中的趋势性和季节性特征嵌入, 就可以实现比 Informer 和 Autoformer 更好的长时间序列预测性能。S. Wang 等人^[231]提出 TimeMixer 模型, 其利用多层前馈神经网络, 并结合时间序列中的趋势性和季节性特征嵌入, 实现了与 iTransformer 模型几乎同等的预测性能。综合以上文献可得结论, (1) Transformer 的各种变种模型在同等实验条件下, 其序列特征提取和预测性能要好于传统循环神经网络。(2) 是否基于 Transformer 模型对模型的预测性能影响不大, 比如 TimeMixer 模型和 iTransformer 模型在预测性能上相差不大。DLinear 甚至超过 Autoformer 模型和 Informer 模型。此外还值得注意的是, NLinear 模型^[230]即是 DLinear 模型, 但其不采用时间序列中的趋势性和季节性特征嵌入, 在多个数据集上均达到了不输 Transformer 的性能。这与文献^[232-234]结论相似, 这些文献均认为位置嵌入对 Transformer 的影响十分微妙。如果 Transformer 要实现其通用性, 那么就不应该针对不同模态的场景数据考虑过多技术性的位置嵌入。因此, 如果不将多模态数据中的结构信息作为位置信息嵌入 Transformer 模型, 那么 Transformer 模型实际上近似于前馈神经网络。而此前的 Transformer 及其变种模型在多模态特征提取上的性能优势, 可能更多来自于其一些技巧性的操作和巨大的参数量。

总之，整理以上论述可以得到以下观点，(1) 首先，利用 Transformer 模型可以实现通用的多模态特征提取。(2) Transformer 模型的多头自注意力机制可能并无必要，真正重要的是“自注意力机制”，而不是“多头”。(3) Transformer 需要将多模态数据中的结构信息作为位置信息嵌入，但在实际使用时，这种位置嵌入具有一定的技术性，而技术和通用性是矛盾的。(4) 在不考虑任何技术性操作时，Transformer 模型和前馈神经网络一样，其性能更多取决于其参数量。

3.2.5 基于生成式大模型的多模态场景数据生成

生成式大模型（大人工智能模型，Foundation Models^[25]）是多模态场景数据生成不断发展的一个必然趋势，其主要特点就是模型参数量极大，以及用于模型训练的数据量极大，从而使得生成式大模型具备了一些“小”生成模型所不具备的特性。一方面，随着多模态场景数据量不断增加，其需要的编码器和解码器数量也随之不断增加（见图 3.11），将如此之多的人工神经网络组合起来，其总的参数量也就必然地变得非常大。另一方面，多模态场景数据生成的规模理论 (Scaling Law^[235,236]) 终于被发现，其认为基于人工神经网络的生成模型性能主要依赖于模型的参数量和数据量，由此引起了对具有更大参数量和数据量的生成模型的研究热潮，最终诞生了生成式大模型。就如同对待摩尔定律一样，此时的人们相信更大的模型参数量和数据量会带来更加优秀的模型性能。当然，其结果没有辜负人们的期待，自以 ChatGPT 为代表的商用大语言模型诞生以来，各类生成式大模型如雨后春笋般出现，如图 3.17 所示。其功能也不仅仅限于对话，而是逐渐扩展至图像生成、代码生成、机器人指令生成等等，应用场景越来越多，至今仍在飞速发展中。

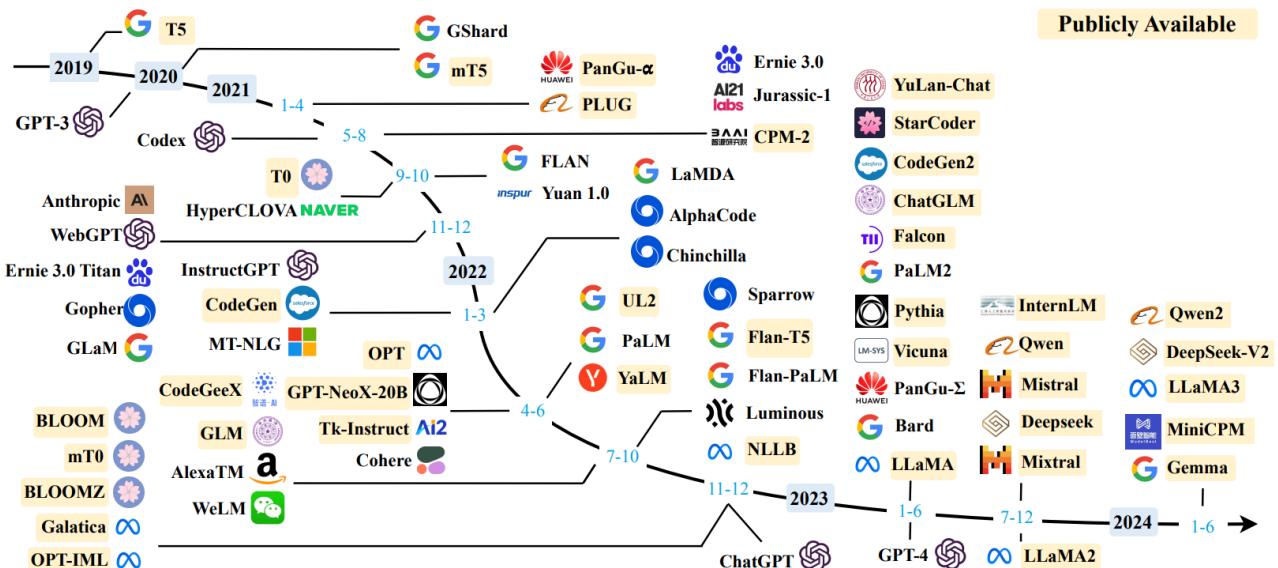


图 3.17: 2019 年 -2024 年生成式大模型发展时间线（出自文献^[237] 中 Figure 3）

由于生成式大模型具有极高的商用特性，因而其实现难度往往不在技术层面，也即其需要考虑的更多是，如何让生成式大模型的商用收益高于模型训练的成本和数据获取的成本，只有如此，才能使得一个商用的生成式大模型随着用户数据的不断输入，而不断地迭代更新模型参数，从而获得更好的性能，以获得更大的收益。而不得不说，坚信唯技术论的研究人员很容易忽略这一点。当然，本文只在这里讨论生成式大模型在多模态场景数据生成方面所采用的基础技术，其中就包括预训练 (pre-train) 和微调 (fine-tune) 技术、提示 (prompt) 技术和混合专家 (Mixture of Experts) 模型等等。

首先是预训练和微调技术，其二者最早出现于 GPT-1 (Generative Pre-trained Transformer-1)^[238] 模型。简言之，其首先将生成模型在大规模多模态场景数据上进行广泛的预训练，然后针对特定场景下收集的新数据微调模型参数，从而实现使得模型在特定场景任务上实现更好的泛化能力。如此做的直接原因是，针对特定场景下收集的新数据，如果每次都构建新的生成模型，然后初始化模型参数并训练，那

么其训练成本会变得非常大。而预训练和微调技术则保证了，可以利用已经训练好的生成模型，仅经过少量的参数微调，就可以使得模型在特定场景下获得同等甚至更好的泛化性。此外，其还有另一个更深层次的原因，实际上预训练和微调技术隐含假设了，新收集的场景数据和用于预训练的场景数据之间，在特征上十分相似甚至相同，那么实际上新收集的场景数据并不“新”，因而其不会改变预训练的生成模型所学习到的场景数据分布，所以并不需要重新训练一个生成模型。因此，直到 GPT-3 模型^[239]，其使用了 45TB（过滤筛选后共 570GB）的纯文本数据用于预训练，其参数量也达到了 1750 亿参数，这使得 GPT-3 在诸多文本场景下仅经过极少量的参数微调就可以实现很高的文本生成泛化性能，并且这一规律在 GPT-4^[240] 模型中的多模态场景下同样适用。

然后是提示技术，该技术最早出现于 GPT-2^[241] 模型，后在 GPT-3 模型中得到广泛的使用和测试，其利用预训练的模型，仅通过输入文本提示，就可以输出相应的文本，从而可以解决少样本学习、单样本学习和零样本学习等问题。在多模态场景数据生成中，提示技术的主要思想就是让生成模型根据输入的提示词定向地生成场景数据，其输入的提示词可以是上下文文本，也可以是图像，或者其他，而并不一定只能输入文本。总之，这种提示技术实际上是一种条件场景数据生成（见本文章节 3.2.1），其关键之处在于处理提示词和目标的生成场景数据之间的关联性。

然后是混合专家模型，准确地说，其是一种生成式大模型的设计架构，简言之，就是一个生成模型性能不够，那就搭建更多的生成模型，让其共同完成一个生成任务^[242,243]，如图 3.18 所示。这种生成式大模型设计架构广泛存在于 Switch Transformer 模型^[244]，其用于自然语言处理。还有 V-MoE 模型^[245]，其用于计算机视觉。还有华为 M6 模型^[246]，其用于处理多模态数据，而值得一提的是，华为 M6 模型是因为在设计大模型时考虑到了计算资源有限等问题，所以选用了混合专家模型。在混合专家模型中，尽管有多个专家模型共同参与生成任务，但实际上不是所有专家模型都参与，而是只有部分专家模型参与，混合专家模型会采用门控机制以控制专家模型的稀疏表达^[243]，定义如下：

$$z = \sum_{i \in \text{TopK}} P_i \times E_i(x), \quad (3.61)$$

这里 $E_i(x)$ 是第 i 个专家模型的输出，全部专家共 M 个， $\sigma_i(x)$ 会给第 i 个专家模型打分， TopK 是一个包含了前 k 个得分最高的专家模型的集合， $M \gg k$ ，并将其得分归一化，从而得到 $[P_{i_1}, \dots, P_{i_k}] = \text{softmax}([\sigma_{i_1}(x), \dots, \sigma_{i_k}(x)])$ ，最终得到 k 个专家模型输出的加权平均 z ，如图 3.18 所示。混合专家模型实际类似一种在多模态间共享模型的机制，如前文所述，在多模态场景下，一个生成式大模型中会有多个子模型分别针对不同模态进行特征提取和生成，但这些子模型并不总是能得到使用，因此这种子模型与模态的一对一关系十分浪费计算资源。因而可以将其构造成混合专家模型，使得所有子模型都可以被视为专家，共同处理所有模态的输入场景数据，而此外只需要针对不同模态设计相应特征嵌入机制即可。

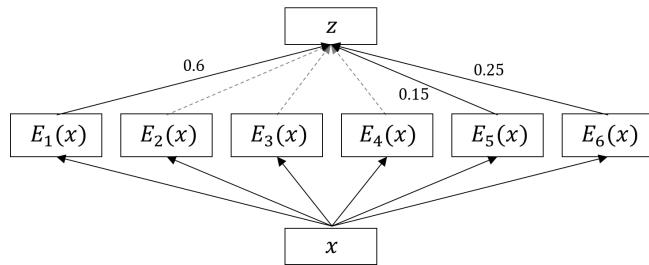


图 3.18：混合专家模型示意图

总之，相比于前述的生成模型理论，基于生成式大模型的多模态场景数据生成更偏向于技术应用，甚至已经进入了商业的范畴。仅仅依靠上述技术也只能初步的搭建一个生成式大模型，但要让其真正的能够运行起来，还需要特定场景下的领域知识，此外还要能在具体的场景下实现稳定的数据采集，以及实现快速的程序响应等等。而最重要的，还是要有成熟的商业模式，而这些都不在本文的讨论范围内，本文旨在论述如何利用生成模型实现场景数据生成。此外，生成式大模型目前仍在飞速发展中，其理论技

术日新月异，因此其使用的很多技术能否经得起时间的考验还不能妄下定论。

3.3 基于跨时空模态生成对抗网络的场景数据生成框架

本章节和下一章节会提供两个基于生成对抗网络的跨时空模态场景数据生成技术应用，作为多模态场景数据生成的数值案例。本章节主要翻译自作者论文^[247]，并在其基础上重新整理与复杂系统场景数据生成有关的观点和结论。由于该论文与本文在符号定义上可能会有不同，本文会尽量保证符号定义上的一致性。该论文^[247]主要研究新能源电网中的场景数据生成问题，其研究目的是生成场景数据，从而估计新能源电网中的不确定性，为新能源电网的安全控制、优化与经济调度提供决策边界。随着目前新能源电网的不断发展，新能源的能源占比不断攀升，但同时也给传统电网带来了许多不确定性问题。不同于传统电网，新能源电网中的不确定性非常高，尤其是风能和太阳能，其能源供应的稳定性十分依赖于自然环境，而自然环境的运行又有极高的不确定性，这部分不确定性会传导至传统电网中，从而使传统电网变得脆弱^[59,60]。因此，需要利用场景数据生成技术估计这部分来自于新能源的不确定性，从而针对其不确定性优化决策配置，进而提高电网整体的稳定性。因此本文提出一种基于跨时空模态生成对抗网络的场景数据生成框架，主要针对新能源电网中的时空模态数据，实现跨模态的场景数据生成。本文将其模型与算法在风能和太阳能数据集上进行了测试，从而达到 SOTA (state-of-the-art) 水平。本文还设计了一个随机日前经济调度技术应用场景，从而验证了该论文方法在具体的电网安全应用场景中的有效性。

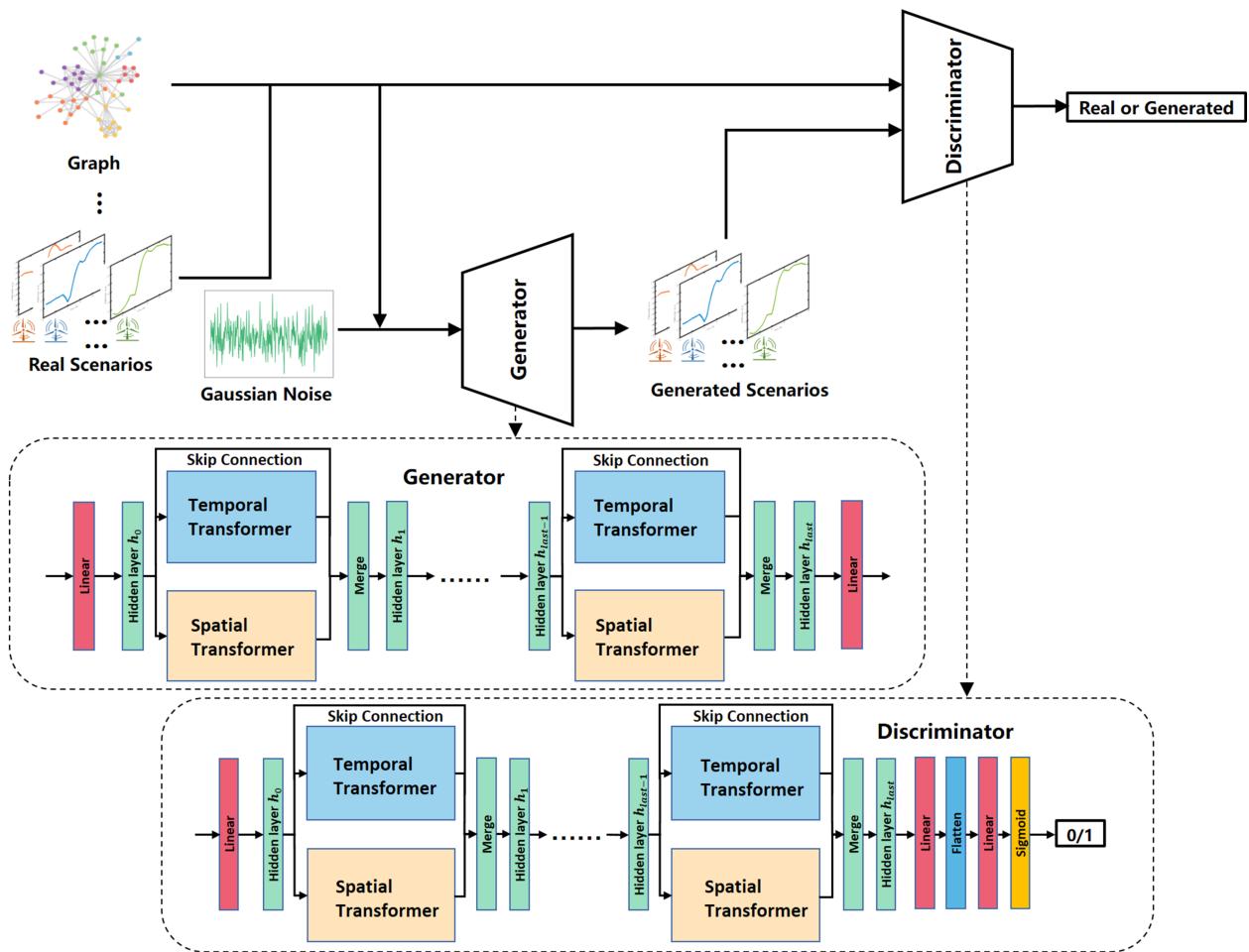


图 3.19: 基于跨时空模态生成对抗网络的场景数据生成框架（出自文献^[247] 中 Figure 1）。

3.3.1 跨时空模态生成对抗网络原理

本文设计了一种跨时空模态的生成对抗网络框架，其原理如图 3.19 所示，该生成对抗网络的基本运行原理与图 3.7 一致。这里时间模态主要是指含有时序特性的数据，空间模态主要是指图结构数据，图结构数据可以描述电网中不同电力单元之间的交互性。由于前文针对多模态生成对抗网络已经有了详尽的公式推导和示意图展示，可参考章节 3.2.1，因此本文这里从公式 (3.17) 和公式 (3.48) 之后直接给出 min-max 博弈的目标函数，如下：

$$\min_g \max_{f: \|f\|_L \leq 1} \mathbb{E}_{(\mathbf{x}, \mathcal{G}) \sim P_{\text{data}}(\mathbf{x}, \mathcal{G})} [f(\mathbf{x}, \mathcal{G})] - \mathbb{E}_{\mathbf{x} \sim P_g(\mathbf{x}|\mathcal{G}), \mathcal{G} \sim P_{\text{data}}(\mathcal{G})} [f(\mathbf{x}, \mathcal{G})], \quad (3.62)$$

这里 \mathcal{G} 是包含 N 个节点的图结构数据， $\mathbf{x} \in \mathbb{R}^{N \times T}$ 是与图 \mathcal{G} 中节点对应的长度为 T 的多变量时间序列数据。此外， f 是判别器，满足 1-Lipschitz 约束， g 是生成器，用于生成场景数据。于是从公式 (3.62) 中解出生成器和判别器各自优化的目标函数，如下：

$$\begin{aligned} L_f &= f(\hat{\mathbf{x}}, \mathcal{G}) - f(\mathbf{x}, \mathcal{G}) + \lambda (\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}}, \mathcal{G})\| - 1)^2, \\ L_g &= -f(\hat{\mathbf{x}}, \mathcal{G}) + \frac{\mu}{CI(I_{i,t}=1)} \sum_{i,t} I_{i,t} \|\hat{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}\|^2, \end{aligned} \quad (3.63)$$

这里 $\hat{\mathbf{x}} \sim P_g(\mathbf{x}|\mathcal{G})$ 表示生成的场景数据，用于区分 \mathbf{x} 。 $(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}}, \mathcal{G})\| - 1)^2$ 是梯度惩罚项，原理可参考公式 (3.18)。 $CI(I_{i,t}=1)$ 是条件计数函数，会计算出 \mathbf{x} 中的符合 $I_{i,t} = 1$ 条件的元素个数。 $I_{i,t}$ 是指示函数，表示矩阵 \mathbf{x} 中位置 (i, t) 的对应元素未缺失， $I_{i,t} = 0$ 则表示缺失。本文跨时空模态生成对抗网络设计时考虑了时间序列 \mathbf{x} 中可能存在的数据缺失问题，所以通过该惩罚项降低重构误差。此外， λ 和 μ 是惩罚项的权重参数。

3.3.2 时空 Transformer 原理

为了参数化公式 (3.62) 中的生成器 g 和判别器 f ，本文设计了时空 Transformer 模型用于时空模态数据特征提取，如图 3.19 所示。该时空 Transformer 模型由多个时空 Transformer 模块组合而成，每个模块又由一个时间 Transformer 和一个空间 Transformer 组成，如图 3.20 所示，而不同的模块之间采用密集连接结构提高特征提取效率。

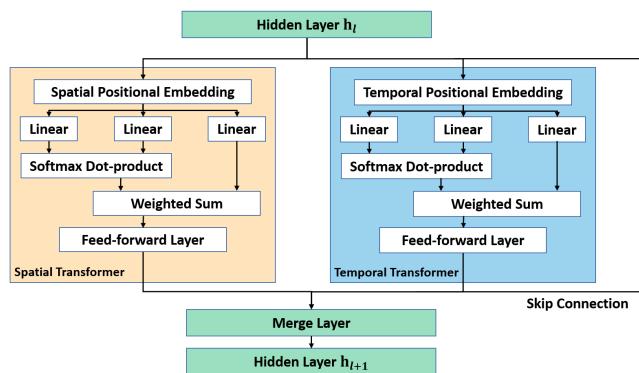


图 3.20：第 $l + 1$ 层的时空 Transformer 模块（出自文献^[247] 中 Figure 2）

对于第 $l + 1$ 层的时空 Transformer 模块，如图 3.20 所示，其输入是第 l 层的输出 $\mathbf{h}_l \in \mathbb{R}^{N \times T \times C}$ ，输出是 $\mathbf{h}_{l+1} \in \mathbb{R}^{N \times T \times C}$ ，这里 C 是隐藏层的嵌入维度。针对空间 Transformer 模块，进行如下计算：

$$\mathbf{x}^S = \mathbf{h}_l + E^S, \quad (3.64)$$

这里 $E^S = A\mathbf{w}_A^S$ ， $A \in \mathbb{R}^{N \times N}$ 是由图结构数据 \mathcal{G} 导出的邻接矩阵， $\mathbf{w}_A^S \in \mathbb{R}^{N \times C}$ ，然后将 E^S 扩展到

$\mathbb{R}^{N \times T \times C}$ 并加到 \mathbf{h}_l 上，从而实现空间位置嵌入。于是得到

$$\begin{aligned}\mathbf{q}^S &= \mathbf{x}^S \mathbf{w}_q^S, \\ \mathbf{k}^S &= \mathbf{x}^S \mathbf{w}_k^S, \\ \mathbf{v}^S &= \mathbf{x}^S \mathbf{w}_v^S,\end{aligned}\tag{3.65}$$

这里 $\mathbf{w}_q^S, \mathbf{w}_k^S, \mathbf{w}_v^S \in \mathbb{R}^{C \times C}$ 是图 3.20 中线性层参数，于是得到

$$M^S = \text{softmax}\left(\frac{\mathbf{q}^S(\mathbf{k}^S)^\top}{\sqrt{d^S}}\right),\tag{3.66}$$

这里 $\mathbf{q}^S(\mathbf{k}^S)^\top$ 表示在维度 N 做张量积， d^S 是缩放因子，从而 $M^S \in \mathbb{R}^{T \times T \times C}$ 。于是得到

$$U^S = M^S \mathbf{v}^S,\tag{3.67}$$

这里表示在维度 T 做张量积，从而 $U^S \in \mathbb{R}^{N \times T \times C}$ 。于是得到

$$Y^S = \text{ReLU}(\text{ReLU}(U^S \mathbf{w}_0^S) \mathbf{w}_1^S) \mathbf{w}_2^S,\tag{3.68}$$

这里 $\mathbf{w}_0^S, \mathbf{w}_1^S, \mathbf{w}_2^S \in \mathbb{R}^{C \times C}$ 是图 3.20 中前馈神经网络层参数，通过 $\text{ReLU}(\cdot)$ 函数激活。

针对时间 Transformer 模块，进行如下计算：

$$\mathbf{x}^T = \mathbf{h}_l + E^T,\tag{3.69}$$

这里 $E^T = \text{Embedding}([1, \dots, T]) \in \mathbb{R}^{T \times C}$ ，其中 $[1, \dots, T] \in \mathbb{R}^{1 \times T}$ 表示时间序列下标。该公式将 E^T 扩展到 $\mathbb{R}^{N \times T \times C}$ ，然后加到 \mathbf{h}_l 上，从而实现时序位置嵌入。于是得到

$$\begin{aligned}\mathbf{q}^T &= \mathbf{x}^T \mathbf{w}_q^T, \\ \mathbf{k}^T &= \mathbf{x}^T \mathbf{w}_k^T, \\ \mathbf{v}^T &= \mathbf{x}^T \mathbf{w}_v^T,\end{aligned}\tag{3.70}$$

这里 $\mathbf{w}_q^T, \mathbf{w}_k^T, \mathbf{w}_v^T \in \mathbb{R}^{C \times C}$ 是图 3.20 中线性层参数，于是得到

$$M^T = \text{softmax}\left(\frac{\mathbf{q}^T(\mathbf{k}^T)^\top}{\sqrt{d^T}}\right),\tag{3.71}$$

这里 $\mathbf{q}^T(\mathbf{k}^T)^\top$ 表示在维度 T 做张量积， d^T 是缩放因子，从而 $M^T \in \mathbb{R}^{N \times N \times C}$ 。于是得到

$$U^T = M^T \mathbf{v}^T,\tag{3.72}$$

这里表示在维度 N 做张量积，从而 $U^T \in \mathbb{R}^{N \times T \times C}$ 。于是得到

$$Y^T = \text{ReLU}(\text{ReLU}(U^T \mathbf{w}_0^T) \mathbf{w}_1^T) \mathbf{w}_2^T,\tag{3.73}$$

这里 $\mathbf{w}_0^T, \mathbf{w}_1^T, \mathbf{w}_2^T \in \mathbb{R}^{C \times C}$ 是图 3.20 中前馈神经网络层参数，通过 $\text{ReLU}(\cdot)$ 函数激活。

最后，将 Y^S, Y^T 和 \mathbf{h}_l 拼接，通过一个融合层 (merge layer) 实现时空特征融合，如下：

$$\mathbf{h}_{l+1} = \text{ReLU}([Y^S, Y^T, \mathbf{h}_l] \mathbf{w}_m),\tag{3.74}$$

这里 $\mathbf{w}_m \in \mathbb{R}^{3C \times C}$ 。最终得到第 $l+1$ 层的时空 Transformer 模块的输出 $\mathbf{h}_{l+1} \in \mathbb{R}^{N \times T \times C}$ 。

此外特别说明，在图 3.62 中，判别器和生成器均在一开始通过一个线性层将输入场景数据 $\mathbf{x} \in \mathbb{R}^{N \times T}$ 扩展到 $\mathbb{R}^{N \times T \times C}$ 。并且判别器在最后通过

$$O_{\text{pred}} = \text{Sigmoid}(\text{Flatten}(\mathbf{h}_{\text{last}} \mathbf{w}_{d_0}) \mathbf{w}_{d_1}),\tag{3.75}$$

输出标签 0 或 1 的概率。这里 \mathbf{h}_{last} 表示最后一个时空 Transformer 模块的输出特征， \mathbf{w}_{d_0} 和 \mathbf{w}_{d_1} 是对应的前馈神经网络层参数。

3.3.3 跨时空模态场景数据生成算法

本文设计用于训练跨时空模态生成对抗网络的算法，并最终用于场景数据生成，如算法 1 所示，简言之，生成器 g 和判别器 f 交替训练 n_{epoch} 个周期，当参数收敛，则可以用生成器 g 生成场景数据。该算法的时间复杂度为 $\mathcal{O}((NT^2C^2 + N^2TC^2)(l_g n_g + 2l_f n_f)n_{\text{epoch}})$ ，这里 l_g 和 l_f 分别是生成器 g 和判别器 f 中时空 Transformer 模块的数量（层数），而 n_g 和 n_f 分别是生成器 g 和判别器 f 在一个周期内迭代优化其参数的次数。但实际上，由于 $n_{\text{epoch}} \gg \max(N, T, C)$ ，且 $\min(N, T, C) \gg 4 > \max(l_g, l_f, n_g, n_f)$ ，因此，实际上其时间复杂度可以近似为 $\mathcal{O}((NT^2C^2 + N^2TC^2)n_{\text{epoch}})$ ，可以发现算法 1 的时间复杂度主要取决于常数 N, T 和 C ，以及迭代周期 n_{epoch} 。

算法 1：跨时空模态场景数据生成

Input: 场景数据 \mathbf{x} 和 \mathcal{G} ，参数化生成器 g 和判别器 f ；

Output: 训练好的生成器 g ；

```

1 用 Adam 优化器初始化生成器  $g$  和判别器  $f$  的参数，学习率为  $\alpha$ ;
2 while 未到迭代周期  $n_{\text{epoch}}$  do
3   for  $k = 1, 2, \dots, n_f$  do
4     在  $\mathbf{x}$  中选择一批样本;
5     采样  $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$  和  $\epsilon \sim U(0, 1)$ ;
6      $\hat{\mathbf{x}} \leftarrow g(\mathbf{x}, \mathcal{G}, \mathbf{z})$ ;
7      $\tilde{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon) \hat{\mathbf{x}}$ ;
8      $L_f \leftarrow f(\hat{\mathbf{x}}, \mathcal{G}) - f(\mathbf{x}, \mathcal{G}) + \lambda (\|\nabla_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}}, \mathcal{G})\| - 1)^2$ ;
9      $f \leftarrow \text{Adam}(\nabla_f L_f)$ ;
10    for  $k = 1, 2, \dots, n_g$  do
11      在  $\mathbf{x}$  中选择一批样本;
12      采样  $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ ;
13       $\hat{\mathbf{x}} \leftarrow g(\mathbf{x}, \mathcal{G}, \mathbf{z})$ ;
14       $L_g \leftarrow -f(\hat{\mathbf{x}}, \mathcal{G}) + \frac{\mu}{CI(I_{i,t}=1)} \sum_{i,t} I_{i,t} \|\hat{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}\|^2$ ;
15       $g \leftarrow \text{Adam}(\nabla_g L_g)$ ;
16  return 生成器  $g$ .

```

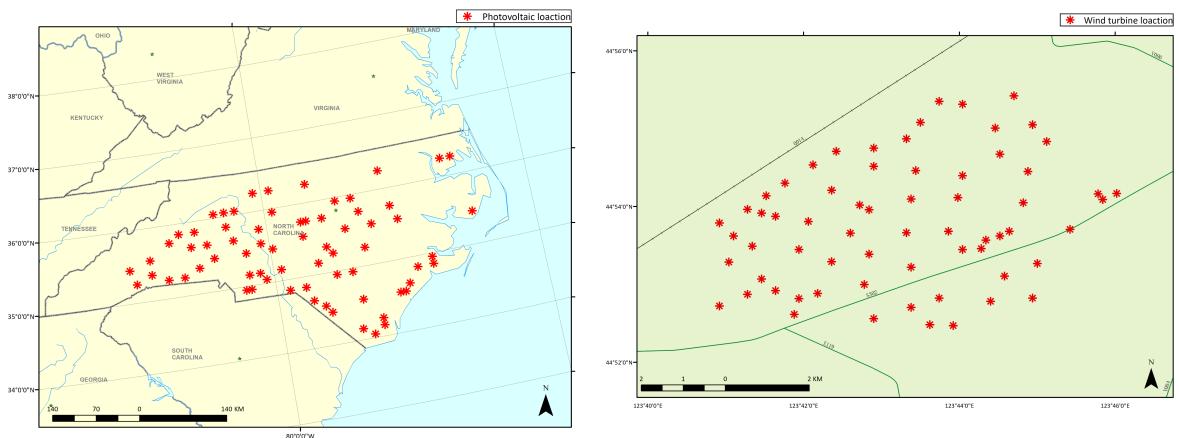
3.3.4 实验设置

本文在仿真数据集上和真实数据集上均做了相应的模型性能测试，其中光伏电站仿真数据采样自 National Renewable Energy Laboratory (NREL) 数据集^[248]，其包含了 2006 年美国北卡罗来纳州全年光伏发电仿真数据，该数据中的光伏电站数量为 69，时间粒度为 15 分钟，光伏电站容量为 10MW~134MW，其电站的空间分布位置如图 3.21(a) 所示。风电数据集为真实数据集，采样自中国吉林省一风能电厂，其包含了 66 个风机，容量均为 1500KW，持续时间 10 个月，其风机空间分布位置如图 3.21(b) 所示。

本文从数据集中随机选择样本，并将其划分成训练集 \mathbf{x}^{tr} 和测试集 \mathbf{x}^{te} ，本文将跨时空模态生成对抗网络模型在训练集上训练并生成场景样本，并试图让生成场景样本和测试集的数据分布相同。因此，本文采用 MMD 距离度量观测场景数据和生成场景数据之间差异，参考公式 (3.2) 可得

$$D_{\text{MMD}}(\hat{\mathbf{x}}, \mathbf{x}^{\text{te}}) = \frac{1}{T(T-1)} \sum_{m \neq n}^T [K(\hat{\mathbf{x}}_{:m}, \hat{\mathbf{x}}_{:n}) + K(\mathbf{x}_{:m}^{\text{te}}, \mathbf{x}_{:n}^{\text{te}})] - \frac{2}{T^2} \sum_{m=1}^T \sum_{n=1}^T K(\hat{\mathbf{x}}_{:m}, \mathbf{x}_{:n}^{\text{te}}), \quad (3.76)$$

这里 $K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$ 是再生核函数， $\hat{\mathbf{x}}$ 是生成的场景样本。本文将 MMD 距离归一化成 MMD 得分，



(a) 光伏电站空间布局范围 $34.05^{\circ}\text{N} \sim 36.25^{\circ}\text{N}, 76.15^{\circ}\text{W} \sim 82.65^{\circ}\text{W}$ (b) 风力电站空间布局范围 $44.875^{\circ}\text{N} \sim 44.924^{\circ}\text{N}, 123.682^{\circ}\text{E} \sim 123.767^{\circ}\text{E}$

图 3.21: 新能源电站空间布局 (出自文献^[247] 中 Figure 3)。

定义如下:

$$\text{MMD} = 1 - \frac{D_{\text{MMD}}(\hat{\mathbf{x}}, \mathbf{x}^{\text{te}}) - D_{\text{MMD}}(\mathbf{x}^{\text{tr}}, \mathbf{x}^{\text{te}})}{D_{\text{MMD}}(\mathbf{x}^{\text{tr}}, \mathbf{x}^{\text{te}})}, \quad (3.77)$$

此外, 为了评估时空模态在跨模态表示转换中的作用, 定义:

$$\begin{aligned} d_S &= \text{MMD}_{\text{cGAN}} - \text{MMD}_{\text{cGAN-T}}, \\ d_T &= \text{MMD}_{\text{cGAN}} - \text{MMD}_{\text{cGAN-S}}, \end{aligned} \quad (3.78)$$

进而定义时空模态各自的贡献度指标, 如下:

$$\begin{aligned} \text{con}_S &= \frac{d_S}{d_S + d_T} \times 100\%, \\ \text{con}_T &= \frac{d_T}{d_S + d_T} \times 100\%. \end{aligned} \quad (3.79)$$

此外, 由于本文跨时空模态生成对抗网络考虑了数据中可能存在的噪声, 见公式(3.63)中的重构误差项, 因此这里还要定义数据中的缺失率, 如下:

$$\eta = \frac{CI[I_{i,t} = 0]}{N \times T} \times 100\%, \quad (3.80)$$

此外, 本文跨时空模态生成对抗网络中, 图 \mathcal{G} 构造于数据集中的电站 GPS 位置, 本文利用球坐标计算两电站 i 和 j 之间的空间距离 d_{ij} , 于是可得图 \mathcal{G} 导出的邻接矩阵, 定义如下:

$$A_{ij} = \begin{cases} e^{-\frac{d_{ij}}{\sigma}}, & i \neq j, \\ 0, & i = j, \end{cases} \quad (3.81)$$

这里 σ 是规范化因子, 光伏数据集中 $\sigma = 100$, 风电数据集中 $\sigma = 1$ 。此外, 算法 1 中, n_f 和 n_g 在光伏数据集中分别设置为 3 和 2, 在风电数据集中分别设置为 2 和 2。 n_{epoch} 设置为 1000。学习率 α 设置为 0.0001。批大小 (batch size) 设置为 32。惩罚项权重 λ 和 μ 均设置为 10。此外, 生成器和判别器中的时空 Transformer 均为两层, 且嵌入特征维度 C 为 32, 时间序列长度 T 为 96。

此外, 为了比较本文跨时空模态生成对抗网络 (cGAN) 和其它基准模型的性能, 本文引入如下基准模型:

1. Autoregressive Integrated Moving Average (ARIMA)^[249]: 一种广泛使用的时间序列分析模型, 主要用于建模和预测时间序列, 模型中有三个通用参数, 也即自回归项的数量 P 、差分次数 D 和滑动平均项的数量 Q , 这里分别设置为 4、0 和 0。

2. 长短期记忆 (Long Short-Term Memory, LSTM) 网络^[250]: 一种针对时间序列隐式特征提取的人工神经网络模型, 如图 3.12(b)所示, 这里隐藏层神经元数设为电站数 N , 共两层隐藏层。
3. 门控循环单元 (Gated Recurrent Unit, GRU) 网络^[250]: LSTM 模型的一个变种, 如图 3.12(b)所示, 其特点是模型参数规模比 LSTM 模型更加轻量, 这里隐藏层神经元数设为电站数 N , 共两层隐藏层。
4. 变分自编码器 (Variational Autoencoder, VAE)^[251]: 一种生成模型, 通过最大化证据下界以最大化似然概率, 从而最优化模型参数, 可参考本文章节 3.1.2。其中编码器和解码器各采用一个三层的前馈神经网络参数化, 每层神经元数均为 2048, 激活函数为 Leaky ReLU, 也即 $\text{Leaky ReLU}(x) = \max(0.2x, x)$ 。
5. Wasserstein GAN (WGAN)^[59]: 生成对抗网络的一个变种, 其在生成对抗网络的基础上采用 Wasserstein 距离替换了 KL 散度, 从而优化模型训练过程中的梯度求解, 可参考本文章节 3.1.3。其中的生成器和判别器各采用一个三层的前馈神经网络参数化, 参数设置与 VAE 相同。
6. C-StyleGAN2-SE^[252]: 生成对抗网络的一个变种, 特点是针对电网中多样式的场景数据进行条件生成。其采用原文献中默认设置的模型参数。

在基准模型的基础上, 本文共设置两个实验, 其一测试本文跨时空模态生成对抗网络和其它基准模型的场景数据生成性能, 也即 MMD 得分。其二在一个风力发电并网的仿真模型上测试生成的场景数据对随机日前经济调度的作用, 从而说明基于本文模型的场景数据生成技术的实际应用价值。

3.3.5 实验结果

本文在具有不同缺失率的数据集上进行了场景数据生成测试, NREL 光伏电站仿真数据集测试结果如表 3.2 所示, 吉林省风电数据集测试结果如表 3.3 所示。总结如下: (1) 本文跨时空模态生成对抗网络 (cGAN) 在两个数据集上, 以及在不同的数据缺失率下, 其性能评估 (MMD 得分) 均达到 SOTA 水平。(2) 在模型中, 时间模态和空间模态对场景数据生成的性能提升效果不同, 空间模态对性能影响更大。

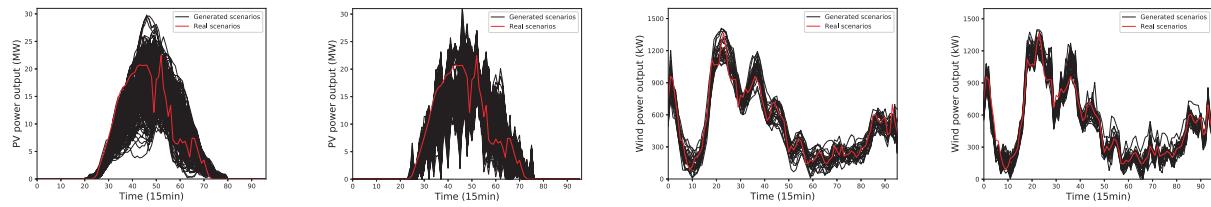
从表 3.2 和表 3.3 可以看到 C-StyleGAN2-SE 模型的 MMD 得分仅次于本文模型, 为了更直观地展示其二者在时空模态场景数据生成上的区别, 本文绘制图 3.22, 其展示了两个模型在光伏和风电场景数据生成结果上的区别。可以直观看出, 二者均可以生成一组形似真实场景的数据, 且具备一定的多样性。本文将其二者生成的场景数据整合成直方图以展示数据分布, 如图 3.23 所示, 可以发现本文模型的生成场景数据分布跟接近真实场景数据分布, 在风电数据集上的对比更加明显。

表 3.2: NREL 光伏电站仿真数据集场景数据生成测试结果

缺失率/MMD 得分/模型	ARIMA	LSTM	GRU	VAE	WGAN	C-StyleGAN2-SE	cGAN-S	cGAN-T	cGAN
0%	0.519	0.881	0.890	0.727	0.786	0.899	0.892 ($con_S : 87.9\%$)	0.823 ($con_T : 12.1\%$)	0.903
5%	0.511	0.877	0.883	0.692	0.782	0.893	0.890 ($con_S : 95.3\%$)	0.813 ($con_T : 4.7\%$)	0.894
10%	0.510	0.872	0.875	0.691	0.781	0.888	0.886 ($con_S : 93.2\%$)	0.810 ($con_T : 6.8\%$)	0.892
15%	0.505	0.871	0.869	0.682	0.775	0.870	0.883 ($con_S : 95.1\%$)	0.809 ($con_T : 4.9\%$)	0.887
20%	0.504	0.866	0.868	0.673	0.772	0.868	0.879 ($con_S : 97.4\%$)	0.805 ($con_T : 2.6\%$)	0.881

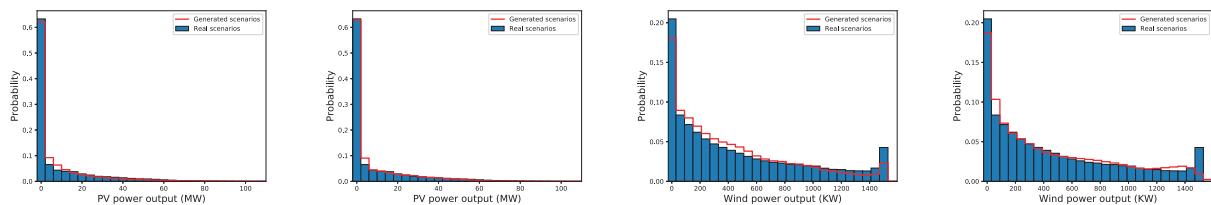
表 3.3: 吉林省风电数据集场景数据生成测试结果

缺失率/MMD 得分/模型	ARIMA	LSTM	GRU	VAE	WGAN	C-StyleGAN2-SE	cGAN-S	cGAN-T	cGAN
0%	0.621	0.919	0.932	0.784	0.839	0.929	0.933 ($cons_S : 93.5\%$)	0.852 ($cont_T : 6.5\%$)	0.939
5%	0.620	0.898	0.914	0.701	0.798	0.914	0.930 ($cons_S : 98.8\%$)	0.849 ($cont_T : 1.2\%$)	0.931
10%	0.617	0.851	0.895	0.682	0.782	0.903	0.925 ($cons_S : 95.7\%$)	0.840 ($cont_T : 4.3\%$)	0.929
15%	0.615	0.849	0.882	0.689	0.751	0.881	0.909 ($cons_S : 91.9\%$)	0.837 ($cont_T : 8.1\%$)	0.916
20%	0.612	0.824	0.859	0.677	0.742	0.874	0.890 ($cons_S : 98.4\%$)	0.831 ($cont_T : 1.6\%$)	0.891



(a) C-StyleGAN2-SE 生成的光伏场景数据 (b) 本文模型生成的光伏场景数据 (c) C-StyleGAN2-SE 生成的风电场景数据 (d) 本文模型生成的风电场景数据

图 3.22: 本文模型和 C-StyleGAN2-SE 模型生成的新能源电网场景数据对比。



(a) C-StyleGAN2-SE 生成的光伏场景数据分布 (b) 本文模型生成的光伏场景数据分布 (c) C-StyleGAN2-SE 生成的风电场景数据分布 (d) 本文模型生成的风电场景数据分布

图 3.23: 本文模型和 C-StyleGAN2-SE 模型生成的新能源电网场景数据分布对比。

为了进一步验证本文跨时空模态生成对抗网络在具体的电网应用场景中的有效性，本文还设计了一个随机日前经济调度技术应用场景，如图 3.24(a)所示，该图展示了一个 IEEE-39-bus 测试系统，本文将风电数据集中的 66 个风机聚类成五个风电场，代入该 IEEE-39-bus 测试系统中形成风力发电并网，其中风机依据位置进行聚类，聚类结果的空间位置分布如图 3.24(b)所示，对应编号见表 3.4。然后整个电网的运行调度优化如下目标函数：

$$\min \sum_{t \in T} \left(\sum_{i \in CG} C_{i,t}^{uc} + \frac{1}{N_s} \sum_{s \in S} \sum_{i \in CG} C_{i,s,t}^{ed} \right). \quad (3.82)$$

具体的符号解释和经济调度运行原理可见补充材料^[253]。其经济调度结果见表 3.5，可见在本文模型生成的场景数据上进行经济调度，其结果最接近实际电网最低运行成本。

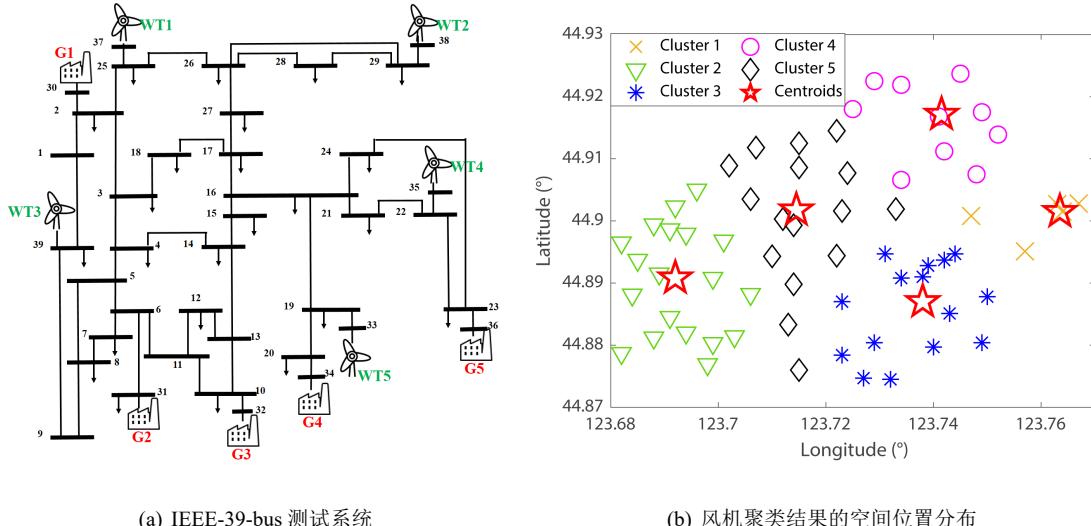


图 3.24: 风力发电并网仿真实验示意图

表 3.4: 风机聚类结果

Cluster	Bus Loc.	# WT
1	37	1-13, 18, 19, 20, 21, 26, 57
2	38	37, 40, 43-45, 47-52, 54, 55, 66
3	39	17, 24, 25, 27, 29-31, 34, 35, 38, 39, 46, 53
4	35	22, 28, 32, 33, 36, 41, 42, 61, 63-65
5	33	14-16, 23, 56, 58-60, 62

表 3.5: 随机日前经济调度结果

	Raw	ARIMA	LSTM	GRU	VAE	WGAN	C-StyleGAN2-SE	Our Model
Expected Cost (\$)	93,626,436	86,549,168 (-7.56%)	90,088,410 (-3.78%)	91,809,385 (-1.94%)	87,192,371 (-6.87%)	88,200,425 (-5.80%)	91,897,417 (-1.85%)	92,291,324 (-1.43%)

3.4 基于跨时空模态生成对抗网络的完全数据缺失插补技术

本章节翻译自作者论文^[254]，用于提供另一个多模态场景数据生成案例。在光伏并网发电的电力网络中，常常因为通讯故障^[61,62] 或外部的网络攻击^[63,64]，使得电网中的部分节点不可观测，从而该电网节点的观测数据完全缺失，如图 3.25 所示，本文将该问题归约为完全数据缺失插值问题。为了解决该问题，本文利用跨时空模态生成对抗网络建模观测数据中的不确定性，然后通过生成场景数据以插补不可观测电网节点的缺失数据。本文利用基于 Node2vec 算法的图矢量化嵌入技术、时序标签嵌入技术（参考公

式(3.69))和光伏电站容量标签嵌入技术，将对应模态的场景数据均嵌入成矢量表示，从而实现时空模态融合表示。进而输入跨时空模态生成对抗网络，最终实现从图结构数据到时序数据的跨模态场景数据生成和缺失值插补。本文在光伏数据集上进行了数值仿真测试，验证本文缺失值插补方法在完全数据缺失插值问题中性能达到SOTA。

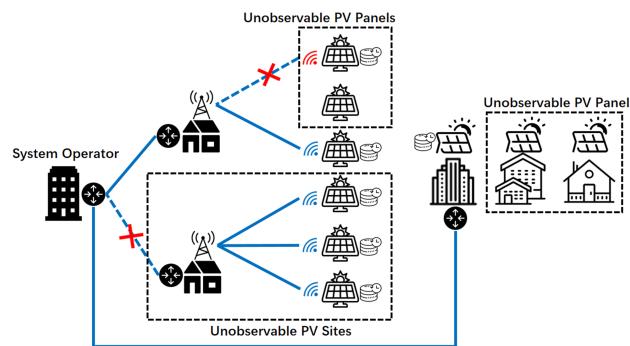


图 3.25: 不可观测的光伏电力单元示意图

3.4.1 跨时空模态生成对抗网络与完全数据缺失问题的联系

本文在章节3.3中已经提出了跨时空模态生成对抗网络框架，因此给定观测数据 $\mathbf{x} \in \mathbb{R}^{N \times T}$ 、图 \mathcal{G} 和光伏电站容量标签 c ，可以从公式(3.62)直接解出生成器 g 和判别器 f 的目标函数，如下：

$$\begin{aligned} L_f &= f(\hat{\mathbf{x}}, \mathcal{G}, c) - f(\mathbf{x}, \mathcal{G}, c) + \lambda (\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}}, \mathcal{G}, c)\| - 1)^2, \\ L_g &= -f(\hat{\mathbf{x}}, \mathcal{G}, c) + \frac{\mu}{CI(I_{i,t}=1)} \sum_{i,t} I_{i,t} \|\hat{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}\|^2, \end{aligned} \quad (3.83)$$

这里 $\hat{\mathbf{x}} \sim P_g(\mathbf{x} | \mathcal{G}, c)$ 表示生成的场景数据，用于区分 \mathbf{x} 。 $(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}}, \mathcal{G}, c)\| - 1)^2$ 是梯度惩罚项，原理可参考公式(3.18)。 $CI(I_{i,t}=1)$ 是条件计数函数，会计算出 \mathbf{x} 中的符合 $I_{i,t} = 1$ 条件的元素个数。 $I_{i,t}$ 是指示函数，表示矩阵 \mathbf{x} 中位置 (i, t) 的对应元素未缺失， $I_{i,t} = 0$ 则表示缺失。此外， λ 和 μ 是惩罚项的权重参数。因此本章节的跨时空模态生成对抗网络如图3.26所示。

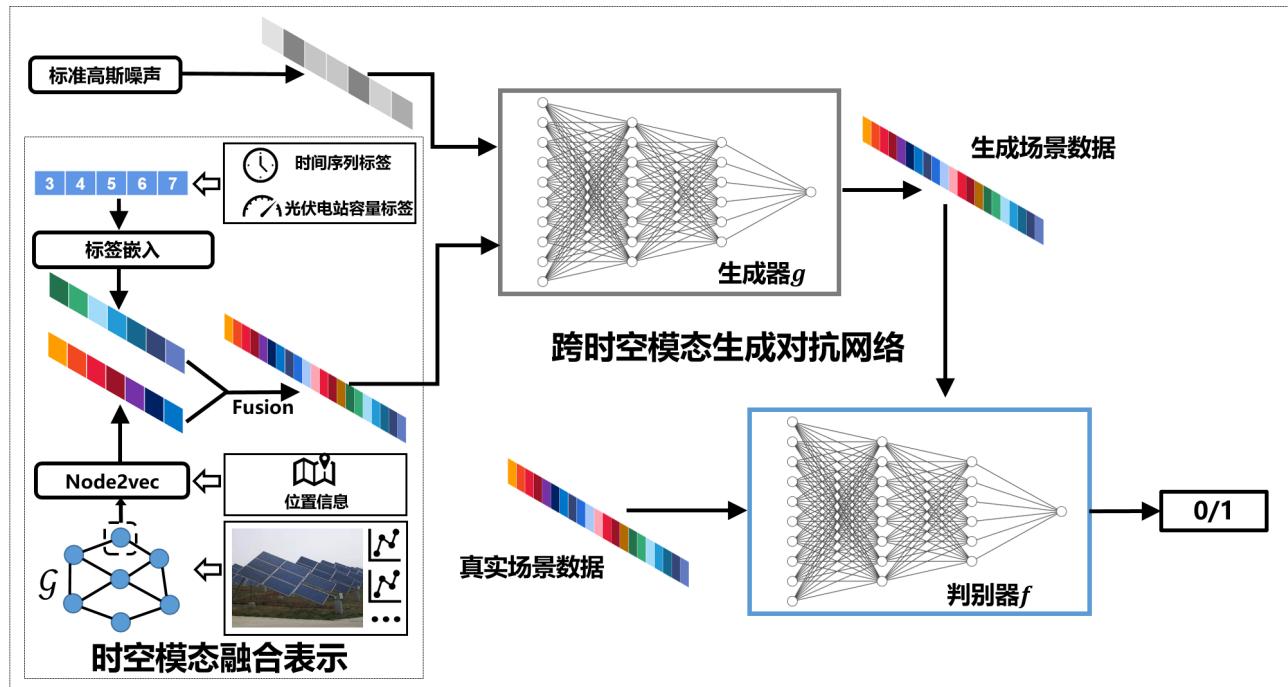


图 3.26: 基于跨时空模态生成对抗网络的完全数据缺失插值方法示意图

由公式(3.83)引出两个问题，第一个问题是跨时空模态生成对抗网络与完全数据缺失问题的联系。完全数据缺失问题是一类特殊的数据缺失问题，如图3.25所示，这意味着部分电网节点完全不可观测，且没有观测数据可以使用。在这种特殊情况下仍然使用时空Transformer（见章节3.3.2）则很难实现较好的特征提取效果。主要原因在于时空Transformer本质上还是一种人工神经网络，其主要功能在于学习观测的时间序列数据的分布特征，然而在完全数据缺失问题中，不可观测的电网节点没有时间序列数据可供提取特征，因而时空Transformer在该问题下也就失效了。此外，图结构数据不能为不可观测的电网节点提供全部的时空关联信息，因为光伏电站的选址定容和电站的地理位置并没有直接的联系。因此，为了确定一个完全不可观测的电网节点的容量，则需要在时空场景数据的基础上额外引入光伏电站的容量标签以提供更多的插值信息。于是，第二个问题是新增的容量标签，如何将其引入跨时空模态生成对抗网络的框架内，为此，本文引入时空模态融合表示方法，见下文。

3.4.2 时空模态融合表示

正如章节3.2.2所述，当模态数量越来越多，则自然而然地会使用多模态融合表示技术以整合所有多模态场景数据，因此本文提出时空模态融合表示技术以整合光伏电网中的时空模态数据和容量标签数据。本文利用基于Node2vec算法的图矢量化嵌入技术、时序标签嵌入技术和光伏电站容量标签嵌入技术，将对应模态的场景数据均嵌入成矢量表示，然后代入生成对抗网络的生成器 g 和判别器 f 中，如图3.26所示。

其中对时序标签和容量标签均采用同一种标签嵌入技术，如图3.27(a)所示，其设置一个标签嵌入层，内置一个参数矩阵 $\mathbf{w}_{\text{embed}} \in \mathbb{R}^{T \times d}$ 。然后针对一个时序样本，其对应的时序标签为 $[1, 2, 3]$ ，也即这个时序样本的时间戳，将其输入标签嵌入层则会索引到矩阵 $\mathbf{w}_{\text{embed}}$ 中对应的第1、2、3行，由这三行参数矢量构成时序标签的嵌入特征 $\mathbf{t} \in \mathbb{R}^{3 \times d}$ 。该标签嵌入技术原理对容量标签 $c \in \mathbb{Z}^+$ 同理，从而也可以将容量标签嵌入成 d 维矢量 $\mathbf{c} \in \mathbb{R}^{1 \times d}$ 。

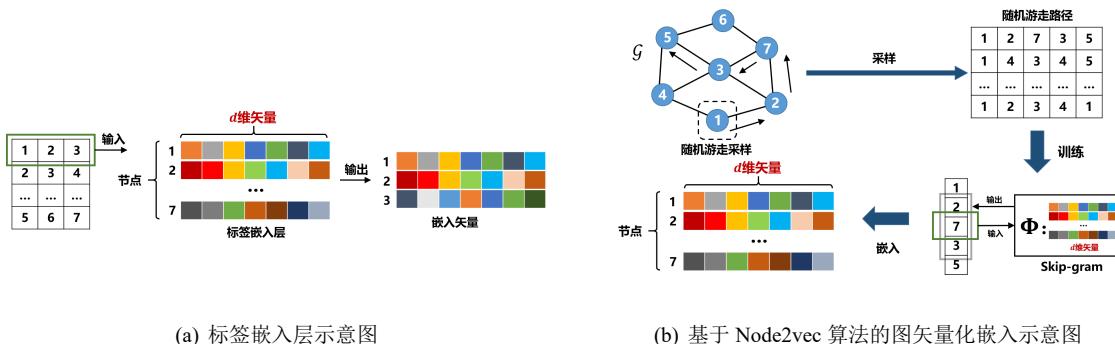


图3.27: 时空模态融合表示示意图

此外对于图结构数据则采用Node2vec算法进行图矢量化嵌入，如图3.27(b)所示，其首先在图 $G = (\mathbf{V}, \mathbf{E})$ 上随机游走采样，得到采样路径，该路径中的每个节点标签的“上下文”记录了 G 中该节点的邻居信息。于是通过skip-gram模型构造该节点标签到其“上下文”的映射 Φ ，skip-gram模型会构造矩阵 $\mathbf{w}_{\text{embed}} \in \mathbb{R}^{N \times d}$ ，同前文标签嵌入技术一样，当输入标签[7]，则会对应到 $\mathbf{w}_{\text{embed}}$ 中的第7行，从而得到一个 d 维矢量，并且同时选择该标签的“上下文”标签[2, 7, 3]对应的 d 维矢量。然后通过反向传播算法优化如下目标函数：

$$\max_{\mathbf{w}_{\text{embed}}} \sum_{j=1}^{N'} \log(p(\mathbf{v}_i \mathbf{v}_j)) + \sum_{k=1}^{N''} \log(p(-\mathbf{v}_i \mathbf{v}_k)) \quad (3.84)$$

这里 \mathbf{v}_j 是正样本，也即[2, 7, 3]对应 d 维矢量中的一个， N' 是正样本的个数。 \mathbf{v}_k 是负样本，也即[2, 7, 3]之外的其它节点对应的 d 维矢量中的一个， N'' 是负样本的个数，负样本在正样本之外的节点中随机选择一小部分。 $p(\cdot)$ 是Sigmoid函数，把矢量的点积映射到区间[0, 1]之间，用于表示给定节点 i ，则“上下文”

中出现节点 j 或不出现节点 k 的概率，因此该目标函数最大化正样本出现的概率，最小化负样本出现的概率。如上模型训练过程对随机游走采样路径中的每一个节点都适用，经过训练， $\mathbf{w}_{\text{embed}}$ 中的每一行都描述了对应节点的 d 维嵌入矢量，从而整理得到节点嵌入表示 $\mathbf{v} \in \mathbb{R}^{1 \times d}$ 。

当得到所有的矢量化嵌入表示特征，则将其全部拼接起来，作为生成对抗网络的输入。值得一提的是， \mathbf{t} 和 \mathbf{v} 作为坐标，描述了一个节点在一段时间的时间序列，因此其生成的时序场景数据可以明确地对应到时间序列数据的时空位置。

3.4.3 跨时空模态完全数据缺失插补算法

本文为此提出跨时空模态完全数据缺失插补算法，如算法 2 所示，其首先执行 Node2vec 算法得到图 G 的矢量化表示 \mathbf{v} ，然后随机选择一批样本并从中提取出时间序列标签嵌入特征 \mathbf{t} ，然后结合容量标签嵌入 \mathbf{c} ，然后采样 $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ ，将所有嵌入特征拼接并输入生成器 $g([\mathbf{z}, \mathbf{v}, \mathbf{t}, \mathbf{c}])$ 即可。此外，生成器 g 和判别器 f 均采用了基本的前馈神经网络参数化，且选择了 RMSprop 优化器训练其参数，学习率为 α 。其余部分的原理与基本的生成对抗网络一致，故不再过多赘述。

算法 2: 跨时空模态完全数据缺失插补

Input: 时间序列数据 \mathbf{x} ，图结构数据 G ，容量标签数据 c ，学习率 α ，迭代周期 n_{epoch} 、 n_f 和 n_g ；

Output: 插值后的时间序列数据；

/* Step 1: 时空模态融合表示 */

1 从 G 中采样随机游走路径；

2 将随机游走路径输入 skip-gram 模型，并训练该模型参数；

3 获得 G 中各个点对应的 d 维矢量表示，记为 \mathbf{v} ；

/* Step 2: 训练生成对抗网络 */

4 用 RMSprop 优化器初始化生成器 g 和判别器 f 的参数，学习率为 α ；

5 while 未到迭代周期 n_{epoch} do

6 for $k = 1, 2, \dots, n_f$ do

7 在 \mathbf{x} 中选择一批样本，记该样本对应时间序列标签 \mathbf{t} ，同时获得容量标签嵌入 \mathbf{c} ；

8 采样 $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ 和 $\epsilon \sim U(0, 1)$ ；

9 拼接 $[\mathbf{z}, \mathbf{v}, \mathbf{t}, \mathbf{c}]$ ；

10 $\hat{\mathbf{x}} \leftarrow g([\mathbf{z}, \mathbf{v}, \mathbf{t}, \mathbf{c}])$ ；

11 $\tilde{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon) \hat{\mathbf{x}}$ ；

12 $L_f \leftarrow f(\hat{\mathbf{x}}) - f(\mathbf{x}) + \lambda (\|\nabla_{\tilde{\mathbf{x}}} f(\tilde{\mathbf{x}})\| - 1)^2$ ；

13 $f \leftarrow \text{RMSprop}(\nabla_f L_f)$ ；

14 for $k = 1, 2, \dots, n_g$ do

15 在 \mathbf{x} 中选择一批样本，记该样本对应时间序列标签 \mathbf{t} ；

16 采样 $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ ；

17 $\hat{\mathbf{x}} \leftarrow g([\mathbf{z}, \mathbf{v}, \mathbf{t}, \mathbf{c}])$ ；

18 $L_g \leftarrow -f(\hat{\mathbf{x}}) + \frac{\mu}{CI(I_{i,t}=1)} \sum_{i,t} I_{i,t} \|\hat{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}\|^2$ ；

19 $g \leftarrow \text{RMSprop}(\nabla_g L_g)$ ；

/* Step 3: 完全数据缺失插补 */

20 遍历每个样本，采样 $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ 并通过 $g([\mathbf{z}, \mathbf{v}, \mathbf{t}, \mathbf{c}])$ 生成场景数据，替换 \mathbf{v} 和 \mathbf{t} 对应位置的缺失数据；

21 return 插值后的时间序列数据。

3.4.4 实验设置

为了验证本文模型性能，本文选择了两个光伏数据集，其采样自 National Renewable Energy Laboratory (NREL) 数据集^[248]，其包含了 2006 年美国北卡罗来纳州 (North Carolina, NC) 和亚利桑那州 (Arizona, AZ) 全年光伏发电仿真数据。其中 NC 数据集包含光伏电站数量 149，容量范围为 20MW~39MW，AZ 数据集包含光伏电站数量 49，容量范围为 39MW~118MW。因此二者各有特点，NC 数据集中节点数量多，且容量范围小，而 AZ 数据集中节点数量少，且容量范围大。其二者的光伏电站空间位置分布如图 3.28 所示。

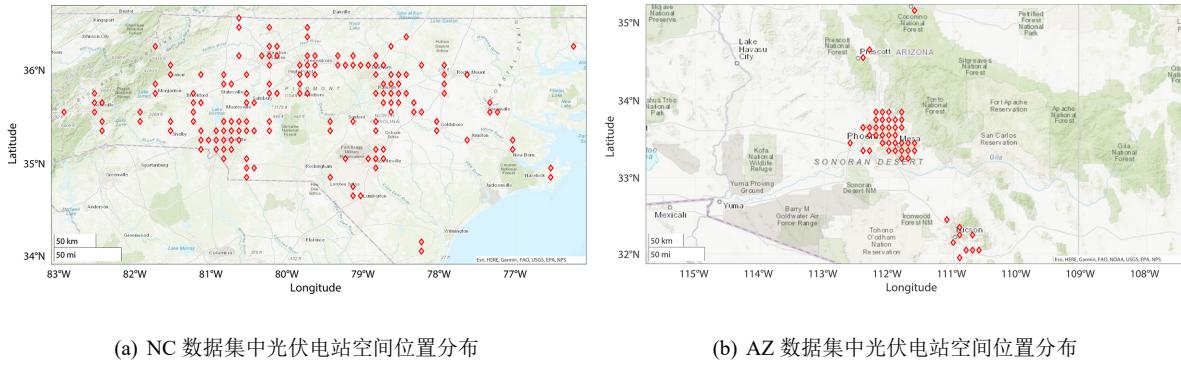


图 3.28: 光伏电站空间位置分布

本文插值实验不划分训练集和测试集，仅对完整的观测数据中随机选择部分节点并隐藏其全部观测数据，从而构造完全数据缺失。对于缺失的数据则将其值置为零，于是可以定义缺失率如下：

$$\eta = \frac{CI[I_{i,t} = 0]}{N \times T} \times 100\%, \quad (3.85)$$

这里 $I_{i,t} = 0$ 表示时序观测场景数据 $\mathbf{x} \in \mathbb{N} \times \mathbb{T}$ 中位置 (i, t) 的值缺失。 $CI[I_{i,t} = 0]$ 是条件计数函数，描述缺失值的个数。

此外，图 \mathcal{G} 的网络结构来自于 GPS 数据，如图 3.28 所示，从而其导出的邻接矩阵 A 构造如下：

$$A_{ij} = \begin{cases} e^{-\frac{d_{ij}}{\sigma}} & , i \neq j, \\ 0 & , i = j, \end{cases} \quad (3.86)$$

这里 d_{ij} 是节点 i 和节点 j 的 GPS 位置计算出的球面距离， $\sigma = 100$ 是规格化因子。

此外，本文采用均方根误差 (Root Mean Square Error, RMSE) 指标和平均绝对误差 (Mean Absolute Error, MAE) 指标以评估模型的插值性能，其定义如下：

$$\text{RMSE} = \sqrt{\frac{1}{N \times T} \sum_{i=1}^N \sum_{t=1}^T (\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t})^2}, \quad (3.87)$$

$$\text{MAE} = \frac{1}{N \times T} \sum_{i=1}^N \sum_{t=1}^T |\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t}|.$$

这里 $\hat{\mathbf{x}}$ 是生成的时序场景数据。此外，为了评估不同模态数据对插值性能的提升作用，本文在插值任务中隐藏部分模态数据，从而评估有该模态和没有该模态情况下模型性能的差异。本文定义指标如下：

$$\text{性能提升} = \frac{\text{指标}_{\text{Model}} - \text{指标}_{\text{Model}'}}{\text{指标}_{\text{Model}'}} \times 100\%. \quad (3.88)$$

这里指标是指 RMSE 或 MAE，Model 是完整的本文模型，Model' 是在完整模型上隐藏部分模态功能的模型。

此外，本文跨模态生成对抗网络 (CM-GAN) 中生成器 g 和判别器 f 分别采用两个三层的前馈神经网络实现，生成器 g 的每层神经元数均为 1024，输出层维度为 48，判别器 f 的每层神经元数均为 512，输出层维度为 1，其均采用 Leaky ReLU 函数 ($\text{Leaky ReLU}(x) = \max(0.2x, x)$) 激活。此外，生成器 g 输入层维度为 562，其中 512 是节点嵌入特征 \mathbf{v} 的维度，时序标签 \mathbf{t} 和容量标签 \mathbf{c} 的嵌入特征维度均为 20，采样高斯噪声的维度是 10。此外，公式 (3.83) 中惩罚项权重参数 $\lambda = \mu = 10$ 。RMSprop 优化器学习率 $\alpha = 0.00001$ ，迭代周期 $n_{\text{epoch}} = 1000$ 、 $n_f = 5$ 和 $n_g = 3$ 。

此外，为了比较本文跨时空模态生成对抗网络 (CM-GAN) 和其它基准模型的完全数据缺失插值性能，本文引入如下基准模型：

1. Bayesian Gaussian CANDECOMP/PARAFAC (BGCP)^[255]: 一种贝叶斯张量分解方法，用于对时空交通缺失数据进行插补，本文将其用于新能源电网中的完全数据缺失问题，其模型中张量的秩假设为 10 (一个超参数)。
2. Low-rank Autoregressive Tensor Completion (LATC)^[256]: 一种低秩自回归张量补全方法，用于对时空交通缺失数据进行插补，其假设数据张量具有低秩性，该张量秩假设为 10 (一个超参数)，从而可以低秩表述和重构，重构的张量可用以插补缺失值，本文将其用于新能源电网中的完全数据缺失问题。
3. Transformer^[200,257]: 基本的 Transformer 模型，具体原理见本文章节 3.2.4。
4. Multi-View Spatial-Temporal Graph Network (MVSTGN)^[258]: 一种多视角的时空 Transformer 模型，原理上与本文提出的时空 Transformer 模型 (见本文章节 3.3.2) 几乎一致，但其主要用于交通数据缺失插补，本文将其直接用于新能源电网中的完全数据缺失问题。
5. Cross-Modal Attention Network (CMAN)^[259]: 一种基于动态跨模态注意力网络模型的两阶段预测方法，本文主要使用其中动态跨模态注意力网络模型解决本文新能源电网中的完全数据缺失问题，所以只考虑其中第一阶段，也即文献^[259] 中的公式 (24) 和公式 (25)，其中的参数 $\sigma_1 = \sigma_2 = 1$ 。
6. SolarGAN^[260]: 一种条件生成对抗网络，可参考公式 (3.47)。特点 (1) 非多模态场景数据生成，只只使用了光伏出力的时间序列。(2) 时间序列与高斯噪声相加并输入生成器 (3) 使用门控循环单元网络提取时序特征，可参考本文章节 3.2.3。
7. C-StyleGAN2-SE^[252]: 生成对抗网络的一个变种，特点是针对电网中多样式的场景数据进行条件生成。其采用原文献中默认设置的模型参数。

以上基准模型，如无特殊声明，则均采用原文献中默认的参数设置。

3.4.5 实验结果

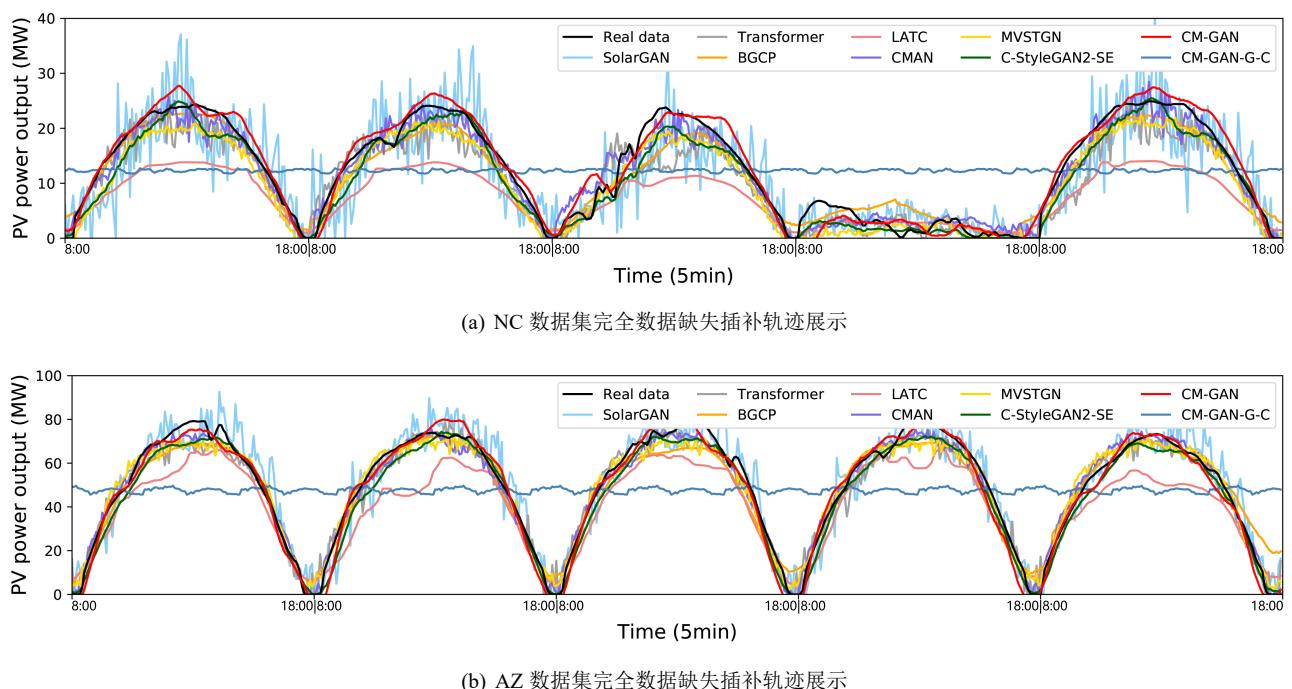
实验结果如表 3.6 和表 3.7 所示，面对不同的数据缺失率，本文模型在完全数据缺失问题上实现了 SOTA 水平的缺失值插补性能，其缺失的光伏出力插值轨迹如图 3.29 所示。值得注意的是，图结构数据和容量标签数据均帮助提升了本文模型的性能，如图 3.29 所示，在 CM-GAN-G-C 中，如果去掉图结构数据和容量标签数据，仅仅依靠时间序列数据训练生成对抗网络，则不可观测的完全数据缺失节点对应的生成数据只能捕捉轨迹均值。而在表 3.6 中，图结构数据和容量标签数据对插值性能提供了 60.5% ~ 67% 的提升，在表 3.7 中，则对插值性能提供了 68.2% ~ 72.5% 的提升。至此，综合两篇文献^[247,254] 中的主要结论，可知多模态数据对基于生成模型的场景数据生成性能有着重要影响。

表 3.6: NC 数据集完全数据缺失插补结果

η	5%		10%		15%		20%		25%	
	RMSE	MAE								
BGCP	0.104	0.079	0.106	0.082	0.110	0.082	0.113	0.080	0.113	0.083
LATC	0.182	0.105	0.188	0.108	0.192	0.110	0.198	0.116	0.207	0.123
Transformer	0.118	0.088	0.121	0.090	0.125	0.092	0.127	0.093	0.130	0.094
MVSTGN	0.111	0.083	0.112	0.083	0.114	0.085	0.114	0.086	0.117	0.088
CMAN	0.112	0.081	0.116	0.085	0.119	0.088	0.125	0.089	0.129	0.095
SolarGAN	0.111	0.084	0.111	0.084	0.113	0.085	0.122	0.090	0.130	0.098
C-StyleGAN2-SE	0.104	0.076	0.105	0.077	0.109	0.081	0.110	0.082	0.111	0.083
CM-GAN	0.103	0.075	0.103	0.076	0.101	0.075	0.103	0.076	0.104	0.076
CM-GAN-G-C	0.261 (-60.5%)	0.226 (-66.8%)	0.261 (-60.5%)	0.227 (-66.5%)	0.261 (-61.3%)	0.227 (-67.0%)	0.263 (-60.8%)	0.227 (-66.5%)	0.263 (-60.5%)	0.228 (-66.7%)
CM-GAN-T	0.129 (-20.2%)	0.092 (-18.5%)	0.129 (-20.2%)	0.095 (-20.0%)	0.131 (-22.9%)	0.098 (-23.5%)	0.132 (-22.0%)	0.098 (-22.5%)	0.132 (-21.2%)	0.098 (-22.4%)
CM-GAN-T-G	0.105 (-1.9%)	0.077 (-2.6%)	0.105 (-1.9%)	0.077 (-1.3%)	0.106 (-4.7%)	0.078 (-3.8%)	0.107 (-3.7%)	0.078 (-2.6%)	0.107 (-2.8%)	0.078 (-2.6%)
CM-GAN-T-C	0.109 (-5.5%)	0.079 (-5.1%)	0.110 (-6.4%)	0.081 (-6.2%)	0.110 (-8.2%)	0.081 (-7.4%)	0.110 (-6.4%)	0.082 (-7.3%)	0.112 (-7.1%)	0.083 (-8.4%)

表 3.7: AZ 数据集完全数据缺失插补结果

η	5%		10%		15%		20%		25%	
	RMSE	MAE								
BGCP	0.128	0.111	0.142	0.123	0.118	0.101	0.116	0.099	0.115	0.098
LATC	0.189	0.147	0.197	0.154	0.203	0.160	0.212	0.165	0.219	0.169
Transformer	0.121	0.097	0.124	0.101	0.126	0.103	0.127	0.105	0.128	0.105
MVSTGN	0.118	0.089	0.119	0.091	0.123	0.093	0.124	0.096	0.127	0.097
CMAN	0.094	0.070	0.095	0.073	0.098	0.074	0.099	0.074	0.101	0.077
SolarGAN	0.119	0.093	0.119	0.092	0.120	0.098	0.118	0.096	0.119	0.097
C-StyleGAN2-SE	0.101	0.083	0.103	0.084	0.103	0.085	0.104	0.088	0.105	0.089
CM-GAN	0.079	0.058	0.082	0.059	0.084	0.062	0.084	0.061	0.083	0.062
CM-GAN-G-C	0.263 (-70.0%)	0.211 (-72.5%)	0.263 (-68.8%)	0.212 (-72.2%)	0.264 (-68.2%)	0.213 (-70.9%)	0.265 (-68.3%)	0.214 (-71.5%)	0.265 (-68.7%)	0.214 (-71.0%)
CM-GAN-T	0.105 (-24.8%)	0.083 (-30.1%)	0.108 (-24.1%)	0.083 (-28.9%)	0.110 (-23.6%)	0.087 (-28.7%)	0.110 (-23.6%)	0.087 (-29.9%)	0.111 (-25.2%)	0.087 (-28.7%)
CM-GAN-T-G	0.085 (-7.1%)	0.065 (-10.8%)	0.085 (-3.5%)	0.066 (-10.6%)	0.087 (-3.4%)	0.068 (-8.8%)	0.089 (-5.6%)	0.070 (-12.9%)	0.089 (-6.7%)	0.071 (-12.7%)
CM-GAN-T-C	0.096 (-17.7%)	0.076 (-23.7%)	0.098 (-16.3%)	0.077 (-23.4%)	0.101 (-16.8%)	0.078 (-20.5%)	0.102 (-17.6%)	0.078 (-21.8%)	0.103 (-19.4%)	0.079 (-21.5%)

图 3.29: 缺失率 $\eta = 5\%$ 情况下的完全数据缺失插补轨迹展示

第四章 基于因果推断的场景数据生成

至此，本文已经论述了基于系统辨识的场景数据生成中，知识驱动系统辨识和数据驱动系统辨识之间此消彼长的关系。本文还论述了基于生成模型的场景数据生成中，逐渐从确定性转向不确定性的建模方法。然而，其二者都不得不服从 $P(x) = P(y)$ 的约束，这里 x 是观测的场景数据， y 是从模型中生成的场景数据（参考本文章节 1.2）。正如大卫·休谟在《人类理解研究》中所言，“我们从观测中发现事物之间的因果性，它引导我们从一个事实推理出另一个事实，并且它还时常使我们相信一些毫无根据的事实，这使得我们超越了自身的记忆和感知”。尽管大卫·休谟对人类能否通过自身理性发现事物之间的因果性抱持怀疑态度，但他也并不否认因果性的存在。这在本文的语境下，可以描述为存在场景数据生成方法使得 $P(x) \neq P(y)$ ，因此本文这里重点讨论如何利用因果推断实现场景数据生成，本文沿着从潜在结果模型^[33] 到结构因果模型^[31,34] 的主线，引出对复杂系统时空因果的讨论，最终使得生成场景数据超出观测场景数据的边界。

4.1 反事实推理

4.1.1 潜在结果模型

当我们想知道事物 T 对另一事物 Y 是否有因果影响时，我们就需要设计并实施随机对照实验。简言之，其会设置两个分组，分别是实验组和对照组，在对照组中，其会设置 $T = 0$ 以表示不接受处理，从而观测 Y 的值。而在实验组中，其设置 $T = 1$ 以表示接受处理，从而尝试观测不同的 Y ，如果 Y 发生显著变化，则认为 T 对 Y 有因果影响，否则没有。当然这只是在理想的实验条件下才能得出的结论，实际的实验过程要更加复杂。首先，实验中实际除了 T 和 Y 之外还有一组协变量 \mathbf{X} ，协变量 \mathbf{X} 中可能会有混杂因子，其同时影响 T 和 Y ，从而产生混杂偏倚，于是干扰得出正确的因果关系。比如，在著名的辛普森悖论^[261] 中，辛普森按照性别划分样本并分别做对照试验，其在两组对照实验中均得到的相同的结论，但当其将所有样本混合之后，对照实验的结论却完全发生了逆转，因此在这个案例中，性别就是混杂因子，其会对使对实验结论产生混杂偏倚。当然混杂因子还有其它可能，因此，为了消除协变量 \mathbf{X} 中的混杂因子对 T 和 Y 的影响，就需要将 T 尽可能随机地分配到实验组和对照组中，从而消除混杂因子导致的混杂偏倚。因此，实际上因果推断的核心思想是朴素的，也即通过比较接受处理 T 后目标变量 Y 的变化，从而发现因果关系。但真正的难题是去除混杂，因为在真实世界中，可能影响实验结果的混杂因子实在是太多了，往往很难通过随机化 T 从而一一去除 \mathbf{X} 中的混杂影响。此外，这种随机化处理需要大量的实验样本作为对照，而真实世界中的很多事物并不允许人为地干预多次，从而使得获取大量对照样本十分困难。

因此，整理上述变量可得表 4.1，从而构造潜在结果模型，用于系统性地去除混杂和评估因果效应。潜在结果模型的关键之处在于将目标变量 Y 表示成潜在结果变量 $Y(1)$ 和 $Y(0)$ ，这里不妨设处理变量 T 为二值变量，如果 $T = 0$ ，则表示在对照组中观测到 $Y(0)$ ，否则 $T = 1$ 表示在实验组中观测到 $Y(1)$ 。为了划分实验组和对照组，所以观测单位至少有两个，当然实际测试时远不止两个，观测单位越多越容易消除观测误差。此外，这里还有两个隐含假设，其一是假设观测单位之间没有交互，如果存在交互，那么就无法判断其中一个观测单位的状态是否受到另一个观测单位的影响。其二是假设因果机制在时间上连

表 4.1: 潜在结果模型示例

观测单位 <i>i</i>	协变量 X	潜在结果变量		处理变量 <i>T</i>
		<i>Y</i> (1)	<i>Y</i> (0)	
1	x ₁	<i>y</i> ₁ (1)	?	1
2	x ₂	?	<i>y</i> ₂ (0)	0
3	x ₃	?	<i>y</i> ₃ (0)	0
4	x ₄	<i>y</i> ₄ (1)	?	1
5	x ₅	<i>y</i> ₅ (1)	?	1
:	:	:	:	:
<i>M</i>	x _{<i>M</i>}	?	<i>y</i> _{<i>M</i>} (0)	0

续稳定，也即因果机制不会随着时间而发生改变。因为在部分特殊场景下，观测单位数量有限，从而需要在不同时间测试多次，这时就需要假设因果机制在时间上连续稳定。

于是，从表 4.1 可以引出潜在结果模型的两个重要概念。其一是反事实，如果一件事物 *T* 导致了事物 *Y* 的发生，那么如果事物 *T* 没有发生，则事物 *Y* 也不会发生，这种从“如果”设置的条件中推理出的结论被称为反事实的^[262]。这包含了两个关键点，首先，反事实推理应该在观测之后，需要先观测到“事物 *T* 导致了事物 *Y* 的发生”，然后才能通过“如果”设置新的发生条件，从而推理反事实的 *Y*，这也就使得反事实推理总是建立在观测之上。此外的第二个关键点是，反事实推理出的 *Y* 是不可观测的，可以观测的 *Y* 被称为事实，与反事实相对。因此，理论上反事实可以超越观测的边界，其基于因果关系而从观测到的事实延展至反事实，而反事实可能不在观测的事实内。于是回到表 4.1，对于一个观测单位，当 *T* 取值固定，则可以观测到 *Y* 的一个潜在结果，比如当 *T* = 1 则观测到 *Y*(1)，那么此时 *Y*(0) 就是不可观测的反事实 *Y*，在表 4.1 中标记为“?”，这对 *T* = 0 同理。因此，对于一个观测单位，其总是有一个潜在结果被观测到，而另一个潜在结果没有被观测到的潜在结果被称为反事实。

第二个重要概念是平均因果效应 (Average Causal Effect, ACE)，平均因果效应是实验组和对照组之间潜在结果的期望差异。给定协变量 **X**，其可以定义如下：

$$\text{ACE}(T \rightarrow Y | \mathbf{X}) = \mathbb{E}[Y(1) - Y(0) | \mathbf{X}], \quad (4.1)$$

也即给定协变量 **X** 的条件平均因果效应。如果对协变量 **X** 求期望，可以定义总体的平均因果效应如下：

$$\text{ACE}(T \rightarrow Y) = \mathbb{E}_{\mathbf{X}}[\text{ACE}(T \rightarrow Y | \mathbf{X})] = \mathbb{E}_{\mathbf{X}}[\mathbb{E}[Y(1) - Y(0) | \mathbf{X} = \mathbf{x}]] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)] \quad (4.2)$$

有了平均因果效应，就可以量化和估计采取处理 *T* 对目标变量 *Y* 的因果影响，进而判断是否应该采取处理。但是求解公式 (4.2) 的真正的难点在于，潜在结果变量 *Y*(1) 和 *Y*(0) 只能观测其一，而另一个是反事实的，依定义不能从观测中得到，所以如表 4.1 所示，反事实值都是缺失的。此外的另一个难点是，由于对于不同的观测单位，其协变量 **X** 的取值往往不同，尤其是在高维 **X** 的情况下，样本量相对于样本维度是稀疏的，因此时常找不到相同 **X** 条件下的两个实验组样本和对照组样本用于计算公式 (4.2)。

由此，为了在潜在结果模型中估计反事实值，一种解决方法是匹配法。简言之，对于观测单位 *i*，协变量值为 **x**_{*i*}，处理变量值为 *t*_{*i*}，目标变量值为 *y*_{*i*}(*t*_{*i*})，则在观测样本中匹配样本，找到样本 **x**_{*j*}，满足 **x**_{*j*} = **x**_{*i*} 和 *t*_{*j*} = 1 - *t*_{*i*}，于是反事实值 *y*_{*i*}(1 - *t*_{*i*}) = *y*_{*j*}(*t*_{*j*})，该方法又称为精确匹配。精确匹配主要是匹配协变量 **X**，而在高维情况下，样本量相对于样本维度是稀疏的，因而精确匹配难以执行，所以实际使用的主要还是近似匹配方法。近似匹配方法主要有两种，其一是卡钳匹配^[263-266]，简言之，就是给协变量 **X** 的匹配设置一个误差范围，在误差范围内即算匹配成功。其二是倾向得分匹配^[267-270]，简言之，就是构建逻辑斯蒂回归模型，输入一组协变量 **X**，输出处理变量 *T*，从而将协变量从高维降低至一维，再根据倾向得分进行卡钳匹配。逻辑斯蒂回归模型是经典的二分类模型，其会输出处理变量 *T* = 1 的条件概率 *P*(*T* = 1 | **X**)，这就是倾向得分，也即给定一组协变量 **X** 条件下，样本属于实验组的概率。

第二种解决方法是插值法，简言之，就是把表 4.1 中的反事实值视作缺失值，然后利用模型拟合观测数据，然后从模型中采样并插补缺失值。比较常用的模型是统计中的线性或非线性回归模型^[271-274]。此外，还有一些张量分解方法^[275-277]也可以实现插值，此类方法将表 4.1 中数据建模成张量，通过低秩分解张量并重构以实现缺失值插补。但需要注意的是，张量分解方法在使用前需要额外的低秩假设，也即假设张量可以分解为多个低秩张量的组合以简化表示。此外，前述章节的所有系统辨识方法和生成模型方法对此问题均适用，此二者无非是模型含义不同，但本质上都是利用模型拟合观测数据，然后从模型中采样并插补缺失值。

4.1.2 潜在结果模型的基础假设

要通过潜在结果模型推理反事实，需要一些基础假设。首先是 SUTVA (stable unit treatment value assumption) 假设，定义如下：

假设 2 (SUTVA^[278]). 在表 4.1 中，

1. 观测单元之间互不干扰。也即对观测单元 $i = 1, \dots, M$ ，无论是 $y_i(1)$ 还是 $y_i(0)$ 不会受到其它观测单元所接受处理的影响。
2. 没有隐藏的处理变量。也即，如果观测单元 i 接受处理 $t_i = 1$ ，则观测到的潜在结果是 $y_i(1)$ ，如果接受的处理 $t_i = 0$ ，则观测到的潜在结果是 $y_i(0)$ 。

SUTVA 假设保证了观测单元之间的独立性，不仅没有交互，而且没有未加考虑的处理变量，对于观测单元 i ，其观测的潜在结果 $y_i(t_i)$ 只取决于其处理变量值 t_i 。但这个假设并不总是满足的，尤其是在社交网络场景或流行病场景下，观测单元之间存在大量的交互。

第二条假设是一致性假设，定义如下：

假设 3 (一致性^[43,44]). 在表 4.1 中，如果观测单元 i 接受处理 t_i ，则所观测到的潜在结果 $y_i(t_i)$ 与其它接受同样处理的观测单元所观测到的潜在结果相同。

通过一致性假设，可以得到

$$\begin{aligned}\mathbb{E}[Y(1)|T = 1] &= \sum_{i=1}^M y_i \times t_i = \mathbb{E}[Y|T = 1], \\ \mathbb{E}[Y(0)|T = 0] &= \sum_{i=1}^M y_i \times (1 - t_i) = \mathbb{E}[Y|T = 0].\end{aligned}\tag{4.3}$$

该公式对含有协变量情况下的条件期望同样有效，也即 $\mathbb{E}[\mathbb{E}[Y(1)|T = 1, \mathbf{X}]] = \mathbb{E}[\mathbb{E}[Y|T = 1, \mathbf{X}]]$ ，且 $\mathbb{E}[\mathbb{E}[Y(0)|T = 0, \mathbf{X}]] = \mathbb{E}[\mathbb{E}[Y|T = 0, \mathbf{X}]]$ 。一致性假设在潜在结果模型中十分重要，其实际上假设了未观测的反事实值可以从其它具有同等协变量条件却接受不同处理的观测单位的观测样本中得到，这构成了前述章节 4.1.1 中匹配法和插值法的理论基础。同时，一致性假设也使得基于潜在结果模型的反事实推理不会超出观测的范围，这是潜在结果模型的优点，其推理出的反事实值可以通过观测样本验证其推理的正确性，所以一些使用潜在结果模型的研究论文常采用一些显式的评估指标以衡量反事实推理的准确性，但在使用结构因果模型^[31,34]的研究论文中就鲜有这种准确性评估方面的便利。但是，反事实推理不能超出观测范围也是潜在结果模型的缺点，因为该假设对反事实推理的约束条件过强，不符合反事实推理的一般性定义，也即反事实推理基于观测并经过因果关系延展至观测之外，这也是潜在结果模型和结构因果模型的根本分歧之一，本文后面会详细论述。

第三条假设是正值假设，定义如下：

假设 4 (正值假设^[267]). 在表 4.1 中， $0 < P(T = 1|\mathbf{X} = \mathbf{x}) < 1$ 。

正值假设保证了，对于 $\mathbf{X} = \mathbf{x}$ ，总是有一些样本被随机的分配到实验组和对照组。如果所有样本都被分配到实验组，则对照组没有样本可供对照。同样地，如果所有样本都被分配到对照组，则实验组没有样本可供评估处理 T 的平均因果效应。

第四条假设是可忽略性假设，定义如下：

假设 5 (可忽略性^[43,44]). 在表 4.1 中， $T \perp\!\!\!\perp Y(1), Y(0) | \mathbf{X}$ 对每个观测单位均成立。

可忽略性假设保证了，所有的混杂因子都包含在协变量 \mathbf{X} 内，没有未被观测的混杂因子会对处理结果造成混杂偏倚。简言之，所有可能产生混杂偏倚的混杂因子都已被观测，所以仅去除协变量 \mathbf{X} 中的混杂因子就可以消除所有混杂偏倚。因此，可忽略性假设又称为无混杂假设，该假设常被用于保证实验中样本选择的随机性^[279]，从而避免人为操作的倾向性。

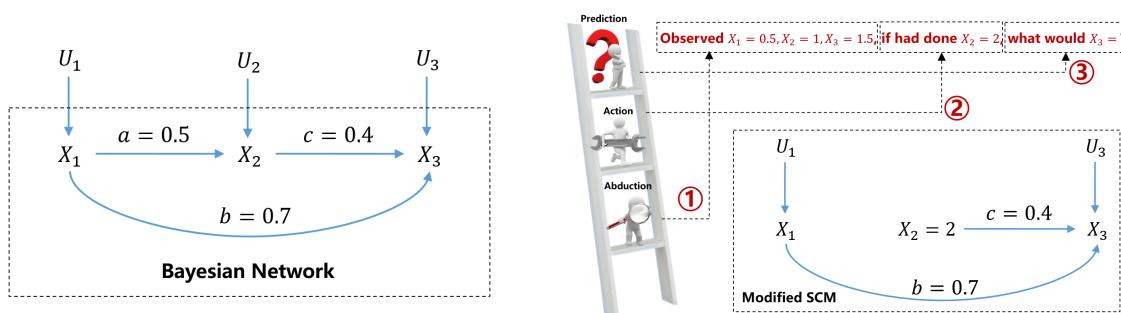
基于这些基础假设，可以简化公式 (4.2) 中的平均因果效应，如下：

$$\begin{aligned} \text{ACE}(T \rightarrow Y) &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y(1) - Y(0) | \mathbf{X} = \mathbf{x}]], \\ &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y(1) | T = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y(0) | T = 0, \mathbf{X} = \mathbf{x}]] \quad /* \text{ 可忽略性 } */ \\ &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y | T = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y | T = 0, \mathbf{X} = \mathbf{x}]]. \quad /* \text{ 一致性 } */ \end{aligned} \quad (4.4)$$

由此，平均因果效应可以利用观测样本中计算出。也正因此，潜在结果模型仅仅保证能够从一个事实推导出另一个事实，但是当涉及真正的反事实时，其会利用诸多假设将反事实也同样约束在可观测的事实范围内，因而潜在结果模型无法外推数据，甚至其原理本身就是排斥外推数据的。

4.1.3 结构因果模型

在对于反事实的定义（见表 4.1）上，结构因果模型与潜在结果模型相同，但在推理反事实的问题上，结构因果模型选用了不同的基础假设，从而得到了不同的结论，也正是这样的分歧，使得结构因果模型理论上具备数据外推的能力，本文这里首先论述如何利用结构因果模型推理反事实。结构因果模型由一组变量和一组响应函数构成，其中变量又可以划分成内生变量 $\mathbf{V} = \{X_1, \dots, X_N\}$ 和外生变量 $\mathbf{U} = \{U_1, \dots, U_N\}$ 。内生变量是可观测的，而外生变量通常是不可观测的，因此可以理解为外部输入的激励或噪声。外生变量间相互独立，并且，内生变量和外生变量在编号上一一对应，也即一个外生变量根据编号指向唯一的一个内生变量。此外，响应函数 $\mathbf{F} = \{f_1, \dots, f_N\}$ 用于描述变量间的函数关系。



(a) 结构因果模型（出自文献^[280] 中 Figure 4.1 或文献^[66] 中 Figure 2） (b) 因果阶梯（出自文献^[280] 中 Figure 4.2 或文献^[66] 中 Figure 3）

图 4.1: 结构因果模型示意图

举个例子，如图 4.1(a)所示，结构因果模型可以描述如下：

$$\begin{aligned} f_1 : X_1 &= U_1, \\ f_2 : X_2 &= aX_1 + U_2, \\ f_3 : X_3 &= bX_1 + cX_2 + U_3, \end{aligned} \tag{4.5}$$

这里 $a = 0.5, b = 0.7, c = 0.4$ 。 U_1, U_2, U_3 均为加性噪声，并且其相互独立。由此可见，公式 (4.5) 具备系统公式 (1.1) 的形式，因此可以视为系统公式的另一种表示。但略有不同的地方在于，结构因果模型会对系统内生变量施加一个有向无环约束，如图 4.1(a)所示，该有向无环约束被建模成一个有向无环图，根据连边含义不同，其又被称为贝叶斯网络或因果网络^[31]。该约束使得变量间的因果关系不能有环，比如不能互为因果等。

为了推理反事实，结构因果模型构造了因果阶梯，如图 4.1(b)所示。因果阶梯主要分为三步，定义如下：

1. 漂因 (abduction): 从观测数据中推断外生变量 \mathbf{U} 的值。
2. 干预 (action): 对结构因果模型施加干预，比如对图 4.1(a)中结构因果模型施加 $\text{do}(X_2 = 2)$ ，若如此做，则 X_2 会被赋予值 2，并且所有指向 X_2 的箭头都会被修改为无，如图 4.1(b)所示。
3. 预测 (prediction): 利用修改后的结构因果模型重新计算目标变量的值，该值为反事实值。

于是，当询问反事实问题，比如“现在已知 $X_1 = 0.5, X_2 = 1, X_3 = 1.5$ ，如果当初 $X_2 = 2$ ，那么现在 $X_3 = ?$ ”这样一个反事实问题，回答该问题首先需要漂因，也即从观测数据中计算外生变量 \mathbf{U} 的值，如下：

$$\begin{aligned} U_1 &= 0.5, \\ U_2 &= 1 - 0.5 \times 0.5 = 0.75, \\ U_3 &= 1.5 - 0.7 \times 0.5 - 0.4 \times 1 = 0.75. \end{aligned} \tag{4.6}$$

进而，干预 $X_2 = 2$ ，也即 $\text{do}(X_2 = 2)$ 以获得修改后的结构因果模型。最终，重新计算 X_3 ，如下：

$$X_3(\text{do}(X_2 = 2)) = 0.5 \times 0.7 + 2 \times 0.4 + 0.75 = 1.9. \tag{4.7}$$

可见，干预后重新计算的反事实值 $X_3(\text{do}(X_2 = 2))$ 不同于观测值 X_3 。

此外，这里 $\text{do}(\cdot)$ 又被称为 do 算子，用于描述因果阶梯中的干预操作。通过模块性假设，其还可以推广到概率形式下的多变量干预，模块性假设定义如下：

假设 6 (模块性^[34])。如果有一组变量 $\{X_{j_1}, \dots, X_{j_p}\} \subset \mathbf{V}$ 被干预，于是对于任意变量 $X \in \mathbf{V}$ ，其满足：

1. 如果 $X \notin \{X_{j_1}, \dots, X_{j_p}\}$ ，则 $P(X|\mathbf{Pa}(X))$ 保持不变。
2. 如果 $X \in \{X_{j_1}, \dots, X_{j_p}\}$ ，则当 x 是对变量 X 的干预值时， $P(X = x|\mathbf{Pa}(X)) = 1$ ，否则 $P(X = x|\mathbf{Pa}(X)) = 0$ 。

这里 $\mathbf{Pa}(X)$ 是变量 X 在图 4.1(a)贝叶斯网络中的直接前驱节点的集合，也即变量 X 的因果父母。这意味着，如果对变量 X_{j_1}, \dots, X_{j_p} 施加一组干预，则变量 X_{j_1}, \dots, X_{j_p} 的值会被固定，且这些变量同其因果父母的直接因果连接将被修改为无，在由新网络结构构成的结构因果模型中，这些因果父母变量将不再对变量 X_{j_1}, \dots, X_{j_p} 的取值造成影响，如图 4.1(b)所示。此外，其它未被干预的变量的概率分布不会改变。

总之，结构因果模型利用因果阶梯可以实现反事实推理，因果阶梯的三个步骤还有另一种更为常用的称呼，也即关联 (association)、干预 (intervention) 和反事实 (Counterfactuals)，其与漂因、干预和预测一

一对应^[31]。前者的提出更多是为了描述反事实推理的层级，结构因果模型通过将反事实推理分为三级，从而说明潜在结果模型以及前述系统辨识方法和生成模型方法均无法突破 $P(x) = P(y)$ ，是因为这些模型和方法仍停留在关联发现的层级，而反事实推理层级则不需要约束 $P(x) = P(y)$ 。

4.1.4 贝叶斯网络与因果网络

可以注意到，在图 4.1(a)中，内生变量之间的因果关系构成了一个有向无环图，并且本文称之为贝叶斯网络，但实际上，这里更合适的称呼是因果网络。J. Pearl 曾在其书^[34]中正式声明，因果网络不同于贝叶斯网络，其主要的区别在于对连边语义的定义上，然而，其随后又在其新书^[31]中重新混淆了这一概念。本文作者在研究结构因果模型的初期也曾困惑于其二者的区别，并且因为经常混用贝叶斯网络和因果网络的概念而遭到审稿人的质疑。因此，该问题是如此之重要，以至于本文需要单独列出一章节以深入探讨该问题。本文的结论是，在图 4.1(a)的结构因果模型中，因果网络和贝叶斯网络都是指有向无环图结构，如果不特别声明具体的连边语义，则二者没有本质区别。

首先，引用 J. Pearl 新书^[31]中的一些名词定义。其一，贝叶斯网络是用于表示概率知识的有向无环图，该书中将贝叶斯网络分为概率贝叶斯网络和因果贝叶斯网络。其二，该书还定义结构因果模型由一组响应函数和一组变量组成，变量包括内生变量和外生变量，由作用在内生变量上的响应函数关系同样构成一个有向无环图，该书称之为因果网络 (causal diagram)。于是该书^[31]陈述：

"The main differences between Bayesian networks and causal diagrams lie in how they are constructed and the uses to which they are put. A Bayesian network is literally nothing more than a compact representation of a huge probability table. The arrows mean only that the probabilities of child nodes are related to the values of parent nodes by a certain formula (the conditional probability tables) and that this relation is sufficient. That is, knowing additional ancestors of the child will not change the formula. Likewise, a missing arrow between any two nodes means that they are independent, once we know the values of their parents. We saw a simple version of this statement earlier, when we discussed the screening-off effect in chains and links. In a chain $A \rightarrow B \rightarrow C$, the missing arrow between A and C means that A and C are independent once we know the values of their parents. Because A has no parents, and the only parent of C is B , it follows that A and C are independent once we know the value of B , which agrees with what we said before."

"If, however, the same diagram has been constructed as a causal diagram, then both the thinking that goes into the construction and the interpretation of the final diagram change. In the construction phase, we need to examine each variable, say C , and ask ourselves which other variables it 'listens' to before choosing its value. The chain structure $A \rightarrow B \rightarrow C$ means that B listens to A only, C listens to B only, and A listens to no one; that is, it is determined by external forces that are not part of our model."

"The response function is the key ingredient that gives SCMs the power to handle counterfactuals. It is implicit in Rubin's potential outcome paradigm but a major point of difference between SCMs and Bayesian networks, including causal Bayesian networks. In a probabilistic Bayesian network, the arrows into Y mean that the probability of Y is governed by the conditional probability tables for Y , given observations of its parent variables. The same is true for causal Bayesian networks, except that the conditional probability tables specify the probability of Y given interventions on the parent variables. Both models specify probabilities for Y , not a specific value of Y . In a structural causal model, there are no conditional probability tables."

从这两段话可以得出以下结论：

1. 概率贝叶斯网络即是贝叶斯网络，因为贝叶斯网络表示的都是概率知识。
2. 因果贝叶斯网络是一类贝叶斯网络，所以其也是一类概率贝叶斯网络。并且，因果贝叶斯网络和概率贝叶斯网络的网络结构相同，其区别主要是对连边语义的定义上。
3. 每一个结构因果模型都会关联一个因果网络，对于图 4.1(a)这样的结构因果模型，其关联的因果网络具有与贝叶斯网络相同的有向无环图结构。
4. 因果贝叶斯网络是一类具有有向无环图结构的因果网络。

因此，图 4.1(a)中的结构因果模型所关联的具有有向无环结构的因果网络实际上就是同样具有有向无环结构的贝叶斯网络，而贝叶斯网络在其中是概率的，但当其被放置在结构因果模型中时，其连边语义就变成了因果的。总之，唯有其有向无环的网络结构是不变的，如图 4.2 所示。

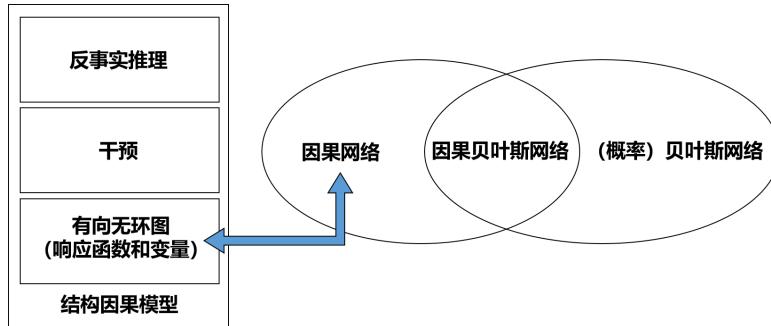


图 4.2: 结构因果模型中贝叶斯网络与因果网络的联系

抛开 J. Pearl 的书不论，该有向无环图在研究发展的不同阶段实际上也有不同的名字。一开始，其被称为贝叶斯网络^[281–285]。后来，其被称为因果图模型 (causal graphical model) 用于强调其因果性^[34, 280, 286–288]。再后来，因果发现研究兴起，其又被称为因果网络^[289–293]。但实际上，从有向无环图视角来看，其网络结构都是表示马尔可夫知识，这种知识可以是概率的，也可以是函数的，但无论如何，其都被解释为因果的。

更重要的是，通过不同的连边语义区分贝叶斯网络和因果网络很容易令人感到困惑。在结构因果模型研究早期，其响应函数以线性函数居多，如图 4.1(a)所示，因此其网络连边的权重可以直接反映线性函数中不同变量的系数，该系数就是因果效应。但是，结构因果模型发展至今，其研究早已不局限于线性形式，而非线性函数的因果效应则无法仅通过一个数值进行描述，比如函数 $\sin(x)$ 和函数 $\sin(2x)$ 的因果效应是无法量化比较的。所以在非线性形式的结构因果模型中，实际上仅仅能够通过判断因果效应是否为零，从而判断因果连边是否存在，但因果连边的权重数值大小其实并不能真正反映因果效应。所以，通过对网络连边语义施以不同定义从而区分因果网络和贝叶斯网络，在非线性形式的结构因果模型中既无可能，也无必要。

因此，在具有图 4.1(a)形式的结构因果模型中，无论其是线性的或是非线性的，因果网络和贝叶斯网络都是指有向无环图结构，如果不特别声明具体的连边语义，则二者没有本质区别。此外，需要特别声明的是，结构因果模型中的因果网络并不一定要采用传统的贝叶斯网络形式，实际上还可以采用有向有环图模型 (Directed Cyclic Graphical)^[294–297]，或者马尔可夫网 (Markov Network)^[298, 299]，或者完全时间图 (Full Time Graph)^[35]，根据以上论述，这些网络模型也可以被放置于结构因果模型中，从而使得其连边具有因果语义，因此也可以是因果网络。所以，由于本文中主要讨论的是建立在形如贝叶斯网络的因果网络之上的结构因果模型，为避免歧义，图 4.1(a)中的结构因果模型均采用贝叶斯网络建模因果关系。

4.1.5 贝叶斯网络的基础假设

如前所述，具有图 4.1(a)形式的结构因果模型都会关联一个贝叶斯网络，该贝叶斯网络描述了不同变量（网络节点）之间的马尔可夫依赖关系，并且需要在其上进行网络结构干预以及推理反事实。因此，结构因果模型建立在贝叶斯网络上，而贝叶斯网络是结构因果模型的基础，所以本文这里讨论贝叶斯网络的基础假设，也即结构因果模型的基础假设。

在此之前，首先需要引出 d-分离的概念，给定一个贝叶斯网络 $G = (\mathbf{V}, \mathbf{E})$ ，d-分离定义如下：

定义 8 (d-分离^[281–285]). 令 \mathbf{X}, \mathbf{Y} 和 \mathbf{Z} 为内生变量 $\mathbf{V} = \{X_1, \dots, X_N\}$ 的三个互斥的子集，并且令 \mathbf{p} 表示从 \mathbf{X} 中节点到 \mathbf{Y} 中节点的任意一条路径（无关方向），于是， \mathbf{Z} 阻隔了路径 \mathbf{p} ，当且仅当存在节点 $v \in \mathbf{p}$ 满足以下其中一条：

1. v 具有 v-结构 (两节点 $a, b \in \mathbf{p}$ 指向 v , 也即 $a \rightarrow v \leftarrow b$), 且 v 及其下游节点不在 \mathbf{Z} 中。
2. v 不具有 v-结构, 且 v 在 \mathbf{Z} 中。

于是, 如果 \mathbf{Z} 阻隔了所有的路径 \mathbf{p} , 则称 \mathbf{Z} d-分离了 \mathbf{X} 和 \mathbf{Y} , 记为 $\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} | \mathbf{Z}$ 。

基于 d-分离, 可以引出马尔可夫性假设 (或因果马尔可夫假设) 和忠实性假设, 定义如下:

假设 7 (马尔可夫性^[281-285]). 概率分布 P 对于贝叶斯网络 \mathcal{G} 是马尔可夫的, 如果对于 V 的任意三个互斥子集 $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ 满足

$$\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} | \mathbf{Z} \Rightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}. \quad (4.8)$$

假设 8 (忠实性^[281-285]). 概率分布 P 对于贝叶斯网络 \mathcal{G} 是忠实的, 如果对于 V 的任意三个互斥子集 $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ 满足

$$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \Rightarrow \mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} | \mathbf{Z}. \quad (4.9)$$

基于这两条假设可以直观地发现, 在观测数据中, 贝叶斯网络 \mathcal{G} 与概率分布 P 的独立性之间一一对应。所以, 如果已知一个贝叶斯网络 \mathcal{G} 满足这两条假设, 那么可以做如下分解:

$$P(X_1, \dots, X_N) = \prod_{i=1}^N P(X_i | \mathbf{Pa}(X_i)), \quad (4.10)$$

这里 $\mathbf{Pa}(X_i)$ 是变量 X_i 在 \mathcal{G} 中的所有直接前驱, 也即因果父母。

此外, 贝叶斯网络通常还需要满足因果充分性假设, 定义如下:

假设 9 (因果充分性^[281-285]). 内生变量集 \mathbf{V} 满足因果充分性, 如果没有任何不可观测的隐变量是 \mathbf{V} 中两个或更多变量的共因。

这意味着, 如果一个贝叶斯网络 (或结构因果模型) 是因果充分的, 那么就不会有其它隐藏路径同时指向 \mathbf{V} 中的两个内生变量, 并且所有的变量信息均已被收集, 足以准确地描述内生变量间的全部因果关系。实际上, 潜在结果模型中的可忽略性假设 (见假设 5) 同样暗含因果充分性, 但因果充分性假设要比可忽略性假设更弱。尽管如此, 因果充分性假设在具体实践时仍然难以完全满足, 正如前文一直强调的, 真实世界系统是十分复杂的, 因此通常难以收集可用于描述一个复杂系统的全部信息。

4.2 潜在结果模型和结构因果模型的根本分歧

4.2.1 可忽略性和后门准则

首先, 从概率统计的角度来说, 潜在结果模型中的可忽略性假设在计算上等价于结构因果模型中的充分性假设和后门准则, 也即当结构因果模型满足充分性假设, 则利用后门准则校正干预后分布可以等价地使潜在结果模型满足可忽略性假设, 从而计算反事实值。而相比之下, 在潜在结果模型中, 要满足可忽略性假设则不是一件容易的事, 其通常需要经过大量的技术性分析和处理。为解释该原理, 这里定义后门准则如下:

定义 9 (后门准则^[34,280,287,288]). 在表 4.1 中, 称一组协变量 \mathbf{X} 满足后门准则, 如果一个贝叶斯网络 \mathcal{G} 建立在包括变量 \mathbf{X}, Y, T 的一组变量上, 且存在连边 $T \rightarrow Y$, 满足

1. \mathbf{X} 中的变量均不在处理变量 T 的下游;

2. \mathbf{X} 阻隔了 T 和 Y 之间的每条含有指向 T 的路径，这些路径称为后门路径。

实际上，给定三个变量 X, Y, T ，其构成的后门路径只有一种情况，如图 4.3 所示。因此，给定一组协变量 \mathbf{X} ，若 \mathbf{X} 满足后门准则，则可以得到

$$\begin{aligned} P(Y|\text{do}(T=1)) &= \sum_{\mathbf{x}} P(Y|\text{do}(T=1), \mathbf{X}=\mathbf{x})P(\mathbf{X}=\mathbf{x}|\text{do}(T=1)) \\ &= \sum_{\mathbf{x}} P(Y|T=1, \mathbf{X}=\mathbf{x})P(\mathbf{X}=\mathbf{x}), \end{aligned} \quad (4.11)$$

这对于 $\text{do}(T=0)$ 同理可得^[34,280,287,288]。

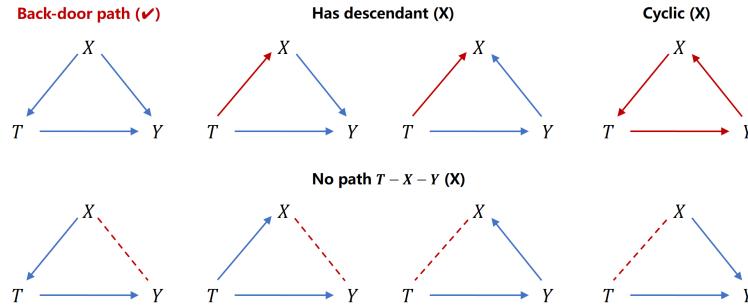


图 4.3: 后门路径示意图（出自文献^[66] 中 Figure 4）

如果该贝叶斯网络 \mathcal{G} 满足因果充分性，也即没有隐变量可以同时影响 \mathbf{X}, Y, T ，则表 4.1 中潜在结果的期望可以计算如下：

$$\begin{aligned} \mathbb{E}[Y|\text{do}(T=1)] &= \sum_y y \times P(Y=y|\text{do}(T=1)) \\ &= \sum_y \sum_x y \times P(Y=y|T=1, \mathbf{X}=\mathbf{x})P(\mathbf{X}=\mathbf{x}) \\ &= \sum_x P(\mathbf{X}=\mathbf{x}) \sum_y y \times P(Y=y|T=1, \mathbf{X}=\mathbf{x}) \\ &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y|T=1, \mathbf{X}]]. \end{aligned} \quad (4.12)$$

这与潜在结果模型的计算结果相同，如下：

$$\begin{aligned} \mathbb{E}[Y(1)] &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y(1)|\mathbf{X}]] \quad /* \text{全期望公式} */ \\ &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y(1)|T=1, \mathbf{X}]] \quad /* \text{可忽略性假设} */ \\ &= \mathbb{E}_{\mathbf{X}} [\mathbb{E}[Y|T=1, \mathbf{X}]] \quad /* \text{一致性假设} */ \end{aligned} \quad (4.13)$$

这对于 $\text{do}(T=0)$ 同理可得。于是对比公式 (4.12) 和公式 (4.13)，可以得到 $\mathbb{E}[Y|\text{do}(T=1)] = \mathbb{E}[Y(1)]$ 和 $\mathbb{E}[Y|\text{do}(T=0)] = \mathbb{E}[Y(0)]$ 。因此在结构因果模型中，平均因果效应（见公式 (4.4)）可以被转换为

$$\text{ACE}(T \rightarrow Y) = \mathbb{E}[Y|\text{do}(T=1)] - \mathbb{E}[Y|\text{do}(T=0)] \quad (4.14)$$

因此，潜在结果模型中的可忽略性假设实际上可以被拆解成结构因果模型中的因果充分性假设和后门准则。相比之下，可忽略性假设在去混杂问题上更具技术性，为了满足该假设，通常需要经过复杂的分层分析^[267,279] 才能去除混杂因子，而这种分层分析在变量数较多时几乎无法实现。而结构因果模型则具备一个贝叶斯网络可以显式地描述因果关系，其根据网络结构就可以轻易找到一组满足后门准则的协变量，从而完成干预后分布的校正，其计算结果与潜在结果模型的反事实推理结果相同。此外如前所述（见假设 9），可忽略性假设实际上暗含了因果充分性假设，而后者要比前者更弱，或者说更易满足。因此，结构因果模型在去混杂问题上要比潜在结果模型更具优势，而结构因果模型面对的真正挑战在于如何准确地发现因果关系。

4.2.2 反事实可证伪性

重新回到图 4.1(b)，如果对潜在结果模型问询“现在已知 $X_1 = 0.5, X_2 = 1, X_3 = 1.5$ ，如果当初 $X_2 = 2$ ，那么现在 $X_3 = ?$ ”，要回答这个问题，那么理论上应该至少收集两个观测样本，这两个样本具有相同的 $X_1 = 0.5$ ，但是不同的 X_2 ，一个 $X_2 = 1$ ，另一个 $X_2 = 2$ ，这样才能满足正值假设（见假设 4）。然后，由一致性假设（见假设 3），通过精确匹配可以推出 X_3 的反事实的值等于那个取 $X_2 = 2$ 的样本中的 X_3 的值。但是这在结构因果模型中却是不同的，通过因果阶梯，即便只有一个观测样本也可能完成反事实推理，见本文章节 4.1.3。那么，我们怎么验证结构因果模型的反事实推理结果正确与否呢？

这是一个十分困难的问题，因为其关系到其推理的反事实能否外推数据。依据定义，反事实值是未观测的（或不可观测的），见表 4.1 中的“?”。而为了验证反事实推理的正确与否，则至少需要两个样本用作比较。一般而言，我们不可能回到过去，然后保持相同的条件重新做另一个对照试验，并且，我们也不可能在不同的时间，保证执行一个具有相同条件的另一个对照试验。因此，如果没有正值假设和一致性假设，那么结构因果模型的反事实推理结果不能被证明，也不能被证伪。这种关于反事实可证伪性的结论在 J. Pearl 的书^[31] 中被模棱两可地否定了，该书认为结构因果模型不需要潜在结果模型中的一致性假设和正值假设，所以根据本文思路，其会推导出结构因果模型的反事实推理不可证伪的结论。

因此，如果有正值假设和一致性假设，那么反事实推理是可证伪的。而在潜在结果模型中，如果没有正值假设，而只有一致性假设，那么也可以推导出一种更弱的反事实可证伪性，在这种情况下，可以通过回归模型推理反事实，但推理的结果却可以被回归模型验证。举个例子，如图 4.4 所示，不妨设只有处理变量 T 和目标变量 Y ，并且观测到样本 $(T, Y) = \{(0, 0.5), (1, 1), (2, 2.5), (3, 2)\}$ ，那么通过线性回归拟合观测样本，可以得到表达式 $Y = 0.5T + 0.5$ 。于是，我们问询“如果当初 $T = 4$ ，那么现在 $Y = ?$ ”，根据回归模型，其反事实推理的结果是 $Y = 2.5$ ，并且该结果可以被回归模型验证。比如我们随意地推理一个值，如果该值与回归模型的推理结果不同，那么该随意推理的值就被证伪了。因此，回归模型描述了观测的边界，超出回归模型划定的边界，才算是真正的外推数据。因此，如果结构因果模型也假设了一致性，那么结构因果模型的反事实推理也就具备了可证伪性，其推理的反事实边界就是回归模型的边界。

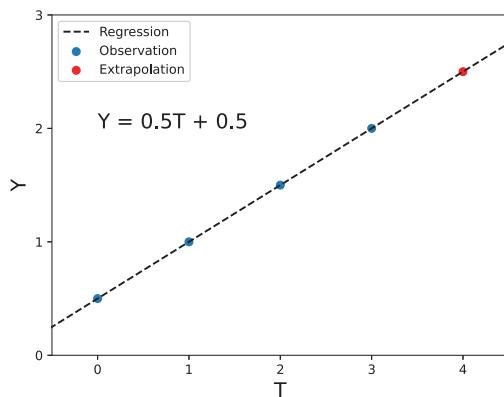


图 4.4：数据外推示意图（出自文献^[66] 中 Figure 5）

此外，还有另一种关于贝叶斯网络的解释，为了构建一个结构因果模型，其需要预设一个贝叶斯网络，但是如何从单样本中发现一个贝叶斯网络呢？这通常是不可能的。因此，一个合理的解释是，其网络结构可以来自于系统的机制，比如物理机制或通讯机制，或者从别的领域引入了网络知识。但是这种说法的问题在于，如前所述，这种对系统机制的理解或外部网络知识的引入，同样是来自于场景数据生成过程，如图 1.2 所示，是因为其在本结构因果模型之外的其它场景数据生成过程中学到了外部知识 g' ，从而才能构建本结构因果模型中的贝叶斯网络。所以，这个问题又被追溯到，在之外的那些场景数据生成过程中，那些外部知识 g' 又是如何被发现的呢？因此，算上本次和之外的场景数据生成过程，实际上至少会有两个及以上的样本参与结构因果模型的反事实推理，所以其反事实推理的可证伪性来源于此。当

然，这只不过是一种保证结构因果模型具备可证伪性的解释，仅从基础假设上来说，即便随机地生成一个贝叶斯网络，结构因果模型也是可以执行反事实推理的，而随机生成网络结构并不会为本次结构因果模型的构建引入任何知识，于是刚才所言的由外部知识带来额外样本的说法就又不成立了。

总之，反事实可证伪性来源于正值假设和一致性假设，基于此，潜在结果模型具备反事实可证伪性。如果仅保留一致性假设，那么潜在结果模型会具备一种更弱的反事实可证伪性，而其反事实推理的边界就是建立在观测数据之上的回归模型的边界，因此其推理的反事实仍然没有超出观测。而由于结构因果模型不做正值假设和一致性假设，因此结构因果模型不具备反事实可证伪性，但也正因如此，其推理的反事实可以超出观测，从而外推数据。因此，反事实可证伪性和数据外推是互斥的关系，可证伪则不算外推，外推则不可证伪。因此，结构因果模型实际上提供了一种思考和推理反事实的方法，其强调推理的过程是反事实的，而不是推理的结果是反事实的，因为其推理的结果在潜在结果模型的定义下并不总是可证伪的。而正相反，潜在结果模型则强调推理的结果是反事实的，这种反事实的结果应当可证伪。这也就是为什么，在基于潜在结果模型的反事实推理研究中常见到对推理准确性的评估，而在基于结构因果模型的反事实推理研究中，则不常见到这种对推理准确性的评估。这就是潜在结果模型和结构因果模型在反事实可证伪性上的根本分歧。

4.2.3 直接因果和间接因果

潜在结果模型和结构因果模型在对直接因果和间接因果的处理上也存在着方法上的不同。首先定义直接因果和间接因果，如下：

定义 10 (直接因果和间接因果). 设存在一条有向的因果路径连通处理变量和目标变量，如果路径上除了此二变量之外还有其它变量，那么该因果路径被称为间接因果，如果没有其他变量，那么该因果路径被称为直接因果。

具体地说，潜在结果模型强调对统计技术的运用，从而保证满足其基础假设，才能够保证反事实推理的准确性，而这种统计技术通常需要大量的人工操作，所以技术性较强。因此，当协变量中既有混杂因子，又有间接因果存在时，这种统计技术就会使问题变得非常复杂。而相对地，结构因果模型则强调对计算机技术的运用，从而自动化地实现反事实推理，因此结构因果模型十分关注反事实推理算法的设计，并通过一些列理论佐证算法计算结果的准确性。与潜在结果模型相比，结构因果模型的一大优势，就是有一个贝叶斯网络可以显式地描述直接因果关系和间接因果关系。

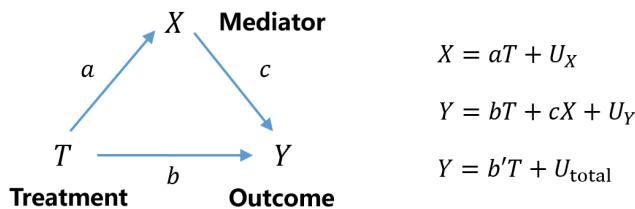


图 4.5: 中介分析示意图^[300]。这里 a, b, c 是系数， U_X, U_Y 是偏置项。 b' 和 U_{total} 分别是总因果效应的系数和偏置项。

在潜在结果模型中，间接因果路径上的变量又被称为中介变量，对于中介变量的分析方法在这些文献^[301-303]中已经有了深入的讨论。但为了尽可能通过图模型可视化潜在结果模型中的中介分析，本文这里引入 Baron-Kenny 模型^[300,304]，如图 4.5 所示， $T \rightarrow Y$ 是直接因果，而 $T \rightarrow X \rightarrow Y$ 是间接因果。为了分析中介因果效应，R. M. Baron 和 D. A. Kenny^[300] 提出了因果步骤法，其构建三个回归模型，假设该回归模型充分地拟合了观测数据，于是中介因果效应存在，如果满足：

1. 系数 a, b, c 显著；
2. $|b| < |b'|$ 。

不过不止这一种判据，其还可以通过检测零假设 $H_0 : b' - b = 0$ 来判断^[305–307]。或者，通过检测零假设 $H_0 : ab = 0$ 来判断^[308,309]。如果回归模型正确建立，那么可以利用回归模型推理反事实，就如图 4.1(a)一样。当然，不论什么方法，其中介分析都基于回归模型，回归模型和系统辨识（见本文第二章）在基本原理和使用方法上几乎是一样的，因此当系统变量数较大时，通过回归模型分析中介因果效应的效率会变得很低，其通常需要反复地拟合多个变量的观测数据并检测其系数的显著程度，过程复杂且计算效率低。并且，回归模型对于非线性案例的中介分析往往难以执行，最重要的原因就是非线性回归模型难以构建，于是同系统辨识遇到的困境一样，如果没有明确的领域知识限制系统基函数的选择，那么采用人工神经网络虽然准确性高，但可解释性低，采用稀疏辨识方法可解释性高，但准确性相对低，很难平衡二者。

相比之下，结构因果模型在变量数较大的情况下则拥有可视化因果关系和高计算效率上的优势。I. Shpitser 和 J. Pearl^[67] 提出了一种递归的多节点干预后分布校正算法，给定一个建立在贝叶斯网络 \mathcal{G} 上的结构因果模型，当对 \mathcal{G} 中的多个节点进行干预时，其干预后分布可以通过递归算法进行校正。本文作者^[66] 也提出了一个递归的多节点干预后分布校正算法，见本文算法 3，本文称其为前向反事实推理算法。需要说明的是，本文作者设计该算法时并未参考文献^[67]，本文作者也是后来经人提醒才发现前人已有了相关工作。经过比较，尽管二者都是基于递归思想，但前向反事实推理算法对递归过程的描述可能更加清楚，本文为此举了一个例子，如图 4.6 所示。本文读者可以将两篇文献对比阅读，也许思想上可以互为补充。

算法 3 中实现第 1 行和第 3 行的代码可以有多种，比如一些深度优先搜索算法和宽度优先搜索算法的变种^[310]，所以对于贝叶斯网络 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ，这两类算法的时间复杂度均为 $\mathcal{O}(|\mathbf{V}| + |\mathbf{E}|)$ 。同时，如果忽略算法 3 中公式 (4.15) 的分解和校正时间，则代码第 9 行的时间复杂度为 $\mathcal{O}(|\mathbf{V}|)$ ，因为算法 3 最多仅需要遍历所有节点，并且该过程实际上可以并行。所以，算法 3 的总时间复杂度为 $\mathcal{O}(|\mathbf{V}|^2 + |\mathbf{V}||\mathbf{E}|)$ 。所以即便贝叶斯网络 \mathcal{G} 规模非常大，算法 3 仍然能在多项式时间内推理反事实，这是结构因果模型的计算效率优势。

算法 3：前向反事实推理 (Forward Counterfactual Inference)

Input: 一个建立在 $\mathbf{V} = \{X_1, \dots, X_N\}$ 之上的贝叶斯网络 \mathcal{G} 。待干预变量 X_{j_1}, \dots, X_{j_p} 和目标变量 X_i ；

Output: 干预后分布 $P(X_i|\text{do}(X_{j_1}), \dots, \text{do}(X_{j_p}))$ ；

- 1 在贝叶斯网络 \mathcal{G} 中搜索从源头 X_{j_1}, \dots, X_{j_p} 到目标 X_i 的全部有向因果路径；
- 2 **if** X_{j_1}, \dots, X_{j_p} 都是 X_i 的直接因果父母 **then**
- 3 在贝叶斯网络 \mathcal{G} 中向后搜索从源头 X_{j_1}, \dots, X_{j_p} 到目标 X_i 的后门路径（共因，见图 4.3）；
- 4 **if** 不存在后门路径 **then**
- 5 **return** $P(X_i|X_{j_1}, \dots, X_{j_p})$ ；
- 6 **else**
- 7 对于所有后门路径，选择一组源头变量的直接前驱，记为 $\mathbf{W} \subset \mathbf{V}$ ，然后依据后门准则（见定义 9），
- 8 **return** $\sum_w P(X_i|X_{j_1}, \dots, X_{j_p}, W)P(W)$ ；
- 9 否则，对于 q 个起始于 X_{j_p} 的因果路径，选择 q 个直接后继的中介变量 $X_{j_{p_1}}, \dots, X_{j_{p_q}}$ ，于是可得一组“源头 - 中介”对，比如 $(X_{j_p}, X_{j_{p_q}})$ ，对 X_{j_1}, \dots, X_{j_p} 均如此做，若无中介变量则忽略该路径；
- 10 于是分解 $P(X_i|\text{do}(X_{j_1}), \dots, \text{do}(X_{j_p}))$ 可得

$$\begin{aligned} P(X_i|\text{do}(X_{j_1}), \dots, \text{do}(X_{j_p})) &= \sum_{x_{j_{1_1}}:j_{p_q}} P(X_i|\text{do}(X_{j_{1_1}}), \dots, \text{do}(X_{j_{2_1}}), \dots, \text{do}(X_{j_{p_q}})) \\ &\quad \times \prod_p \prod_q P(X_{j_{p_q}}|\text{do}(X), X \in \mathbf{Pa}(X_{j_{p_q}}) \subseteq \{X_{j_1:j_p}\}), \end{aligned} \quad (4.15)$$

这里 $\mathbf{Pa}(X_{j_{p_q}})$ 是 $X_{j_{p_q}}$ 的因果父母；

- 11 依据算法 3 递归求解公式 (4.15) 中的每个分布 $P(\cdot|\text{do}(\cdot), \dots, \text{do}(\cdot))$ ；
- 12 **return** $P(X_i|\text{do}(X_{j_1}), \dots, \text{do}(X_{j_p}))$.

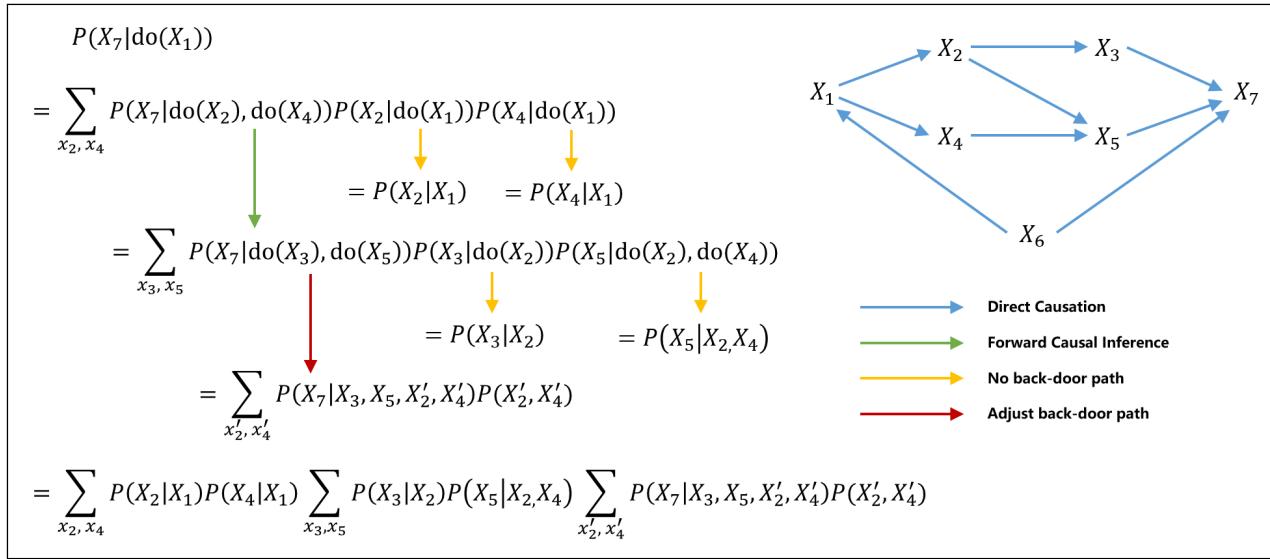


图 4.6: 前向反事实推理算法示例。这里 X_i 和 X'_i 相同, 写成两个符号是为了区分两次递归。

4.3 复杂系统时空因果推断

至此, 本文已经讨论了潜在结果模型和结构因果模型的基本原理, 以及其二者的共通之处和根本分歧, 但仍然缺少对复杂系统中时空因果的建模、理论基础和应用技术等方面的探讨。在复杂系统中, 系统变量之间常有交互, 且这种交互会有时间上的延迟, 可参考本文章节 1.2 和章节 2.2, 本文称这种含有时延的因果交互为时空因果关系, 那么时空因果关系应该如何建模并在其上推理反事实?

为了扩展潜在结果模型在时空因果上的应用, 断点回归 (Regression Discontinuity Design) 方法^[311,312] 和双重差分 (Differences in Differences) 方法^[313,314] 常被用于设计准实验并推断因果关系。然而, 准实验并不是完全的随机实验, 所以该方法可能会引入潜在的估计偏差。而与此同时, 结构因果模型则没有在时空因果推断上形成统一的框架。比如, 有些研究^[297] 利用基于有向无环图的结构因果模型推理反事实, 但有向无环图无法描述时空因果的性质。比如, 有些研究^[35] 设计了完全时间图以描述时空因果, 但却没有建立基于完全时间图的反事实推理框架。比如, 有些研究^[315] 直接将基本的结构因果模型应用在马尔可夫过程上, 然而其使用的贝叶斯网络不仅限制了节点间的空间交互性质, 也忽略了在时间上的高阶马尔可夫性质。总之, 本文在前人研究的基础上, 于本文集中探讨复杂系统中时空因果的建模、理论基础和应用技术, 并尝试建立统一的时空因果推断框架。

4.3.1 贝叶斯网络中的马尔可夫等价关系

贝叶斯网络总是具有有向无环的网络结构, 但该网络结构却并不总是合理的, 比如在建立在变量 $\mathbf{V} = \{X_1, \dots, X_N\}$ 之上的具有高阶马尔可夫性的非线性系统中, 定义如下:

$$X_{i,t} = f_i(X_{j,t-\tau}, j=1, \dots, N, \tau=0, \dots, \tau_{\max}) + \xi_{i,t}, \quad (4.16)$$

这里 $i = 1, \dots, N$, 且 $X_{i,t}$ 表示时刻 t 的变量 X_i , 不同时刻的 X_i 视为不同变量。 $\xi_{i,t}$ 是变量 X_i 在时刻 t 所受影响的加性噪声。 τ_{\max} 是个常数, 表示最大时延。可以发现, 系统公式 (4.16) 是系统公式 (1.1) 的一个离散时间的含有加性噪声的案例, 其具有广泛的研究意义, 而正是在这样一个系统模型中, 贝叶斯网络却无法正确建模。举个例子, 如图 4.7(a) 所示, 其仅有两个变量 X 和 Y , 但是由于其均在 $t-1$ 时刻影响 t 时刻的对方, 从而可以直观地发现其应当互为因果。然而, 贝叶斯网络受到有向无环的约束, 不允许出现因果路径环路, 因此贝叶斯网络无法正确建模图 4.7(a) 这样一个特例。因此在这个特例中, 贝叶斯网络 $X \rightarrow Y$ 和贝叶斯网络 $X \leftarrow Y$ 实际上具有一种等价关系。

T. Verma 和 J. Pearl^[316] 指出, 贝叶斯网络中存在马尔可夫等价关系, 是因为存在一族贝叶斯网络, 其

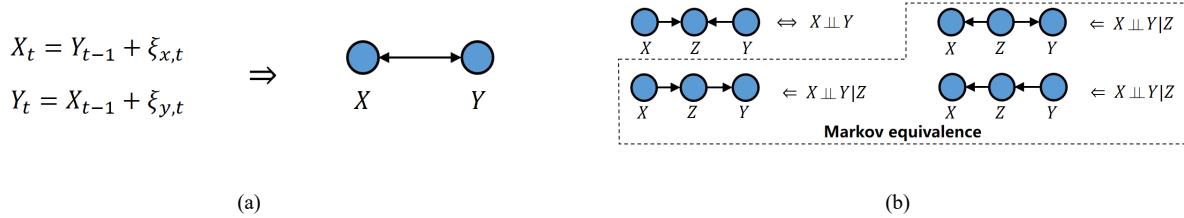


图 4.7: 马尔可夫等价性示意图。这里 X_t 和 Y_t 分别表示在时刻 t 的变量 X 和 Y , ξ 表示噪声项或偏置项。

网络结构不同,但却具有相同的独立性关系。如图 4.7(b)所示,对于任意的三个变量 X 、 Y 和 Z ,如果在观测样本中检测出 $X \perp\!\!\!\perp Y$,则根据忠实性假设(见假设 8)可以推出 $X \rightarrow Z \leftarrow Y$ 。然而,如果在观测样本中检测出 $X \perp\!\!\!\perp Y|Z$ 则能同时推出 $X \leftarrow Z \rightarrow Y$ 、 $X \rightarrow Z \rightarrow Y$ 和 $X \leftarrow Z \leftarrow Y$ 这三种网络结构,并且通常无法区分哪一个结构才是正确的,因而这三种网络结构具有相同的马尔可夫等价关系,又称为马尔可夫等价类。当然,实际的贝叶斯网络中通常不止三个变量,但是图 4.7(b)描述的却是最基本的四种网络结构,由这四种网络结构排列组合成的马尔可夫等价类可能非常庞大。

而在一个如公式(4.16)所示的多变量系统中,如果用贝叶斯网络建模该系统中变量间的因果关系,则会引发三个问题:(1)瞬时因果和一阶因果只能表示其一。由于在系统公式(4.16)中,存在 $\tau = 0$ 和 $\tau = 1$ 两种情况,而贝叶斯网络不具备在时间维度描述因果关系的能力,因此当使用贝叶斯网络描述系统公式(4.16)中的因果关系时,其连边仅能描述两个变量在 $\tau = 0$ 或 $\tau = 1$ 情况下的因果关系中的一个。 $\tau = 0$ 的情况常见于现代控制系统中的控制器设计^[317],而 $\tau = 1$ 的情况则常见于一阶马尔可夫过程,比如文献^[315],均不可忽视。(2)不能描述高阶马尔可夫性,如系统公式(4.16)。 $\tau = 0$ 和 $\tau = 1$ 的情况尚且不能兼顾, $\tau > 1$ 的情况则更不能。(3)不能描述变量间的交互性,如图 4.7(a)所示,当变量间存在含时延的因果交互时,为了满足有向无环约束,必然需要丢弃其中部分有向连边,于是其网络结构损失了这部分变量间的因果交互信息。因此,为了更准确的建模系统公式(4.16)中存在的时空因果关系,传统的贝叶斯网络则不够适用。

4.3.2 时空贝叶斯网络

如图 4.2 所示,结构因果模型建立在一个因果网络上,传统的结构因果模型选择了(因果)贝叶斯网络作为底层的因果网络,但其并不是唯一的选择,如果更换其它的因果网络模型,则会构成不同的结构因果模型。但其真正的难点并不在于更换因果网络模型,而是在于更换了底层的因果网络之后,如何基于该网络进行干预和反事实推理。幸运的是,由于本文提出了前向反事实推理算法(见算法 3),因此给定任意一个有向无环图,在其上进行多节点干预,总能递归地校正干预后分布,从而推理反事实。因此,要想合理地基于结构因果模型建模时空因果关系并推理反事实,需要具备两个要素:(1)有向无环的因果网络;(2)前向反事实推理算法。算法已经具备,那么只剩下新的因果网络需要设计,因此本文于此提出时空贝叶斯网络模型用于建模时空因果关系,其网络结构可以准确描述时空因果关系中带时延的因果交互,并且其网络结构唯一,不存在马尔可夫等价类。

如图 4.8 所示,时空贝叶斯网络定义为一组带有不同时间截的有向无环图,建立在变量集 $\mathbf{V} = \{X_1, \dots, X_N\}$ 上。在贝叶斯网络中,节点并不完全等价于变量 X_i ,而是表示不同时刻的变量 $X_{i,t-\tau}$,因此不同时刻的变量也可以视为不同的变量。因此,基于马尔可夫性假设(见假设 7)和忠实性假设(见假设 8),给定一个时空贝叶斯网络 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ 可以得到如下分解:

$$P(X_{1,t}, \dots, X_{N,t}, X_{1,t-1}, \dots, X_{N,t-T+1}) = \prod_{\tau=0}^{T-1} \prod_{i=1}^N P(X_{i,t-\tau} | \mathbf{Pa}(X_{i,t-\tau})), \quad (4.17)$$

这里时间截取值在范围 $0 \sim T - 1$ 内,表示观测到时间步数为 T 的时序样本。理论上,时空贝叶斯网络允许出现无限的时间步数,但具体实践中不可能实现,因此这里只考虑最大时间步数 T 而不失一般性。于

是自然而然地定义时序假设如下：

假设 10 (时序假设). 在时空贝叶斯网络 \mathcal{G} 中，原因节点（变量）在结果节点（变量）前，或与结果节点（变量）处于同一时刻。

由时序假设其实也可以推出时空贝叶斯网络的有向无环性，该性质不言自明，或可参考文献^[65]中 Proposition 1。此外，为了保证无混杂因子，这里还需要加上因果充分性假设，见假设 9 和本文章节 4.2.1。

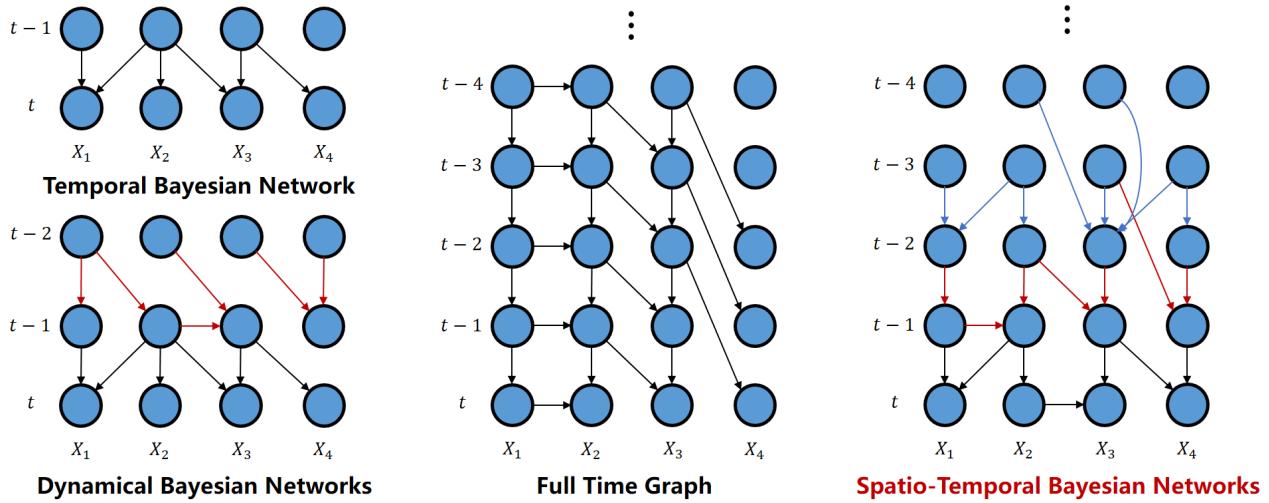


图 4.8: 多种时空因果图模型示意图（出自文献^[66]中 Figure 8）

表 4.2: 多种时空因果图模型比较

	时序贝叶斯网络	动态贝叶斯网络	完全时间图	时空贝叶斯网络
高阶因果关系	✗	✗	✓	✓
瞬时因果关系	✗	✓	✓	✓
非平稳因果关系	✗	✓	✗	✓

时空贝叶斯网络并不是无源之水，实际上类似的网络结构早已提出，如图 4.8 所示。其一是时序贝叶斯网络 (Temporal Bayesian Network)^[298,318]，其用于建模具有一阶马尔可夫性质的系统中的因果关系。其二是动态贝叶斯网络 (Dynamical Bayesian Networks)^[298]，其用一组时序贝叶斯网络以描述具有因果关系切换性质的系统。值得注意的是，动态贝叶斯网络在提出的开始只有两层时序贝叶斯网络，且其中允许瞬时因果关系的出现，当然实际上可以扩展到多层。其三是完全时间图 (Full Time Graph)^[35]，其本质上是一种允许瞬时因果关系的时序贝叶斯网络，但其网络结构不随时间切换。算上本文提出的时空贝叶斯网络，该四种贝叶斯网络的对比见表 4.2，可以发现时空贝叶斯网络具有更加广义的网络结构，可以涵盖另外三种贝叶斯网络。此外，由于时空贝叶斯网络完全由多个局部的有向无环结构拼接而成，所以其也同样具有全局的有向无环结构，因此，基于时空贝叶斯网络的结构因果模型可以通过前向反事实推理算法进行反事实推理。

4.3.3 时空因果等价性

本文章节 4.1.4 已经论证了贝叶斯网络中的因果性，这里同样从格兰杰因果和信息转移的角度论证时空贝叶斯网络中的因果性，从而证明在时空贝叶斯网络中，其网络连边存在着多重因果含义，且其不同含义之间构成等价，本文称之为时空因果等价性。

首先，格兰杰因果^[319] 是一种用于建模时空因果的基本模型，其将时空因果关系定义为一个变量对另一个变量在时间上的可预测性。可预测性^[320,321] 定义如下：

$$\begin{aligned}\mathcal{F}_{Y \rightarrow X} &= \log \left[\frac{\text{Var}(\xi_t)}{\text{Var}(\xi'_t)} \right], \\ \xi_t &= X_t - \sum_{\tau=1}^{\tau_{\max}} X_{t-\tau} w_{t-\tau}, \\ \xi'_t &= X_t - \sum_{\tau=1}^{\tau_{\max}} X_{t-\tau} w_{t-\tau} - \sum_{\tau=1}^{\tau_{\max}} Y_{t-\tau} u_{t-\tau},\end{aligned}\tag{4.18}$$

这里 $X_{t-\tau}$ 和 $Y_{t-\tau}$ 分别是变量 X 和变量 Y 在时刻 $t-\tau$ 的状态变量，注意这里一个变量在不同时刻可以被视为不同的变量，用下标时刻 $t-\tau$ 加以区分。此外， ξ 和 ξ' 是噪声项或偏置项， w 和 u 是系数，同样用下标时刻 $t-\tau$ 加以区分。 τ_{\max} 是常数，表示最大时延。不妨设一共 K 个观测样本，因此 $t = \tau_{\max} + 1, \dots, K$ ，注意这里会损失 τ_{\max} 个样本。并且， $\text{Var}(\cdot)$ 描述了这 K 个样本的方差。由此可见，可预测性 $\mathcal{F}_{Y \rightarrow X}$ 实际上是指变量 Y 对变量 X 的不确定性的降低程度。

公式 (4.18) 中主要利用线性自回归模型定义格兰杰因果，而非线性的格兰杰因果则较为复杂，尤其是在公式 (1.1) 这样的多变量系统中。原因之一是，非线性自回归模型往往难以建模。原因之二是，多变量情况下，格兰杰因果需要对所有变量逐点对地做回归分析从而判断两个变量之间是否有时空因果关系，这种因果关系发现方法十分低效。为了解决这个问题，L. Barnett 和 T. Bossomaier^[322] 利用对数似然率量化转移熵，从而证明双变量情况下转移熵和可预测性等价。转移熵^[323] 是一种用于衡量两个系统变量之间信息流动的非对称度量工具，定义如下：

$$\begin{aligned}\mathcal{T}_{Y \rightarrow X} &= H(X_t | X_{t-1}^{(\tau_{\max})}) - H(X_t | X_{t-1}^{(\tau_{\max})}, Y_{t-1}^{(\tau_{\max})}) \\ &= -\frac{1}{K - \tau_{\max}} \sum_{t=\tau_{\max}+1}^K \log \frac{f(x_{t-1}^{(\tau_{\max})}; \boldsymbol{\theta}_0)}{f(x_{t-1}^{(\tau_{\max})}, y_{t-1}^{(\tau_{\max})}; \boldsymbol{\theta})} \\ &= -\frac{1}{K - \tau_{\max}} \log \frac{\mathcal{L}(\hat{\boldsymbol{\theta}}_0 | X, Y)}{\mathcal{L}(\hat{\boldsymbol{\theta}} | X, Y)} \\ &= \mathcal{F}_{Y \rightarrow X}.\end{aligned}\tag{4.19}$$

这里 $X_{t-1}^{(\tau_{\max})} = \{X_{t-1}, \dots, X_{t-\tau_{\max}}\}$ ，且 $x_{t-1}^{(\tau_{\max})}$ 是其状态值， $Y_{t-1}^{(\tau_{\max})}$ 和 $y_{t-1}^{(\tau_{\max})}$ 可同理定义。此外， $H(Y|X) = -\sum_{x,y} P(x,y) \log P(y|x)$ 和 $f(\cdot|\boldsymbol{\theta})$ 是由 $\boldsymbol{\theta}$ 参数化的非线性函数。 $\mathcal{L}(\cdot|X, Y)$ 是变量 X 和 Y 的概率似然函数。

$\boldsymbol{\theta} \in \Theta$ ，且 Θ 是全参数构成的集合。 $\boldsymbol{\theta}_0 \in \Theta_0 = \{\boldsymbol{\theta} \in \Theta | f(x_{t-1}^{(\tau_{\max})}, y_{t-1}^{(\tau_{\max})}; \boldsymbol{\theta}) \text{ 不依赖于 } y_{t-1}^{(\tau_{\max})}\}$ 。 $\hat{\boldsymbol{\theta}}_0$ 和 $\hat{\boldsymbol{\theta}}$ 分别是对应参数的估计值。由此可见， $\mathcal{T}_{Y \rightarrow X}$ 通过对数似然率量化了从变量 Y 到变量 X 的信息转移和变量 Y 对变量 X 的不确定性的降低程度，且二者等价。

因此，综合公式 (4.19) 和系统公式 (4.16)，可以发现时空贝叶斯网络的连边具有因果含义，其含义包括了马尔可夫性（d-分离，见定义 8）、变量间的独立性、可预测性和信息转移。于是，不妨设有一个建立在系统公式 (4.16) 之上的时空贝叶斯网络 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ，于是可得两条引理，如下：

引理 4.3.1. 在时空贝叶斯网络 \mathcal{G} 中，对于任意的变量 X 和变量 Y ，以及一组变量 \mathbf{Z} ，设 Y 和 \mathbf{Z} 在 X 前发生或同时刻发生，则 $X \perp\!\!\!\perp_{\mathcal{G}} Y | \mathbf{Z}$ 当且仅当转移熵 $\mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = 0$ 。

证明. 对于变量 X 和变量 Y ，由马尔可夫性假设可得 $X \perp\!\!\!\perp_{\mathcal{G}} Y | \mathbf{Z} \Rightarrow X \perp\!\!\!\perp Y | \mathbf{Z}$ ，因此转移熵 $\mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = H(X | \mathbf{Z}) - H(X | Y, \mathbf{Z}) = 0$ 。另一方面，如果 $\mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = 0$ ，则可得 $X \perp\!\!\!\perp Y | \mathbf{Z}$ ，进而由忠实性假设推出 $X \perp\!\!\!\perp_{\mathcal{G}} Y | \mathbf{Z}$ ，证毕。□

引理 4.3.2. 在时空贝叶斯网络 \mathcal{G} 中，对于任意的变量 X 和变量 Y ，以及一组变量 \mathbf{Z} ，设 Y 和 \mathbf{Z} 在 X 前发生或同时刻发生，则转移熵 $\mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = 0$ 当且仅当可预测性 $\mathcal{F}_{Y \rightarrow X | \mathbf{Z}} = 0$ 。

证明. 不妨设给定关于 X 、 Y 和 \mathbf{Z} 的 K 个联合样本，记为 $(x^{(K)}, y^{(K)}, \mathbf{z}^{(K)})$ ，其收集于系统公式 (4.16)，这里 $x^{(K)} = \{x_1, \dots, x_K\}$ ，对 $y^{(K)}$ 和 $\mathbf{z}^{(K)}$ 也如此定义，这些联合样本根据下标时间戳一一对应。并且，这里假设这些联合样本满足遍历性，见假设 1，也即该联合样本构成的系统状态轨迹随时间访问了系统状态空间中的所有区域。于是定义转移熵如下：

$$\mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = H(X | \mathbf{Z}) - H(X | Y, \mathbf{Z}). \quad (4.20)$$

同时，定义用 $\boldsymbol{\theta}$ 参数化的预测模型 f ，如下：

$$f(x | y, \mathbf{z}; \boldsymbol{\theta}) = P(X | Y, \mathbf{Z}; \boldsymbol{\theta}). \quad (4.21)$$

这里 $P(\cdot | \cdot; \boldsymbol{\theta})$ 是参数化的条件概率分布。并且，假设预测模型 f 的参数是唯一可辨识的，也即对于 $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ ，则 $f(\cdot | \cdot; \boldsymbol{\theta}_1) \neq f(\cdot | \cdot; \boldsymbol{\theta}_2)$ 。因此，存在唯一的正确参数 $\boldsymbol{\theta}^*$ 满足

$$f(x | y, \mathbf{z}; \boldsymbol{\theta}^*) = P(X | Y, \mathbf{Z}; \boldsymbol{\theta}^*) = P(X | Y, \mathbf{Z}). \quad (4.22)$$

于是，由于遍历性假设，且联合样本根据下标时间戳一一对应，可知对于 $t = 1, \dots, K$ ，预测模型 f 中样本 x_t 仅取决于样本 y_t 和 \mathbf{z}_t ，于是可得关于参数 $\boldsymbol{\theta}$ 的概率似然函数，如下：

$$\begin{aligned} & \mathcal{L}_K(\boldsymbol{\theta} | x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}) \\ &= P(x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}; \boldsymbol{\theta}) \\ &= P(x_K | x^{(K-1)}, y^{(K)}, \mathbf{z}^{(K)}; \boldsymbol{\theta}) P(y_K, \mathbf{z}_K | x^{(K-1)}, y^{(K-1)}, \mathbf{z}^{(K-1)}; \boldsymbol{\theta}) P(x^{(K-1)}, y^{(K-1)}, \mathbf{z}^{(K-1)}; \boldsymbol{\theta}) \quad (4.23) \\ &= f(x_K | y_K, \mathbf{z}_K; \boldsymbol{\theta}) q(y_K, \mathbf{z}_K) \mathcal{L}_{K-1} \\ &= \prod_{t=1}^K f(x_t | y_t, \mathbf{z}_t; \boldsymbol{\theta}) q(y_t, \mathbf{z}_t), \end{aligned}$$

这里 $P(x_t | x^{(t-1)}, y^{(t)}, \mathbf{z}^{(t)}; \boldsymbol{\theta}) = f(x_t | x^{(t-1)}, y^{(t)}, \mathbf{z}^{(t)}; \boldsymbol{\theta}) = f(x_t | y_t, \mathbf{z}_t; \boldsymbol{\theta})$ 。此外，参数 $\boldsymbol{\theta}$ 仅影响变量 X ，于是 $P(y_t, \mathbf{z}_t | x^{(t-1)}, y^{(t-1)}, \mathbf{z}^{(t-1)}; \boldsymbol{\theta})$ 可以被简化为 $q(y_t, \mathbf{z}_t)$ 。 $q(y_t, \mathbf{z}_t)$ 几乎处处不等于零，且与参数 $\boldsymbol{\theta}$ 无关。于是，通过最大化概率似然函数 \mathcal{L}_K 的平均对数，可得

$$\begin{aligned} \ell(\boldsymbol{\theta} | x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}) &= \frac{1}{K} \log \mathcal{L}_K(\boldsymbol{\theta} | x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}) \\ &\propto \frac{1}{K} \sum_{t=1}^K \log f(x_t | y_t, \mathbf{z}_t; \boldsymbol{\theta}). \end{aligned} \quad (4.24)$$

因此当 $K \rightarrow \infty$ ，由 Birkhoff 遍历定理（见定理 1.2.1）可得

$$\ell \xrightarrow{\text{a.s.}} \mathbb{E}[\log f(x | y, \mathbf{z}; \boldsymbol{\theta})], \quad (4.25)$$

进而可得

$$\mathbb{E}[\log f(x | y, \mathbf{z}; \boldsymbol{\theta})] \xrightarrow{\text{a.s.}} -H(X | Y, \mathbf{Z}). \quad (4.26)$$

于是可以定义一个嵌套的零模型 $H_0 : \boldsymbol{\theta} \in \Theta_0$ ，这里 $\Theta_0 = \{\boldsymbol{\theta} \in \Theta | \text{给定 } \mathbf{Z}, X \text{ 条件独立于 } Y\}$ ，并且 $\Theta_0 \subseteq \Theta$ ， Θ 是全参数集。于是可得似然率，如下：

$$\Lambda(x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}) = \frac{\mathcal{L}_K(\hat{\boldsymbol{\theta}}_0 | x^{(K)}, y^{(K)}, \mathbf{z}^{(K)})}{\mathcal{L}_K(\hat{\boldsymbol{\theta}} | x^{(K)}, y^{(K)}, \mathbf{z}^{(K)})}, \quad (4.27)$$

这里 $\hat{\boldsymbol{\theta}}_0$ 和 $\hat{\boldsymbol{\theta}}$ 分别是对 Θ_0 和 Θ 的最大似然估计结果。可以直观发现， $\Lambda(x^{(K)}, y^{(K)}, \mathbf{z}^{(K)})$ 反映了给定 \mathbf{Z} ，待预测变量 X 对变量 Y 的依赖程度，因此可以定义可预测性如下：

$$\mathcal{F}_{Y \rightarrow X | \mathbf{Z}} = -\frac{1}{K} \log \Lambda(x^{(K)}, y^{(K)}, \mathbf{z}^{(K)}), \quad (4.28)$$

这里 $\mathcal{F}_{Y \rightarrow X | \mathbf{Z}} \in [0, +\infty]$ 。于是依据转移熵公式 (4.20)、公式 (4.25) 和公式 (4.26)，当 $K \rightarrow \infty$ ，可得 $\mathcal{F}_{Y \rightarrow X | \mathbf{Z}} \xrightarrow{\text{a.s.}} \mathcal{T}_{Y \rightarrow X | \mathbf{Z}}$ ，证毕。 \square

由此二引理可得时空因果等价性定理，如下：

定理 4.3.3 (时空因果等价性). 在时空贝叶斯网络 \mathcal{G} 中，对于任意的变量 X 和变量 Y ，以及一组变量 \mathbf{Z} ，设 Y 和 \mathbf{Z} 在 X 前发生或同时刻发生，则

$$X \perp\!\!\!\perp_{\mathcal{G}} Y | \mathbf{Z} \Leftrightarrow \mathcal{T}_{Y \rightarrow X | \mathbf{Z}} = 0 \Leftrightarrow \mathcal{F}_{Y \rightarrow X | \mathbf{Z}} = 0. \quad (4.29)$$

因此，时空贝叶斯网络 \mathcal{G} 也是一种因果网络，其连边的因果性由格兰杰因果赋予，同时其因果路径还描述了信息转移的路径。

4.3.4 时空因果唯一性

结构因果模型在推理反事实时，需要经过关联、干预、反事实推理等步骤（可参考本文章节 4.1.3），其中的关联部分就涉及回溯因果关系并构建（因果）贝叶斯网络等内容。然而，如图 4.7 所示，贝叶斯网络中存在马尔可夫等价类，所以其从观测样本中回溯的贝叶斯网络可能存在多种等价表示，也即网络结构不唯一，最后会导致在其上推理的反事实结果不唯一。而在时空贝叶斯网络中，当假设时空贝叶斯网络中不存在瞬时因果时，时空贝叶斯网络的网络结构是唯一的，本文称之为时空因果唯一性，本文从信息路径阻隔的角度为时空因果的唯一性提供解释。从图 4.7(b) 中可以看出，在传统的贝叶斯网络中存在马尔可夫等价类，其原因之一是因为条件独立性往往对应着分叉结构和链式结构，从而当从观测样本中检测出条件独立性时，则无法分辨其网络结构是链式的还是分叉的。但时空贝叶斯网络由一组有向无环的子网络结构组成，而在这些子网络结构中没有链式结构，比如，如果有链式结构 $X \rightarrow Z \rightarrow Y$ 被检测出，则由时空因果等价性（见定理 4.3.3）可知 $X \perp\!\!\!\perp_{\mathcal{G}} Y | Z \Rightarrow \mathcal{T}_{Y \rightarrow X | Z} = 0 \Rightarrow \mathcal{F}_{Y \rightarrow X | Z} = 0$ ，所以当观测到变量 Z 时，变量 X 对变量 Y 不具有任何的可预测性，从而 $X \rightarrow Z$ 就可以省略，或者说 $X \rightarrow Z$ 其实是另一个子有向无环网络中的连边，不在当前时刻的有向无环网络中。从而每一个子有向无环网络均可以由独立性和条件独立性一一对应地实现表示，于是每一个子有向无环网络均可以唯一表示，从而时空贝叶斯网络具有唯一的网络结构表示，没有马尔可夫等价类。本文从以下一系列命题证明这一结论。

首先，定义马尔可夫毯 (Markov blanket) 如下：

定义 11 (马尔可夫毯). 对于一个有向无环图 $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ ，以及一个目标变量 $X \in \mathbf{V} = \{X_1, \dots, X_N\}$ ，如果存在变量集 $\mathbf{M} \subset \mathbf{V}$ 满足

$$X \perp\!\!\!\perp_{\mathcal{G}} \mathbf{V} \setminus (\mathbf{M} \cup \{X\}) | \mathbf{M}, \quad (4.30)$$

则最小的变量集 \mathbf{M} 称为马尔可夫毯，记为 $\mathbf{M}(X)$ ，其包含了对于变量 X 的所有预测信息。

如果存在马尔可夫等价类，则因果方向很难正确辨识，因为正向和反向都可能被辨识出来，但只有一个正确的。因此，在有向无环图 \mathcal{G} 中，目标变量 X 的后继节点也可能被包括进 X 的马尔可夫毯 $\mathbf{M}(X)$ 中，也即其后继节点也可以为预测变量 X 提供预测信息，从而导致 $\mathbf{Pa}(X) \subseteq \mathbf{M}(X)$ ，这通常并不是我们想要的。而不存在瞬时因果的时空贝叶斯网络则具备 $\mathbf{Pa}(X) = \mathbf{M}(X)$ 的性质，这才是我们通常想要的，由此才可以仅依靠观察贝叶斯网络中变量 X 的因果父母以设计预测模型，而无需考虑其它更多变量。

为了证明不存在瞬时因果的时空贝叶斯网络具备 $\mathbf{Pa}(X) = \mathbf{M}(X)$ 的性质，本文引出命题如下：

命题 4.3.4. $\forall Y \in \mathbf{Pa}(X)$ ，变量 Y 满足 $\mathcal{F}_{Y \rightarrow X | \mathbf{Pa}(X) \setminus \{Y\}} > 0$.

证明. 对于任意变量 $Y \in \mathbf{Pa}(X)$ ，没有任何变量集可以 d-分离变量 X 和变量 Y ，所以根据时空因果等价性定理（见定理 4.3.3），由 $Y \not\perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{Pa}(X) \setminus \{Y\}$ 可以推出 $\mathcal{F}_{Y \rightarrow X | \mathbf{Pa}(X) \setminus \{Y\}} > 0$. \square

命题 4.3.5. $\forall Y$ 在变量 X 之前，且 $Y \notin \mathbf{Pa}(X)$ ，则变量 Y 满足 $\mathcal{F}_{Y \rightarrow X | \mathbf{Pa}(X)} = 0$.

证明. 在时空贝叶斯网络 \mathcal{G} 中, 对于任意发生在变量 X 之前的变量 Y , 且 $Y \notin \text{Pa}(X)$, 可得 $Y \perp\!\!\!\perp_{\mathcal{G}} X | \text{Pa}(X)$, 因为所有从变量 Y 到变量 X 的信息路径都被 $\text{Pa}(X)$ 阻隔。于是可得 $Y \perp\!\!\!\perp_{\mathcal{G}} X | \text{Pa}(X) \setminus \{Y\}$, 因为 $\text{Pa}(X) = \text{Pa}(X) \setminus \{Y\}$ 。因此根据时空因果等价性定理(见定理 4.3.3), 可得 $\mathcal{F}_{Y \rightarrow X | \text{Pa}(X) \setminus \{Y\}} = 0$ 。□

于是易得如下推论:

推论 4.3.6. $\forall Y$ 在变量 X 之前, 则 $Y \notin \text{Pa}(X)$ 当且仅当 $\mathcal{F}_{Y \rightarrow X | \text{Pa}(X) \setminus \{Y\}} = 0$.

推论 4.3.6 解释了时空贝叶斯网络中的信息路径阻隔原理, 也即变量 X 的因果父母 $\text{Pa}(X)$ 阻隔了所有从其它非因果父母变量出发并最终指向变量 X 的因果路径。进而可得如下命题:

命题 4.3.7. 对于任意嵌套的变量集 $\mathbf{M} \supseteq \text{Pa}(X)$, 并且所有 \mathbf{M} 中的变量都在变量 X 前, 于是 $\forall Y \in \mathbf{V} \setminus \mathbf{M} \cup \{X\}$, 有 $Y \perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{M}$, 也即 $\mathcal{F}_{Y \rightarrow X | \mathbf{M}} = 0$ 。

证明. 令 $\mathbf{M} \supseteq \text{Pa}(X)$, 并且所有 \mathbf{M} 中的变量都在变量 X 前。对任意的 $Y \in \mathbf{V} \setminus \mathbf{M} \cup \{X\}$, 可得 $Y \notin \text{Pa}(X)$ 。于是根据命题 4.3.5, 可得 $\mathcal{F}_{Y \rightarrow X | \text{Pa}(X)} = 0$, 进而根据时空因果等价性定理, 可得 $Y \perp\!\!\!\perp_{\mathcal{G}} X | \text{Pa}(X)$ 。这意味着 $\text{Pa}(X)$ 阻隔了变量 X 和变量 Y 之间的所有路径, 如图 4.9 所示。又因为 $\mathbf{M} \supseteq \text{Pa}(X)$, 所以 \mathbf{M} 也阻隔了变量 X 和变量 Y 之间的所有路径。因此, $Y \perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{M}$ 。□

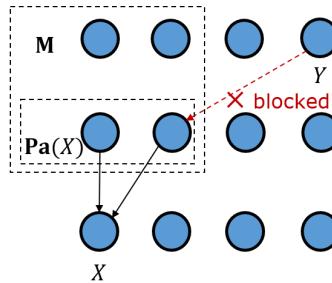


图 4.9: 命题 4.3.7 示意图 (出自文献^[65] 中 Figure 3(a))。对于任意的 $\mathbf{M} \supseteq \text{Pa}(X)$, $\forall Y \in \mathbf{V} \setminus \mathbf{M} \cup \{X\}$, \mathbf{M} 阻隔了所有从变量 Y 到变量 X 的信息路径。

于是仅需要少量证明可得如下推论:

推论 4.3.8. 在时空贝叶斯网络 \mathcal{G} 中, 对任意变量 X 均满足 $\text{Pa}(X) = \mathbf{M}(X)$ 。

证明. 可以分类讨论。首先, 不妨设 $\mathbf{M}(X)$ 是变量 X 的马尔可夫毯。于是一方面, 依据命题 4.3.7, 如果 $\text{Pa}(X) \subseteq \mathbf{M}(X)$, 则 $\text{Pa}(X) = \mathbf{M}(X)$ 显然成立, 如图 4.9 所示, $\mathbf{M}(X)$ 是命题 4.3.7 中 \mathbf{M} 的最小实现。另一方面, 如果 $\text{Pa}(X) \supset \mathbf{M}(X)$, 于是 $\exists Y$ 在变量 X 前, 且 $Y \in \text{Pa}(X)$, 但是 $Y \notin \mathbf{M}(X)$ 。于是可得 $Y \not\perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{M}(X)$, 因为存在从变量 Y 到变量 X 的直接连边, 所以 $\mathbf{M}(X)$ 不能 d 分离它们, 如图 4.10 所示。然而, 由于 $\mathbf{M}(X)$ 是变量 X 的马尔可夫毯, 所以根据定义 11 可得 $Y \perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{M}(X)$, 于是二者矛盾, 因此只有 $\text{Pa}(X) = \mathbf{M}(X)$ 这一种情况。□

因此, 在时空贝叶斯网络 \mathcal{G} 中, 变量 X 的预测信息仅取决于其因果父母 $\text{Pa}(X)$ 。并且, 易知同时刻的不同变量的因果父母可以并行辨识, 因为假设了该贝叶斯网络 \mathcal{G} 中不会出现瞬时因果关系, 所以同时刻变量之间没有连边, 一个变量也不会为另一个同时刻变量提供预测信息, 由此可以引出时空贝叶斯网络的可分解性, 如下:

定理 4.3.9 (时空因果可分解性). 当时空贝叶斯网络 \mathcal{G} 中不存在瞬时因果时, 其对于时刻 t 的变量 $X_{i,t}, i = 1, \dots, N$, 可以被分解为 N 个子图 $\mathcal{G}_{i,t} = (\mathbf{V}, \mathbf{E}_{i,t})$, 于是 \mathcal{G} 中的 d 分离与这些子图 $\mathcal{G}_{i,t}$ 中的 d 分离相同, 也即 $\perp\!\!\!\perp_{\mathcal{G}} \iff \perp\!\!\!\perp_{\mathcal{G}_{i,t}}$ 。

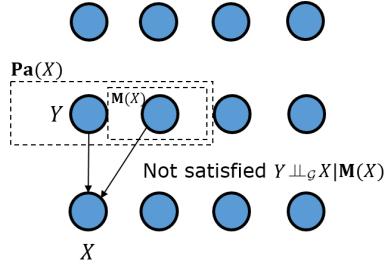


图 4.10: 推论 4.3.8 示意图 (出自文献^[65] 中 Figure 3(b))。如果 $\mathbf{Pa}(X) \supset \mathbf{M}(X)$, 则 $\exists Y \in \mathbf{Pa}(X)$, 但 $Y \notin \mathbf{M}(X)$, 于是可得 $Y \not\perp\!\!\!\perp_{\mathcal{G}} X | \mathbf{M}(X)$ 。

证明. 根据推论 4.3.8, 预测 $X_{i,t}, i = 1, \dots, N$ 仅取决于其因果父母 $\mathbf{Pa}(X_{i,t})$ 。并且, 由于该时空贝叶斯网络 \mathcal{G} 中没有瞬时因果关系, 所以这些同在时刻 t 的变量间没有连接, 此外也没有连接可以从时刻 t 回到过去。由此可知: (1) 对于时刻 t 的任意一个变量, 比如 $X_{1,t}$, 其它变量如 $X_{2,t}, \dots, X_{N,t}$ 不会出现 在集合 $\mathbf{Pa}(X_{1,t})$ 中。(2) 对于时刻 t 的任意一个变量, 比如不存在任何既经过变量 $X_{2,t}, \dots, X_{N,t}$ 且最终指向 $X_{1,t}$ 的因果路径。所以, 在时空贝叶斯网络 \mathcal{G} 中, 当仅关注变量 $X_{1,t}$ 的因果父母时, 包含变量 $X_{2,t}, \dots, X_{N,t}$ 的因果连边及相关因果路径都可以被忽略或隐藏, 于是可以得到一个关于变量 $X_{1,t}$ 的子图 $\mathcal{G}_{1,t}$, 这对于变量 $X_{2,t}, \dots, X_{N,t}$ 同理, 且 $\mathcal{G}_{2,t}, \dots, \mathcal{G}_{N,t}$ 同样分别可得。所以, 时空贝叶斯网络 \mathcal{G} 对于时刻 t 的变量 $X_{i,t}, i = 1, \dots, N$, 可以被分解为 N 个子图 $\mathcal{G}_{i,t} = (\mathbf{V}, \mathbf{E}_{i,t})$, 这里 $\mathbf{E}_{i,t}$ 是由 $\mathbf{Pa}(X_{i,t})$ 导出的指向变量 $X_{i,t}$ 的直接因果连边构成的集合。所以, \mathcal{G} 在时刻 t 也可以理解为由子图 $\mathcal{G}_{1,t}, \dots, \mathcal{G}_{N,t}$ 线性组合而成, 如图 4.11 所示。□

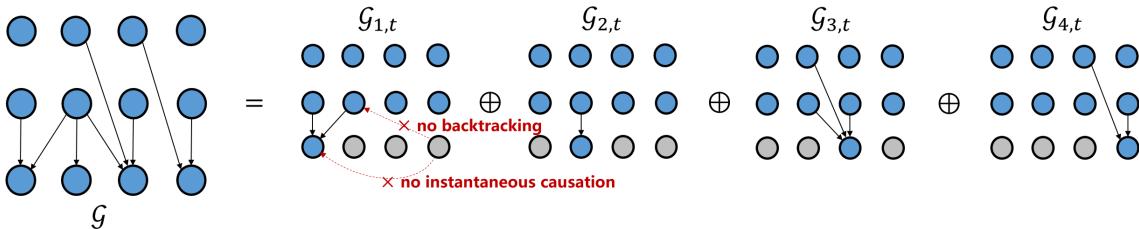


图 4.11: 时空贝叶斯网络可分解性示意图 (出自文献^[65] 中 Figure 4)。这里, \oplus 表示子图的线性组合。

通过时空贝叶斯网络的可分解性, 最终可得其唯一性定理, 如下:

定理 4.3.10 (时空因果唯一性). 当时空贝叶斯网络 \mathcal{G} 中不存在瞬时因果时, 其网络结构可唯一表示。

证明. 根据时空因果可分解性定理 (见定理 4.3.9), 可以得到时空贝叶斯网络 \mathcal{G} 的 N 个子图 $\mathcal{G}_{i,t}$ 。并且, 在子图 $\mathcal{G}_{i,t}$ 中, 变量 $X_{i,t}$ 的因果父母唯一, 可以先证明这一结论。

令 X 表示任意目标变量 $X_{i,t}, i = 1, \dots, N$, 假设其因果父母不唯一, 于是可以令 $\mathbf{Pa}(X)$ 和 $\widetilde{\mathbf{Pa}}(X)$ 分别表示变量 X 在子图 $\mathcal{G}_{i,t}$ 和子图 $\widetilde{\mathcal{G}}_{i,t}$ 中的因果父母。于是, $\forall Y \in \widetilde{\mathbf{Pa}}(X)$, 没有一个不包含变量 X 和变量 Y 的变量集 Z 可以满足 $Y \perp\!\!\!\perp_{\widetilde{\mathcal{G}}_{i,t}} X | Z$ 。所以, $\mathbf{Pa}(X) \setminus \{Y\}$ 作为一个可能出现的变量集 Z , 其不满足 $Y \perp\!\!\!\perp_{\widetilde{\mathcal{G}}_{i,t}} X | \mathbf{Pa}(X) \setminus \{Y\}$ 。同时, 由时空因果可分解性定理可得 $\perp\!\!\!\perp_{\widetilde{\mathcal{G}}_{i,t}} \Rightarrow \perp\!\!\!\perp_{\mathcal{G}} \Rightarrow \perp\!\!\!\perp_{\mathcal{G}_{i,t}}$ 。综合二者, 进而可从 $Y \not\perp\!\!\!\perp_{\widetilde{\mathcal{G}}_{i,t}} X | \mathbf{Pa}(X) \setminus \{Y\}$ 推出 $Y \not\perp\!\!\!\perp_{\mathcal{G}_{i,t}} X | \mathbf{Pa}(X) \setminus \{Y\}$ 。因此, 根据推论 4.3.6 和因果等价性定理 (见定理 4.3.3), 可得 $Y \in \mathbf{Pa}(X)$ 。所以, $\widetilde{\mathbf{Pa}}(X) \subseteq \mathbf{Pa}(X)$, 反过来同理可得 $\widetilde{\mathbf{Pa}}(X) \supseteq \mathbf{Pa}(X)$, 所以 $\mathbf{Pa}_X = \widetilde{\mathbf{Pa}}_X$, 并且 $\mathcal{G}_{i,t} = \widetilde{\mathcal{G}}_{i,t}$ 。

因此, 当时空贝叶斯网络 \mathcal{G} 中不存在瞬时因果时, 其在时刻 t 可以唯一地分解为 N 个子图 $\mathcal{G}_{i,t}$, 并且反过来, \mathcal{G} 也可以由 N 个唯一表示的子图 $\mathcal{G}_{i,t}$ 线性组合得到, 所以时空贝叶斯网络 \mathcal{G} 具有时空因果唯一性, 其中不存在马尔可夫等价类。□

4.4 基于高阶因果熵的唯一因果网络发现

4.4.1 高阶因果熵算法原理

如章节 4.2.2 所述，结构因果模型建立在一个贝叶斯网络上，但该贝叶斯网络的结构却并不一定预先可知，其通常可以从相关的领域知识中获得，或者也可以从观测数据中辨识出，由后者又衍生出因果发现领域^[289-293]。因此，通过因果发现，相比于基于系统辨识和生成模型的场景数据生成方法，基于结构因果模型的反事实推理所需要的领域知识更少。本文已经论述了复杂系统中关于时空因果的建模、理论基础和应用技术，再结合相应的时空因果发现，则可以构建完整的基于时空贝叶斯网络的时空图反事实框架。因此，本文于此提出高阶因果熵算法，如算法 4 所示，用于唯一辨识时空贝叶斯网络的结构，本文称之为唯一因果网络发现。

算法 4：高阶因果熵算法

Input: 变量 $\mathbf{V} = \{X_1, \dots, X_N\}$ 和最大因果时延 τ_{\max} ;

Output: 时空贝叶斯网络 \mathcal{G} ;

```

1 for  $i = 1, \dots, N$  并行地 do
2   初始化  $\mathbf{Pa}(X_{i,t}) = \emptyset$ ;
3   for  $\tau = 1, \dots, \tau_{\max}$  do
4     令条件集  $\mathbf{Z} = \{X_{1,t-\tau}, \dots, X_{N,t-\tau}\}$ ;
5     for  $j = 1, \dots, N$  do
6        $\mathcal{T}_{X_{j,t-\tau} \rightarrow X_{i,t} | \mathbf{Z} \setminus \{X_{j,t-\tau}\}} = H(X_{i,t} | \mathbf{Z} \setminus \{X_{j,t-\tau}\}) - H(X_{i,t} | \mathbf{Z})$ ;
7       if  $\mathcal{T}_{X_{j,t-\tau} \rightarrow X_{i,t} | \mathbf{Z} \setminus \{X_{j,t-\tau}\}} > \alpha$  then
8         添加  $X_{j,t-\tau}$  进  $\mathbf{Pa}(X_{i,t})$ ;
9       else
10        从  $\mathbf{Z}$  中删除  $X_{j,t-\tau}$ ;
11   for 每一个  $Y \in \mathbf{Pa}(X_{i,t})$  do
12      $\mathcal{T}_{Y \rightarrow X_{i,t} | \mathbf{Pa}(X_{i,t}) \setminus \{Y\}} = H(X_{i,t} | \mathbf{Pa}(X_{i,t}) \setminus \{Y\}) - H(X_{i,t} | \mathbf{Pa}(X_{i,t}))$ ;
13     if  $\mathcal{T}_{Y \rightarrow X_{i,t} | \mathbf{Pa}(X_{i,t}) \setminus \{Y\}} < \beta$  then
14       从  $\mathbf{Pa}(X_{i,t})$  中删除  $Y$ ;
15 用  $\mathbf{Pa}(X_{1,t}), \dots, \mathbf{Pa}(X_{N,t})$  重构  $\mathcal{G}$ ;
```

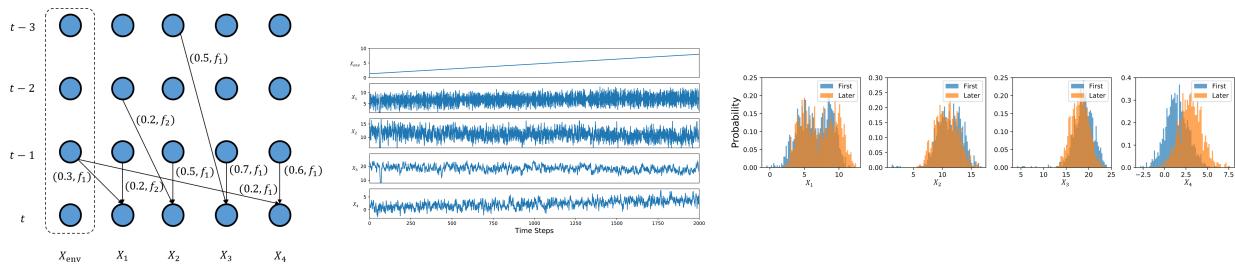
算法 4 主要分为两个部分，第 3 行到第 10 行代码的功能是为初始的空 $\mathbf{Pa}(X_{i,t})$ 添加因果父母节点，第 11 行到第 14 行代码的功能是为 $\mathbf{Pa}(X_{i,t})$ 删除弱因果父母节点。其辨识原理可参考推论 4.3.6，由此算法 4 在第 3 行到第 10 行代码中遍历时延 $\tau = 1, \dots, \tau_{\max}$ ，并在第 11 行到第 14 行代码中遍历 $\mathbf{Pa}(X_{i,t})$ 中的节点，其原理均相同。这里 τ_{\max} 表示待遍历的最大因果时延，是个参数。此外，直接检测可预测性的算法复杂度较高，而由时空因果等价性定理（见定理 4.3.3）可知，时空贝叶斯网络中的马尔可夫性（d-分离）可以由零转移熵等价辨识出，也即当检测到零转移熵时，则可以判断为没有因果连边，反之则有，因此通过检测零转移熵而等价检测零可预测性则较为容易，所以在算法 4 中通过检测零转移熵以判断是否存在因果连边。为此，其设置了参数 α 和 β ，当转移熵估计值低于 α 和 β ，则将检测的转移熵视为零。此外，该算法取名“因果熵”是出自文献^[318] 中提出的最优因果熵算法，本文高阶因果熵算法相对于最优因果熵算法的优势在于可以检测高阶因果关系，而最优因果熵算法只能检测一阶因果关系（一阶马尔可夫性）。还必须强调的是，因果熵与转移熵原理相同，区别只在于因果熵用在时空贝叶斯网络中，从而其具有因果含义，保留因果熵名称的原因有二，一是为了强调因果性，二是为了保证研究脉络的连续性。

此外，高阶因果熵算法的时间复杂度为 $\mathcal{O}(N^3 \tau_{\max})$ 。其主要的计算取决于搜索每个 t 时刻变量的因果父母，而对于每个 t 时刻变量，搜索其因果父母都需要遍历 $N \tau_{\max}$ 个“过去”变量，所以这部分的时

间复杂度为 $\mathcal{O}(N\tau_{\max})$ 。然后由时空因果可分解性定理（见定理 4.3.9）可知，对于复杂系统中的可观测变量 $\mathbf{V} = \{X_1, \dots, X_N\}$ ，可以并行检测每个变量在 t 时刻的因果父母，且检测过程和结果相互独立，所有检测结果最后可以线性组合出一个完整的贝叶斯网络。因此，每个 t 时刻变量的因果父母可以并行搜索，仅考虑最后合并的时间复杂度 $\mathcal{O}(1)$ 。此外，本文使用 KSG 估计器^[324] 计算因果熵，其最坏情况下时间复杂度为 $\mathcal{O}(N^2)$ ，通过 KD 树算法可以加速其中的样本近邻搜索过程。综上可得总体的时间复杂度为 $\mathcal{O}(N^3\tau_{\max})$ ，由时空因果唯一性定理（见定理 4.3.10）可知，其检测出的网络结构唯一。

4.4.2 实验设置

本文将高阶因果熵算法在仿真数据上进行测试，该仿真数据采样自一个满足时空贝叶斯网络结构约束的系统，见公式 (4.16)。举个例子，如图 4.12(a)所示，每条连边上设置了函数 f_1 和 f_2 ，及其相应的权重。其中 $f_1(x) = x$, $f_2(x) = x + 5x^2e^{-\frac{x}{20}}$ ，于是变量的状态可以由其历史状态计算得出，比如 $x_{1,t} = 0.2f_2(x_{1,t-1}) + 0.3f_1(x_{\text{env},t-1}) + N(0, 1)$ ，这里 $N(0, 1)$ 是标准高斯噪声。此外，为了仿真趋势性时间序列，本文设置了环境变量 X_{env} 作为变量 X_1 和变量 X_4 的共同因果父母，而 X_{env} 没有因果父母，因此其状态轨迹设置为 1.3 ~ 8.0 之间持续 2000 个采样步数的递增序列，如图 4.12(b)所示。其它四个变量的状态轨迹也同样展示在图 4.12(b)中。本文取这四个变量的前 1000 步样本和后 1000 步样本，并对其分别计算数据分布，如图 4.12(c)所示，可见其前后 1000 步的数据分布发生明显的偏移。实际实验中，本文设置了多个具有图 4.12(a)中案例形式的系统并从中采样系统轨迹，其具有不同的连边权重和网络结构，采样步数固定为 2000 步，本文经过反复测试以消除实验误差。



(a) 时空贝叶斯网络案例（出自文(b) 趋势性时间序列数据（出自文献^[65] 中 Figure 5）
献^[65] 中 Figure 6） (b) 趋势性时间序列数据（出自文献^[65] 中 Figure 6） (c) 时间序列中的分布偏移（出自文献^[65] 中 Figure 7）

图 4.12: 仿真时间序列数据展示。

此外，为了评估算法性能，本文引入真正例率 (True Positive Rate, TPR) 和假正例率 (False Positive Rate, FPR) 等指标，定义如下：

$$\begin{aligned} TPR &= \frac{\text{真正例}}{\text{真正例} + \text{假反例}}, \\ FPR &= \frac{\text{假正例}}{\text{假正例} + \text{真反例}}. \end{aligned} \quad (4.31)$$

此外，为了比较高阶因果熵算法和其它基准算法的性能，本文引入如下基准算法：

1. 格兰杰因果 (Granger Causality, GC) 检测算法^[319]: 构建回归模型，通过辨识模型参数并评估因果效应显著程度，以此判断是否存在因果关系，具体原理还可参考本文章节 4.3.3。
2. Convergent Cross Mapping (CCM) 算法^[325]: 通过将系统从高维流形表示嵌入到低维，从而得到关于系统变量的多个低维的流形表示（含因果时延），再通过低维流形之间的可重构性判断对应系统变量之间是否具有存在因果关系。
3. PCMCI 算法^[326]: 该算法主要分为两部分，首先通过 PC 算法^[327,328] 辨识不含瞬时因果的时空贝叶斯网络结构，再通过瞬时条件独立 (Momentary Conditional Independence, MCI) 检测算法辨识瞬时因果

关系，最后将二者结果合并。

4. DYNOTEARs 算法^[329]: 通过构建线性回归模型直接拟合观测数据，并根据模型参数判断是否存在因果关系。

本文设置高阶因果熵算法中的参数 $\tau_{\max} = 5$, $\alpha = 0.01$, $\beta = 0.02$, 其它算法均采用其对应文献中默认参数设置。

4.4.3 实验结果

首先，所有对比算法在图 4.12(a)中时空贝叶斯网络上测试的 ROC (receiver operating characteristic) 曲线如图 4.13(a)所示，易见高阶因果熵算法的 AUC (area under curve) 面积超过其它基准算法。此外，本文还测试了所有对比算法在不同网络规模下的性能，如图 4.14中箱图所示，随着网络规模逐渐上升，保持 FPR 指标值尽可能低于 0.1，高阶因果熵算法的 TPR 指标值虽然逐渐下降，但仍然超过其它基准算法并保持在较高水平。

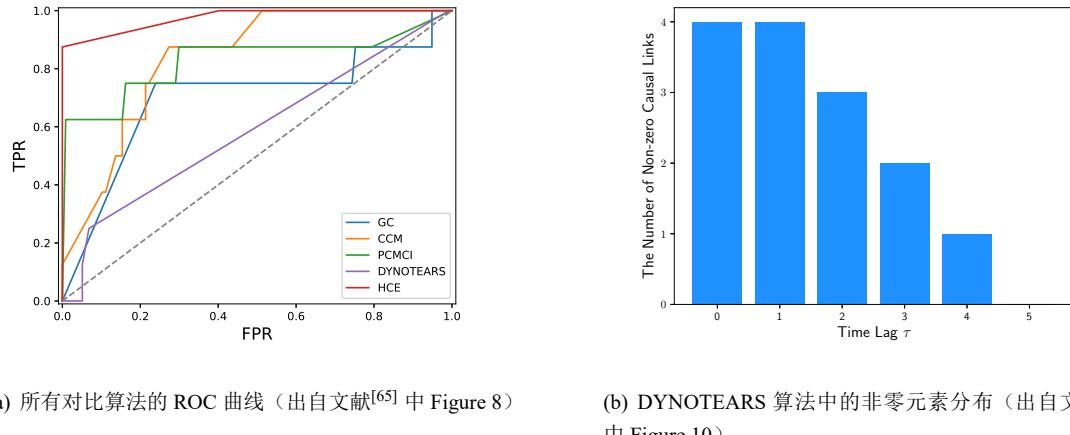


图 4.13: 高阶因果熵算法及其它基准算法性能对比。

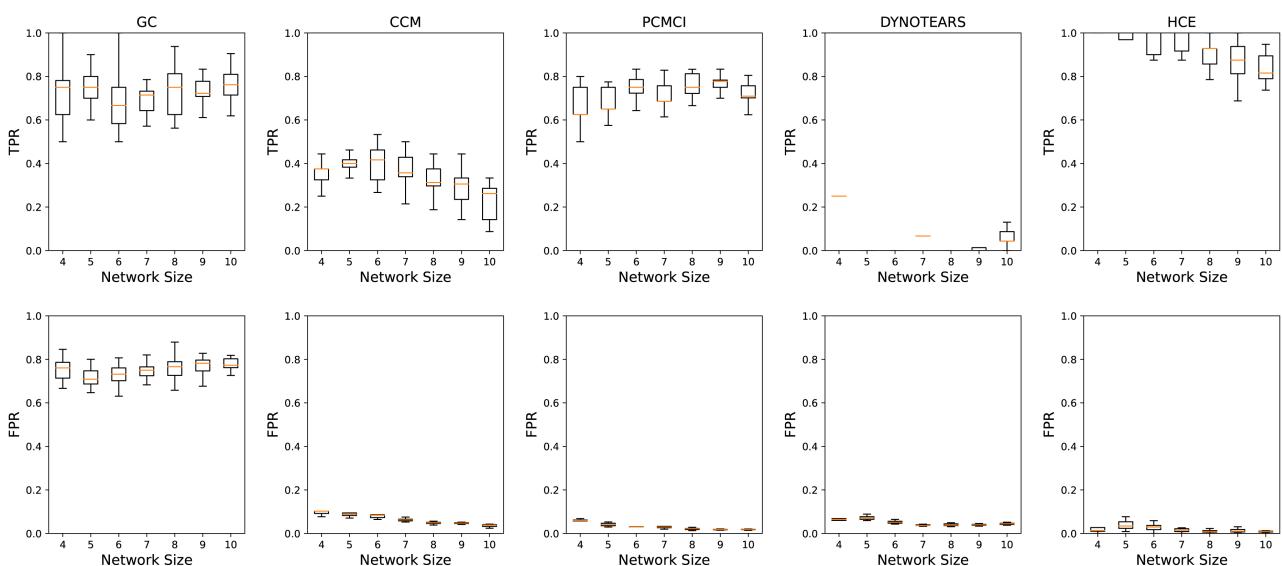


图 4.14: 高阶因果熵算法及其它基准算法性能对比（出自文献^[65] 中 Figure 9）。

相比之下，格兰杰因果检测算法之所以不准确，主要是因为这里格兰杰因果检测是对双变量逐对检测，因而没有利用时空贝叶斯网络中的信息路径阻隔的原理（见本文推论 4.3.6），所以检测出了许多虚

假连边。另一个值得特别说明的算法是 DYNOTEARs 算法，该算法在本仿真数据集上表现较差，原因有二，(1) 最主要的原因是 DYNOTEARs 算法中假设回归模型是线性的，所以不适应本仿真数据集中的非线性函数。(2) DYNOTEARs 算法在回归模型中考虑了瞬时因果，以至于不精确的数据拟合结果中出现了大量的瞬时因果连边，如图 4.13(b)所示，而这些瞬时因果连边不在本仿真数据集的数据生成机制中。基于回归模型并通过拟合数据以辨识因果关系，其准确率往往取决于数据拟合的准确性，而从 DYNOTEARs 算法的结果来看，该方法在大规模的因果网络辨识问题上效果较差。由于回归模型和系统辨识（见本文第二章）在建模和参数辨识上有很多相近之处，因而基于回归模型的因果发现算法性能瓶颈与系统辨识问题的瓶颈无二。

4.4.4 神经高维高阶因果熵估计

本章节讨论高维高阶因果熵的估计。在算法 4 中，本文采用了 KSG 估计器^[324] 计算转移熵 T ，但实际应用中存在一些特殊情况，比如图 4.15(a)中展示了由十只鸽子构成的鸽群的飞行轨迹，图 4.15(b)中展示了由六只狗构成的狗群和狗主人之间的相对运动轨迹，在这些特殊的集群行为中，其时空因果关系是高维高阶的，也即集群中每个个体可视作一个变量，而这些变量本身又是由多种属性描述的，甚至是多模态的。因此，高维高阶因果熵相比于前文高阶因果熵（见章节 4.4）需要额外处理集群中个体变量的多属性，而 KSG 估计器^[324] 则较难处理这种“高维”问题，于是文献^[330] 引入了基于人工神经网络的高维高阶因果熵估计方法用于解决此问题。本文于此论述该方法及其理论框架如何进一步拓展对唯一因果发现的研究。

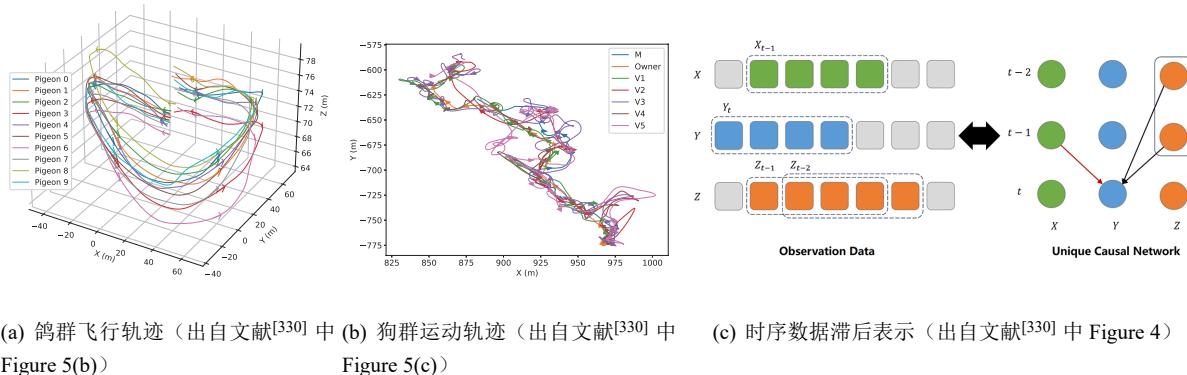


图 4.15: 存在高维高阶时空因果关系的时序数据。

高维高阶因果熵估计问题的关键点有二，(1) 在高阶因果熵算法中存在两种特殊情况，一种条件集 Z 非空，另一种则条件集 Z 为空集，当条件集 Z 为空集，则因果熵退化为双变量互信息。(2) 如果采用人工神经网络估计高维高阶因果熵，则如何用时间序列样本表示和区分各个时刻的变量。针对关键点 (1)，文献^[330] 引入 Donsker-Varadhan 表示定理，如下：

定理 4.4.1 (Donsker-Varadhan 表示^[331])。KL 散度（见公式 (3.3)）存在变分下界，也即对任意分布 P 和 Q ，存在

$$D_{KL}(P||Q) = \sup_{f \in \mathcal{F}} \{ \mathbb{E}_P[f] - \log \mathbb{E}_Q[e^f] \}, \quad (4.32)$$

这里 \mathcal{F} 表示一族函数 $f : \Omega \rightarrow \mathbb{R}$ 构成的集合， Ω 是场景数据域（样本空间）。

于是在高阶因果熵算法中，对任意变量 X 、 Y 和条件集 Z ，如果条件集 Z 为空，则因果熵可以表示

如下：

$$\begin{aligned}
 \mathcal{T}_{Y_{t-\tau} \rightarrow X_t} &= \iint P(x_t, y_{t-\tau}) \log \frac{P(x_t, y_{t-\tau})}{P(x_t)P(y_{t-\tau})} dx_t dy_{t-\tau} \\
 &= D_{\text{KL}}(P(x_t, y_{t-\tau}) || P(x_t)P(y_{t-\tau})) \\
 &= \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{P(x_t, y_{t-\tau})} [f(x_t, y_{t-\tau})] - \log \mathbb{E}_{P(x_t)P(y_{t-\tau})} [e^{f(x_t, y_{t-\tau})}] \right\}.
 \end{aligned} \tag{4.33}$$

而如果条件集 \mathbf{Z} 非空则可得

$$\begin{aligned}
 \mathcal{T}_{Y_{t-\tau} \rightarrow X_t | \mathbf{Z}} &= \int \cdots \int P(x_t, y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|}) \times \log \left(\frac{P(x_t | y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})}{P(x_t | \mathbf{z}_{1:|\mathbf{Z}|})} \right) dx_t dy_{t-\tau} d\mathbf{z}_{1:|\mathbf{Z}|} \\
 &= D_{\text{KL}}(P(x_t, y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|}) || P(x_t | \mathbf{z}_{1:|\mathbf{Z}|})P(y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})) \\
 &= \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{P(x_t, y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})} [f(x_t, y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})] - \log \mathbb{E}_{P(x_t | \mathbf{z}_{1:|\mathbf{Z}|})P(y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})} [e^{f(x_t, y_{t-\tau}, \mathbf{z}_{1:|\mathbf{Z}|})}] \right\}.
 \end{aligned} \tag{4.34}$$

这里 x_t 、 $y_{t-\tau}$ 和 $\mathbf{z}_{1:|\mathbf{Z}|}$ 分别表示对应变量的样本。于是，针对公式 (4.33) 和公式 (4.34) 中的函数 $f(\dots)$ ，可以用人工神经网络参数化即可，具体原理可以参考本文章节 3.2.3 和章节 3.2.4 中的多模态特征提取相关内容。因此，高维高阶因果熵估计将高阶因果熵算法扩展到了多模态因果发现。此外，针对关键点 (2)，还需要考虑用时间序列样本表示和区分各个时刻的变量，文献^[330] 中引入了滞后表示方法，如图 4.15(c) 所示，通过错位选择时间序列样本，从而可以通过不同的错开的位数以表示各个时刻变量之间的滞后关系。具体实验结果可参考文献^[330]，在文献中有更详细的展示。

第五章 总结与展望

5.1 本文总结

本文旨在促进人工智能基础理论与安全技术的发展。本文针对复杂系统场景数据生成问题展开研究，研究基于系统辨识、生成模型和因果推断等三类复杂系统场景数据生成方法。本文从数据分布偏移（一致）这一问题，沿着系统辨识、生成模型和因果推断这一主线展开论述，形成一整套复杂系统场景数据生成理论框架和技术方法论，并最终论证了构造生成器 g 并实现 $P(x) \neq P(y)$ 的可行性。

从复杂系统场景数据生成视角来看，纵观全文，可以发现一些趋势。首先，复杂系统建模所需要的领域知识越来越少。在基于系统辨识的场景数据生成方法中，系统模型的构建往往十分依赖领域知识，尤其是稀疏辨识方法，而人工神经网络（神经微分方程）的使用使得其对领域知识的依赖程度逐渐降低，但同时系统辨识的可解释性却也同时降低，直到物理信息神经网络在二者间取了折中。而在基于生成模型的场景数据生成方法中，其不对复杂系统直接建模，而是转向建模复杂系统中的不确定性，由于其完全采用了人工神经网络进行建模，所以对领域知识的依赖程度较低。而在基于因果推断的场景数据生成方法中，其依据（因果）贝叶斯网络中蕴含的马尔可夫性对生成模型中的不确定性进一步分解，见公式(4.10)和公式(4.17)。尽管在早期的因果推断研究中，因果关系同样由领域知识给出，但随着因果发现算法的出现，从观测场景数据中直接推断出因果关系而不依赖于领域知识成为可能。因此，基于因果推断的场景数据生成方法对领域知识的依赖程度更低。

其二，模型的参数量从少量到越来越多，但可解释性却越来越低。更多的领域知识会使得复杂系统变得“简单”，但领域知识的获取并不容易。如前所述，为使得复杂系统建模对领域知识的依赖程度降低，那么面对变得“复杂”的系统就需要更加复杂的模型才足以描述，于是模型的参数量逐渐增加，最终诞生了大人工智能模型，比如参考本文章节 3.2.5 中的生成式大模型。但与此同时，这种庞大而复杂的人工神经网络模型使得对模型的理解变得困难，进而模型的可解释性逐渐降低。在面对这一趋势时，有许多研究曾尝试避免这种可解释性的降低，但对应用的需求超过了对可解释性的需求，以至于有一段时间人们更关注于把这类人工神经网络模型用起来，而不是解释模型本身。但随着应用范围愈加广泛，并触及到一些存在高可靠要求的应用场景，如电力网络和通信网络，这种可解释性的需求正在逐渐升高。

其三，复杂系统场景数据生成逐渐能够实现数据外推，但外推数据不可证伪。无论是基于系统辨识，或是基于生成模型，其场景数据生成总是约束数据分布一致 ($P(x) = P(y)$)，这是为了保证生成的场景数据不会越出观测，从而使得场景数据生成的准确性可以在观测数据中得到验证。但在基于因果推断的场景数据生成方法中则不然，因果推断使得其可以从观测的场景数据中辨识出因果关系，进而在因果关系的基础上外推数据，从而将观测的场景数据外推至不可观测的反事实。潜在结果模型发现了这个问题，于是做出了一致性假设（见假设 3）以使其推断出的反事实仍然满足数据分布一致。但结构因果模型则认为没有必要做这一假设，也正因如此，其可以突破数据分布一致约束，实现 $P(x) \neq P(y)$ 。然而，反事实可证伪性和数据外推是互斥的关系，可证伪则不算外推，外推则不可证伪。因此，结构因果模型实际上提供了一种思考和推理反事实的方法，其强调推理的过程是反事实的，而不是推理的结果是反事实的，因为其推理的结果在潜在结果模型的定义下并不总是可证伪的。而在这一问题上，潜在结果模型则倾向于可证伪的反事实推理，其强调反事实推理的结果是可证伪的，因而与结构因果模型产生了根本性的分歧。

总之，本文整合了系统辨识、生成模型和因果推断这三个重要的研究领域，并在复杂系统场景数据生成这一框架下集中讨论其共性，并以此解决了复杂系统中的数据分布偏移情况下的场景数据生成问题，从而形成一整套复杂系统场景数据生成理论框架和技术方法论，并最终论证了构造生成器 g 并实现 $P(x) \neq P(y)$ 的可行性。

5.2 未来研究展望

正如本文章节 1.1 所言，下一代人工智能或可将知识和数据结合，采用“大数据、大模型、大算力”模式解决人工智能算法在复杂场景下的泛化性问题。或可引入协同交互机制，采用“人、机、环境协同交互”模式解决人工智能在动态开放环境中所面临的数据分布偏移问题。或可引入因果机制，从而在封闭环境中实现观测场景数据的外推。此外，本文还认为，科学技术的进步总是在具体的场景和具体的问题中产生的，所以应当更加关注具体的场景数据，才能总结出更符合数据规律的理论与技术。在实践中检验真理，本文作者会在未来研究中始终践行这一点。

参考文献

- [1] 工业和信息化部, 中央网络安全和信息化委员会办公室, 国家发展和改革委员会, et al. 国家人工智能产业综合标准化体系建设指南（2024 版）[EB/OL]. <https://www.gov.cn/zhengce/zhengceku/202407/P020240702716282797987.pdf>. 2024.
- [2] TURING, A. M. Computing machinery and intelligence[J]. Mind, 1950, 59(236):433–460.
- [3] McCarthy, John, Minsky, Marvin L., Rochester, Nathaniel, et al. A proposal for the Dartmouth Summer Research Project on artificial intelligence[J]. AI Magazine, 1955, 27(4):12.
- [4] Simon, Herbert A. Models of Man: Social and Rational[M]. Wiley, 1957.
- [5] Newell, Allen and Simon, Herbert A. Computer science as empirical inquiry: symbols and search[J]. Communications of the ACM, 1976, 19(3):113–126.
- [6] Newell, Allen. Physical symbol systems[J]. Cognitive Science, 1980, 4(2):135–183.
- [7] Riguzzi, Fabrizio, Bellodi, Elena, and Zese, Riccardo. A history of probabilistic inductive logic programming[J]. Frontiers in Robotics and AI, 2014, 1(6):1–5.
- [8] McCulloch, Warren S. and Pitts, Walter. A logical calculus of the ideas immanent in nervous activity[J]. The Bulletin of Mathematical Biophysics, 1943, 5(4):115–133.
- [9] Hebb, Donald Olding. The Organization of Behavior: A Neuropsychological Theory[M]. Psychology press, 1949.
- [10] Rosenblatt, Frank. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological Review, 1958, 65(6):386.
- [11] Linnainmaa, Seppo. Taylor expansion of the accumulated rounding error[J]. BIT Numerical Mathematics, 1976, 16(2):146–160.
- [12] Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088):533–536.
- [13] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, et al. Deep residual learning for image recognition[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.
- [14] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, et al. Identity mappings in deep residual networks[C]. In: Computer Vision – ECCV 2016. 2016. 630–645.
- [15] Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5):359–366.

- [16] Funahashi, Ken-Ichi. On the approximate realization of continuous mappings by neural networks[J]. *Neural Networks*, 1989, 2(3):183–192.
- [17] Lu, Zhou, Pu, Hongming, Wang, Feicheng, et al. The expressive power of neural networks: a view from the width[C]. In: *Advances in Neural Information Processing Systems*. 2017.
- [18] Liang, Shiyu and Srikant, R. Why deep neural networks for function approximation?[C]. In: *International Conference on Learning Representations*. 2017. Available: <https://openreview.net/forum?id=SkpSlKIel>.
- [19] Raghu, Maithra, Poole, Ben, Kleinberg, Jon, et al. On the expressive power of deep neural networks[C]. In: *International Conference on Machine Learning*. 2017. 2847–2854.
- [20] Vapnik, V.N. An overview of statistical learning theory[J]. *IEEE Transactions on Neural Networks*, 1999, 10(5):988–999.
- [21] Devaney, Robert. *An Introduction to Chaotic Dynamical Systems*[M]. Benjamin Cummings, 1986.
- [22] Vapnik, Vladimir N. *Statistical Learning Theory*[M]. Wiley, 1998.
- [23] 张钹, 苏航, 朱军. 迈向第三代人工智能 [J]. *中国科学: 信息科学*, 2020, 50(9):1281–1302.
- [24] Wu, Fei, Lu, Cewu, Zhu, Mingjie, et al. Towards a new generation of artificial intelligence in China[J]. *Nature Machine Intelligence*, 2020, 2(6):312–316.
- [25] Bommasani, Rishi, Hudson, Drew A., Adeli, Ehsan, et al. On the opportunities and risks of foundation models[EB/OL]. <https://arxiv.org/abs/2108.07258>. 2022.
- [26] Ha, David and Schmidhuber, Jürgen. Recurrent world models facilitate policy evolution[C]. In: *Advances in Neural Information Processing Systems*. 2018.
- [27] LeCun, Yann. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27[EB/OL]. <https://arxiv.org/abs/2108.07258>. 2022.
- [28] Zhou, Zhi-Hua. Open-environment machine learning[J]. *National Science Review*, 2022, 9(8):nwac123.
- [29] Zador, Anthony, Escola, Sean, Richards, Blake, et al. Catalyzing next-generation artificial intelligence through neuroai[J]. *Nature Communications*, 2023, 14(1):1597.
- [30] Ji, Jiaming, Qiu, Tianyi, Chen, Boyuan, et al. AI alignment: a comprehensive survey[EB/OL]. <https://arxiv.org/abs/2310.19852>. 2024.
- [31] Pearl, Judea and Mackenzie, Dana. *The Book of Why: The New Science of Cause and Effect*[M]. Basic Books, 2018.
- [32] Rubin, Donald B. Causal inference using potential outcomes[J]. *Journal of the American Statistical Association*, 2005, 100(469):322–331.
- [33] Imbens, Guido W. and Rubin, Donald B. *Causal Inference in Statistics, Social, and Biomedical Sciences*[M]. Cambridge University Press, 2015.
- [34] Pearl, Judea. *Causality: Models, Reasoning, and Inference (2nd Edition)*[M]. Cambridge university press, 2009.

- [35] Peters, Jonas, Janzing, Dominik, and Schölkopf, Bernhard. Elements of Causal Inference: Foundations and Learning Algorithms[M]. The MIT Press, 2017.
- [36] Gao, Ting-Ting and Yan, Gang. Autonomous inference of complex network dynamics from incomplete and noisy data[J]. *Nature Computational Science*, 2022, 2(3):160–168.
- [37] Prasse, Bastian and Mieghem, Piet Van. Predicting network dynamics without requiring the knowledge of the interaction graph[J]. *Proceedings of the National Academy of Sciences*, 2022, 119(44):e2205517119.
- [38] Nelles, Oliver. Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes (2nd Edition)[M]. Springer, 2020.
- [39] Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes[EB/OL]. <https://arxiv.org/abs/1312.6114>. 2022.
- [40] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, et al. Generative adversarial nets[C]. In: Advances in Neural Information Processing Systems. 2014.
- [41] Dinh, Laurent, Krueger, David, and Bengio, Yoshua. NICE: Non-linear independent components estimation[EB/OL]. <https://arxiv.org/abs/1410.8516>. 2015.
- [42] Sohl-Dickstein, Jascha, Weiss, Eric, Maheswaranathan, Niru, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]. In: International Conference on Machine Learning. 2015. 2256–2265.
- [43] Rubin, Donald B. Inference and missing data[J]. *Biometrika*, 1976, 63(3):581–592.
- [44] Rubin, Donald B. Bayesian inference for causal effects: the role of randomization[J]. *The Annals of Statistics*, 1978, 6(1):34–58.
- [45] Yang, Jingkang, Zhou, Kaiyang, Li, Yixuan, et al. Generalized out-of-distribution detection: a survey[J]. *International Journal of Computer Vision*, 2024. Available: <https://doi.org/10.1007/s11263-024-02117-4>.
- [46] Zhou, Kaiyang, Liu, Ziwei, Qiao, Yu, et al. Domain generalization: a survey[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, 45(4):4396–4415.
- [47] Wang, Jindong, Lan, Cuiling, Liu, Chang, et al. Generalizing to unseen domains: a survey on domain generalization[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35(8):8052–8072.
- [48] Zhuang, Fuzhen, Qi, Zhiyuan, Duan, Keyu, et al. A comprehensive survey on transfer learning[J]. *Proceedings of the IEEE*, 2021, 109(1):43–76.
- [49] Gretton, Arthur, Borgwardt, Karsten, Rasch, Malte, et al. A kernel method for the two-sample problem[C]. In: Advances in Neural Information Processing Systems. 2006.
- [50] Gretton, Arthur, Borgwardt, Karsten M., Rasch, Malte J., et al. A kernel two-sample test[J]. *Journal of Machine Learning Research*, 2012, 13(25):723–773.
- [51] 宋学锋. 复杂性、复杂系统与复杂性科学 [J]. 中国科学基金, 2003, 5(2):262–269.
- [52] 郭雷. 系统学是什么 [J]. 系统科学与数学, 2016, 36(3):291–301.
- [53] 钱学森, 戴汝为, 于景元. 一个科学新领域——开放的复杂巨系统及其方法论 [J]. 自然杂志, 1990, 12(1):3–10.

- [54] 齐磊磊. 系统科学、复杂性科学与复杂系统科学哲学 [J]. 系统科学学报, 2012, 20(3):7–11.
- [55] 狄增如 and 陈晓松. 复杂系统科学研究进展 [J]. 北京师范大学学报 (自然科学版), 2022, 58(3):371–381.
- [56] 丁璟韬, 徐丰力, 孙浩, et al. 人工智能驱动的复杂系统研究前沿 [J]. 电子科技大学学报, 2024, 53(33):455–461.
- [57] Hadamard, Jacques. Sur les problèmes aux dérivées partielles et leur signification physique[J]. Princeton University Bulletin, 1902, 13:49–52.
- [58] Birkhoff, George D. Proof of the ergodic theorem[J]. Proceedings of the National Academy of Sciences, 1931, 17(12):656–660.
- [59] Chen, Yize, Wang, Yishen, Kirschen, Daniel, et al. Model-free renewable scenario generation using generative adversarial networks[J]. IEEE Transactions on Power Systems, 2018, 33(3):3265–3275.
- [60] Li, Jinghua, Zhou, Jiasheng, and Chen, Bo. Review of wind power scenario generation methods for optimal operation of renewable energy systems[J]. Applied Energy, 2020, 280:115992.
- [61] Gu, Chaojun and Jirutijaroen, Panida. Dynamic state estimation under communication failure using kriging based bus load forecasting[J]. IEEE Transactions on Power Systems, 2015, 30(6):2831–2840.
- [62] Heidari Kapourchali, Mohammad, Sepehry, Mojtaba, and Aravinthan, Visvakumar. Fault detector and switch placement in cyber-enabled power distribution network[J]. IEEE Transactions on Smart Grid, 2018, 9(2):980–992.
- [63] Ge, Xiaohua, Han, Qing-Long, Zhang, Xian-Ming, et al. Distributed event-triggered estimation over sensor networks: a survey[J]. IEEE Transactions on Cybernetics, 2020, 50(3):1306–1320.
- [64] Zhong, Xingsi, Jayawardene, Iroshani, Venayagamoorthy, Ganesh Kumar, et al. Denial of service attack on tie-line bias control in a power system with PV plant[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(5):375–390.
- [65] Kang, Mingyu, Chen, Duxin, Meng, Ning, et al. Identifying unique spatial-temporal Bayesian network without Markov equivalence[J]. IEEE Transactions on Artificial Intelligence, 2024. DOI: 10.1109/TAI.2024.3483188.
- [66] Kang, Mingyu, Chen, Duxin, Pu, Ziyuan, et al. Spatio-temporal graphical counterfactuals: an overview[EB/OL]. <https://arxiv.org/abs/2407.01875>. 2024.
- [67] Shpitser, Ilya and Pearl, Judea. Identification of joint interventional distributions in recursive semi-Markovian causal models[C]. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2006. 1219–1226.
- [68] Donoho, D.L. Compressed sensing[J]. IEEE Transactions on Information Theory, 2006, 52(4):1289–1306.
- [69] Donoho, D.L., Elad, M., and Temlyakov, V.N. Stable recovery of sparse overcomplete representations in the presence of noise[J]. IEEE Transactions on Information Theory, 2006, 52(1):6–18.
- [70] Candes, E.J., Romberg, J., and Tao, T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information[J]. IEEE Transactions on Information Theory, 2006, 52(2):489–509.
- [71] Mallat, S.G. and Zhang, Zhifeng. Matching pursuits with time-frequency dictionaries[J]. IEEE Transactions on Signal Processing, 1993, 41(12):3397–3415.

- [72] Pati, Y.C., Rezaiifar, R., and Krishnaprasad, P.S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition[C]. In: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers. 1993. 40–44.
- [73] Blumensath, Thomas and Davies, Mike E. On the difference between orthogonal matching pursuit and orthogonal least squares[EB/OL]. <https://eprints.soton.ac.uk/142469/1/BDOMPvsOLS07.pdf>. 2007.
- [74] Elad, Michael. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing[M]. Springer, 2010.
- [75] Tibshirani, Robert. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1996, 58(1):267–288.
- [76] Candes, E.J. and Tao, T. Decoding by linear programming[J]. IEEE Transactions on Information Theory, 2005, 51(12):4203–4215.
- [77] Chen, Scott Shaobing, Donoho, David L., and Saunders, Michael A. Atomic decomposition by basis pursuit[J]. SIAM Review, 2001, 43(1):129–159.
- [78] Boyd, Stephen, Parikh, Neal, Chu, Eric, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends® in Machine learning, 2011, 3(1):1–122.
- [79] Boyd, Stephen and Vandenberghe, Lieven. Convex Optimization[M]. Cambridge University Press, 2004.
- [80] Xu, Zongben, Zhang, Hai, Wang, Yao, et al. $L_{1/2}$ regularization[J]. Science China Information Sciences, 2010, 53:1159–1169.
- [81] Xu, Zongben, Chang, Xiangyu, Xu, Fengmin, et al. $L_{1/2}$ regularization: a thresholding representation theory and a fast solver[J]. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(7):1013–1027.
- [82] Brunton, Steven L., Proctor, Joshua L., and Kutz, J. Nathan. Discovering governing equations from data by sparse identification of nonlinear dynamical systems[J]. Proceedings of the National Academy of Sciences, 2016, 113(15):3932–3937.
- [83] Rudy, Samuel H., Brunton, Steven L., Proctor, Joshua L., et al. Data-driven discovery of partial differential equations[J]. Science Advances, 2017, 3(4):e1602614.
- [84] Pan, Wei, Yuan, Ye, Gonçalves, Jorge, et al. A sparse bayesian approach to the identification of nonlinear state-space systems[J]. IEEE Transactions on Automatic Control, 2016, 61(1):182–187.
- [85] Yuan, Ye, Tang, Xiuchuan, Zhou, Wei, et al. Data driven discovery of cyber physical systems[J]. Nature Communications, 2019, 10(1):4894.
- [86] Haar, Alfred. Zur theorie der orthogonalen funktionensysteme[J]. Mathematische Annalen, 1911, 71(1):38–53.
- [87] Bishop, Christopher M. Pattern Recognition and Machine Learning[M]. Springer, 2006.
- [88] Buhmann, Martin Dietrich. Radial basis functions[J]. Acta numerica, 2000, 9:1–38.
- [89] Nyquist, H. Certain topics in telegraph transmission theory[J]. Transactions of the American Institute of Electrical Engineers, 1928, 47(2):617–644.

- [90] Morlet, Jean, Arens, Georges, Fourgeau, Eliane, et al. Wave propagation and sampling theory-Part I: complex signal and scattering in multilayered media[J]. *Geophysics*, 1982, 47(2):203–221.
- [91] Morlet, Jetal, Arens, Georges, Fourgeau, Eliane, et al. Wave propagation and sampling theory-Part II: sampling theory and complex waves[J]. *Geophysics*, 1982, 47(2):222–236.
- [92] Chen, Ricky T. Q., Rubanova, Yulia, Bettencourt, Jesse, et al. Neural ordinary differential equations[C]. In: *Advances in Neural Information Processing Systems*. 2018.
- [93] Jordan, Michael I. Serial order: a parallel distributed processing approach[EB/OL]. <https://cseweb.ucsd.edu/~gary/PAPER-SUGGESTIONS/Jordan-TR-8604-OCRed.pdf>. 1986.
- [94] Elman, Jeffrey L. Finding structure in time[J]. *Cognitive Science*, 1990, 14(2):179–211.
- [95] Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory[J]. *Neural Computation*, 1997, 9(8):1735–1780.
- [96] Gers, Felix A., Schmidhuber, Jürgen, and Cummins, Fred. Learning to forget: continual prediction with LSTM[J]. *Neural Computation*, 2000, 12(10):2451–2471.
- [97] Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[EB/OL]. <https://arxiv.org/abs/1412.3555>. 2014.
- [98] Raissi, M., Perdikaris, P., and Karniadakis, G.E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. *Journal of Computational Physics*, 2019, 378:686–707.
- [99] Karniadakis, George Em, Kevrekidis, Ioannis G, Lu, Lu, et al. Physics-informed machine learning[J]. *Nature Reviews Physics*, 2021, 3(6):422–440.
- [100] Baydin, Atilim Gunes, Pearlmutter, Barak A., Radul, Alexey Andreyevich, et al. Automatic differentiation in machine learning: a survey[J]. *Journal of Machine Learning Research*, 2018, 18(153):1–43.
- [101] ichi Funahashi, Ken and Nakamura, Yuichi. Approximation of dynamical systems by continuous time recurrent neural networks[J]. *Neural Networks*, 1993, 6(6):801–806.
- [102] Zhang, Huaguang, Wang, Zhanshan, and Liu, Derong. A comprehensive review of stability analysis of continuous-time recurrent neural networks[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, 25(7):1229–1262.
- [103] Lim, Soon Hoe. Understanding recurrent neural networks using nonequilibrium response theory[J]. *Journal of Machine Learning Research*, 2021, 22(47):1–48.
- [104] Brunel, Nicolas and Hakim, Vincent. Fast global oscillations in networks of integrate-and-fire neurons with low firing rates[J]. *Neural Computation*, 1999, 11(7):1621–1671.
- [105] Campbell, Shannon R., Wang, DeLiang L., and Jayaprakash, Ciriyam. Synchrony and desynchrony in integrate-and-fire oscillators[J]. *Neural Computation*, 1999, 11(7):1595–1619.
- [106] Hodgkin, Alan L and Huxley, Andrew F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. *The Journal of Physiology*, 1952, 117(4):500.
- [107] Ozer, Mahmut, Perc, Matjaž, and Uzuntarla, Muhammet. Stochastic resonance on Newman-Watts networks of Hodgkin-Huxley neurons with local periodic driving[J]. *Physics Letters A*, 2009, 373(10):964–968.

- [108] Rulkov, Nikolai F. Regularization of synchronized chaotic bursts[J]. Physical Review Letters, 2001, 86:183–186.
- [109] Rulkov, Nikolai F. Modeling of spiking-bursting neural behavior using two-dimensional map[J]. Physical Review E, 2002, 65:041922.
- [110] Topal, Irem and Eroglu, Deniz. Reconstructing network dynamics of coupled discrete chaotic units from data[J]. Physical Review Letters, 2023, 130:117401.
- [111] Bullmore, Ed and Sporns, Olaf. Complex brain networks: graph theoretical analysis of structural and functional systems[J]. Nature Reviews Neuroscience, 2009, 10(3):186–198.
- [112] Lynn, Christopher W and Bassett, Danielle S. The physics of brain network structure, function and control[J]. Nature Reviews Physics, 2019, 1(5):318–332.
- [113] Meena, Chandrakala, Hens, Chittaranjan, Acharyya, Suman, et al. Emergent stability in complex network dynamics[J]. Nature Physics, 2023, 19(7):1033–1042.
- [114] Schaeffer, Rylan, Miranda, Brando, and Koyejo, Sanmi. Are emergent abilities of large language models a mirage?[C]. In: Advances in Neural Information Processing Systems. 2023. 55565–55581.
- [115] Pál Erdős, Alfréd Rényi. On random graphs I[EB/OL]. https://www.renyi.hu/~p_erdos/1959-11.pdf. 1959.
- [116] Gilbert, E. N. Random graphs[J]. The Annals of Mathematical Statistics, 1959, 30(4):1141–1144.
- [117] Watts, Duncan J and Strogatz, Steven H. Collective dynamics of ‘small-world’ networks[J]. Nature, 1998, 393(6684):440–442.
- [118] Newman, M.E.J. and Watts, D.J. Renormalization group analysis of the small-world network model[J]. Physics Letters A, 1999, 263(4):341–346.
- [119] Barabási, Albert-László and Albert, Réka. Emergence of scaling in random networks[J]. Science, 1999, 286(5439):509–512.
- [120] Arenas, Alex, Díaz-Guilera, Albert, Kurths, Jurgen, et al. Synchronization in complex networks[J]. Physics Reports, 2008, 469(3):93–153.
- [121] Kuramoto, Yoshiki. Self-entrainment of a population of coupled non-linear oscillators[C]. In: International Symposium on Mathematical Problems in Theoretical Physics. 1975. 420–422.
- [122] Rodrigues, Francisco A., Peron, Thomas K. DM., Ji, Peng, et al. The Kuramoto model in complex networks[J]. Physics Reports, 2016, 610:1–98.
- [123] Kalman, R. E. Mathematical description of linear dynamical systems[J]. Journal of the Society for Industrial and Applied Mathematics Series A Control, 1963, 1(2):152–192.
- [124] Lin, Ching-Tai. Structural controllability[J]. IEEE Transactions on Automatic Control, 1974, 19(3):201–208.
- [125] Liu, Yang-Yu, Slotine, Jean-Jacques, and Barabási, Albert-László. Controllability of complex networks[J]. Nature, 2011, 473(7346):167–173.
- [126] Yuan, Zhengzhong, Zhao, Chen, Di, Zengru, et al. Exact controllability of complex networks[J]. Nature Communications, 2013, 4(1):2447.

- [127] Wang, Xiaofan and Chen, Guanrong. Pinning control of scale-free dynamical networks[J]. *Physica A: Statistical Mechanics and its Applications*, 2002, 310(3):521–531.
- [128] Wang, Xiao Fan and Chen, Guanrong. Synchronization in small-world dynamical networks[J]. *International Journal of Bifurcation and chaos*, 2002, 12(1):187–192.
- [129] Yu, Wenwu, Chen, Guanrong, Lü, Jinhu, et al. Synchronization via pinning control on general complex networks[J]. *SIAM Journal on Control and Optimization*, 2013, 51(2):1395–1416.
- [130] Yu, Wenwu, Ren, Wei, Zheng, Weixing, et al. Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics[J]. *Automatica*, 2013, 49(7):2107–2115.
- [131] Chen, Guanrong. Pinning control and controllability of complex dynamical networks[J]. *International Journal of Automation and Computing*, 2017, 14:1–9.
- [132] Li, Aming, Cornelius, Sean P, Liu, Y-Y, et al. The fundamental advantages of temporal networks[J]. *Science*, 2017, 358(6366):1042–1046.
- [133] Angulo, Marco Tulio, Moog, Claude H, and Liu, Yang-Yu. A theoretical framework for controlling complex microbial communities[J]. *Nature Communications*, 2019, 10(1):1045.
- [134] Angulo, Marco Tulio, Aparicio, Andrea, and Moog, Claude H. Structural accessibility and structural observability of nonlinear networked systems[J]. *IEEE Transactions on Network Science and Engineering*, 2020, 7(3):1656–1666.
- [135] D’Souza, Raissa M, di Bernardo, Mario, and Liu, Yang-Yu. Controlling complex networks with complex nodes[J]. *Nature Reviews Physics*, 2023, 5(4):250–262.
- [136] Albert, Réka, Jeong, Hawoong, and Barabási, Albert-László. Error and attack tolerance of complex networks[J]. *nature*, 2000, 406(6794):378–382.
- [137] Artimo, Oriol, Grassia, Marco, De Domenico, Manlio, et al. Robustness and resilience of complex networks[J]. *Nature Reviews Physics*, 2024, 6(2):114–131.
- [138] Kuznetsov, Yuri A, Kuznetsov, Iu A, and Kuznetsov, Y. *Elements of Applied Bifurcation Theory*[M]. Springer, 1998.
- [139] Gao, Jianxi, Barzel, Baruch, and Barabási, Albert-László. Universal resilience patterns in complex networks[J]. *Nature*, 2016, 530(7590):307–312.
- [140] Tu, Chengyi, Grilli, Jacopo, Schuessler, Friedrich, et al. Collapse of resilience patterns in generalized Lotka-Volterra dynamics and beyond[J]. *Physical Review E*, 2017, 95(6):062307.
- [141] Jiang, Junjie, Huang, Zi-Gang, Seager, Thomas P., et al. Predicting tipping points in mutualistic networks through dimension reduction[J]. *Proceedings of the National Academy of Sciences*, 2018, 115(4):E639–E647.
- [142] Laurence, Edward, Doyon, Nicolas, Dubé, Louis J., et al. Spectral dimension reduction of complex dynamical networks[J]. *Physical Review X*, 2019, 9(1):011042.
- [143] Gao, Jianxi. Intrinsic simplicity of complex systems[J]. *Nature Physics*, 2024, 20(2):184–185.
- [144] Thibeault, Vincent, Allard, Antoine, and Desrosiers, Patrick. The low-rank hypothesis of complex systems[J]. *Nature Physics*, 2024, 20(2):294–302.

- [145] Vegué, Marina, Thibeault, Vincent, Desrosiers, Patrick, et al. Dimension reduction of dynamics on modular and heterogeneous directed networks[J]. PNAS Nexus, 2023, 2(5):pgad150.
- [146] Koch, James, Chen, Zhao, Tuor, Aaron, et al. Structural inference of networked dynamical systems with universal differential equations[J]. Chaos: An Interdisciplinary Journal of Nonlinear Science, 2023, 33(2): 023103.
- [147] Gurobi. <https://www.gurobi.com/>. 2024.
- [148] Pastor-Satorras, Romualdo and Vespignani, Alessandro. Epidemic spreading in scale-free networks[J]. Physical Review Letters, 2001, 86:3200–3203.
- [149] Pastor-Satorras, Romualdo, Castellano, Claudio, Van Mieghem, Piet, et al. Epidemic processes in complex networks[J]. Reviews of Modern Physics, 2015, 87:925–979.
- [150] MacArthur, Robert. Species packing and competitive equilibrium for many species[J]. Theoretical Population Biology, 1970, 1(1):1–11.
- [151] Harush, Uzi and Barzel, Baruch. Dynamic patterns of information flow in complex networks[J]. Nature Communications, 2017, 8(1):2181.
- [152] Paulsen, Vern I. and Raghupathi, Mrinal. An Introduction to the Theory of Reproducing Kernel Hilbert Spaces[M]. Cambridge University Press, 2016.
- [153] Chen, Zhitang, Zhang, Kun, Chan, Laiwan, et al. Causal discovery via reproducing kernel Hilbert space embeddings[J]. Neural Computation, 2014, 26(7):1484–1517.
- [154] Kullback, S. and Leibler, R. A. On information and sufficiency[J]. The Annals of Mathematical Statistics, 1951, 22(1):79–86.
- [155] Thomas, MTCAJ and Joy, A Thomas. Elements of Information Theory[M]. Wiley, 2006.
- [156] Vallender, S. S. Calculation of the Wasserstein distance between probability distributions on the line[J]. Theory of Probability & Its Applications, 1974, 18(4):784–786.
- [157] Panaretos, Victor M. and Zemel, Yoav. Statistical aspects of Wasserstein distances[J]. Annual Review of Statistics and Its Application, 2019, 6(1):405–431.
- [158] Kingma, Diederik P., Rezende, Danilo J., Mohamed, Shakir, et al. Semi-supervised learning with deep generative models[C]. In: Advances in Neural Information Processing Systems. 2014.
- [159] Kipf, Thomas N. and Welling, Max. Variational graph auto-encoders[EB/OL]. <https://arxiv.org/abs/1611.07308>. 2016.
- [160] Kipf, Thomas N. and Welling, Max. Semi-supervised classification with graph convolutional networks[C]. In: International Conference on Learning Representations. 2017. Available: <https://openreview.net/forum?id=SJU4ayYg1>.
- [161] Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein generative adversarial networks[C]. In: International Conference on Machine Learning. 2017. 214–223.
- [162] Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, et al. Improved training of Wasserstein GANs[C]. In: Advances in Neural Information Processing Systems. 2017.

- [163] Bińkowski, Mikołaj, Sutherland, Danica J., Arbel, Michael, et al. Demystifying MMD GANs[C]. In: International Conference on Learning Representations. 2018. Available: <https://openreview.net/forum?id=r1lU0zWCW>.
- [164] Mirza, Mehdi and Osindero, Simon. Conditional generative adversarial nets[EB/OL]. <https://arxiv.org/abs/1411.1784>. 2014.
- [165] Yoon, Jinsung, Jarrett, Daniel, and van der Schaar, Mihaela. Time-series generative adversarial networks[C]. In: Advances in Neural Information Processing Systems. 2019.
- [166] Karras, Tero, Aittala, Miika, Laine, Samuli, et al. Alias-free generative adversarial networks[C]. In: Advances in Neural Information Processing Systems. 2021.
- [167] Casella, George and Berger, Roger. Statistical Inference (2nd Edition)[M]. Duxbury Press, 2002.
- [168] Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using Real NVP[C]. In: International Conference on Learning Representations. 2017. Available: <https://openreview.net/forum?id=HkpbnH9lx>.
- [169] Kingma, Durk P and Dhariwal, Prafulla. Glow: Generative flow with invertible 1x1 convolutions[C]. In: Advances in Neural Information Processing Systems. 2018.
- [170] Pumarola, Albert, Popov, Stefan, Moreno-Noguer, Francesc, et al. C-Flow: Conditional generative flow models for images and 3D point clouds[C]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.
- [171] Rezende, Danilo and Mohamed, Shakir. Variational inference with normalizing flows[C]. In: International Conference on Machine Learning. 2015. 1530–1538.
- [172] Papamakarios, George, Nalisnick, Eric, Rezende, Danilo Jimenez, et al. Normalizing flows for probabilistic modeling and inference[J]. Journal of Machine Learning Research, 2021, 22(57):1–64.
- [173] Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. Denoising diffusion probabilistic models[C]. In: Advances in Neural Information Processing Systems. 2020.
- [174] Anderson, Brian D.O. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3):313–326.
- [175] Song, Jiaming, Meng, Chenlin, and Ermon, Stefano. Denoising diffusion implicit models[C]. In: International Conference on Learning Representations. 2021. Available: <https://openreview.net/forum?id=St1giarCHLP>.
- [176] Nichol, Alexander Quinn and Dhariwal, Prafulla. Improved denoising diffusion probabilistic models[C]. In: International Conference on Machine Learning. 2021. 8162–8171.
- [177] Austin, Jacob, Johnson, Daniel D., Ho, Jonathan, et al. Structured denoising diffusion models in discrete state-spaces[C]. In: Advances in Neural Information Processing Systems. 2021. 17981–17993.
- [178] Kawar, Bahjat, Elad, Michael, Ermon, Stefano, et al. Denoising diffusion restoration models[C]. In: Advances in Neural Information Processing Systems. 2022. 23593–23606.
- [179] Liu, Jiawei, Wang, Qiang, Fan, Huijie, et al. Residual denoising diffusion models[C]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2024. 2773–2783.

- [180] Hyvärinen, Aapo. Estimation of non-normalized statistical models by score matching[J]. *Journal of Machine Learning Research*, 2005, 6(24):695–709.
- [181] Vincent, Pascal. A connection between score matching and denoising autoencoders[J]. *Neural Computation*, 2011, 23(7):1661–1674.
- [182] Song, Yang and Ermon, Stefano. Generative modeling by estimating gradients of the data distribution[C]. In: *Advances in Neural Information Processing Systems*. 2019.
- [183] Song, Yang, Sohl-Dickstein, Jascha, Kingma, Diederik P, et al. Score-based generative modeling through stochastic differential equations[C]. In: *International Conference on Learning Representations*. 2021. Available: <https://openreview.net/forum?id=PxTIG12RRHS>.
- [184] 苏剑林. 生成扩散模型漫谈（五）：一般框架之 SDE 篇 [EB/OL]. <https://spaces.ac.cn/archives/9209>. 2022.
- [185] Zhang, Zijian, Zhao, Zhou, Yu, Jun, et al. ShiftDDPMs: Exploring conditional diffusion models by shifting diffusion trajectories[C]. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023. 3552–3560.
- [186] Baltrušaitis, Tadas, Ahuja, Chaitanya, and Morency, Louis-Philippe. Multimodal machine learning: a survey and taxonomy[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, 41(2):423–443.
- [187] Zhao, Jing, Xie, Xijiong, Xu, Xin, et al. Multi-view learning overview: recent progress and new challenges[J]. *Information Fusion*, 2017, 38:43–54.
- [188] Li, Yingming, Yang, Ming, and Zhang, Zhongfei. A survey of multi-view representation learning[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2019, 31(10):1863–1883.
- [189] Perozzi, Bryan, Al-Rfou, Rami, and Skiena, Steven. DeepWalk: Online learning of social representations[C]. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2014. 701–710.
- [190] Grover, Aditya and Leskovec, Jure. node2vec: Scalable feature learning for networks[C]. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. 855–864.
- [191] Mikolov, Tomas, Chen, Kai, Corrado, Greg, et al. Efficient estimation of word representations in vector space[EB/OL]. <https://arxiv.org/abs/1301.3781>. 2013.
- [192] Le, Quoc and Mikolov, Tomas. Distributed representations of sentences and documents[C]. In: *Proceedings of the 31st International Conference on Machine Learning*. 2014. 1188–1196.
- [193] Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, et al. Enriching word vectors with subword information[J]. *Transactions of the Association for Computational Linguistics*, 2017, 5:135–146.
- [194] Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks[C]. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2014. 3104–3112.
- [195] Rahate, Anil, Walambe, Rahee, Ramanna, Sheela, et al. Multimodal Co-learning: challenges, applications with datasets, recent advances and future directions[J]. *Information Fusion*, 2022, 81:203–239.

- [196] McMahan, Brendan, Moore, Eider, Ramage, Daniel, et al. Communication-efficient learning of deep networks from decentralized data[C]. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. 2017. 1273–1282.
- [197] Warnat-Herresthal, Stefanie, Schultze, Hartmut, Shastry, Krishnaprasad Lingadahalli, et al. Swarm learning for decentralized and confidential clinical machine learning[J]. Nature, 2021, 594(7862):265–270.
- [198] Mittal, Sarthak, Bengio, Yoshua, and Lajoie, Guillaume. Is a modular architecture enough?[C]. In: Advances in Neural Information Processing Systems. 2022. 28747–28760.
- [199] Schuster, M. and Paliwal, K.K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11):2673–2681.
- [200] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, et al. Attention is all you need[C]. In: Advances in Neural Information Processing Systems. 2017.
- [201] Lecun, Y., Bottou, L., Bengio, Y., et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278–2324.
- [202] Zeiler, Matthew D., Krishnan, Dilip, Taylor, Graham W., et al. Deconvolutional networks[C]. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010. 2528–2535.
- [203] Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition[EB/OL]. <https://arxiv.org/abs/1409.1556>. 2015.
- [204] Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, et al. Rethinking the inception architecture for computer vision[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.
- [205] Huang, Gao, Liu, Zhuang, van der Maaten, Laurens, et al. Densely connected convolutional networks[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
- [206] Szegedy, Christian, Liu, Wei, Jia, Yangqing, et al. Going deeper with convolutions[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.
- [207] Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]. 2017.
- [208] Huang, Gao, Chen, Danlu, Li, Tianhong, et al. Multi-scale dense networks for resource efficient image classification[C]. In: International Conference on Learning Representations. 2018. Available: <https://openreview.net/forum?id=Hk2aImxA>.
- [209] Scarselli, Franco, Gori, Marco, Tsoi, Ah Chung, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2009, 20(1):61–80.
- [210] Bruna, Joan, Zaremba, Wojciech, Szlam, Arthur, et al. Spectral networks and locally connected networks on graphs[EB/OL]. <https://arxiv.org/abs/1312.6203>. 2014.
- [211] Henaff, Mikael, Bruna, Joan, and LeCun, Yann. Deep convolutional networks on graph-structured data[EB/OL]. <https://arxiv.org/abs/1506.05163>. 2015.
- [212] Niepert, Mathias, Ahmed, Mohamed, and Kutzkov, Konstantin. Learning convolutional neural networks for graphs[C]. In: International Conference on Machine Learning. 2016. 2014–2023.

- [213] Liu, Ziqi, Chen, Chaochao, Li, Longfei, et al. Geniepath: Graph neural networks with adaptive receptive paths[C]. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019. 4424–4431.
- [214] Veličković, Petar, Cucurull, Guillem, Casanova, Arantxa, et al. Graph attention networks[C]. In: International Conference on Learning Representations. 2018. Available: <https://openreview.net/forum?id=rJXMpikCZ>.
- [215] Defferrard, Michaël, Bresson, Xavier, and Vandergheynst, Pierre. Convolutional neural networks on graphs with fast localized spectral filtering[C]. In: Advances in Neural Information Processing Systems. 2016.
- [216] Chen, Ming, Wei, Zhewei, Huang, Zengfeng, et al. Simple and deep graph convolutional networks[C]. In: International Conference on Machine Learning. 2020. 1725–1735.
- [217] Chien, Eli, Peng, Jianhao, Li, Pan, et al. Adaptive universal generalized pagerank graph neural network[C]. In: International Conference on Learning Representations. 2021. Available: <https://openreview.net/forum?id=n6j17fLxrP>.
- [218] He, Mingguo, Wei, Zhewei, Huang, zengfeng, et al. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation[C]. In: Advances in Neural Information Processing Systems. 2021. 14239–14251.
- [219] Correia, Gonçalo M., Niculae, Vlad, and Martins, André F. T. Adaptively sparse transformers[C]. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019. 2174–2184.
- [220] Clark, Kevin, Khandelwal, Urvashi, Levy, Omer, et al. What does BERT look at? An analysis of BERT’s attention[C]. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: analyzing and Interpreting Neural Networks for NLP. 2019. 276–286.
- [221] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019. 4171–4186.
- [222] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, et al. An image is worth 16x16 words: Transformers for image recognition at scale[C]. In: International Conference on Learning Representations. 2021. Available: <https://openreview.net/forum?id=YicbFdNTTy>.
- [223] Liu, Ze, Lin, Yutong, Cao, Yue, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. 10012–10022.
- [224] Trockman, Asher and Kolter, J Zico. Patches are all you need?[J]. Transactions on Machine Learning Research, 2023. Available: <https://openreview.net/forum?id=rAnB7JSMXL>.
- [225] Zhou, Haoyi, Zhang, Shanghang, Peng, Jieqi, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]. 2021. 11106–11115.
- [226] Wu, Haixu, Xu, Jiehui, Wang, Jianmin, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting[C]. In: Advances in Neural Information Processing Systems. 2021. 22419–22430.

- [227] Zhou, Tian, Ma, Ziqing, Wen, Qingsong, et al. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting[C]. In: International Conference on Machine Learning. 2022. 27268–27286.
- [228] Zhang, Yunhao and Yan, Junchi. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting[C]. In: International Conference on Learning Representations. 2023. Available: <https://openreview.net/forum?id=vSVLM2j9eie>.
- [229] Liu, Yong, Hu, Tengge, Zhang, Haoran, et al. iTransformer: Inverted transformers are effective for time series forecasting[C]. In: International Conference on Learning Representations. 2024. Available: <https://openreview.net/forum?id=JePfAI8fah>.
- [230] Zeng, Ailing, Chen, Muxi, Zhang, Lei, et al. Are transformers effective for time series forecasting?[C]. 2023. 11121–11128.
- [231] Wang, Shiyu, Wu, Haixu, Shi, Xiaoming, et al. Timemixer: Decomposable multiscale mixing for time series forecasting[C]. In: International Conference on Learning Representations. 2024. Available: <https://openreview.net/forum?id=7oLshfEIC2>.
- [232] Haviv, Adi, Ram, Ori, Press, Ofir, et al. Transformer language models without positional encodings still learn positional information[C]. In: Findings of the Association for Computational Linguistics: EMNLP 2022. 2022. 1382–1390.
- [233] Kazemnejad, Amirhossein, Padhi, Inkit, Natesan Ramamurthy, Karthikeyan, et al. The impact of positional encoding on length generalization in transformers[C]. In: Advances in Neural Information Processing Systems. 2023. 24892–24928.
- [234] Chi, Ta-Chung, Fan, Ting-Han, Chen, Li-Wei, et al. Latent positional information is in the self-attention variance of transformer language models without positional embeddings[C]. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2023. 1183–1193.
- [235] Kaplan, Jared, McCandlish, Sam, Henighan, Tom, et al. Scaling laws for neural language models[EB/OL]. <https://arxiv.org/abs/2001.08361>. 2020.
- [236] Hoffmann, Jordan, Borgeaud, Sebastian, Mensch, Arthur, et al. Training compute-optimal large language models[EB/OL]. <https://arxiv.org/abs/2203.15556>. 2022.
- [237] Zhao, Wayne Xin, Zhou, Kun, Li, Junyi, et al. A survey of large language models[EB/OL]. <https://arxiv.org/abs/2303.18223>. 2024.
- [238] Radford, Alec, Narasimhan, Karthik, Salimans, Tim, et al. Improving language understanding by generative pre-training[EB/OL]. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. 2018.
- [239] Brown, Tom B., Mann, Benjamin, Ryder, Nick, et al. Language models are few-shot learners[EB/OL]. <https://arxiv.org/abs/2005.14165>. 2020.
- [240] OpenAI, Achiam, Josh, Adler, Steven, et al. GPT-4 technical report[EB/OL]. <https://arxiv.org/abs/2303.08774>. 2024.
- [241] Radford, Alec, Wu, Jeffrey, Child, Rewon, et al. Language models are unsupervised multitask learners[EB/OL]. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. 2019.

- [242] Jacobs, Robert A., Jordan, Michael I., Nowlan, Steven J., et al. Adaptive mixtures of local experts[J]. *Neural Computation*, 1991, 3(1):79–87.
- [243] Shazeer, Noam, Mirhoseini, Azalia, Maziarz, Krzysztof, et al. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer[C]. In: International Conference on Learning Representations. 2017. Available: <https://openreview.net/forum?id=B1ckMDqlg>.
- [244] Fedus, William, Zoph, Barret, and Shazeer, Noam. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity[J]. *Journal of Machine Learning Research*, 2022, 23(120):1–39.
- [245] Riquelme, Carlos, Puigcerver, Joan, Mustafa, Basil, et al. Scaling vision with sparse mixture of experts[C]. In: Advances in Neural Information Processing Systems. 2021. 8583–8595.
- [246] Lin, Junyang, Men, Rui, Yang, An, et al. M6: a Chinese multimodal pretrainer[EB/OL]. <https://arxiv.org/abs/2103.00823>. 2021.
- [247] Kang, Mingyu, Zhu, Ran, Chen, Duxin, et al. A cross-modal generative adversarial network for scenarios generation of renewable energy[J]. *IEEE Transactions on Power Systems*, 2024, 39(2):2630–2640.
- [248] (NREL), National Renewable Energy Laboratory. 光伏电站仿真数据 [EB/OL]. <https://www.nrel.gov/grid/solar-power-data.html#panel2>. 2022.
- [249] Xiao, Dongliang and Qiao, Wei. Hybrid scenario generation method for stochastic virtual bidding in electricity market[J]. *CSEE Journal of Power and Energy Systems*, 2021, 7(6):1312–1321.
- [250] Li, Hui, Ren, Zhouyang, Xu, Yan, et al. A multi-data driven hybrid learning method for weekly photovoltaic power scenario forecast[J]. *IEEE Transactions on Sustainable Energy*, 2021, 13(1):91–100.
- [251] Qi, Yuchen, Hu, Wei, Dong, Yu, et al. Optimal configuration of concentrating solar power in multienergy power systems with an improved variational autoencoder[J]. *Applied Energy*, 2020, 274:115124.
- [252] Yuan, Ran, Wang, Bo, Sun, Yeqi, et al. Conditional style-based generative adversarial networks for renewable scenario generation[J]. *IEEE Transactions on Power Systems*, 2023, 38(2):1281–1296.
- [253] Kang, Mingyu. 外链补充材料 [EB/OL]. https://drive.google.com/file/d/1h7qt9I5r3-UylH7c-qwLL1MZW_D6qxos/view?usp=share_link. 2022.
- [254] Kang, Mingyu, Zhu, Ran, Chen, Duxin, et al. CM-GAN: a cross-modal generative adversarial network for imputing completely missing data in digital industry[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(3):2917–2926.
- [255] Chen, Xinyu, He, Zhaocheng, and Sun, Lijun. A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation[J]. *Transportation Research Part C: Emerging Technologies*, 2019, 98:73–84.
- [256] Chen, Xinyu, Lei, Mengying, Saunier, Nicolas, et al. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(8):12301–12310.
- [257] Wang, Chen, Wang, Ying, Ding, Zhetong, et al. A transformer-based method of multienergy load forecasting in integrated energy system[J]. *IEEE Transactions on Smart Grid*, 2022, 13(4):2703–2714.
- [258] Yao, Yang, Gu, Bo, Su, Zhou, et al. MVSTGN: a multi-view spatial-temporal graph network for cellular traffic prediction[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(5):2837–2849.

- [259] Zhao, Shan, Hu, Minghao, Cai, Zhiping, et al. Dynamic modeling cross-modal interactions in two-phase prediction for entity-relation extraction[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(3):1122–1131.
- [260] Zhang, Wenjie, Luo, Yonghong, Zhang, Ying, et al. SolarGAN: multivariate solar data imputation using generative adversarial network[J]. *IEEE Transactions on Sustainable Energy*, 2021, 12(1):743–746.
- [261] Simpson, E. H. The interpretation of interaction in contingency tables[J]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1951, 13(2):238–241.
- [262] Lewis, David K. *Counterfactuals*[M]. Blackwell, 1973.
- [263] Althauser, Robert P. and Rubin, Donald. The computerized construction of a matched sample[J]. *American Journal of Sociology*, 1970, 76(2):325–346.
- [264] Cochran, William G. and Rubin, Donald B. Controlling bias in observational studies: a review[J]. *Sankhyā: The Indian Journal of Statistics, Series A (1961–2002)*, 1973, 35(4):417–446.
- [265] Rubin, Donald B. Multivariate matching methods that are equal percent bias reducing, I: some examples[J]. *Biometrics*, 1976, 32(1):109–120.
- [266] Stuart, Elizabeth A. and Rubin, Donald B. Matching with multiple control groups with adjustment for group differences[J]. *Journal of Educational and Behavioral Statistics*, 2008, 33(3):279–306.
- [267] Rosenbaum, Paul R. and Rubin, Donald B. The central role of the propensity score in observational studies for causal effects[J]. *Biometrika*, 1983, 70(1):41–55.
- [268] Rubin, Donald B. and Thomas, Neal. Matching using estimated propensity scores: relating theory to practice[J]. *Biometrics*, 1996, 52(1):249–264.
- [269] Rubin, Donald B. Estimating causal effects from large data sets using propensity scores[J]. *Annals of Internal Medicine*, 1997, 127(8):757–763.
- [270] Caliendo, Marco and Kopeinig, Sabine. Some practical guidance for the implementation of propensity score matching[J]. *Journal of Economic Surveys*, 2008, 22(1):31–72.
- [271] Pearson, Karl. LIII. On lines and planes of closest fit to systems of points in space[J]. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, 2(11):559–572.
- [272] Cox, D. R. The regression analysis of binary sequences[J]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1958, 20(2):215–232.
- [273] Sims, Christopher A. Macroeconomics and reality[J]. *Econometrica*, 1980, 48(1):1–48.
- [274] Blanchard, Olivier J and Quah, Danny. The dynamic effects of aggregate demand and supply disturbances[EB/OL]. 1988.
- [275] Kolda, Tamara G. and Bader, Brett W. Tensor decompositions and applications[J]. *SIAM Review*, 2009, 51(3):455–500.
- [276] Acar, Evrim, Dunlavy, Daniel M., Kolda, Tamara G., et al. Scalable tensor factorizations for incomplete data[J]. *Chemometrics and Intelligent Laboratory Systems*, 2011, 106(1):41–56.
- [277] Aggarwal, Charu C et al. *Recommender Systems*[M]. Springer, 2016.

- [278] Rubin, Donald B. Randomization analysis of experimental data: The Fisher randomization test comment[J]. *Journal of the American Statistical Association*, 1980, 75(371):591–593.
- [279] Rubin, Donald B. Estimating causal effects of treatments in randomized and nonrandomized studies[J]. *Journal of Educational Psychology*, 1974, 66(5):688–701.
- [280] Pearl, Judea. *Causal Inference in Statistics: A Primer*[M]. Wiley, 2016.
- [281] Pearl, Judea. Bayesian networks: a model of self-activated memory for evidential reasoning[C]. In: *Proceedings of the 7th Conference of the Cognitive Science Society*, University of California, Irvine, CA, USA. 1985.
- [282] Pearl, Judea. Fusion, propagation, and structuring in belief networks[J]. *Artificial Intelligence*, 1986, 29(3): 241–288.
- [283] Pearl, Judea. Evidential reasoning using stochastic simulation of causal models[J]. *Artificial Intelligence*, 1987, 32(2):245–257.
- [284] Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*[M]. Morgan Kaufmann, 1988.
- [285] Geiger, Dan, Verma, Thomas, and Pearl, Judea. Identifying independence in bayesian networks[J]. *Networks*, 1990, 20(5):507–534.
- [286] Pearl, Judea. [Bayesian analysis in expert systems]: Comment: graphical models, causality and intervention[J]. *Statistical Science*, 1993, 8(3):266–269.
- [287] Pearl, Judea. Causal diagrams for empirical research[J]. *Biometrika*, 1995, 82(4):669–688.
- [288] Pearl, Judea. Causal inference in statistics: an overview[J]. *Statistics Surveys*, 2009, 3:96–146.
- [289] Glymour, Clark, Zhang, Kun, and Spirtes, Peter. Review of causal discovery methods based on graphical models[J]. *Frontiers in Genetics*, 2019, 10:524.
- [290] Schölkopf, Bernhard, Locatello, Francesco, Bauer, Stefan, et al. Toward causal representation learning[J]. *Proceedings of the IEEE*, 2021, 109(5):612–634.
- [291] Vowels, Matthew J., Camgoz, Necati Cihan, and Bowden, Richard. D'ya like DAGs? a survey on structure learning and causal discovery[J]. *ACM Computing Surveys*, 2022, 55(4):1–36.
- [292] Assaad, Charles K, Devijver, Emilie, and Gaussier, Eric. Survey and evaluation of causal discovery methods for time series[J]. *Journal of Artificial Intelligence Research*, 2022, 73:767–819.
- [293] Hasan, Uzma, Hossain, Emam, and Gani, Md Osman. A survey on causal discovery methods for i.i.d. and time series data[J]. *Transactions on Machine Learning Research*, 2023.
- [294] Lacerda, Gustavo, Spirtes, Peter, Ramsey, Joseph, et al. Discovering cyclic causal models by independent components analysis[C]. In: *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 2008. 366–374.
- [295] Spirtes, Peter. Directed cyclic graphical representations of feedback models[C]. In: *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 1995. 491–498.

- [296] Forré, Patrick and Mooij, Joris M. Markov properties for graphical models with cycles and latent variables[J]. arXiv preprint arXiv:1710.08775, 2017.
- [297] Bongers, Stephan, Forré, Patrick, Peters, Jonas, et al. Foundations of structural causal models with cycles and latent variables[J]. *The Annals of Statistics*, 2021, 49(5):2885 – 2915.
- [298] Koller, Daphne and Friedman, Nir. *Probabilistic Graphical Models: Principles and Techniques*[M]. MIT Press, 2009.
- [299] Barnard, Aubrey and Page, David. Causal structure learning via temporal Markov networks[C]. In: Proceedings of the Ninth International Conference on Probabilistic Graphical Models. PMLR, 2018. 13–24.
- [300] Baron, Reuben M and Kenny, David A. The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations[J]. *Journal of Personality and Social Psychology*, 1986, 51(6):1173–1182.
- [301] Imbens, Guido W. and Rubin, Donald B. Bayesian inference for causal effects in randomized experiments with noncompliance[J]. *The Annals of Statistics*, 1997, 25(1):305–327.
- [302] Mealli, Fabrizia and Rubin, Donald B. Assumptions allowing the estimation of direct causal effects[J]. *Journal of Econometrics*, 2003, 112(1):79–87.
- [303] Rubin, Donald B. Direct and indirect causal effects via potential outcomes[J]. *Scandinavian Journal of Statistics*, 2004, 31(2):161–170.
- [304] Hayes, Andrew F. *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach* (3rd Edition)[M]. Guilford Publications, 2022.
- [305] MacKinnon, David P. and Dwyer, James H. Estimating mediated effects in prevention studies[J]. *Evaluation Review*, 1993, 17(2):144–158.
- [306] MacKinnon, David P., Lockwood, Chondra M., and Williams, Jason. Confidence limits for the indirect effect: distribution of the product and resampling methods[J]. *Multivariate Behavioral Research*, 2004, 39(1):99–128.
- [307] MacKinnon, David P., Fairchild, Amanda J., and Fritz, Matthew S. Mediation analysis[J]. *Annual Review of Psychology*, 2007, 58:593–614.
- [308] Sobel, Michael E. Asymptotic confidence intervals for indirect effects in structural equation models[J]. *Sociological Methodology*, 1982, 13:290–312.
- [309] Sobel, Michael E. Some new results on indirect effects and their standard errors in covariance structure models[J]. *Sociological Methodology*, 1986, 16:159–186.
- [310] Cormen, Thomas H, Leiserson, Charles E, Rivest, Ronald L, et al. *Introduction to Algorithms* (4th Edition) [M]. MIT Press, 2022.
- [311] Hahn, Jinyong, Todd, Petra, and der Klaauw, Wilbert Van. Identification and estimation of treatment effects with a regression-discontinuity design[J]. *Econometrica*, 2001, 69(1):201–209.
- [312] McCrary, Justin. Manipulation of the running variable in the regression discontinuity design: a density test[J]. *Journal of Econometrics*, 2008, 142(2):698–714.

- [313] Bertrand, Marianne, Duflo, Esther, and Mullainathan, Sendhil. How much should we trust differences-in-differences estimates?[J]. *The Quarterly Journal of Economics*, 2004, 119(1):249–275.
- [314] Lechner, Michael. The estimation of causal effects by difference-in-difference methods[J]. *Foundations and Trends in Econometrics*, 2011, 4(3):165–224.
- [315] Ness, Robert, Paneri, Kaushal, and Vitek, Olga. Integrating Markov processes with structural causal modeling enables counterfactual inference in complex systems[C]. In: *Advances in Neural Information Processing Systems*. 2019.
- [316] Verma, Thomas and Pearl, Judea. Equivalence and synthesis of causal models[C]. In: *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. USA: , 1991. 255–270.
- [317] Dorf, Richard C. and Bishop, Robert H. *Modern Control Systems* (13th Edition)[M]. Pearson, 2016.
- [318] Sun, Jie, Taylor, Dane, and Bollt, Erik M. Causal network inference by optimal causation entropy[J]. *SIAM Journal on Applied Dynamical Systems*, 2015, 14(1):73–106.
- [319] Granger, Clive WJ. Investigating causal relations by econometric models and cross-spectral methods[J]. *Econometrica: Journal of the Econometric Society*, 1969, 37(3):424–438.
- [320] Geweke, John. Measurement of linear dependence and feedback between multiple time series[J]. *Journal of the American Statistical Association*, 1982, 77(378):304–313.
- [321] Geweke, John. Measures of conditional linear dependence and feedback between time series[J]. *Journal of the American Statistical Association*, 1984, 79(388):907–915.
- [322] Barnett, Lionel and Bossomaier, Terry. Transfer entropy as a log-likelihood ratio[J]. *Physical Review Letters*, 2012, 109(13):138105.
- [323] Schreiber, Thomas. Measuring information transfer[J]. *Physical Review Letters*, 2000, 85(2):461.
- [324] Kraskov, Alexander, Stögbauer, Harald, and Grassberger, Peter. Estimating mutual information[J]. *Physical Review E*, 2004, 69(6):066138.
- [325] Sugihara, George, May, Robert, Ye, Hao, et al. Detecting causality in complex ecosystems[J]. *Science*, 2012, 338(6106):496–500.
- [326] Runge, Jakob, Nowack, Peer, Kretschmer, Marlene, et al. Detecting and quantifying causal associations in large nonlinear time series datasets[J]. *Science Advances*, 2019, 5(11):eaau4996.
- [327] Spirtes, Peter and Glymour, Clark. An algorithm for fast recovery of sparse causal graphs[J]. *Social Science Computer Review*, 1991, 9(1):62–72.
- [328] Malinsky, Daniel and Spirtes, Peter. Causal structure learning from multivariate time series in settings with unmeasured confounding[C]. In: *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery*. PMLR, 2018. 23–47.
- [329] Pamfil, Roxana, Sriwattanaworachai, Nisara, Desai, Shaan, et al. DYNOTEARS: Structure learning from time-series data[C]. In: *International Conference on Artificial Intelligence and Statistics*. 2020. 1595–1605.
- [330] He, Yanyan, Kang, Mingyu, Chen, Duxin, et al. NHCE: a neural high-order causal entropy algorithm for disentangling coupling dynamics[J]. *IEEE Transactions on Network Science and Engineering*, 2024, 11(6): 5930–5942.

- [331] Donsker, M. D. and Varadhan, S. R. S. Asymptotic evaluation of certain Markov process expectations for large time-III[J]. Communications on Pure and Applied Mathematics, 1976, 29(4):389–461.

致谢

最应当感谢的是我的导师虞文武老师、副导师陈都鑫老师、外导高建喜老师和姚琦伟老师，师恩如山。正是在几位老师的共同培养和支持下，我才能够在学术上独当一面，开辟自己的学术道路，千言万语道不尽心中的热忱。此外还要特别感谢王和师兄、刘洪喆师兄和朱然师兄，给了我许多学术上的支持和帮助。然后就是感谢一下朱然、贺岩岩、韦梦立、孟宁、杨伊凡、邹燕等几位重要的论文合著者。然后还要感谢一下邓婕、点点、兔兔、祁老师、佳文、岩岩、孟宁、邵其、正哥、永健、伊凡，我第一次跟这么多好朋友一起创业，一起过生日，是正哥的瓜和邵其的火锅把我们聚在一起，1912的旋转木马，珍珠泉的秀丽山水，鱼嘴的夕阳，江上的风和奇迹，承载了太多回忆，我希望你们永远快乐。然后还要感谢东南大学提供了研究平台，这种支持不仅仅是指研究条件，更是指学术氛围。我想我毕业后无论过去多久都还会想起，没有什么能比光荣的东大标志更使我感动，大草坪上迷人的夜景，图书馆前壮丽的黄昏，阳光照耀下的梧桐叶，还有庄严巍峨的大礼堂。最后我想说，学术的乐趣在于不可预料的命运和未来，其光荣在于平淡，其艰巨在于漫长，唯一的正途就是强大。希望同志们能够前赴后继，在求知的道路上不断前进，向着胜利，直到永远。