第六篇

评价学应用

第二十五章

评价学驱动的人工智能

本章阐述了人工智能的基础概念、基本假设及问题定义。在众多的人工智能范式中，本章重点聚焦于数据驱动的深度学习范式。尽管该范式目前处于主流地位，但其内在的“黑盒”属性仍存在可解释性挑战。

本章由康国新、高婉铃和我共同探讨、阐述和实现。

25.1 现有人工智能范式的缺点

早期的人工智能研究由符号主义范式（Symbolic Paradigm）主导。该范式的核心观点为人类智能可以通过符号逻辑与显式规则得以完整表征 [205, 204, 213, 95]。这一范式为图灵测试 [313] 奠定了理论基础，将智能定义为“符号处理”的能力。专家系统是符号主义的巅峰，其核心是将人类经验形式化为逻辑规约，并植入规则驱动的推理引擎中 [40, 69, 185, 65]。然而，此类系统在扩展性方面存在固有缺陷，且由于缺乏从数据中自主学习的能力，使其应用场景受到了极大限制。

上述局限性促使了联结主义范式（Connectionist Paradigm）的兴起。该范式受到生物神经网络的启发 [257, 120, 125]。赫布（Hebb）在其开创性理论中，将突触调节与学习行为联系起来，为神经科学与机器学习之间架起了一座理论桥梁 [115]。在多层神经架构与高效训练算法的基础上，深度学习应运而生，自此模型能够从大规模数据中自动挖掘统计规律 [165, 107, 170, 318, 74]。

区别于符号主义与早期联结主义，当代人工智能范式已逐步转向数据驱动的深度学习。该范式基于一种核心假说。

深度学习假说 1（数据深度表示假说）. *通过对海量数据集统计规律的深度表示，可实现对人类智能的有效逼近 [147]。*

这种以数据为中心的原理在大语言模型中得到了最为显著的印证 [37]，且其性能表现遵循尺度定律（Scaling Laws）——即性能增益与训练规模、参数量及计算算力的扩张呈现出高度可预测的线性或幂律关系 [149]。然而，单纯依赖规模扩张的路径正面临愈发严峻的“数据瓶颈”。在高质量人工标注数据趋于饱和且成本高昂的背景下，合成数据生成正成为极具前景的替代方案。

尽管合成数据具有良好的可扩展性，但它也引入了新的复杂问题 [281, 199, 21]。合成数据的质量从根本上受限于数据的生成模型，而这些模型通常采用黑盒架构，缺乏透

明度与可解释性。这种可解释性的缺失使得研究者难以将下游模型中出现的误差或偏差，追溯至合成数据中特定的属性根源。当模型性能下降时，究竟是由于数据覆盖度不足、语义一致性缺失，还是更深层的表示缺陷所致，依然难以归因和溯源。

总结而言，当前的合成数据存在以下显著缺陷：首先，生成模型可能无法准确拟合真实数据的统计分布，由此引入的偏差会严重损害模型的泛化能力；其次，合成样本常含有难以察觉的逻辑矛盾或特征畸变，这些瑕疵可能对预训练过程造成负面干扰。此外，合成数据面临多样性匮乏与模态崩溃的固有风险。生成器往往会产生变化有限的样本，导致模型过度拟合于狭窄的模式，而在真实世界多样的数据上表现不佳。

为了提升合成数据的质量，增强生成模型的可解释性已刻不容缓。若无法理解生成模型究竟学习到了哪些知识，又系统性地忽略了哪些信息，那么盲目扩张合成语料库规模将极易引入伪相关，并导致模型失配。

这些观察结果推动了向基于评价学（Evaluatology）的 AI 范式转变，其中系统的归因分析和可解释性不再是事后补充，而是 AI 开发周期的核心组成部分。无论数据来源如何，所有数据本质上都是在特定条件下生成的。然而，当前主流的 AI 训练方法在很大程度上忽略了这些生成条件，而仅仅专注于数据本身。这种缺陷导致数据质量参差不齐，且难以评价，限制了模型的可解释性和因果发现能力，并使模型在面对新场景时变得脆弱。

我们的研究直觉是将数据及其求索条件明确纳入训练过程，可显著提升人工智能的有效性与透明度。即使在数据有限的情况下，利用数据与求索条件之间的相互作用，仍能揭示更深层的因果结构，使模型能够捕捉数据多样性背后不变的信息本质。学习建立在求索条件感知的因果关系基础上，我们正迈向更具可解释性、可归因性且真正智能的系统。

25.2 深度学习的基础概念和问题定义

我们介绍数据驱动的深度学习范式的基础概念和基本原则。

25.2.1 基础概念

深度学习概念定义 1 (模型架构（Model Architecture）)**.** 在数据驱动的人工智能范式中，模型架构通常指深度神经网络，其本质是实现输入与输出间映射关系的函数近似器。深度神经网络架构通过协同训练数据规模与计算资源的持续增长，实现模型性能的有效增长 *[171, 318, 101]*。

深度学习概念定义 2 (数据集（Data Set）)**.** 数据集由大规模有标签或无标签的样本构成，是模型训练的核心和基础 *[165, 109, 302]*。

深度学习概念定义 3 (损失函数（Loss Function）)**.** 损失函数 $\mathcal{L}$ 用于量化模型输出与基准事实（*ground truth*）之间的差值。模型训练的核心目标在于最小化该函数在给定数据集上的累积误差，即 $\min \mathcal{L}(\mathcal{D};\theta)$。从而使模型能够学习输入与输出之间的映射关系 *[257, 101, 56, 29]*。

25.2.2 问题定义

问题定义 13 (深度学习问题定义)**.** 在训练数据集 $\mathcal{D}$ 充足、模型参数规模 θ 足够大且计算资源 $\mathcal{C}$ 不构成瓶颈的条件下，深度学习模型 f_θ 通常可以通过最小化经验风险 *(Empirical Risk Minimization, ERM)* 有效地应对日益复杂的现实任务 $\mathcal{T}$ *[316, 170, 101, 149, 124]*:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\mathcal{D}; \theta), \tag{25.1}$$

其中 $\mathcal{L}$ 表示损失函数，$\hat{\theta}$ 则代表通过最小化经验风险 *(Empirical Risk Minimization, ERM)* 所获得的优化模型参数。通过准确度等统计指标，可对模型性能 $(Performance)$ 进行测试:

$$Performance = \mathcal{M}(f_{\hat{\theta}}, \mathcal{T}), \tag{25.2}$$

其中 $\mathcal{M}$ 表示定量的统计指标，且假设计算成本 $\mathcal{C}$ 与模型参数量及训练数据规模成比例增长 *[149]*:

$$\mathcal{C} \propto \theta \cdot \mathcal{D}. \tag{25.3}$$

然而，这些性能指标是通过黑盒模型得到的，难以提供深入的因果可解释性 *[181, 76, 224]*。

25.3 基于评价学的强人工智能路径

基于评价学的人工智能范式推动人工智能研究发生深刻转变：研究问题不再只是“模型性能是否足够好”，而是在“在什么评价条件下性能好”、“为什么模型性能好”以及“什么关键设计改变会让模型性能更好”。

25.3.1 自包含系统

为突破上述困境，新范式从现实世界系统中隔离出自包含系统。自包含系统是可运行、可实验的简化系统。它把评价对象、影响对象和混杂对象集合视为一个整体。其形式化表示如下式。

$$\mathrm{SCS} = \{\,\mathrm{EO},\ \mathrm{AO},\ \mathrm{COs}\,\}. \tag{25.4}$$

其中评价对象是被评价和设计的对象，具体的例子包括视频检索网络、编码器-解码器架构或推荐算法等算法架构，设计的目标是在海量的配置空间中找到使总体影响最优的评价对象配置。

混杂对象和评价对象一起向影响对象（AO）施加影响。混杂对象定义了模型运行的上下文环境，包括但不限于训练数据、实验配置、超参和环境变量等。

为确保训练数据具备可溯源性与可复现性，无论是基于观察还是实验所得的训练样本，均须在明确的求索条件下产生，包括但不限于量的定义、实现与赋值，数据产生的具体场景、采集流程以及潜在偏差等。

基于自包含系统，新范式可以探索评价对象（原因对象）的设计空间。这一探索可以分为三步。暴力枚举确保对设计空间的全面覆盖。启发式方法利用先验因果知识聚焦高潜力区域。剪枝剔除冗余或无效的设计路径。这三步使新范式以更低成本系统地探索设计空间。

25.3.2 对象因果模型

基于自包含系统可以建立对象因果模型。对象因果模型刻画评价对象、混杂对象和影响对象之间的影响机制。它的核心任务是从总体影响中剥离评价对象的真实影响。

新范式通过基础求索活动建立对象因果模型。测试和测量量化模型在不同配置下的性能。评价阐明特定设计导致性能变化的原因，并且从总体影响中分离评价对象的真实影响。这一过程产生科学可解释的因果理解。

在对象因果模型下，可以进一步探索评价与设计这一对偶关系。评价问题固定评价对象，变化评价条件，旨在揭示评价对象的真实影响。设计问题固定评价条件，变化评价对象，旨在寻找实现最优总体影响的配置。

借助这一因果理解，新范式可以实现智能化。它首先依据已学得的因果原理探索多种替代设计，作为新的评价对象。它最终执行自动设计，寻找实现最优总体影响的配置。由此，人工智能从因果理解走向自主的设计。这正是基于评价学的强人工智能路径。

25.4 案例研究

我们提供了一个案例展示基于评价学的人工智能范式如何加速数据库的自动设计。

在数据库自动设计中，评价对象是数据库索引 [49]。影响对象是最小独立运行的数据库系统。混杂对象集合涵盖影响索引性能的对象。这些对象或者其结构和属性包括数据分布与偏斜模式、模式演变与更新频率、存储布局以及数据压缩策略。

受 CPU 评价学的启发 [323]，混杂对象并非一成不变。负载与访问分布会根据利益相关方的需求动态调整。这使评价能够反映真实的生产环境变化，而不依赖静态的评价基准。

为隔离索引的真实影响，新范式构建自包含系统。它把索引、数据库系统和混杂对象集合视为一个整体。在此系统中，新范式探索索引的设计空间。以行存数据库为例，加速用户数据访问通常需要在选定列上构建索引。对于一个包含 n 列的数据表，若允许任意选择列子集及其顺序，设计空间会组合爆炸。其空间复杂度如公式 25.5。其中 k 表示索引所包含的列数，取值从 1 到 n。

$$\sum_{k=1}^{n} \frac{n!}{(n-k)!}. \tag{25.5}$$

该空间呈阶乘级增长，在计算上难以处理。新范式通过三步降低计算量。暴力枚举接近穷举。启发式探索最可能被访问的列，并在这些列上建立索引，剪枝剔除冗余或低效的设计路径。

在自包含系统之上，新范式建立索引的对象因果模型。它通过基础求索分离索引的真实影响。测试和测量量化索引在不同工作负载下的性能。评价从总体影响中推断每个候选索引的真实影响，并阐明特定索引结构导致性能提升或下降的原因。这一过程产生对索引设计如何影响系统性能的科学理解。

借助这一因果理解，新范式进入智能设计阶段。它通过自我演进替代的索引形式，生成新的结构变体。它执行自动设计，创造最能满足利益相关方需求的索引。这超越了从 B 树、哈希或位图等现有模板中选择的局限，有可能支持全新索引结构的创新，从而实现评价学驱动的数据库智能设计。

25.5 总结

本章提出了基于评价学的人工智能范式。该范式把智能的形成建立在两个核心概念之上：自包含系统和对象因果模型。该范式利用评价与设计之间的对偶关系：评价时固定评价对象，变化评价条件；自动设计则固定评价条件，变化评价对象。沿着这条路径，人工智能得以从不透明的数据驱动黑盒，走向可解释、基于因果且自我演进的智能。

第二十六章

评价学在大模型中的应用

本章基于我们已经发表的论文 [310] 改写。在这个工作 [310] 的基础上，我们也基于评价学方法发布了大模型排行榜（https://llmeval.benchcouncil.org/），为理论框架的验证与持续迭代提供了可复现、可对比的实证平台。本章由高婉铃撰写。

大模型评价（LLM evaluation）是在给定任务与交互设定下，依据模型对测试实例的响应，对其能力、行为与可用性作出判断的过程。它广泛存在于通用知识推理、数学与代码、指令遵循、多模态理解、安全对齐、人类偏好比较以及面向部署的服务评测等场景中。

传统做法通常将评价问题表述为：在固定评价基准（benchmark）上运行模型、与参考答案或偏好信息对照，并用准确性、得分等指标排序 [116, 52, 238]。

在基础模型快速迭代之后，社区进一步提出了大量标准化测试集、动态刷新 benchmark、人工竞技场与统一评测框架 [137, 142]，使“测什么、怎么测”在形式上日益丰富。

然而，从评价学（evaluatology）视角看，大模型评价研究面临的核心困难并不只是“benchmark 分数能否进一步提高”，而是“分数的变化究竟由什么对象、在什么评价条件下产生”。

当前大量实验仍采用以下方式来比较结果：在若干公开数据集上，将完整模型或完整测试脚本作为黑盒，用单一默认配置下的准确性、pass@k 或胜率进行排序。这种做法虽然简洁，却难以回答更关键的问题：测试到的性能来自被评价模型本身，还是来自数据集选择、提示模板、思维链（CoT）设置、解码参数、语言与格式，乃至软件硬件栈所引入的混杂？

已有工作表明，仅改变提示、解码或数据视图即可使同一模型的报告性能出现大幅波动 [26, 283, 209]；若评价对象（EO）与影响对象（AO）未被清晰界定，则指标越多，反而越容易掩盖真实影响来源，并将 benchmark 表现与模型能力混为一谈。

本章以评价学为基础，讨论大模型的评价 [310]。核心观点是：LLM 评价不应仅被当作“模型 + benchmark”的整体黑盒，而应被置于自包含系统（self-contained system, SCS）之中，明确区分评价对象（EO）、影响对象（AO）、混杂对象（COs），并在评价条件（EC）容许的配置空间下，揭示评价对象及其内部组件对 AO 的影响。

26.1　大模型的传统评价方式及其局限

既往大模型评价在操作层面主要体现为：在公开或自建 benchmark 上运行候选模型，依据参考答案、规则判定、人工标注或偏好，完成模型比较与结果发布。

相应流程既涵盖固定题库上的自动判分，也涵盖交互式偏好采集、持续更新的任务实例，以及更接近应用情境的闭环评测；判分机制既包括确定性匹配与单元测试，也包括模型辅助评判。尽管外在形态多样，其内在组织方式具有较强一致性，即将测试结果压缩为少量汇总的指标，并将该指标视为模型的能力表征。

在这一组织方式下，benchmark 通常对任务内容、输入输出形式及评分准则作出较为明确的规定，而对提示设计、是否启用分步推理、输出格式约束、解码超参数、交互轮次、结果解析方式及运行环境等过程性配置，则往往缺乏统一、完整的约束。因而，公开报告中的得分虽然在名义上表示模型在某一 benchmark 上的性能，实质上反映的是模型在特定配置下得到的测试结果。当不同机构、不同版本或不同实验脚本在上述过程中存在配置上的差异时，分数的不同并不能直接体现模型能力的差异。

传统做法在产业迭代与学术沟通中发挥了重要作用，但其局限源自评价方法本身，具有结构性，并非仅由个别 benchmark 或个别实现疏漏所致。

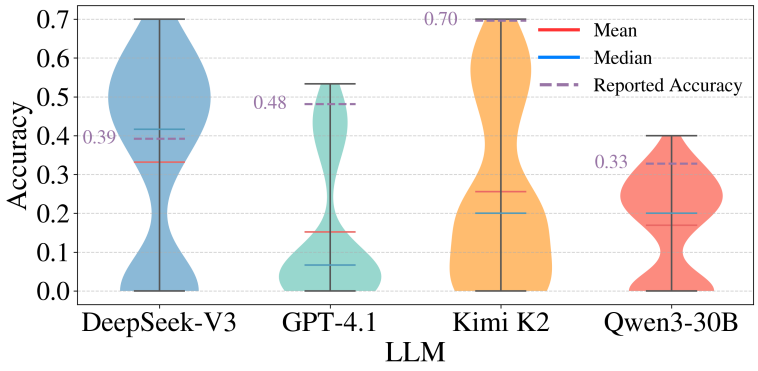


图 26.1: 固定数据集和模型下的准确性差异：162 种评价配置组合 [310]。图上的 Reported Accuracy 是官方报告的测试结果，其中 DeepSeek-V3 源自 [182]，GPT-4.1 源自 [216]，Kimi K2 源自 [305]，Qwen3-30B 源自 [333]。

- 第一，评价缺少严谨的方法。公开结果多数仅仅报告某模型在某一 benchmark 上达到某一分数，却不揭示，也自然无法再现其所采用的提示策略、解码设定、交互方式及判分流程。
- 第二，单点配置报告难以反映配置空间中的结果分布。如图 26.1所示，当过程性配置可在合理范围内变动时，同一模型在同一 benchmark 上的表现应被理解为分布于不同评价条件配置之下，而非必然对应单一数值。仅报告默认值、官方报告值或某一最优配置下的结果，既不呈现离散程度，也难以揭示模型排序随配置改变而发生反转的情形。
- 第三，静态实例上的正确性与能力判断之间缺少必要的区分。模型在固定测试实例上的正确响应，可能源于真实的推理能力，也可能源于题型熟悉、模板匹配或训练

数据与测试数据之间的内容关联。已有研究表明，题干改写、数值替换或干扰构造等轻微变动即可引起性能显著波动，说明传统静态得分对模型有效性或者泛化能力的指示能力有限，在开放域、多语言与多风格应用场景中尤为明显。

- *第四，单因素经验难以支撑系统层面的解释。*关于提示、格式、语言、解码等因素影响的讨论虽已较为丰富，但多发生于彼此独立的实验设定之中，较少在统一定义的自包含系统配置空间内，系统分析多因素并存时的主效应、交互效应（采用第 十四章介绍的 DoE 方法）及其跨模型稳定性。其结果是，局部改进经验难以上升为可泛化、可检验的系统模型。

- *第五，不同评价和测试范式之间缺少统一的描述框架。*参考答案驱动方式便于自动化，但在开放生成任务上依赖规则或模型打分；人类偏好评测更贴近使用体验，却受采样人群与交互协议制约；环境化与动态 benchmark 更接近部署场景，又引入额外协议变量。传统实践常在各类范式之间切换与采用，却较少在同一描述体系下同时报告评价和测试系统的配置、结果分布与评价结果差异来源。

综上，传统大模型评价的主要不足并不在于未能给出分数，而在于缺少严谨的方法获得与报告分数。这个报告分数实质上是通过对影响对象的测试获得的评价结果。基于现有方法的排序结果，无法再现完整的评价和测试系统，亦难以判断评价结果的差异应归因于模型本身还是评价和测试过程的配置。后续讨论将在这一认识基础上，进一步说明如何将评价从结果汇总变为揭示原因对象的影响，并在此理论框架下提供可解释性。

26.2　大模型评价的自包含系统

在评价学中，评价的核心是揭示评价对象对影响对象产生的影响。对于大模型评价，基于评价学实验方法的核心是构建自包含系统（self-contained system,），使评价对象、评价条件与评价结果之间的关系得以明确表述。

评价对象（*EO*）是被评价的核心对象，记为 O。它既可以是完整的大模型系统，也可以是在特定部署形态下实际参与响应生成的系统单元。例如，在比较两个开源模型孰优孰劣时，O 通常是被测模型本身；在考察 API 推理服务是否稳定可复现时，O 可定义为包含模型权重、推理引擎及相关运行支撑的统一系统；若研究问题转向某一子模块（如对齐模块、工具调用接口或后处理组件）的独立贡献，则该子模块亦可被指定为 O。

影响对象（*AO*）是获得模型评分所依赖的最小完整运行系统，其中包括被测大模型、推理与调度软件、benchmark 执行与判分流程，以及相应的硬件与系统环境。准确性、通过率、胜率、Elo 值、偏好得分或任务成功率等，均是在这一整体系统中通过实验直接得到的测试结果，而不是脱离该系统单独存在的抽象数值。

混杂对象集合（*COs*）是除 O 之外仍会对 AO 产生影响的对象集合。在大模型评价中，COs 可包括：评测基准的版本与实例构成、参考答案与标注规范、自动或人工评判规则、作为裁判的辅助模型、与 O 协同工作的检索器或外部工具、训练语料与测试内容之间的关联结构，以及硬件平台与推理实现带来的非确定性因素等。若这些对象未被明确建模或控制，AO 上观测到的结果波动中就会混入 COs 的作用，从而使 O 的独立影响难以隔离。

评价条件（*EC*）是在给定 SCS 中移除 O 后形成的评价组件及其具体配置，记为 c。对于大模型评价，EC 可实例化为 benchmark 及任务子集选择、提示组织方式、是否启

用分步推理、交互轮次、输出格式约束、解码超参数、结果解析与判分流程、随机种子、并发与超时设置、互补组件开关以及计算预算等。若 O 为某一大模型，则其所面对的题库版本、提示模板、解码设定及评分流程均属于 EC 的组成部分；若 O 为包含推理服务的系统单元，则服务参数、批处理策略与接口层配置同样应纳入 EC。EC 所包含的可容许配置空间，构成完整评价系统的规格化描述；仅给出 benchmark 名称，通常不足以唯一确定 c。

用形式化语言表示，在评价条件 c 下，评价对象 O 对 AO 产生的总体影响记为:

$$OE(O \mid c). \tag{26.1}$$

它即构成可测试的评价结果。在大模型评价中，$OE(O \mid c)$ 可由准确性、通过率、偏好胜率、任务成功率等指标表示。若使用损失函数 $\mathcal{L}$ 表示误差型或风险型指标，则 $\mathcal{L}(O, c)$ 是 $OE(O \mid c)$ 的一个具体测量形式。这说明任何一次 benchmark 得分或 leaderboard 排序，都不是 O 的孤立属性，而是 O 与 EC 共同作用后、在 AO 上产生的可测试结果。若忽略 EC，就容易把评价系统配置变化误认为对象能力变化，进而使模型比较失去可解释基础。

在本章所讨论的大模型评价场景中，评价的目标仍是揭示评价对象 O 在评价条件 c 下对 AO 的影响；但这一影响并非直接测量得到，而是通过测试在 AO 上获得的。测试要求对象处于运行或执行状态，使模型对测试实例产生响应，并将该响应与预期结果、规则判定或任务行为进行比较；因此，准确性、通过率、胜率、Elo 值等，本质上都是测试结果，而不是脱离运行过程单独存在的抽象分数。单次测试以检验为基础，即运行测试用例并将其输出与检验基准的预期结果进行比对。在此层次上，测试与测量共同构成量的赋值手段：测量获得对象的结构属性，测试获得对象在特定条件下必须运行后才能显现的行为属性。对应地，影响既可能体现为测量结果的变化，也可能体现为测试结果的变化；在大模型评价中，后者更为常见。于是，评价、测试与测量之间的关系可概括为：测量或测试为影响对象变化的量赋值，而评价揭示影响。这一区分有助于说明：benchmark 分数并非模型的结构属性，而是可运行系统在特定 EC 下执行测试后的结果。

26.3 评价条件的组件化分解

在自包含系统框架下，大模型评价并非仅由评价对象 O 单独决定，而是由 EC 中若干彼此耦合的组成组件共同塑造。前一节已将 EC 界定为移除 O 后的衍生评价系统。

本节进一步说明：为何需要将 EC 进行组件化分解，以及在大模型场景中应识别哪些关键组成组件。组件化分解的基本出发点是：一次可重复的大模型测试，通常同时涉及“测什么”、“如何问”、“如何生成”以及“如何判定”等多个彼此独立的决策。若这些决策未被分别定义，则 AO 上的测试结果波动无法归因于具体的条件，模型比较也容易退化为不确定条件下的比较。因此，组件化分解的目标，不是增加评价流程的复杂度，而是将完整评价系统拆分为可命名、可配置、可对照的组成单元，使 $OE(O \mid c)$ 中的 c 具有明确结构，而非仅由一个 benchmark 名称间接指代。

26.3.1 分解原则

大模型评价条件（EC）的组件划分应满足三点要求。第一，结果相关性：该组件对 AO 上的测试结果具有实质影响，而非纯实现细节。第二，可独立配置：在固定其他组件的

情况下，该组件可以在有限候选值之间切换，从而支持对照实验。第三，边界清晰：该组件与 O 的边界明确，避免将模型能力本身误写为配置项。

满足上述条件的组件，可视为 EC 的基本组件；其全部合法取值构成该组件的配置域，多个组件配置域的笛卡尔积则形成可容许的评价条件配置空间。需要指出，组件化分解并不预设各组件彼此独立。恰恰相反，大模型评价中的许多影响恰恰来自组件之间的联合作用；分解的意义在于为后续分析提供结构化坐标，而非事先假定某个单因素对于评价的充分性。

26.3.2 三类功能组成

结合大模型评价的常见流程，EC 宜理解为由三类功能组成的组件。三类之间按职责划分，与具体实现中的模块顺序不必一一对应。

任务与实例组件。 这类组件回答“测什么”，主要影响进入 AO 的输入实例及其呈现方式。典型组件包括：

- 语言选择（*language*）：测试实例采用何种自然语言或跨语言设定；
- 题型结构（*question format*）：如选择题、填空题或其他可解析作答格式；
- 实例改写（*question paraphrase*）：是否在保持语义与参考答案不变的前提下，对题干作等价改写，以检验模型是否依赖固定表述或表面记忆。

其中，实例改写并非改变测试目标，而是改变实例的可辨识结构，用于区分模型的真实推理能力，还是模型只是在预训练阶段记住了答案。

交互与提示组件。 这类组件回答“如何向模型提问并组织上下文”，主要影响模型接收到的指令形式与推理路径。典型组件包括：

- 示例提示（*few-shot*）：是否提供示范样例及其数量设定；
- 思维链（*CoT*）：是否要求模型给出中间推理过程；
- 多轮交互（*multi-turn*）：是否允许多轮追问、澄清或补答。

这些组件共同决定模型在 AO 中接收到的交互界面，而不等同于模型权重本身。

生成与控制组件。 这类组件回答“模型如何产生输出”，主要影响响应的随机性、长度与重复行为。典型组件包括：

- 温度（*temperature*）：控制采样随机性；
- *nucleus* 采样（*top_p*）：控制候选词集的累积概率阈值；
- 存在惩罚（*presence penalty*）：抑制重复主题或重复片段；
- 最大 *token* 生成长度（*max tokens*）：限制输出规模，影响完整性、成本与延迟。

上述参数在开源的本地部署推理系统与 API 服务中均可配置，是 AO 运行配置中影响结果的关键组件，而实际取值在实践中又常采用默认配置。

除上述三类外，AO 中还通常存在*解析与判分组件*，如答案抽取规则、正则匹配、单元测试、模型辅助裁判或人工标注流程。严格而言，它们也是评价系统不可缺少的环节；在组件化分解中，可视其是否作为 EC 的对照对象，纳入 EC 描述，或作为 AO 的固定结构单独说明。本书后续讨论中，若 EC 强调“可切换配置”，则判分脚本中可参数化的部分应被显式标识；若某 benchmark 的判分规则不可变，则宜作为 AO 的结构（组成组件）的前提予以声明。

26.3.3 配置域与评价条件空间

显式定义组成组件后，需为每一组件指定*配置域*，即该组件在研究中允许取值的范围。配置域的确定，应在“覆盖实际使用差异”与“保持空间可分析”之间折中：域过窄，难以代表真实使用；域过宽，则全组合规模迅速膨胀，使系统测试不可行。

以常见的知识推理类测试为例，语言选择可取多种自然语言；题型可在选择题与填空题之间切换；实例改写、示例提示、分步推理与多轮交互通常可在启用与禁用之间切换；解码参数则可在若干代表性水平下取值，以覆盖保守生成、常规性生成与高随机性生成等不同使用场景。各组件配置域确定后，一次具体测试对应其中的一个配置向量 c；全部合法配置向量构成 EC 的可容许空间。

因此，组件化分解的最终产物，不是组件清单本身，而是一个*结构化的评价条件空间*。在该空间中，不同的 c 对应 AO 中的不同运行方式，从而对应不同的 $OE(O \mid c)$。只有在这一空间上开展系统采样、对照或分解分析，才能进一步讨论：某一得分差异究竟来自 O 的变化，还是来自某一组件或其联合配置的变化。

26.3.4 与自包含系统的关系

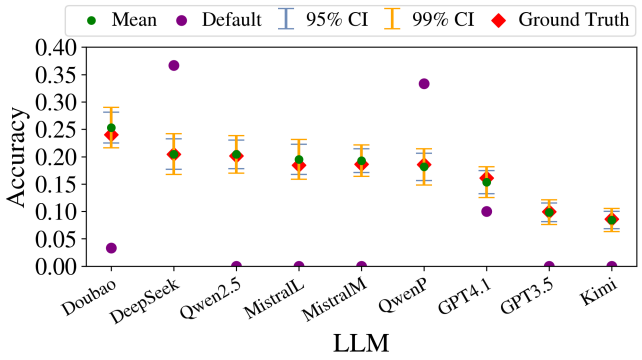
回到 SCS 的定义。评价对象 O 是分析焦点；AO 是产生可测试结果的完整运行系统；EC 则是 SCS 移除 O 后剩余系统中所有可配置的组件集合。组件化分解因此构成连接“抽象评价系统”与“可操作实验设计”的中间层：没有分解，EC 只是未规格化的默认值；完成分解后，EC 才成为可枚举、可比较、可解释的配置空间。这一分解也解释了传统 benchmark 的概念为何不足以代表完整评价系统：benchmark 主要标识任务与实例来源，而交互方式、生成控制及判分解析等组件往往仍留在 AO 中，但未写入公开报告。组件化分解的目的正是将这些隐含决策转化为可研究的显式组件，为后续的系统评价、归因分析与条件化比较奠定基础。

26.4 LLM Evaluatology：大模型评价归因的案例

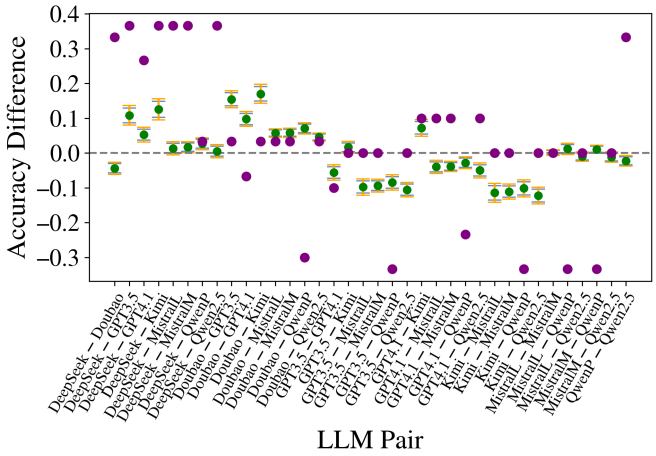
LLM Evaluatology [310] 可以被看作评价学在大模型评价系统中的一个实例化。它并不试图提出一个新的黑盒测试与评价方法，而是将一次完整评价运行所依赖的过程性配置——任务与实例、交互与提示、生成与控制，以及判分与解析规则——在统一的评价条件空间中加以显式定义、组合与抽样，并在因果模型下区分评价对象 O 的真实影响与数据集、提示和解码等混杂因素的影响，从而估计各组成组件的期望影响、交互效应以及不同评价条件下结果的稳定性。其贡献不在于给出“哪一个模型最终获胜”，而在于提供

一套审计大模型内部影响机制的方法：使 AO 上可测试到的分数变化，尽可能归因到可复现、可对照的组件配置上，而非笼统地对应不同的模型上。

我们选取各模型排名前五的重要影响组件，并固定其他组件，以此构建一个真值空间，在这个空间上遍历获得真值，并在此空间上将 LLM Evaluatology 与传统大模型评价方法进行了对比，如图 26.2 所示。实验结果表明，测试结果真值落在 LLM Evaluatology 所给出的置信区间内，而传统方法的估计结果则显著偏离真值，且均落在该置信区间之外。上述结果进一步验证了 LLM Evaluatology 评价方法的有效性与可靠性。



(a) AIME 基准 [4] 下大模型的准确率置信区间。图上的 default 是默认配置下得到的实验结果。在该默认配置下，调用各模型 API 时不显式指定解码相关超参数（temperature、top_p、presence penalty、max tokens），采用各模型预设的默认值；数据集采用原始题目，不考虑语言选择、题型结构、实例改写等组件的影响；不使用 few-shot、CoT 或 multi-turn 等提示或交互格式。



(b) AIME 基准下不同大模型准确率差异的置信区间。

图 26.2: LLM Evaluatology 与传统大模型评价方法的对比 [310]。

该案例揭示了若干有意义的洞察。

- 第一，过程性配置可能比模型名称更决定评价结果。在 AIME [4] 等基准上，题目呈现方式、CoT 提示、生成长度、少样本、多轮交互与作答语言等 EC 组件的配

置，往往比更换 O 更能解释 AO 上准确性的波动；同一 O 在不同合法配置下，准确率可从接近零升至约 0.93。若只比较模型名称或默认脚本，容易把格式与提示的贡献误归因于“模型本身”。

- 第二，默认单点配置不能充当完整归因的必要基线。社区常用的默认设定只是 EC 空间中的一个点：在 AIME [4] 上，162 种系统抽样配置相对默认配置可带来约 70% 的准确率摆动；在 SWE-bench [142]、MMBench [184] 与 Arena-Hard [177] 上，默认配置下的测试结果也常落在更广 EC 抽样所构造的置信区间之外。因此，O 是否优于 O'，应在相同 EC 空间中通过系统抽样检验，而非仅在单点配置比较。

- 第三，组件效应具有条件性与交互性。CoT 提示、少样本或多轮交互并非普遍有效；联合建模下，主效应不显著而交互项显著的情形并不少见 [1]。同一实例的变体变换（如语义无关或逻辑误导的干扰插入）可进一步区分记忆与模型的真实推理能力。评价某一提示或增强策略时，须明确其在何种任务与配置组合下生效，并警惕由于数据重叠所导致的混杂。

- 第四，评价结论的置信度与均值同等重要。在 MMLU [116]、GPQA [238] 等榜单上，不同模型在默认配置下成对比较的结论与基于 EC 配置空间抽样的置信区间比较之间，可能出现排序反转或“虚假差距”。若只报告单点配置下的测试结果或均值，便可能忽略模型性能之间排序结果对于 EC 配置空间的条件敏感性；系统化抽样与基于置信区间的结果报告，旨在给出可归因、可复现、对配置扰动更稳健的结论。

这些发现共同说明：大模型评价领域的科学进步不应以“新模型是否赢得排行榜”为唯一标准，而应以“新组件或新对象是否在明确评价条件空间中对 AO 产生可归因、可复现、稳定的影响”为标准。LLM Evaluatology 所示范的路径——SCS 定义、EC 的组件分解、因果模型与归因分析——为这一标准提供了可操作的分析方法；研究者若希望声称某次能力提升来自模型本身，必须在同一 EC 空间中证明：在控制过程性配置与实例变换之后，O 对 AO 的影响仍成立，而非仅反映某一单点配置下的测试结果。

26.5 总结

大模型评价更应关注预测推理在开放问题上的性能，是评价学可以发挥重要作用的应用场景。传统评价主要围绕单一配置下的 benchmark 得分和模型排行榜展开，容易将 AO 上获得的测试结果差异笼统归因于评价对象 O，事实上，这些测试结果容易受到 EC 组件配置以及测试和评价过程中不同配置的影响，造成归因困难和结论脆弱。评价学提供了一种更系统的视角：将一次完整测试和评价置于自包含系统中，明确评价对象、评价条件、影响对象、混杂对象；将评价条件分解为任务与实例、交互与提示、生成与控制等可配置的组成组件；要求用评价条件配置空间下的测试结果分布取代单点配置下的测试结果；并通过因果建模控制混杂，实现更可靠的组件归因。

LLM Evaluatology 案例表明，大模型评价领域真正需要的不是新的黑盒排名，而是更清晰的评价对象定义、更严格的评价条件控制，以及更可解释的过程性归因。

[1] 由于使用的是 DoE 方法，这里使用的词语是效应，而不是影响

第二十七章

评价学在时间序列预测系统中的应用

本章基于我们已经发表的论文 [325]，由范帆达撰写。

推理学概念定义 17 (时间序列预测（time-series forecasting）)**.** 时间序列预测是根据对象的历史观察推断其未来变化的过程。

作为推理学（见第 七章）的应用，时间序列预测被广泛应用于能源调度、交通管理、金融风险、气象预报和公共卫生等场景。近年来，不断有新的方法被提出，报告的性能越来越好，模型也越来越复杂 [341, 342, 211, 183, 324]。但是，预测误差下降并不意味着新模型的核心组件得到了改进。一个完整模型通常包括输入变换、数据视图、编码器、解码器、输出变换、训练和推理等配置；若只在单一配置上比较均方误差，很难判断改进究竟来自哪个对象，也难以区分稳定能力与偶然出现的最佳结果。本章以 *CombinationTS* [325] 为案例，说明如何将评价学应用于时间序列预测研究，将表面的模型排名上升到机制性的组件影响归因。

27.1 从模型排名到组件归因

问题定义 14 (时间序列预测问题定义)**.** 设长度为 L 的 N 变量时间序列为

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L]^\top \in \mathbb{R}^{L \times N}, \tag{27.1}$$

其中，$\mathbf{z}_t \in \mathbb{R}^N$ 表示时刻 t 对所有变量的测量或测试结果。给定回看（*lookback*）窗口长度 T 和预测长度 P，对于任意满足 $T \le t \le L - P$ 的时间点 t，构造历史输入与未来预测目标分别为

$$\mathbf{X}_t = [\mathbf{z}_{t-T+1}, \dots, \mathbf{z}_t]^\top \in \mathbb{R}^{T \times N}, \tag{27.2}$$

以及

$$\mathbf{Y}_t = [\mathbf{z}_{t+1}, \dots, \mathbf{z}_{t+P}]^\top \in \mathbb{R}^{P \times N}. \tag{27.3}$$

时间序列预测的目标是基于训练样本集 $\mathcal{D} = \{(\mathbf{X}_t, \mathbf{Y}_t)\}_{t \in \mathcal{I}}$，学习参数化预测模型

$$f_{\boldsymbol{\theta}} : \mathbb{R}^{T \times N} \to \mathbb{R}^{P \times N}, \tag{27.4}$$

使其仅利用截至时刻 t 的历史测量或测试结果，同时预测未来 P 个时刻所有 N 个变量的取值：

$$\hat{\mathbf{Y}}_t = f_{\boldsymbol{\theta}}(\mathbf{X}_t). \tag{27.5}$$

模型参数通过最小化预测值与真实未来序列之间的经验损失进行学习，即

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{X}_t, \mathbf{Y}_t) \in \mathcal{D}} \ell(f_{\boldsymbol{\theta}}(\mathbf{X}_t), \mathbf{Y}_t)\,, \tag{27.6}$$

其中，$\mathcal{I}$ 为训练样本对应的时间索引集合，$\ell(\cdot,\cdot)$ 为预测损失函数。

MSE 和 MAE 已在第 七章的预测误差与校准定义中给出。本章沿用该定义，将 MSE 和 MAE 作为时间序列预测系统上的误差型测量量；在多步多变量预测中，可把每个预测位置 (p, n) 视作该定义中的一个预测项 i。MSE 和 MAE 是在预测系统上测得的反映系统总体影响的量，而不是某个预测组件本身的影响。单个损失值只能描述某个模型配置与某个任务配置共同作用后的测量结果，不能自动回答“改变某个组件造成了多大变化”。传统排行榜式评价因而容易出现两个问题。

第一，归因缺口。完整模型之间往往同时存在多项差异，模型排名只能比较整套配置，无法把变化归因于输入变换、数据视图或编码器中的某一个对象。

第二，基准危机。统一固定配置可能没有覆盖不同组件的有效配置空间，形成公平性陷阱；分别搜索并报告每个模型的最好结果，又可能把偶然的超参数匹配当作一般能力，形成最佳结果陷阱。

27.2 时间序列预测评价的自包含系统

根据评价学，时间序列预测组件的评价应首先构造自包含系统（*self-contained system, SCS*），明确评价对象、影响对象、混杂对象和评价条件。

评价对象（EO）。评价对象记为 $o \in O$，可以是完整预测模型，也可以是输入变换、嵌入、编码器、解码器或训练策略中的一个具体实例。例如，研究“Transformer [318] 编码器是否必要”时，某种编码器实例是 EO；研究“周期分解是否改善预测”时，周期输入变换是 EO。

影响对象（*AO*）。直接 AO 是最小的能独立运行的预测任务系统。在时间预测系统这个场景下，AO 包括 EO。

MSE、MAE、预测区间覆盖率、训练开销和推理延迟是在该 AO 上测量或测试得到的量。电网调度系统、库存决策系统等使用预测结果的系统可以作为间接 AO。

混杂对象（*COs*）。除 EO 外仍会影响直接 AO 的对象构成 COs，包括数据集及其分布、互补模型组件、训练算法、软件实现、计算平台和随机数生成机制等。单个混杂对象记为 CO，多个混杂对象统一记为 COs。

评价条件（*EC*）。从 SCS 中移除 EO 后，由直接 AO 和 COs 形成的衍生系统。一个具体 EC 记为 $c \in C$，可以包括数据集、数据划分、回看窗口、预测长度、隐藏维度、学习率、批大小、随机种子、训练预算以及互补组件。

按照全文的数学记号，测量或测试过程记为 m，在 EC c 下评价 EO o 得到测量或者测试结果

$$m(c, o) = r, \qquad r \in R. \tag{27.7}$$

若关注误差型量，可以将其中一个结果分量写为

$$r_\ell = \ell(c, o), \tag{27.8}$$

其中 ℓ 表示 MSE 等损失。公式 27.8 是总体测量结果，仍然混合了 EO 与 EC 的共同作用；只有引入参照对象并控制 EC，才能进一步推断 EO 的影响。

27.3 预测系统的组件分解

CombinationTS 将预测系统的模型分解为五个功能组件：

$$f_o = \mathcal{T}_{\text{out}}^{-1} \circ \mathcal{D} \circ \Phi \circ \mathcal{E} \circ \mathcal{T}_{\text{in}}, \tag{27.9}$$

其中，$\mathcal{T}_{\text{in}}$ 为输入变换 (input transformation)，$\mathcal{E}$ 为嵌入 (embedding) 或数据视图，Φ 为编码器 (encoder)，$\mathcal{D}$ 为解码器 (decoder)，$\mathcal{T}_{\text{out}}^{-1}$ 为输出逆变换 (output transformation)。

输入变换直接作用于原始序列，用于归一化或注入趋势、周期与多尺度等结构先验。例如，RevIN 用于缓解分布偏移 [154]，趋势–季节分解和多尺度下采样用于显式地建模不同的时间结构 [336, 324]。

嵌入的作用是决定如何组织最小的 Token 单元，同时把原始的时间序列空间映射到潜空间。常见的有：Point-wise view [280] 以时间点为基本单元，Patch-wise view [211] 以连续时间片段为 token，Variate-wise view [183] 以变量为 token。

编码器是在潜空间上建模 Token 的高阶依赖关系，一般采用 MLP [304]、Transformer [211, 183]、或不引入可学习参数的 Identity 映射 [336]。

解码器的作用是把潜空间映射回我们实际要预测的序列。其中，单层的线性映射使用得最多。

输出逆变换是输入变换的逆过程。例如如果采用了 RevIN 归一化 [154]，那么模型需要加上变换之前的均值并乘以归一化之前的方差。假如，采用了类似趋势-季节分解或者多尺度分解，那么一般采用简单的加和来合并多个分支。

模型的组件分解的关键不是把模型拆成更多名称，而是允许在不同的粒度上揭示原因对象的影响。例如，将其中一个组件视为 EO，并把其余组件纳入 COs 和 EC。例如，以编码器为 EO 时，嵌入、输入变换、解码器、数据与训练配置都必须在配对比较中保持一致或按照同一规则变化。

27.4 从测量结果到配对影响

设 o 是待评价组件，o_0 是具有相同功能接口的参考组件 (reference component)。在同一 EC c_k 下分别运行二者，可以定义基于误差型指标的配对差异：

$$\Delta_k(o, o_0) = \ell(c_k, o) - \ell(c_k, o_0). \tag{27.10}$$

对于 MSE 等越小越好的指标，$\Delta_k < 0$ 表示 o 相对于 o_0 降低了预测误差。与直接比较两组独立平均值相比，公式 27.10 固定了同一次比较中的 EC，更接近 EO 对 AO 的推断影响。

设目标 EC 分布为 $p(c)$，则参考评价条件下的平均配对影响为

$$\Delta_p(o, o_0) = \mathbb{E}_{c\sim p(C)}[\ell(c, o) - \ell(c, o_0)]. \tag{27.11}$$

在有限样本 $\{c_k\}_{k=1}^K$ 上，可以估计

$$\widehat{\Delta}_p(o, o_0) = \sum_{k=1}^K w_k \Delta_k(o, o_0), \qquad w_k \geq 0, \quad \sum_{k=1}^K w_k = 1. \tag{27.12}$$

均匀权重回答 “在采样条件中平均影响如何”，而由真实部署频率或利益相关方需求确定的 w_k 回答 “在目标使用场景中平均影响如何”。因此，不存在脱离 EC 分布的唯一平均性能。

CombinationTS 同时报告三个描述性统计量：

$$\hat{\mu}(o) = \frac{1}{K}\sum_{k=1}^K \ell(c_k, o), \qquad \hat{\sigma}(o) = \sqrt{\frac{1}{K-1}\sum_{k=1}^K [\ell(c_k, o) - \hat{\mu}(o)]^2}, \tag{27.13}$$

$$L_{\text{best}}(o) = \min_{1\leq k\leq K} \ell(c_k, o). \tag{27.14}$$

$\hat{\mu}$ 描述采样 EC 下的平均性能，$\hat{\sigma}$ 描述对 EC 变化的敏感性，L_{best} 描述已探索条件中的峰值潜力。它们是 EO 在 EC 分布下的结果特征；若要声称某个组件产生了影响，还应报告式 27.10 的差异分布、置信区间或配对差异的统计显著性检验。

分层配对评价协议　CombinationTS 在 Weather、Electricity 和四个 ETT 子集上开展实验，并以 MSE 为主要量。预测长度为 $P \in \{96, 192, 336, 720\}$；实验还改变回看窗口、编码器层数、隐藏维度 d_{hid}、学习率和互补组件，并构造 $K = 600$ 个分层抽样 EC，即每个数据集抽取 100 个条件 [325]。所有 EO 使用同一组 $\{c_k\}_{k=1}^K$，这是评价协议中最重要的控制。若不同组件各自抽取一组 EC，即使样本量相同，组件差异仍可能混入条件采样差异。配对设计既提高比较效率，也使式 27.10 可以直接计算。

27.5　实验结果与评价学解释

以下表格沿用 CombinationTS 的标记方式。蓝色粗体和蓝色粗体下划线分别表示 $\hat{\mu}$ 的最优值和次优值；在报告 L_{best} 的表中，红色粗体和红色粗体下划线分别表示其最优值和次优值；绿色表示来自不同实验协议、不能与配对 EC 估计直接比较的外部结果。

27.5.1　嵌入与编码器

表 27.1 展示了 Patch-wise embedding 在三个代表性数据集上的 $\hat{\mu}$ 均为最低，同时，$\hat{\mu}$ 最优的组件不一定取得最小的 L_{best}：平均能力与偶然峰值可能属于不同 EO。因此，不能用一次最佳结果替代 EC 分布下的总体表现。

数据集	Point-wise		Patch-wise		Variate-wise		Time-to-dim	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
ETTh1	0.4367	0.0364	**0.4277**	0.0370	**0.4285**	0.0327	0.4398	0.0342
ETTm2	0.2663	0.0762	**0.2562**	0.0725	**0.2596**	0.0723	0.2612	0.0728
Electricity	0.1855	0.0252	**0.1773**	0.0246	**0.1827**	0.0249	0.1851	0.0264

表 27.1: Embedding 作为 EO 时的结果。指标为 MSE，越低越好；表中为四个预测长度的汇总结果。其中，Point-wise 是对多通道的单个点进行映射；Patch-wise 是对单通道的 Patch 级的序列进行映射；Variate-wise 是对单个通道的整个通道进行映射；Time-to-dim 是无参数的嵌入，是把整个通道当作嵌入空间。

数据集	MLP		Transformer		Identity	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
ETTh1	**0.4960**	0.0938	0.5095	0.1004	**0.4392**	0.0424
ETTm2	**0.2784**	0.0759	0.2909	0.0772	**0.2621**	0.0756
Electricity	**0.1851**	0.0253	**0.1776**	0.0242	0.1888	0.0275

表 27.2: Encoder 作为 EO 时的结果。指标为 MSE，越低越好；表中为四个预测长度的汇总结果。其中，Transformer 和 MLP 是主流的 encoder 技术路线；Identity 表示空的 Encoder。

表 27.2 展示了 *Identity* 悖论（*Identity Paradox*）：在 ETTh1 和 ETTm2 上，无可学习参数的 Identity encoder （即不需要编码器）具有最低的平均 MSE，且波动不高于复杂编码器；但在 Electricity 上，Transformer 更好。

我们使用单侧 Mann-Whitney U 检验比较采样结果。这里的 Mann-Whitney U 检验是一种基于排序的统计比较方法。我们把 Identity 与参照 encoder 在多个采样 EC 下得到的 MSE 放在一起，从小到大排序，然后判断 Identity 是否更常落在较小误差的一侧。较小的 p 值表示：如果两种 encoder 实际上没有系统差异，那么观察到当前这种排序优势的概率很小。

合并全部数据集时，Identity 相对于 Transformer 的 $p < 0.0001$，相对于 MLP 的 $p = 0.0115$ [325]。这些结果支持“复杂编码器并非普遍必要”，但不能推出“Identity 在所有 EC 或所有数据分布上都更优”。图 27.1 进一步把同一组配对 EC 下的 MSE 分布可视化，显示 Identity 的优势并不只来自单个最佳点。

27.5.2　输入变换的条件效用

输入变换没有脱离 EC 的普遍优劣。Cycle 在 ETTh1 和 ETTh2 上取得最低 $\hat{\mu}$，说明显式周期分解能够在部分周期性数据上降低平均误差；MultiScale 和 TrendSeasonal 在其他数据集上更有优势，但在 Weather 上 Baseline 最好。绿色 TimeMixer 数值虽然更低，却来自不同模型和不同搜索协议，不能作为共享 EC 下组件影响的直接证据。该差异说明：输入分解的价值不仅取决于分解本身，还取决于后续组件能否利用和交互这些信号。表 27.4 用相对 RevIN 的 MSE 变化率展示了同一点：同一类输入先验在不同数据集上

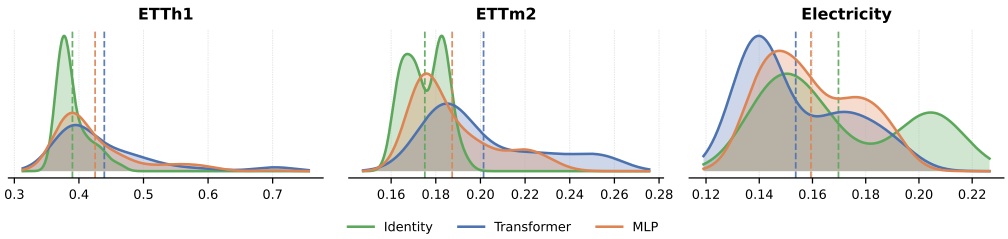


图 27.1: Encoder 作为 EO 时的 MSE 分布。虚线表示各分布的平均水平；越靠左表示 MSE 越低。该图来自 CombinationTS 海报，用于补充表 27.2 中的平均值与稳定性结果。

数据集	Baseline			Cycle			MultiScale			TrendSeasonal			TimeMixer [324]
	$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}	L_{best}
ETTh1	0.3975	0.0315	**0.3749**	**0.3891**	0.0283	**0.3743**	**0.3921**	0.0183	0.3829	0.3947	0.0248	0.3811	**0.3610**
ETTh2	**0.3013**	0.0116	0.2868	**0.2969**	0.0115	**0.2818**	0.3071	0.0343	0.2874	0.3024	0.0135	**0.2849**	**0.2710**
ETTm1	0.3451	0.0206	**0.3139**	0.3431	0.0193	**0.3137**	**0.3380**	0.0171	0.3159	**0.3384**	0.0167	0.3166	**0.2910**
ETTm2	0.1827	0.0037	**0.1738**	**0.1825**	0.0019	**0.1804**	0.1830	0.0011	0.1818	**0.1821**	0.0013	0.1805	**0.1640**
Electricity	0.2061	0.0238	0.1679	0.1999	0.0225	**0.1580**	**0.1952**	0.0202	**0.1580**	**0.1976**	0.0197	0.1729	**0.1290**
Weather	**0.1882**	0.0094	**0.1599**	**0.1917**	0.0017	**0.1899**	0.1950	0.0021	0.1915	0.1934	0.0008	0.1920	**0.1470**

表 27.3: Input Transformation 作为 EO 时的结果。蓝色标记配对协议下的 $\hat{\mu}$，红色标记配对协议下的 L_{best}；绿色 TimeMixer [324] 结果来自其原始论文直接汇报的值。其中，Baseline 是只做了归一化的变换；Cycle 是 Cyclenet [178] 中的变换，它去掉了一个周期的信号；MultiScale 分解采用的是降采样，并分别对多个分支进行建模；TrendSeasonal 分解采用的是 DLinear [335] 中的趋势-季节分解。

的作用方向可能不同。

27.5.3　频域变换的影响边界

SimpleTM [50] 相对于 iTransformer [183] 在三个代表性数据集上均降低了 $\hat{\mu}$，但其 $\hat{\sigma}$ 并未一致下降。因此，更准确的结论是频域处理相对于时域改善了平均性能，而不是普遍提高了稳定性。同时，SimpleTM 在 ETTh1 上仍未超过 Identity，在 ETTm2 上也仅有较小优势；没有 Identity 参照时，“优于 iTransformer” 容易被误读为 “频域编码器不可替代”。图 27.2 将 $\hat{\mu}$ 与 $\hat{\sigma}$ 同时画出，避免只看平均 MSE 而忽略 EC 敏感性。

数据集	周期分解	多尺度分解	趋势–季节分解
ETTh1	+2.1%	+1.4%	+0.7%
ETTh2	+1.5%	-1.9%	-0.4%
ETTm1	+0.6%	+2.1%	+1.9%
ETTm2	+0.1%	-0.2%	+0.3%
Electricity	+3.0%	+5.3%	+4.1%
Weather	-1.9%	-3.6%	-2.8%

表 27.4: 不同输入变换与参考模块 (RevIN) 的 MSE 变化率。负值表示 MSE 降低，正值表示 MSE 升高；符号差异说明输入变换的影响依赖 EC，而不是脱离条件的固定属性。

数据集	iTransformer [183]		SimpleTM [50]		Variate+Identity	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
ETTh1	0.4401	0.0335	**0.4324**	0.0354	**0.4271**	0.0355
ETTm2	0.2870	0.0803	**0.2697**	0.0824	**0.2721**	0.0798
Electricity	**0.1858**	0.0333	**0.1851**	0.0329	0.1974	0.0351

表 27.5: Variate-wise embedding 下不同 encoder 的配对 EC 汇总结果。指标为 MSE，越低越好。

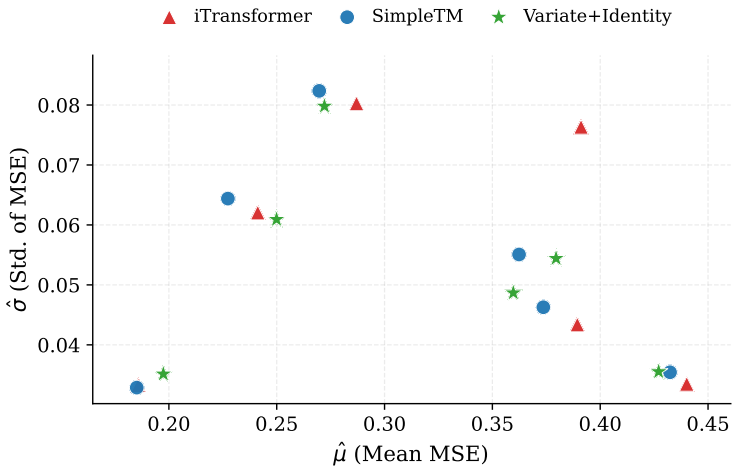


图 27.2: 频域处理、时域注意力与 Identity 参照的 $\hat{\mu}$–$\hat{\sigma}$ 对比。每个点对应一个数据集下的汇总结果；SimpleTM 的平均性能改善并不自动意味着稳定性同步改善。

27.6　组件交互影响

单个组件的影响可能依赖于另一个组件。设 e_0, e_1 为两种 embedding，ϕ_0, ϕ_1 为两种

encoder，在同一 EC c_k 下可以定义二阶配对差异：

$$\Delta_{\text{int},k} = [\ell(c_k, e_1, \phi_1) - \ell(c_k, e_0, \phi_1)] - [\ell(c_k, e_1, \phi_0) - \ell(c_k, e_0, \phi_0)]. \tag{27.15}$$

若 $\Delta_{\text{int},k} \neq 0$，说明 embedding 的变化在两个 encoder 下产生了不同结果，即二者存在交互影响。此时，把完整模型的改进全部归因于 embedding 或 encoder 都不充分。

公式 27.15 也解释了 CombinationTS 中的若干现象：Transformer 对数据视图具有较高敏感性，结构化 embedding 可以改变其工作状态；多尺度与趋势–季节分解是否有效，则取决于后续组件是否提供跨尺度或跨分支交互。因此，评价报告不应只给出组件的主影响，还应至少检查关键组件之间的交互。

27.7　评价结论的适用范围

CombinationTS 的实验能够支持采样 EC 范围内的组件比较，但评价结论仍受到三个边界约束。

第一，结论依赖目标 *EC* 分布。六个公开数据集和选定超参数空间并不等于全部真实部署条件。若目标场景具有不同的非平稳性、采样频率、缺失模式或计算约束，应重新定义 $p(c)$ 与权重 w_k。

第二，低预测误差不等于高应用价值。MSE 改善首先是对直接 AO 的影响证据。它是否改善电网调度成本、库存风险或医疗决策，还需要在相应的间接 AO 上测量或者测试反映影响的量和利益相关方价值函数。

第三，统计差异不自动等于机制解释。配对差异、置信区间和统计显著性检验可以支持影响归因，但组件为何产生该影响仍需结合表示分析、消融实验和对象因果模型进行推理。

27.8　总结

时间序列预测系统的评价不应停留在单个模型、单个配置和单个误差值上。本章以 CombinationTS 为案例，将预测模型分解为输入变换、嵌入、编码器、解码器和输出变换等组件，并在评价学框架中明确了 EO、直接与间接 AO、CO/COs 以及 EC。

本章进一步区分了测量结果与对象影响：$\ell(c, o)$ 是 EO 与 EC 共同作用后在 AO 上取得的误差，而相同 EC 下相对于参照对象的配对差异 $\Delta_k(o, o_0)$ 才是组件推断影响的基本证据。分层配对采样、目标 EC 分布、组件交互影响和简单参照对象共同构成了比排行榜更严格的评价方案。只有当一个组件在明确的目标 EC 分布下相对于参照对象稳定产生改进，并且结论能够经受交互分析和统计检验时，才能认为它对时间序列预测系统产生了可复现且具有适用边界的影响。

第二十八章

科技研究机构评价

本章采用评价学（Evaluatology）对科学技术研究机构进行评价，由范帆达和我共同撰写。

28.1 引言

无论是作为独立实体，还是隶属于高校或企业，科技研究机构（science and technology research institute，STRI）都在科学与技术（science and technology，S&T）进步中发挥着关键的驱动作用。以往针对 STRI 的评价工作往往过于简化，主要将其绩效简化为论文数量、被引次数或其他文献计量指标的简单量化，这属于一种观察性方法。这种做法只能捕捉到科学研究机构所产生总体影响中的一个狭窄的侧面。

在评价学中，科技研究机构被形式化地视为一个评价对象（*EO*）。科技研究机构评价的本质在于揭示该 EO 对一组影响对象（*AOs*）所产生的影响。从这一视角出发，STRI 的作用不应仅局限于学术影响本身，而应涵盖更为广泛的结果维度，影响对象包括人类、国家以及特定的产业等。

此外，这些 AOs 上的可观察结果——无论是通过测量还是测试获得——都不可避免地同时受到 EO 与 混杂对象 （COs）的共同影响。在实践中，对 AO 的影响，诸如一国的战略契合度、某一产业的创新活力，或人类社会的可持续发展指数等，不仅反映了由 EO 直接传递的影响，也包含了由 COs 引入的影响，例如人才资源、国际合作、知识产权保护以及技术—资本环境等因素。因此，评价学要求对 AO 上的每一项测量或测试结果，沿着 EO $\Rightarrow$ AO 与 COs $\Rightarrow$ AO 两条相对简单的路径进行分解，从而确保结果能够准确隔离、识别可归因于 EO 本身的影响。在更复杂的场景下，EO的影响可以通过COs传递给 AO。唯有通过这种精确的归因，自包含系统才能对 STRI 在其 AOs 上所产生的真实影响给出有效且无偏的估计或者推测。

最后，有必要认识到，EO 本身并非一个单一、同质的整体。一个科技研究机构由多个内部组成组件或者机制构成——例如人才培养、评价机制、公共平台、学术合作、管理策略以及技术专长等——每一要素都作为关键组成部分，形成其各自的影响路径。这些内部要素共同作用于 AOs，产生异质性的影响，从而在 SCS 中形成分层的因果链条。因此，严格的评价必须对 EO 内部各组成要素的贡献进行解耦，识别它们通过何种内部影响机制影响不同的 AOs，并进一步分析这些要素之间的相互作用如何放大或削弱 STRI 的总体影响。用于评价 STRI 的自包含系统结构如图 28.1 所示。

本章其余部分将围绕上述视角逐步展开。第 28.2 节首先回顾了历史上主导 STRI 评价的文献计量学方法，并指出其在 SCS 框架下的局限性。第 28.3 节将 AO 从狭义的学术产出扩展至对国家发展、人类进步与产业提升等更广泛的影响。第 28.4 节进一步构建因果逻辑，隔离 EO 产生的真实影响和 COs 产生的影响。最后，第 28.5 节对 EO 进行内部组成组件的分解，并考察其内部影响机制如何共同作用于 AOs，产生可测量或者可测试的影响。综合而言，这些内容共同奠定了对科技研究机构（STRI）进行评价的因果与结构基础。

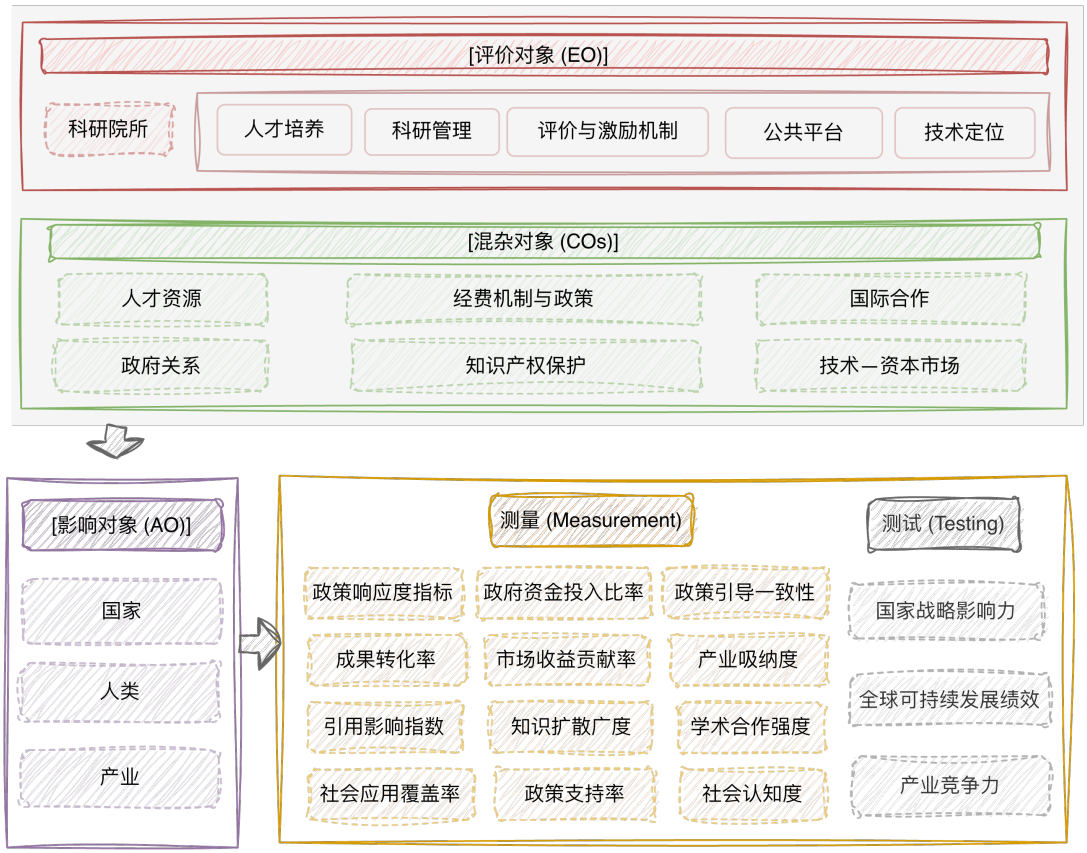


图 28.1: 评价科技研究机构的自包含系统

28.2　学术成果的传统评价方法

在相当长的一段时期内，对科技研究机构（STRI）的评价一直由以学术为导向的评价体系所主导。传统框架几乎将科学与技术（S&T）完全等同于学术成果，主要依赖论文数量、引用影响、期刊排名、高被引论文以及科学共同体内的奖项等文献计量指标。

期刊评价： 学术期刊的质量与影响力通常通过一系列定量指标来衡量，这些指标由不同的数据提供方进行定义并发布。

- *影响因子（Impact Factor, IF）*：作为最为成熟且广泛认可的指标，影响因子由 Clarivate 每年在其 *Journal Citation Reports*（JCR）中发布，其计算公式为：

$$\text{IF}_{\text{年份}Y} = \frac{\text{年份}Y\text{ 对}(Y-1)\text{ 与}(Y-2)\text{ 年发表论文的引用次数}}{(Y-1)\text{ 与}(Y-2)\text{ 年发表的可被引文献总数}}. \tag{28.1}$$

  影响因子反映了期刊论文在发表后两年内所获得的平均引用次数。长期以来，它被视为衡量期刊质量的“金标准”，但也因计算窗口较短、容易受到少数高被引论文的影响，以及不同学科之间可比性不足而受到批评 [94]。

- *CiteScore*：CiteScore 由 Elsevier 基于其 Scopus 数据库提出，是影响因子的主要替代指标之一。与 IF 不同，CiteScore 采用覆盖引用与发文的四年时间窗口，并纳入更广泛的文献类型（如综述、快报等），旨在提供更加全面、透明且稳健的期刊评价指标 [306]。其计算公式为：

$$\text{CiteScore}_Y = \frac{\sum_{i=Y-4}^{Y-1} \text{引用次数}_i}{\sum_{i=Y-4}^{Y-1} \text{发表文献数}_i}. \tag{28.2}$$

- *SCImago 期刊排名（SCImago Journal Rank, SJR）*：SJR 同样基于 Scopus 数据库构建，并引入了类似于 Google PageRank 的算法思想 [218]。该指标不仅考虑引用数量，还考虑引用来源的“质量”，即来自高声望期刊的引用会被赋予更高权重。因此，SJR 更侧重于衡量期刊的学术声望，而非单纯的引用流量 [81]。由于其迭代计算特性，SJR 并不存在一个简单的分数表达式。

- *来源标准化每篇论文影响力（Source Normalized Impact per Paper, SNIP）*：SNIP 指标旨在解决跨学科比较中的不公平问题。它通过以学科领域的“引用潜力”对期刊的原始引用影响进行归一化，从而衡量论文在其所属领域内的相对影响力。当 SNIP 值大于 1.0 时，表示该期刊的引用影响高于其学科领域的平均水平 [176]。其计算公式为：

$$\text{SNIP} = \frac{\text{单位论文原始影响（RIP）}}{\text{相对引用潜力（RCP）}}. \tag{28.3}$$

期刊排名和分区： 除单一指标之外，期刊分区提供了一种更为直观的层级化分类方式，有助于研究者快速判断期刊在其学科领域中的相对地位。

- *JCR 分区（JCR Quartiles）*：该分区体系由 Clarivate 发布，依据期刊在其所属学科类别中的影响因子进行排序，并将排序结果划分为四个等分区间：*Q1*（前 25%）、*Q2*（25%–50%）、*Q3*（50%–75%）以及 *Q4*（后 25%）[58]。

- *中科院分区（CAS Partition）*：中科院分区在中国学术界被广泛采用。该体系基于期刊三年平均影响因子，并采用“金字塔式”的分布模型。在每一学科内部，排名前 5% 的期刊被划分为 一区，6%–20% 为 二区，21%–50% 为 三区，其余期刊归为 四区。其中，一区与二区中最具影响力的期刊还会被进一步标注为“*Top* 期刊”[54]。

会议排名和分区： 在计算机科学等快速发展的领域中，由于审稿周期较短、能够迅速传播前沿研究成果，顶级学术会议往往被认为比许多期刊更具声望。对学术会议的评价通常依赖于基于同行的专家评估，而非单一的定量计算公式。

- *CCF* 国际学术会议推荐列表（*CCF Recommended International Conference List*）：该列表由中国计算机学会（China Computer Federation, CCF）制定，将计算机科学领域的国际学术会议划分为 *A*、*B*、*C* 三个等级。其中，*A* 类代表学术影响力最高的顶级会议。其评价标准具有多维性，综合考虑会议历史、审稿质量、论文录用率以及整体学术影响力等因素。例如，神经信息处理系统大会（*NeurIPS*）被评定为 *A* 类会议 [53]。
- *CORE* 排名（*CORE Ranking*）： CORE 排名由澳大利亚计算机研究与教育协会（Computing Research and Education Association of Australasia）发布，是另一套在国际上广泛认可的计算机科学会议评价体系。该体系将学术会议划分为四个等级：*A**（旗舰级）、*A*（优秀）、*B*（良好）和 *C*（标准）。在该体系中，*NeurIPS* 被评定为 *A** 级会议 [63]。

综上所述，传统的学术成果评价在很大程度上依赖于一套成熟但受限的文献计量指标与排名体系。在这一范式下，被评价对象（EO）往往通过一组狭义界定的学术产出进行隐含评判，而其对国家发展、人类进步与产业提升等更广泛层面的影响则基本未被系统考察。因此，这类以学术为中心的方法所捕捉到的，更多只是科研活动的表层表现，而非研究机构对其受影响对象所作出的完整因果贡献。

28.3 超越学术影响：面向科技研究机构的自包含系统框架

SCS 构成了评价学的核心评价模型。它形式化地刻画了在特定求索条件下，评价对象（EO）如何在考虑混杂对象（COs）影响的同时，对一组影响对象（AOs）产生可测量或可测试的影响。

在这一框架中，科技研究机构（STRI）的各类活动通过其影响路径加以解释，使得 AO 上的每一项可测量或可测试结果，都能够追溯并归因于 EO、COs 或二者之间的相互作用。因此，SCS 通过将 EO、COs 和 AO 三个组成部分整合到一个统一的结构中，对 STRI 的因果架构进行建模，从而以概念和方法上的一致性将机构能力、环境背景和社会影响联系起来。

位于 SCS 顶层的是评价对象（EO），它代表了直接从事创新与知识创造活动的研究实体。每一个 EO 由多个内部组成要素构成——包括人才培养、科研管理、评价与激励机制、公共平台（数据/基础设施）以及技术定位——这些要素共同构成了其内部组成结构及影响机制。EO 的这些内部要素共同塑造了组织在科学与技术知识的生成、转化与传播方面的内在能力，并决定了 EO 在 SCS 中通过何种方式、以何种路径最终对其影响对象（AOs）产生影响。

位于 EO 和 AO 之外的是混杂对象，它们代表在 SCS 中对 EO 产生影响、约束或放大作用的对象。典型的 COs 包括人才资源、经费机制与政策、国际合作、政府关系、知识产权保护以及技术—资本市场等。EO–COs 的交互刻画了政策激励、资源流动与知识交换所作用的动态边界，并由此塑造了 EO 的内部能力如何被转化为在 AOs 上可测量、可测试的具体影响。

这些影响最终体现在被影响对象之中，AOs 构成了承载并显现科学与技术活动影响的具体领域。需要强调的是，AOs 不应被限定为论文或引用等学术产出。在 STRI 的语境下，AOs 涵盖了以下三个更高层级的社会影响领域：

- **国家发展（*National Development*）**——国家竞争力、战略安全以及政策契合度；
- **人类福祉（*Human Progress*）**——知识积累、社会福祉、可持续性、公平性与全球福祉；
- **产业提升（*Industrial Advancement*）**——技术升级、生产率提升以及经济结构转型。

这些领域共同反映了科学与技术（S&T）在整个人类文明层面所能产生的不同影响。因此，对 S&T 的全面评价不仅应量化学术成就本身，还必须刻画并衡量在 SCS 中通过 EO ⇒ AO 与 COs ⇒ AO 因果路径所产生的多领域影响。

为了在评价学框架下将 SCS 具体化并应用于科技研究机构的评价，评价者依赖于两类基本的求索方式：测量（*measurement*）与测试（*testing*）。二者共同作用，对 AOs 上的可观察影响进行赋值，并检验关于 EO 影响的命题或模型是否符合检验基准（*test oracle*）。

测量在特定的 EC 条件下，对沿着 EO ⇒ AO 与 COs ⇒ AO 路径所产生的可观察影响进行赋值。每一项指标都对应于这些因果关系的一种可测量体现。例如，科研成果转化率与技术商业化率刻画了由 EO 产生的科学与技术如何向产业与经济层面的 AOs 传播；国际科研合作强度与国际合作论文占比等指标则用于捕捉跨国知识流动，反映 EO 对全球科学交流的贡献。面向政策的指标——包括产业政策敏感度与国家战略契合度——衡量 EO 活动对国家层面 AOs 的影响；而人类发展指数（*Human Development Index, HDI*）则将测量扩展至人文 AOs，通过把科学与技术进步与人类福祉的改善相联系来加以刻画。因此，测量依托于来自行政记录、科研产出数据库、合作网络、政策文件以及社会经济统计等可观察数据，并将其转化为具有可比性的数值量。

测试是一种验证过程，其通过运行测试用例来判断关于 EO 对其 AOs 产生影响的命题或模型是否符合既定的检验基准。在 STRI 的语境下，检验基准规定了在给定 SCS 中科学与技术活动所应达到的强制性或期望性结果，例如国家战略影响力的目标水平、全球可持续发展绩效的基准，或产业竞争力所需达到的阈值。测试用例是一种预先定义的求索条件，由选定的 EO 内部要素、COs、AOs、时间窗口以及数据样本共同构成，在该条件下对测量指标进行计算。测试过程通过执行这些测试用例，并将实际测量得到的结果与对应检验基准所规定的结果进行比较，从而给出通过或未通过的判定，或对诸如“STRI 与国家战略保持一致”或“STRI 显著促进全球可持续发展”等命题作出接受或拒绝的结论。

通过反复迭代的循环过程——其中测量和测试提供定量化输入，同时测试对明确定义的检验基准进行验证——该方法论确保评价结果既具有坚实的数值基础，又在逻辑上保持一致性。

28.4 精准归因：识别 EO 的真实影响

在 SCS 中，评价的因果结构本质上是相互联结的。系统中的各类对象——包括科技研究机构、期刊、会议、公共平台——随着时间变化持续发生交互，并在不同对象之间生成相

互重叠的影响。在这一自包含系统中，诸如政策激励、经费项目、合作机会以及平台可见性等混杂对象并非静态的背景条件，而是能够主动对影响对象产生影响、并调制 EO 影响的活跃对象。因此，任何在 AO 上观察到的结果都体现为一种总体影响，其中既包含 EO 产生的真实影响，也包含 COs 所引入的影响，以及二者交互作用所产生的影响。

精准归因的目标在于隔离由 *EO 产生的真实影响*，即对这些交织在一起的因果来源进行解耦。从经验上看，评价者只能直接观测到 AO 上的结果。设 $\widehat{Y}_{\mathrm{AO}}$ 表示在某一 AO 上测得的影响，该影响汇聚了多条因果路径的共同贡献：

$$\widehat{Y}_{\mathrm{AO}} = f_{\mathrm{EO}}\big(\mathrm{EO} \Rightarrow \mathrm{AO}\big) + f_{\mathrm{COs}}\big(\mathrm{CO} \Rightarrow \mathrm{AO}\big) + f_{\mathrm{int}}\big(\mathrm{EO}, \mathrm{CO} \Rightarrow \mathrm{AO}\big) + \varepsilon, \tag{28.4}$$

其中，f_{EO} 表示 EO 对 AO 产生的真实影响，f_{COs} 表示由 COs 产生的影响，f_{int} 刻画了 COs 通过界面（影响机制）对 EO 作用于 AO 的影响进行放大、削弱或重塑的过程，而 ε 表示噪声项。由于评价者只能观测到 $\widehat{Y}_{\mathrm{AO}}$，要识别由 EO 产生的真实影响，必须在明确定义的求索条件下，通过因果重构（*causal reconstruction*）过程来加以实现 [258]。

为将 EO 和 COs 的影响加以区分，EO 的真实影响可以表示为：

$$\begin{aligned} \text{EO 的真实影响} = \mathbb{E}\Big[\widehat{Y}_{\mathrm{AO}} \mid \mathrm{EO} = 1,\, \mathrm{COs} = \text{constant}\Big] \\ - \mathbb{E}\Big[\widehat{Y}_{\mathrm{AO}} \mid \mathrm{EO} = 0,\, \mathrm{COs} = \text{constant}\Big], \end{aligned} \tag{28.5}$$

该表达式是对 COs 进行控制（conditioning on COs）。从概念上看，这对应于在相同条件下进行的一种反事实比较：*如果 EO 的影响不存在，AO 的结果将会呈现为何种状态？*

在固定条件下，可以采用多种方法论途径来执行上述因果重构。其中一类方法依赖于影响分解（*effect decomposition*）[8]，将 AO 上的测量或者测试结果划分为可归因于 EO、COs 及其交互作用产生的影响等不同组成部分。

另一类方法采用受控比较策略（*controlled comparison strategies*）[102]，在相同的 COs 条件下，对不同的 EO 进行比较，从而消除评价条件差异所带来的影响。此外，还可以应用结构重构技术（*structural reconstruction techniques*）[259]，例如基于约束或基于评分的因果路径重构方法，从观察数据中推断 EO ⇒ AO 关系的结构形态。最后，测试为验证所推断影响提供了一种机制，用以检验其是否符合既定的检验基准。这些方法共同作用，确保所推断的 EO 影响反映的是 EO 的真实影响，而非由 COs 所引入的扰动。

在评价学框架下，精准归因将评价从单纯的描述性比较提升为一种因果问责（*causal accountability*）形式。评价者不再仅仅对不同 STRI 的观察绩效进行对比，而是通过追溯影响路径，考察绩效产生的原因，并识别 COs 如何对 EO 影响进行调制、放大或混杂。例如，某一 STRI 较高的论文产出规模或突出的技术转化表现，既可能源于其真实的内部能力，也可能主要受益于有利的 COs，如充足的经费支持、优势合作关系或特定的时间条件。若无法对这些来源加以区分，评价结果便容易将外部条件优势与 EO 的内在能力混为一谈。

最终，精准归因将 STRI 的科学与技术进步重塑为一种经评价条件调整后的因果影响，从而在共享的评价条件中揭示 EO 的真实影响。这一原则为下一步工作奠定了基础：*追踪 EO 的内部组成要素：结构及机制*，以阐明其内部影响机制如何协同作用，从而在 AOs 上产生可测量或测试的总体影响。

28.5　追踪内部机制：EO 内部组件层面的归因分析

在将 EO 的真实影响与 COs 的影响加以区分之后，还需要进一步的分析步骤，以理解该影响是在 EO 内部是如何生成的。EO 并非一个单一、同质的对象，而是一个由多个相互依赖的内部组成要素所构成的结构化系统，这些要素共同决定了其科学与技术（S&T）进步的形成过程。

这些组成要素——例如人才培养、评价与激励机制、公共服务平台、管理机制与策略以及技术定位——作为内部影响机制发挥作用。它们之间的协同交互最终塑造了 EO 在 SCS 中对各类 AOs 所产生的可测量、可测试的具体影响。

设 $\mathbf{c}_{\mathrm{EO}} = \{c_1, c_2, \ldots, c_n\}$ 表示 EO 的内部组成要素集合。每一个组成要素 c_i 都会以单独或协同的方式，对 AO 上所观测到的结果 $\widehat{Y}_{\mathrm{AO}}$ 产生贡献。因此，由 EO 产生的总体影响可以表示为：

$$f_{\mathrm{EO}}(\mathbf{c}_{\mathrm{EO}} \Rightarrow \mathrm{AO}) = \sum_i g_i(c_i \Rightarrow \mathrm{AO}) + \sum_{i<j} g_{ij}(c_i, c_j \Rightarrow \mathrm{AO}) + \varepsilon_{\mathrm{EO}}, \tag{28.6}$$

其中，f_{EO} 表示 EO 对 AO 产生的真实影响，g_i 表示组成要素 c_i 的直接影响，g_{ij} 刻画不同组成要素之间交互作用所产生的高阶影响，而 $\varepsilon_{\mathrm{EO}}$ 用于表征那些无法被直接观测到的残余内部影响。

为了在给定的 CO 配置下评估各内部组成要素的边际贡献，我们考察 AO 结果对 c_i 的敏感性：

$$\frac{\partial \widehat{Y}_{\mathrm{AO}}}{\partial c_i} = \underbrace{\frac{\partial f_{\mathrm{EO}}}{\partial c_i}}_{\text{EO 内部组件的直接影响}} + \underbrace{\sum_k \frac{\partial f_{\mathrm{int}}}{\partial c_i}\frac{\partial x_k}{\partial c_i}}_{\text{CO 影响}}. \tag{28.7}$$

其中，$\widehat{Y}_{\mathrm{AO}}$ 表示在 AO 上测得的影响。f_{int} 表示 EO 内部组成要素与 COs 之间的交互作用函数，x_k 表示第 k 个 CO 变量（如经费水平、合作机会或政策激励等）。第一项刻画了该内部组成要素本身所产生的真实贡献；第二项则反映了 COs 对该要素效应的调制作用——例如经费水平、合作机会或政策激励如何放大或削弱某一具体机制的贡献。这种基于微分的表述为组件层面的归因分析提供了定量基础，有助于澄清各内部机制如何共同塑造 EO 在 AOs 上所产生的整体影响。

从评价学的视角来看，这一分析构成了一种机制归因（*mechanistic attribution*）。它将评价问题从“该 EO 产生了多少真实影响?”转变为“哪些内部机制产生了这些影响，以及它们是通过何种相互作用实现的?”。这种洞察使得对 STRI 的改进能够更具针对性，为循证政策设计提供依据，并在 COs 异质性的条件下，实现对不同 EO 之间更加公平的比较。

最终，组件层面的归因揭示了 STRI 的影响并非源自孤立的单一结构或者机制，而是由多种内部结构或机制之间的协同配合下产生的——每一种结构或机制都在 SCS 中留下可测量的因果影响，并共同决定了 EO 在 AOs 上所呈现的可测量或可测试影响。

28.6　本章小结

本章表明，对科技研究机构（STRI）的评价需要从以文献计量为主的观察性方法转向一种以因果为基础的分析框架，用以揭示 STRI 及其内部组成要素如何在不同的影响对象

产生可测量或可测试的总体影响。

第二十九章

实验床原理、方法论与案例研究

本章系统化地阐述了何谓实验床（testbed），并介绍实验床的基本原理、方法论以及一个典型案例研究。本章由高婉铃和我共同完成。

29.1 什么是实验床？

实验床——无论是作为实验平台、仿真环境，还是完整的模拟系统——都是工程领域中评价设计选择与实现权衡不可或缺的工具。

然而，实验床本身并不存在统一、严格的形式化定义。

评价学概念定义 17（实验床）. 实验床是一种为某一类或多类原因对象或者评价对象（*UO* 或 *EO*）设计和实现的评价模型或评价系统，用以揭示原因对象的影响，并探索原因对象的设计空间。

29.2 实验床原理

实验床的根本目的在于通过可控、可重复且可解释的实验，使 *EO* 对其对应的 *AO* 所产生的因果影响得以测量或测试。

理想情况下，实验床应当设计或者实现自包含系统或者对象因果模型。根据其实现方式，实验床可分为两类：

- 基于 *SCS* 的实验床：提供可运行、可隔离、可重复的实验环境，通过测量或测试获得评价结果；
- 基于 *OCM* 的实验床：提供数字化、可推演、可检验的对象影响模型，主要通过建模和计算获得评价结果。

无论基于 SCS 还是 OCM，一个完美实验床都应当能够在不同 COs 条件下识别、隔离并推断 EO 对 AO 的影响。

理想状态下，它应满足四个特征：能够正确识别 AO 与 COs；能够完全隔离无关对象；能够推断 EO 的真实影响；并且可以自由操控评价模型的完整状态空间。然而，受限于现实条件，在绝大多数情况下我们只能构建不完美的 SCS。因此，实验床首先应当遵

循受控真实（*controlled realism*）这一核心原则：即在复现 SCS 功能性与因果关系、或保持 OCM 机制结构有效性的同时，为研究者提供足够的控制能力、可实验能力或者模型推演能力，从而实现对 EO 影响的准确推断。

任何实验床的设计均应遵循以下三项基本原则：

（1）代表性（Representativeness）： 实验床必须以足够的精准性构建近似一个完美或不完美 SCS 或者 OCM，使其所得结果在真实系统中仍然具有外部有效性或者泛化能力。代表性确保 EO、AO 与 COs 之间的关键关系得以忠实保留，即便模型经过简化亦然。例如，即使缺乏真实物理电路，只要硬件模拟器能够保持时序与依赖关系特征，其评价结论仍具有一定的有效性。

（2）可控性（Controllability）： 实验床应允许对 EO、AO 与 COs 的配置进行显式操控。对于基于 SCS 的实验床，这种受控实验能力使不完美的实验环境转化为可分析的评价模型；对于基于 OCM 的实验床，这种受控建模能力使研究者能够显式设定对象状态、影响路径与机制参数，并对不同配置进行推演与比较。在理想场景下（完美 SCS 或 OCM），所有无关对象的影响均可被消除；而在实践中，实验床只能尽可能逼近这一状态。

（3）透明性与可再现性（Transparency and Reproducibility）： 实验床必须支持对其内部状态的完全可见性，并允许实验或模型推演以确定性或统计上有界的结果进行重复。对于 SCS，这意味着实验条件、对象配置与测量、测试过程应当可追溯；对于 OCM，这意味着对象结构、影响关系和模型假设应当可解释、可审查。透明性保证结果的可解释性——研究者能够将结果追溯至其潜在原因；而可再现性则确保实验结论能够被独立验证。

总体而言，实验床将评价学（Evaluatology）的核心目标具体化：构建一个可测量和测试、可操控、可推断的环境，使研究能够从单纯观察过渡到基于实验的因果理解。无论是基于 SCS 还是基于 OCM，实验床始终是在技术与资源约束下对这一目标的具体实现。

29.3　实验床的基本方法学

在上述原理基础之上，实验床方法学定义了评价者如何构建、运行与改进实验床，以在现实约束条件下实现可靠的因果推断。实验床的方法论基础，建立在其与两类评价模型——完美评价模型、不完美评价模型之间的对应关系之上。这两类评价模型体现了准确性与可行性之间的不同权衡，并分别可通过 SCS 或 OCM 两种方式实现。

（1）完美实验床： 完美实验床代表一种理论上的理想情形。在该环境中，所有无关对象均被完全隔离，所有相关交互均被显式建模，因果结构完全透明，研究者可以推断 EO 的真实影响。

- 基于 *SCS* 的实现：构建一个完全可控、完全隔离、完全可重复的实验环境。例如，在完全确定性的算法基准评价中，所有输入、随机种子与计算状态均可被固定与操控，从而实现完全可重复的评价。
- 基于 *OCM* 的实现：构建一个机制完整、结构完备、可完全推演的对象因果模型。例如，在理想化的药代动力学模型中，药物、身体各器官与生化反应路径均被完整地建模形式化，从而在系统能实际测量或测试之前可推演 EO 对 AO 的影响。

然而，由于其极高的复杂性与抽象成本，完美实验床在现实中往往难以实现。

（2）不完美实验床： 不完美实验床旨在逼近真实系统，但不可避免地包含部分无法完全控制、隔离或表达的因素。换言之，尽管实验床试图刻画 EO 与 AO 之间的因果关系，一些来自周围环境或未观察到的 COs 的影响可能仍然存在。

- 基于 *SCS* 的实现：在受控但不完全隔离的环境中运行实验。例如，在不同环境温度下评价 CPU 性能时，由于热影响的存在，性能结果可能发生变化。这种不完美隔离使评价结果更具现实代表性，但也为因果推断带来不确定性。
- 基于 *OCM* 的实现：建立部分机制已知、部分机制近似的对象因果模型。例如，在药物疗效评价中，模型能够刻画主要代谢通路，但尚未完全覆盖个体差异、合并用药或长期副作用等复杂机制。此时，模型仍可用于推断 EO 的影响，但其外部有效性有待检验。

因此，不完美实验床在因果严谨性与现实有效性之间实现了一种务实的平衡。

（3）评价流程： 在所有实验床类型下，实验床设计与实施的一般流程均包含以下六个规范阶段：

1. 对象定义：定义被评价对象 EO、影响对象 AO，以及混杂对象集合 COs；
2. 评价模型构建：在实验床中实现 SCS 或 OCM，形式化 EO、AO 与 COs 的相互关系，定义对象的属性、机制等信息；
3. 条件采样：生成一个具有代表性的 EC 子集 C'，覆盖 COs 与 AO 参数的关键变化；
4. 结果测量或推演：在受控条件下执行实验，或在 OCM 上进行推演，获得结果分布 $oe(o|c')$（其形式化定义见第 8.4.1 节），并通过重复实验或重复推演处理随机性；
5. 影响推断：结合 SCS 或 OCM，采用统计分析方法（如方差分析、回归分析或协方差分解）估计 EO 对 AO 的推断影响；
6. 假设检验：对 EO 对 AO 的推断影响进行统计假设检验。

该结构化方法学构成了一个统一框架：完美实验床保证理论有效性，不完美实验床提供现实真实性。二者共同使得评价学能够适用于科学研究与工程实践；而 SCS 与 OCM 则分别提供了物理实验与机制建模两种实现路径。

29.4 案例研究

为展示实验床原理在实践中的应用，下面选取来自不同评价领域的代表性案例，说明不同基于 SCS 的实验床方法论如何实现。

案例一：CPU 性能评价： 在硬件性能评价基准（benchmark） 中，*EO* 为 CPU，*AO* 为计算系统（包括操作系统、内存与磁盘），而 *COs* 则由工作负载、数据集、线程数等构成。

不同类型的实验床体现了评价学在严谨性与可行性之间的权衡。

完美实验床意味着在完全隔离并穷尽 AO 与 COs 空间的条件下评价 CPU——这是一个在实践中难以实现的理想状态。

不完美实验床则通过在有限（limited）AO 上运行标准化评价基准程序（如 SPEC CPU [284]），在受控但不完全隔离的条件下评价 CPU 性能。

案例二：药物疗效评价： 在生物医学评价中，*EO* 为药物化合物，*AO* 为人体，而 *COs* 包括饮食、压力与环境暴露等因素。完美实验床对应于一种完全可控的理论生理模型，在该模型中 AO 与 COs 均可被精确操控，从而彻底隔离药物的生化影响——这一情形在现实中并不存在。

临床试验可被视为非完美实验床，其中随机化与盲法被用作构造近似等价评价条件的手段。

讨论： 跨领域案例共同揭示了一个反复出现的权衡关系：评价成本随准确性的提升而显著增加。因此，实验床设计的实践艺术在于保持关键因果结构的同时，使评价在操作层面上可行。此类实验床正是评价学基本愿景的具体体现：将抽象的因果推断转化为可复现且以证据为基础的评价实践，从而弥合理论与实践之间的鸿沟。

29.5 本章小结

本章在评价学框架下建立了实验床的统一理论与方法学基础。