arXiv:2304.03210v1 [q-bio.MN] 6 Apr 2023

基因组尺度生物网络恢复的因果发现与最优实验设计

Ashka Shah 芝加哥大学
美国伊利诺伊州芝加哥
shahashka
@uchicago.edu

Arvind Ramanathan 阿贡国家实验室 美国伊利诺伊州莱蒙特
ramanathana@anl.gov

Valerie Hayot-Sasson 芝加哥大学 美国伊利诺伊州芝加哥
vhayot@uchicago.edu

Rick Stevens 芝加哥大学 美国伊利诺伊州芝加哥
stevens
@cs.uchicago.edu

摘要

发现基因组尺度网络中的因果关系，对于识别从基因到可观测性状（例如细胞功能差异、疾病、耐药性等）的通路至关重要。基于图模型的因果学习器依赖干预样本来确定网络中边的方向。然而，尚无研究证明这些模型能够扩展至基因组规模，其数量级约为 10_3-10_4 个基因。我们提出了一种新的学习器 SP-GIES，它能够联合利用干预数据集和观察性数据集进行学习；对于包含 1,000 个节点的网络，其速度比现有学习器快近 4 倍。SP-GIES 在包含 1,000 个节点的网络上取得了 0.91 的 AUC-PR 分数，并可扩展至包含 2,000 个节点的网络，是现有研究所处理规模的 4 倍。我们还展示了 SP-GIES 如何改进下游最优实验设计策略，以选择要在系统上实施的干预实验。这是通过自主实验设计实现大规模因果发现的重要进展。

关键词

因果关系、结构学习、最优实验设计、基因型-表型映射

1 引言

生物网络描述生物系统中各变量之间的因果相互作用。这些变量可以是基因、转录因子、蛋白质和代谢物。这些变量之间的相互作用以及外部环境因素，共同决定了物种（或基因型）的基因组如何映射到可观测的性状（或表型）（见图 1）。从基因型预测表型是系统生物学的核心挑战之一（[22], [25], [16]）。基因型—表型模型有助于预测细胞功能、疾病、耐药性以及许多其他与生物学和公共卫生相关的问题。本文聚焦于恢复相互作用层级中第一层的生物网络（即基因调控网络），并将其称为“基因组尺度网络”，因为变量的数量对应于一个基因组中的基因数量。恢复基因组尺度网络对于理解表型变化的驱动因素和识别新的药物靶点至关重要。此外，深入了解基因组尺度网络还能提升下游表型预测的准确性模型，包括使用机器学习的模型（见图 1 中的潜在主题）。

生物网络是根据实验数据推断出来的。生物网络的恢复是一个逆向工程问题：给定实验数据，是什么样的网络产生了这些数据？现有的网络恢复方法主要分为两类：成对信息论方法和图模型方法。前者的优势在于它们具有高度并行性，并且已被证明对大规模生物数据集([7], [14], [18]) 有效。这些方法的主要缺点是它们只能从一种数据分布中学习——例如，在对系统实施干预后，它们无法纳入新的实验数据。第二类网络恢复方法是图模型，通常称为结构学习器。这些方法的优势在于它们具有直接的因果解释（边的方向意味着因果关系），因此已经形成了一系列能够联合学习观测数据分布和干预数据分布([10], [31], [29]) 的方法——我们称之为“联合学习器”。这种从不同分布中联合学习的能力，能够将统计模型（例如贝叶斯网络）转化为因果模型（例如因果贝叶斯网络）。图模型的缺点是，它们无法很好地随图中节点数量扩展（图搜索空间是组合式的），并且联合学习器没有并行实现。这意味着联合结构学习器尚未在大规模数据集上得到评估——我们至多观察到对包含500 个节点的网络进行的评估。鉴于大多数物种的基因组规模约为 10_3-10_4 个基因，因此有必要将这些联合结构学习器扩展到更大的网络。在本文中，我们介绍了一种新的混合方法结构学习器，名为 SP-GIES[1]，它兼具两类方法的优势。

因果模型优于统计模型，因为网络中的有向因果边使我们能够识别从基因型到表型的通路或作用机制。在生物学和医学应用中，研究人员希望同时对功能结果及其产生机制进行建模，从而理解这些机制，并通过疗法加以操控。鉴于这一优先事项，这类应用领域对因果模型有着迫切需求。因此，有必要纳入系统干预产生的数据。我们估计，可能的干预空间

[1]代码可在 https://github.com/shahashka/SP-GIES 获取。

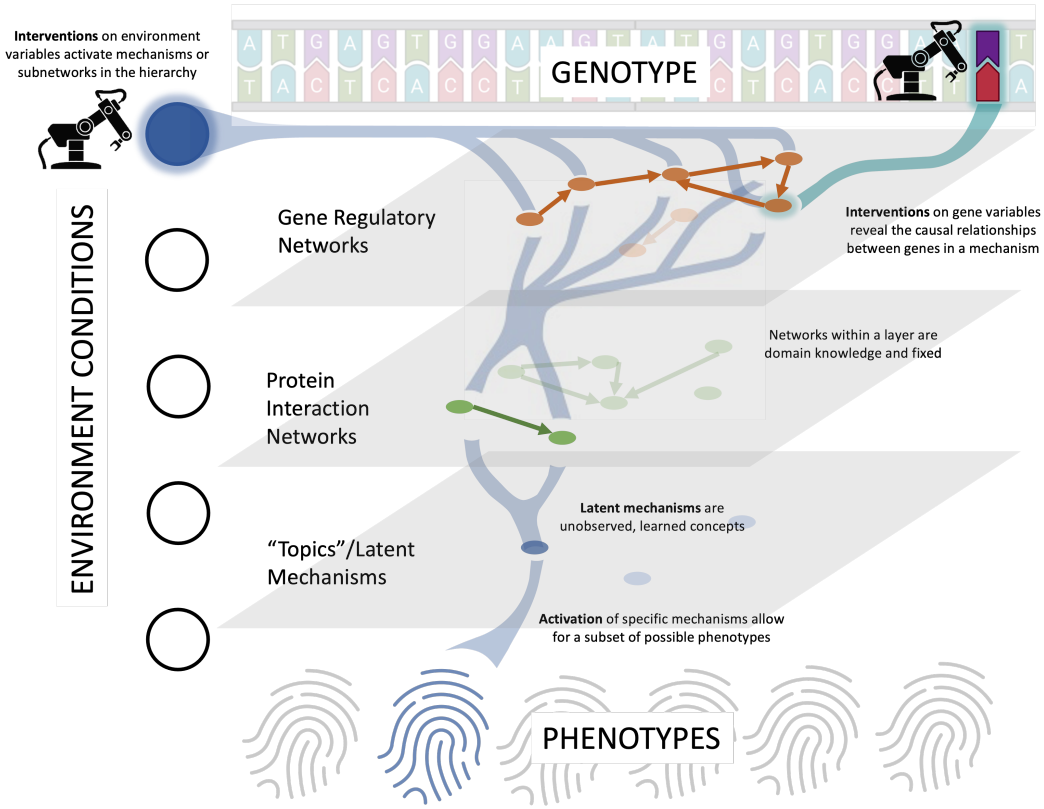

图 1：部分已知的生物网络层级。该层级中的各层可表示为因果网络。干预会激活作用机制，并揭示机制内部的因果关系。可能的干预空间十分庞大，因此需要采用自主设计循环，例如人工智能驱动的实验设计和机器人实验室。理论上，恢复所有网络即可实现完整的基因型到表型映射。然而在实践中，不可能完整捕获数据分布并控制混杂变量。因此，可将作用机制表示为主题潜变量，并通过主题建模进行学习。

在环境空间和基因空间中约为 10_4-10_5，具体取决于所研究的物种。对这一干预空间进行暴力采样，实验成本高昂且效率低下。因此，我们需要设计反馈循环，根据领域知识和已学得的知识选择有利的干预。最优实验设计（最优实验设计，OED）研究利用期望收益选择贝叶斯网络中的变量进行干预，以推动因果发现（[29], [11], [1], [8]）。这一范式使我们能够推理新获得的知识如何与现有领域知识相衔接，并为将人工智能融入科学领域提供前进方向。

本工作的贡献如下：

(1) 实现了一种名为骨架引导贪心干预等价搜索（SP-GIES）的新型混合结构学习方法，该方法基于现有的 GIES，能够联合利用观测数据和干预数据进行学习。与现有方法相比，SP-GIES 的网络恢复准确率更高，在大规模网络上的求解速度也更快。(2) 将 SP-GIES 应用于最优实验设计反馈循环，以选择针对基因的最优干预。(3) 分析这些方法的复杂度，并探讨未来的发展方向，包括构建统一的因果算法库，以通过干预实现基因组尺度的网络恢复，从而完成基因型到表型映射。

2 预备知识

2.1 因果贝叶斯网络

令 $G = (V, E)$ 为由一组顶点V和有向边 E定义的有向无环图。图中的顶点表示随-机变量 $X_1 ... X_p$。根据贝叶斯网络的马尔可夫假设，每个变量 X_i 在给定其父节点的条件下，都与其非后代条件独立。贝叶斯网络的联合分布

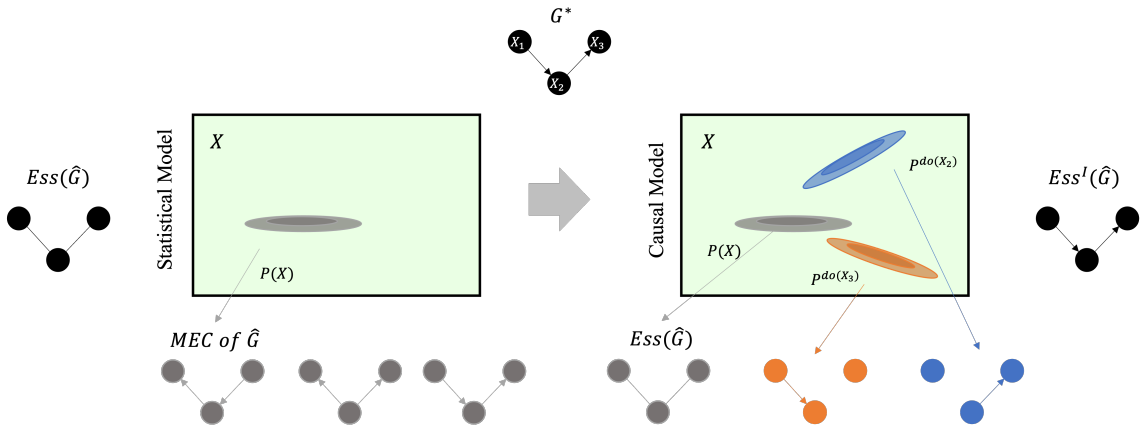

图2：如何利用干预数据将纯统计模型（例如贝叶斯网络）转换为因果模型（例如因果贝叶斯网络）。通过对干预分布进行建模，干预本质图 $Ess^I(G)$ 相比本质图 $Ess(G)$ 更接近真实的底层因果图。本图改编自 Schölkopf等人。[27]

网络分解为 $P(\mathbf{X}) = \prod_{i=1}^{p} P(X_i|Pa(X_i))$ [28]。以下定义描述了与结构学习相关的贝叶斯网络的重要性质。

定义*2.1*（马尔可夫等价类（*MEC*））。贝叶斯网络 G 由所有共享相同条件独立关系的有向无环图组成。

定义 *2.2*（本质图（*Ess*））。本质图 Ess(G) 是一个部分定向图，能够唯一表示某个有向无环图的 MEC。只有在等价类中方向一致的边才带有有向箭头，其余边均保持无向 [2]。

因果贝叶斯网络是一种边表示因果与效应关系的贝叶斯网络 [20]。例如，如果存在从 X_i 到 X_j的有向边，这意味着不存在未观测混杂变量导致它们之间的相关性，并且 X_i 是 X_j的直接原因。

2.2 基于观测数据的结构学习

根据一组实例（每个节点的采样值 X_i）构建图结构称为结构学习。根据数据学习图 $\hat{}G$ 时，采用以下假设。

(1) **因果充分性** - 所有随机变量均已观测到，即不存在隐变量 (2) **因果马尔可夫假设** - 数据由一组随机变量上的底层贝叶斯网络 (G^*, θ^*) 生成 X(3) **忠实性假设** - 由 (G^*, θ^*) 诱导的 P^* 关于 X的分布，除了 G^*结构所蕴含的独立性之外，不满足其他独立性条件

给定一个包含 $D = \{X_1 ... X_p\}$ 的数据集，该数据集包含来自 P^* 的 N 个样本——假设这些数据独立同分布（iid）。任务是学习一个模型 $\hat{}M = (\hat{}G, \hat{}\theta)$，该模型定义的分布 $\hat{}P$ 能够最佳拟合真实分布 P^* [13]。

同一 MEC 中的所有图都会产生相同的观测分布。因此，仅凭观测数据，我们所能做到的最好结果是恢复真实图的MEC。一些图模型方法仅从观测数据中学习结构。这些方法分为基于约束的方法和基于评分的方法。此外，还有仅从观测数据中学习的成对信息论方法。我们将在第 3 节讨论这些方法。

2.3 干预数据下的结构学习

除了基于观测数据进行结构学习之外，我们还希望将干预数据集纳入学习过程。对一组随机变量 $I \subseteq X$ 实施干预会移除指向随机变量 I的入边，并将联合分布设为新的干预分布 P^{do}（$X_i=x$）。这里，我们将节点 X_i设为值 x。在本文中，我们假设采用“硬”干预，即将节点设为一个常数值，因为这模拟了我们在生物学应用中能够实施的干预类型——例如基因敲除实验。这与“软”干预相反，后者是从某个分布中采样 $X_i = x$。

定义 *2.3*（干预分布）。硬干预后的联合分布为：

$$P^{do(X_i=x)} = \begin{cases} \prod_{j \neq i}^{p} P(X_j|Pa(X_j)) & if X_i = x \\ 0 & otherwise \end{cases}$$

联合利用观测数据和干预数据进行学习，可以正确识别因果关系并确定图中边的方向。基于观测数据估计出的 $Ess(G)$，可以进一步细化为 I-本质图（$Ess^I(G)$）。这是一种部分定向的有向无环图，表示一个干预马尔可夫等价类（I-MEC）（见图 2）。目前已有多种方法能够根据观测数据和干预数据联合估计结构。有关这些结构学习器的完整列表，请参阅 Vowels 等人 [30]。

表 1：本文涵盖的所有学习器的属性。方法按学习器的性质逐行分类。数据类型指学习器处理观测数据和/或干预数据的能力。评估数据集列出了各篇论文用于评估的基准数据集。最后，表中还列出了算法经评估所使用的最大节点数，以及相应论文报告的算法最坏情况复杂度。 p 表示节点数， n 表示样本数。 k 表示图中任意节点的最大度数。请注意，对于联合学习器，平均情况下的复杂度并非指数级；不过，精确计算取决于团的大小和网络拓扑。

		数据类型		评估数据集			扩展性	
	论文	观测。	干预。	随机	DREAM	大规模	最大节点数	最坏情况运行时间
成对信息论	*CLR* [7]	✓				✓	4,345	$O(p^3n^3)$
	ARACNE-AP [14]	✓				✓	1,331	$O(p^3+p^2n^2)$
	Pinna 等人 [23]		✓	✓	✓		100	$O(p^3+np)$
图模型	*PC* [34],[15], [17]	✓		✓	✓	✓	5,361	$O(p^{k+2})$
	GIES [10]	✓	✓	✓	✓		500	$O(2^p)$
	IGSP [31]	✓	✓				24	$O(2^p)$
	MCMC Mallows [24]	✓	✓		✓		10	$O(2^p)$
混合方法	*SP-GIES* [这篇论文]	✓	✓	✓	✓	✓	2,000	$O(2^p)$

表 2：论文中涵盖的最优实验设计方法的属性。最优实验设计准则指用于选择下一次实验的效用函数。请注意，此处列出的复杂度仅对应于选择下一次干预或干预集合。此处的每种算法还具有从后验分布中采样的成本，这等同于表 1 中学习器的复杂度。

	最优实验设计准则		评估数据集			扩展性	
论文	信息论	边定向	随机	DREAM	大规模	最大节点数	最坏情况运行时间
Ness 等人 [19]		✓		✓		17	$O(\|E\|*p!)$
Hauser 和 Bühlmann [11]		✓				40	多项式于 p
Tong 和 Koller [29]	✓					12	$O(pn)$
ABCD [1]	✓		✓	✓		10	$O(p^2n)$
BED [8]	✓		✓	✓		100	p的多项式

2.4 最优实验设计

我们可以通过对系统进行采样获得新的干预数据，但我们希望优先选择能够最快恢复真实因果图的干预实验。这可以通过在潜在 ˆ 干预集合上最大化期望效用函数 U 来实现：$I = \arg\max_I \mathbb{E}_{G|D}[U]$ 其中 U是一种效用函数，例如互信息、定向边数量等。

我们的目标是：给定干预样本和观测样本的混合数据，以及关于图中因果边的一些先验信息，恢复潜在的因果贝叶斯网络。我们将其表述为在有向无环图空间上的贝叶斯推断问题。我们设有先验 $P(G)$，用于编码关于潜在有向无环图的任何先验结构知识。应用贝叶斯定理可得到后验分布 $P(G|D) \propto P(D|G)P(G)$。似然为$P(D|G) = \int_\theta P(D,\ \theta|G)d\theta = \int_\theta P(D|\theta,\ G)P(\theta|G)d\theta$，其中我们已将图参数 [29] 边缘化掉。 D是观测数据和干预数据的混合。通过上一节所述的联合结构学习器对后验分布进行采样。使用最优实验设计从选定的干预中采样新数据后，将新数据与现有数据集连接起来，并使用结构学习器重新对后验进行采样。

3 相关工作

表 1 和表 2 分别总结了生物网络恢复和最优实验设计方面的相关工作。

3.1 利用干预数据进行生物网络恢复的结构学习

成对信息论学习器使用计算得到的指标来估计两个变量的依赖关系。通过对该指标应用阈值，可以获得系统的网络表示。Pinna等人 [23] 使用偏差矩阵（干预样本与稳态样本之间的差异）来估计初始网络。该算法赢得了DREAM4网络恢复挑战赛，并且在当时代表了基因调控网络重建的最先进水平。Faith等人 [7] 提出了CLR（相关性上下文似然），用于计算大肠杆菌K-12基因调控网络中基因之间的成对B样条互信息。CLR使用MATLAB实现，并配备了用于计算成对互信息的快速并行内核。Margolin等人 [18] 开发了ARACNE（准确细胞网络重建算法），这是一种类似的基于成对互信息的网络恢复算法。ARACNE还采用了DPI（数据处理不等式）算法，用于移除间接相互作用。随后，ARACNE升级为ARACNE-AP [14],，使用Java实现，以处理自适应分区并计算互信息——这使计算性能提升了200倍。

基于图模型的学习器比成对方法揭示的结构精细得多，因为它们基于贝叶斯网络。PC 算法是一种基于约束的结构学习器，只处理观测数据。PC 通过执行

条件独立性检验来学习图：检验成对节点在给定其他节点条件下是否独立。自 PC 算法问世以来，人们已为其开发了许多并行实现和优化方法（[15], [17]），其中包括一种使用 GPU 的实现（[34]）。现有的联合学习观测数据和干预数据的结构学习器（联合学习器）要么基于评分，要么采用评分与约束相结合的混合方法。Rau 等人 [24] 提出了一种名为 MCMC Mallows 的算法，该算法基于在潜在高斯有向无环图空间中进行的马尔可夫链蒙特卡洛采样，并对评分最高的有向无环图进行优化。MCMC Mallows 在 DREAM4网络上进行了评估，在其中两个网络上的表现优于 Pinna等人 [23] 的方法。Hauser 和 Bühlmann [10] 提出了贪婪干预等价搜索（GIES），这是一种贪婪的基于评分的方法，是原始 GES 学习器的扩展。GIES 在最坏情况下的复杂度关于 p （图中的节点数）呈指数级；不过，作者指出并通过实证表明，GIES 的效率高于这一最坏情况所暗示的水平。GIES 在 DREAM4 上进行了评估，其得分位居所有参与者的前三分之一。干预贪婪最稀疏置换（IGSP，Wang 等人 [31]）是 GSP 学习器的一种结合评分与约束的非参数扩展。IGSP 为随机变量的每一种排列关联一个有向无环图，并通过交换排列中的元素来贪婪地更新该图。IGSP 在蛋白质信号数据 [26] 上的表现与 GIES 相当，在包含 24 个基因的真实基因表达数据集上则优于 GIES。IGSP 在最坏情况下的复杂度也关于 p呈指数级。通过限制图中每个节点的最大度数，可以用多项式复杂度实现 IGSP 和 GIES；然而，对于含有枢纽节点（连接度较高的节点）的大规模网络，最大度数可能难以确定。我们在第4节中展示这些联合学习器与我们提出的学习器 SP-GIES 的实证扩展性比较。

3.2 生物网络恢复的最优实验设计

选择最优干预的一种方法，是选择能使互信息最大化或使熵减少最多的干预集合。Tong 和 Koller [29] 从图和参数的概率分布中采样一组排序。随后，他们利用这组排序计算熵项，并选择期望后验损失最低的干预。Agrawal等人 [1] 通过 ABCD策略扩展了 Tong 和 Koller [29] 的工作。ABCD策略是批量实验设计的一种贪心实现。他们使用基于干预预期熵减小量的效用函数。然而，这需要计算图空间和参数空间上的期望；他们通过自助法和加权重要性采样，设计出一种可处理的算法。

选择最优干预的另一种方法，是选择能够产生最多定向边的干预。Ness 等人 [19] 采用最优实验设计来恢复蛋白质信号网络 [26]。他们使用一种效用函数，该函数基于给定观测 MEC 和批次中其他干预时某项干预的期望信息增益。然而，该算法对批次大小呈阶乘级依赖。Ghassami 等人[8] 采用本质图中定向边的期望数量。

作为效用函数。本质图的真实网络首先使用类似 PC 算法的基于约束的方法进行估计。Hauser 和Bühlmann [11] 同样提出了一种基于骨架图的定向边数量的效用函数。

4 方法

这里我们介绍混合方法算法——骨架引导的贪心干预等价搜索（SP-GIES）。骨架是一个无向图，其边对应于有向无环图（DAG）中的边。该算法依次使用观测学习器估计骨架，再使用联合图模型结构学习器 GIES 来确定边的方向。该两步算法如下：

(1) 使用 ARACNE、CLR 或 PC，仅利用观测样本生成骨架
(2) 使用(1) 的输出限制 GIES 学习器的可能边集。使用观测数据和干预数据，通过 GIES 进行联合学习。

如果使用PC算法，则（2）的输入是一个Ess(G)，它由部分定向的有向无环图或PDAG 表示。对于（2），我们选择GIES，因为它有一个开源实现，是 R 中广泛使用的pcalg库的一部分。对于扩展性研究，我们选择使用PC 算法的 R 实现。这是因为 CLR 和 ARACNE 分别使用MATLAB和 Java 实现。由于 GIES 只有R 实现，因此在扩展性研究中，我们选择使用 PC 的 R 实现。

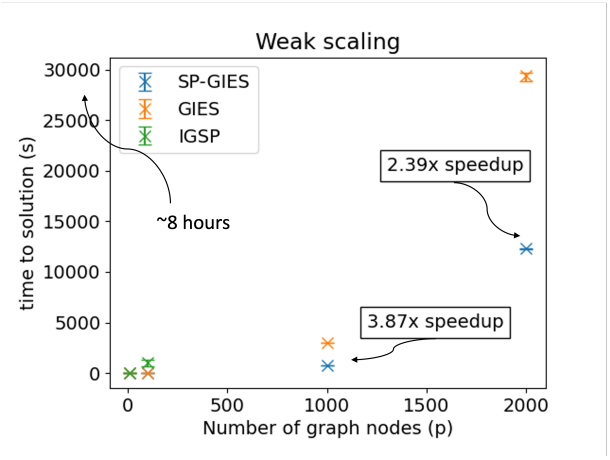

图 3：比较 SP-GIES、GIES 和 IGSP 算法的弱扩展研究。样本数量固定为n=1,000。数据点为 3 次运行的平均值。研究在一台 2.8 Ghz AMD EPYC Milan 7543P 32 核 CPU 和一块 NVIDIA A100 GPU 上进行。IGSP 在 1,000 和 2,000 个节点上的运行超出了我们的配额，因此未在此处绘制。

虽然（1）很容易在多处理器上并行化或部署到 GPU 上，但计算瓶颈在于 GIES 步骤（见图 4）。仍看图 3，我们可以看到，SP-GIES 比 GIES 更快地得到结果。为了解其中原因，我们研究了 GIES 算法。GIES 是一种贪婪评分型结构学习器。评分函数是贝叶斯信息

表 3：在规模为 10 的随机网络上对学习器进行评估。生成了三种不同类型的随机网络：Erdős–Rényi [6], 无标度网络 [4], 和小世界 [32]。 p 和 k 指用于生成图的参数，而非此前所用的节点数和度数。结果取自 30 个随机图的平均值，每个图均使用 100 个数据样本进行学习。算法上的上标表示所用的数据类型（O 表示观测数据，OI 表示观测数据与干预数据的混合）。对于 SP-GIES，分别对 Erdős–Rényi、无标度网络和小世界网络使用 ARACNE、CLR 和 PC 进行骨架估计，因为这些方法的表现最佳。

规模为 10 的随机网络									
算法	Erdős–Rényi（$p = 0.5$）			无标度网络（$k = 2$）			小世界（$p = 0.5, k = 2$）		
	SHD	SID	AUC-PR	SHD	SID	AUC-PR	SHD	SID	AUC-PR
PC^O	22.77	70.73	0.35	11.27	61.00	0.62	8.23	36.33	0.59
CLR^O	22.67	71.43	0.34	17.27	68.30	0.35	10.83	51.17	0.43
ARACNE-APO	21.83	79.00	0.36	16.63	76.70	0.37	10.10	52.80	0.43
GES^O	23.27	53.34	0.47	**11.20**	**28.57**	**0.69**	10.23	**17.27**	0.67
GIESOI	23.93	**39.23**	0.59	14.13	36.43	0.68	21.30	17.70	0.55
PinnaOI	22.67	57.60	0.26	18.00	56.93	0.28	13.97	29.73	0.13
IGSPOI	**20.00**	67.27	0.38	15.00	53.23	0.41	6.73	22.43	0.55
SP-GIESOI	24.00	39.87	0.59	11.34	48.53	0.64	**5.90**	26.53	**0.68**
NULL	21.00	66.83	**0.61**	16.00	61.30	0.58	10.00	30.00	0.55

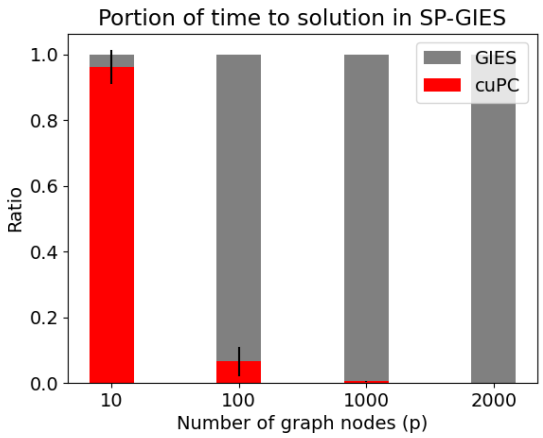

图 4：SP-GIES算法各步骤的运行时间占比。步骤（1）是 cuPC，步骤（2）是 GIES。可以看到，随着图规模增大，瓶颈在于 GIES 步骤。

该准则基于给定当前数据时图的最大似然估计。从一个空本质图开始，在算法的前向阶段，向 $Ess(G)$ 添加一条边，使新本质图的得分最高。算法遍历所有可能的边，直到找到得分最高的相应本质图。后向阶段与此类似，但操作是删除边而非添加边。通过先估计骨架来限制 GIES 算法的可能边集合，我们缩小了两个阶段的搜索空间，从而实现加速。此外，由于我们使用了 PC 算法的快速 GPU 实现（ Zarebavani 等人， [34]），预先估计骨架所带来的开销极小。

表 1 显示，所有联合学习器的最坏情况复杂度均呈指数级——这归因于组合搜索空间在图上。然而，平均经验复杂度与网络的拓扑结构有关，即与最大团的大小有关。由于平均复杂度难以计算，我们在图 3 中展示了，随着网络规模增大，SP-GIES 达到解的速度更快。我们参照 Hauser 和 Bühlmann [10] 的类似研究，对 SP-GIES 进行了扩展性研究，不过我们将规模扩展到了 p=2,000 个节点，而 GIES 论文中的规模为 p=500 个节点。在我们分配的 72 小时运行时限内，IGSP 未能完成针对 1,000 和 2,000 个节点的任务。我们发现，由于前文讨论的边集限制，SP-GIES 达到解的速度比 IGSP 和 GIES 都快。

5 数据集

我们在不同规模的数据集上将 SP-GIES 与现有学习器进行了比较。随机网络使用 networkxPython库生成。观测数据基于线性高斯模型生成。我们生成了100 个观测样本，并为每个节点生成了一个干预样本。DREAM4计算机模拟挑战赛提供了通过随机微分方程合成生成的基因调控网络的观测数据和基因敲除干预数据。该数据集包含 10 个观测样本，并为每个节点提供一个干预样本。真实网络是更大的E.coli 基因调控网络的子网络。RegulonDB 数据集由Faith 等人提供 [7]。该网络是当前估计的 E.coliK12 基因调控网络，包含1146 个节点、3179条边和 524个真实实验样本。在这 524 个实验样本中，有 35个样本取自基因敲除实验。RegulonDB 数据集还包含环境干预。现有学习器目前仅对基因变量进行建模，因此任何环境干预样本都被视为观测数据。未来的工作包括对环境条件变量进行显式建模。

每个学习器均通过三项指标进行评估：结构汉明距离（SHD）、结构干预距离（SID）和AUC-PR。SHD是生成的有向无环图之间的L1误差

表4：对规模为10的DREAM4网络上的学习器进行评估。该数据集包含11个观测样本和10个干预样本。干预样本来自基因敲除实验。对于SP-GIES，骨架估计使用了ARACNE，但网络4使用的是CLR。请注意，对于网络3，我们在GIES子程序中使用了自适应学习（adaptive="triples"），以使SP-GIES学习器达到最佳性能。对于其他网络，使用自适应学习并未带来显著改进。

DREAM4计算机模拟规模 10																
算法	网络1				网络2				网络3				网络4				网络5			
	结构	汉明	距离	SID AUC-PR	结构	汉明	距离	SID AUC-PR	结构	汉明	距离	SID AUC-PR	结构	汉明	距离	SID AUC-PR	结构	汉明	距离	SID AUC-PR
PC^O	14	39	0.31		12	56	0.43		16	64	0.26		14	45	0.16		11	57	**0.59**	
CLR^O	13	36	0.45		**10**	56	0.53		14	62	0.36		14	47	0.19		15	62	0.33	
ARACNE-APO	**11**	28	0.56		12	56	0.45		13	61	0.43		15	45	0.06		12	57	0.46	
GES^O	13	29	0.49		15	52	0.30		17	64	0.15		19	48	0.10		11	51	0.14	
GIESOI	13	31	0.51		12	49	0.51		16	**40**	0.49		15	**26**	0.34		9	39	0.37	
PinnaOI	**11**	**22**	0.49		11	56	0.08		14	61	0.57		**12**	42	0.37		11	38	0.47	
IGSPOI	13	27	0.07		13	54	0.08		16	65	0.07		15	38	0.06		13	48	0.19	
SP-GIESOI	**11**	23	**0.63**		**10**	**34**	**0.6**		**12**	47	**0.63**		**12**	35	0.33		**6**	**30**	0.45	
NULL	12	27	0.57		11	53	0.58		14	61	0.57		**12**	36	**0.56**		11	46	0.56	

表5：学习器在大规模网络上的评估。Pinna等人 [23] 要求每个基因有一个干预样本；由于RegulonDB数据集不提供此类样本，我们未在该数据集上评估Pinna。GESO需要将最大度设置为10才能收敛。IGSP在72小时内未能在这些网络上收敛。

大规模网络的评估						
算法	RegulonDB 1146 个节点，3179 条边			小世界 1000 个节点，1000 条边		
	SHD	SID	AUC-PR	SHD	SID	AUC-PR
ARACNE-APO	3,752	25,979	0.04	1,000	3,883	0.50
CLR^O	**3,095**	**18,069**	0.30	2,620	116,560	0.09
PC^O	3,963	23,908	0.01	467	2,447	0.82
GES^O	8,712	74,224	0.005	1,523	**879**	0.88
GIESOI	8,355	80,580	0.006	1,623	1,097	0.84
PinnaOI	x	x	x	16,577	4,334	0.002
SP-GIESOI	3,154	22,114	0.10	**341**	975	**0.91**
NULL	3,179	18,294	**0.50**	1,000	3,883	0.50

以及真实有向无环图。SID统计由公式2.3 定义的错误干预分布数量。AUC-PR是精确率-召回率曲线下的面积。这三个指标都是因果发现中常用的评估指标。然而，我们注意到，SHD 和 AUC-PR 都偏向于空图。假设一个结构学习器 A 学习了一个 G_A，其中没有边，并且具有 $SHD_A = |E|$。另一个学习器 B 学习了一个 G_B，其中具有 $SHD_B > |E|$，但正确学习出了真实图中的一些因果边。根据SHD， G_A 更好，因此我们可能会认为，对于这些数据而言， A 是更好的学习器。然而，我们也可以为 G_B 提出论据，因为它捕捉到了比 G_A 更多的因果关系。这意味着，较低的 SHD 并不总是表示一个最符合数据分布的模型。对于 AUC-PR，也可以提出类似的论点。SID 的缺点是，对于稀疏网络，其计算复杂度关于节点数量呈二次增长；对于稠密网络，则呈三次增长。Peters和 Bühlmann [21]通过实验证明了这一点，最多涉及 50 个节点。

6 结果

我们在随机网络、DREAM4 网络和基因组尺度网络上的评估结果分别如表 3、表 4 和表 5 所示。我们评估的观测学习器包括 PC、ARACNE 和 CLR。我们评估的干预学习器包括 GIES、IGSP、Pinna 和 SP-GIES。由于蒙特卡洛抽样存在可扩展性方面的挑战，我们未将 MCMC Mallows 纳入评估。Rau 等人 [24] 对 MCMC Mallows 在小型 DREAM4 和随机数据集上的表现进行了评估，可供参考。此外，我们还纳入了一个不含任何依赖关系（无边）的网络得分作为参照，并将其命名为 NULL。对于 CLR 和 ARACNE，必须选择一个阈值来筛选边。对于随机网络和 DREAM4 网络的 CLR，我们选择的阈值使保留的边约为排名前 10% 的边。对于 RegulonDB 数据集的 CLR，我们遵循 Faith 等人 [7] 的研究，选择了能达到 60% 精确率的阈值。对于 ARACNE，我们使用了与 p 值 10^{-8} 相对应的阈值——这是 Lachmann 等人 [14] 使用的默认值。

对于规模为 10 的随机网络，表现最佳的学习器取决于网络类型。在无标度网络上，不使用干预数据的 GES 表现最佳；在小世界网络上，SP-GIES 表现最佳。对于 Erdős–Rényi 网络，GIES 和 SP-GIES 的表现相当。对于 DREAM4，在全部五个网络和所有指标类型中，SP-GIES 最常获得最佳分数。我们发现，加入干预数据能够提升

能够处理这两种数据类型的算法（即 GIES 和 SP-GIES）的表现。

对于 RegulonDB 网络，得分最高的方法是CLR。这是因为CLR 方法会计算一个阈值来剪枝边，使学习器相较于真实网络达到60% 的精确率——这使得比较不公平，因为真实网络必须事先已知。我们使用CLR 对 SP-GIES 进行骨架估计。一个出乎意料的结果是，SP-GIES似乎使 CLR 骨架的初始估计变得更差。为了理解这一点，我们放大研究了 RegulonDB中的一个子网络，该子网络包含三个枢纽（连接非常多的节点）。图 5显示，CLR 正确识别出了该子网络中的3 个枢纽。然而，SP- GIES 从这一初始估计中移除了许多边，并在其他位置添加了新边。因此，使用 SP-GIES时未检测到任何枢纽。一个可能的原因是，GIES学习器使用最大似然估计为每个候选图评分。该计算假设数据采样自线性高斯分布——但对于 RegulonDB 这类真实数据集而言，情况并非如此，因为其中可能存在非线性效应。由于互信息能够捕捉非线性，我们在 CLR 中没有看到这一问题的体现。为了检验这一理论，我们根据RegulonDB 子网络的拓扑结构生成了合成线性高斯数据。图 5显示，使用合成数据时，SP-GIES能够检测到该子网络中的3 个枢纽。数据集的非线性只能解释部分现象，因为与 CLR 估计相比，使用合成数据得到的SP-GIES估计仍然更偏离真实网络。对于 SP-GIES 这类联合学习器的适用范围而言，还需要进一步分析其中的数据条件和网络拓扑结构。

对于我们在此评估的少数几个网络，空值图取得了最佳得分。许多现实世界的网络，包括基因调控网络，都是稀疏的。因此，空值学习器通常表现得非常出色，因为真实网络在组合空间中接近空网络。有趣的是，在 SID 指标下，空值图从未取得最佳得分。这可能表明，因果网络恢复方法应使用 SID 而不是其他指标进行评估。然而，要理解这一结论的有效性，还需要进一步分析，而这超出了本文的范围。

为了了解 SP-GIES 在最优实验设计方面相较于 GIES 的性能提升，我们在 DREAM4计算机模拟 10 节点数据集上，使用两种不同的最优实验设计准则评估了这两种算法。由于最优实验设计的时间复杂度涉及后验采样，以及每轮干预时的最优基因选择，我们将评估限制在小型网络上。我们遵循 [1], 的框架，其中包括对后验分布进行自助采样。图 6 展示了一个 DREAM4 计算机模拟网络的示例结果。我们观察到三种趋势。首先，边方向确定策略的表现优于信息增益策略。其次，SP-GIES学习器提升了学习器的初始性能，但很容易陷入停滞；随着干预样本增加，算法的性能提升速度无法持续保持在与基于 GIES 的策略相同的水平。尽管如此，SP-GIES x OED 的运行表现仍优于 GIES x OED，而且加入干预数据后，准确率也有所提高。我们推测，SP-GIES 联合学习器可能会陷入停滞，因为固定骨架的同时也限制了学习器的搜索空间。这可能导致算法中的 GIES 步骤

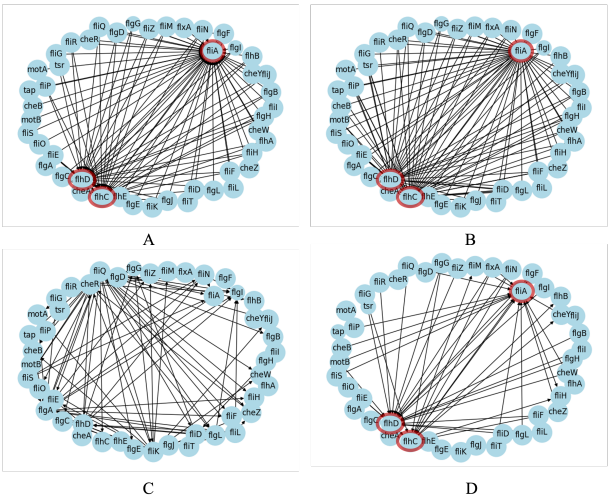

图 5：（A）由RegulonDB数据集给出的真实网络子网络，其中包含 3 个以红色突出显示的枢纽。（B）由 CLR 估计的无向子网络，正确识别了这 3 个枢纽。阈值的选择使得真实网络中有 60% 得到恢复。（C）使用 CLR 骨架由 SP-GIES 估计的子网络。对于真实数据，未检测到枢纽。（D）使用合成数据和 CLR 骨架由 SP-GIES 估计的子网络。检测到了这 3 个枢纽，但该图比 CLR 估计结果和真实网络稀疏得多。

陷入局部最优。这促使我们考虑在骨架中纳入一定程度的不确定性，或将骨架之外的边集随机纳入搜索空间——这留待未来工作探讨。第三，我们发现随机策略的表现与最优实验设计策略相当。换言之，优先进行某些实验而非其他实验，并不一定能更准确地估计因果网络。一种可能的解释是，网络拓扑可能导致对某些节点进行干预并不比对其他节点更有利。另一种原因是，最优实验设计策略依赖良好的模型（因果图）来估计后验分布。缺乏良好模型时，效用函数的计算就会不准确，进而可能优先选择错误的实验。尽管与 GIES 相比，SP-GIES 能更准确地估计因果图，但这可能仍不足以支持最优实验设计。这与我们在主动学习中观察到的情况类似：当模型性能不佳时，随机标注新样本有助于改进模型。

7 讨论

理解基因组尺度网络对于建立从基因型到表型的因果映射非常重要。然而，由于能够联合从混合数据中学习的结-构学习器具有较高的计算复杂度，从干预数据集恢复基因组尺度网络尚未实现。现有方法，如 GIES、IGSP、MCMCMallows 及其他方法，仅在最多 500 个节点的小型网络上进行评估，并表现出最坏情况下的指数级复杂度

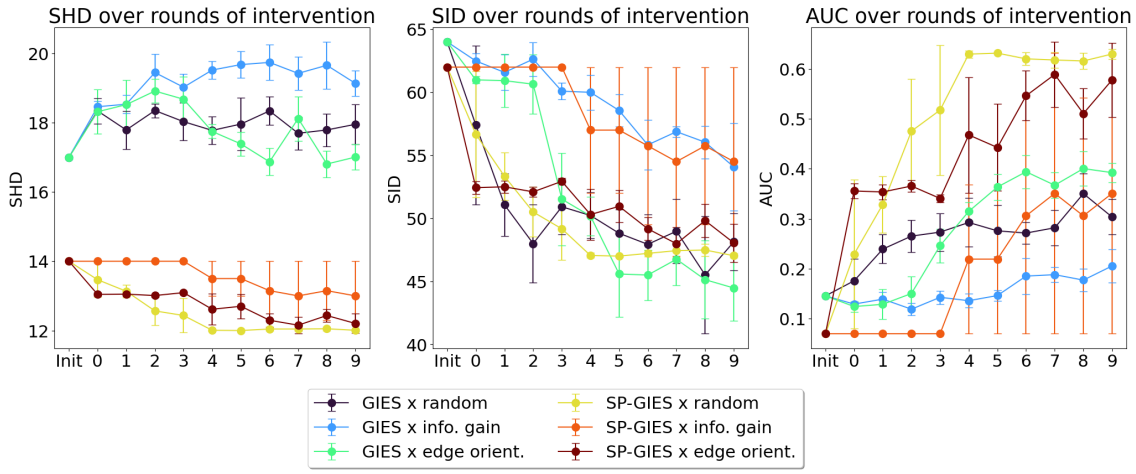

图 6：对 DREAM410 节点网络#3在10轮干预中的最优实验设计策略进行评估。在每轮干预中，将与所选干预相对应的一个新的干预数据样本添加到当前数据集中。GIES和 SP-GIES 是联合学习器。指标取 30 次运行的平均值。

复杂度。这使得后续的最优实验设计技术在计算上难以处理，因为复杂度还要乘上选择最优干预的计算量。这是通过自主设计循环大规模实现因果发现时面临的一个巨大问题。

在这篇论文中，我们指出，为观测数据集构建的结构学习器已有适用于多处理器和 GPU 的并行实现。我们的 SP-GIES 学习器首先利用这些并行实现估计骨架。这会限制后续联合结构学习器的边集，并缩短求解时间。然而，SP-GIES 仍受联合学习器 GIES 扩展性的瓶颈制约——例如，运行包含 10_4 个节点的网络，24 小时后仍未完成。要实现对规模约为 10_3-10_4的基因组尺度网络的恢复，并对规模约为 10_4-10_5的干预空间进行采样，我们必须在联合学习器的分布式并行实现方面取得突破。目前，联合结构学习器尚无并行实现，因为它们是基于评分的学习器（或约束与评分混合型学习器）。在算法的每一步中，都会对候选图进行评分，通常使用基于整个图的似然 $P(D|G)$ 的函数。正如我们在第 2 节所见，有向无环图的联合分布是条件概率的乘积，而每个节点的条件概率取决于其父节点。因此，似然的计算是顺序进行的，通常采用贪心策略对这一空间进行优化。如何将图划分为多个子图以供各个计算内核处理，并不直观。另一方面，基于约束的方法使用成对条件独立性检验来确定边。一种简单的扩展性提升方法是为每项计算单独生成一个计算线程；不过，也可以将变量分组为多个块，以进一步减少计算量并提升性能。基于评分的学习器尚未实现这种性能提升，这也成为现有联合学习器面临的障碍。

SP-GIES在RegulonDB数据集（这是与生物学最相关的数据集）上的表现表明，未来的研究也应将非线性纳入联合学习器。一种方法是修改GIES评分函数。由于只有少数函数存在最大似然估计的闭式解，另一种选择是使用非参数学习器，例如IGSP。然而，IGSP无法扩展到RegulonDB数据集的规模。此外，IGSP学习器在随机网络和DREAM4网络上的表现总体看来较差。我们注意到，该算法的应用场景与本文使用的数据集截然不同。IGSP专为具有足够干预数据样本的情况而设计，以确保假设检验能够成功。对于本文使用的数据集，每个节点只有少量干预样本。Gretton等人 [9]、、Yu等人 [33]、、Zheng等人 [35]等人的研究探讨了非线性数据集的结构学习。近期研究通过将组合优化重新表述为连续优化来使用神经网络。然而，这些方法在低数据量场景下表现不佳。要实现真实生物数据集上的大规模网络恢复，就需要将非线性结构学习器中的策略纳入联合学习器。

在评估现有方法并设计 SP-GIES 方法时，我们发现学习器和最优实验设计策略由多种语言实现，包括 Java、MATLAB、R、Python 和 C。这是因为，对生物网络进行因果发现所涉及的研究领域多种多样，包括生物学、统计学、计算机科学、机器学习和表征学习。近期，人们开始关注统一因果发现工具，例如 CausalDiscoveryToolbox [12]、、Pgmpy [3]和 CausalDAG [5] 都是 Python 库，分别以不同程度支持图模型和/或因果建模。然而，在 CausalDiscoveryToolbox 中，该库是 R 代码的 Python 封装，而 R 代码又是 C 代码的封装。对于 Pgmpy，目前尚未实现联合干预学习器。CausalDAG 为带干预的因果发现提供了整体上最佳的框架，尽管许多实现仍不完整。为了更好地并行实现因果算法和最优实验设计♀♀♀♀♀♀

策略，我们应鼓励与超级计算社区开展合作。因此，有必要为这些模型和算法提供专门的 Python 或 C 实现，因为这些语言在超级计算社区中很受欢迎。

8 结论

我们提出了 SP-GIES，这是一种联合结构学习器，它利用并行观测学习器来估计骨架，并初始化 GIES 联合学习器。我们针对不同规模的数据集和各种指标，对 SP-GIES 与现有方法进行了系统评估。结果表明，对于规模最大为 2,000 个节点的网络，以及最大为 1,146 个节点的生物网络，SP-GIES 均能提供更高的网络恢复准确率。据我们所知，联合学习器此前尚未实现这种规模的网络恢复。与现有方法 GIES 相比，SP-GIES 能更快地得出结果，最高可实现 3.87 倍加速。我们还展示了 SP-GIES 能够改进后续的最优实验设计策略。未来的工作包括实现 SP-GIES 的分布式并行版本，并将 SP-GIES 纳入自主实验设计循环中。

致谢

本研究作为 CANDLE 项目的一部分，获得了 DOE-Exascale Computing Project（17-SC-20-SC）的支持。本研究使用了阿贡领导计算设施的资源。该设施是 DOE 科学办公室用户设施，由合同 DE- AC02-06CH11357 提供支持；此外，本研究还获得了美国国立卫生研究院国家过敏和传染病研究所资助编号 P01AI165077（AR）的支持。本项目全部或部分由联邦资金资助，资金来自美国卫生与公众服务部下属美国国立卫生研究院国家过敏和传染病研究所，合同编号 75N93019C00076，授予芝加哥大学。本项目全部或部分由联邦资金资助，资金来自能源部科学办公室生物与环境研究计划，FWP# 34903，授予阿贡国家实验室。

参考文献

[1] Raj Agrawal、Chandler Squires、Karren Yang、Karthikeyan Shanmugam 和 Caroline Uhler。2019。ABCD-strategy：面向目标因果结构发现的预算约束实验设计。载于第 *22* 届国际人工智能与统计学会议。PMLR，3400–3409。[2] Steen A Andersson、David Madigan 和 Michael D Perlman。1997。无环有向图的马尔可夫等价类刻画。《统计学年鉴》25，2（1997），505–541。[3] Ankur Ankan 和 Abinash Panda。2015。Pgmpy：使用 Python 的概率图模型。载于第 *14* 届 *Python* 科学会议论文集（*SCIPY 2015*）。Citeseer。[4] Albert-László Barabási 和 Eric Bonabeau。2003。无标度网络。《科学美国人》288，5（2003），60–69。[5] Chandler Squires。2018。CausalDAG：因果模型的创建、操作与学习。https://github.com/uhlerlab/causaldag [6] Paul Erdős、Alfréd Rényi 等。1960。关于随机图的演化。匈牙利科学院数学研究所出版物5，1（1960），17–60。[7]Jeremiah J Faith、Boris Hayete、Joshua T Thaden、Ilaria Mogno、Jamey Wierzbowski、Guillaume Cottarel、Simon Kasif、James J Collins 和 Timothy S Gardner。2007。通过表达谱汇编对大肠杆菌转录调控进行大规模绘制与验证。《*PLoS* 生物学》5，1（2007），e8。[8] AmirEmad Ghassami、Saber Salehkaleybar、Negar Kiyavash 和 Elias Bareinboim。2018。面向因果结构学习的预算约束实验设计。载于国际机器学习会议。PMLR，1724–1733。[9] Arthur Gretton、Peter Spirtes 和 Robert Tillman。2009。弱加性噪声模型下的非线性有向无环结构学习。《神经信息处理系统进展》第 22 卷（2009）。[10] Alain Hauser 和 Peter Bühlmann。2012。有向无环图干预马尔可夫等价类的刻画与贪心学习。《机器学习研究杂志》13，1（2012），2409–2464。[11] Alain Hauser 和 Peter Bühlmann。2014。利用干预数据主动学习因果模型的两种最优策略。《近似推理国际期刊》55，4（2014），926–939。[12] Diviyan Kalainathan 和 Olivier Goudet。2019。因果发现工具箱：在 Python 中揭示因果关系。arXiv 预印本 arXiv:1903.02278（2019）。[13] Daphne Koller 和 Nir Friedman。2009。概率图模型：原理与技术。MIT 出版社。[14] Alexander Lachmann、Federico M Giorgi、Gonzalo Lopez 和 Andrea Califano。2016。ARACNE-AP：通过互信息的自适应分区推断进行基因网络反向工程。《生物信息学》32，14（2016），2233–2235。[15] Thuc Duy Le、Tao Hoang、Jiuyong Li、Lin Liu、Huawen Liu 和 Shu Hu。2016。采用多核处理器的高维因果发现快速 PC 算法。《IEEE/ACM 计算生物学与生物信息学汇刊》16，5（2016），1483–1495。[16] Nathan E Lewis、Harish Nagarajan 和 Bernhard O Palsson。2012。利用计算机模拟方法的系统发育约束代谢基因型–表型关系。《自然评论：微生物学》10，4（2012），291–305。[17] Anders L Madsen、Frank Jensen、Antonio Salmerón、Helge Langseth 和 Thomas D Nielsen。2017。基于大规模数据集的贝叶斯网络结构学习并行算法。《知识型系统》117（2017），46–55。[18] Adam A Margolin、Ilya Nemenman、Katia Basso、Chris Wiggins、Gustavo Stolovitzky、Riccardo Dalla Favera 和 Andrea Califano。2006。ARACNE：哺乳动物细胞环境中基因调控网络重建算法。载于《BMC 生物信息学》第 7 卷。BioMed Central，1–15。[19] Robert Osazuwa Ness、Karen Sachs、Parag Mallick 和 Olga Vitek。2017。用于推断信号网络的贝叶斯主动学习实验设计。载于计算分子生物学研究国际会议。Springer，134–156。[20] Judea Pearl。1995。从贝叶斯网络到因果网络。载于《人工智能中处理部分知识的数学模型》。Springer，157–182。[21] Jonas Peters 和 Peter Bühlmann。2015。用于评估因果图的结构干预距离。《神经计算》27，3（2015），771–799。[22] Massimo Pigliucci。2010。基因型–表型映射与“基因即蓝图”隐喻的终结。《英国皇家学会哲学汇刊 B：生物科学》365，1540（2010），557–566。[23] Andrea Pinna、Nicola Soranzo 和 Alberto De La Fuente。2010。从基因敲除到网络：通过图分析建立直接因果关系。《PloS one》5，10（2010），e12912。[24] Andrea Rau、Florence Jaffrézic 和 Grégory Nuel。2013。从观测数据和干预基因表达数据联合估计因果效应。《BMC 系统生物学》7，1（2013），1–12。[25] Marylyn D Ritchie、Emily R Holzinger、Ruowang Li、Sarah A Pendergrass 和 Dokyoon Kim。2015。整合数据以揭示基因型–表型相互作用的方法。《自然评论：遗传学》16，2（2015），85–97。[26] Karen Sachs、Omar Perez、Dana Pe'er、Douglas A Lauffenburger 和 Garry P Nolan。2005。从多参数单细胞数据推导因果蛋白信号网络。《科学》308，5721（2005），523–529。[27] Bernhard Schölkopf、Francesco Locatello、Stefan Bauer、Nan Rosemary Ke、Nal Kalchbrenner、Anirudh Goyal 和 Yoshua Bengio。2021。迈向因果表征学习。《CoRR》abs/2102.11107（2021）。arXiv:2102.11107 https://arxiv.org/abs/2102.11107。P. Spirtes、C. Glymour 和 R. Scheines。2000。[28]《因果、预测与搜索》（第 2 版）。MIT 出版社。[29] Simon Tong 和 Daphne Koller。2001。贝叶斯网络中的结构主动学习。载于人工智能国际联合会议，第 17 卷。Citeseer，863–869。[30] Matthew J Vowels、Necati Cihan Camgoz 和 Richard Bowden。2021。你喜欢 DAG 吗？结构学习与因果发现综述。《ACM 计算调查》（CSUR）（2021）。[31] Yuhao Wang、Liam Solus、Karren Yang 和 Caroline Uhler。2017。带干预的基于置换的因果推断算法。《神经信息处理系统进展》第 30 卷（2017）。[32] Duncan J Watts 和 Steven H Strogatz。1998。“小世界”网络的集体动力学。《自然》393，6684（1998），440–442。[33] Yue Yu、Jie Chen、Tian Gao 和 Mo Yu。2019。DAG-GNN：利用图神经网络进行 DAG 结构学习。载于国际机器学习会议。PMLR，7154–7163。[34] Behrooz Zarebavani、Foad Jafarinejad、Matin Hashemi 和 Saber Salehkaleybar。2019。cuPC：用于 GPU 上因果结构学习的基于 CUDA 的并行 PC 算法。《IEEE 并行与分布式系统汇刊》31，3（2019），530–542。[35] Xun Zheng、Bryon Aragam、Pradeep K Ravikumar 和 Eric P Xing。2018。无撕裂的 DAG：用于结构学习的连续优化。《神经信息处理系统进展》

信息处理系统31 (2018)。