北京大学

本科生毕业论文

题目：迈向智能体的物理理论

Toward a Physical Theory of Agents

姓　　名：宋卓洋

学　　号：**********

院　　系：物理学院

专　　业：物理学

导师姓名：朱华星

二〇二六 年 五 月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则一旦引起有碍作者著作权之问题，将可能承担法律责任。

摘要

大语言模型智能体已取得丰富的工程成果，但仍缺乏统一的理论框架来指导设计。微观层面的理论研究虽然活跃，却尚未与智能体的宏观行为建立可追溯的联系，智能体的能力来源与约束仍缺乏原理层面的解释。本文回顾从神经网络基础到智能体应用的完整技术图景，揭示设计中的跨领域共性，并提出一套以物理概念为基础的理论研究纲领。该纲领将智能体架构形式化为算子图，以概率分布作为理论分析的基本对象，通过将分布投影到不同的宏观表示上导出可验证的定量约束。这一方法不仅能导出定量规律，还使得从少量转移核测量出发通过模拟替代高成本的端到端试错成为可能，为智能体设计提供原理层面的指导，也为未来理论发展指出了方向。

关键词：大语言模型，智能体，算子图，投影，细致平衡

Toward a Physical Theory of Agents

Zhuo-Yang Song (Physics)
Supervisor: Professor Hua Xing Zhu

ABSTRACT

Large language model (LLM) agents have produced impressive engineering outcomes, but a unified theory to guide their design remains absent. Although microscopic research on LLM training and generation is active, it has yet to connect with the macroscopic behavior of agents, leaving the origins and limits of agent capabilities without principled explanation. This thesis reviews the full technical landscape from neural-network foundations to agent applications, identifies cross-domain design commonalities, and puts forward a physics-inspired theoretical programme. The programme casts agent architectures as operator graphs whose edges carry probability distributions and derives testable quantitative constraints by projecting these distributions onto different macroscopic representations. Beyond producing quantitative laws, the framework shows that a small set of transition-kernel measurements suffices for simulation to replace expensive end-to-end experimentation, offering principled design guidance and charting a path for future theoretical development.

KEY WORDS: large language model, agent, operator graph, projection, detailed balance

目 录

第一章　引言

大语言模型（Large Language Model, LLM）与工具调用、记忆管理和评估反馈的结合，催生了被称为智能体（Agent）的新型系统。这类系统将 LLM 的概率性生成嵌入搜索和决策循环，使其从孤立的文本补全转变为反复验证和筛选的算子[1-4]。近年来，智能体在科学发现[5-9]、工程优化[10-12]和开放世界交互[13-14]等领域取得了显著进展，其能力边界由 LLM 生成特性与系统架构的耦合共同决定[15-17]。

对智能体的理解目前停留在工程层面。不同系统的架构差异显著：有的依赖形式化验证逐步检查推理的每一步，有的放弃过程验证仅靠评估函数驱动搜索，还有的直接与物理实验设备交互。这些系统的共性仅被经验感知，未被定量刻画。对 LLM 的微观理论研究（注意力机制、训练动力学、词元统计性质）虽然活跃[18-20]，与智能体层面的宏观行为之间却缺乏可追溯的连接。智能体的能力从何而来、受何约束、如何扩展，仍无原理层面的解释。

不同架构的智能体之间是否存在跨领域的定量规律，以及这些规律能否为设计提供超越反复试错的指导，是本文讨论的核心问题。提出这一问题，是为了从实践中提炼原理。这种从宏观可测量量出发建立定量约束的方法，在科学史上有先例可循[21]。

本文的目标是建立一套以物理概念为基础的理论研究纲领。现阶段，智能体实践积累了大量经验但缺少统一的形式化语言，理论活跃于微观层面但未连接到宏观行为；与此同时，LLM 完整输出所诱导的词元序列分布已经可以通过采样、评测和转移核测量进入经验研究，成为连接两端的可操作对象。因此，提出统一纲领并非单纯增加一种解释框架，而是为了使机制解释、智能体工程、评测方法和训练与规模定律研究能够在同一个概率对象上比较、积累和相互约束。为此，本文首先回顾从神经网络到智能体应用的技术背景，考察代表性系统的设计范式与能力表现，提炼跨领域共性。在此基础上，本文引入算子图作为统一的形式化语言，以 LLM 生成的词元序列上的概率分布（即系综）作为理论分析的基本对象，对系综的测量构成连接微观动力学与宏观表现的不可约实验，将分布投影到不同的宏观表示上以提取可测量量，通过线性表示和势能表示两种投影方式从中提取宏观约束和设计原则[22-23]。这些约束在多模型、多任务实验中获得了验证，并通过基于转移核测量的模拟展示了从少量实验数据出发替代高成本端到端试错、系统优化设计参数的能力。这为一个初步但有清晰结构的物理理论框架奠定了基础，并为未来的研究指明了方向：包括表示的拓展与统一、理论的定理化证明，以及训练动力学的引入等。

本文的组织如下。第二章回顾从神经网络、大语言模型到智能体架构的技术背景，

第三章考察代表性智能体系统的设计范式与能力表现，提炼跨领域的共性趋势。在此基础上，第四章发展以物理概念为基础的理论框架：引入算子图作为统一形式化语言，以概率分布（系综）为基本对象，将其投影到线性表示与势能表示两种宏观表示上，分别导出响应率约束与细致平衡条件等定量规律，并通过基于不可约实验的模拟展示设计参数优化的完整工作流。第五章讨论当前框架的开放问题，第六章总结全文并展望未来方向。

第二章　技术背景

本章回顾从神经网络到智能体的技术背景，为后续的理论分析提供必要的概念基础。第 2.1节介绍神经网络与深度学习的基本原理，包括前馈网络、物理对称性在网络设计中的作用以及优化动力学；第 2.2节讨论大语言模型的核心架构、训练范式和理论前沿，重点关注其概率性生成的本质；第 2.3节梳理智能体的形式化定义、关键技术与架构设计模式。三节由底层计算基础逐步过渡到智能体系统，共同构成第三章考察具体智能体实现和第四章发展理论框架的出发点。

2.1　神经网络与深度学习基础

理解智能体的能力约束，需要从其计算基础开始。神经网络通过多层非线性变换实现了对复杂数据的高效表示和处理，为智能体提供了感知、决策和适应的底层能力。尽管 LLM 在现代智能体设计中占据主要地位，最先进的端到端智能体中，小型化、专用化的神经网络架构因其高效性和可解释性仍然是重要组成部分。作为技术图景的第一层，本节从网络架构、优化动力学及其在智能体中的应用三个方面介绍神经网络与深度学习的基础原理。

前馈神经网络与反向传播

深度学习中最基础的网络结构是多层感知机（Multilayer Perceptron, MLP）：一连串线性变换与逐元素非线性函数交替堆叠，把输入逐层映射到目标空间。一个含 $(n-1)$ 个隐藏层的 MLP 写成 (2.1)的形式：

$$f(x) = W_n\sigma(W_{n-1}\sigma(\cdots\sigma(W_1x + b_1)\cdots) + b_{n-1}) + b_n, \tag{2.1}$$

其中 W_i、b_i 是第 i 层的权重和偏置，σ 是预先指定的逐元素非线性函数。把激活函数本身改为可学习的，就得到 KAN[24-25]，它在小规模科学回归任务上往往比固定激活的 MLP 更易解释。

通用逼近定理（Universal Approximation Theorem）表明，具有至少一个隐藏层和非线性激活函数的多层感知机可以任意精度逼近任何定义在紧致集上的连续函数[26]（图 2.1）。这为神经网络作为函数逼近器提供了理论保障，然而深层网络的训练往往面临梯度消失和梯度爆炸问题。1986 年，Rumelhart 等人提出的反向传播算法（Backpropagation）高效地计算损失函数对网络参数的梯度，显著推动了深度学习领域的发展[27]。

这种算法通过链式法则将误差从输出层逐层传播回输入层，使得网络能够通过梯度下降等优化方法进行训练。具体地说，如 (2.2)所示：

$$\frac{\partial \mathcal{L}}{\partial W_i} = \delta_i \cdot a_{i-1}^T, \quad \delta_i = (W_{i+1}^T \delta_{i+1}) \odot \sigma'(z_i), \tag{2.2}$$

$\mathcal{L}$ 为损失函数，a_{i-1} 是第 $i-1$ 层的激活输出，$\sigma'(z_i)$ 是激活函数的导数，$\odot$ 是逐元素乘法。δ_i 是第 i 层的误差项，定义为

$$\delta_i = \frac{\partial \mathcal{L}}{\partial z_i}, \tag{2.3}$$

z_i 是第 i 层线性变换的输入。整个反向过程只需对参数遍历一次，复杂度与参数量同阶，这是深层网络得以在现代硬件上训练的直接原因。

激活函数的选择对于网络的非线性表达能力至关重要；其中，ReLU（Rectified Linear Unit）[28]及其变体因缓解梯度消失问题和计算开销低而成为现代深度网络的主流选择。

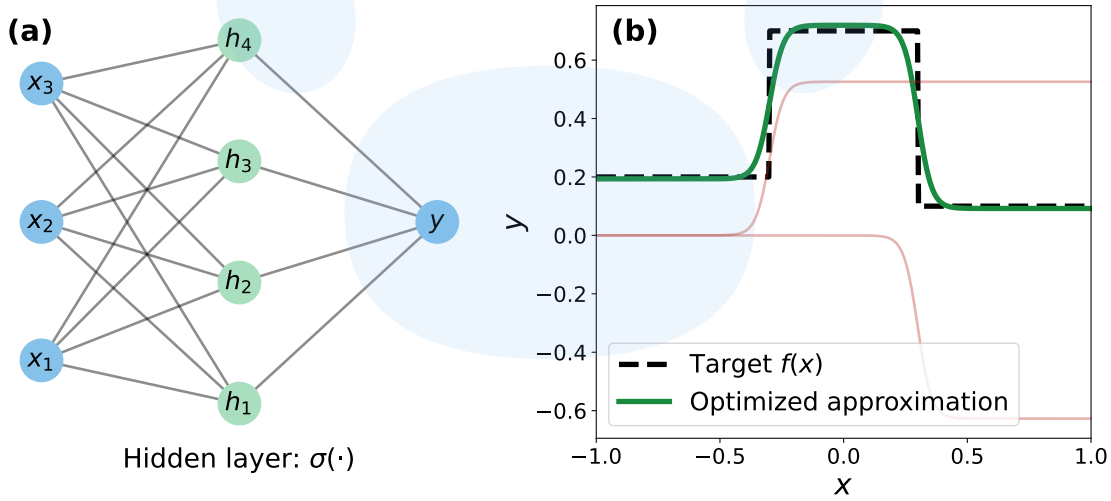

图 2.1　(a) 单层隐藏层的多层感知机示意图；(b) 用单层 sigmoid 激活函数逼近给定函数

物理对称性在神经网络设计中的意义

把数据本身具有的物理对称性写进网络结构，往往比让网络自行学回该对称性更省样本，外推也更稳。

卷积神经网络（Convolutional Neural Network, CNN）是这一思路的早期范例。卷积、局部连接和权重共享对应图像在像素网格上的平移对称性：同一组卷积核在所有位置共享，参数量从全连接的 $O(n^2)$ 量级降到 $O(k^2)$；池化进一步带来局部平移不变性与空间下采样[29]。视觉任务上的实践经验显示，通过发现数据中潜在的物理规律（如平移对称性），神经网络能够更高效地学习和泛化。

循环神经网络（Recurrent Neural Network, RNN）把相似的思路用在序列上：跨时间步共享同一套权重，对应时间方向的平移不变性。简单 RNN 的状态被同一组矩阵反复迭代，梯度在深层时间展开中容易爆炸或消失，使得长程依赖难以被学习；为此，LSTM 和 GRU 引入门控，让状态能在多步之间近似恒等地传输[30-31]，在自然语言处理、语音识别和时间序列任务上长期占主流，直到 Transformer 出现才被取代。

对称性的角色不限于平移变换和缩放。FermiNet 把电子波函数对粒子交换的反对称性写进网络结构，无需显式构造 Slater 行列式即可求解电子基态[32]；类似的思路在分子动力学和自旋系统中也反复出现。

反过来，神经网络也可以成为发现新对称性或新解析关系的工具。AI-Feynman[33]通过神经网络拟合插值，把高维数据中潜在的解析表达式抽取出来；AI-Poincaré[34]则直接学习数据空间上的等变变换，把守恒量和生成元显式呈现。两类工作说明，对称性既可以作为网络的输入约束，也可以是网络的输出对象。

优化与训练动力学

在神经网络的训练过程中，如何确定网络参数以最小化损失函数是一个核心问题。反向传播算法提供了高效计算梯度的方法，但优化过程的动力学特性仍然是一个活跃的研究领域。核心困难在于：在深层网络中，损失函数的非凸性导致了复杂的损失函数景观，包含大量的局部极小点、鞍点和高阶平坦区域，然而在实践中，随机梯度下降（Stochastic Gradient Descent, SGD）及其变体往往能够找到泛化良好的极小点。这一反直觉的现象引发了学界对优化动力学的深入研究。尽管已有不少理论尝试，从过参数化模型的隐式正则化效应[35]，到热力学视角[20]，乃至场论工具[36]，学界仍然缺乏一个全面的理论框架来理解深度网络优化的动力学特性。

工程方面的努力同样值得关注，为了加速训练过程的收敛，研究者们提出了多种优化算法。一种方案是给梯度加上动量，让更新量沿历史方向累积、在平坦区抑制震荡，可以追溯到 Polyak 的早期工作[37]；按参数的历史梯度动态调整学习率是另一种代表性的方案，从 AdaGrad、RMSprop 到 Adam 都属于这一分支[38]，Adam 是目前训练大模型的默认选项，部分原因在于对初始学习率不敏感。另一方面，正则化技术的引入也显著提升了模型的泛化能力。权重衰减、Dropout 和批归一化分别从控制容量、抑制神经元共适应、稳定优化几个角度切入[39]，实际训练里往往按数据规模与网络深度组合使用，有效地防止了过拟合现象的发生。

从神经网络到智能体的技术图景

感知、决策和适应是智能体的三个基本能力，而神经网络为每一项都提供了技术基础。CNN[29]、RNN[30]等架构将高维原始输入压缩为语义丰富的表征向量，使智能体

无需手工设计的特征即可提取对决策有意义的信息。通用逼近定理[26]保证了神经网络能够表示任意复杂的策略函数和价值函数，深度网络直接参数化从状态到动作的映射，使智能体在高维连续空间中快速做出决策。反向传播算法[27]使从感知到决策的整条链路可以被统一优化，智能体在与环境的交互中持续调整自身参数以适应变化[40]。小型神经网络为智能体提供了这些基础能力，而 LLM 将表征和生成能力进一步推向了语言理解与推理的新高度，第 2.2节将展开讨论。

2.2 大语言模型的原理

在现代智能体中，LLM 是最核心的组成部分。LLM 通过在大规模文本数据上预训练，学习到丰富的语言模式和世界知识，从而具备强大的语言生成与理解能力，为智能体提供了归纳、推理和适应的基础。LLM 的成功既得益于 Transformer 架构的设计及后续大量改进，也得益于语料库的极大丰富和计算资源的飞速提升，是深度学习技术的集大成者。与之对应，LLM 也面临着深度学习理论中的诸多挑战，其中最根本的困难在于：LLM 的训练和推理过程是一个高度非线性的“黑箱”，学界对其内部机制缺乏深入理解，导致在设计、优化和评估 LLM 时面临诸多不确定性。理解 LLM 如何支撑智能体的能力，是建立智能体理论的前提，也是推动智能体设计与应用的关键。本节将简要介绍 LLM 的核心架构与关键技术，并讨论其中的理论前沿。

Transformer 架构与注意力机制

LLM 的生成单元是词元（token），每个词元代表一个单词、子词或字符。在一般的 Transformer 架构中[41]，输入词元序列首先被映射到 d_{embed} 维的嵌入空间 $\mathbb{R}^{d_{\text{embed}}}$，形成初始嵌入序列 $\{\boldsymbol{t}_{i,0}\}$。随后，Transformer 通过交替堆叠注意力层和前馈网络层，逐层变换词元表示，产生下一个词元预测的输出。这些输出经过一个线性层映射为词汇表上的 logits，再经 softmax 归一化为概率分布，从而确定下一个词元的生成。

Transformer 的核心是自注意力层，它将某一层的输入序列 $\{\boldsymbol{t}_i\}$ 通过线性投影得到查询（Query）、键（Key）、值（Value）：

$$\boldsymbol{q}_i = W_Q \boldsymbol{t}_i, \quad \boldsymbol{k}_i = W_K \boldsymbol{t}_i, \quad \boldsymbol{v}_i = W_V \boldsymbol{t}_i. \tag{2.4}$$

注意力机制通过计算查询与键的相似度来对值进行加权聚合，形成输出：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right) V, \tag{2.5}$$

其中 d_k 为键的维度。这一机制使得每个位置都能聚合全局信息，并通过 softmax 产生概率权重。多头注意力（multi-head attention）则并行执行多个注意力函数，将结果拼

接后通过前馈网络传播到下一层。

如果将第 ξ 层注意力和前馈网络变换联合视为作用于词元表示流形 $\mathcal{E}_\xi = \{t_i | \forall i,$ 对所有可能的序列输入$\}$ 的算子 $\mathcal{F}_\xi$[18]：

$$\mathcal{F}_\xi : \mathcal{E}_\xi \to \mathcal{E}_{\xi+1}, \quad \boldsymbol{t}_{i,\xi+1} = \boldsymbol{t}_{i,\xi} + F_\xi \left(\sum_{j,k,l} \hat{\Gamma}_{ijkl,\xi} (\hat{\boldsymbol{t}}_{j,\xi} \otimes \hat{\boldsymbol{t}}_{k,\xi}) \cdot \hat{\boldsymbol{t}}_{l,\xi}, \phi(i) \right). \tag{2.6}$$

此处 $\hat{\Gamma}_{ijkl,\xi}$ 为注意力张量的紧凑形式，$\otimes$ 表示外积，F_ξ 为包含位置编码和层归一化的前馈网络所定义的非线性变换，$\phi(i)$ 为位置编码。这一形式表明，Transformer 层可以通过注意力对词元的相关性进行非线性重组，逐步调整流形结构。利用词元关联子分析[18]可以观测到，随着层数加深，词元表示所在的流形空间经历先快速膨胀后逐渐收缩的过程（示意图 2.2），逐渐从高维特征空间投影到约 10 维的低维语义空间，这一维度与人类语义空间的维度十分接近[42]。

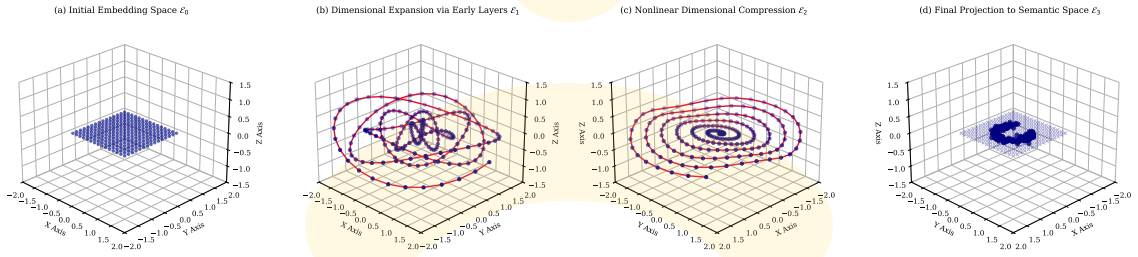

图 2.2　词元表示流形在 Transformer 层间的演化过程示意图[18]

微观词元动力学的理论视角

上述描述说明，LLM 的内部机制可以被理解为词元表示在高维空间中的逐层演化。围绕这一微观端点，若干相互重叠的研究线索已经提供了可用的理论语言。一种理想化路径是从可解模型（solvable model）出发，在简化系统中建立尽可能完整的理论图像。较早的可解神经动力学模型表明，简单系统已经可以表现出集体计算能力和吸引子结构[43]；在现代神经缩放理论中，Maloney 等人[36]通过可解的随机特征模型将幂律缩放与数据分布的频谱结构联系起来。后者讨论的是训练和规模定律（Scaling Laws）等宏观规律，但它也提示了一个重要方向：宏观幂律可能来自较低层次的可解结构。

然而，可解模型通常需要强简化假设。要进一步接近真实的 Transformer 模型，一种自然的中间描述是表示几何（representation geometry）：不直接求解全部参数动力学，而是刻画表示空间如何被逐层变形。深度网络的传播过程可以被视为对输入表示流形的逐层变换，分类能力与流形之间的可分性和几何结构密切相关[44]。在 Transformer 模型中，词元关联子（token correlation）方法进一步揭示了层间表示的维度演化：模型先将词元展开到高维工作空间，再逐步压缩到低维语义子流形[18]。这一路径提供了连接

词元级表示和语义能力的几何语言。

如果说表示几何描述的是表示空间的整体形状，那么机制可解释性（mechanistic interpretability）进一步追问这些形状由哪些局部计算结构产生。Transformer 电路（Transformer Circuits）系列直接以 Transformer 语言模型为对象，将残差流（residual stream）、注意力头组合和 MLP 层视为可分解的计算通道[45]，并把归纳头（induction heads）与上下文学习能力的出现联系起来[46]。在此基础上，稀疏自编码器（sparse autoencoder, SAE）和字典学习方法被用于从中间激活中分解出更接近单义的特征[47]。这类工作并不直接给出宏观性能定律，而是更接近对局部表征和计算单元的解析。

这些几何结构和局部计算结构并不是静态给定的，而是在训练过程中形成、重排并可能发生突变。学习动力学因此更接近复杂系统视角：它与可解模型追求封闭理论图像不同，不要求把所有行为还原为单一可解结构，而是承认训练过程中可能出现涌现、突变和跨尺度重组。例如，顿悟（grokking）现象可以被解释为从惰性训练动力学向丰富特征学习动力学的转变[48]；大模型涌现能力则显示某些任务表现会随规模增长出现非线性变化[49]。这些结果已经接近训练和规模定律的宏观端点，但仍然暗示能力变化背后存在表示结构和动力学状态的重组。

不过，这些微观研究仍有明显局限。它们通常依赖受控模型、局部激活分析或特定训练现象来获得可解释性，因此能够说明 LLM 内部存在可分析的结构，却尚不能直接给出训练后模型在任务中的可观测行为。在微观词元动力学与宏观能力之间建立可计算的桥梁，是第四章需要处理的问题。

大语言模型的训练

LLM 的训练通常分为预训练（pre-training）、有监督微调（supervised fine-tuning, SFT）和基于人类反馈的强化学习（reinforcement learning from human feedback, RLHF）三个阶段[50]。

预训练 预训练是 LLM 获取基础语言能力和世界知识的核心阶段。模型在海量的无标注文本语料上进行自监督学习，典型的目标函数是最大化下一个词元的预测概率（即自回归语言建模，autoregressive language modeling）。给定词元序列 $x_1, x_2, \ldots, x_T$，模型最小化负对数似然：

$$\mathcal{L}_{\mathrm{PT}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta), \tag{2.7}$$

θ 为模型参数。在足够大的语料和足够长的训练时间下，这一极简目标足以让模型从词法、句法一直学到长程语义依赖，输出可以直接被下游任务使用的通用表示。代价是工程上的：单次预训练往往需要数千张 GPU 或 TPU 运行数周到数月，处理的词元规

模在万亿量级。

从理论上理解预训练过程中知识如何被存储和提取，是一个基础而困难的问题。Allen-Zhu 和 Li 的系列受控实验给出过一个重要结论[51]：同一条事实如果只以单一表述出现在预训练数据中，模型可以记住却几乎无法在下游被提取出来；只有当训练语料里这条事实被以释义、改写、翻译等多种方式重复出现，提取才稳定。这一发现表明，预训练不仅是一个记忆过程，更是形成语义结构和泛化能力的关键阶段，为了解 LLM 内部机制提供了重要线索。

有监督微调　预训练后的模型虽然能够生成流畅的文本，但未必能准确遵循人类指令或执行特定任务。有监督微调在高质量的人工标注数据上进一步训练模型，使其学会按照预期方式响应指令。典型的微调数据是“指令—回答”对，训练目标依然是最大化目标回答的条件概率：

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{(x,y)\in\mathcal{D}_{\mathrm{SFT}}} \log P(y \mid x;\theta), \tag{2.8}$$

其中 x 为指令，y 为标注的回答，$\mathcal{D}_{\mathrm{SFT}}$ 为微调数据集。这一阶段通常只需少量高质量数据和较少的训练步数，即可显著提升模型的指令跟随能力和安全性。公式 (2.7)与 (2.8)的结构相似，这表明某些能力或许无须以词元级微观理论解释，而可通过更高层次的抽象来理解。

基于人类反馈的强化学习（RLHF）　仅靠 SFT，模型仍可能在风格、安全和价值观上偏离用户期望。RLHF 把人类对输出的偏好数据引入训练，分为奖励模型与策略优化两步[50,52]。

奖励模型 r_ϕ 从人类的成对偏好中学到打分。给定同一个提示下的两条输出，标注者指明哪一条更好，奖励模型用 Bradley-Terry 形式拟合这种偏好：

$$\mathcal{L}_{\mathrm{RM}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\mathrm{RM}}} \log \sigma\big(r_\phi(x,y_w) - r_\phi(x,y_l)\big), \tag{2.9}$$

其中 y_w 和 y_l 分别为偏好更好和更差的输出。

然后，利用近端策略优化（Proximal Policy Optimization, PPO）[53]微调策略模型 π_θ，以最大化期望奖励：

$$\mathcal{L}_{\mathrm{RL}} = -\mathbb{E}_{x\sim\mathcal{D}_{\mathrm{RL}},y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right], \tag{2.10}$$

其中 π_{ref} 通常是 SFT 模型，β 控制 KL 惩罚的强度。PPO 通过裁剪机制稳定更新，使得模型在提升奖励的同时保持生成多样性。

大模型的生成和推理

预训练目标只是预测下一个词元，但随着模型规模上升，模型在不少需要多步组合的任务上突然变得可用[49]：算术、常识推理、代码生成等任务上的准确率会在某个规模阈值之后非线性提升，而不是平滑增长。这种现象使得 LLM 在智能体中不再只是语言模式匹配器，而是承担更接近智能体推理引擎的角色；也激发了一类把它当作相变现象研究的工作[19]。

思维链（Chain-of-Thought, CoT）提示是最具影响力的推理增强技术之一[54]。通过在提示中加入中间推理步骤的示例，模型被引导生成逐步的逻辑链条来解决问题。在此基础上，自洽性（self-consistency）技术通过多次采样生成多条推理路径，再聚合答案，显著提高了最终结果的可靠性[2]。

更深层次的突破来自强化学习微调：利用可验证的反馈信号（如数学答案的对错、代码能否运行）来优化模型的推理过程，其中通过结果奖励模型（Outcome Reward Model, ORM）或更精细的过程奖励模型（Process Reward Model, PRM）[55]，LLM 能够学会构造更长的、包含回溯和自我验证的 CoT。DeepSeek-R1[56]是这一范式的代表，它通过纯粹的强化学习使模型自发涌现出反思和纠错等复杂推理行为。

规模定律

LLM 的性能与模型规模、数据量和计算资源之间存在幂律关系。Kaplan 等人[57]首先系统性地建立了这一经验关系，发现交叉熵损失 $\mathcal{L}$ 与参数量 N、数据量 D 和计算量 C 分别满足幂律：

$$\mathcal{L}(N) \propto N^{-\alpha_N}, \quad \mathcal{L}(D) \propto D^{-\alpha_D}, \quad \mathcal{L}(C) \propto C^{-\alpha_C}, \tag{2.11}$$

且这些关系在跨越七个数量级的范围内保持稳定。Hoffmann 等人[58]进一步修正了最优配比关系，发现在固定计算预算下，模型参数量和训练数据量应当等比例扩展，即 Chinchilla 最优条件。

规模定律不只发生在预训练侧。Snell 等人[59]考察了推理时计算的扩展：在固定模型下，增加采样次数、延长 CoT 等推理时算力投入同样带来可观的性能上升，在部分任务上甚至比同等代价的预训练扩参更划算。“思考更久”与“模型更大”在这里成为互补的两条路径。

从理论角度看，规模定律的存在暗示 LLM 的学习过程服从某种深层的统计结构。Maloney 等人[36]从随机特征模型出发，把幂律缩放与自然数据的频谱密度联系起来，给出了可精确求解的理论框架；Allen-Zhu 和 Li[60]则从知识存储的角度量化了规模定律：每个参数大约可承载 2 比特知识，并且这一上界在常用的量化与稀疏化方案下基本不

变。

更近期的研究从效率和密度的角度重新审视了规模定律。Xiao 等人[61]提出了容量密度（capacity density）的概念，定义为达到同等性能所需的参考模型参数量与实际模型参数量之比。他们的实证分析揭示了密度定律（Densing Law）：近年来开源 LLM 的容量密度大约每三个月翻一倍。这意味着 LLM 不仅在规模上增长，更在参数效率上持续优化，使得同样规模的模型所蕴含的有效能力随时间推移呈指数增长。

Liu 等人[20]从热力学的视角审视训练过程，发现在河谷形损失景观假设下，温度、熵、热容等热力学量以及热力学三大定律和能量均分定理自然地涌现于训练动力学中。这一框架既为理解训练行为提供了物理直觉，也为学习率调度的设计提供了实用指导。

评估：基准与性能度量

如果说规模定律从资源投入的角度提供了模型性能的宏观预测，那么评估就是从产出的角度度量这些投入是否有效的核心手段。没有可靠的评估，就无法判断一个模型是否比另一个更好，也就无从指导训练和架构的改进。因此，评估与规模定律共同驱动了 LLM 的演化。随着 LLM 能力的提升，评估方法也在不断演化，整体趋势是从静态知识问答走向端到端的能力测试。

早期的评估基准以 MMLU[62]为代表，通过大量选择题覆盖多领域的知识和推理能力。随着模型在这些基准上接近饱和，更精细的评估需求催生了领域专用的基准。PHYBench[63]收录了 500 道原创物理问题，涵盖从高中到物理竞赛的难度范围，并引入了表达式编辑距离（Expression Edit Distance, EED）评分，提供了比二值评分精细得多的评估。即使是当前最强的推理模型也与人类专家存在显著差距。更近期的 PRBench[64]将评估推向了真正的端到端：要求 AI 智能体自主阅读一篇物理论文，从零实现其算法并复现定量结果，当前最强模型的平均得分仅为 34%，端到端复现成功率为零。从 MMLU 到 PHYBench 再到 PRBench，可以观察到评估方法正在从考查孤立知识点向考查完整的科学研究能力演进。

另一方面，端到端评估的昂贵性也促使研究者开发更加高效的评估方法。一系列通过可解释指标鉴定 LLM 能力的研究也在发展。例如词元关联子方法[18]仅需一次前向传播即可估算模型的有效工作维度，在不执行下游任务的前提下预判模型的相对性能，为模型选择和能力诊断提供了计算开销较低的指标。

大语言模型对当代智能体的意义

与确定性工具不同，LLM 的生成过程本质上是概率性的：给定相同的输入，它可以产生多种不同的输出。这一特性使 LLM 在智能体中发挥着独特作用，既可以将集中的输入展宽为多样化的候选，也可以将分散的多路信号收束为集中的输出。

在 FunSearch[5]、AlphaEvolve[6]、IdeaSearch[11]等迭代智能体中，LLM 从当前最优解出发生成多样化的新候选，其概率性展宽特性驱动了搜索。在 Voyager[13]中，LLM 将游戏状态、技能库检索和环境反馈融合为单一动作代码；在 Generative Agents[14]中，LLM 将记忆检索、反思和社交上下文综合为行动决策；在 Inner Monologue[65]中，LLM 将成功检测、场景描述和人类反馈汇聚为机器人规划。展宽与收束是同一概率性生成过程在不同输入条件下的表现。第 2.3节将讨论 LLM 如何在智能体架构中承担核心角色，而第四章将用形式化语言刻画这种概率性生成的物理规律与能力边界。

2.3 智能体的发展

智能体（Agent）是指能够感知环境、做出决策并执行动作以实现目标的系统。从最早的规则系统到当代基于大语言模型的智能体，智能体技术的发展体现了人工智能从符号主义到连接主义、从专用到通用的演化。本节的作用是给出后文所需的技术背景：沿着这一历史线索，介绍智能体的形式化定义、关键技术和架构设计模式，并说明现代 LLM 智能体为何不能只被理解为单次文本生成。

智能体的形式化定义

形式上，智能体可以用马尔可夫决策过程（Markov Decision Process, MDP）来描述。一个 MDP 由五元组 $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ 定义，其中 $\mathcal{S}$ 为状态空间，$\mathcal{A}$ 为动作空间，$\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ 为状态转移函数，$\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 为奖励函数，$\gamma \in [0, 1)$ 为折扣因子。智能体的目标是学习策略 $\mu : \mathcal{S} \to \Delta(\mathcal{A})$，使累积期望奖励 $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$ 最大化。

在更一般的情况下，智能体可能无法直接观测完整的环境状态，而只能获取部分观测 $o_t \in \mathcal{O}$，此时问题推广为部分可观测马尔可夫决策过程（Partially Observable MDP, POMDP）。无论在哪种形式化下，智能体的核心循环都是相同的：观测环境状态，依据策略选择动作，执行动作并接收反馈，然后更新内部状态或策略。

这一形式化虽然简洁，但对于现阶段理解现代 LLM 智能体已经足够：状态空间就是上下文窗口里的自然语言，动作空间则包含生成文本、调用工具与用户交互，奖励信号由任务完成与否提供。$\Delta(\mathcal{S})$ 是 $\mathcal{S}$ 上的概率分布空间，其中的元素正是第四章算子图各条有向边上承载的分布 π。这一视角为后续讨论智能体的物理规律提供了必要的数学框架。

从规则到学习：智能体的演化

通过策略表示方式的几次重大变化，可以了解智能体的演化轨迹。

最早的智能体系统基于人工编写的规则。专家系统（expert system）通过“如果—那么”规则链来模拟人类专家的决策过程，行为树（behavior tree）则为游戏角色和机器人提供了层次化的决策结构。这类系统在规则明确的封闭环境中表现优异，但面对复杂多变的开放环境时，手工编写的规则很快变得不可维护[16]。

强化学习智能体通过与环境的试错交互来自动学习策略，克服了规则系统对人工知识的依赖。Mnih 等人[40]提出的 Deep Q-Network（DQN）首次将深度神经网络与 Q-learning 结合，直接从原始像素输入学习 Atari 游戏策略，标志着深度强化学习的开端。Silver 等人[66]的 AlphaGo 进一步将深度学习与蒙特卡洛树搜索结合，在围棋中击败人类世界冠军。随后的 AlphaZero[67]证明了通过纯粹的自我博弈，无需任何人类先验知识，即可在围棋、国际象棋和将棋中达到超人水平。近端策略优化（Proximal Policy Optimization, PPO）[53]等策略梯度算法的提出则使得强化学习在连续动作空间中变得更加实用和稳定。

然而，强化学习智能体通常需要大量的环境交互才能收敛，且其策略难以迁移到新任务。大语言模型的出现改变了这一局面：LLM 智能体不再需要从零开始学习，而是通过预训练获得了丰富的世界知识和推理能力，从而能够在零样本或少样本的情况下处理复杂任务。这一进步源于 LLM 将策略表示从数值向量（如 Q 值或策略网络的参数）提升到了自然语言层面。

大语言模型智能体的核心技术

在 LLM 智能体的设计中，推理引导、能力扩展和协作组织是绕不开的几件事。

最基础的一步是上下文工程（Context Engineering）。Yao 等人[1]的 ReAct 把推理（Reasoning）与行动（Acting）交织起来：每一步先用一段推理文字分析当前情况，再决定下一个行动（如查询知识库或执行操作），行动结果又反馈回推理。这种模式能明显缓解纯推理链中常见的幻觉（hallucination）[68]和错误累积问题。Yao 等人[4]的思维树（Tree of Thoughts, ToT）把线性推理链扩展成树状搜索，允许模型探索多条路径并在失败时回溯，对需要深度规划的任务提升明显。

工具使用把能力边界从模型自身的参数化知识推到了外部系统。Schick 等人[3]的 Toolformer 表明语言模型可以自行学习何时调用计算器、搜索引擎或翻译系统等外部工具，以及如何将返回结果接到后续的文本生成里。

更复杂的情形是让多个 LLM 协同工作。Hong 等人[69]的 MetaGPT 把标准化操作流程编码进提示序列，让不同角色的 LLM 智能体之间能高效协作；Park 等人[14]的生成式智能体（Generative Agents）则让 25 个 LLM 驱动的智能体在一个模拟城镇里生活、互动、协作，并从中涌现出组织聚会、建立社交关系等社会行为。科学发现领域里，LLM 智能体与外部工具和评估器组合起来同样展示出强大的潜力[5-6,9,11]：LLM 承担创造性

提议，外部模块做确定性的验证和评估，这种分工的具体例子留到第三章详细讨论。

智能体架构的设计模式

随着 LLM 智能体的快速发展，一些共性的架构设计模式逐渐浮现。

最直接的方式是并行搜索：让多个智能体实例同时尝试解决同一问题，然后从中选择最优解。Pass@k 指标和多数投票（majority voting）[2]策略就是这一思路的体现：生成 k 个候选方案，前者度量其中至少一个正确的概率，后者聚合 k 个独立推理路径产生最终输出。这种方法虽然简单，但实践中效果显著。并行搜索的有效性取决于搜索空间的先验结构：当假设空间具有良好的覆盖性质时，增加采样次数能够系统性地提高找到优质解的概率。

比并行搜索更深入的是迭代进化。FunSearch[5]首次证明了 LLM 可以在进化框架中承担创造性角色：将 LLM 作为变异算子生成程序候选，配合系统性的评估和选择机制，FunSearch 在组合数学的开放问题（如 cap set 问题）上发现了超越已知最优解的新构造。AlphaEvolve[6]大幅推广了这一思路，通过集成 Gemini Flash 和 Gemini Pro 两种模型来平衡探索的广度和深度，在矩阵乘法优化等经典问题上取得了突破性结果。The AI Scientist[70]则将迭代进化框架推向科学研究本身，实现了从假设生成、实验设计到论文撰写的全自动化。在高能物理中，Song 等人[8]将类似的进化搜索框架应用于 Feynman 积分约化的优化，通过 LLM 搜索最优的优先级函数实现了数千倍于人类专家设计优化方案的效率提升，展示了该范式在计算密集型科学问题中的巨大潜力。

嵌套与递归是另一个重要的设计模式。在实践中，绝大多数高性能的智能体系统都不是由单个 LLM 调用构成的，而是由多个 LLM 实例各司其职、层次化地组织而成[17]。例如，AlphaEvolve[6]集成了快速模型（负责广度探索）和精确模型（负责深度改进）的分工协作；Meyerson 等人[71]的百万步任务求解系统通过嵌套智能体实现了错误率为零的长程推理。这种嵌套不仅是工程上的便利，更暗示着一条深层的设计原则：当多个 LLM 单元被层次化组织时，系统整体的性能响应可以超越单通道响应率的上界[22]，也即正耦合条件下多通道架构的总响应率能够超越单通道独立缩放时任何一个的上限。从这个角度看，嵌套是将计算资源注入智能体的必要组织形式。本文将在第四章对此给出定量分析。

策略表示的演化与能力边界

策略表示的抽象层次从显式的规则表，到数值化的值函数，再到自然语言层面的推理和决策，不断提升。这一演化使智能体能在越来越开放的环境中运作，但也使理解和预测其行为变得困难：基于 LLM 的智能体是否具有系统性的能力边界，以及这些边界由什么因素决定，是接下来的核心问题。上述架构模式表明工具、记忆、搜索和协

作等维度能够描述系统形态，但还不足以解释不同架构在能力边界上的共性。第三章将从实践角度考察这些边界的具体表现，理论层面的定量分析则构成了第四章的核心内容。

第三章　智能体的实现与应用

为了理解智能体的能力边界，首先需要考察正在运行的各类具体系统与任务。上一章回顾了智能体的概念和架构的图像，本章则关注已经运行在具体环境中的代表性系统，从实践案例中观察哪些约束反复出现。考察的顺序按验证信号的来源安排：封闭计算环境中以评估函数驱动的迭代搜索、通过物理实验或代码库与环境交互的闭环系统，以及目标并不预设的开放世界探索。这些案例为第四章的理论框架提供经验基础。

3.1　迭代搜索与科学发现

形式化推理与数学发现　在形式化证明里，每一步推理都要过检查器，最终结论的可靠性（soundness）建立在这种逐步审查之上。

AlphaProof[7]是这一方向的代表。AlphaProof 将 AlphaZero 风格的强化学习与 Lean 形式化证明助手相结合：模型在大量自动形式化的数学问题上进行训练，学习如何在 Lean 的证明搜索空间中导航。面对困难问题时，AlphaProof 能够在测试时动态调整搜索策略，投入更多计算资源探索更深层的证明路径。在 2024 年国际数学奥林匹克竞赛（IMO）中，AlphaProof 达到银牌水平，成功求解了包括被认为最难的第 6 题在内的多道竞赛题，609 名参赛选手中，仅有 5 人解出该题。

然而，逐步验证也将智能体严格限制在可形式化的问题范围内。对于许多科学和工程问题，如发现新的数学构造、优化物理计算策略或设计实验方案，推理的中间步骤无法被形式化验证，这就需要一种不同的正确性保证机制。

迭代进化范式　当无法验证推理的每一步时，可以放弃过程验证，转而通过严格的评估函数验证最终结果的正确性。迭代进化范式正是基于这一思路：LLM 在每轮迭代中提出候选解，评估函数量化最终输出的质量，择优保留后反馈给 LLM 进行下一轮搜索。中间的推理步骤可能是不透明的，但只要评估器能够可靠地判断最终解的优劣，搜索过程就能稳定收敛到高质量的解。与 LLM 的单次生成不同，这一范式的核心在于搜索，即通过大量迭代探索解空间。

这一范式的原型由 FunSearch[5]首次提出。FunSearch 通过维护一个按评估得分分级的程序库，在每轮迭代中从高分程序中采样作为提示，由 LLM 生成新的候选程序，经评估后择优保留。LLM 在这里的角色不是直接给出最终答案，而是作为进化搜索中的变异算子，利用其对代码结构和数学模式的理解来提出有前景的候选。利用这一框

架，FunSearch 在 cap set 问题上发现了超越数十年人类最优的构造，实现了 LLM 在数学开放问题上的首次突破，这是任何单次的 LLM 生成都无法可靠地找到的。

AlphaEvolve[6]在 FunSearch 的基础上进行了大幅扩展，集成了 Gemini Flash 和 Gemini Pro 两种模型，前者负责广度优先的快速探索，后者负责深度优先的精细改进。AlphaEvolve 在矩阵乘法优化中找到了 56 年来对 Strassen 算法的首次改进，并在 Google 数据中心调度优化中节约了约 0.7% 的 Google 全球计算资源。类似的搜索框架也被应用于量子电路设计：Cao 等人[10]利用 LLM 辅助搜索，在 1+1 维 XY 自旋链上自主发现了仅含 4 个参数的紧凑变分拟设，捕捉了边界诱导的对称性破缺（能量偏差低于 1%），并在“祖冲之”量子处理器上得到验证；该框架进一步推广到 2+1 维标量场论，产生了据作者所知该类模型的首个可扩展拟设。The AI Scientist[70]则将迭代进化框架推广到完整的科学研究流程，实现了从假设生成、实验设计、结果分析到论文撰写的全自动化，尽管其质量控制仍依赖外部审稿机制。

把上述系统放在一起看，LLM 驱动搜索的对象按照推理强度落在一个连续谱上。一端是计算密集型任务，算法框架几乎确定，搜索的实际目标是其中某个抽象的无法先验得知的策略函数；另一端是推理任务，LLM 的世界知识直接进入搜索过程，要找的是具有物理意义的解，解空间因此带上语义结构。下面分别展示这两端的典型案例。

符号回归智能体 符号回归，即从数据中自动发现数学表达式，是科学发现里反复出现的核心问题。遗传编程靠对表达式树的随机突变和交叉来搜索候选，AI-Feynman[33]则把神经网络插值与符号操作组合。这类句法搜索原则上能覆盖整个表达式空间，但实际上常受制于两个困难：组合爆炸让搜索效率低下，缺乏约束的句法变异又容易产出过拟合的复杂表达式。

IdeaSearchFitter[11]利用 LLM 作为语义层面的搜索算子（图 3.1）。在多岛屿进化框架中，每个岛屿维护一组候选表达式及其自然语言“理由”（rationale）。LLM 在生成新的候选表达式时，不仅依据数据拟合的反馈，还依据已有候选的自然语言描述来进行“语义变异”。例如，如果当前最优表达式被描述为“包含指数衰减项”，LLM 可能建议“加入幂律修正以描述长程行为”。这种语义层面的引导使得搜索过程偏向于产生物理上合理、概念上连贯的表达式。

在 Feynman 符号回归数据库（FSReD）上的测试表明，IdeaSearchFitter 在噪声鲁棒性和表达式简洁性方面优于多个传统基线。在真实科学数据上的应用中：在部分子分布函数（PDF）的参数化问题中，IdeaSearchFitter 发现的紧凑表达式不仅拟合精度高，而且具有清晰的物理动机，这是纯句法搜索方法难以做到的。LLM 的科学知识和语言理解直接参与搜索，使得解空间具有语义结构。

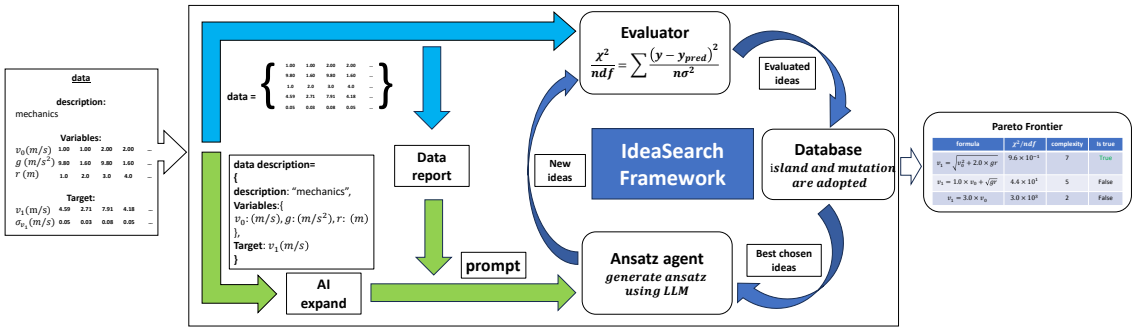

图 3.1　IdeaSearchFitter 的多岛屿语义搜索框架[11]

分部积分程序优化智能体　Feynman 积分约化是高能物理精确计算的核心瓶颈。分部积分（Integration-by-Parts, IBP）恒等式[72]把大量 Feynman 积分写成少数主积分的线性组合，代价是生成和求解恒等式组的开销随问题复杂度迅速膨胀。Laporta 算法[73]给出了系统化的解决方案，但在多圈、多腿的前沿问题上常常遇到内存耗尽或时间不可接受的状况。

Song 等人[8]把 FunSearch 框架应用于 IBP 约化上（图 3.2）。他们注意到约化的效率高度依赖求解顺序，而求解顺序可以由一个优先级函数刻画。通过使用 LLM 与遗传算法协同搜索这个优先级函数，该方法得到了相比传统排序大幅领先的策略。

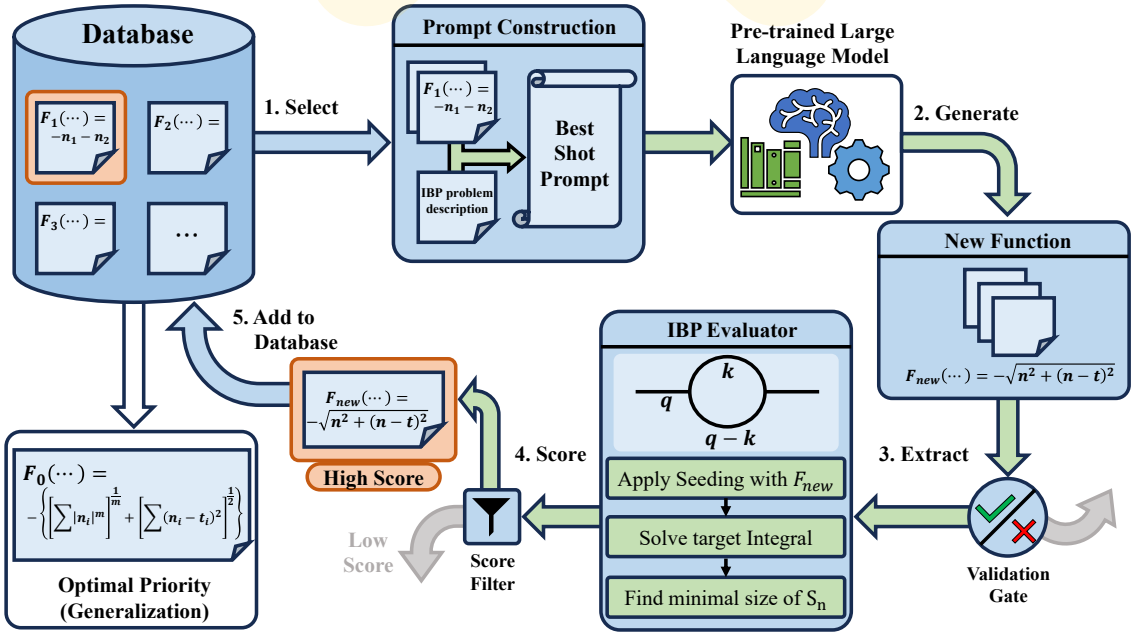

图 3.2　LLM 驱动的 IBP 约化优化框架流程图[8]

这个例子给出了一条清晰的进化路径（图 3.3）。从 Laporta 优先级出发，首先得到 Box 型优先级，把测试目标积分所需的 seeding 数从约 200 压到约 100；继续进化收敛到 Ellipse 型优先级，再降到约 50。三者在指标空间中的几何形状截然不同：Laporta 优

先级的等优先级线是对角直线，与目标积分位置无关；Box 优先级是矩形，开始注意到目标位置；Ellipse 是围绕目标的椭圆，聚焦于其邻域。这条由粗到精的轨迹说明，搜索可以从人类基线出发逐步贴近问题本身的结构。对一类点数与分子较多的 Feynman 积分，效率提升达到 ~ 3,000 倍，且新得到的优先级函数可直接搬到任意圈数和腿数的情形。这表明即便面对这样高度抽象的算法优化问题，LLM 驱动搜索仍能给出可解释的产物，不只是提升了性能，也加深了对 IBP 约化结构本身的理解。

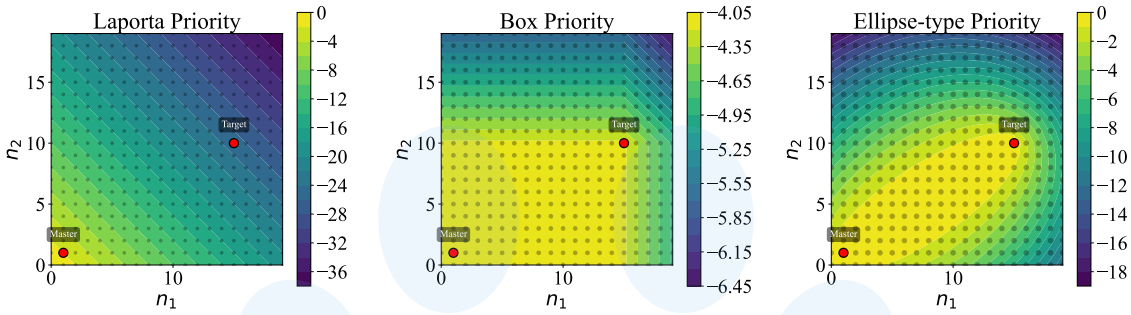

图 3.3　进化搜索逐步发现的三种优先级函数在指标空间 (n_1, n_2) 中的可视化[8]

3.2　环境反馈驱动的智能体

迭代进化范式中的验证完全在封闭的计算环境中完成。然而，在更广泛的智能体应用场景中，验证信号来自与环境的直接交互，包括物理世界的实验反馈与数字环境的测试用例。

闭环实验智能体　科学发现的一个重要瓶颈是湿实验（wet lab experiment）。在这里，智能体不仅要能生成假设和代码，还要能直接与物理世界交互，将验证的闭环延伸到实验台上。

Boiko 等人[9]构建的 Coscientist 是这一方向的代表性工作。Coscientist 以 GPT-4 为核心，集成了网络搜索、文献检索、代码执行和云端机器人实验平台等多个功能模块。在一个典型的任务中，Coscientist 自主完成了以下流程：通过文献检索确定目标反应的实验方案，编写控制代码驱动液体处理机器人执行合成操作，分析实验结果并根据反馈优化反应条件。在 Suzuki 和 Sonogashira 偶联反应的优化中，Coscientist 在几轮自主实验迭代后成功找到了高产率条件。

Bran 等人[74]提出的 ChemCrow 采取了不同的策略。ChemCrow 不依赖闭环实验执行，而是为 LLM 配备了 18 个专用化学工具，包括分子性质预测、反应可行性评估、安全性检查和文献搜索等。通过这些工具，ChemCrow 能够在不实际执行实验的情况下完

成从有机催化剂设计到新型发色团发现等任务。工具驱动的范式降低了安全风险，同时保持了领域专业性。

两套系统对照着看，科学智能体设计中的权衡就显出来：闭环实验给出的是最真实的反馈，但要付安全、成本和执行可靠性的代价；工具驱动更安全也更高效，可靠性却被工具自身的精度和 LLM 对其输出的解读能力卡住。Liang 等人[75]的工作进一步把这种特征暴露出来：即使在完全数字化的沙盒环境中，随着交互轮次的增长，上下文里堆积的工具描述和环境反馈会挤占决策真正需要的信息，性能反而下降。这意味着工具数量本身补不上闭环实验的缺失，智能体的能力边界存在更深层的结构性约束。

软件工程智能体　软件工程是 LLM 智能体最具商业影响力的应用领域之一，同时也是智能体-环境接口设计的重要试验场。与闭环实验智能体类似，软件工程智能体的验证机制来自环境反馈，即测试用例的通过与否，但其环境是完全数字化的代码库而非物理世界。

Yang 等人[12]提出的 SWE-agent 引入了智能体-计算机接口（Agent-Computer Interface, ACI）的概念。类比于人机交互中的用户界面设计，ACI 是专为 LLM 优化的计算环境抽象层。传统的命令行界面是为人类设计的，包含大量对 LLM 而言冗余或误导性的信息；SWE-agent 通过设计简洁的文件查看器、精确的编辑命令和结构化的错误反馈，显著提升了 LLM 在代码库中定位、理解和修复问题的能力。在 SWE-bench 基准上，ACI 的设计使得智能体的成功率从基线显著提高。

SWE-agent 的启发不仅限于软件工程。它揭示了一个在科学智能体设计中同样重要但尚未被充分讨论的维度：智能体的性能不仅取决于 LLM 本身的能力，还取决于环境接口如何向 LLM 呈现信息。在 IdeaSearchFitter 和 IBP 约化优化中，提示模板的设计实际上就是一种 ACI：它决定了 LLM “看到”什么样的数据表示和反馈格式，从而影响搜索的效率和质量。

这一领域的快速发展催生了开源平台 OpenHands[76]和商业产品 Devin[77]等系统，它们将 ACI 设计、多智能体协作和安全沙箱执行整合为端到端的软件工程智能体。截至 2026 年 4 月，前沿 LLM 智能体在 SWE-bench Verified 上已超过 80% 的成功率[78]，展示了 LLM 智能体在结构化编程任务中的强大能力。

3.3　开放世界与具身智能体

前述智能体，无论是证明定理、优化策略、设计实验还是修复代码，都运行在目标明确的任务环境中，成功标准是预先定义的。然而，许多现实场景并不存在明确的目标函数，而是要求智能体在开放世界中自主探索、学习和积累能力，或将能力直接

应用于物理世界和数字环境中的操作。

开放世界探索与技能合成 Wang 等人[13]提出的 Voyager 是这一方向的开创性工作。Voyager 是一个基于 GPT-4 的 Minecraft 智能体，其核心创新在于三个相互耦合的机制：自动课程生成、可增长的技能库和迭代提示反馈。自动课程机制使智能体能够根据当前的探索进度和物品栏状态自主提出下一个应该学习的技能（如“获取木头”→“制作木镐”→“挖掘石头”→“制作石剑”），无需人工设定任务序列。每个成功学习的技能被存储为可复用的 Python 程序代码，构成不断增长的技能库。当面对新任务时，智能体首先从技能库中检索相关的已有技能作为构建模块，然后通过 LLM 生成新的代码来组合和扩展这些技能。

Voyager 在多个维度上显著超越了基线方法：获取的独特物品数量是此前最优方法的 3.3 倍，探索距离是 2.3 倍，技术树推进速度是 15.3 倍，且是首个达到钻石工具层级的 LLM 智能体。更重要的是，Voyager 的技能库具有跨场景迁移能力：在一个新的 Minecraft 世界中，智能体可以直接复用已有技能库，而无需从零开始学习。

从架构设计的角度看，Voyager 的“代码即动作”（code-as-action）范式与迭代进化中的“搜索程序”范式有深层的共通性：两者都利用 LLM 生成可执行代码作为中间表示，通过环境反馈进行迭代改进，且产出物（程序/技能）是可解释、可复用的。差异在于目标结构：迭代进化的目标是优化一个已知的评估指标，而开放世界探索的目标是最大化能力的覆盖范围，后者更接近于一种自驱动的能力搜索。

具身智能体与计算机使用 具身智能体和计算机使用智能体把 LLM 的输出直接接到机器人控制或 GUI 操作上，动作空间因此明显比前述系统更广泛和复杂，既包括抽象的代码生成，也包括具体的物理操作。

Liang 等人[15]的 Code as Policies 把 LLM 生成的代码直接当作机器人控制策略。不同于端到端策略学习，它要求 LLM 按自然语言指令生成一段 Python 程序，去调用底层的感知与控制原语。这样做的优势在组合性和可解释性：复杂的操控任务被分解成一系列简单原语的调用，每一步都能被人理解和验证。

计算机使用智能体[79]则把多模态感知与 GUI 交互组合，通过截屏理解当前界面，再生成鼠标点击、键盘输入等操作完成任务。它不依赖预先定义的 API 接口，原则上可以操作任何软件，是目前最通用、也最具挑战性的一类动作空间。

代码生成、API 调用、机器人控制、GUI 操作这类动作空间也可以大致排成一条谱系：越往后越通用，但控制精度和效率也越难保证。同样的权衡在科学智能体设计中也能看到：FunSearch 通过高度结构化的代码接口精确输出，Coscientist 则用更通用、也更脆弱的机器人控制接口换取与物理世界的直接交互。

3.4　共性趋势与理论需求

上述各范式的实践揭示了几条共性趋势。

LLM 驱动的搜索为困难问题提供了一种有效范式。LLM 单次生成难以可靠地解决具有巨大解空间的问题，但当 LLM 被嵌入进化搜索框架时，系统能够通过迭代改进探索解空间。在许多成功案例中，程序代码是重要的中间表示：它是可执行、可验证、可组合的，使得搜索过程的每一步产出，无论是 FunSearch 的数学构造、IdeaSearchFitter 的符号表达式、Voyager 的技能代码还是 SWE-agent 的代码补丁，都能被精确评估和复用。然而，这些具体表示并不完全相同；形式化证明中的证明状态、符号回归中的表达式、软件工程中的补丁、开放世界中的动作代码和 GUI 任务中的操作序列，都只是不同环境对 LLM 输出的任务层投影。更深层的共同对象，是 LLM 在给定上下文和反馈下诱导的词元序列分布。正是这个分布使不同任务中的搜索、验证和复用可以被放入同一个概率框架中比较。

验证机制的设计决定了智能体的能力边界。从 AlphaProof 的形式化证明到 FunSearch 的数值评估，从 Coscientist 的实验反馈到 SWE-agent 的测试用例，不同的验证机制提供了不同层次的正确性保证，也决定了搜索空间的有效范围。更强的验证机制（如形式化证明）提供更可靠的反馈信号，但也限制了适用的问题类型。即使在具备完善验证机制的沙盒环境中，智能体的能力仍然存在固有上限[75]，这表明验证机制是必要条件而非充分条件，智能体的能力边界由更基本的信息处理约束所决定。

环境接口的设计与 LLM 能力同样重要。SWE-agent 的 ACI 设计、Voyager 的技能库检索机制、IdeaSearchFitter 的语义反馈模板，都表明智能体的性能主要取决于信息如何在 LLM 与环境之间流动。这些观察表明，仅从工程层面理解智能体是不够的：需要一个形式化框架来刻画 LLM 概率性生成的内在约束。其重要性在于，相关证据已经分散出现在科学发现、软件工程、开放世界探索、评测方法和训练与规模定律研究中，但如果缺少共同的数学对象，这些结果很难相互检验和累积。以词元序列分布为中间环节，可以把这些领域中的经验规律组织为同一条从生成到行为的可测量链条，这正是第四章的主题。

第四章　智能体的物理规律

第二章与第三章从神经网络基础、LLM 原理、智能体架构到丰富的实现范式，系统梳理了 LLM 智能体的完整技术图景。这些工作积累了丰富的经验成果，但也凸显出一个问题：学界对智能体系统的理论理解滞后于工程实践。本章要回答的问题是：设计一个新的智能体架构时，除经验和试错之外，是否存在可以依据的原理。

本章提出一个处于初步阶段的理论纲领。智能体的物理学面对的是两个已经可观测、但尚未连通的端点：在微观端，是 LLM 内部的 token、logits、注意力和表示几何；在宏观端，是评测分数、Scaling Laws 与架构设计目标。要把二者放进同一套理论语言中，关键不在于直接从微观机制推出全部宏观行为，而在于找到可测量的中间对象。LLM 的生成具有概率性，同一输入可以产生不同输出，因此智能体每一步执行自然地对应一个词元序列上的概率分布。这些分布构成统计物理意义上的系综[80-81]；对单步转移核的测量即为不可约实验，它把词元级微观动力学通过特定的智能体组件和宏观表现连接起来。

将系综作为理论分析的基本对象，本章的任务是发展从中提取稳定宏观量的方法：这里的宏观量是指不随生成随机性变化、可以稳定测量的物理量。系综本身只能通过端到端实验直接采样，而投影将其压缩为可以用数学工具近似和预测的宏观表示。本章给出两种投影方式：一种将分布线性投影为评估基准上的分数[22]；另一种借用统计物理方法，通过对系综计算自由能得到势函数[23,82]。两者都是将分布投影到不同宏观表示上的操作，区别仅在于投影的目标空间。如图 4.1所示，这一纲领形成了一个类似衔尾蛇的闭环：词元级动力学[18-19,83]决定系综 π 的形态；不可约实验测量转移核，并把 π 投影为可解释、可调控的宏观量；这些宏观量再进入仿真、评测、强化学习[20,84]和规模定律[57]（第4.5节），反过来指导架构设计，并最终影响下一轮模型与智能体的构造。只有当宏观量既能解释智能体行为，又能返回并改变设计流程时，这条从微观生成到宏观设计的理论环路才算闭合。这一方法论不试图从第一性原理推导一切，而是在每个层次上建立自洽的描述，再通过跨层次的联系将它们统一起来。

因此，本章以下内容展开讨论这条纲领中的物理桥梁。第4.1节先用算子图把智能体流程写成作用在分布上的组件组合，并说明不可约实验如何替代完整端到端试错；第4.2节和第4.3节分别给出线性表示和势能表示，展示同一个系综如何导出效用响应、细致平衡和搜索时间复杂度；第4.4节把这些定量约束转化为架构设计原则；第4.5节则讨论训练与规模定律如何为这一推理时框架提供背景和未来接口。由此，微观 token 动力学、序列系综、宏观评测与设计目标被组织为一条可测量、可模拟、可反馈的理论

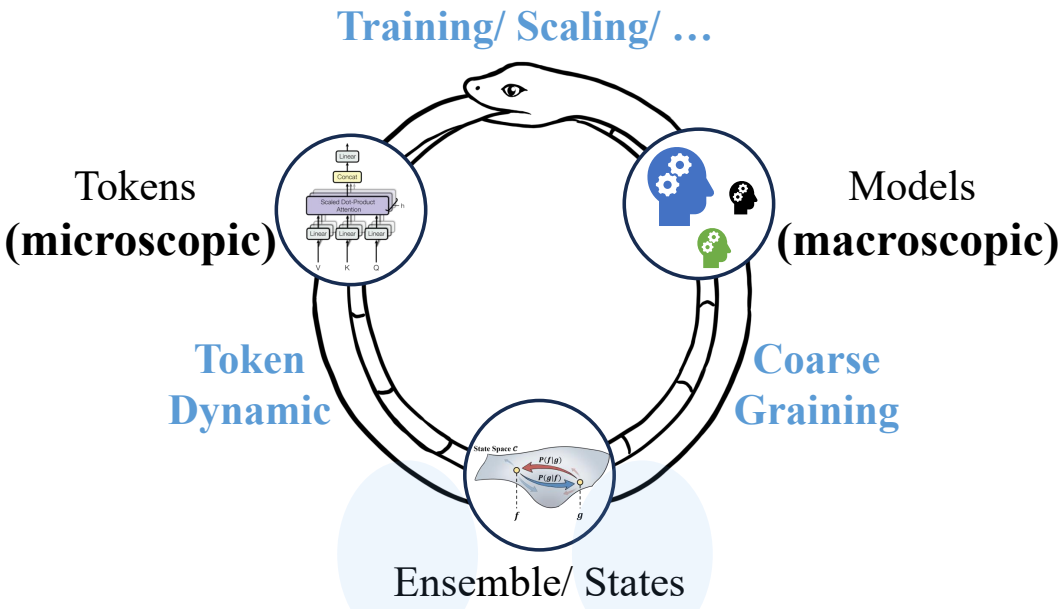

图 4.1　本文研究纲领的理论图景：从微观生成到宏观设计的物理闭环

闭环。

4.1　算子图与算子代数

算子图将给定智能体的完整处理流程表示为一幅有向图：节点是算子（LLM、工具、数据库），有向边携带状态空间 $\mathcal{S}$ 上的概率分布 π。这里的状态空间 $\mathcal{S}$ 由智能体在每个时间步所保留的全部信息定义，包括任务目标、历史摘要、代码、文件系统、API 返回值等。这一构造类似于深度学习中的计算图（computational graph），尽管计算图中的有向边传输张量，而算子图中的有向边传输概率分布。本节定义算子及其组合规则，建立环路算子的代数理论，并以 IdeaSearchFitter 为具体案例展示算子图的构造。

4.1.1　算子与有向边

算子图中的每个节点都是一个算子，表示为一个转移核（transition kernel）$\mathcal{T}((g_1, g_2, \cdots) \leftarrow (f_1, f_2, \cdots))$，将输入边上的状态映射为输出边上的状态。具体的算子在性质上有所不同，例如 LLM 的转移核往往保持为一个概率映射，而工具算子（如数据预处理、数值拟合、评估器等）的转移核往往退化为确定性的一对一映射，即 δ 函数形式的核（对每个输入只在唯一的输出处取非零值）。尽管形式和实现上十分多样，它们都遵循同一种代数形式，区别仅在于转移核的具体结构和所遵守的规律（详见第 4.2，4.3节）。

所有有向边携带的对象为 $\mathcal{S}$ 上的概率分布 π。具体地，$\pi(g)$ 定义为：给定相同的输入，算子每次独立执行时产生输出状态 g 的概率。需要说明的是：确定性的状态同

样可以对应于 δ-型分布，即每次执行都给出同一个输出。因此，工具算子的输入是分布、输出也是分布（确定性映射只是将一个 δ 分布映射为另一个 δ 分布），LLM 算子的输入是分布、输出也是分布（但从 δ 分布展宽为一般分布），有向边上的所有信息流都属于同一类数学对象。图 4.2以数据归一化（工具算子）和提案生成（LLM）为例展示了这一统一性。分布 π 是算子图中最基本的对象，但直接处理 π 往往难以提炼出定量规律。通过对 π 施加不同的数学操作，可以将其投影到不同的表示[85]中。第4.2节将选择一种最简单的投影，即把每个状态映射为在给定基准测试上的性能分数，从而定义效用函数 J 以对智能体的性能边界进行分析。

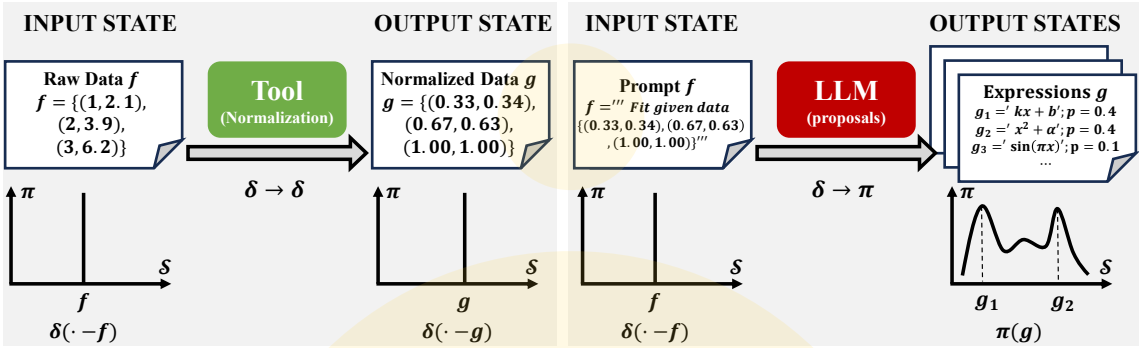

图 4.2　算子图中有向边携带的数学对象示意

4.1.2　粗粒化与可约性

通过算子图，智能体的完整端到端流程被分解为单步算子的组合。这意味着一旦知道了每个组件的转移核的完整描述，就可以通过组合这些核来预测整个系统的行为。尽管状态空间的高维性和复杂性可能使得直接处理转移核变得困难，但如果存在一种合适的粗粒化方法[86]将状态空间映射到一个更小的离散空间，就可能能够在这个粗粒化层面上用对应的矩阵乘法来预测系统的行为。这就使得智能体的完整流程是可约的，即可以通过单步不可约实验的结果来预测多步流水线（pipeline）的行为，而不需要端到端地运行整个系统。可约性的实际意义在于：测量少量基本组件（如一个生成分布、一个修正核、一个工具算子）后，就可以通过矩阵乘法模拟任意由这些组件组装的 pipeline，从而避免对每种架构配置都进行高成本的端到端测试。这为智能体设计打开了一条新的便捷路径：通过不可约实验建立组件库，再通过组合预测筛选最优架构。

为了说明何种粗粒化方法是合适的，需要考虑算子 $\mathcal{T}$ 的结构。定义映射 $\varphi : \mathcal{X} \to \mathcal{S}$，将原始的文本级状态空间 $\mathcal{X}$ 映射到一个有限的离散状态空间 $\mathcal{S}$。在这个粗粒化层面上，文本级分布 $\pi(f)$ 被映射为状态向量 $\vec{\pi}$，其中每个分量 $\pi_i = \sum_{\varphi(f)=i} \pi(f)$ 是所有映射到

状态 i 的文本概率之和。算子 $\mathcal{T}$ 在粗粒化层面上的转移核 T_{ij} 定义为：

$$T_{ij} = \mathbb{E}_{f\sim\rho_i}\left[\sum_{\varphi(g)=j} \mathcal{T}(g \leftarrow f)\right], \tag{4.1}$$

其中 $\rho_i(f) = \pi(f)/\pi_i$ 是状态 i 内的文本条件分布，即在已知粗粒化标签为 i 的条件下，底层文本的概率权重。T_{ij} 同时取决于算子 $\mathcal{T}$ 和态内分布 ρ_i。

这种粗粒化本身并不保证可约性。严格可约性要求 T_{ij} 不依赖于 ρ_i，即无论状态 i 内的文本组成如何变化，转移到状态 j 的概率不变。这等价于 φ 是 $\mathcal{T}$ 的充分统计量：知道 $\varphi(f) = i$ 后，f 的具体内容对预测 $\varphi(g)$ 不提供额外信息。这时，T_{ij} 退化为常数矩阵，粗粒化演化闭合为 $\vec{\pi}' = \vec{\pi}\cdot T$，pipeline 的终端分布可以通过矩阵乘法精确预测：

$$\vec{\pi}_{\text{pred}} = \vec{\pi}_0 \cdot \prod_k T_k. \tag{4.2}$$

为验证式 (4.2)是否可以用于预测实际智能体的行为，本文设计了一组跨模型、跨任务的定量实验（完整方法和评估结果见附录 D.1）。在三个计算型任务域上定义了四态粗粒化映射 φ（正确 C、近似错误 NW、完全错误 FW、无法解析 NA），独立测量了 12 个生成-修正模型组合的生成分布 $\vec{\pi}_0$ 和修正核 T_{ij}^{ref}。在此基础上，构造了 6 种 pipeline 拓扑（图 4.3），用矩阵乘法预测终端分布，并与端到端运行 $N = 200$ 次的经验分布比较。总共 216 条 pipeline，约 58,000 次 API 调用，其中 15,000 次用于不可约测量，43,000 次用于端到端验证。

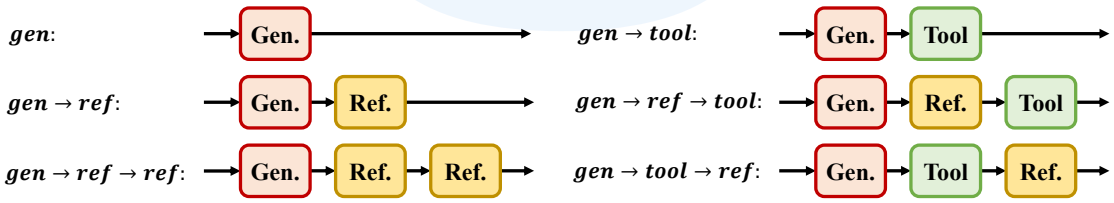

图 4.3　6 种 pipeline 配置的算子拓扑

图 4.4展示了一组代表性的不可约实验。(a) 为 Qwen2.5-7B-Instruct 在组合计数任务上的生成分布 $\vec{\pi}_0$；(b) 为 Qwen2.5-72B-Instruct 的修正核 T_{ij}^{ref}，非对角元素反映了修正能力（如 $T_{NW\to C} = 0.62$）；(c) 为确定性工具核 T_{ij}^{tool}，将所有非 NA 态映射至 C。

图 4.5展示了一组代表性 pipeline 在 6 种配置下的效用 J 的演化。空心圆为矩阵乘法预测，实心方块为端到端经验值（$\pm1\sigma$ 误差棒）。纯 LLM 修正链（gen→ref→ref）使 J 从 0.70 提升至 0.91；确定性 tool 配置完美一致（$L_1 = 0.000$）。效用函数 $J = \pi(\text{C})+0.5\pi(\text{NW})$ 是第4.2节将要讨论的线性表示的一个实例，代表了智能体的端到端正确率。在全部 3 个任务域、12 个模型组合、6 种配置（共 216 条 pipeline）上，预测与经验 J 的平均绝

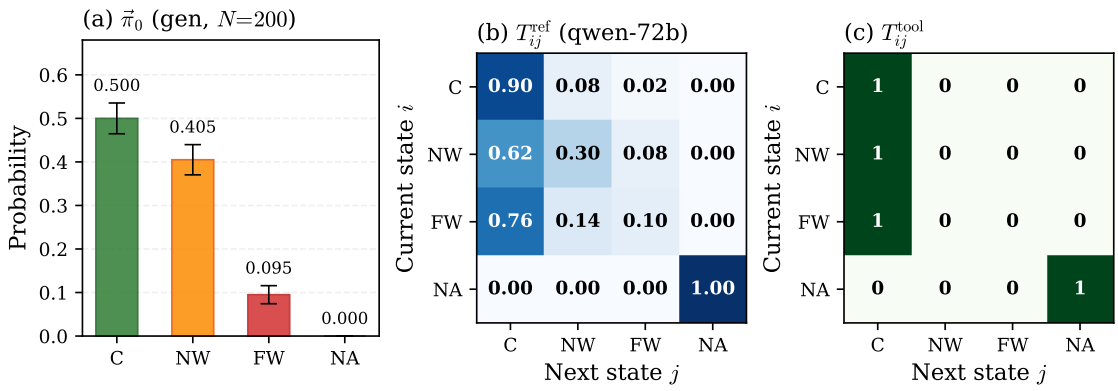

图 4.4　不可约实验测量与工具核（组合计数，Qwen2.5-7B→Qwen2.5-72B）

对误差为 0.022；统计检验表明，在 95% 置信水平下可以接受可约性假设（完整评估与统计检验方法见附录 D.1图 D.1）。

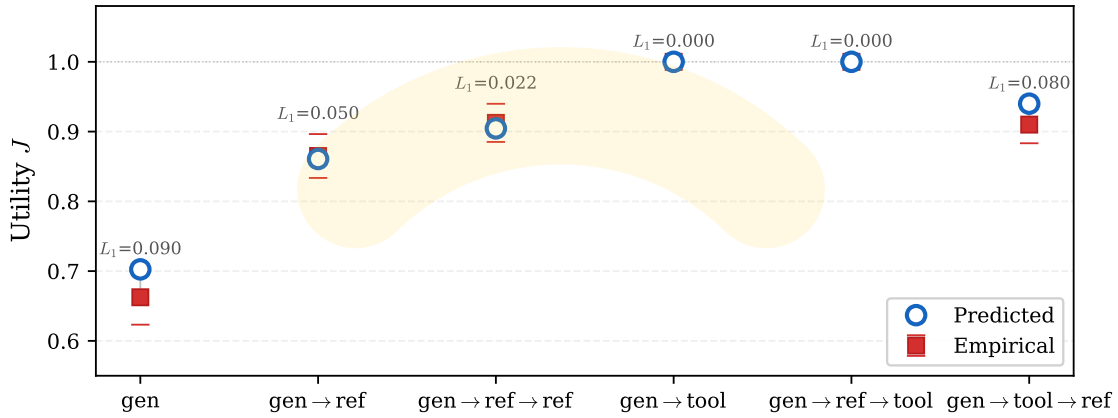

图 4.5　代表性 pipeline 的效用 J 演化（组合计数问题，Qwen2.5-7B-Instruct→Qwen2.5-72B-Instruct）

4.1.3　IdeaSearchFitter 的算子分解

可约性保证了模块化分析的有效性：一旦单步核被测定，任意由这些模块组装的 pipeline 都可以通过矩阵乘法预测。这使得将智能体分解为算子并逐模块分析成为可行的思路。这一方案的实际价值在于避免了智能体端到端评估的高成本：传统的端到端评估需要对每种架构配置都运行完整的测试流程，而可约性允许实验者只测量基本组件的转移核（不可约实验），再通过组合预测来筛选最优架构。本文以 IdeaSearchFitter[11]为具体案例展示算子图的构造方法，随后在第 4.2和 4.3节分别从线性表示和势能表示的角度分析其中的单步核，提炼出可验证的宏观约束。图 4.6将其完整的处理流程翻译为算子图，每个节点标注了对应的算子符号。在该算子图中，输入 π_0 并联进入 $\mathcal{T}_{\text{data}}$ 和

$\mathcal{T}_{\text{prompt}}$，两路输出汇入迭代环路；环路按 $\mathcal{T}_1 \to \mathcal{T}_2 \to \mathcal{T}_{\text{eval}} \to$ DB 的顺序单向循环，收敛后经 $\mathcal{T}_{\text{Pareto}}$ 输出。

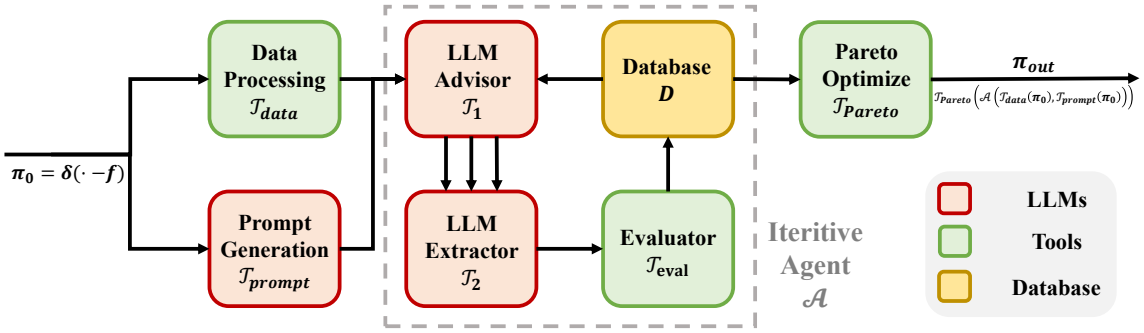

图 4.6　IdeaSearchFitter（fuzzy 模式）的算子图

IdeaSearchFitter 的完整流水线可以写为算子的嵌套结构：

$$\pi_{\text{out}} = \mathcal{T}_{\text{Pareto}}\big(\mathcal{A}\big(\mathcal{T}_{\text{data}}(\pi_0),\ \mathcal{T}_{\text{prompt}}(\pi_0)\big)\big)\,, \tag{4.3}$$

其中 $\pi_0 = \delta(\cdot - X)$ 是输入数据的 δ 分布。$\mathcal{T}_{\text{data}}$（数据归一化、重标定，$\delta$ 核）和 $\mathcal{T}_{\text{prompt}}$（LLM 将数据描述和物理背景组织为 prompt）是两个并联的预处理算子，它们的输出作为两个独立通道同时输入 Agent 算子 $\mathcal{A}$。$\mathcal{A}$ 封装了迭代搜索环路的全部复杂性。它接收多通道输入，内部执行迭代生成并收敛（其内部结构将在第4.3节展开）。$\mathcal{T}_{\text{Pareto}}$（$\delta$ 核）对 $\mathcal{A}$ 的输出进行 Pareto 前沿（Pareto front）优化，确定性地筛选出拟合质量和表达式复杂度之间的最优权衡。

Agent 算子 $\mathcal{A}$ 内部的单步生成操作定义为

$$\text{Gen}(D) = \mathcal{T}_{\text{eval}} \circ \mathcal{T}_2 \circ \mathcal{T}_1\big(\text{Sel}(D);\ \mathcal{T}_{\text{data}}(\pi_0),\ \mathcal{T}_{\text{prompt}}(\pi_0)\big)\,, \tag{4.4}$$

其中 D 是当前数据库状态，$\text{Sel}(D)$ 从中选取历史最优候选及其评分，$\mathcal{T}_1$ 是提案生成器 LLM_1，$\mathcal{T}_2$ 是提取器 LLM_2，$\mathcal{T}_{\text{eval}}$ 是评估器（数值拟合计算 χ^2/ndf 并写入数据库）。在 fuzzy 模式中，$\mathcal{T}_1$ 将集中的输入展宽为分布扇，即每次执行产生完全不同的自然语言理论提案，而 $\mathcal{T}_2$ 将其收敛为一条规范化的符号表达式，使得搜索在广泛的提案空间中探索并可以聚焦到可行的符号形式上。

数据库在每步迭代中积累所有历史状态：

$$\begin{aligned} D_1 &= \text{Gen}(D_0), \\ D_2 &= \text{Gen}(D_0, D_1), \\ D_3 &= \text{Gen}(D_0, D_1, D_2), \\ &\vdots \end{aligned} \tag{4.5}$$

环路持续迭代直到数据库收敛。迭代搜索环路的复杂性被封装在 $\mathcal{A}$ 中，从系统外部看，这只是一个从输入分布到输出分布的单步映射。其内部的马尔可夫描述和势函数 V 将在第4.3节展开。

在算子图的语言下，智能体的设计问题转化为选择最优算子及其组合方式。可约性实验（第4.1.2节）表明，单步核的独立测量足以预测多步流水线的宏观行为，从而保证了将分布投影到宏观表示的合理性。下面两节分别从线性表示和势能表示出发，展示如何从转移核导出可验证的宏观约束。

4.2　线性表示与信息响应率

给定一个经过 pipeline 输出的分布 π 和评分函数 $s : \mathcal{S} \to \mathbb{R}$，效用函数定义为

$$J = \langle s(f) \rangle_\pi = \int_{\mathcal{S}} s(f)\, \pi(f)\, Df. \tag{4.6}$$

J 是分布 π 的线性泛函，因此以下将这种通过 J 分析智能体性能的方法称为线性表示。这是最直观的投影方式：将高维概率分布投影为单个标量。这里评分函数 s 的粒度决定了投影的分辨率：如果 s 是一个非常粗糙的函数（如正确率），则 J 只能反映智能体性能的宏观趋势；如果 s 是一个非常细致的函数（如每个状态的奖励），则 J 可以反映智能体性能的微观细节。pipeline 的计算预算 $\mathcal{B}$（迭代轮数、模型质量等）通过改变 $\pi(\mathcal{B})$ 影响 J，响应率[81] χ 定义为

$$\chi := \frac{\partial J}{\partial \mathcal{B}} = \int_{\mathcal{S}} s(f)\, \frac{\partial \pi(f; \mathcal{B})}{\partial \mathcal{B}}\, Df. \tag{4.7}$$

引入固定 LLM 算子 $\mathcal{T}_{\text{LLM}}$ 作为包装器（wrapper），接收单一输入分布 $\pi(\mathcal{B})$ 并映射为衍生分布 $\pi'(\mathcal{B}) = \mathcal{T}_{\text{LLM}}(\pi(\mathcal{B}))$，Song 和 Zhu[22]提出信息响应率假说：

$$\langle \chi(\pi') \rangle \le \langle \chi(\pi) \rangle, \quad (\mathcal{B} \to \infty), \tag{4.8}$$

即固定 LLM 通道不能放大基础分布的性能响应率。图 4.7给出了直观图像。图中绿色曲线为输入策略集的分布 π，红色曲线为经 LLM 处理后的衍生分布 π'，虚线标注最优状态的位置。LLM 的内在逻辑在两种预算下都将 π' 的中心拉向同一类输出，但净效果不同：小预算下 π 远离最优状态，LLM 将分布拉近最优解，净效果有利；大预算下 π 已接近最优状态，LLM 反而将分布展宽偏离最优解，净效果有害，导致 $J(\pi') \le J(\pi)$。图 4.8在四个完全不同的任务域上验证了这一约束[22]：(a) Tetris（$\mathcal{B}$ 为束搜索宽度）、(b) 0/1 背包（$\mathcal{B}$ 为束搜索宽度）、(c) 世界知识排序（$\mathcal{B}$ 为信噪比）、(d) AIME 题目（$\mathcal{B}$ 为模型规模）。在所有域中，基础分布性能 $J(\pi(\mathcal{B}))$ 的增长率均不低于 LLM 衍生分布性能 $J(\pi'(\mathcal{B}))$。

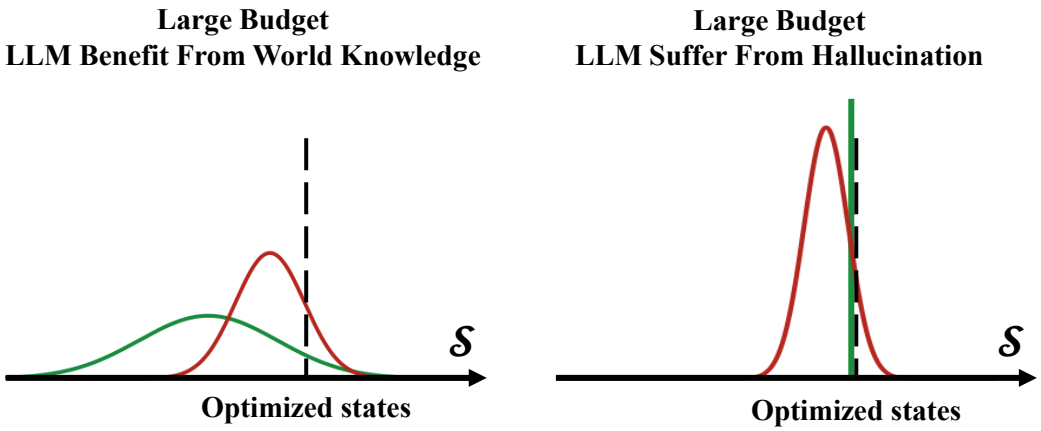

图 4.7　响应率约束 (4.8)的直观解释

4.2.1　响应率理论的应用

上述四个任务域的验证来自 Song 和 Zhu[22]的原始工作，其实验环境经过专门设计以隔离响应率效应。为了检验这一约束在真实智能体框架中是否同样成立，本文补充两个基于已有基准测试和真实模型的例子。在每个例子中，应用的步骤相同：首先在智能体流程中识别作为 wrapper 的 LLM 的位置，确定其输入分布 π 和输出分布 π' 分别对应什么；然后将式 (4.8)代入，得出对宏观可测量量的定量预测；最后用实验数据检验该预测。

推理链长度与正确率的关系　在思维链（Chain-of-Thought）推理中，LLM 在每一步续写时充当 wrapper：尽管每一步的输入上下文随链长增长，但从最外层看，LLM 始终是对其内部全部架构和已有上下文的一个整体包装。具体地，将第 k 步的中间生成视为分布 π_k，续写的下一步输出视为 $\pi_{k+1} = \mathcal{T}_{\mathrm{LLM}}(\pi_k)$。作为对比，平凡处理器（直接返回占位符而不经 LLM 续写）的输出不依赖于链长度，其响应率 $\chi = 0$。式 (4.8)要求 LLM wrapper 的响应率 $\chi' \le \chi = 0$，即 $J(\pi')$ 关于链长度的导数非正：当推理链足够长时，LLM 续写处理的上下文越来越多，而平凡处理器始终不变，因此 $J(\pi')$ 必然下降。如果进一步假设每步的性能衰减率近似恒定，则预测正确解答的概率随链长度指数衰减。Liang 等人[75]在长程智能体任务中独立地观察到了相同的现象：更低的词元消耗反而对应更高的任务完成率，他们将其归因于超过某一阈值后额外词元不再引入有用信息，反而通过注意力稀释降低了推理质量。Wei 等人[87]通过对推理动力学的系统分析进一步证实了这一效应，并将其称为 “overthinking”：推理链存在一个实例相关的完成点，超过该点后继续生成不仅无益，而且引入语义振荡和冗余，导致性能下降。为定量检验这一预测，本文使用 PHYBench[63]的测试时计算扩展数据。由于难题自然倾向于产生更长的推理链且正确率更低，直接使用原始链长度 L_t 可能混淆题目难度与链长

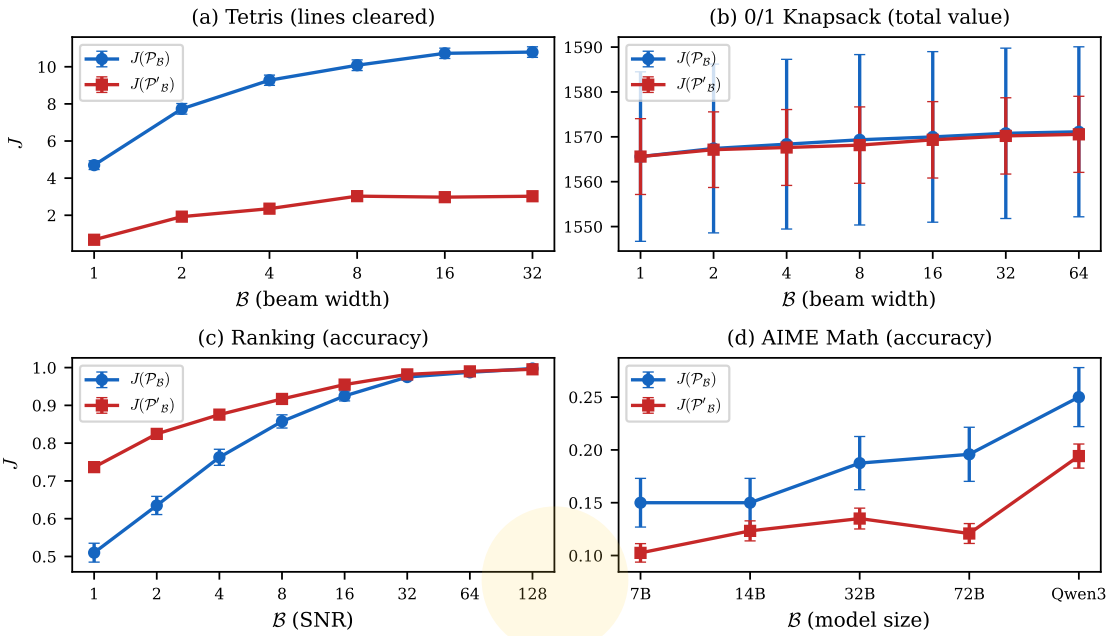

图 4.8　信息响应率假说的跨域验证[22]

度本身的效应。为此，利用每个模型对每道题的多次独立作答（每题 ≥ 16 次）进行逐题中位数归一化：将每次作答的 L_t 除以该模型在该题上的中位链长度 $\tilde{L}$，得到归一化链长度 $\hat{L} = L_t/\tilde{L}$。在此基础上拟合指数衰减模型

$$\bar{p} \sim e^{-\hat{L}/L_0+c}, \tag{4.9}$$

其中 L_0 为无量纲的特征长度，$\hat{L} = 1$ 对应该模型在该题上的典型链长度，拟合时排除最短链区间（第一个分箱点），因为该区间的分布特征与其余区间存在系统性差异（图中以半透明标出）。图 4.9展示了三个代表性模型的结果：归一化后的衰减趋势成立，表明尽管特征长度 L_0 因模型而异，但指数衰减的函数形式具有普适性。误差棒为每 bin 内的二项式标准误，阴影带为 1σ 拟合预测带。全部七个模型的完整分析见附录 D.2。

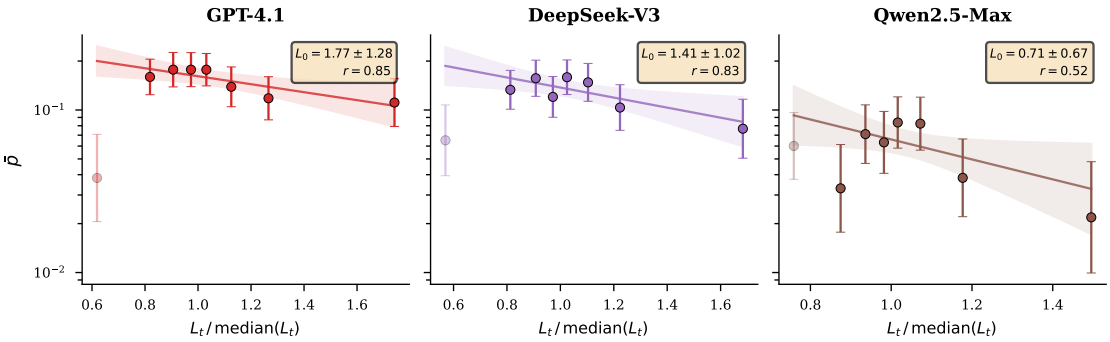

图 4.9　中位数归一化后的思维链长度随正确解答概率的指数衰减

工具集规模的最优权衡 在工具调用型智能体中，LLM 接收任务描述和可用工具列表，选择并调用合适的工具。这里 LLM 充当 wrapper：输入分布 π 对应于直接执行确定性 pipeline（不经 LLM 选择，按固定规则调用工具）的结果分布，输出 $\pi' = \mathcal{T}_{\text{LLM}}(\pi)$ 对应于经 LLM 选择工具后的结果分布。式 (4.8)的预测如下：当工具集较小时，π 的语义简单（可选工具少，组合复杂度低），LLM 能有效地将分布压缩和偏置向最优解，$J(\pi') > J(\pi)$，LLM 有正贡献；当工具数量增大时，π 的组合复杂性超出固定 LLM 的处理边界，LLM 开始在大量工具描述中迷失方向，将已经接近最优的分布展宽，$J(\pi') < J(\pi)$。因此，理论预测智能体性能随工具集规模可能呈非单调变化：先升后降，峰值位置由 LLM 的处理容量决定。为检验这一预测，本文在 PRBench[64]论文复现任务的简化版本上进行实验：固定任务目标，系统性地改变提供给智能体的工具集规模（从少量核心工具到大量冗余工具），分别测量三种模型的任务完成率。图 4.10的结果验证了非单调行为的存在，且随着模型能力的提升，峰值位置向更大规模工具集移动，与 LLM 处理容量增大的预期一致。

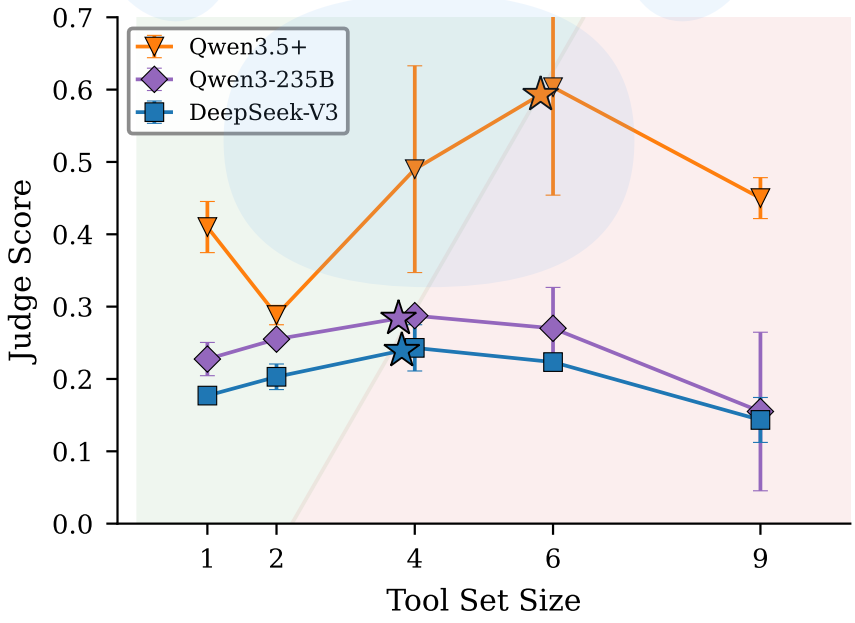

图 4.10 工具集规模与智能体性能的非单调关系

Liang 等人的实验同样观察到这一现象，他们认为这一机制在两个层面上发挥作用：在 prompt 层面，每增加一个工具，其模式和使用说明占据更多的上下文预算，挤压真正与任务相关的信息空间；在策略层面，更大的动作空间增加了工具选择的歧义性，使规划更加脆弱[75]。

4.2.2　多算子耦合与嵌套架构

公式 (4.8)的响应率约束适用于单一 LLM 通道的情形：一个固定的 LLM 算子作为 wrapper 处理基础策略集。然而，实际的智能体往往包含多个可独立缩放的 LLM 算子，这时响应率约束的适用范围需要重新审视。

多通道情形在实践中普遍存在。以式 (4.3)为例，如果原本固定的输入 π_0 随着预算 $\mathcal{B}$ 的增加而演化，即 $\pi_0 = \pi_0(\mathcal{B})$，则输入通道本身也具有独立的计算预算，pipeline 在数据处理中变为双通道系统。为分析一般多通道系统的响应率，可将每个通道的预算 $\mathcal{B}_i$ 视为独立变量，效用函数 J 依赖于所有通道的预算，即 $J = J(\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n)$。在这种情形下，拓展的响应率可以写为[22]：

$$\frac{dJ}{d\mathcal{B}_{\text{ref}}} = \sum_{i=1}^{n} \frac{\partial J}{\partial \mathcal{B}_i} \cdot \frac{d\mathcal{B}_i}{d\mathcal{B}_{\text{ref}}}, \tag{4.10}$$

其中 $\mathcal{B}_{\text{ref}}$ 是参考预算，架构设计者选择的缩放协议决定了当 $\mathcal{B}_{\text{ref}}$ 增大时各通道预算如何共同变化。

以图 4.11所示的生成-选择（generate-then-select）架构为例：生成器 LLM（Gen）产生 k 个候选解，选择器 LLM（Sel）从中挑选最优者。其中分数函数 s 被定义为输出解在 AIME 测试集上的正确率，预算 $\mathcal{B}$ 与模型规模相关。图 4.11展示了这一架构的算子图：其中 Gen 和 Sel 各自对应一个独立的预算通道 $\mathcal{B}_{\text{gen}}$ 和 $\mathcal{B}_{\text{sel}}$（模型规模或生成次数等）。这时，效用函数 $J = J(\mathcal{T}_{sel}(J_{\text{gen}}(\mathcal{B}_{\text{gen}})); \mathcal{B}_{\text{gen}})$，其中 $\mathcal{T}_{sel}$ 是在 $\pi \to J$ 投影上定义的选择器算子，J_{gen} 是直接通过多数投票（majority voting）从生成器输出的分布 π_{gen} 投影得到的效用函数。响应率约束 (4.8)适用于每个单独的通道，则代入公式 (4.10)得到：

$$\frac{dJ}{d\mathcal{B}_{\text{ref}}} = \frac{\partial J}{\partial \mathcal{B}_{\text{sel}}} \cdot \frac{d\mathcal{B}_{\text{sel}}}{d\mathcal{B}_{\text{ref}}} + \frac{\partial J}{\partial J_{\text{gen}}} \cdot \frac{dJ_{\text{gen}}}{d\mathcal{B}_{\text{gen}}} \cdot \frac{d\mathcal{B}_{\text{gen}}}{d\mathcal{B}_{\text{ref}}}. \tag{4.11}$$

式 (4.11)的右端第一项是选择器通道的贡献：当选择器固定（$d\mathcal{B}_{\text{sel}}/d\mathcal{B}_{\text{ref}} = 0$）时该项消失，$J$ 的增长完全由生成器驱动；第二项是生成器经选择器后的贡献：$\partial J/\partial J_{\text{gen}} \le 1$ 度量选择器对生成器性能预算增加的响应。当两个通道共同缩放时，两项的符号和大小决定了三种耦合体制（图 4.11下排）：(a) 解耦体制中仅 Gen 缩放，嵌套曲线与某条固定选择器配置的性能曲线重合；(b) 负耦合中共同缩放反而降低响应率；(c) 正耦合中共同缩放使嵌套曲线超越所有固定线的包络。在正耦合时，尽管选择器对生成器预算的敏感度受到限制，但通过同时提升选择器的预算，生成-选择架构对预算的总响应能够超过任何单一通道的响应上界。这正是 Snell 等人[59]所观察到的 test-time-scaling。

Song 和 Zhu[22]在 AIME 数据集上进一步验证了正耦合体制的存在。在该实验中，五种不同规模的 Qwen 系列模型（7B 至 ~200B）分别充当生成器和选择器。当生成器和选择器使用同一规模模型共同缩放时，性能曲线对模型规模的响应超过了不经选择

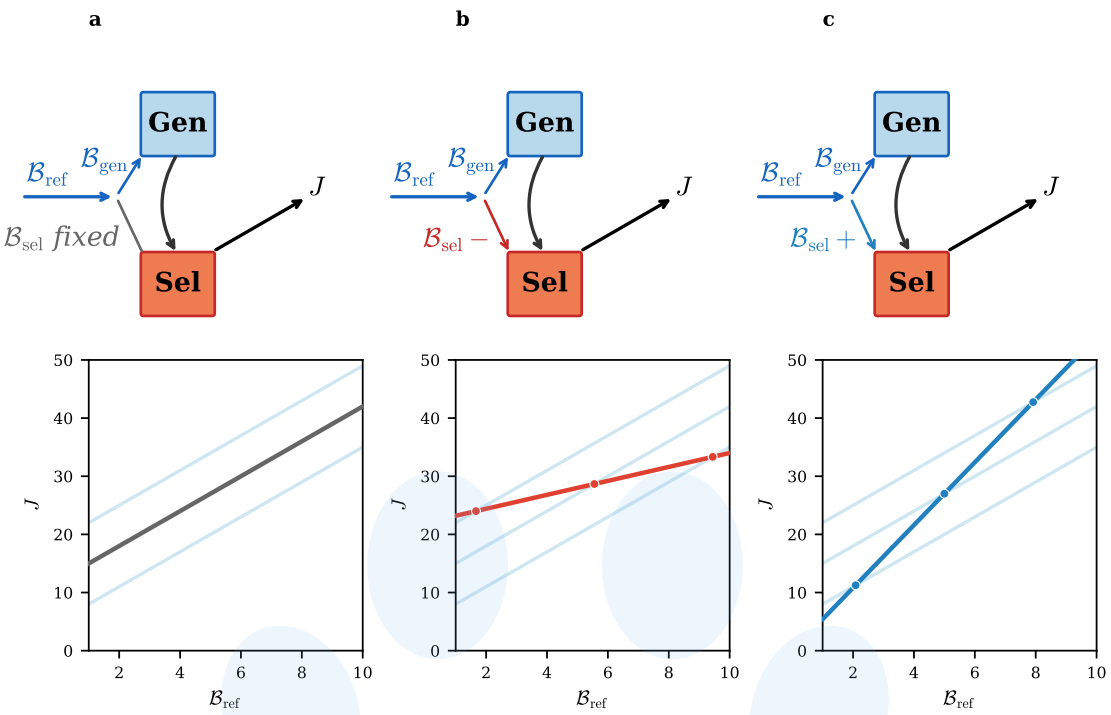

图 4.11　生成-选择架构的算子图与三种层间耦合体制[22]

的多数投票（majority voting）的性能，验证了正耦合的存在。这对智能体的自我演化有直接含义：开放式自我改进要求架构本身随智能体能力提升共同演化或至少允许同步提升架构中各组件的能力。

4.3　势能表示与细致平衡

线性表示从外部衡量了算子 pipeline 的性能边界，但在迭代智能体的场景下，这一框架遇到了根本性的困难。第4.2节的响应率分析依赖于将 pipeline 分解为有限个通道并逐通道施加约束，然而在迭代智能体中，环路算子 $\mathcal{A}$ 的 n 轮迭代对应 n 层嵌套，而 n 通常在数千以上。当嵌套层数与迭代次数同阶时，每个深层通道的响应率约束需要逐层传递到外层再求和，引入大量无法独立测量的中间量，使得式 (4.10)中的总响应率上界变得平凡，即不存在有意义的约束。换言之，线性表示虽然能有效地刻画固定架构或少数通道共同缩放的行为，但对于理解迭代环路内部的收敛性和能力边界是不充分的。

要理解描述迭代过程的环路算子 $\mathcal{A}$ 的行为，需要开发新的数学工具。为此本节从转移核的微观结构中提取一个势函数 V 作为宏观表示，使得稳态分布取得玻尔兹曼形式 $\pi_\infty(f) \propto e^{-\beta V(f)}$。势函数 V 直接刻画了环路内部的能量景观，从而规避了逐层嵌套分析的困难，为迭代智能体的收敛性和搜索时间复杂度提供了定量预测。本节首先建

立转移核的马尔可夫基础，然后展示 LLM 的生成过程满足细致平衡条件，从而自然地导出势函数 V 及其物理意义。

4.3.1　大语言模型生成过程的马尔可夫描述

一般的状态空间 $\mathcal{S}$ 包含异质的对象，如 prompt、API 返回值、代码片段，算子的输入与输出往往属于不同类型，转移过程通常是不可逆的。然而，在许多迭代智能体中，环路中的每个状态都属于同一类对象：IdeaSearchFitter 中每个状态是一条符号表达式[11]，FunSearch 中是一段程序[5]，条件词生成中是一个单词[23]。当状态空间可以限制到这样一个同质的子空间 $C \subseteq \mathcal{S}$ 时，输入与输出在 C 上可比，使得用一套统一的势能函数来描述转移过程成为可能。C 中一个状态的边界如何划定本身是一个开放问题，一种实用的判据可能是检验细致平衡的符合程度。

在 C 上，智能体的迭代生成可以形式化为马尔可夫转移过程。考虑一个以 LLM 为核心的智能体：它接收当前状态 $f \in C$ 作为输入，通过一系列确定性步骤组织和评估该状态以生成提示词，然后将提示词输入 LLM，解析 LLM 的输出得到新状态 $g \in C$。为简化推导，这里只考虑单通道输入输出的情形，这时转移核为[23]

$$\mathcal{T}(g \leftarrow f) = P(g|f), \tag{4.12}$$

即智能体从包含状态 f 的模板通过 LLM 生成转移到包含状态 g 的输出的概率。数据库的内容被纳入状态定义 f 中，从而保证转移核 $\mathcal{T}(g \leftarrow f)$ 仅依赖于当前状态。

需要强调的是，马尔可夫描述的对象是单步生成算子 $\mathcal{T}_{\text{eval}} \circ \mathcal{T}_2 \circ \mathcal{T}_1$，而非包含数据库的完整系统。数据库作为确定性的存储结构，其行为可以用传统方法分析，不需要额外的概率性理论支撑。马尔可夫近似也并非一个严苛的假设。在实际应用中，LLM 智能体系统通常运行远超上下文窗口的步数[5-6,11,13-14]，因此必须反复提炼和压缩历史信息以适应上下文窗口的限制，这恰好使无记忆近似成为对单步生成动力学的自然描述。图 4.12展示了这一形式化框架的示意图。

4.3.2　从能量基模型到细致平衡

LLM 的势能通过把其生成过程视作能量基模型（energy-based model）能够自然地得到[23,82]。考虑序列级别的玻尔兹曼分布 $p_\theta(y|x_f) \propto e^{-E_\theta(x_f,y)}$，其中 $E_\theta(x_f,y) = \sum_{t=1}^{T}\left[-\ell_\theta(y_t|x_f,y_{<t}) + \log\sum_v e^{\ell_\theta(v|x_f,y_{<t})}\right]$ 是词元序列级别的能量函数，ℓ_θ 为 logit，第一项为词元能量，第二项为每步的对数配分函数（熵项），$y = (y_1,\ldots,y_T)$ 是完整的词元

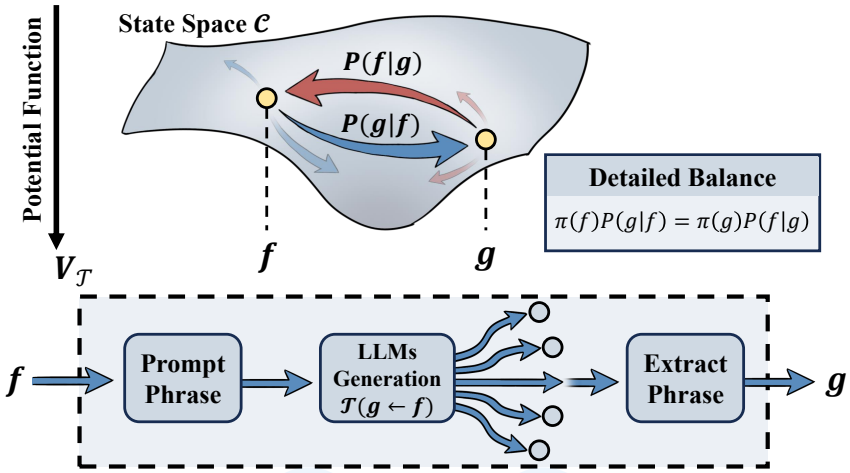

图 4.12　LLM 生成动力学的马尔可夫形式化框架[23]

序列。智能体的状态级转移核通过对所有实现状态 g 的词元序列求和得到：

$$\mathcal{T}(g \leftarrow f) = \sum_{y:\,\mathrm{state}(y)=g} \prod_{t=1}^{T} p_\theta(y_t|x_f, y_{<t}) = \frac{Z_\theta(x_f, g)}{Z_\theta(x_f)}, \tag{4.13}$$

其中 $Z_\theta(x_f, g)$ 定义为 $\sum_{y:\,\mathrm{state}(y)=g} e^{-E_\theta(x_f,y)}$，是限制在状态 g 的微观态扇区的配分函数，$Z_\theta(x_f) := \sum_y e^{-E_\theta(x_f,y)}$ 是总配分函数。这一过程将词元级别的微观动力学投影到语义状态空间上。

定义条件自由能 $F(g|f) := -\log Z_\theta(x_f, g)$，它吸收了从状态 f 出发到达状态 g 的所有词元序列的能量和熵贡献。F 的物理意义是状态间转移的“代价”：$F(g|f)$ 越低，从 f 生成 g 越容易。当两个方向的代价不同时，自由能差 $\Delta F = F(g|f) - F(f|g)$ 决定了生成的方向偏好。在最简单的单步生成中，即固定状态 f，F 退化为 $-\log p_g$，即单纯由生成概率决定，此时两个状态之间 $\Delta F = \ln(p_f/p_g)$；优势态具有低 F（容易被生成），非优势态具有高 F（难以被生成），ΔF 控制了两者之间的对比度。而在多步生成中，为了从 F 中提取出一个不依赖状态对的全局标量，即势函数 V，需要对 F 的结构施加约束。假设条件自由能的反对称部分 $A(f, g) := \frac{1}{2}[F(g|f) - F(f|g)]$ 是状态空间 C 上的恰当 1-形式（exact 1-form），即存在标量函数 $h : C \to \mathbb{R}$ 使得

$$F(g|f) - F(f|g) = h(f) - h(g), \forall f, g \in C. \tag{4.14}$$

这一条件要求沿状态空间中任意闭合路径的反对称自由能之和为零，于是排除了任何宏观概率流的循环。

由式 (4.13)，$\mathcal{T}(g \leftarrow f) = e^{-F(g|f)}/Z_\theta(x_f)$。1-形式假设 (4.14)保证了反对称自由能可以用标量函数 h 表达，这使得从转移核的微观结构中提取一个标量势函数成为可能。

定义势函数[23]

$$V(f) := -h(f) - \log Z_\theta(x_f), \tag{4.15}$$

其中 h 是 1-形式假设 (4.14)中的势函数，$Z_\theta(x_f)$ 是以状态 f 为条件的总配分函数。$V(f)$ 的物理意义是语义状态层面的宏观自由能：它通过对实现状态 f 的所有微观态（词元序列）求和而得到，将词元级的能量 E_θ 和熵贡献投影为单一的热力学标量。图 4.14展示了这一分辨率差异：(a) 在能量模型中，$E_\theta(x, \hat{y})$ 是逐词元定义的能量，推理通过逐词元能量最小化进行；(b) 势函数 $V(f)$ 则定义在语义闭合的状态 f 和 g 上，通过对实现给定状态的所有词元序列求和得到，中间的思维链推理词元不携带 V。

势函数 V 的定义使得转移核自然满足细致平衡条件[80,88]。对转移概率取对数比并利用式 (4.13)和 (4.14)，对所有状态对 (f, g) 直接得到[23]：

$$\log \frac{\mathcal{T}(g \leftarrow f)}{\mathcal{T}(f \leftarrow g)} = \beta V(f) - \beta V(g), \tag{4.16}$$

细致平衡要求转移概率的对数比恰好等于势函数之差，排除了状态空间中的概率流循环：沿任意闭合路径 $f_1 \to f_2 \to \cdots \to f_n \to f_1$，

$$\sum_{i=1}^{n} \log \frac{\mathcal{T}(f_{i+1} \leftarrow f_i)}{\mathcal{T}(f_i \leftarrow f_{i+1})} = \sum_{i=1}^{n} \beta \left(V(f_i) - V(f_{i+1}) \right) = 0. \tag{4.17}$$

这里的 β 是一个尺度因子，对于裸的 LLM 生成过程，$\beta = 1$，当转移矩阵 $\mathcal{T}$ 是通过将 LLM 嵌入一个智能体中定义的时候，β 可以被智能体的生成策略调制。例如，一个使用多数投票（majority voting）策略的 LLM 智能体中，如果每次生成 M 个候选状态，然后从这些候选状态中选择出现次数超过 $n (n > M/2)$ 次的状态作为输出状态，那么这个智能体的转移核具有 $\beta = n$，它的势能则与裸的 LLM 势能一样[23]。图 4.13展示了这一效应的实证验证（实验细节见附录 D.4）：利用第4.1.2节可约性实验中同一模型的生成分布数据，在正确状态 C 为优势态（$\Delta F \approx -0.90$）和非优势态（$\Delta F \approx 2.84$）两种情况下，命中率 P 随投票阈值 n 指数变化，衰减指数 b 由自由能差 $|\Delta F|$ 决定，与对 β 的理论预测符合。

多模型、多任务的实验验证支持了细致平衡这一理论预测[23]。图 4.15直观地展示了 LLM 生成的方向性：以 Claude Sonnet 4.5 模型在条件词生成任务中为例，将状态按势函数 βV 从低到高排列，箭头粗细正比于转移概率。从高势能状态（如 BUZZY、TURKEY）到低势能状态（如 ATTITUDE）的转移概率远大于反向转移，生成过程呈现系统性的向下漂移。这种定向漂移与状态对层面的细致平衡并不矛盾。细致平衡约束的是每一对状态之间的转移概率比 $\mathcal{T}(g \leftarrow f)/\mathcal{T}(f \leftarrow g) = e^{\beta[V(f)-V(g)]}$，而定向漂移是对整体转移统计的描述，二者作用在不同的分辨率上，因此相容。图 4.16进一步验证了细致平衡条件 (4.16)的定量预测：在 IdeaSearch 符号回归任务的 1372 对状态中，

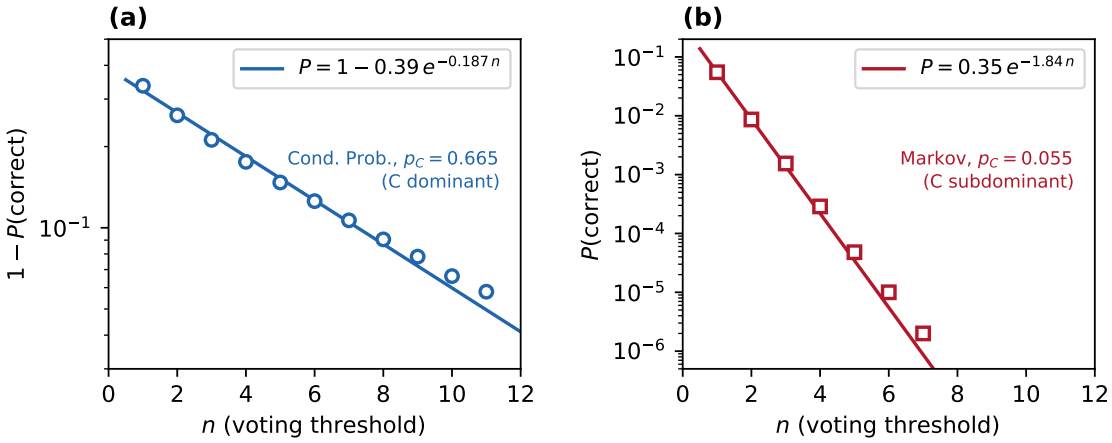

图 4.13　多数投票对生成分布的锐化效果

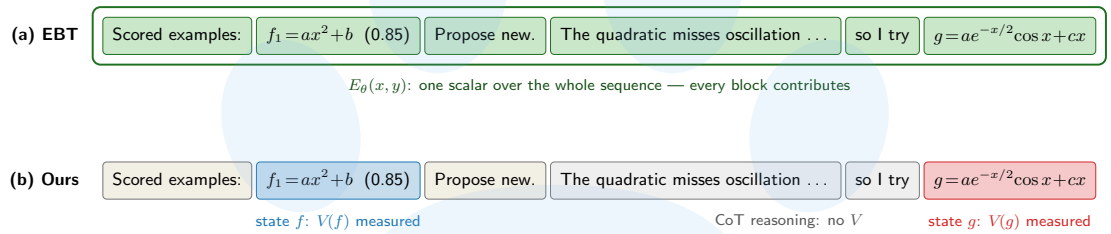

图 4.14　微观词元级能量与宏观状态级势函数的分辨率对比[23]

转移概率的对数比 $\log[\mathcal{T}(g \leftarrow f)/\mathcal{T}(f \leftarrow g)]$ 与势函数差 $\beta(V(f) - V(g))$ 高度线性相关（Pearson 相关系数 $r = 0.92$，$\chi^2/\text{ndf} = 0.83$）。

4.3.3　势函数的提取与物理意义

1-形式假设给出了势函数 V 的微观定义 (4.15)，但在实际调用中通常无法直接接触 LLM 的词元级内部结构。需要一种仅从可观测的转移概率 $\mathcal{T}(g \leftarrow f)$ 出发就能提取 V 的方法。为此定义作用量为全局平均违反程度[23]：

$$S_{\text{act}} = \int_{f \in C} \int_{g \in C} \mathcal{T}(g \leftarrow f)\, K(V(f) - V(g))\, Df\, Dg, \tag{4.18}$$

其中 $K(x) = \exp(-\beta x/2)$ 是描述转移对势函数排序违反程度的凸函数，β 为尺度因子。K 的具体形式并非唯一选择，Song 等人[23]的分析表明提取的势函数 V 对 K 的形式不敏感。最适合描述智能体生成方向性 $\mathcal{T}$ 的势函数 $V_{\mathcal{T}}$ 是使作用量 S_{act} 最小化的那个[89-90]，即满足变分原理

$$\delta S_{\text{act}} = 0. \tag{4.19}$$

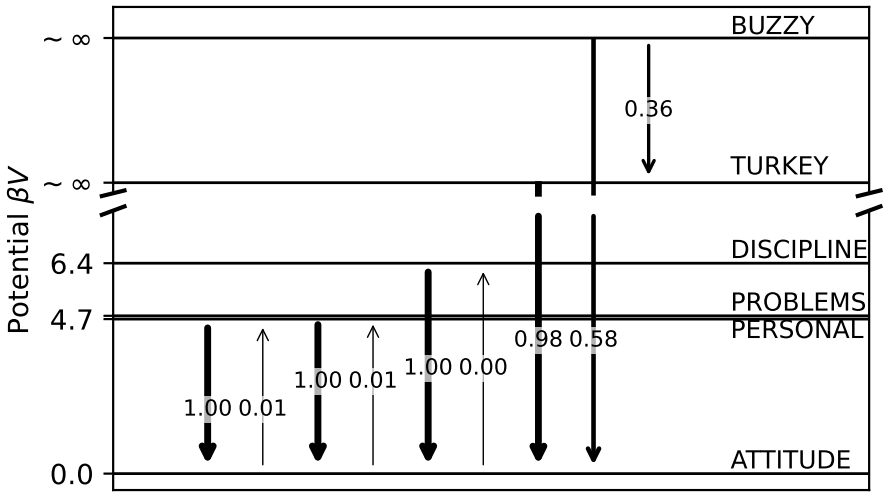

图 4.15　LLM 生成方向性的直观体现[23]

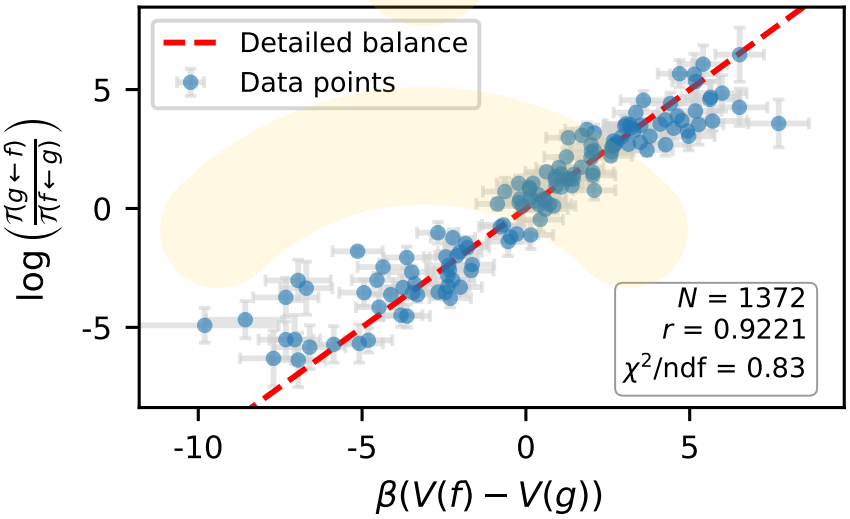

图 4.16　细致平衡条件的实验验证[23]

这一变分条件等价于 $V_{\mathcal{T}}$ 满足以下平衡条件：对所有 $f \in C$，

$$\int_g \mathcal{T}(g \leftarrow f) K'(V_{\mathcal{T}}(f) - V_{\mathcal{T}}(g)) Dg - \int_h \mathcal{T}(f \leftarrow h) K'(V_{\mathcal{T}}(h) - V_{\mathcal{T}}(f)) Dh = 0. \quad (4.20)$$

这正好是在公式 (4.15)中定义的势函数 V，二者最多相差一个常数。因此最小作用量原理提供了一条从转移概率数据中提取 LLM 势函数的路径，而无需了解 LLM 的内部结构，完整的数学论证见文献[23]。

由此提取的 V 具有明确的物理意义。细致平衡仅约束条件自由能的反对称部分。将 $F(g|f)$ 分解为对称部分 $S(f,g)$ 和反对称部分 $A(f,g)$，细致平衡要求反对称部分是恰当 1-形式：$A(f,g) = \frac{1}{2}[h(f) - h(g)]$。对称部分 $S(f,g)$ 编码了两个状态之间互相转

换的难度，完全不受细致平衡约束。回顾式 (4.15)，$V(f)$ 中的 $\log Z_\theta(x_f)$ 正是总配分函数的对数，而 $h(f)$ 吸收了 1-形式条件引入的标量势。最小作用量原理从转移概率数据中提取的 V 正是这个宏观自由能。

这一理论框架表明，尽管不同的 LLM 具有不同的内部结构和训练过程，它们在智能体的状态空间中都以相似的方式隐式地编码了一个通过状态空间上的势函数表示的自由能。这个自由能反映了 LLM 本身对状态“好坏”的全局认知。图 4.17展示了 IdeaSearch 符号回归任务中 70 个状态的转移子图：纵轴为势函数 $V(f)$，横轴为对应表达式的均方误差 $\log_{10}(\mathrm{MSE})$，绿色箭头表示势函数下降的转移，红色箭头表示上升的转移。约 80% 的高概率转移指向低势能状态，验证了势函数作为 LLM 全局认知代理的有效性。

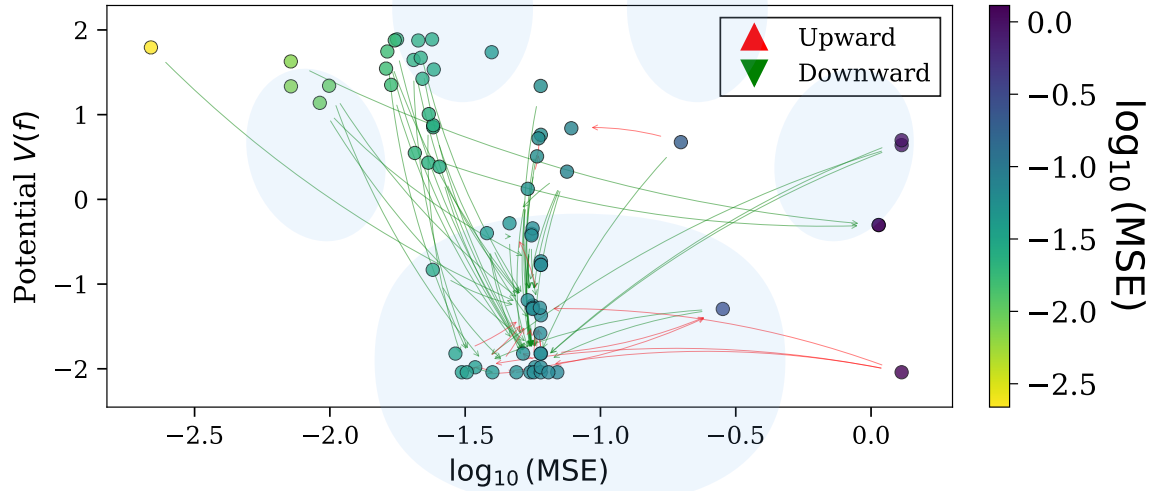

图 4.17　势函数对状态转移方向的预测[23]

4.3.4　评估器与搜索时间复杂度

回顾第4.1节的算子图，评估器算子的作用是用评分函数 $s(f)$ 给势函数景观加一层系统性偏置。它给每个状态 f 计算分数 $s(f)$（如 χ^2/ndf 的负值）并写进数据库；下一轮 LLM 读取这些历史分数时，高分状态更可能最终出现在 prompt 里，条件生成分布也随之被改写。

这一过程可以通过将裸势函数 $V_{\mathrm{bare}}(f)$（LLM 自身编码的全局认知）偏置为有效势函数理解：

$$V_{\mathrm{eff}}(f) = V_{\mathrm{bare}}(f) - \frac{s(f)}{T}. \tag{4.21}$$

T 是 IdeaSearchFitter 的工程配置参数，控制评分对采样的影响强度，与细致平衡中的理论参数 β 不同。评分与势函数直接耦合：$s(f)$ 越大，$V_{\mathrm{eff}}(f)$ 越低，状态在下一轮就

越容易被访问。这种偏置没有改变 LLM 的参数，而是控制 prompt 里的上下文，进而改变了有效能量景观。

裸 LLM 的 V_{bare} 对应模型预训练得到的通用认知，与具体任务的目标函数之间通常并不重合。评估器把任务特定的信号注入能量景观，引导搜索从 V_{bare} 的通用最低点漂移到 V_{eff} 的任务最优区域。在 IdeaSearch 里，这表现为搜索系统性地向低 χ^2/ndf 区域漂移。

评分偏置改变了有效势函数的景观结构，而这一结构直接影响了搜索的时间复杂度。对于第4.1节定义的环路算子 $\mathcal{A}$，稳态分布的玻尔兹曼形式决定了到达目标状态所需的搜索时间。势函数 V 的景观结构控制着这一时间尺度。对于满足细致平衡的可逆马尔可夫链，在封闭状态空间上假设遍历性成立（所有状态互通）且状态划分已给定，稳态分布为玻尔兹曼形式 $\pi_\infty(i) = e^{-\beta V(i)}/Z$，由 Kac 定理[91]，首次到达状态 i 的平均步数为 $\tau_i = Z \cdot e^{\beta V(i)}$。若 βV 在 N 个状态上近似高斯分布（均值 μ，标准差 σ），配分函数通过矩生成函数计算为 $Z = N \cdot e^{-\mu+\sigma^2/2}$，代入得

$$\ln(\tau/N) = \Delta(\beta V) + \frac{\sigma^2}{2}, \tag{4.22}$$

其中 $\Delta(\beta V) = \beta V_{\text{target}} - \mu$。这表明，达到特定目标状态的搜索时间 τ 相对于平均水平 N 的对数增长由两部分组成：第一部分 $\Delta(\beta V)$ 反映了目标状态在能量景观中的高度，即相对于平均水平的势能差距；第二部分 $\sigma^2/2$ 反映了景观的宽度，即熵壁垒。

图 4.18是在 IdeaSearch 符号回归任务的一个 $N = 50$ 状态子图上的模拟检验。（a）展示了四个不同难度目标（V 的第 25、50、75、90 分位数）的 $\ln(\tau/N)$ 随有效景观宽度 $\sigma(T)$ 的变化：曲线的垂直分离反映 ΔV 的贡献，上升斜率反映 σ 的贡献。（b）将全部 24 个数据点投影到势能壁垒 $\Delta V_{\text{eff}} + \sigma^2/2$ 上，线性拟合给出

$$\ln(\tau/N) = 1.5 + 0.2 \cdot \left(\Delta V_{\text{eff}} + \frac{\sigma^2}{2}\right), \quad r = 0.9. \tag{4.23}$$

景观宽度的改变是通过调整 IdeaSearch 的温度参数 T 实现的。拟合斜率低于 Kac 预测值 1，原因在于评估器的分数 $s(f)$ 只能约束输入状态的排序，无法直接控制 LLM 输出状态的分布，因此 s 对有效势函数 V_{eff} 的偏置弱于理论假设，使得 V_{eff} 系统性地平坦于真实势能景观。斜率的大小反映了 LLM 生成的语义继承性：如果 LLM 倾向于为高分输入产生语义相近的高分输出，则 s 的控制力更强，斜率更接近 1；反之，如果生成过程将输入的评分信息大幅打散，则斜率更低。尽管如此，势能壁垒对搜索时间复杂度的显著影响仍然清晰可见，验证了势函数景观结构对搜索效率的控制作用。

需要指出的是，这一模拟中所有可能的状态在实验开始前已经确定，转移核在这些状态之间定义。然而 LLM 的实际生成分布是开放的：它可以产生训练集之外的新表

达式，探索未曾见过的状态空间区域。这种开放性可能是探索新知识的关键。要完全理解智能体探索新知识的机制，需要发展更精细的理论工具来处理开放系统的动力学。

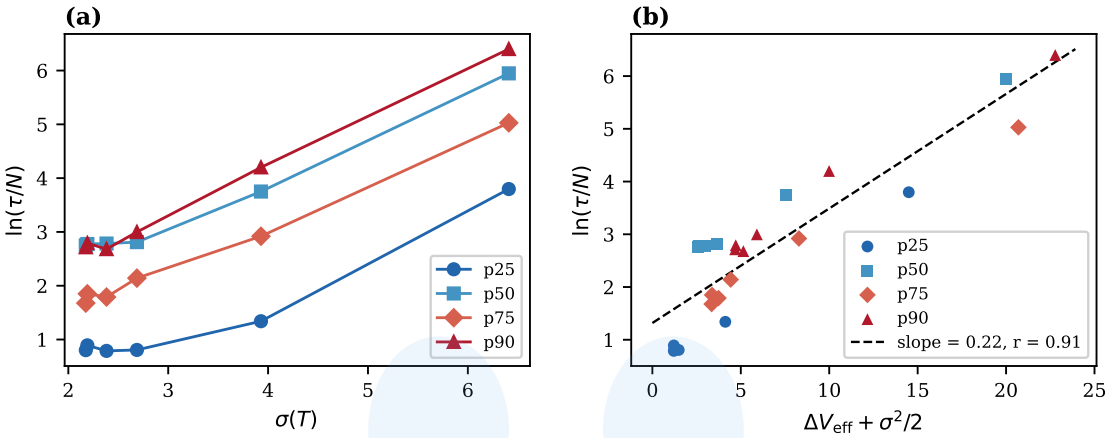

图 4.18　IdeaSearch 能量景观上的到达时间缩放

4.4　统一设计原理

通过算子图的语言，不同智能体架构的设计原则可以放到同一个框架里分析。至此，本文建立了两种不同的表示，以下讨论这些表示在智能体设计中的若干指导原则，并通过 IdeaSearchFitter 的进化过程实验展示理论框架的工程优化和定量分析能力。对于无环的浅层 pipeline，最简单的线性表示已经足够揭示核心约束。响应率约束 (4.8)表明固定的 LLM 算子不能充当计算资源的放大器：在高预算区间，直接增强基础策略（如更强的搜索、更好的提案生成、更可靠的验证）比依赖固定 LLM 来放大增益更为有效。这一约束直接指向一条设计原则：即 LLM 应当作为“薄判断层”，只做从输入到分布的映射判断，而将确定性的重计算交给工具算子。FunSearch[5]中 LLM 根据已有高分程序判断探索方向而评估函数承担验证任务，IdeaSearchFitter[11]中 LLM 进行语义层面的判断而数值拟合由外部评估器完成，IBP 约化优化[8]中 LLM 负责搜索最优的优先级函数而 Feynman 积分的约化由确定性的计算代数系统执行，Voyager[13]中 LLM 判断下一个应该学习的技能而具体的游戏交互由生成的代码在环境中执行。这些架构都采用了“薄判断层”的设计思想。当 pipeline 包含多个可变通道时，嵌套架构能突破单通道的响应率上界，正如公式 (4.10)中所揭示的。AlphaEvolve[6]集成 Gemini Flash 和 Gemini Pro 两种模型的设计，以及量子态制备中[10]哈密顿量复杂度与搜索配置的协同变化，都可以在这一框架下理解。

线性表示对浅层 pipeline 的设计是有用的约束，但放到迭代智能体的场景里就变得平凡（第4.3节）。势能表示从转移核的微观结构里提取出势函数 V，绕过了逐层分析

的麻烦，给出一系列非平凡的定量预测。V 为智能体的状态空间提供了全局排序，LLM 的生成过程表现为偏向低势能状态的概率漂移[23]，V 因此暴露出 LLM 内在认知与真实数据之间的差异。进一步地，势函数的景观结构控制了搜索的时间复杂度：由式 (4.22)，到达目标状态的平均搜索时间 τ 取决于势能差 ΔV 与景观宽度 σ。式 (4.21)中的温度 T 是相应的调控旋钮：T 增大，探索性变强但搜索时间也变长；T 减小，方向性更突出。这些预测并不止于定性原则，下面的工作流可以把它们落到定量执行。

以 IdeaSearchFitter 的核心循环为例，可以看出粗粒化、分数和势能这些理论概念如何支撑起智能体设计的定量分析和优化（完整配置和补充实验见附录 D.6）。IdeaSearchFitter 的核心循环被运行 10 次以采集转移数据，共得到 50288 次状态转移，耗时 58 小时。图 4.19(a) 展示从中估计的 5×5 粗粒化转移核 T_{gen}：向最低分区间转移的概率占主导，跨区间改善的概率不超过 1%。在 T_{gen} 之上构建模拟（详见附录 D.6），图 4.19(b) 展示 500 条模拟轨迹的 68% 和 95% 置信区间带，10 条实际实验轨迹（深蓝色）与模拟预测吻合，说明粗粒化马尔可夫模型确实有预测能力。轨迹在对数横轴上先快速上升，在 $\sim 10^3$ 次 LLM 调用后饱和，这一形态与 FunSearch[5]、IBP 约化优化[8]、IdeaSearch[11]和量子电路设计[10]中报告的分数演化一致，是迭代 LLM 搜索的普遍特征。转移核可以测量，设计参数的系统扫描也就成为可能：图 4.19(c) 在岛屿温度 T 与父代数量 n_{parents} 构成的二维空间上扫描首达时间（First Passage Time，FPT）[11]，定位最优配置。若按当前实验设置把图 4.19(c) 里每个配置都端到端跑一遍，需要约两个月并行运行 250 次 IdeaSearchFitter 的核心循环，并产生超过 5000$ 的 API 费用；而确定转移核之后，模拟在 5 分钟内就能完成全部参数扫描，这正是模拟方法相对于端到端试错的核心优势。

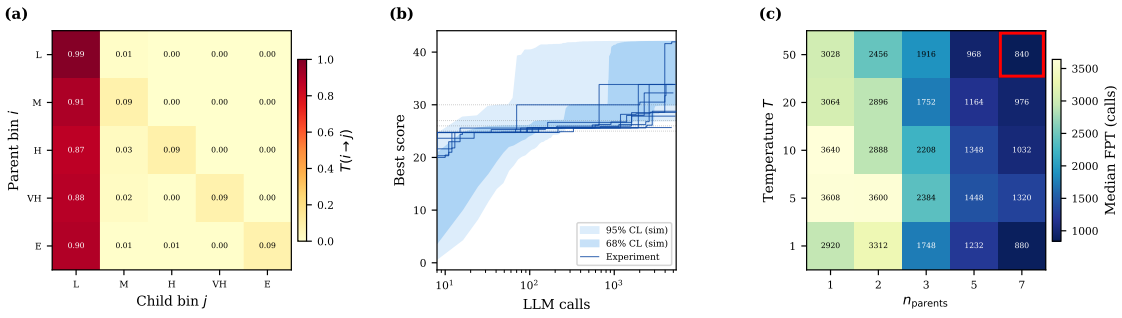

图 4.19　IdeaSearchFitter 进化过程的模拟与参数扫描

进一步地，势函数 V 的景观结构为改善概率提供了唯象理解。图 4.20(a) 展示了单次 LLM 调用后输出结果相对最优的输入改善的概率 $P_{\text{imp}}(s) = P(\text{child} > \text{parent} \mid \text{parent} = s)$ 随亲代分数 s 的变化：在低分区间 P_{imp} 近似平稳（~ 0.70），超过阈值 $s_0 \sim 21$ 后急剧下降。为了解释这种行为，可以假设势函数在通过分数划分状态时，势能随状态分数的变化并不单调，而是具有二次形式 $V = a(s - s^*)^2 + b$，由于跃迁概率由局域能

隙 $\propto |dV/ds|$ 控制，因此改善概率 P_{imp} 也呈现 logistic 衰减：在势能下降沿（$s < s^*$）改善概率高且平稳，在上升沿（$s > s^*$）随能隙增大而衰减。图 4.20(b) 展示了从转移核数据中提取的势函数 V，二次拟合的极值点与 P_{imp} 的 logistic 拐点位置接近，验证了这一唯象解释的合理性。实验者可据此从已有转移数据预测追加实验的改善概率。

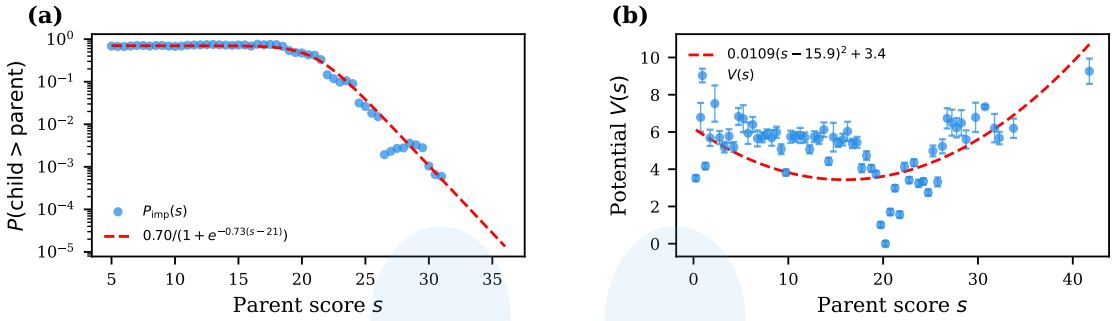

图 4.20　改善概率与势能景观的对应关系

4.5　规模定律与训练的物理视角

前三节的理论框架刻画了 LLM 在推理阶段的行为约束和在智能体层面的表现，但 LLM 本身的训练动力学和特征提取仍然是一个开放问题。在过去几年中，Scaling Laws 为 LLM 和智能体的设计提供了丰富的经验指导，并逐渐获得物理学层面的解释。理解这些经验规律如何融入一个完整的理论框架，是将本章的理论图像从推理时扩展到训练时的关键一步。

Scaling Laws 的研究始于 Kaplan 等人[57]对神经语言模型的经验观察：模型性能（以交叉熵损失衡量）与模型参数量、数据集大小和训练计算量之间存在幂律关系。Hoffmann 等人[58]进一步揭示了固定计算预算下参数量和数据量的最优分配规则。Snell 等人[59]将 Scaling Laws 的视角从训练时扩展到推理时，发现推理阶段增加计算量有时比增大模型参数更有效。Xiao 等人[61]提出了 LLM 的 Densing Law，定义 capacity density 为有效容量与实际参数量之比，发现其大约每三个月翻一倍，且这一趋势在不同架构和训练策略中保持稳定，区分了“增大系统”（增加参数量）和“改进系统”（提升单位参数的有效容量）两种性能提升途径。这些工作建立了 Scaling Laws 的基本经验框架，但幂律指数和相变行为仍缺乏第一性原理的解释。

近期的若干工作开始从物理学角度为 Scaling Laws 提供理论基础。Liu 等人[20]提出了神经热力学定律（Neural Thermodynamic Laws），将损失函数对应于自由能，数据集对应于热浴，训练步骤对应于弛豫过程，从而将 Scaling Laws 中观察到的幂律行为与临界指数联系起来。Maloney 等人[36]通过可解的随机特征模型揭示了幂律缩放与数据

频谱结构之间的联系，Sun 等人[19]的相变分析则将 LLM 的能力涌现与临界现象联系起来。Allen-Zhu 和 Li[60]从信息论的角度分析了模型容量的上界。在多智能体系统层面，Kim 等人[17]发现当多个智能体协作解决问题时，系统性能存在协调开销导致的亚线性区间和相变行为，表明多智能体系统可能像物理系统一样存在临界现象。

这些从不同角度出发的研究逐渐汇聚为一幅初步的物理图像，然而将这些经验描述与推理时的理论框架连接起来仍然是一个未来的重要挑战。

4.6　从定量约束到设计原理

本章从算子图出发，通过两种表示从 LLM 智能体的概率性生成中提取了定量规律：线性表示给出了响应率 χ 的上界约束，势能表示给出了细致平衡条件和搜索时间复杂度。两种表示的预测均已在多模型、多任务的实验中得到验证，并在 IdeaSearchFitter 的进化过程实验中展示了从转移核测量到设计参数优化的完整工作流。

回到本章开头提出的纲领视角：智能体的物理学并不是只研究一条端到端分数曲线，也不是只研究模型内部的 token 几何，而是要在二者之间建立可测量的桥梁。概率分布（系综）π 是这座桥梁的基本对象，线性表示和势能表示分别提供了从分布中提取宏观量的两种最直观的方法。对单步转移核 $\mathcal{T}(g \leftarrow f)$ 的测量是不可约实验，而智能体的宏观表现可以通过对各宏观量的分析约化为这些不可约实验的叠加。不可约实验为宏观行为的预测提供了稳健的实验基础，同时与词元级机制相联系，使得智能体的能力来源、约束与扩展可以在统一的理论框架下被讨论。

这也说明了理论与端到端实验的不同意义：理论分析的是投影后的抽象量（效用分数或势函数），端到端实验则产生系综中一个具体的词元序列；后者可能直接具有应用价值，而系综本身是用于检测智能体宏观表现的理论工具。IdeaSearchFitter 的实验提供了一个具体例证：少量端到端实验产出的转移数据，经粗粒化后可在数分钟内完成设计空间的全参数扫描，而势函数的景观结构则为改善概率的衰减行为提供了唯象解释。这一例子把宏观评测和设计目标重新接回了微观生成：向下，宏观量可以追溯到特定环境中 LLM 完整输出的系综；向上，这些宏观量进入仿真与参数扫描，并反馈到真实架构设计中。然而，若干关键环节，例如两种表示的统一与拓展、1-形式假设的推导、响应率假说的严格证明、训练与推理的衔接，仍然开放。这些问题的讨论将在第五章展开。

第五章　讨论与开放问题

第四章基于算子图的理论框架在 LLM 智能体的性能约束上给出了若干不平凡的定量预测，并在多个任务领域得到实证支持，但这一框架仍处于初步阶段，若干关键假设和开放问题还需要进一步讨论与验证。本章围绕这些假设和问题展开讨论，结合相关工作对照分析，以明确智能体的物理理论可能的发展方向。

粗粒化映射的选择　本文的理论框架依赖于粗粒化映射 φ 将高维文本状态空间投影为有限离散空间，但 φ 的选择目前是经验性的。在第四章的可约性实验中，四态映射 $\{C, NW, FW, NA\}$ 基于相对误差阈值手工设计，针对的是计算型任务的特定结构。不同的任务域可能需要完全不同的状态划分，目前没有一般性的原则来指导 φ 的构造。

因此，可约性是否成立同样是经验性的。式 (4.1)中的粗粒化核 T_{ij} 依赖于态内分布 ρ_i，只有当 T_{ij} 对 ρ_i 的变化不敏感时，矩阵乘法预测才可靠。第四章的实验在 216 条 pipeline 上验证了这一点，但这是特定 φ、特定任务域和特定模型下的经验结果，而非定理保证。原则上，过细的 φ 会使态内样本不足、核估计不可靠，过粗的 φ 则会将本质不同的文本归入同一态，破坏 T_{ij} 对 ρ_i 的独立性。

如何发展一套系统的粗粒化方法论，是推进这一框架的关键环节之一。统计物理中的重整化群（Renormalization Group）提供了一个概念上的类比：好的粗粒化应保留对宏观行为有决定性影响的自由度，而积掉不相关的微观细节，使得粗粒化后的描述在不同尺度上自洽[92]。在本文的语境下，这对应于寻找使 T_{ij} 最大程度独立于 ρ_i 的映射 φ。一种可能的形式化路径是将 φ 视为 $\mathcal{T}$ 的近似充分统计量，并通过最小化 T_{ij} 对 ρ_i 的变分来自动学习最优的状态划分。这一可能的方向旨在将粗粒化从经验设计推向可优化的数学对象，目的是为不同任务域提供统一的构造原则。

两种表示的关系　线性表示测量 $\mathcal{T}$ 对资源输入的响应特性，势能表示测量 $\mathcal{T}$ 在状态空间上的结构特征。两者作用于同一转移核，一个刻画“投入更多资源能得到多少额外性能”，一个刻画“状态之间转移的内在偏好”。

两类问题在统计物理中有成熟的对应。Kubo[93]建立的涨落-耗散定理连接了系统内禀的涨落谱与对外部扰动的线性响应，两者都是同一哈密顿量的不同侧面，定理给出了从涨落预测响应的精确公式。Wolpert 和 Macready[94]的 no-free-lunch 定理约束了任何固定映射在所有问题上的平均表现，与响应率约束 $\langle\chi(\pi')\rangle \le \langle\chi(\pi)\rangle$ 在结构上相似，尽管前者约束的是所有问题上的平均表现，后者约束的是特定分布的响应梯度，两

者都表明固定映射的性能改善存在不可逾越的上界。

目前的挑战在于，涨落-耗散定理的推导依赖于哈密顿量的具体形式和平衡态假设，而 LLM 中承担哈密顿量角色的物理量，尽管可能是势能 V，但其具体定义和性质尚不清晰。另一方面，Cover 和 Thomas[95]的数据处理不等式（DPI）给出了信息论的上界：信息通过固定信道后不会增加。但 DPI 约束的是互信息 $I(X;Y)$ 而非效用函数 J。从信息约束到性能约束的映射本身是核心难点：Shannon 定理给出信道容量上界，响应率给出性能对预算的响应上界，两者类比但并不等价。互补的表示暗示着这两条线索有望在未来得到统一，从而建立从转移核的结构特征到性能约束的可推导关系。

1-形式假设的物理根源 势能表示的核心前提是条件自由能的反对称部分构成状态空间上的恰当 1-形式（式 (4.14)）。该假设在 Song 等人[23]的实证分析中获得了广泛支持（代表性结果见图 4.16），但其物理根源尚待阐明。1-形式假设与统计力学中的 Kolmogorov 可逆性条件有形式类比：后者要求沿闭合路径的概率流之和为零，这正是式 (4.17)所表达的条件。满足此条件时，系统的长程行为由 V 的景观结构决定，这为理论分析提供了有力的工具。在 Kolmogorov 条件中，可逆性是物理系统平衡态的推论；在 LLM 生成中，可逆性是否是训练目标（最大化条件概率）的必然结果，或是自然语言语义空间本身的“保守性”，目前无法判定。跨模型架构、跨训练目标、跨语言和跨模态的比较研究将有助于揭示 1-形式假设的物理根源。

作为对照，Blondel 等人[82]建立了自回归语言模型与能量基模型之间的数学等价性，证明了自回归模型的词元级 logit 累积隐式定义了序列级能量函数：

$$E(y_{1:T}) = -\sum_t s_t(y_t|y_{<t}),$$

与本文式 (4.13)中的粗粒化过程在形式上具有一致性。但 Blondel 等人的框架停留在词元序列层面，未将能量函数投影到语义状态空间，因而未达到势函数 V 和细致平衡条件。1-形式假设恰好填补了这一间隙：它将反对称自由能投影为状态空间的标量函数，使得从微观能量到宏观势函数的映射成为可能。

响应率假说的严格证明 响应率假说目前以经验假说形式提出，尽管得到了 Song 和 Zhu[22]在多个任务领域、多个模型规模上的实证支持，并在第四章的一系列问题中做出了非平凡的定量预测，但其严格证明仍是一个重要的开放问题。DPI 提供了直观动机，但 DPI 约束的是互信息 $I(X;Y)$，即接收端对发送端信息的不确定性减少，而响应率约束的是效用函数 J 对预算 $\mathcal{B}$ 的梯度。两者的数学对象表面上不同：前者是概率分布之间的泛函，后者是性能分数对预算变量的导数。

一种可能的证明路径是：首先建立 J 的信息论解释，将“性能”映射为可比较的

信息量，使 DPI 的上界可以直接传递到 χ。Cover 和 Thomas[95]中关于互信息与决策误差的关系（Fano's inequality）提供了起点：当互信息 I 有上界时，决策误差也有对应的下界，这暗示了从信息约束到性能约束的可推导性。其次在可解模型（线性化注意力机制、Maloney 等人[36]的随机特征模型）中进行精确推导以揭示一般结构。最后分析预算分配策略和饱和行为如何进入约束。这条路径一旦走通，将为响应率假说提供严格的数学基础，并可能揭示更一般的智能体性能约束。

训练与推理的衔接　第四章将 LLM 视为固定算子，但 V 的景观结构和 χ 的响应特性很可能由训练过程决定。

Liu 等人[20]的神经热力学定律将训练动力学纳入热力学语言：损失函数对应自由能，数据集对应热浴，训练步骤对应弛豫过程。Maloney 等人[36]通过可解模型揭示了幂律缩放与数据频谱结构的联系。尽管这些是训练侧的物理理论，而本文是推理侧的物理理论，公式 (4.15)已经揭示了在推理侧的宏观势函数 V 可以通过训练侧的微观能量函数 E 进行粗粒化得到，这为两者的衔接提供了数学入口。进一步的研究有望将 RLHF 等训练方法引入的偏好信号形式化为对 V 的特定偏置，从而能够为训练方法的设计提供理论指导。更一般地，如果将多轮训练—推理—评估的闭环视为 LLM 层面的宏观动力学，则本文的分布框架可能为刻画这种跨尺度的反馈回路提供数学语言。

多智能体协作中的集体效应　Kim 等人[17]发现多智能体系统性能随智能体数量并非单调增长，存在亚线性区间和相变行为。Hogg 等人[96]的研究指出搜索问题本身可以呈现相变：当问题复杂度超过临界值，搜索效率急剧下降。这两个结果在统计物理中有自然的对应，多体系统的集体行为由耦合强度决定。

放回算子图的语言里，多个 $\mathcal{A}$ 算子组合在一起，在状态空间上引入有效耦合。第三章的实践已经给出两端的样本：Voyager 跨世界复用技能库属于弱耦合的重用，AI 小镇按角色分工则属于强耦合的协作。不同耦合体制下的性能特征还需要系统刻画，工程上的意义却已经明确：理解这些特征能够在部署之前，靠小规模测量预判一次协作会带来增益还是开销。

框架的适用范围　算子图方法论本身并不限于 LLM；只要系统在状态空间上产生概率性转移，就可以形式化成转移核。真正重要的是 1-形式假设和响应率假说的适用范围：若两者在非 LLM 系统中不成立，这些约束就属于 LLM 特有的物理规律；若在更广的智能体系统里依然成立，那么这些约束可能反映了更一般的智能体原理。

进化算法中的固定变异算子是否也有类似响应率约束？直观上，即使变异算子不随种群规模缩放，增加种群规模带来的边际收益也可能通过集体效应被变异放大，这

与 $\langle\chi(\pi')\rangle \leq \langle\chi(\pi)\rangle$ 的结构相似。RL 策略网络是否满足某种细致平衡？policy gradient 的更新是方向性的而非对称的，细致平衡在此情形下不应成立。通过实证分析进行比较将有助于区分 LLM 特有的约束与更一般的智能体原理。

理论的发展方向 当前的理论框架仍然处于经验理论阶段，然而一个有清晰结构、有可验证预测、有明确发展方向的研究纲领正在形成：通过在两种表示之间建立联系，并对其核心假设进行定理化，智能体的性能约束有望从经验观察过渡到形式理论。建立从训练侧到推理侧的衔接，将为理论提供更丰富的实用空间，并为训练方法的设计提供理论指导。这一纲领的每一步都具有挑战性，但也都具有明确的路径和潜在的突破点。

第六章　结论

本文旨在为智能体设计提出一套以物理概念为基础的理论研究纲领。在实践上，随着人工智能，尤其是 LLM 领域的飞速发展，智能体系统的设计经验日益丰富，并在包括科学研究在内的一系列领域取得了显著成果。然而，这些实践上的努力往往是经验性的，缺乏一个系统的理解与复用的方法。另一方面，虽然对 LLM 的微观机制（如注意力机制、训练动力学等）的理论研究非常活跃，但这些研究尚未与智能体层面的宏观行为建立起可追溯的联系。在这样的背景下，智能体的能力来源、受何约束以及如何扩展等宏观设计问题仍然缺乏原理层面的解释。

本文首先回顾了从神经网络基础到智能体应用的完整技术背景，考察了不同设计范式下智能体的能力表现与共性趋势。在此基础上，本文引入算子图作为统一的形式化语言，将不同智能体架构纳入同一框架描述，并将概率分布投影到不同的宏观表示上以提取可测量量，分别以线性表示和势能表示两种投影方式为这些架构的能力极限做出了定量陈述。两种表示的预测均已在多模型、多任务的实验中获得验证，并指向了可操作的设计原则，包括薄判断层架构、嵌套耦合体制的选择和温度对搜索方向性的调控。这表明这一统一的理论框架已经具备了一定的指导能力。

这一纲领的核心在于将 LLM 生成的词元序列分布视为统计物理意义上的系综，以物理学的手段从系综中提取不随生成随机性变化的宏观可测量量。线性表示和势能表示是从系综中提取宏观量的两种最直观的投影方式，前者将分布线性投影为性能分数，后者借用统计物理中的自由能方法将分布投影为势函数。在这一视角下，对单步转移核的系综测量构成不可约实验，连接了词元级的微观动力学、特定智能体环境与智能体的宏观表现，而智能体整体的宏观表现可以约化为这些不可约实验的组合。由此，理论不再只描述端到端分数曲线，也不只停留在模型内部机制，而是给出一条从微观生成到宏观设计的可测量路径。

这一纲领仍处于初步阶段。两种表示的深层联系、响应率假说的严格证明、1-形式假设的物理根源以及训练动力学与推理约束的衔接，是当前面临的主要开放问题，解决这些问题将标志着这一框架从经验描述向公理化的转变。尽管存在这些挑战，一个有清晰结构、有可验证预测、有明确发展路径的研究纲领正在形成。从经验观察过渡到形式理论，从推理侧扩展到训练侧，这一纲领的推进有望形成完整的物理闭环：宏观量向下追溯到特定环境中 LLM 完整输出的系综，向上进入仿真、评测和参数扫描，并最终反馈到真实的架构设计中。这为智能体设计提供了超越反复试错的原理基础，也为理解智能系统的能力边界开辟一条以物理概念指导的探索路径。

参考文献

[1] YAO S, ZHAO J, YU D, et al. ReAct: Synergizing Reasoning and Acting in Language Models[EB/OL]. 2023. https://arxiv.org/abs/2210.03629. arXiv: 2210.03629 [cs.CL].

[2] WANG X, WEI J, SCHUURMANS D, et al. Self-Consistency Improves Chain of Thought Reasoning in Language Models[EB/OL]. 2023. https://arxiv.org/abs/2203.11171. arXiv: 2203.11171 [cs.CL].

[3] SCHICK T, DWIVEDI-YU J, DESSÌ R, et al. Toolformer: Language Models Can Teach Themselves to Use Tools[EB/OL]. 2023. https://arxiv.org/abs/2302.04761. arXiv: 2302.04761 [cs.CL].

[4] YAO S, YU D, ZHAO J, et al. Tree of Thoughts: Deliberate Problem Solving with Large Language Models[EB/OL]. 2023. https://arxiv.org/abs/2305.10601. arXiv: 2305.10601 [cs.CL].

[5] ROMERA-PAREDES B, BAREKATAIN M, NOVIKOV A, et al. Mathematical discoveries from program search with large language models[J/OL]. Nature, 2024, 625(7995): 468-475. https://doi.org/10.1038/s41586-023-06924-6. DOI: 10.1038/s41586-023-06924-6.

[6] NOVIKOV A, VŨ N, EISENBERGER M, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery[EB/OL]. 2025. https://arxiv.org/abs/2506.13131. arXiv: 2506.13131 [cs.AI].

[7] HUBERT T, MEHTA R, SARTRAN L, et al. Olympiad-level formal mathematical reasoning with reinforcement learning[J/OL]. Nature, 2026, 651(8106): 607-613. https://doi.org/10.1038/s41586-025-09833-y. DOI: 10.1038/s41586-025-09833-y.

[8] SONG Z Y, YANG T Z, CAO Q H, et al. Explainable AI-assisted optimization for Feynman integral reduction[J/OL]. Journal of High Energy Physics, 2026. arXiv: 2502.09544 [hep-ph]. https://arxiv.org/abs/2502.09544.

[9] BOIKO D A, MACKNIGHT R, KLINE B, et al. Autonomous chemical research with large language models[J/OL]. Nature, 2023, 624(7992): 570-578. https://doi.org/10.1038/s41586-023-06792-0. DOI: 10.1038/s41586-023-06792-0.

[10] CAO Q H, HOU Z Y, LI Y Y, et al. Scalable Quantum State Preparation via Large-Language-Model-Driven Discovery[EB/OL]. 2025. http://arxiv.org/abs/2505.06347. arXiv: 2505.06347 [quant-ph].

[11] SONG Z Y, CAI Z, ZHANG S, et al. Iterated Agent for Symbolic Regression[EB/OL]. 2025. http://arxiv.org/abs/2510.08317. arXiv: 2510.08317 [physics.comp-ph].

[12] YANG J, JIMENEZ C E, WETTIG A, et al. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering[EB/OL]. 2024. https://arxiv.org/abs/2405.15793. arXiv: 2405.15793 [cs.SE].

[13] WANG G, XIE Y, JIANG Y, et al. Voyager: An Open-Ended Embodied Agent with Large Language Models[EB/OL]. 2023. https://arxiv.org/abs/2305.16291. arXiv: 2305.16291 [cs.AI].

[14] PARK J S, O'BRIEN J C, CAI C J, et al. Generative Agents: Interactive Simulacra of Human Behavior[EB/OL]. 2023. https://arxiv.org/abs/2304.03442. arXiv: 2304.03442 [cs.HC].

[15] LIANG J, HUANG W, XIA F, et al. Code as Policies: Language Model Programs for Embodied Control[EB/OL]. 2023. https://arxiv.org/abs/2209.07753. arXiv: 2209.07753 [cs.RO].

[16] SAPKOTA R, ROUMELIOTIS K I, KARKEE M. AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges[J/OL]. Information Fusion, 2026, 126: 103599. https://www.sciencedirect.com/science/article/pii/S1566253525006712. DOI: https://doi.org/10.1016/j.inffus.2025.103599.

[17] KIM Y, GU K, PARK C, et al. Towards a Science of Scaling Agent Systems[EB/OL]. 2026. https://arxiv.org/abs/2512.08296. arXiv: 2512.08296 [cs.AI].

[18] SONG Z Y, LI Z, CAO Q H, et al. Bridging the Dimensional Chasm: Uncover Layer-wise Dimensional Reduction in Transformers through Token Correlation[EB/OL]. 2025. http://arxiv.org/abs/2503.22547. arXiv: 2503.22547 [cs.CL].

[19] SUN Y, HAGHIGHAT B. Phase Transitions in Large Language Models and the $O(N)$ Model[EB/OL]. 2025. https://arxiv.org/abs/2501.16241. arXiv: 2501.16241 [cs.LG].

[20] LIU Z, LIU Y, GORE J, et al. Neural Thermodynamic Laws for Large Language Model Training[EB/OL]. 2025. https://arxiv.org/abs/2505.10559. arXiv: 2505.10559 [cs.LG].

[21] CARNOT S. Réflexions sur la puissance motrice du feu et sur les machines propres à développer cette puissance[M/OL]. Bachelier, 1824. https://gallica.bnf.fr/ark:/12148/bpt6k29063f.

[22] SONG Z Y, ZHU H X. A Theory of LLM Information Susceptibility[EB/OL]. 2026. http://arxiv.org/abs/2603.23626. arXiv: 2603.23626 [cs.LG].

[23] SONG Z Y, CAO Q H, LUO M X, et al. Detailed balance in large language model-driven agents[EB/OL]. 2025. http://arxiv.org/abs/2512.10047. arXiv: 2512.10047 [cs.LG].

[24] LIU Z, WANG Y, VAIDYA S, et al. KAN: Kolmogorov-Arnold Networks[EB/OL]. 2025. https://arxiv.org/abs/2404.19756. arXiv: 2404.19756 [cs.LG].

[25] LIU Z, MA P, WANG Y, et al. KAN 2.0: Kolmogorov-Arnold Networks Meet Science[EB/OL]. 2024. https://arxiv.org/abs/2408.10205. arXiv: 2408.10205 [cs.LG].

[26] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators[J/OL]. Neural Networks, 1989, 2(5): 359-366. https://doi.org/10.1016/0893-6080(89)90020-8. DOI: 10.1016/0893-6080(89)90020-8.

[27] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J/OL]. Nature, 1986, 323(6088): 533-536. https://doi.org/10.1038/323533a0. DOI: 10.1038/323533a0.

[28] NAIR V, HINTON G E. Rectified Linear Units Improve Restricted Boltzmann Machines[C/OL]//Proceedings of the 27th International Conference on Machine Learning (ICML). 2010: 807-814. https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf.

[29] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J/OL]. Proceedings of the IEEE, 1998, 86(11): 2278-2324. https://doi.org/10.1109/5.726791. DOI: 10.1109/5.726791.

[30] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J/OL]. Neural Computation, 1997, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735. DOI: 10.1162/neco.1997.9.8.1735.

[31] CHUNG J, GULCEHRE C, CHO K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[EB/OL]. 2014. https://arxiv.org/abs/1412.3555. arXiv: 1412.3555 [cs.NE].

[32] PFAU D, SPENCER J S, MATTHEWS A G D G, et al. Ab initio solution of the many-electron Schrödinger equation with deep neural networks[J/OL]. Phys. Rev. Res., 2020, 2: 033429. https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429. DOI: 10.1103/PhysRevResearch.2.033429.

[33] UDRESCU S M, TAN A, FENG J, et al. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity[C/OL]//LAROCHELLE H, RANZATO M, HADSELL R, et al. Advances in Neural Information Processing Systems: vol. 33. Curran Associates, Inc., 2020: 4860-4871. https://proceedings.neurips.cc/paper_files/paper/2020/file/33a854e247155d590883b93bca53848a-Paper.pdf.

[34] LIU Z, TEGMARK M. Machine Learning Hidden Symmetries[J/OL]. Phys. Rev. Lett., 2022, 128: 180201. https://link.aps.org/doi/10.1103/PhysRevLett.128.180201. DOI: 10.1103/PhysRevLett.128.180201.

[35] NEYSHABUR B. Implicit Regularization in Deep Learning[EB/OL]. 2017. https://arxiv.org/abs/1709.01953. arXiv: 1709.01953 [cs.LG].

[36] MALONEY A, ROBERTS D A, SULLY J. A Solvable Model of Neural Scaling Laws[EB/OL]. 2022. https://arxiv.org/abs/2210.16859. arXiv: 2210.16859 [cs.LG].

[37] QIAN N. On the momentum term in gradient descent learning algorithms[J/OL]. Neural Networks, 1999, 12(1): 145-151. https://doi.org/10.1016/S0893-6080(98)00116-6. DOI: 10.1016/S0893-6080(98)00116-6.

[38] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[EB/OL]. 2017. https://arxiv.org/abs/1412.6980. arXiv: 1412.6980 [cs.LG].

[39] OPPERMAN C P, BOSMAN A S, MALAN K M. Regularisation in neural networks: a survey and empirical analysis of approaches[J/OL]. IEEE Transactions on Artificial Intelligence, 2025: 1-15. https://doi.org/10.1109/tai.2025.3644334. DOI: 10.1109/tai.2025.3644334.

[40] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with Deep Reinforcement Learning[EB/OL]. 2013. https://arxiv.org/abs/1312.5602. arXiv: 1312.5602 [cs.LG].

[41] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[EB/OL]. 2023. https://arxiv.org/abs/1706.03762. arXiv: 1706.03762 [cs.CL].

[42] VALERIANI L, DOIMO D, CUTURELLO F, et al. The geometry of hidden representations of large transformer models[EB/OL]. 2023. https://arxiv.org/abs/2302.00294. arXiv: 2302.00294 [cs.LG].

[43] HOPFIELD J J. Neural networks and physical systems with emergent collective computational abilities[J/OL]. Proceedings of the National Academy of Sciences, 1982, 79(8): 2554-2558. https://doi.org/10.1073/pnas.79.8.2554. DOI: 10.1073/pnas.79.8.2554.

[44] COHEN U, CHUNG S, LEE D D, et al. Separability and geometry of object manifolds in deep neural networks[J/OL]. Nature Communications, 2020, 11(1): 746. https://doi.org/10.1038/s41467-020-14578-5. DOI: 10.1038/s41467-020-14578-5.

[45] ELHAGE N, NANDA N, OLSSON C, et al. A Mathematical Framework for Transformer Circuits[J]. Transformer Circuits Thread, 2021.

[46] OLSSON C, ELHAGE N, NANDA N, et al. In-context Learning and Induction Heads[J]. Transformer Circuits Thread, 2022.

[47] BRICKEN T, TEMPLETON A, BATSON J, et al. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning[J]. Transformer Circuits Thread, 2023.

[48] KUMAR T, BORDELON B, GERSHMAN S J, et al. Grokking as the transition from lazy to rich training dynamics[C/OL]//The Twelfth International Conference on Learning Representations. 2024. https://openreview.net/forum?id=vt5mnLVIVo.

[49] WEI J, TAY Y, BOMMASANI R, et al. Emergent Abilities of Large Language Models[EB/OL]. 2022. https://arxiv.org/abs/2206.07682. arXiv: 2206.07682 [cs.CL].

[50] CHRISTIANO P, LEIKE J, BROWN T B, et al. Deep reinforcement learning from human preferences[EB/OL]. 2023. https://arxiv.org/abs/1706.03741. arXiv: 1706.03741 [stat.ML].

[51] ALLEN-ZHU Z, LI Y. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction[EB/OL]. 2024. https://arxiv.org/abs/2309.14316. arXiv: 2309.14316 [cs.CL].

[52] BAI Y, KADAVATH S, KUNDU S, et al. Constitutional AI: Harmlessness from AI Feedback[EB/OL]. 2022. https://arxiv.org/abs/2212.08073. arXiv: 2212.08073 [cs.CL].

[53] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal Policy Optimization Algorithms[EB/OL]. 2017. https://arxiv.org/abs/1707.06347. arXiv: 1707.06347 [cs.LG].

[54] WEI J, WANG X, SCHUURMANS D, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models[EB/OL]. 2023. https://arxiv.org/abs/2201.11903. arXiv: 2201.11903 [cs.CL].

[55] LIGHTMAN H, KOSARAJU V, BURDA Y, et al. Let's Verify Step by Step[EB/OL]. 2023. https://arxiv.org/abs/2305.20050. arXiv: 2305.20050 [cs.LG].

[56] GUO D, YANG D, ZHANG H, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning[J/OL]. Nature, 2025, 645: 633-638. https://doi.org/10.1038/s41586-025-09422-z. DOI: 10.1038/s41586-025-09422-z.

[57] KAPLAN J, MCCANDLISH S, HENIGHAN T, et al. Scaling Laws for Neural Language Models[EB/OL]. 2020. https://arxiv.org/abs/2001.08361. arXiv: 2001.08361 [cs.LG].

[58] HOFFMANN J, BORGEAUD S, MENSCH A, et al. Training Compute-Optimal Large Language Models[EB/OL]. 2022. https://arxiv.org/abs/2203.15556. arXiv: 2203.15556 [cs.CL].

[59] SNELL C, LEE J, XU K, et al. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters[EB/OL]. 2024. https://arxiv.org/abs/2408.03314. arXiv: 2408.03314 [cs.LG].

[60] ALLEN-ZHU Z, LI Y. Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws[EB/OL]. 2024. https://arxiv.org/abs/2404.05405. arXiv: 2404.05405 [cs.CL].

[61] XIAO C, CAI J, ZHAO W, et al. Densing Law of LLMs[EB/OL]. 2024. https://arxiv.org/abs/2412.04315. arXiv: 2412.04315 [cs.AI].

[62] HENDRYCKS D, BURNS C, BASART S, et al. Measuring Massive Multitask Language Understanding[EB/OL]. 2021. https://arxiv.org/abs/2009.03300. arXiv: 2009.03300 [cs.CY].

[63] QIU S, GUO S, SONG Z Y, et al. PHYBench: Holistic Evaluation of Physical Perception and Reasoning in Large Language Models[C/OL]//The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2026. https://openreview.net/forum?id=brG8FPq1cf.

[64] QIU S, DENG J, DENG Y, et al. PRBench: End-to-end Paper Reproduction in Physics Research[EB/OL]. 2026. https://arxiv.org/abs/2603.27646. arXiv: 2603.27646 [cs.CL].

[65] HUANG W, XIA F, XIAO T, et al. Inner Monologue: Embodied Reasoning through Planning with Language Models[EB/OL]. 2022. https://arxiv.org/abs/2207.05608. arXiv: 2207.05608 [cs.RO].

[66] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search[J/OL]. Nature, 2016, 529(7587): 484-489. https://doi.org/10.1038/nature16961. DOI: 10.1038/nature16961.

[67] SILVER D, HUBERT T, SCHRITTWIESER J, et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm[EB/OL]. 2017. https://arxiv.org/abs/1712.01815. arXiv: 1712.01815 [cs.AI].

[68] HUANG L, YU W, MA W, et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions[J/OL]. ACM Transactions on Information Systems, 2025, 43(2): 1-55. http://dx.doi.org/10.1145/3703155. DOI: 10.1145/3703155.

[69] HONG S, ZHUGE M, CHEN J, et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework[EB/OL]. 2024. https://arxiv.org/abs/2308.00352. arXiv: 2308.00352 [cs.AI].

[70] LU C, LU C, LANGE R T, et al. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery[EB/OL]. 2024. https://arxiv.org/abs/2408.06292. arXiv: 2408.06292 [cs.AI].

[71] MEYERSON E, PAOLO G, DAILEY R, et al. Solving a Million-Step LLM Task with Zero Errors[EB/OL]. 2025. https://arxiv.org/abs/2511.09030. arXiv: 2511.09030 [cs.AI].

[72] CHETYRKIN K, TKACHOV F. Integration by parts: The algorithm to calculate β-functions in 4 loops[J/OL]. Nuclear Physics B, 1981, 192(1): 159-204. https://www.sciencedirect.com/science/article/pii/0550321381901991. DOI: https://doi.org/10.1016/0550-3213(81)90199-1.

[73] LAPORTA S. HIGH-PRECISION CALCULATION OF MULTILOOP FEYNMAN INTEGRALS BY DIFFERENCE EQUATIONS[J/OL]. International Journal of Modern Physics A, 2000, 15(32): 5087-5159. eprint: https://doi.org/10.1142/S0217751X00002159. https://doi.org/10.1142/S0217751X00002159. DOI: 10.1142/S0217751X00002159.

[74] BRAN A M, COX S, SCHILTER O, et al. ChemCrow: Augmenting large-language models with chemistry tools[EB/OL]. 2023. https://arxiv.org/abs/2304.05376. arXiv: 2304.05376 [physics.chem-ph].

[75] LIANG J, HAN J, LI W, et al. GenericAgent: A Token-Efficient Self-Evolving LLM Agent via Contextual Information Density Maximization (V1.0)[EB/OL]. 2026. https://arxiv.org/abs/2604.17091. arXiv: 2604.17091 [cs.CL].

[76] WANG X, LI B, SONG Y, et al. OpenHands: An Open Platform for AI Software Developers as Generalist Agents[EB/OL]. 2025. https://arxiv.org/abs/2407.16741. arXiv: 2407.16741 [cs.SE].

[77] WU S. Introducing Devin, the first AI software engineer[Z]. https://cognition.ai/blog/introducing-devin. 2024.

[78] JIMENEZ C E, YANG J, WETTIG A, et al. SWE-bench: Can Language Models Resolve Real-world Github Issues?[C/OL]//The Twelfth International Conference on Learning Representations. 2024. https://openreview.net/forum?id=VTF8yNQM66.

[79] HU S, OUYANG M, GAO D, et al. The Dawn of GUI Agent: A Preliminary Case Study with Claude 3.5 Computer Use[EB/OL]. 2024. https://arxiv.org/abs/2411.10323. arXiv: 2411.10323 [cs.AI].

[80] 汪志诚. 热力学 · 统计物理[M]. 5 版. 高等教育出版社, 2013.

[81] LANDAU L D, LIFSHITZ E M. 统计物理学 I[M]. 5 版. 高等教育出版社, 2011.

[82] BLONDEL M, SANDER M E, VIVIER-ARDISSON G, et al. Autoregressive Language Models are Secretly Energy-Based Models: Insights into the Lookahead Capabilities of Next-Token Prediction[EB/OL]. 2026. https://arxiv.org/abs/2512.15605. arXiv: 2512.15605 [cs.LG].

[83] YANG Z, BAO C, BIN Y, et al. Turbulence-like 5/3 spectral scaling in contextual representations of language as a complex system[EB/OL]. 2026. https://arxiv.org/abs/2604.05536. arXiv: 2604.05536 [cs.CL].

[84] HU S, CAI X, HUANG Y, et al. Emergent Slow Thinking in LLMs as Inverse Tree Freezing[EB/OL]. 2026. https://arxiv.org/abs/2509.23629. arXiv: 2509.23629 [cs.AI].

[85] 曾谨言. 量子力学[M]. 4 版. 科学出版社, 2007.

[86] PESKIN M E, SCHROEDER D V. 量子场论导论[M]. 世界图书出版公司, 2020.

[87] WEI Z, PANG L, LIU J, et al. The Evolution of Thought: Tracking LLM Overthinking via Reasoning Dynamics Analysis[EB/OL]. 2026. https://arxiv.org/abs/2508.17627. arXiv: 2508.17627 [cs.CL].

[88] 刘川. 热力学与统计物理[M]. 北京大学出版社, 2024.

[89] 刘川. 理论力学[M]. 北京大学出版社, 2024.

[90] LANDAU L D, LIFSHITZ E M. 力学[M]. 5 版. 高等教育出版社, 2007.

[91] KAC M. On the notion of recurrence in discrete stochastic processes[J/OL]. Bulletin of the American Mathematical Society, 1947, 53(10): 1002-1010. https://doi.org/10.1090/S0002-9904-1947-08927-8. DOI: 10.1090/S0002-9904-1947-08927-8.

[92] WILSON K G, KOGUT J. The renormalization group and the ϵ expansion[J/OL]. Physics Reports, 1974, 12(2): 75-199. https://doi.org/10.1016/0370-1573(74)90023-4. DOI: 10.1016/0370-1573(74)90023-4.

[93] KUBO R. Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems[J/OL]. Journal of the Physical Society of Japan, 1957, 12(6): 570-586. https://doi.org/10.1143/JPSJ.12.570. DOI: 10.1143/JPSJ.12.570.

[94] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization[J/OL]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82. https://doi.org/10.1109/4235.585893. DOI: 10.1109/4235.585893.

[95] COVER T M, THOMAS J A. Elements of Information Theory[M/OL]. John Wiley & Sons, 2001. https://doi.org/10.1002/0471200611.

[96] HOGG T, HUBERMAN B A, WILLIAMS C P. Phase transitions and the search problem[J/OL]. Artificial Intelligence, 1996, 81(1): 1-15. https://www.sciencedirect.com/science/article/pii/0004370295000445. DOI: https://doi.org/10.1016/0004-3702(95)00044-5.

[97] OpenAI. Introducing GPT-4.1 in the API[Z]. https://openai.com/index/gpt-4-1/. 2025.

[98] OpenAI. GPT-4o System Card[EB/OL]. 2024. https://arxiv.org/abs/2410.21276. arXiv: 2410.21276 [cs.CL].

[99] OpenAI. Introducing OpenAI o3 and o4-mini[Z]. https://openai.com/index/introducing-o3-and-o4-mini/. 2025.

[100] DeepSeek-AI. DeepSeek-V3 Technical Report[EB/OL]. 2025. https://arxiv.org/abs/2412.19437. arXiv: 2412.19437 [cs.CL].

[101] TEAM Q. Qwen2.5 Technical Report[EB/OL]. 2025. https://arxiv.org/abs/2412.15115. arXiv: 2412.15115 [cs.CL].

[102] TEAM Q. QwQ-32B: Embracing the Power of Reinforcement Learning[EB/OL]. 2025. https://qwenlm.github.io/blog/qwq-32b/.

附录 A　术语表

下表列出本文涉及的核心术语及其中英文对照。

表 A.1　术语表

中文术语	英文术语与说明
	大语言模型基础
Transformer 架构	Transformer，基于自注意力机制的神经网络架构
词元	Token，大语言模型的基本生成单元，代表一个单词、子词或字符
注意力机制	Attention Mechanism，通过查询-键-值计算聚合全局信息
词元关联子	Token Correlation，用于刻画 Transformer 层间词元表示相关性和有效维度变化的指标
可解模型	Solvable Model，在简化假设下可解析求解、用于提取神经网络训练或缩放行为核心结构的理论模型
表示几何	Representation Geometry，从表示流形、可分性和维度演化角度研究神经网络内部表征的方法
机制可解释性	Mechanistic Interpretability，将模型行为分解为可理解的局部特征、电路或计算单元的研究方向
Transformer 电路	Transformer Circuits，将残差流、注意力头和多层感知机层视为可分解计算通道的机制可解释性框架
稀疏自编码器	Sparse Autoencoder (SAE)，通过稀疏字典学习从模型激活中分解较单义特征的工具
顿悟	Grokking，训练中测试性能在长时间停滞后突然提升的现象，通常被视为学习动力学转变的例子
预训练	Pre-training，在海量无标注文本上的自监督学习阶段

接下页

表 A.1（续）

中文术语	英文术语与说明
监督微调	Supervised Fine-Tuning (SFT)，在高质量标注数据上训练以遵循指令
人类反馈强化学习	Reinforcement Learning from Human Feedback (RLHF)，使模型行为与人类偏好对齐
规模定律	Scaling Laws，模型性能与参数量、数据量、计算量之间的幂律关系
	推理与采样
思维链	Chain-of-Thought (CoT)，通过生成中间推理步骤来增强复杂问题求解能力
自洽性	Self-Consistency，多次采样后通过多数投票聚合答案以提高可靠性
多数投票	Majority Voting，从多个候选输出中选择出现频率最高的结果
	智能体概念
马尔可夫决策过程	Markov Decision Process (MDP)，智能体的经典形式化框架
上下文工程	Context Engineering，通过设计提示和环境反馈的呈现方式来引导 LLM 推理
多智能体协作	Multi-Agent Collaboration，多个 LLM 智能体各司其职、层次化协作
形式化推理	Formal Reasoning，利用形式化证明系统逐步验证推理的每一步
迭代进化	Iterative Evolution，LLM 作为变异算子嵌入搜索循环，通过评估反馈迭代改进
闭环实验系统	Closed-Loop Experimentation，智能体直接与物理实验设备交互并反馈优化
智能体-计算机接口	Agent-Computer Interface (ACI)，为 LLM 优化的计算环境抽象层

接下页

表 A.1（续）

中文术语	英文术语与说明
	理论框架
算子图	Operator Graph，将有向边上的概率分布作为统一数学对象的形式化语言
算子	Operator，算子图中的节点，形式化为转移核
转移核	Transition Kernel，描述算子从输入状态分布到输出状态分布的概率映射
系综	Ensemble，词元序列上的概率分布，对系综的测量即为不可约实验，是连接微观动力学与宏观表现的桥梁
流水线	Pipeline，算子的有序组合，描述智能体从输入到输出的完整处理流程
投影	Projection，将文本级状态空间映射到有限离散状态空间的映射 φ，或将概率分布映射到宏观表示的操作
可约性	Reducibility，pipeline 的终端分布可从单步算子核的矩阵乘法预测的性质
线性表示	Linear Representation，将状态分布通过效用函数 $J = \langle s \rangle_\pi$ 投影为标量性能分数的线性表示
响应率	Susceptibility $\chi = \partial J / \partial \mathcal{B}$，效用函数对计算预算的导数
势能表示	Potential Representation，从转移核的反对称结构中提取标量势函数的表示
势函数	Potential Function V，状态空间上的宏观自由能，反映 LLM 对状态的全局排序
细致平衡	Detailed Balance，转移概率的对数比等于势函数之差的约束条件
1-形式假设	1-Form Hypothesis，条件自由能的反对称部分构成状态空间上的恰当 1-形式

接下页

表 A.1（续）

中文术语	英文术语与说明
效用函数	Utility Function $J = \langle s \rangle_\pi$，将状态分布投影为标量性能分数的线性泛函
参数自举	Parametric Bootstrap，从估计量的后验分布（如 Dirichlet）中重采样以量化有限样本噪声的统计方法
不可约实验	Irreducible Experiment，对单步转移核 $\mathcal{T}(g \leftarrow f)$ 的独立测量，是投影分析的基本单元
重整化群	Renormalization Group，通过逐步积掉微观自由度来构造不同尺度上自洽描述的方法论
最小作用量原理	Least Action Principle，从可观测转移概率中提取势函数的变分方法
首达时间	First Passage Time (FPT)，马尔可夫链从初始状态首次到达目标状态所需的步数
改善概率	Improvement Probability $P_{\text{imp}}(s)$，单次 LLM 调用后子代分数超过最优亲代的概率
能隙	Energy Gap ΔV，状态转移所需翻越的势能差，控制跃迁概率的指数抑制强度
岛屿	Island，IdeaSearch[11]中并行独立运行的搜索实例，岛屿之间不共享中间结果
棘轮机制	Ratchet，全局最优分数只升不降的约束，保证搜索过程的单调进展

附录 B　实现案例的详细实验结果

本附录为正文第三章引用的三项作者代表性工作提供详细实验数据。数据来源均标注于对应小节开头。

B.1　IdeaSearchFitter 的详细结果

本小节数据来源：Song, Cai, Zhang et al.[11]。代码仓库：https://github.com/IdeaSearch。

FSReD 基准测试

IdeaSearchFitter 在 Feynman 符号回归数据库（FSReD，120 个问题）上与主流基准方法进行了系统比较。表 B.1给出了不同噪声水平下的恢复率。

表 B.1　FSReD 上的恢复率比较

方法	$\gamma = 0.0$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$
IdeaSearchFitter	**82.5%**	**80.0%**	**79.2%**	**71.7%**
IdeaSearchFitter（含单位）	81.7%	74.2%	73.3%	69.2%
PySR	48.3%	34.2%	31.7%	25.8%
Operon	20.8%	7.5%	0.0%	0.0%
AI-Feynman	53.3%	33.3%	13.3%	0.8%

在搜索效率方面，IdeaSearchFitter 通常在 10 分钟以内达到 $>$ 70% 的恢复概率，Epoch 中位数（3/4 分位数）在 $\gamma = 0.0$ 时为 2.00，在 $\gamma = 0.1$ 时为 10.00；启用单位检查后可降至 3.50。中位词消耗量（词元用于 LLM 推理）在 1648 至 3757 之间。

真实世界数据集的模型发现

IdeaSearchFitter 在 PMLB 数据集上展现了良好的机制对齐能力：

- **葡萄园产量**：发现 $\hat{y} = \alpha \log(\frac{9}{2}\text{lugs}_{1989} + \text{lugs}_{1990}) - \beta$，其中 $\alpha \approx 6.93$，$\beta \approx 3.27$。对数压缩编码了农业系统的记忆饱和效应，复杂度 13 节点下的验证 NMSE 为 0.113。
- **超新星光变曲线**：发现 $\hat{y} = \exp(-\alpha(\frac{t-|t|}{2})^2 - \beta(\frac{t+|t|}{2}))$，利用 $|t|$ 天然实现“快升慢降”的物理机制分离。复杂度 22 节点下的验证 NMSE 为 0.0144。

PDF 参数化应用

在高能物理的质子部分子分布函数（Parton Distribution Function, PDF）应用中，IdeaSearchFitter 直接从 CT18NNLO 数据网格搜索 $f_i(x, Q)$ 的二维解析形式。对 $\bar{u}$ 分布，发

现的表达式为

$$x f_{\bar{u}}(x,Q) = p_1 \log\!\big(1 + p_2 \log Q\big) \log\!\Big(\tfrac{1}{x}\Big) (1-x)^{\,p_4 + p_5 \log Q}, \tag{B.1}$$

该形式包含 4 个参数，复杂度 24 个节点。$\log(1/x)$ 项编码了 Regge 动机的小 x 增长，$(1-x)^{p_4+p_5\log Q}$ 项编码了大 x 抑制及其对能标 Q 的 DGLAP 对数演化。从训练域 $Q \in [2,5]$ GeV 到外推域 $Q = 100\,\mathrm{GeV}$，IdeaSearchFitter 的验证 χ^2 为 34.00，显著优于 PySR 的 128.98，且在前沿复杂度下无过拟合迹象。

B.2 IBP 约化优化的详细结果

本小节数据来源：Song, Yang, Cao et al.[8]。

优先级函数搜索

FunSearch 对单圈无质量 bubble 积分进行 5000 epoch×10 次的搜索后，发现了最优优先级函数簇，由人类专家推广为：

$$F_0(I(n_i); I_{\text{target}}(t_i)) = -\left\{ \left[\sum_i |n_i|^m \right]^{1/m} + \left[\sum_i (n_i - t_i)^2 \right]^{1/2} \right\}, \tag{B.2}$$

其中 $m \in \{1, 2, 4, \infty\}$。实际应用采用 $m = 2$（椭圆型）。

五圈平面六粒子相空间积分族

图 B.1展示了五圈六粒子相空间积分的平面和非平面拓扑结构。实线为无质量传播子，虚线为外动量。

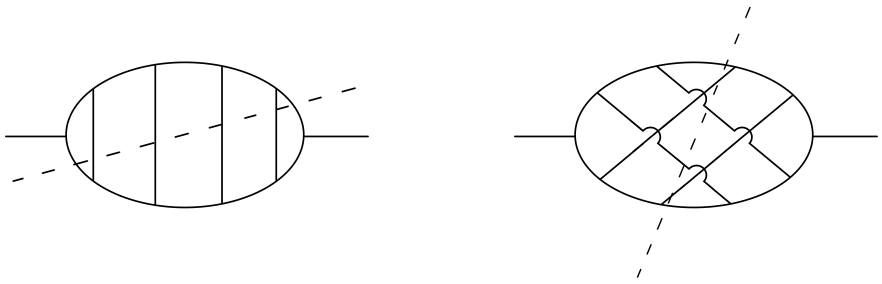

图 B.1　五圈六粒子相空间积分族的费曼图拓扑[8]

表 B.2给出了平面积分族中代表性目标积分的约化性能对比。格式为（种子积分数 / 学习时间 (s) / 单点求解时间 (s)）。“OOM” 表示超出 400 GB 内存限制。

表 B.2　平面六粒子积分族的 IBP 约化性能对比

积分	改进种子	优先级函数 F_0	改进倍数
s_6	108215 / 551.9 / 5.02	46390 / 226.8 / 4.0	2.33
s_8	791164 / 4271.4 / 36.2	101549 / 624 / 20.9	7.79
s_{10}	4388968 / OOM	268909 / 2198.8 / 132.7	16.32
s_{12}	19901220 / OOM	802488 / 12508.9 / 1462.3	**24.8**
d_5	21740796 / OOM	7109 / 47.7 / 2.24	**3058**

对于含大量分子和点数的子区积分 d_5，F_0 将种子积分从约 2174 万减少到 7109，改进倍数达 3058，使原先需要超过 400 GB 内存的任务可在标准笔记本电脑上于两分钟内完成。

泛化能力

在一圈 bubble 积分上搜索出的 F_0 被直接推广到五圈层面和非平面拓扑，无额外训练，验证了该方法的可扩展性，体现了 FunSearch “搜索程序而非解”范式的优势。

B.3　量子电路设计的详细结果

本小节数据来源：Cao, Hou, Li et al.[10]。

1+1 维 XY 自旋链：人-AI 协作发现

LLM（DeepSeek-Distill-Qwen-32B，温度 1.0）在 100 小时内发现了一个 5 层、4 参数的紧凑变分拟设。图 B.2展示了该拟设的量子线路结构：每层由 R_y 单量子比特旋转门和 CNOT 双量子比特门交替构成，其中第二、三层的旋转角 $F_{\theta_{2,3},i_q}$ 依赖于量子比特位置 i_q，编码了 LLM 自主发现的空间调制模式。LLM 自主发现了空间调制旋转变分参数模式：

$$F^{\mathrm{AI}}_{\theta_{2,3},i} = \theta_2 + \theta_3 \cdot \sin\left(\frac{i}{n}\pi\right), \tag{B.3}$$

该形式打破了平移不变性以反映开边界条件，而这一物理洞察并非由人类显式提供。

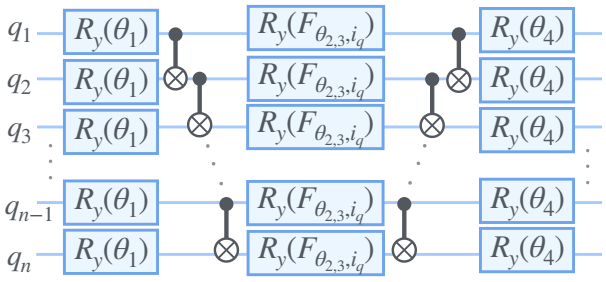

图 B.2　人-AI 协作发现的紧凑变分拟设量子线路[10]

人类专家受此启发，将其精炼为更符合物理直觉的形式：

$$F^{\text{human}}_{\theta_{2,3},i} = \theta_2 + \theta_3 \cdot \cos^{2n}\left(\frac{i}{n}\pi\right), \tag{B.4}$$

该形式在边界处锐化空间依赖性，同时保持体对称性。

最终人-AI 协作拟设在 $n = 9$ 时达到：

- 基态保真度 $F = 0.980$
- 相对能量偏差 $|\Delta E/E| < 0.4\%$

通过将 $n = 4$ 至 $n = 10$ 的优化参数拟合为平滑函数并外推，该拟设可扩展至 $n = 35$，外推能量偏差 $< 0.6\%$，单量子比特保真度 $F_s \gtrsim 0.99$。

“祖冲之”量子处理器验证

上述紧凑拟设在“祖冲之”量子处理器上成功制备了 20 位点 XY 模型的基态，通过零噪声外推有效缓解了双量子比特门误差。该任务是传统高参数设计难以完成的。

2+1 维标量场论：人-AI 协作

受 XY 模型中 LLM 自主发现对称性破缺的启发，人类专家主动引导 LLM 在对称性守恒子空间内搜索。该搜索发现了一个 3 参数、浅层深度的拟设。该电路在匹配传统 6 参数、深度 72 设计保真度的同时，将电路深度减少了 3 倍，验证了人类引导、AI 催化的协作范式在更难问题上的有效性。这一从 LLM 自主发现到人类精炼再到联合推广的过程，是该工作的核心贡献。

附录 C　物理理论文章的补充材料

本附录为正文第四章依赖的两篇物理理论文章提供正文限于篇幅未能展开的关键推导和实验细节。所有数据来源均标注于对应小节。

C.1　响应率框架的补充

本小节数据来源：Song & Zhu[22]。

自演化的现象学模型

正文第四章将响应率假说与智能体的自我改进能力相关联。响应率理论提供了一个描述此问题的现象学模型（图 C.1）。令数据质量函数 $p(b)$ 和模型输出质量 $l(b)$ 分别表示 pipeline 产生的训练数据质量和模型在该数据上的输出质量，两者均依赖于模型能力 b。自演化动力学由以下微分方程描述：

$$\frac{db}{dr} = \eta \cdot [p(b) - l(b)], \tag{C.1}$$

其中 r 为训练轮次，$\eta > 0$ 为学习率。固定点 b^* 满足 $p(b^*) = l(b^*)$，其稳定性由导数比较决定：若 $p'(b^*) > l'(b^*)$，该固定点为排斥子，能力低于 b^* 则退化，高于 b^* 则自我改进；若 $p'(b^*) < l'(b^*)$，则为吸引子，改进与退化均被限定于有限范围内。

跨领域实验的完整范围

图 4.8展示了响应率假说在四个任务域上的跨域验证。四个子图中，蓝色为基础分布性能 $J(\pi(\mathcal{B}))$，红色为 LLM 衍生分布性能 $J(\pi'(\mathcal{B}))$，在所有域中 $J(\pi)$ 的增长率均不低于 $J(\pi')$，支持响应率约束。

响应率假说在四个结构差异显著的任务领域中进行了验证：

- **Tetris**：10×20 棋盘，6 行预填充垃圾行，18 种固定方块朝向，50 步上限。性能 J 为消除行数。预算 $\mathcal{B}$ 为 beam width（$\{1, 2, 4, 8, 16, 32\}$）。基策略为深度优先回溯 beam search（lookahead=3）。
- **0/1 背包**：50 件物品，容量为总重的 30%。性能 J 为总价值。基策略按价值密度排序 beam search。
- **世界知识排名**：4 个数据集（15 国 GDP、15 国人口、8 行星直径、12 动物体重）。性能 J 为正确识别 rank-1 的比例。预算 $\mathcal{B}$ 为信噪比（$\{1, 2, 4, 8, 16, 32, 64, 128\}$）。

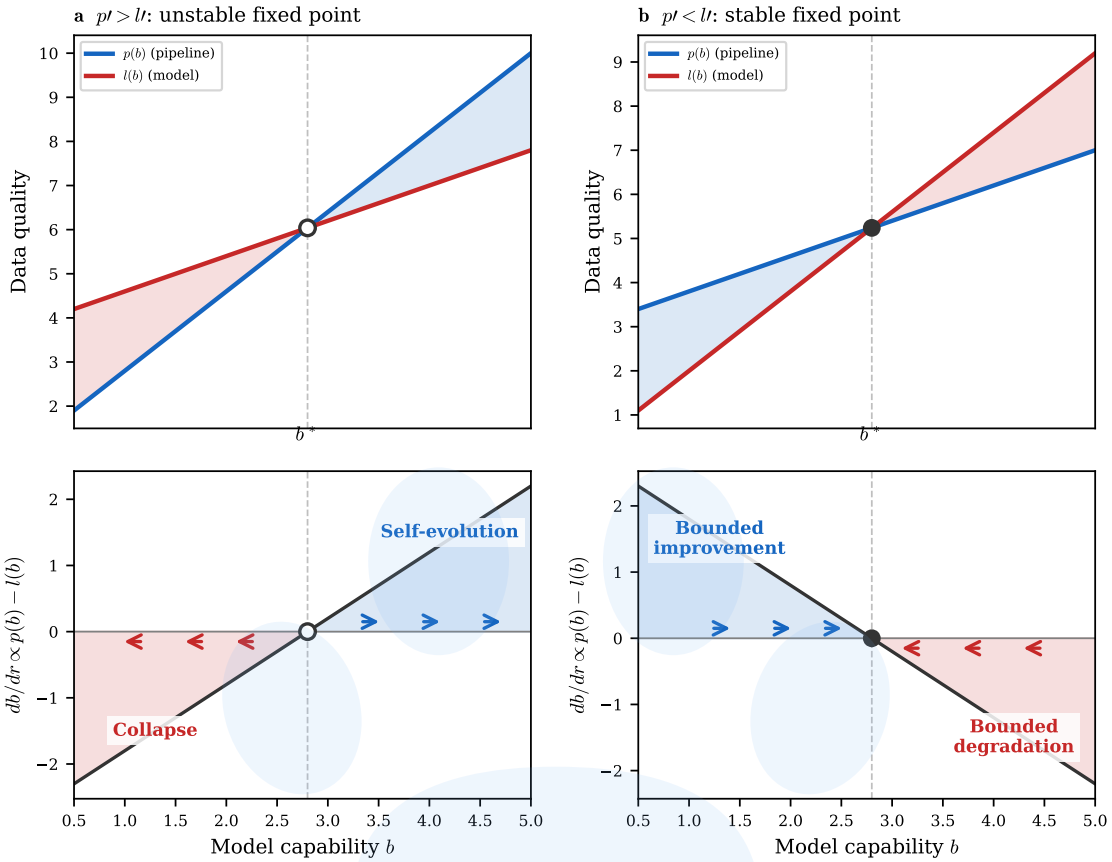

图 C.1　自演化的现象学模型[22]

- **AIME 数学**：60 道 2024-2025 年 AIME 题目。生成器模型大小 $\mathcal{B}_{\text{gen}}$ 和选择器模型大小 $\mathcal{B}_{\text{sel}}$ 为两个预算变量，样本数 $k \in \{1, 3, 5, 9, 15, 17, 19, 21\}$。

所有实验使用五个 Qwen 系列模型：Qwen2.5-Instruct-7B/14B/32B/72B 和 Qwen3-Max（~200B）。

C.2　细致平衡框架的补充

本小节数据来源：Song, Cao, Luo & Zhu[23]及其 Supplemental Material。

最小作用量原理与变分条件的等价性

正文第四章定义了作用量 $S_{\text{act}}[V] = \sum_{f,g} K(V(f) - V(g))\mathcal{T}(g \leftarrow f)$，其中 $K(x) = e^{-\beta x/2}$ 为描述转移对势函数排序违反程度的凸函数。最小作用量势函数 $V_{\mathcal{T}}$ 满足变分条件 $\delta S_{\text{act}}/\delta V(f) = 0$。当 $K(x)$ 为凸函数时，变分条件与最小作用量原理等价。

必要性：假设最小作用量 $V_{\mathcal{T}}$ 不满足变分条件，则存在状态 f_0 使 $\delta S_{\text{act}}/\delta V(f_0) \neq 0$。构造 $V'_{\mathcal{T}}(f) = V_{\mathcal{T}}(f) - \epsilon\delta_{f,f_0}(\delta S_{\text{act}}/\delta V(f_0))$，展开 $S_{\text{act}}[V'_{\mathcal{T}}]$ 后取 ϵ 充分小可得到 $S_{\text{act}}[V'_{\mathcal{T}}] <$

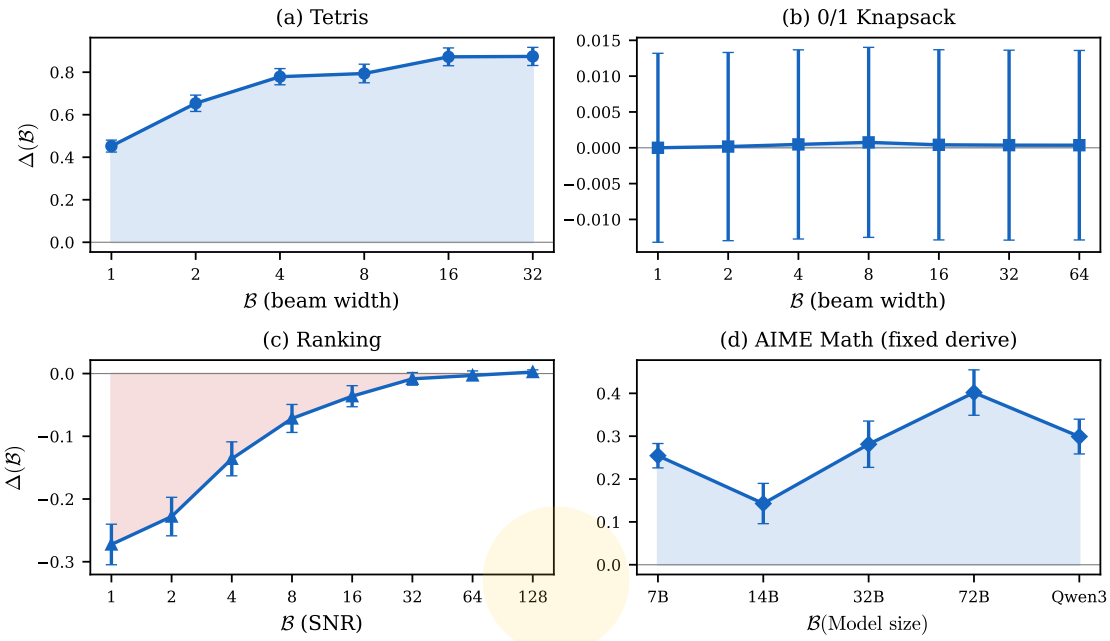

图 C.2　信息响应率假说的跨域验证[22]

$S_{\text{act}}[V_{\mathcal{T}}]$，与最小作用量矛盾。

充分性（K 为凸函数时）：对任意 V_1, V_2 及 $0 \le \lambda \le 1$，

$$S_{\text{act}}[\lambda V_1 + (1-\lambda)V_2] \le \lambda S_{\text{act}}[V_1] + (1-\lambda) S_{\text{act}}[V_2], \tag{C.2}$$

其中不等号来自 $K(x)$ 的凸性及 $\mathcal{T}(g \leftarrow f) \ge 0$。故 S_{act} 为凸泛函，变分条件的解即全局最小点。此外，任何满足 $K'(x) - K'(-x)e^{-\beta x} = 0$ 的 $K(x)$ 均可使细致平衡成为变分原理的充分条件[23]，对 $K(x)$ 的具体形式无特定要求。

两个验证智能体的设计

为了对细致平衡做交叉验证，作者设计了两个构造截然不同的智能体。

条件词生成智能体：状态空间由字母索引和为 100 的英文单词构成（如 ATTITUDE, EXCELLENT），LLM 读取当前状态后生成新词。实验覆盖了五个能力档次的模型（GPT-5 nano, Claude Sonnet 4.5, Gemini 2.5 Flash 等）。能力更强的模型对指令的遵循更稳定，相应得到的状态空间刻画也更具区分意义。

IdeaSearchFitter 智能体：在符号回归任务中运行 10 次专家模式，采样温度设为 1000.0 以均匀采样状态空间。

补充验证实验

文献[23]的 Supplemental Material 中还有以下几项补充验证：

1. **闭环路径验证**：在状态转移图中搜索 620 个三元组 (f, g, h)，验证闭合路径上正向与反向转移核对数之和近似为零，与细致平衡一致。
2. $K(x)$ **函数的稳健性**：使用 $K(x) = \log(1 + e^{-\beta x})$ 替代正文中的 $K(x) = e^{-\beta x/2}$，恢复的势函数分布与正文结果基本一致。
3. **未测转移对的检验**：对 $f \leftarrow g$ 被测量但 $g \leftarrow f$ 未被测量的状态对（IdeaSearchFitter 中 18935 对，条件词生成中 8805 对），势函数差与转移核对数比之间的关系满足不等约束。
4. **随机智能体作为反例**：以 GPT-5 nano 对随机整数串的转移作为反例，结果不满足细致平衡。当状态空间对模型而言缺乏可解释的语义时，细致平衡不再成立，可见这一性质的根源在于 LLM 对状态空间的语义理解，而不是符号系统本身的属性。
5. **极端智能体的解析分析**：Claude Sonnet 4.5 在条件词生成任务中仅涉及 5 个提示词，其转移矩阵可被较准确地估计，支持对最小作用量和势函数分布的解析计算。

多数投票与势函数的关系

多数投票（Majority Voting）策略在势能表示框架中等效于缩放 β 参数而不改变势函数 V 的分布。设原始转移核为 $\mathcal{T}(g \leftarrow f)$，在 M 个候选中选择出现 $n > M/2$ 次的状态，新转移核为

$$\mathcal{T}'(g \leftarrow f) = \sum_{k=n}^{M} \binom{M}{k} [\mathcal{T}(g \leftarrow f)]^k [1 - \mathcal{T}(g \leftarrow f)]^{M-k} = I_{\mathcal{T}(g \leftarrow f)}(n, M - n + 1), \quad \text{(C.3)}$$

其中 $I_x(a, b)$ 为正则化不完全 Beta 函数。当 $\mathcal{T}$ 较小时，转移核比值近似满足

$$\frac{\mathcal{T}'(g \leftarrow f)}{\mathcal{T}'(f \leftarrow g)} \approx \left(\frac{\mathcal{T}(g \leftarrow f)}{\mathcal{T}(f \leftarrow g)} \right)^n, \quad \text{(C.4)}$$

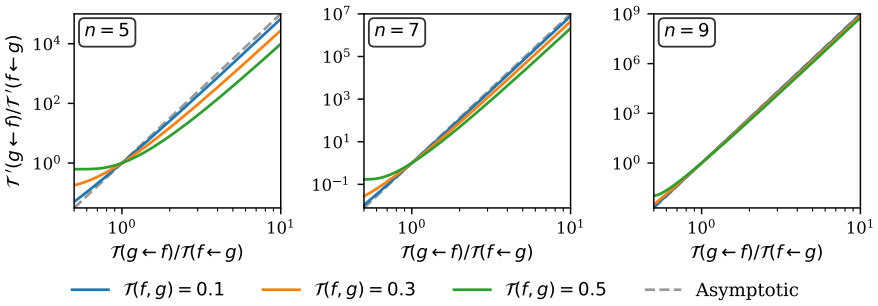

图 C.3　多数投票对势函数缩放的数值验证[23]

即 $V_{\mathcal{T}'} \approx n \cdot V_{\mathcal{T}}$。图 C.3以 $M = 10$ 为例进行了数值验证，三个子图分别对应 $n = 5, 7, 9$，

不同颜色表示不同的 $\mathcal{T}(g \leftarrow f)$ 取值；当 $\mathcal{T}$ 较小时近似成立。这意味着提高选择阈值 n 等效于降低作用量 $S_{\text{act}} \sim 1/n$，从而增强智能体生成动力学的方向性。

IdeaSearch 自动发现势函数形式

文献[23]还使用 IdeaSearch 自动搜索势函数的表达形式：将势函数表示为预定义函数和算子的组合，以最小化作用量为目标，经 4000 轮搜索后发现了具有清晰物理解释的最优势函数形式，进一步支持了势能表示作为可测量物理工具的有效性。

附录 D　本文原创实验的配置

本附录记录正文第四章中各项本文原创实验的详细配置，以便复现。本文所使用的基于算子图的粗粒化方法和智能体模拟开源于https://github.com/SonnyNondegeneracy/agentsimulator，提供可约性验证与多数投票模拟的完整复现脚本。

D.1　可约性验证实验

本节为正文图 4.3–4.5的实验配置。

任务域与投影映射

实验选择三个计算型任务域，每个域固定一道题（表 D.1），使不同模型间的结果具有可比性。三个任务域共享相同的四态投影映射 φ: Text $\to \mathcal{S} = \{\mathrm{C}, \mathrm{NW}, \mathrm{FW}, \mathrm{NA}\}$，完整的提取与分类流程如算法 1中 ParseBoxed 和 Classify 过程所示。

表 D.1　可约性实验的三个任务域

任务域	题目	标准答案
条件概率	盒子 A 中有 4 个红球和 6 个白球，盒子 B 中有 7 个红球和 3 个白球。先等概率随机选一个盒子，再从选中的盒子中不放回地连续取两个球。已知第一个取出的球是红球，求第二个球也是红球的概率。	$\frac{6}{11} \approx 0.5455$
马尔可夫稳态	一个马尔可夫链有 3 个状态，转移矩阵 P = [[0.5, 0.3, 0.2], [0.2, 0.6, 0.2], [0.3, 0.3, 0.4]]，求稳态分布中状态 1 的概率 π_1。	$\frac{9}{28} \approx 0.3214$
组合计数	将 5 个不同的球放入 3 个不同的盒子中，要求每个盒子至少放 1 个球。共有多少种不同的放法?	150

智能体组件设计

实验涉及生成、修正、工具三种算子，连同投影分类函数的完整操作流程如算法 1所示。生成算子的 temperature 设为 1.0 以获得非退化的初始分布。修正算子去除上一步输出中的 `<think>` 标签以避免思维链内容干扰修正模型的判断。工具算子为确定性的正则表达式操作，不涉及语言模型调用。

模型配置与核测量

实验使用的模型配置如表 D.2所示，共 $2 \times 6 = 12$ 个生成-修正组合 ×3 个任务域 =36 组不可约实验。

生成分布 $\vec{\pi}_0$ 的测量：对每个生成模型在每个任务域上独立采样 $N_{\mathrm{gen}} = 200$ 次，对输出进行投影分类并统计频率。修正核 T_{ij}^{ref} 的测量：从生成模型的输出中为每个状态

表 D.2　可约性实验的模型配置

角色	数量	模型
生成	2	Qwen2.5-7B-Instruct, Qwen2.5-32B-Instruct
修正	6	Qwen2.5-72B-Instruct, Qwen-Plus, Qwen3.5-Plus, Qwen3.6-Plus, DeepSeek-V4-Pro, GLM-5.1

i 选取 $n_T = 50$ 条代表性文本，输入修正模型，统计修正后落入各状态 j 的频率，构成 4×4 转移矩阵。工具核 T^{tool} 为确定性矩阵（行列顺序为 C, NW, FW, NA）：

$$T^{\text{tool}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{D.1}$$

即所有非 NA 态映射至 C，NA 映射到自身，无估计噪声。6 种 pipeline 拓扑见正文图 4.3，预测分布均通过式 (4.2)的矩阵乘法计算，每条 pipeline 端到端采样 $N_{\text{emp}} = 200$ 次得到经验分布。共 $36\times6 = 216$ 条 pipeline，约 58,000 次 API 调用。

统计检验

由于生成分布和修正核均从有限样本估计，预测分布 $\vec{\pi}_{\text{pred}}$ 和经验分布 $\vec{\pi}_{\text{emp}}$ 都带有采样噪声。为量化这两项噪声的联合效应，采用参数自举（parametric bootstrap）：将观测到的计数视为多项分布的充分统计量，从对应的 Dirichlet 后验中重采样，模拟在相同采样量下预测与经验分布的预期波动。据此为每条 pipeline 构造效用 J 的 95% 置信区间和 L_1 距离的 95% 分位阈值 τ_{95}。

结果

图 D.1给出全部 216 条 pipeline 的完整评估。(a) 中预测效用 J_{pred} 与经验效用 J_{emp} 的一致性良好（Pearson $r = 0.935$，MAE = 0.022），205/216 (94.9%) 在参数自举 95% 置信区间内。(b) 展示了按配置类型分组的 L_1 距离，红线标记 per-config τ_{95} 阈值，全局 202/216 (93.5%) 通过 $L_1 < \tau_{95}$ 检验。两种检验的通过率均与零假设的 95% 理论预期一致。gen→tool→ref 在两种检验中均为通过率最低的配置（83%/86%），定量印证了态内分布 ρ_i 偏移的效应。

SOTA 修正模型的核收敛

图 D.2以条件概率任务（gen = Qwen2.5-7B-Instruct）为代表性案例，展示了四个修正模型的修正核 T^{ref}。定义核距离 $L_1 = \sum_{ij} |T^{\text{ref}}_{ij} - T^{\text{tool}}_{ij}|$，从 Qwen2.5-72B（$L_1 = 0.92$）到 Qwen3.6-Plus（$L_1 = 0.04$），修正核逐步趋向工具核 T^{tool} 的形态：非对角元素消失，所有非 NA 状态以接近 1 的概率映射到 C。这表明工具不是智能体架构中的特殊组件，

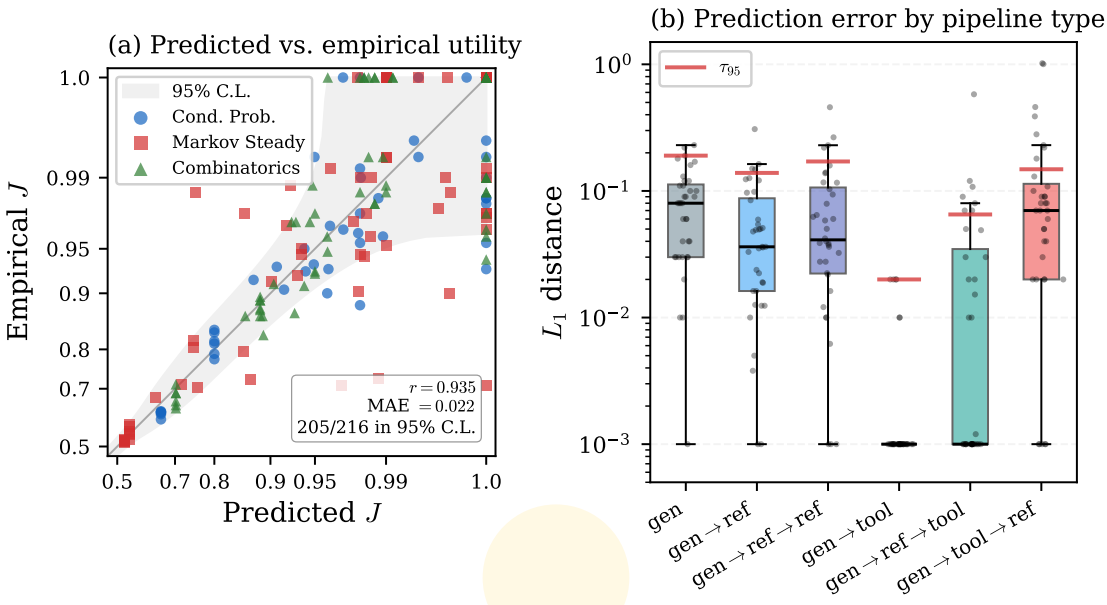

图 D.1　216 条 pipeline 的可约性验证：(a) 效用 J 散点图；(b)L_1 距离箱线图

而是算子能力的极限形式。

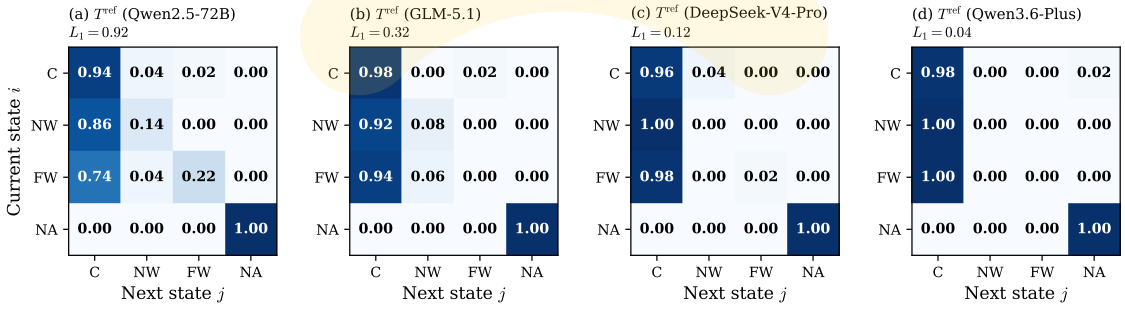

图 D.2　修正核向工具核的收敛（条件概率任务，gen = Qwen2.5-7B-Instruct）。L_1 为与 T^{tool} 的距离

D.2　思维链长度与正确率的指数衰减

本节为正文图 4.9的实验配置，并给出全部七个模型的难度控制分析。实验基于 PHYBench[63]的测试时计算扩展（test-time scaling, TTS）数据。PHYBench 在 TTS 实验中选取了开源的 100 道题目，对每个模型每道题独立采样 16 次以上。本文使用的数据涵盖六种模型：GPT-4.1[97]、GPT-4o[98]、o4-mini[99]、DeepSeek-R1[56]、DeepSeek-V3[100]和 Qwen2.5-Max[101]，以及数据量较少的 QwQ-32B[102]，共 17882 条记录，每条记录包含模型名称、prompt 词元数、completion 词元数和模型输出的答案。

分析流程如下：对每个模型，以 completion 词元数作为思维链长度 L_t 的度量，将

L_t 按等百分位分 bin，统计每 bin 中正确解答的比例 $\bar{p}$（正确性由 EED 分数是否超过阈值判定）。误差棒采用二项式标准误 $\sqrt{\bar{p}(1-\bar{p})/n_{\text{eff}}}$，其中 n_{eff} 为该 bin 内的独立题目数（而非总样本数），以反映题间独立性。对 $\bar{p}$ 与 L_t 的关系拟合指数衰减模型 $\bar{p} \sim e^{-L_t/L_0+c}$，提取特征长度 L_0，并给出 1σ 拟合预测带。

表 D.3　思维链衰减实验配置

参数	取值
数据集	PHYBench TTS 数据（100 道开源题，每模型每题 ≥16 次采样）
模型	GPT-4.1, GPT-4o, o4-mini, DeepSeek-R1, DeepSeek-V3, Qwen2.5-Max, QwQ-32B
预算变量	completion 词元数（思维链长度 L_t）
性能指标	正确解答概率 $\bar{p}$（EED ≥ 阈值的比例）
拟合模型	$\bar{p} \sim e^{-L_t/L_0+c}$，提取特征长度 L_0
误差估计	二项式标准误（对数轴），1σ 拟合预测带

难度控制分析

跨题目的观测统计中，难题自然倾向于产生更长的推理链且正确率更低，因此原始的 $\bar{p}$–L_t 衰减可能部分反映了题目难度与链长度的关联而非链长度本身的因果效应。为控制这一混淆因素，利用每个模型对每道题的多次独立作答（15–30 次），对链长度进行逐题逐模型的中位数归一化：对每个模型 m 和每道题 i，计算该模型在该题上所有作答的中位链长度 $\tilde{L}_{m,i}$，然后将每次作答的链长度归一化为 $L_t/\tilde{L}_{m,i}$。该操作消除了题目难度对链长度的系统性影响：归一化后的值反映的是同一模型在同一道题上某次作答相对于其典型长度的偏离程度。

表 D.4给出了数据集中全部具有足够样本量的七个模型的归一化结果。拟合时排除第一个分箱点，因为极短链区域的分布特征与其余区间存在系统性差异。

表 D.4　思维链衰减的难度控制结果。$\hat{L} = L_t/\tilde{L}$ 为归一化链长度，L_0 为拟合特征长度，$r = \sqrt{1 - \text{SS}_{\text{res}}/\text{SS}_{\text{tot}}}$ 为拟合相关系数。

模型	推理模型	中位 L_t (tokens)	题目数	L_0	r	归一化后
GPT-4.1	否	1826	88	1.77	0.85	衰减
GPT-4o	否	741	98	1.44	0.70	衰减
DeepSeek-V3	否	845	90	1.41	0.83	衰减
Qwen2.5-Max	否	982	89	0.71	0.52	衰减
o4-mini	是	4203	84	−4.08	0.53	反转
DeepSeek-R1	是	9186	86	−2.74	0.54	反转
QwQ-32B	是	7652	16	—	≈ 0	数据不足

结果呈现清晰的二分格局。全部四个非推理模型在消除难度因素后仍保持显著的指数衰减（$r = 0.52$–0.85），归一化 L_0 在 0.7–1.8 之间。两个数据充分的推理模型则呈

现反转（$L_0 < 0$），更长的推理链反而与更高的正确率关联。这一差异可在响应率框架下自然理解：推理模型通过专门的强化学习训练[56]获得了将额外推理词元转化为有效分析的能力，其有效预算上限 $\mathcal{B}$ 显著高于非推理模型。在有效预算范围之内，更长的推理链通过引入更深层的问题分解和自我验证而提升性能；式 (4.8)所预测的衰减仅在超出有效预算上限后才发生。换言之，思维链推理的价值在于提高了有效预算 $\mathcal{B}$ 本身，而非违反响应率约束。

D.3　工具集规模与性能的非单调关系

本节为正文图 4.10的实验配置。实验设计了一项固定任务，系统地改变智能体可用的工具数量，观察任务完成质量如何随工具集规模变化。任务选取为密度矩阵重正化群（DMRG）算法的实现和验证：给定 PRBench[64]中一篇 DMRG 论文的 PDF，要求智能体在沙箱环境中阅读论文、实现算法、生成指定格式的数据文件，由自动评分器（Judge）从方法理解、代码正确性、数据精度和完整性四个维度打分。

实验中工具集从最小的单工具（仅 python）逐步扩展到全部 9 种工具，中间设置了 2、4、6 三个档位。每个模型-工具集组合重复 3 次。共 75 次运行（5 个模型 × 5 种工具集规模 × 3 次重复），每次运行最多 30 轮迭代。

表 D.5　工具集规模实验配置

参数	**取值**
任务	DMRG 算法实现与验证（基于 PRBench 论文复现框架）
模型	DeepSeek-V3, DeepSeek-R1, Qwen-Plus, Qwen3-235B, Qwen3.5+
工具集规模	$\{1, 2, 4, 6, 9\}$
每组重复	3 次
每次最大迭代轮数	30
性能指标	Judge Score（0–1，四维度加权平均）
工具集构成	1: python 2: bash, python 4: bash, python, read_file, write_file 6: bash, python, read_file, write_file, glob, grep 9: 全部工具（含 search, web_fetch, edit_file）

正文图 4.10展示了其中 3 个模型（DeepSeek-V3, Qwen3-235B, Qwen3.5+）的结果，呈现倒 U 型曲线：工具数量过少时智能体缺乏必要的操作能力，过多时上下文窗口中的工具描述占用大量词元，反而降低了有效推理空间。

D.4 多数投票对生成分布的锐化效果

本节为正文图 4.13的实验配置。实验利用第4.1.2节可约性实验中已有的生成分布数据，通过蒙特卡罗模拟验证多数投票阈值 n 对正确率的指数调制效应。

数据来源：使用 Qwen2.5-32B-Instruct 在两个任务上的生成分布（每任务 $N = 200$ 个独立样本），选取正确状态 C 为优势态（条件概率任务，$p_C = 0.665$，$p_{NW} = 0.270$，$p_{FW} = 0.065$）和非优势态（马尔可夫稳态任务，$p_C = 0.055$，$p_{NW} = 0.945$）两种情况。

模拟流程：对于投票阈值 $n = 1, 2, \ldots, 11$，取 $M = 2n - 1$（满足 $n > M/2$ 的最大奇数 M）。每组配置独立执行 5×10^5 次 bootstrap 投票模拟：从 $N = 200$ 个样本中有放回地抽取 M 个，统计各状态出现次数，若某状态出现 $\geq n$ 次则该状态胜出，否则判为无效。拟合函数为 $P(n) = 1 - ae^{-bn}$（优势态）和 $P(n) = ae^{-bn}$（非优势态）。图 4.13中，(a) 以对数纵轴展示优势态（$p_C = 0.665$）的错误率 $1 - P$ 随 n 的指数衰减，(b) 展示非优势态（$p_C = 0.055$）的命中率 P 随 n 的指数衰减；数据点为蒙特卡罗模拟结果，曲线为指数拟合。

表 D.6　多数投票锐化实验配置

参数	**取值**
模型	Qwen2.5-32B-Instruct
任务（优势态）	条件概率（$p_C = 0.665$）
任务（非优势态）	马尔可夫稳态（$p_C = 0.055$）
样本数 N	200（来自可约性实验的生成数据）
投票阈值 n	$1, 2, \ldots, 11$
候选数 M	$2n - 1$（$= 1, 3, 5, \ldots, 21$）
每组 MC 模拟次数	5×10^5
拟合结果（优势态）	$P = 1 - 0.39\, e^{-0.187\, n}$
拟合结果（非优势态）	$P = 0.35\, e^{-1.84\, n}$

拟合指数 b 与自由能差 ΔF 的关系如下。在单步生成中 $V(i) = -\ln p_i$，正确状态 C 与优势态之间的自由能差为 $\Delta F = \ln(p_{\text{dom}}/p_C)$。若 $\beta_{\text{eff}} = n$ 严格成立，即有效分布 $p_{\text{eff}}(i) \propto p_i^n$，则 $b = |\Delta F|$。

表 D.7　拟合指数 b 与自由能差 $|\Delta F|$ 的比较

	$\lvert\Delta F\rvert$	b（拟合）
非优势态（$p_C = 0.055$）	2.84	1.84
优势态（$p_C = 0.665$）	0.90	0.19

拟合值 b 系统性地小于 $|\Delta F|$，这一偏差与附录 D.4 中 $\beta_{\text{eff}} = n$ 推导的适用条件吻合：该近似要求 $\mathcal{T} \ll 1$，而本实验的状态空间仅有 2–4 个态，优势态的转移概率（0.945

和 0.665）已不满足此条件。即便如此，命中率随 n 呈指数变化的形式在两种情形下都依然成立，与 β_{eff} 随投票阈值 n 上升、智能体生成方向性随之增强的预测相符。

D.5　搜索时间与势能景观的缩放关系

本节为正文图 4.18给出实验配置。实验基于 IdeaSearchFitter 符号回归任务中提取的转移矩阵数据，用蒙特卡罗模拟检验搜索时间 τ 与势能景观参数之间是否服从 Arrhenius 型关系。

数据准备：从 IdeaSearchFitter 数据库中选取一个代表性子图，采样 50 个状态，利用已测量的转移频次构造经验转移矩阵。势函数 V 和温度参数 β 来自正文第四章中细致平衡框架的拟合结果。

模拟流程：在 6 个不同温度下进行随机游走模拟。温度 T 通过缩放转移概率实现：$\mathcal{T}_T(g \leftarrow f) \propto \mathcal{T}(g \leftarrow f)^{1/T}$。目标状态分别定义为势函数 V 位于第 25、50、75、90 百分位以上的状态。每组配置独立运行 300 次，记录首次到达目标状态的步数 τ，最大步数限制为 5×10^6。

正文图 4.18由两个子图组成。子图 (a) 给出 $\ln(\tau/N)$ 随有效景观宽度 $\sigma(T)$ 的变化，四条曲线对应不同百分位的目标状态。子图 (b) 给出 $\ln(\tau/N)$ 与组合势垒 $\Delta V_{\text{eff}} + \sigma^2/2$ 之间的线性关系（斜率 0.2，$r = 0.9$），与 Arrhenius 型缩放律一致。

表 D.8　搜索时间缩放实验配置

参数	**取值**
数据来源	IdeaSearchFitter 符号回归转移矩阵数据库
子图状态数 N	50
温度 T	$\{1.0, 2.0, 5.0, 10.0, 100.0, 10^6\}$
目标状态	势函数 V 的第 25、50、75、90 百分位以上
每组独立运行	300 次
最大步数	5×10^6
拟合模型	$\ln(\tau/N) = a + b \cdot (\Delta V_{\text{eff}} + \sigma^2/2)$

D.6　IdeaSearch 进化过程的模拟验证

本节为正文第4.4节中 IdeaSearchFitter 实验验证的详细配置和补充结果。本节的进化模拟基于算子图框架，使用本附录开头提到的 agentsimulator 库实现，复用其粗粒化和马尔可夫模拟模块。实验在两个数据集上进行：IdeaSearchFitter 符号回归和 IBP 约化优化（双圈 bubble Feynman 积分，系延续 Song 等人[8]工作的进一步实验）。IBP 约化优化任务的计算瓶颈在评估器的大规模本地符号计算而非 API 调用，端到端实验难以扩展，因此基于转移核的模拟方法在此类任务中具有重要意义。

模拟方法与建模假设

IdeaSearchFitter 的完整单步操作 $\mathrm{Gen}(D) = \mathcal{T}_{\mathrm{eval}} \circ \mathcal{T}_2 \circ \mathcal{T}_1(\mathrm{Sel}(D))$ 包含数据库选择、LLM 提案生成、符号提取和数值评估四个环节（式 (4.4)）。模拟分两层结构捕捉其进化动力学的核心特征。

第一层为从数据估计的粗粒化转移核 $T_{\mathrm{gen}}(i \to j) = P(\text{child} \in \text{bin } j \mid \text{parent} \in \text{bin } i)$，以父代分数为条件从端到端转移数据中估计，捕获给定父代水平后 $\mathcal{T}_{\mathrm{eval}} \circ \mathcal{T}_2 \circ \mathcal{T}_1$（LLM 生成与评估链）的组合效应，Laplace 平滑（$\alpha = 10^{-6}$）确保遍历性。第二层在 T_{gen} 的基础上建模数据库选择机制 $\mathrm{Sel}(D)$：从分数不超过当前全局最优的区间中按 Boltzmann 权重 $\propto \exp(s/T)$ 采样父代区间，然后用 T_{gen} 生成子代。模拟采用棘轮（ratchet）机制：全局最优分数只升不降，即每步生成的子代仅在其分数不低于当前记录时才更新全局状态，否则结果被丢弃。多岛屿并行搜索中，每步各岛屿独立采样一个子代，全局最优取所有岛屿的历史最佳。

这一简化模型的两个核心建模选择是：T_{gen} 将生成-评估链的随机性压缩为分数区间之间的转移概率，温度加权的 $\mathrm{Sel}(D)$ 将数据库的动态选择效应建模为 Boltzmann 采样。模型对若干机制进行了简化处理。多父代的效应通过每个父代独立生成一个候选再从中选择来近似，捕捉了多父代扩展搜索宽度的基本效果，但未建模多个父代同时出现在 prompt 中时的语义交互：实际实验中不同父代的组合可能产生非独立的协同效应。岛屿模型被简化为完全独立的并行采样（每步各岛屿独立生成子代，取全局最优），未纳入实际 IdeaSearch 中岛屿间的迁移机制和共享数据库的复杂动态，因此模拟中的岛屿配置并非对任何一种实际运行模式的忠实复现，而是对并行搜索效应的理想化近似。此外，LLM 自身的采样温度已被 T_{gen} 的统计宽度隐式吸收。图 4.19(b) 中模拟预测与实验轨迹的一致性表明，尽管存在上述简化，模型仍然成功捕捉了 IdeaSearchFitter 的核心动力学特征，未来的工作可以在此基础上逐步引入更细粒度的机制建模以提升预测精度并对最优化参数空间进行更细致的探索。

表 D.9　进化模拟实验配置

参数	**IdeaSearchFitter（正文）**	**IBP（附录）**
独立运行数	10	3
状态转移总数	50288	3834
总运行时间	58.1 h	48.8 h
粗粒化分箱	$[0, 25, 26, 27, 30, 42]$	$[0, 29, 30, 31, 33, 35]$
状态标签	L, M, H, VH, E	L, M, H, VH, E
实测配置 $(T, n_{\mathrm{par}}, n_{\mathrm{isl}})$	$(1000, 1, 8)$	$(15, 5, 12)$
FPT 模拟次数	每配置 2000 次	每配置 2000 次

API 调用费用估算

IdeaSearchFitter 的 50288 次状态转移中，每次转移需要一次 LLM 调用。根据 LLM 输出长度（均值 3401 字符，约 972 词元）和输入长度（系统提示与父代信息，约 3300 词元/次）估算，总词元消耗约为 2.2×10^8。实验使用 DeepSeek-V3、GPT-5、GPT-5-mini、Gemini-2.5-Flash、Grok-4、Gemini-Pro、Gemini-2.5-Pro 和 Qwen3 共 8 个模型接近等比例混合调用，实际 API 费用约 200$。

首达时间分布与多岛屿效应

IdeaSearchFitter[11]采用岛屿模型（island model）进行并行搜索，每个岛屿独立维护搜索状态并定期迁移。图 D.3展示了 IdeaSearchFitter 数据集的 FPT 补充分析。(a) 为当前配置下的 FPT 分布直方图，中位 FPT 约为 3080 次 LLM 调用。(b) 展示了 FPT 随岛屿数量 n_{isl} 的变化：增加岛屿数可以降低墙钟时间（$\tau_{\mathrm{wall}} = \tau_{\mathrm{calls}}/n_{\mathrm{isl}}$），但总 LLM 调用次数轻微上升，反映了多岛屿并行搜索的协同开销：独立岛屿之间不共享中间结果，因此并行化仅加速墙钟时间而增加总计算量。

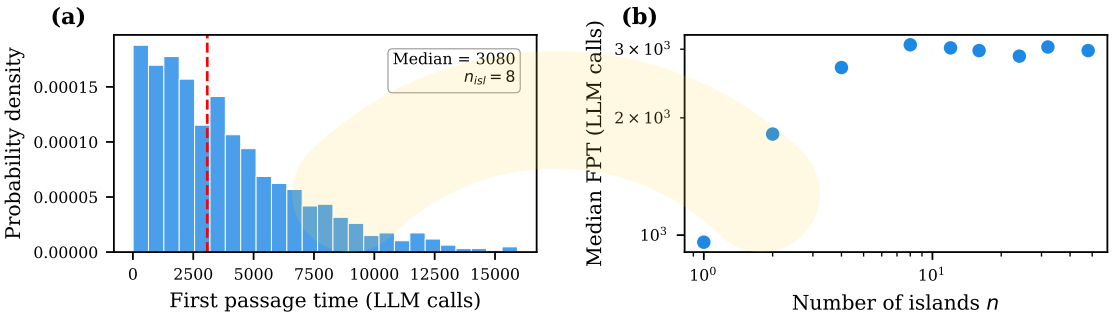

图 D.3　IdeaSearchFitter 数据集的 FPT 分布与岛屿数量效应

IBP 约化优化的对应结果

在双圈 bubble Feynman 积分的 IBP 约化优化中，3 次独立运行共产生 3834 次转移，耗时 48.8 小时。图 D.4和图 D.5展示了与正文 IdeaSearchFitter 数据集对应的分析。IBP 的转移核同样为下三角主导，改善概率整体较低（~ 0.01）。尽管两个数据集的动力学参数差异显著，改善概率的 logistic 衰减形式和势能景观的定性结构保持一致。

改善概率 logistic 形式的唯象解释

正文第4.4节指出，二次势函数 $V(s) = a(s-s^*)^2 + c$ 的景观结构与改善概率 $P_{\mathrm{imp}}(s)$ 的 logistic 衰减形式定性一致。此处从能隙的角度给出解释。对于从分数 s 出发的一次 LLM 调用，子代分数改善 δs 所需翻越的能隙为

$$\Delta V = V(s+\delta s) - V(s) \approx 2a(s-s^*)\,\delta s. \tag{D.2}$$

当 $s < s^*$ 时，$\Delta V < 0$，改善沿势能下降方向进行，能隙为零或为负，改善概率高且近似平稳。当 $s > s^*$ 时，$\Delta V > 0$ 且随 $(s-s^*)$ 线性增长：即能隙随分数增大而持续变

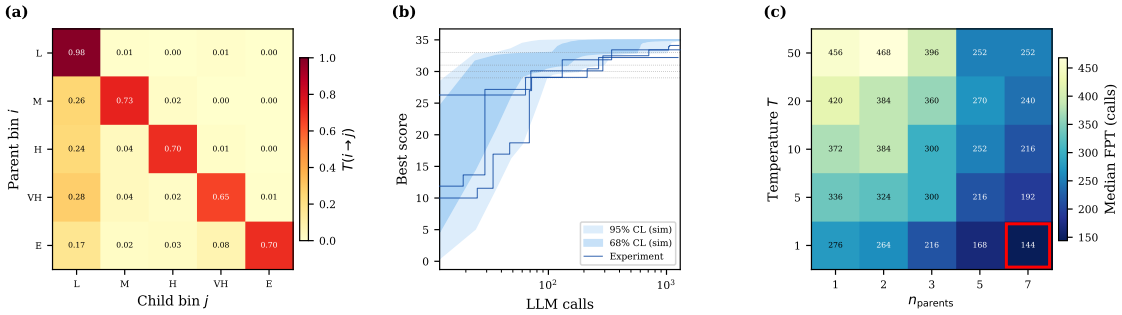

图 D.4　IBP 约化优化的进化模拟与参数扫描

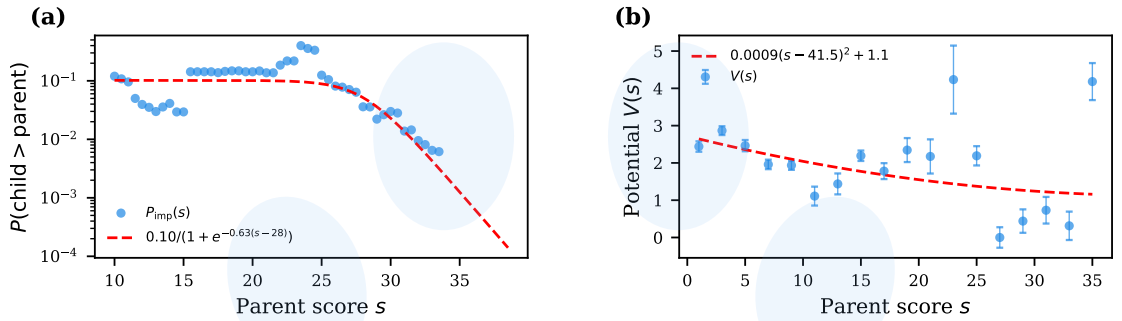

图 D.5　IBP 约化优化的改善概率与势能景观

宽。由于跃迁概率受能隙的指数抑制，能隙的线性增长导致改善概率从平台值 a_0 过渡到指数衰减，这一从平稳到急剧衰减的转变正是 logistic 函数的特征形状。实际拟合得到 $P_{\text{imp}}(s) = a_0/(1+\exp(-b(s-s_0)))$，其中 logistic 拐点 s_0 位于势阱的底部，与能隙开始变宽的区域对应。两个数据集（IdeaSearchFitter 和 IBP）的改善概率均呈现这一衰减形式，表明势能景观的能隙结构这一唯象模型可以解释 LLM 优化过程中性能提升的概率随分数变化的模式。

算法 1 可约性实验的完整操作流程

procedure PARSEBOXED(文本 x)　▷ 正则提取
　$x' \leftarrow$ 去除 x 中所有 `<think>...</think>` 标签
　matches $\leftarrow$ 正则匹配 x' 中所有 `\boxed{…}` 的内容
　if matches $= \emptyset$ **then return** None
　end if
　return float(last(matches))　▷ 非数值则返回 None
end procedure

procedure CLASSIFY(文本 x，标准答案 a^*)　▷ 投影 φ
　$v \leftarrow$ PARSEBOXED(x)
　if v = None **then return** NA
　end if
　$\varepsilon \leftarrow |v - a^*|/|a^*|$
　if $\varepsilon < 0.01$ **then return** C（正确）
　end if
　if $\varepsilon < 0.50$ **then return** NW（近似错误）
　end if
　return FW（完全错误）
end procedure

procedure GEN(题目 q)　▷ 生成算子
　prompt $\leftarrow q +$ “将最终答案写在 `\boxed{}` 中”
　return LLM(prompt, $T = 1.0$)
end procedure

procedure REF(上一步输出 x，原题 q)　▷ 修正算子
　$x' \leftarrow$ 去除 x 中所有 `<think>...</think>` 标签
　prompt $\leftarrow$ “以下是某人对一道题的解答：” $+ x' +$ “题目：” $+ q +$ “请仔细检查解答中的推理过程和计算是否正确。如有错误请修正。给出你的最终答案，写在 `\boxed{}` 中。”
　return LLM_{ref}(prompt)
end procedure

procedure TOOL(文本 x，标准答案 a^*)　▷ 工具算子
　$x' \leftarrow$ 去除 x 中所有 `<think>...</think>` 标签
　return 正则替换 x' 中所有 `\boxed{…}` 的内容为 a^*　▷ NA 态不变
end procedure