arXiv:2303.10154v1 [cs.NE] 15 Mar 2023

表观遗传算法：用于调控染色体表达的自强化注意力机制

Mohamed Djallel DILMI[1] , Hanene AZZAG[2] , Mustapha LEBBAH[3]

[1]LIPN（法国国家科学研究中心 UMR 7030），索邦巴黎北方大学[2]LIPN（法国国家科学研究中心 UMR 7030），索邦巴黎北方大学[3]DAVID，巴黎萨克雷大学 电子邮箱：dilmi@lipn.univ-paris13.fr / azzag@lipn.univ-paris13.fr / lebbah@uvsq.fr

摘要。 遗传算法是生物启发式方法的典型代表。它们通过模拟突变、交叉和选择等多种算子来模仿自然选择。近期关于表观遗传调控过程的发现表明，这些过程发生在遗传基础“之上”或“之外”，涉及能够影响并改善基因表达的变化。这引发了一个问题：如何通过模拟表观遗传算子来改进遗传算法（GAs）。本文提出了一种新的表观遗传算法，用于模拟称为DNA甲基化的表观遗传现象。我们的表观遗传算法的新颖之处主要在于利用注意力机制和深度学习，这与基因增强/沉默概念十分契合。本文通过理论论证和实证研究，展示了所提出的表观遗传算法能够高效地解决比简单遗传算法更复杂的问题。例如，面对本文所述的两个非凸（多峰）优化问题时，该算法表现良好，且能够出色地克服局部最优解的局限，从而找到全局最优解。

关键词：表观遗传算法、遗传算法、深度学习、注意力机制、多峰优化问题

1 引言

在生物学中，表观遗传学是一项非常活跃的研究领域，因为它能帮助我们解答遗传学无法回答的问题。它结合了查尔斯·达尔文和让-巴蒂斯特·拉马克的思想。因此，有关表观遗传学的深入研究文献十分丰富。例如，Jablonka和Raz（2009年）对细胞表观遗传机制进行了全面综述，其中包含了一些重要概念，我们将在本文中尝试加以模仿。

在计算机科学领域，已有一些研究尝试将表观遗传机制引入进化计算，以用于优化。第2节回顾了

一些现有论文。本文提出一种新的表观遗传算法（EGA）方法，通过引入一种模拟DNA甲基化过程的全新注意力机制（Wolffe和Matzke，1999年）。因此，第3节首先概述表观遗传现象及其相关机制；随后，第4节阐述所提出的注意力机制；第5节则详细介绍利用注意力机制模拟DNA甲基化的表观遗传算法。该自监督框架能够学习解码阶段和表观遗传操作，无需对信息进行编码，从而提高优化问题的求解效率。论文最后一节（第6节）中的实验表明，所提出的改进提高了效率，并降低了陷入局部最优解的风险，只需经过少数几代即可达到全局最优解。

它涵盖多种机制，包括DNA甲基化 Wolffe和 Matzke（1999）（向DNA分子添加甲基）、组蛋白修饰 Ban- nister和 Kouzarides（2011）（改变DNA缠绕其上的蛋白质），以及非编码RNA分子；这些机制共同调控基因的活性。

从某种意义上说，表观遗传学起着基因表达控制系统的作用，使细胞能够应对环境变化或发育过程中的变化，而不会永久改变底层遗传密码。这些变化可以逆转，有时甚至可以遗传，因此是塑造生物体发育和进化轨迹的关键因素。

表观遗传学研究对许多领域都具有重要意义，包括癌症研 究、神经科学以及对复杂疾病的认识 Egger 等（2004）；Feinberg 和 Vogelstein（1983）；Simó-Riudalbas 和 Esteller（2014）。在医学领域，研究人员正在开发表观遗传疗 法，通过控制基因表达或靶向特定的表观遗传标记，为癌症和其他疾病提供潜在疗法。

总体而言，表观遗传学加深了我们对基因、环境与健康之间关系的理解，并为研究和治疗干预开辟了新途径。一个很好的例子是同卵双胞胎：他们拥有相同的DNA，但外貌却可能不同，van Dongen 等（2021）。这是因为在生命历程中，我们会经历不同的事情，因此彼此也会有所不同。在细胞中，存在许多机制来控制DNA的读取和表达。其中一些机制称为表观遗传修饰，可以在不改变DNA序列的情况下开启或关闭特定基因。DNA甲基化是最知名、研究最广泛的机制之一，发生于甲基添加到DNA碱基时。它动态调控基因转录的能力启发了本文提出的方法。

2 现状

近年来，表观遗传学启发了计算机科学领域的多篇文章。在本节中，我们将探讨其中的一些文章。
在 Periyasamy 等人（2008年）的研究中，作者提出了一种受生物分子所采用的优化策略启发的代内表观遗传算法，并提出了一种基于智能体的

细胞建模与仿真环境。作者旨在利用其表观遗传算法，研究经跨代表观遗传机制分化的不同细胞类型中癌症的发育模式。

Sousa和Costa（2010）提出了人工生命（ALife）领域的另一种方法，作者提出了EpiAL模型，该模型利用动态环境影响生物体的调控以及后天获得的表观遗传标记的遗传。EpiAL模型从人工生命的视角研究表观遗传现象的合理性及其与进化系统的相关性。作者计划进一步完善该模型，重点关注生物学知识和问题求解技术。

Tanev和Yuta（2008）提出了一种新的遗传编程方法，称为表观遗传编程，该方法通过组蛋白修饰实现对基因表达的显式控制。他们利用多型现象的累积效应实现表型多样性，并通过沉默基因型组合，保护个体免受交叉带来的破坏性影响。

Jablonka和Raz（2009）试图论证／证明，表观遗传学概念可应用于计算机优化中的信息表示和操作，以解决各种组合问题。尽管他们对表观遗传理论作了颇为出色的介绍，但其表观遗传算法仍较为初级，并采用随机变异。

据我们所知，尽管“表观遗传学”一词经常被提及，但这些研究中没有一项提出过能够良好模拟生物学机制及其特性（例如可逆性、发生于……之上等）的表观遗传算法。

在过去三到四年中，研究者提出了基于人工智能的模型来模拟表观遗传机制；在 Ezzarii 等人（2020年）的研究中，作者提出通过在给定数据集上训练 XGBoost分类器，学习一组表观遗传规则。
在相反的研究方向中，Huang 等人（2021年）提出了一种用于脑部表观遗传分析的机器学习方法，该方法揭示了与阿尔茨海默病相关的激酶。他们表明，可以利用机器学习技术提取表观遗传因素。这促使本研究采用一种基于注意力机制（第4节）的全新深度学习框架（第5节），以模拟表观遗传现象。

3 表观遗传学：概述

表观遗传学研究的是基因表达或细胞表型的变化，这些变化并不涉及底层DNA序列的改变**?** 。它涵盖多种机制，包括DNA甲基化 Wolffe和Matzke（1999年）（向DNA分子添加甲基）、组蛋白修饰 Bannister和Kouzarides（2011年）（DNA缠绕其上的蛋白质发生变化），以及非编码RNA分子；这些机制共同调节基因的活性。

I从某种意义上说，表观遗传学就像基因表达的控制系统，使细 ls
t以响应环境变化或发育过程中的变化，而不永久性地 y

改变底层的遗传密码。这些变化可能是可逆的，有时甚至可以遗传，因此它们是塑造生物体发育和进化轨迹的关键因素。

表观遗传学研究对许多领域都具有重要意义，包括癌症研究、神经科学以及对复杂疾病的理解（Egger 等（2004）；Feinberg 和 Vogelstein（1983）；Simó-Riudalbas 和 Esteller（2014））。在医学领域，人们正开发表观遗传疗法，试图通过控制基因表达或靶向特定的表观遗传标记，将其作为治疗癌症及其他疾病的潜在手段。

总的来说，表观遗传学加深了我们对基因、环境与健康之间关系的理解，并为研究和治疗干预开辟了新途径。同卵双胞胎就是一个很好的例子：他们拥有相同的DNA，却可能长得并不一样（van Dongen 等（2021））。这是因为在一生中，我们会有不同的经历，因此也会变得各不相同。在细胞中，存在许多机制来控制DNA的读取和表达。其中一些机制称为表观遗传修饰，可以在不改变DNA序列的情况下开启或关闭特定基因。DNA甲基化是最为人熟知、研究最广泛的机制之一，它发生于甲基添加到DNA碱基之时。DNA甲基化动态调控基因转录的能力，启发了本文所提出的方法。

4 形式化

现在，我们简要介绍了本文模拟的表观遗传操作；接下来，我们提出一种新的注意力机制形式化实现，能够很好地拟合/模拟DNA甲基化。

4.1 自强化注意力

有时，处理序列 $x = [x_1, x_2, \cdots, x_p]$ 的传统表示并不容易。处理 x v x $=$ $[v_i]_{i=1}^{d_v}$ d_v v x 的向量化表示通常更有用，该表示记作 ( )，其中 是向量的大小 ( )。

为了捕捉向量 v（x）中各个特征的相关性，我们计算一个表示向量 a，称为**强化注意力**（图 1）。强化权重 $a = [a_i]_{i=1}^{d_v}$ 的每个分量 a_i 表示特征 v_i 与 v 的相关性，且不依赖任何额外信息。因此，较大的值 a_i 会增强 v_i 的相关性，而较小的 a_i 值则会抑制特征 v_i。如图 1 所示，输出 $o = [o_i]_{i=1}^{d_v}$ 是根据计算所得的注意力 a 对向量进行强化后得到的结果，其定义为 a 与 v 的逐元素乘积：

$$o = [a_i \times v_i]_{i=1}^{d_v}. \tag{1}$$

为什么采用自强化注意力？在这种意义上，元素值向量 $[v_i]_{i=1}^{d_v}$会受到注意力权重 $[a_i]_{i=1}^{d_v}$ 的影响，从而提供许多可能的调控方案/表达形式。注意力权重向量 a 可以增强或抑制某些向量的

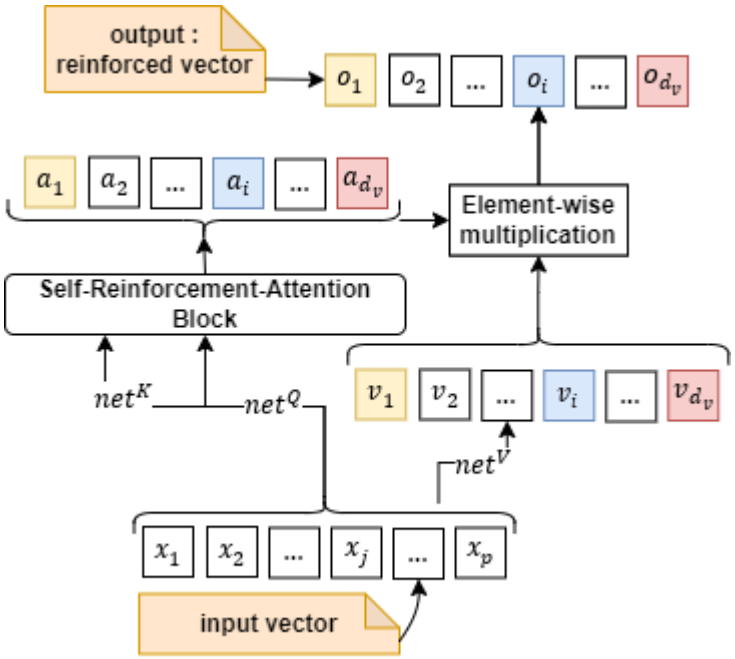

图1：自强化注意力

部分，并在关键的特定位置消除迭代元素冲突，修改/优化其输出函数。例如，要使函数最小化：

$$f(x) = x - 1, \tag{2}$$

如果受到注意力权重 $a^{(1)} = [0, \frac{10}{9}]$或 $a^{(2)} = [1, 0]$的影响，表示为值向量 $v\ (x) = [1, 9]$ 、维度为 $d_v = 2$ 的候选解 $x = 1.9$ 会更符合该函数。强化权重 a 可能表明，增强第一个分量/元素，同时抑制第二个分量/元素，从而得到候选解 $o^{(1)} = o^{(2)} = [1, 0]$。强化权重 a 提出了不同的发展路径，但值向量 v的表达稳定性保持不变。

这个例子证实了关于表观遗传学的研究，尤其是关于同卵双胞胎的研究：他们拥有相同的DNA序列 x ，但有两种不同的甲基化状态 $a^{(1)}$ 和 $a^{(2)}$ ，从而产生不同的表型 $o^{(1)}$ 和 $o^{(2)}$。另一个例子是求函数的根：

$$g(x) = x^2 - 3x + 2, \tag{3}$$

可写为：

$$g(x) = (x - 1.5)(x - 2.5), \tag{4}$$

并且有两个根 $x_1 = 1.5$ 和 $x_2 = 2.5$。

用值向量 $v\ (x) = [5, 5]$ 表示的候选根 $x = 5.5$ ，若受到注意力权重 $a^{(1)} = [\frac{1}{5}, 1]$或 $a^{(2)} = [\frac{2}{5}, 1]$的影响，便符合函数的两个根。此时

强化权重 $a^{(1)}$ 和 $a^{(2)}$ 再次提出两种不同的发育路径，使候选解分别到达与 $x_1 = 1.5$ 和 $x_2 = 2.5$对应的两个根 $o^{(1)} = [1, 5]$ 和 $o^{(2)} = [1, 5]$。因此，它们赋予候选根可塑性。这一机制与 Waddington 关于表观遗传学（可塑性）的观点相近（Noble（2015）；Waddington（1956）；见图 2）。

4.2 缩放点积自强化注意力

为了计算注意力权重 $a = [a_i]_{i=1}^{d_v}$，我们提出了一种名为"缩放点积强化注意力"的实现方法。输入由维度为 d_k 的 d_v 查询向量 $Q = (q_i)_{1 \le i \le d_v}$ 和 d_v 键向量 $K = (k_i)_{1 \le i \le d_v}$，以及维度为 d_v 的值向量 $v = (v_i)_{1 \le i \le d_v}$ 组成，所有这些都由输入向量 x 推断得出。我们计算查询向量 q_i 与键向量 k_i 的点积，并将每个点积除以 $\sqrt{d_k}$，以减弱数值过大的影响。随后，我们应用激活函数（例如 Sigmoid，.....）来获得强化权重 $A = (a_i)_{1 \le i \le d_v}$。这些步骤可用以下表达式描述：

$$a_i = f\left(\frac{q_i \cdot k_i}{\sqrt{d_k}}\right). \tag{5}$$

为了推断两个矩阵 K 和 Q 以及值向量 v，我们提出了三个神经网络 net^K、net^Q 和 net^v。它们可以采用任意架构；唯一需要满足的约束是输出形状。

如图 3 所示，我们实现的强化注意力与 Vaswani 等人（2017年）提出的传统缩放点积自注意力不同，因为它生成的是一个维度为 d_v 的向量，而不是矩阵。

4.3 多头自强化注意力

与其仅使用键向量、查询向量和值执行单一注意力函数，不如使用不同的查询向量Q^h和键向量 K^h 的学习向量化表示，执行 N 个并行函数，$1 \le h \le N$，同时保持值向量 v 不变，从而得到一系列 N 个强化向量 $O = [o^h]_{h=1}^{N}$。

这一考虑使一个候选解能够发展出不同的路径，从而利用注意力权重所提供的可塑性探索不同的可能性，如第 4.1 节公式 3 所示。

5 表观遗传学模型的贡献

由于文献将表观遗传学描述为一系列参与调控基因转录的机制和过程，这些机制和过程能够在不改变基因和/或染色体的情况下产生表型变异，因此，在人工生命中对其进行建模扩展了遗传算法，增加了一个用于调节评估的补充步骤

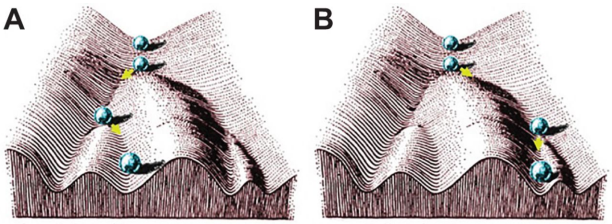

图 2：Waddington发育景观图，图片来源：Noble（2015）

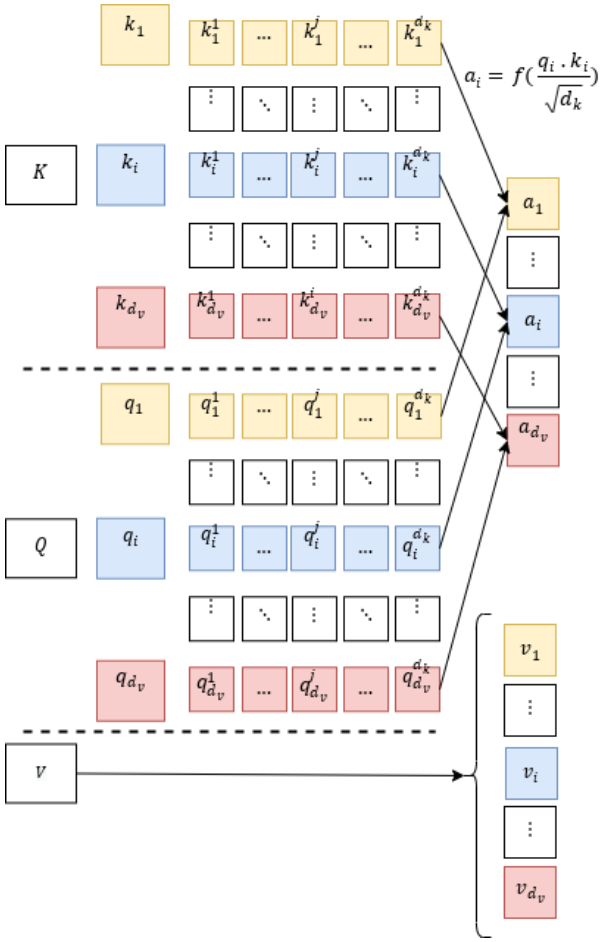

图 3：缩放点积强化注意力

一。算法1将表观遗传学算法作为遗传算法的扩展进行了介绍（第7、8和9行）。

算法1 表观遗传算法 EpiGeAl

1: **输入**： N_i, p, r_m, r_c, $DeepiGen$, $Params$。▷ 参数 = DeepGeni 参数（例如 d_k, d_v，注意力头数 N，...）
2: **输出**： $best$, $score$, $callback$ ▷ 回调 = DeepiGen 及参数的已保存副本 用于调节最佳解转录的。
3: $t \leftarrow 0$。 ▷ t = 代索引
4: $P(0) \leftarrow PopulationInitialization(N_i,\ p)$ ▷ P = 种群
5: $Q(0)$ $C(0) \leftarrow \emptyset$ ▷ Q = 辅助种群
6: **当不满足** 终止条件()**时**
7: $DeepiGenFit(P(t)\ DeepiGen)$
8: $Scores(t) \leftarrow Evaluation(P(t)\ DeepiGen, Params)$
9: $best, score, callback = FindBest(P(t)\ Scores(t)\ DeepiGen, Params)$
10: $Q\ (t)\ \leftarrow Selection\ (P\ (t)\ \ Scores\ (t)\ )$ ▷ Q 包含幸存基因
11: $C(t) \leftarrow Reproduction(Q(t))$ ▷ C 包含后代基因
12: $C(t) \leftarrow Mutation(C(t)\ r_m)$
13: $P(t+1) \leftarrow PopulationUpdating(C(t)\ Q(t))$
14: $t \leftarrow t+1$

5.1 表观遗传学模型卡顿

所提出的表观遗传学模型 DeepiGene 使用多个模块：染色体编码器和表型嵌入器，用于将序列转换为一个实值函数，该函数将染色体表示空间映射到适应度分数 $\mathbb{R}$。图 4 展示了所提出的 DeepiGen 的整体架构。设有一条染色体 x，它代表给定问题的一个候选解。它将经过两个主要模块：

- 染色体编码器模块（CEB）将 x 嵌入到一个维度为 d_v 的自持表征向量中；因此，它将染色体表征空间 X 映射到 $\mathbb{R}^{d_v}$：

$$\begin{aligned} CEB &: X \to \mathbb{R}^{d_v} \\ & x \mapsto x^s_{vec} \end{aligned} \tag{6}$$

因此，首先将染色体 x 嵌入到一个维度为 d_v 的向量 x_{vec} 中。这一步可以代表编码任务的一种潜在解。事实上，编码是一项与问题相关的任务，通常被认为至关重要，因为它会影响结果 [ref]；人们并不知道所提出的编码函数是否适合该问题。此外，这提供了一种自适应的解，因为它可以通过反馈进行学习。

其次，表示 x 自维持一致性的表观遗传信息被嵌入到一个与 d_v 维度相同的向量 e_{vec} 中：

$$\begin{aligned} EF &: X \to \mathbb{R}^{d_v} \\ & x \mapsto e_{vec} \end{aligned} \tag{7}$$

自维持表观染色体 x_{vec}^{s} 是 x_{vec}与 e_{vec} 的逐元素乘积:

$$x_{vec}^{s} = e_{vec} \times x_{vec} \tag{8}$$

这一表述遵循表观遗传学的基本原理，例如可逆性、“发生于.....之上”以及对结果的影响。

- 表型解码器模块（PDB）接收一个自维持表观染色体 x_{epi}^{s}，并返回一个解 s，该解符合优化问题的解空间 S:

$$\begin{aligned} PDB\colon \mathbb{R}^{d_v} &\to S \\ x_{epi}^{s} &\mapsto s \end{aligned} \tag{9}$$

随后，对 s 执行适应度评估步骤，以获得一个评分。

我们不必只考虑一种自维持一致性的可能性，还可以考虑由表观遗传学平衡产生的多种自维持一致性。因此，对于一个染色体 x，我们可以得到一个由 N 个表观遗传学自维持向量化表示 e 组成的序列:

$$e = [e_{vec}^{h}]_{1 \le h \le N} \tag{10}$$

其中 $e_{vec}^{h} \in \mathbb{R}^{d_v}$。染色体编码器模块（CEB）将返回一个由 N个自维持表观遗传染色体 $[x_{vec}^{h,s}]_{1 \le h \le N}$ 组成的序列，满足:

$$\begin{aligned} CEB_M\colon X &\to \mathbb{R}^{d_v \times N} \\ x &\mapsto [x_{vec}^{h,s}]_{1 \le h \le N} \end{aligned} \tag{11}$$

其中 $x^{h,}{}_{vec}^{s} = e_{vec}^{h} \times x_{vec}$。然后，表型解码器模块（PDB）将返回与这些表观染色体 $[x_{vec}^{h,s}]_{1 \le h \le N}$ 相对应的 N 个潜在解 $[s^{h}]_{1 \le h \le N}$。最后，我们建议在评估每个解的适应度后应用最优池化。

这一考量使染色体 x能够呈现不同的表达形式 $s = \{s^{1}, \cdots, s^{h}, \cdots, s^{N}\}$。后者可以沿着不同的允许轨迹变化，从而产生符合局部约束的不同结果或解。

5.2 自强化注意力在我们的表观遗传学模型中的应用

如前所述，我们的 DeepiGen 在染色体编码器模块（CEB）中采用了所提出的自强化注意力。事实上，自维持一致性 e_{vec} 可以视为强化权重 a （参见 4.1）。随后，可以使用缩放点积自强化注意力对其进行嵌入。因此，自维持表观遗传染色体 $[x_{vec}^{h,s}]_{1 \le h \le N}$对应于 $O = [o^{h}]_{h=1}^{N}$中的强化向量。

强化注意力机制旨在捕捉染色体中基因之间的不相容性，并调节染色体中编码的信息，从而模拟生物学中的自我调节过程。

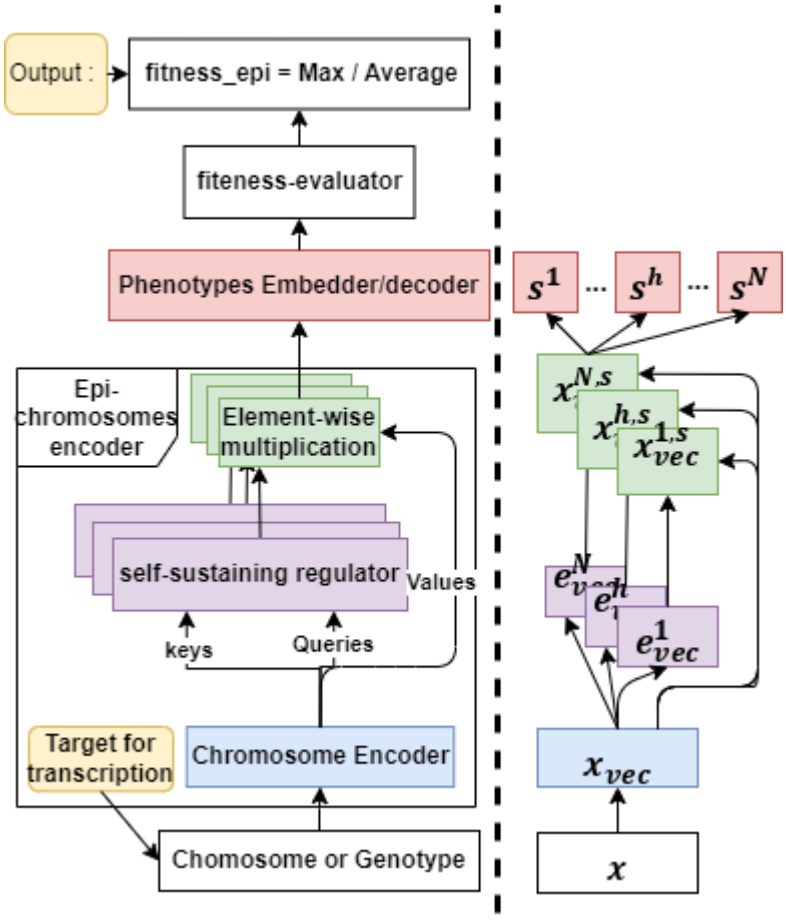

图 4：将表观遗传学视为自我维持调控算子的拟议架构

5.3 表观遗传学模型的训练策略

我们选择训练表观遗传学模型 DeepiGen，以优化目标函数，即最大化种群得分。具体而言，就是使用以下损失函数训练神经网络：

$$\mathcal{L} = \sum_{n=1}^{N_i} \frac{1}{score[n]} \tag{12}$$

对于多头强化注意力，我们添加扩散约束，以强制头间离散。此类现象的发生将通过扩散率 r_{diff} 加以控制。因此，在某些代中，我们将最大化种群中每条染色体所关联的所有转录的方差。

5.4 框架属性

我们的方法具有以下几项属性：

- 智能染色体编码：所提出的方法通过模块（PDB）自动完成解码阶段。此外，借助神经网络进行可学习编码，所提出的算法为一个经典老问题提供了出色的答案：如何编码染色体？因此，我们可以对所有问题使用二进制编码。

- 探索与开发之间的平衡：经典遗传算法需要通过变异率进行巧妙的探索；较高的变异率会导致系统不稳定，无法充分开发已探索区域。在我们提出的算法中，神经网络 DeepiGen 专注于开发新的潜在区域，这有助于实现纳什均衡。

6 实验

遗传算法的局限之一是，在非凸问题中，它们往往会收敛到局部最优解甚至任意点，而非全局最优解。为研究我们的方法的表现，我们决定聚焦于多峰优化问题。

6.1 多峰问题的优化：收敛性与动态特性研究

为了研究所提出的表观遗传学框架及其在非凸问题上的动态过程的收敛性，我们提出在两个著名的多峰优化问题上进行测试：BUMPY Keane（1995）和 STALAGMITE 函数**?** 。

6.1.1 BUMPY 函数

Keane（1995）是多峰优化问题之一，其产生一系列峰值，峰值会随着与原点距离的增加而逐渐变小。其目标函数可定义为取值 $(x,\ y)$ 在 $\mathcal{D}_B =]0,\ 10]^2$ 时：

$$f(x,y) = \frac{sin^2(x-y)sin^2(x+y)}{\sqrt{x^2+2y^2}}$$

BUMPY 问题仅取决于两个变量 x 和 y，这有助于展示该函数，并在第 6.4 节对 DeepiGen 的结果进行定性讨论。图 5 显示，真正的最大值 0.675 出现在 (1.393,0.006)，但该问题还具有一些其他特性，使其颇难求解：

- 首先，该问题关于 $x = y$具有相当强的对称性，因此在 (0.031,1.441) 处存在一个高度为 0.47 的相似峰值；如果算法陷入这个局部最优解，该峰值就可能构成相当大的竞争。

- 接下来的两个峰值分别为 0.365 和 0.274，分别位于 (1.593,0.471) 和 (0.475,1.578)。这两个峰值的竞争力都不如第一个。

因此，首要挑战是测试我们的方法能否达到高度为 0.675、位于 (1.393,0.006) 的全局最优解，即便初始种群经过设计，刻意偏向位于 (0.031,1.441)、高度为 0.47 的局部最优解；见图7。

BUMPY 函数面临的第二个挑战，是评估采用多头架构 4.3 所带来的影响。也就是说，我们想探究：其他注意力头都在做什么？它们会探索全局最优解以外的其他解，还是只会收敛到同一个最优解?

6.1.2 STALAGMITE 函数

? 于 2020年作为 SKILLLYNC 学员的多峰优化问题提出。其灵感来自一些洞穴地面上形成的石笋结构，可在定义域 $\mathcal{D}_S = [0,1]^2$ 上定义为四个函数的乘积：

$$f(x,y) = f_{1,x}f_{2,x}f_{1,y}f_{2,y}.$$

其中，

$$f_{1,x} = [sin(5.1\pi x + 0.5)]^6,$$
$$f_{1,y} = [sin(5.1\pi y + 0.5)]^6,$$
$$f_{2,x} = exp[-4ln(2)\frac{(x-0.0667)^2}{0.64}],$$
$$f_{2,y} = exp[-4ln(2)\frac{(y-0.0667)^2}{0.64}],$$

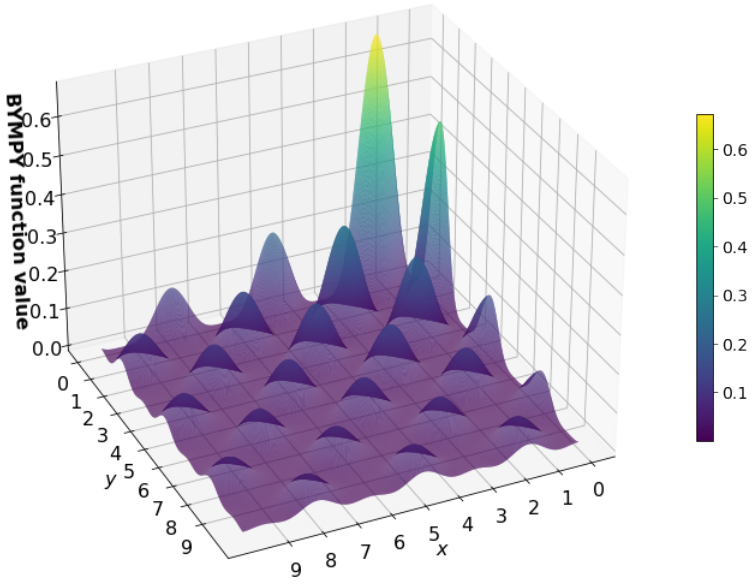

图 5：BUMPY 函数生成的曲面 $0 < x,\ y \leq 10$

显而易见，STALAGMITE 函数是对称的，即，

$$\forall (x, y) \in \mathcal{D}_S, f(x, y) = f(y, x).$$

图 6 显示，全局最大值 1 出现在 (0.06，0.06) 处，但也展现了石笋问题的另一项特性：在 (0.262，0.067) 和 (0.067，0.262) 处还存在两个高度为 0.845 的峰值；如果算法陷入其中一个局部最优解，这两个峰值便可能颇具竞争力。

因此，这一特性引出了一个问题：如果数学约束迫使算法忽略位于 (0.06，0.06) 处、高度为 1 的全局最优解，我们的算法会有怎样的表现？由于可能存在两个局部最优解，算法在寻找新的全局最优解时，是否会受到种群初始化的影响（旧的局部最优解为位于 (0.262，0.067) 或 (0.067，0.262) 处、高度为 0.845 的解）？

6.2 实验概要

为确保我们的方法能够妥善应对上述四项挑战，我们进行了四项实验：

- 实验1（全局最优解）：我们测试了基本模型（即一个自强化注意力机制）达到全局最优解的能力。

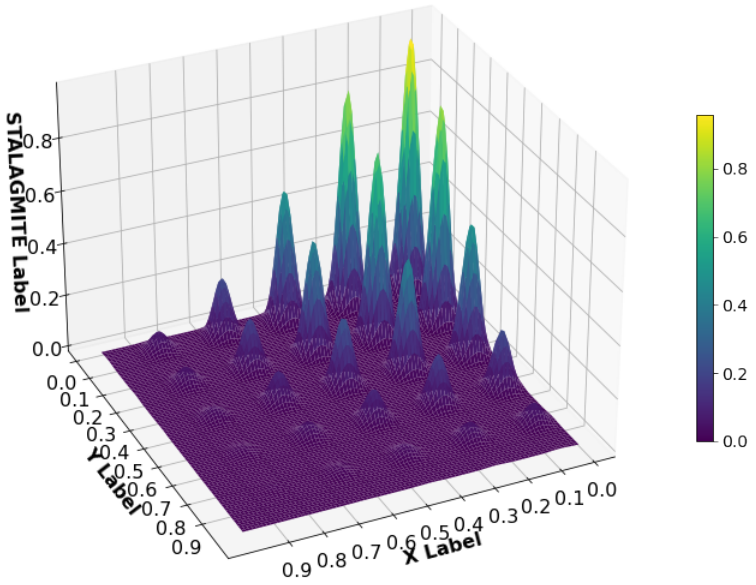

图6：由 STALAGMITE 函数生成的表面 $0 \leq x,\ y \leq 1$

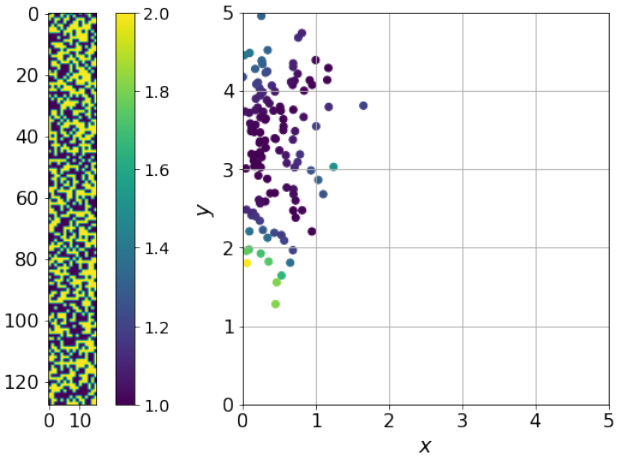

图7：实验所考虑的初始种群

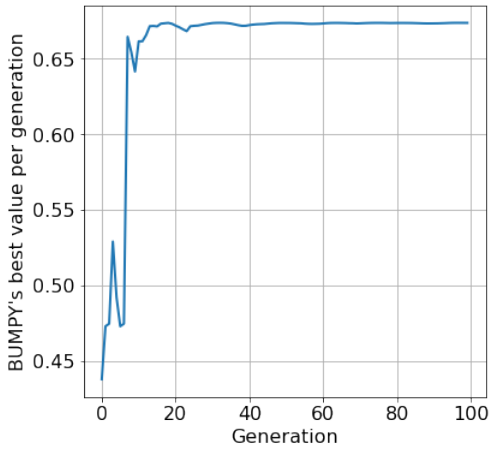

图 8：BUMPY 函数每代找到的最佳值的进化

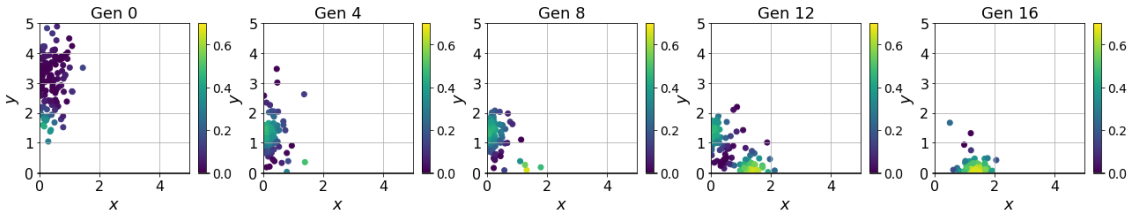

图 9：种群经过多代迁移，最终到达 BUMPY 函数位置 (1.39,0.006) 处高度为 0.675 的全局最优峰

注意力头。为此，我们针对 BUMPY 和 STALAGMITE 函数测试了不同的基准测试配置。最初特别关注种群，目的是有意促使算法偏向局部最优解。

- 实验2（双胞胎）：我们测试了采用多注意力头自强化注意力架构的影响。换句话说，算法能否找到不止一个最优解？也就是说，不同的注意力头是否会关注不同的区域和／或特征?

6.3 参数化

我们的 DeepiGen 有十个自由度：首先是遗传算法中广为人知的五个经典参数：

1. N_i ： 种群规模。2. p 染色体的维度。
3. r_{mut} ： 变异率。4. r_c 交叉率。5. nb_{iter}
最大迭代次数。

除上述五个参数外，我们提出的模型 DeepGeni 还包括：

6. d_k ： 键和值查询编码器的输出维度。7. $embed$ ： 用于确定是否嵌入染色体的布尔值。8. d_v ： 值嵌入器的输出维度（注意：当 $embed == False$时，$d_v = p$ 和 net^V 为恒等变换）。9. N：DeepGeni 模型中的注意力头数。10. r_{diff} ： 控制头间离散的扩散率。

对于 6.2 中介绍的两项主要实验，我们测试了 1000 种不同的配置，并调整这些参数，以确保所提出算法的稳健性。因此，对于经典遗传算法参数：种群规模参数 N_i取值为 $[32, 64, 128, 256, 512, 1024],$；染色体维度 p取值为 $[8, 16, 32],$；变异率 r_{mut}在 0.05 到 0.3 之间变化，步长为 0.01；交叉率 r_c 取值为 $[0.7, 0.8, 0.9]$ ，以允许部分染色体存活；最后，最大迭代次数设为 $nb_{iter} = 100$。

此外，对于 DeepiGen 的参数：查询向量和键向量的维度 d_k 在所有 $embed == True$ 情况下均取值于 $[16, 32, 64],$ ；值向量的维度 d_v 取值于 $[8, 16, 32, 64]$。重要参数是自强化注意力头的数量 N，其取值于 $[1, 8, 16]$。因此，扩散率

r_{diff} 在所有 $N > 1$ 中取值范围为 0.05 至 0.3，步长为 0.05。
然而，正如第 5.4 节所讨论的，本文仅考虑了二进制编码。实数编码被假定由值编码器 net^v 隐式实现。最后需要测试的是种群初始化对所得结果的影响，因此我们特意考虑了较难的情形；例如，针对 BUMPY 函数和 STALAGMITE 函数分别采用的初始种群都偏向于局部最优解。

6.4 结果与讨论

6.4.1 实验1

在测试的一千种（1000）配置中，所有配置都成功找到或达到 BUMPY 函数和 STALAGMITE 函数的全局最优解。此外，考虑到对全局最优解施加惩罚的约束，我们的算法轻松找到了次优解。

本节选择展示 BUMPY 函数的所得结果，所采用的配置如下：初始化种群包含 $N_i = 128$ 个维度为 $p = 16$ 的染色体，变异率为 $r_{mut} = 0.1$，交叉率为 $r_c = 0.7$；键与查询的输出维度取值为 $d_k = 32$ 。此外，不使用值嵌入，因此 $embed = False$，且仅使用一个注意力头 $N = 1$。
图7左侧显示了 $N_i = 128$ 个染色体的初始种群，右侧显示了相应的解码值（x， y）。可以看到，它们都位于位置 (0.47, (0.031，1.441) 处高度为 0.47 的第二个峰附近，因此所有凸优化方法（例如梯度下降）都有陷入这个局部最优解的风险。图 8 展示了最佳最优值随代数的演变。可以看到，我们的算法在第十六代 $t = 16$ 达到了全局最优解。实际上，该算法从第八代 $t = 8$ 起便开始考察全局最优解所在的这个凸区域。

问：训练过程中发生了什么?
正如预期的那样，如图 9 所示，由于优化步骤 (7)，模型在第一代就达到了局部最优解。在第四代 $t = 4$，模型开始探索全局凸区域；到第八代（$t = 8$），解码解与接近全局最优解的一个解相吻合 $s_{opt} = 0.67$。发现这一点后，模型在接下来的几代中专注于全局最优解附近的这个最优凸区域。因此，惯性逐渐减小，所有解都收敛到全局最优解。

值得注意的是，有时会发生变异，并试图探索远离中心值的区域。

6.4.2 实验2

考虑使用多个注意力头进行注意力机制，从而增强或抑制某些基因，使算法能够考虑与同一条染色体相关的多个最优值，如第 4.1 节中针对 BUMPY 函数的情况所述。当然，这取决于

注意力头所关注的区域。

确实，当头数为 $N = 16$ 、扩散率为 $r_{diff} = 0.3$ 时，我们的算法完美模拟了同卵双胞胎中观察到的表观遗传学现象；对于同一条染色体，两个不同的注意力头成功关联到／抵达了两个不同的最优峰值。

图10展示了 BUMPY 函数的最优值随代数的进化过程；可以看到，在第 $t = 36$ 代，不同的注意力头聚集成三个主要簇，分布在最优峰值和次优峰值附近；并且如第 $t = 48$ 代所示，它们继续探索其他区域。因此，对于二进制编码的染色体：

$$x = [0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 0.0]$$

染色体编码器模块（CEB_M）包含多个
自强化注意力头，返回多个自维持表观遗传染色体（自强化向量） $[x_{vec}^{h,s}]_{1 \le h \le 16}$。训练后，我们的（$CEB_M$）中的两个注意力头 $h = 9$ 和 $h = 15$ 针对给定的染色体 x，生成两个自维持表观遗传染色体：

$$\begin{aligned} x_{vec}^{9,s} = [&0.89, 1.06, 0.41, 0.74, 0.7, 0.71, 0.78, 0.43, \\ &0.45, 1.3, 0.77, 0.46, 0.84, 1.73, 1., 0.51], \end{aligned}$$

and

$$\begin{aligned} x_{vec}^{15,s} = [&0.66, 0.34, 1.74, 0.56, 0.68, 1.22, 0.43, 0.46, \\ &1.37, 1.26, 0.42, 0.78, 0.68, 0.64, 0.53, 1.25], \end{aligned}$$

如图11所示，训练后的表型解码器模块（PDB）返回的两个解码染色体如下：

$$PDB(x_{vec}^{9,s}) = (1.426, 0.036)$$

and

$$PDB(x_{vec}^{15,s}) = (0.166, 1.354)$$

从图11以及数值 $x^{9,\ s}_{vec}$ 和 $x^{15,\ s}_{vec}$可以看出，要激活第一个解（1.426，0.036），位置13上的基因被增强，而位置2、7和8上的基因被沉默。类似地，要激活第二个解（0.166，1.354），位置2上的基因被增强，而位置6、7、10和14上的基因被沉默。另一方面，在考虑 STALAGMITE 函数并施加迫使其忽略全局最优解的约束后，该函数成功找到了高度为 0.845 的两个对称解，其位置分别为 (0.265，0.065) 和 (0.065， 0.265)，但它们是通过两个不同的注意力头 $h = 5$ 和 $h = 13$，使用了两条不同的染色体。

实验2所得结果证实了图2所示的 Waddington发育景观图。

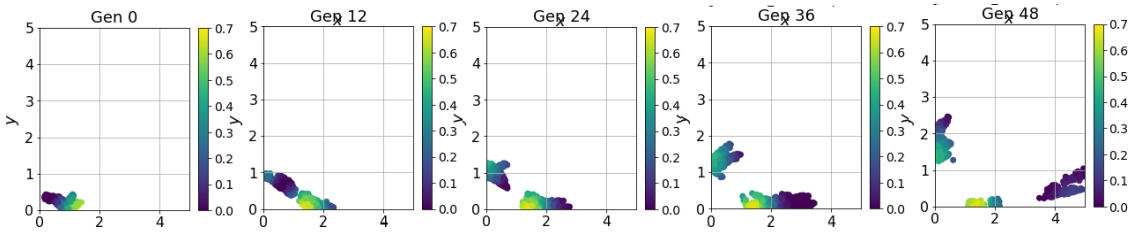

图10：在 BUMPY 函数中，种群经过多代迁移，分别到达位于 (1.39,0.006) 和 (0.365,0.274) 的高度分别为 0.675 和 0.475 的多重最优峰



图11：针对同一条染色体，注意力头8和15分别提出的两个注意力数组，使其在 BUMPY 函数中分别到达近似全局最优解 0.67 和近似次优解 0.44，位置分别为 (1.426,0.036) 和 (0.166,1.354)

7 结论与当前工作

本文提出了新的表观遗传算法：利用一种新颖的注意力机制对遗传算法（遗传算法）进行改进。该改进模拟了 DNA甲基化的表观遗传过程（增强/沉默）。我们的这一方法的主要贡献之一是采用自监督方式；解码阶段和表观遗传操作均可学习。这使我们不再纠结于如何编码信息这一由来已久的问题。随后，在讨论注意力与表观遗传机制之间的相似性后，最后一节中的实验表明，所提出的改进使算法能够高效地拟合多峰优化问题。通过在几代之后达到全局最优解，算法效率得以提升，同时降低了陷入局部最优解的风险。此外，使用多个注意力头表明，同一染色体可根据所在区域表示多个最优解。

此外，本文所呈现的结果属于一项研究的一部分，该研究探讨了利用注意力机制在遗传算法中模拟表观遗传过程的可能性。作为研究的一部分，我们计划从理论上研究该方法的收敛性，并考察在各种遗传算法中整合其他模拟表观遗传过程的改进方法；因此，我们也在开展有关开发一种考虑外部因素的表观遗传算法的研究。

参考文献

Bannister, A. J. and Kouzarides, T. (2011). 组蛋白修饰对染色质的调控。《细胞研究》, 21(3):381–395. Egger, G., Liang, G., Aparicio, A., and Jones, P. A. (2004). 人类疾病中的表观遗传学及表观遗传疗法的前景。《自然》, 429(6990):457–463. Ezzarii, M., Ghazi, H. E., Ghazi, H. E., and Bouanani, F. E. (2020). 基于表观遗传算法的网络攻击检测技术。《*IEEE Access*》, 8:199482–199491. Feinberg, A. P. and Vogelstein, B. (1983). 低甲基化可区分某些人类癌症中的基因与其正常对应基因。《自然》, 301(5895):89–92. Huang, Y., Sun, X., Jiang, H., Yu, S., Robins, C., Armstrong, M. J., Li, R., Mei, Z., Shi, X., Gerasimov, E. S., De Jager, P. L., Bennett, D. A., Wingo, A. P., Jin, P., Wingo, T. S., and Qin, Z. S. (2021). 一种脑部表观遗传分析机器学习方法揭示了与阿尔茨海默病相关的激酶。《自然·通讯》, 12(1):4472. Jablonka, E. and Raz, G. (2009). 跨代表观遗传遗传：流行程度、机制及其对遗传与进化研究的意义。《生物学季刊评论》, 84:131 – 176. Keane, A. (1995). 多峰问题的遗传算法优化：对收敛性和稳健性的研究。《工程中的人工智能》, 9(2):75–83. Noble, D. (2015). Conrad Waddington与表观遗传学的原点。《实验生物学杂志》, 218(6):816–818. Periyasamy, S., Gray, A., and Kille, P. (2008). 表观遗传算法。载于*2008*年 *IEEE*进化计算大会（*IEEE*计算智能世界大会）, pages 3228–3236. Simó-Riudalbas, L. and Esteller, M. (2014). 靶向癌症的组蛋白修饰模式：针对写入酶、擦除酶和读取蛋白的药物。《英国药理学杂志》, 172(11):2716–2732. Sousa, J. A. B. and Costa, E. (2010). 在人工生命中设计一种表观遗传方法：EpiAL模型。载于$ICAART$。Tanev, I. and Yuta, K. (2008). 表观遗传编程：通过组蛋白修饰将表观遗传学习融入遗传编程。《信息科学》, 178(23):4469–4481. 包含特别专题：遗传与进化计算。van Dongen, J., Gordon, S. D., McRae, A. F., Odintsova, V. V., Mbarek, H., Breeze, C. E., Sugden, K., Lundgren, S., Castillo-Fernandez, J. E., Hannon, E., Moffitt, T. E., Hagenbeek, F. A., van Beijsterveldt, C. E. M., Jan Hottenga, J., Tsai, P.-C., Hottenga, J.-J., de Geus, E. J. C., Spector, T. D., Min, J. L., Hemani, G., Ehli, E. A., Paul, F., Stern, C. D., Heijmans, B. T., Slagboom, P. E., Daxinger, L., van der Maarel, S. M., Willemsen, G., Montgomery, G. W., Reversade, B., Ollikainen, M., Kaprio,

J., Spector, T. D., Bell, J. T., Mill, J., Caspi, A., Martin, N. G., Boomsma, D. I., Consortium, B. I. O. S., and of DNA甲基化联盟, G. (2021年). 同卵双胞胎携带早期基因组编程的持久表观遗传特征. 《自然·通讯》, 12(1):5618. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017年). 注意力机制就是一切. $CoRR$, abs/1706.03762. Waddington, C. H. (1956年). 双胸表型的遗传同化. 进化, 10(1):1–13. Wolffe, A. P. and Matzke, M. A. (1999年). 表观遗传学：通过抑制进行调控。《科学》, 286(5439):481–486.