

2025 年 12 月 15 日至 12 月 21 日周报

何瑞杰

中山大学, 大湾区大学

目录

1. 项目进展	2
1.1. 使用神经网络学习生命游戏的演化动力学	2
2. 文献阅读	4
2.1. Scalable Diffusion Models with Transformers	4
3. 学习进度	7
3.1. 生成模型理论	7
3.2. 随机微分方程	7
3.3. 量子力学	7
4. 下周计划	10
参考文献	11

速 览

本周阅读了 DiT, 是 Diffusion 的一个高效高性能的工程化实践。在生命游戏项目中初步实现了朴素的规则推断机制, 但不具有鲁棒性, 也欠缺在实际训练集上验证。最后本周学习了少许 SDE 和量子力学的内容。

1. 项目进展

1.1. 使用神经网络学习生命游戏的演化动力学

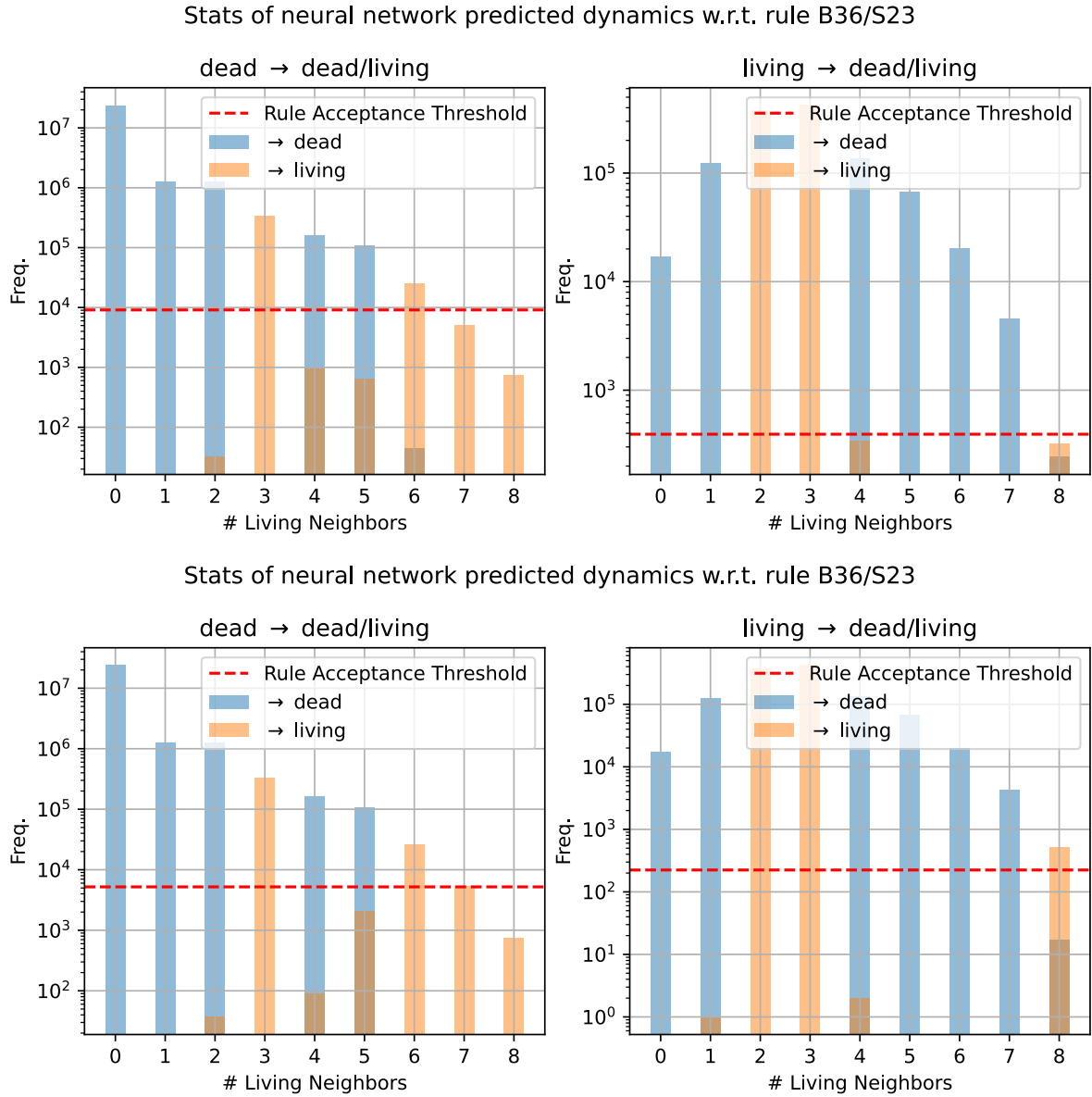


图 1 同一次 B36/S23 规则数据训练中生命游戏演化统计数据及硬编码阈值对比

首先考虑的是直接根据频数、正确率和模型预测确信程度设置阈值。第一个条件为频数需要大于等于某个设定的阈值 α ，其值与正确率和该类 (x_t 时刻细胞状态是存活还是死亡) 相关；第二个条件是模型对存活的预测要足够确信，即预测为存活的频数至少为预测为死亡频数的 β 倍。对应于 Python 代码如下：

```
def infer_rule_str(self, counters, acc) -> Tuple[List, List]:
    dd, dl = sum(counters[0].values()), sum(counters[1].values())
    ld, ll = sum(counters[2].values()), sum(counters[3].values())

    th_ratio = 0.6
    self.d_th = int(th_ratio * (1-acc/100) * (dd+dl))
    self.l_th = int(th_ratio * (1-acc/100) * (ld+ll))
```

```

print(dd, dl, ld, ll, self.d_th, self.l_th, acc)

d_all = counters[0] + counters[1]
l_all = counters[2] + counters[3]

filtered_b = sorted(list(filter(lambda x:x[1]>self.d_th, d_all.items())),
key=lambda x:x[0])
filtered_s = sorted(list(filter(lambda x:x[1]>self.l_th, l_all.items())),
key=lambda x:x[0])

self.born = []
self.survive = []

list_str = lambda x:list(map(lambda k:str(int(k)), x))

for i,_ in filtered_b:
    if counters[1][i] > 10 * counters[0][i]:
        self.born.append(i)

for i,_ in filtered_s:
    if counters[3][i] > 10 * counters[2][i]:
        self.survive.append(i)

return list_str(self.born), list_str(self.survive)

```

另外考虑在上述统计中邻居存活数分布极度偏倚的情况。考虑将统计用数据序列添加扰动，如以 γ 概率将一个死细胞翻转为活细胞；或在某些情况下增加训练数据中刚开始若干轮被采样到的概率。

将等变网络的群改变为 $p8$ （包括以 45° 为单位的旋转变换和平移变换），只需将下列函数中参数改为 $n=8$ 即可。

2. 文献阅读

2.1. Scalable Diffusion Models with Transformers

<https://arxiv.org/abs/2212.09748v2> | William Peebles, Saining Xie

一般的 Diffusion 系列模型所使用的主干都是 U-Net，本文提出了一个基于 Transformer 的替代方案，拥有良好的扩展性能和 SOTA (state of the art) 的 FID 生成分数。

2.1.1. 隐扩散模型 (latent diffusion model, LDM)

相比于一般的扩散模型，隐扩散模型工作在 VAE 等带有瓶颈结构的潜在空间中。其特点为相比直接在图像空间 (pixel space) 中运行扩散模型，隐空间维数远小于图像空间，这样可以显著降低计算开销和推理时间。注意 LDM 的工作空间只是从图像空间改为隐空间，因此扩散模型范畴内的方法，如 DDPM、DDIM、无类引导等方法均可挪用在 LDM 上。

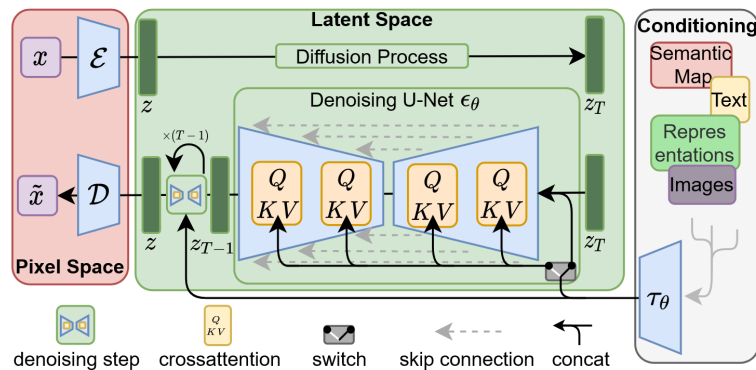


图 2 隐扩散模型的架构[1]

2.1.2. 视觉 Transformer (Vision Transformer, ViT)

视觉 Transformer 是将先前在自然语言处理 (natural language processing, NLP) 中大放异彩的 Transformer 架构引入计算机视觉 (Computer vision, CV) 领域的一次成功尝试。Transformer 架构的核心为注意力机制 (attention mechanism)，其本质为对加上位置编码的序列每项计算得到的键-值对进行匹配，其匹配方式是做内积，然后以此确定其他项相对于某一项的权重。换言之，注意力机制赋予序列中的每项不同的注意力，这是源于自然语言中词元 (token) (组) 之间的依赖关系，例如代词往往指示的是一个也许在千里之外的另一个名词。

现在将视线从 NLP 转向 CV，ViT 的核心思想是在图像中构造视觉 token，其方法为将图片按照网格划分为若干小块 (patch)，然后将这些小块经过映射后作为对应于该输入图像的视觉 token，最后输入 Transformer 模块。ViT 继承了 Transformer 的有点，具有可拓展性、高并行度、可优化、全局感受野等优点。

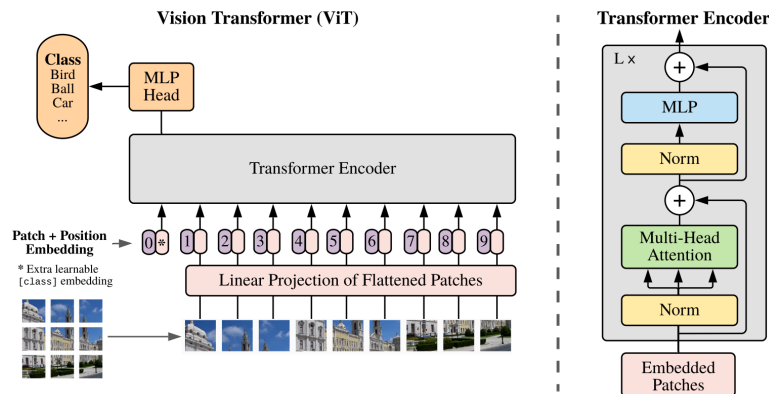


图 3 视觉 Transformer 的架构[2]

2.1.3. 扩散 Transformer (Diffusion Transformer, DiT)

DiT 可以说是融合了 ViT 和 LDM 这两个框架，或者说 DiT 是将 LDM 中主干换成 ViT 后的产物。为适配条件生成等任务，需要在模型中引入一个时间标志 t 和一个类别标志 c 。其中前者是将标量时间映射后的时间向量，后者可以是离散的标签，或文生图中对应生成提示词 (prompt) 的嵌入向量 (embedding vector)。作者提出了三种 DiT 模块

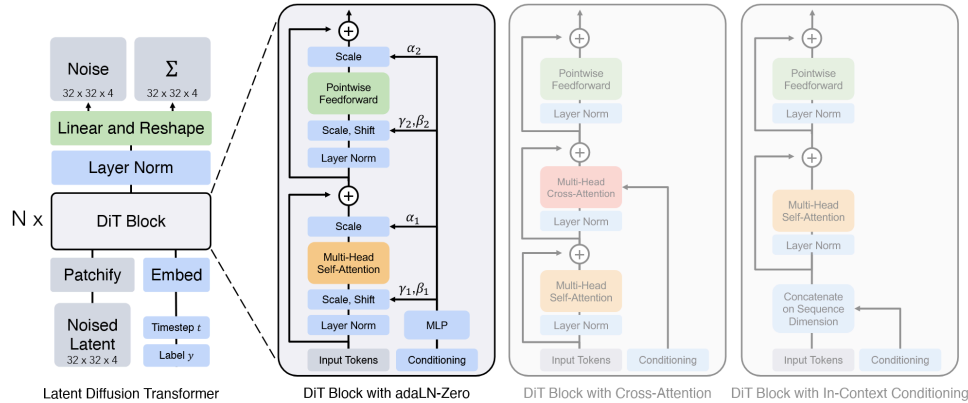


图 4 隐扩散模型的架构[1]

1. **上下文条件 (In-context conditioning)**：将时间标志和类别标志作为两个额外的 token 附加到输入序列中，与图像 token 平等。
2. **交叉注意力块 (Cross-attention)**：将时间标志和类别标志连接成一个短的序列，与图像 token 序列分开，并在中间的交叉注意力模块中进入。
3. **自适应层归一化 (adaLN)**：将层归一化 (layer norm, LN) 的参数改为由时间标志和类别标志而非从图像 token 中学习得到，并同等地施加在全体图像 token 上。
 - **adaLN-Zero 块**：对 adaLN 的修改，引入一个自时间标志和类别标志学习得到的缩放系数，作用于多头注意力后的缩放模块中。MLP 对应于输出 α 的部分为零初始化，这样在训练初期整个 DiT 模块近似为恒等映射。因为在扩散模型中，若加噪系数 $\sigma \rightarrow 0$ ，那么就有 $x_k \approx x_{k+1}$ ，这样有利于模型的快速训练。

2.1.4. 实验

Transformer 发力了。

2.1.4.1. DiT 模块比较

作者训练了四个最高计算量的 DiT-XL/2 模型，每个使用不同的 DiT 块。adaLN-Zero 块产生的 FID 低于其他两种，但计算效率最高（上下文条件 119.4 GFlops，交叉注意力模块 119.4 Gflops，adaLN 和 adaLN-Zero 118.6 Gflops）。adaLN-Zero 显著优于普通 adaLN。

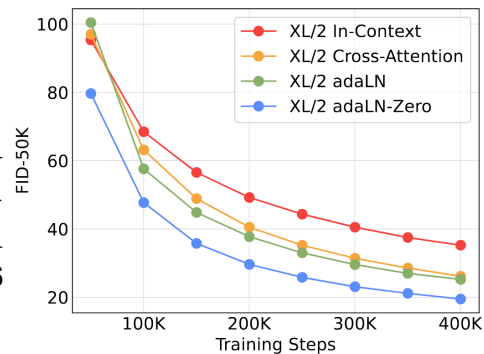


图 5 DiT 模块比较

2.1.4.2. 模型规模与块大小的规模法则

作者训练了 12 个 DiT 模型，覆盖模型配置 (S, B, L,

XL) 和块大小 (8, 4, 2)。实验结果显式增加模型规模和减小块大小都能显著改善扩散模型。当保持块大小不变增加模型大小时, FID 显著降低。保持模型规模不变减小块大小时, FID 显著降低。模型参数量并不能唯一确定模型的质量, 当保持模型规模不变而减小块大小时, Transformer 的总参数基本不变, 只有计算量增加。这些结果表明, 模型的计算量是其性能的关键。

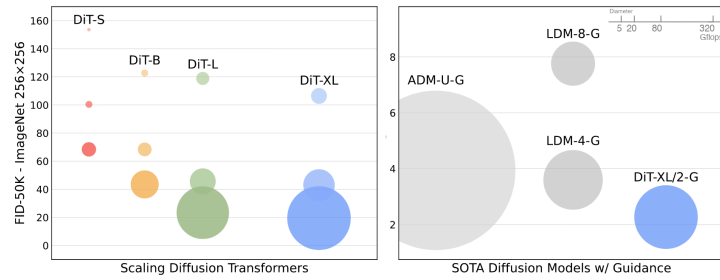


图 6 不同规模 DiT 模型的性能（左）及与 SOTA 模型的比较（右）
圆圈大小代表模型的计算量

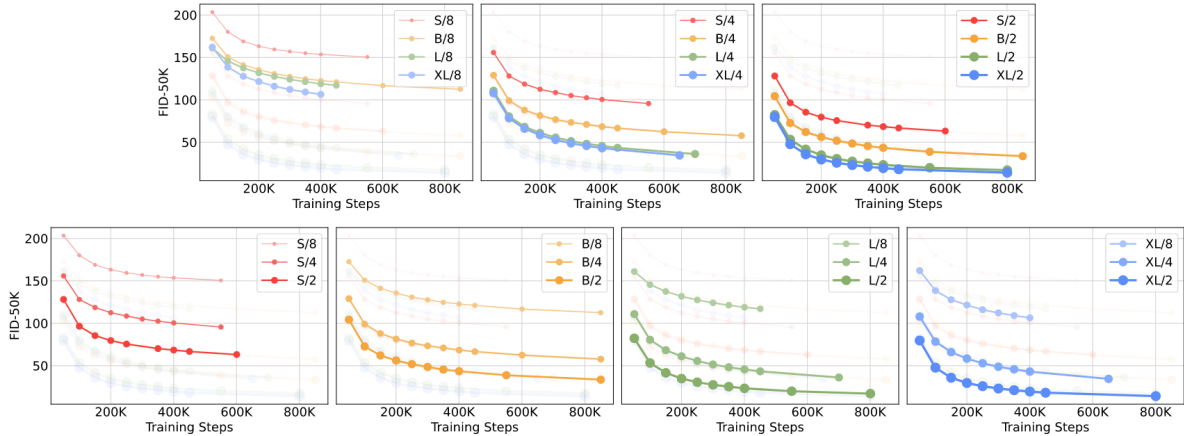


图 7 增加模型大小和减小块大小均能显著提升模型性能

2.1.4.3. 与 SOTA 模型的比较

在 ImageNet 256×256 的条件生成基准上，使用无分类器引导的 DiT-XL/2 优于所有先前的扩散模型。DiT-XL/2 在各种评估指标上均优于所有先前生成模型，包括之前的 SOTA StyleGAN-XL。在 512×512 分辨率上，DiT-XL/2 再次优于所有先前的扩散模型。即使 token 数量增加，DiT-XL/2 仍保持计算效率。

2.1.4.4. 模型计算 vs 采样计算

较小模型每图像使用的采样计算比较大模型多 5 倍，较大模型仍保持更好的 FID。一般来说，增加采样计算无法弥补模型的不足。我的推测是小模型无法对噪声和协方差矩阵做很好的估计，这样添加再多采样步数也无济于事。

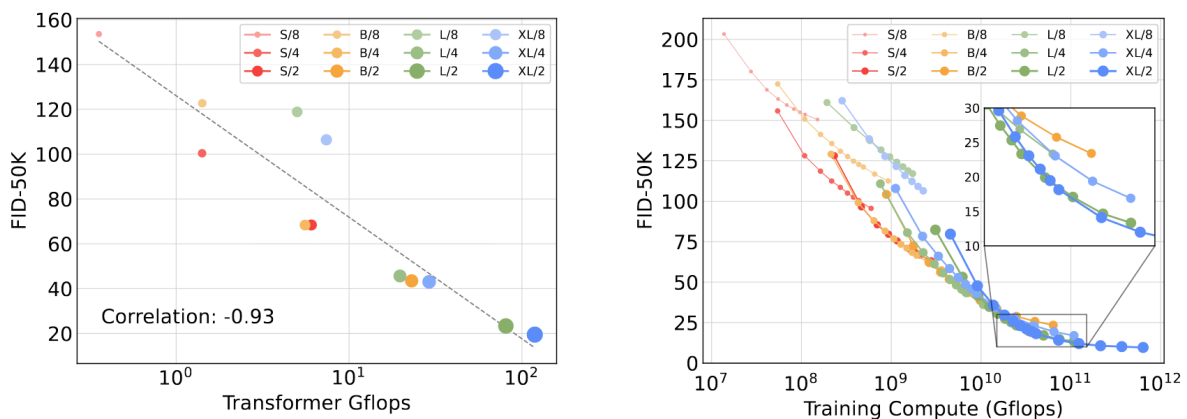


图 8 左：DiT 模块的计算量与 FID 分数显著相关；右：模型越大，对计算性能的利用越高效。

3. 学习进度

3.1. 生成模型理论

本周参阅了 MIT 的生成模型课程笔记，这是一本五十页的小册子，本周读完了大半部分内容。由于先前阅读了不少相关方面的论文，阅读起来没什么障碍，不过该讲义依然给予我了一些比较优雅的视角。

3.2. 随机微分方程

本周阅读了 SDE 解的存在性和唯一性定理的证明部分。

3.3. 量子力学

在 1921 年和 1922 年，O.Stern 和 W.Gerlach 进行了下面的实验。他们将银放在一个留有一个小孔的加热炉中加热，然后在银原子逃逸的路径上设置一个非均匀磁场。根据经典力学的预测，银原子束在通过磁场后应该会发生扩散，因为每个原子的磁矩方向是随机的。然而实验结果显示，银原子束在通过磁场后分裂成了两个离散的部分。如果这个磁矩是由旋转角动量产生的，那么我们应该观察到一个连续分布。因此这说明存在一个未知的内秉角动量，它在某一方向上只能取两个值。

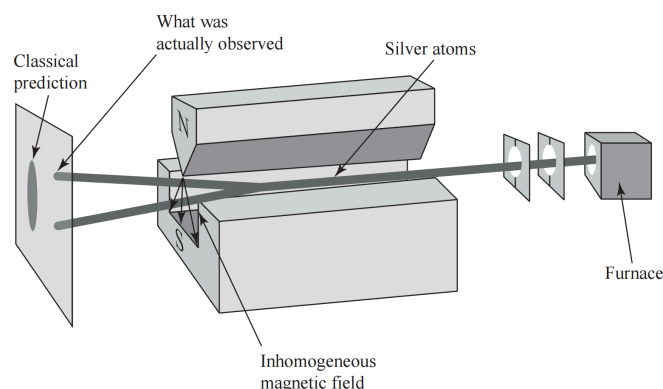


图 9 Stern 和 Gerlach 的实验¹

故事没有结束，接下来再看下面一个级联的 Stern-Gerlach 实验。上述实验对应下图中 (a) 子图的左边。当在 \hat{z} 方向施加非均匀磁场时，银原子束分裂成两束。现在遮挡其中一束，对另一束考虑下面三种处理

1. 再次通过一个 \hat{z} 方向的非均匀磁场，银原子束不再分裂；
2. 通过一个 \hat{x} 方向的非均匀磁场，银原子束分裂成两束；
3. 通过一个 \hat{x} 方向的非均匀磁场后，遮挡其中一束；再通过一个 \hat{z} 方向的非均匀磁场，银原子束再次分裂成两束。

¹本节内容参考自[3]

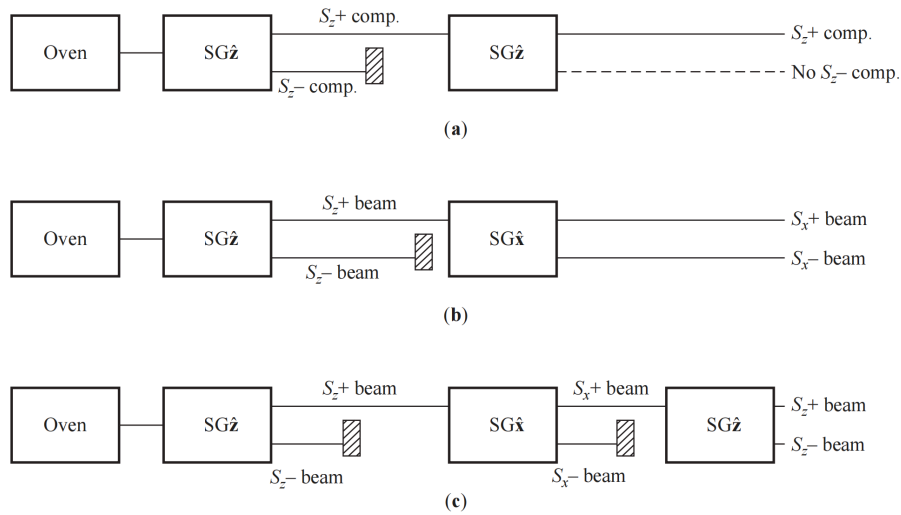


图 10 级联 Stern-Gerlach 的实验

该实验表明，每次经过某一方向的磁场，相当于对银原子的内秉角动量进行一次测量，测量结果只能取两个值的其中一个。如果已在某个方向上测量过，不经过其他操作再次测量时得到的结果不会变化。然而若在进行了某一方向的一次测量后再经过另一方向的测量，最初的测量结果会被“抹去”，换言之，不能同时确定两个方向上的测量结果。

为研究如上的量子现象，需要引入对应的数学工具。量子力学研究复 Hilbert 空间 \mathcal{H} 、其上的线性算子以及其对偶对象（复对偶 Hilbert 空间 \mathcal{H}^* 和其上的线性算子）。在量子力学中，称复 Hilbert 空间中的元素为态矢量 (state vector)，写作 $|a\rangle$ ，因此也称右矢量 (ket)，其对偶对象 $\langle a|$ 为左矢量 (bra)。复 Hilbert 空间中两个右矢 $|a\rangle$ 和 $|b\rangle$ 的内积为 $\langle a|b\rangle$ ，英文就叫 bra(c)ket。注意它可以写成 $\langle a||b\rangle$ ，而左边一项是右矢 $|a\rangle$ 的对偶，这是由 Riesz 表示定理保证的。由内积公理，对任意右矢 $|a\rangle$ ，有 $\langle a|a\rangle \geq 0$ ；对任意两个右矢 $|a\rangle$ 和 $|b\rangle$ ，内积有共轭对称性 $\langle a|b\rangle = \langle b|a\rangle^*$ 。若 \mathcal{H} 为至多可数维，那么存在一个正交归一基 (orthonormal basis) $\{e_i\}_{i \in \mathbb{N}}$ ，可以表出任意右矢 $|x\rangle$ 为

$$|x\rangle = \sum_{i \in \mathbb{N}} x_i |e_i\rangle \quad (1)$$

若将基矢 $|e_j\rangle$ 之对偶作用于 $|x\rangle$ ，得到

$$\langle e_j|x\rangle = \sum_{i \in \mathbb{N}} x_i \langle e_j|e_i\rangle = \sum_{i \in \mathbb{N}} x_i \delta_{i,j} = x_j \quad (2)$$

这样我们就能“形式上”地将右矢 $|x\rangle$ 写作一个“列矢量” x 的形式。需要注意的是，列矢量 x 是 \mathcal{H} 中的抽象元素 $|x\rangle$ 在基（或称表象） $\{e_i\}_{i \in \mathbb{N}}$ 下的表示，类似地不难证明其对偶 $\langle x|$ 可以写成 $x^{\top*} = x^\dagger$ 。

量子力学中称 \mathcal{H} 上的线性算子 $\hat{A}: \mathcal{H} \rightarrow \mathcal{H}$ 为算符。算符 \hat{A} 作用在右矢 $|x\rangle$ 上写作 $\hat{A}|x\rangle$ ，得到另一个右矢，其对偶的情形为 $\langle x|\hat{A}^\dagger$ ，其中 \square^\dagger 表示算子的 Hermite 共轭。若将一个右矢和一个左矢形式上地乘在一起，即 $|a\rangle\langle b|$ ，它定义了一个算符。事实上上述结果也可以写成张量积的形式，即 $|a\rangle\langle b| = |a\rangle \otimes \langle b|$ 。现在还是考虑一个基 $\{e_i\}_{i \in \mathbb{N}}$ 。一方面，右矢 $|x\rangle$ 可以写成基矢的线性组合，也可以被基矢作用得到在基矢“方向”上的“分量”或投影，有

$$|x\rangle = \sum_{i \in \mathbb{N}} |e_i\rangle \cdot x_i = \sum_{i \in \mathbb{N}} |e_i\rangle \langle e_i|x\rangle = \left[\sum_{i \in \mathbb{N}} |e_i\rangle \langle e_i| \right] |x\rangle = 1 |x\rangle. \quad (3)$$

于是我们成功构造了恒等算符 1。另一方面，算符 \hat{A} 作用在右矢 $|x\rangle$ 得到新的右矢 $|y\rangle$ ，用一个基矢 $|e_j\rangle$ 作用在 $|y\rangle$ 上得到

$$\langle e_j|y\rangle = \langle e_j|\hat{A}|x\rangle = \langle e_j|\hat{A} \cdot 1 \cdot |x\rangle = \sum_{i \in \mathbb{N}} \langle e_j|\hat{A}|e_i\rangle \langle e_i|x\rangle \quad (4)$$

若令 $A_{j,i} = \langle e_j|\hat{A}|e_i\rangle$ ，则有 $y_j = \langle e_j|y\rangle = \sum_{i \in \mathbb{N}} A_{j,i} x_i$ 。这表明算符 \hat{A} 在基 $\{|e_i\rangle\}_{i \in \mathbb{N}}$ 下的矩阵表示为 $\mathbf{A} = (\mathbf{A}_{j,i})$ ，相应地，其对偶算符 \hat{A}^\dagger 的矩阵表示为 $\mathbf{A}^\dagger = (\mathbf{A}_{i,j}^*)$ ，即为原算符矩阵的共轭转置。考虑算符 \hat{A} 同时作用在右矢 $|x\rangle$ 和左矢 $\langle y|$ 上，有

$$\langle x|\hat{A}|y\rangle = \langle x|[\hat{A}|y\rangle] = [\langle y|\hat{A}^\dagger|x\rangle]^* \quad (5)$$

如果 $\hat{A} = \hat{A}^\dagger$ ，我们称其为 **Hermite 算符**，Hermite 算符的所有本征值（特征值）为实数，每个本征值对应的本征态（特征向量）都是正交的。

4. 下周计划

论文阅读

1. 生成模型: EDM、LDM、ADM
2. Neural SDE 相关经典论文

项目进度

1. 使用神经网络学习生命游戏的演化动力学
 - 实现提取规则后的即时测试
 - 对四种不同的情况设置四种不同的阈值
 - 考虑基于 likelihood 的规则搜索算法
2. 耦合约瑟夫森结
 - 实现简单的 Neural SDE

理论学习 随机过程课即将考试, 本周和下周不设其他学习内容。

1. 随机过程课程
 - 总复习, 复习第一章和第二章

参考文献

- [1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models.” 2021.
- [2] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *CoRR*, 2020, [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [3] J. J. Sakurai, *Modern Quantum Mechanics (Revised Edition)*, 1st ed. Addison Wesley, 1993. [Online]. Available: <http://www.worldcat.org/isbn/0201539292>