

当热噪声开始干活：一篇关于密码子优化、伊辛模型和热力学硬件的长文

副标题

把一段蛋白质序列改写成更适合宿主表达的 mRNA，看上去像生物学问题；在这篇新论文里，它被改写成了一个能量地形上的采样问题。真正有意思的，不只是结果能不能算出来，而是计算机到底该把热噪声当成敌人，还是当成工具。

核心一句话

这篇论文最重要的洞见是：当一个问题本质上是在巨大的组合空间里找低能量解时，我们也许不必在传统 GPU 上“强行算完”，而可以让物理系统自己朝低能态松弛，再把这个过程直接当作计算。

想象你在给一段中文改写成更适合某个平台发布的版本。意思不能变，但有些词太生硬，有些句式太拗口，有些标点和长度还得照顾平台偏好。你不会只问“同义词对不对”，而是同时在意读起来是否顺、结构是否紧凑、整体风格是否一致。密码子优化就是这种任务的分子版：蛋白质序列不变，但编码它的 mRNA 可以有很多“同义写法”，而不同写法在宿主细胞里并不等价。

这正是 [Andraz Jelencic](#) 和 [Ross C. Walker](#) 的新论文想抓住的地方。它不是在问“密码子优化能不能做”，这个问题早就有人在做；它问的是另一层更尖锐的问题：**如果这个任务本来就是一个组合优化问题，那为什么还要默认用 GPU 上的通用算法硬算，而不能让专门为概率采样设计的热力学硬件来做？**

这听起来像是一个硬件工程话题，实际上却先碰到了生物学的基本事实。自然界的遗传密码是“冗余”的：61 个编码氨基酸的密码子，对应 20 种氨基酸。也就是说，很多氨基酸都有多个同义密码子可选。可“同义”并不等于“同样好用”。宿主细胞更偏爱某些密码子，另一些则更少见；这会影响翻译速度、mRNA 稳定性，进而影响蛋白表达量。关于这一点，综述文献早就说得很清楚：密码子偏好不是语法上的细枝末节，而是能细调表达水平的生物学旋钮。[Quax 等人](#)的综述和更早的 [Gustafsson 等人](#)综述都强调过，所谓“同义密码子”并不真的同义。

于是，密码子优化的问题就出现了：给定一条蛋白质序列，我们要从每个位置的同义密码子集合里选出一个，让最终的 mRNA 满足几个往往彼此拉扯的目标。论文采用的是一个很典型、也很适合讲清楚的版本：一方面希望密码子尽量贴近宿主常用频率，另一方面希望整体 GC 含量接近目标值，还要避免长串重复碱基。换句话说，这不是单一目标的“找最优”，而是多个约束之间的折中。

如果只用生物学语言描述，这像是在给一句话做风格润色，同时要求它读起来顺口、不要太多重复、还要符合平台的长度要求。可一旦把它翻译成数学语言，问题就变了。每个位置都不是一个连续变量，而是一个离散选择；每个选择又会和邻居、全局统计量发生关系。这样的对象

非常适合写成能量函数：

$$E(c_1, \dots, c_L) = w_f \sum_{p=1}^L u(c_p) + w_R \sum_{p=1}^{L-1} r(c_p, c_{p+1}) + w_{GC} \left(\frac{1}{3L} \sum_{p=1}^L g(c_p) - \rho_T \right)^2$$

你不必盯着公式本身看太久。它表达的其实是三件事：第一，某些密码子在宿主中更常见，偏离它们要付出代价；第二，相邻密码子之间的重复模式会被惩罚；第三，整体 GC 比例不要偏离目标太远。这一步很关键，因为一旦你把问题写成“找最低能量的状态”，很多物理和算法工具就突然变得可用了。

从“算答案”到“让系统自己沉下去”

传统计算机处理这类问题，往往是靠迭代搜索。可以是遗传算法，可以是模拟退火，也可以是更复杂的启发式优化。它们都在做一件事：在庞大的组合空间里来回试探，试着别卡在局部最优里。这个过程本身不奇怪，奇怪的是我们通常把“热噪声”视为干扰，必须尽量压下去；而在概率采样和退火问题里，热噪声恰恰是帮助系统跳出局部谷底的资源。

这就是热力学计算的直觉起点。论文把自己放在这个框架里：不是让芯片尽可能安静、尽可能确定，而是让它在给定能量函数下自然采样，让物理系统朝 Boltzmann 分布靠近。与其说它是在“执行算法”，不如说它是在“模拟一个会自动寻找低能态的物理过程”。

这个想法并不新。早在 [Thermodynamic Computing](#) 这类研究里，研究者就已经把计算想成一种受控的热物理过程。它的核心不是“更快地做同样的事”，而是“换一种更符合任务本质的方式做事”。如果目标本来就是抽样、优化、近似推断，那计算机不必每一步都和噪声搏斗；它可以借噪声一臂之力。

这也解释了为什么这篇论文会把目光放到药物研发。药物研发里有大量问题并不是“精确算一个闭式解”，而是“在巨大空间中找足够好的方案”。构象搜索、分子设计、结合亲和力估计、蛋白序列优化，很多都和能量地形、离散选择以及概率采样有关。密码子优化尤其合适，因为它既真实，又足够简单，能把“问题如何映射到硬件”这件事讲得很干净。

先别谈硬件，先看问题长什么样

论文选的例子是 SARS-CoV-2 刺突蛋白，长度 1273 个氨基酸，对应 3819 个 mRNA 核苷酸。作者用大肠杆菌 K-12 作为宿主背景，优化目标则包括三部分：密码子使用偏好、GC 含量和重复惩罚。这个组合不是随便挑的，它基本对应了很多实际 codon optimization 流程中最常见的一类指标。

这里有一个容易被忽略的事实：密码子优化不是在“找唯一正确答案”，而是在“在许多彼此近似的候选里找一个更合适的折中”。这意味着问题天然就不是线性的，也不适合用单一评分维度理解。把它写成能量函数后，很多局部与全局的关系才清晰起来：单个位置上的 codon

usage 是局部项，重复惩罚是相邻项，而 GC 含量是全局项。局部与全局在同一个目标函数里打架，这正是组合优化的典型特征。

在传统 GPU 语境里，这类任务往往由遗传算法、模拟退火或者一些定制启发式方法处理。论文里也确实把一个经过仔细调参的遗传算法作为基线，避免“拿一个明显不公平的对手来比较”。这点很重要，因为很多硬件故事最容易犯的毛病，就是把对照组设得过弱。这里作者至少在形式上承认了：如果你要比较的是能耗，那先要让解的质量站在同一水平线上。

结果是，三种方法在解的质量上都差不多。论文报告，在标准参数下，Potts 模型采样、Ising 模型采样和遗传算法得到的分数都大致落在 234 到 240 的区间内；在更难参数设置下，分数则都上移到 444 到 453 左右，但彼此仍然接近。**这意味着比较的重点不再是“谁能找到唯一更好的解”，而是“谁用更少的代价找到同样好的解”。**

这也是整篇论文真正的转折点。因为一旦质量相近，能耗就不再是附属指标，而成了主角。

Potts 模型：让每个位置先变成“多选题”

数学上最自然的表达方式，是把每个氨基酸位置看成一个 Potts 变量。你可以把它理解成多状态离散变量：不是二元开关，而是“从几个同义密码子里选一个”。每个位置的状态数不一样，有的氨基酸只对应一个密码子，有的则可能有二、三、四甚至六个选择。

这种表示法有一个好处：它和问题本身的结构贴得很近。每个位置就是一个 categorical variable，相邻位置之间通过重复惩罚产生耦合，整个图结构像一条链。链结构很朴素，但正因为朴素，反而适合快速采样。对一条链来说，奇偶位置可以分块更新：给定一侧，另一侧条件独立。这就是所谓的 block Gibbs 更新。**换句话说，问题在图上越“瘦”，采样就越容易并行化。**

论文中所谓的 Potts 硬件路径，就是直接在这种多状态表示上做采样。理论上这最贴近原始问题，因为你不需要先把多选题拆成若干二选一再绕回来。作者的结果也支持这一点：Potts 采样所需迭代步数最少，最终得分略优，而且变量数也最少。对 1273 个氨基酸的位置，它就是 1273 个多状态变量，结构直观，代价也直接。

可是问题来了：真实存在的热力学采样单元，近中期更可能先是二元的 p-bit 硬件，而不是直接支持多状态的 p-dit 硬件。也就是说，最自然的表示法并不一定最容易落到芯片上。于是论文转向了另一条更工程化的路：把 Potts 模型再编译成 Ising 模型。

Ising 模型：把多选题拆成一串二选一

Ising 模型是二元自旋模型，每个变量只能取两种状态。很多物理和优化硬件都更容易实现这种形式，因为它只需要处理“向上/向下”这类二态随机变量。问题是，密码子选择不是二态，而是多态。怎么把多态装进二态硬件里？

论文采用的是 domain-wall encoding，中文里可以直译成“域壁编码”或“温度计编码”。直觉上，它不是让每个状态都分配一个独立开关，而是让一串自旋像温度计一样排列：前面若干个是 +1，后面若干个是 -1，二者之间的那个边界位置就代表状态编号。状态不是由“哪一个灯亮了”决定，而是由“亮灭的分界线在哪儿”决定。

这个比喻非常有用，因为它揭示了 domain-wall encoding 的关键优势。最常见的 one-hot 编码像是在几盏灯里只允许一盏亮，约束需要一个很密的惩罚团；而域壁编码像是在一条链上找分界线，约束只要邻近相连就够了。**前者更像“整齐的名单”，后者更像“找边界”，而边界天生是局部的。**

这件事的重要性不只是节省变量，更重要的是图结构更稀疏、可并行性更高、混合速度更好。论文引用并利用了 domain-wall encoding 的既有理论工作，尤其是 [Chancellor 2019](#) 以及 [Berwald、Chancellor 和 Dridi 2021/2022](#) 的研究。这些工作证明了，至少在只允许二次相互作用的前提下，域壁编码在变量数上已经接近最优；同时，相比 one-hot，它的约束图更稀疏，也更利于采样。

对这篇论文而言，domain-wall encoding 不是抽象装饰，而是工程落点。因为热力学硬件目前更接近 p-bit，而 p-bit 本质上更适合二态采样。于是，Potts 问题先被写成 Potts，再被编译成 Ising，再在热力学芯片上运行。这个“翻译链”听上去绕，但每一步都在服务一个现实约束：**让问题适配硬件，而不是幻想硬件自动适配所有问题。**

这里还有一个更细微但很关键的点。域壁编码里的单比特翻转，不只是翻转状态，它实际上是在相邻状态间挪动边界。这意味着局部移动在语义上对应“换一个相邻密码子选择”，而不是把整个位置从头乱改一遍。作者还刻意按 GC 含量给密码子排序，使得一次局部翻转对 GC 的影响尽量平滑。这个设计很像在给能量地形铺一条更顺的路：每一步都别跳得太猛，搜索会更稳定。

为什么 GC 约束必须“变通”处理

论文里最值得注意的工程技巧之一，是对 GC 含量项的处理。理论上的 GC 目标是一个全局平方惩罚：所有位置加起来以后，整体 GC 比例不要偏离目标太远。可这样的全局耦合如果直接展开，会变成大量全连接相互作用，而这和热力学芯片偏好的稀疏局部连接不合。

作者的解决办法是用一个自适应线性项去近似它：根据当前链的 GC 偏差，动态调整一个系数，让系统在 annealing 过程中逐步把整体 GC 往目标推。你可以把它理解成：不把全局目标一次性钉死，而是在每轮采样后看一眼偏差，再把“朝哪个方向压”的力度改一改。

这一步体现了硬件计算里一个常被低估的原则：**好的映射不是把原始公式原封不动搬进芯片，而是找一个在硬件上等价、或者至少足够近似的实现方式。**如果全局项会毁掉图结构，那就不能只因为“数学上更漂亮”而硬塞；你得找到一个保持问题精神、但不破坏硬件可执行性的替代项。

这类变通并不会神秘地让问题消失。它只是把一个难以直接实现的约束，改写成在退火过程中逐步生效的偏置。论文也没有把这说成理论上完全等价的变换，而是诚实地把它叫作自适应近似。这个克制很重要，因为它区分了“我已经精确重写了问题”和“我找到一个足够好、可落地的工程近似”。

三种方法，谁更好？

论文比较了三种方案：直接的 Potts 采样、经过域壁编码的 Ising 采样、以及一个经过调参的遗传算法基线。它们的解质量都在可接受且相近的范围内。换句话说，**这不是“新方法终于算出了旧方法算不出的答案”，而是“大家都能算到差不多的答案，但算的代价完全不同”。**

在标准参数下，Potts 采样通常只需要 10 次 sweep，Ising 采样则要 2×10^4 次，遗传算法则是 200 个个体、1000 代。表面上看，Potts 与 Ising 之间差了两个数量级的迭代步数；但如果把硬件纳入考虑，差距会被进一步放大。论文把单次 Ising 采样单元的能耗估成约 1.3 飞焦每个自旋每个 Gibbs 步，按 3147 个自旋和 2×10^4 步计算，总能耗大约是 82 nJ。

Potts 芯片还没有真正落地，所以作者只能用基于原型测量外推出来的模型估算：对于 1273 个节点、10 次 sweep，大约是 0.084 nJ。这个数字很漂亮，但它必须被正确理解为“投影值”，不是“已在商用芯片上实测的值”。论文在这点上说得很清楚：Ising 路径是更近的现实，Potts 路径是更自然但更远的未来。

和 GPU 的比较才是最戏剧化的部分。作者给出两种估算：一种是按峰值 FLOP 的理想下界，一种是按实际运行时间乘 TDP 的测量上界。无论按哪种口径，热力学芯片都明显更省电。对 Potts 采样来说，理想化 GPU 下界大约是 13 微焦耳，而测量口径可到 245 毫焦耳；对 Ising 采样来说，数值则大约是 20 毫焦耳到 84 焦耳。作者据此给出总量级结论：相对于常规 GPU 和遗传算法组合，节能幅度可达到 10^5 到 10^9 倍这个级别。

这里不要被“十亿倍”这种数字直接带跑。真正值得看的是两个层次。第一，优化质量并没有因为换硬件而明显下降，这说明问题映射没有把生物目标扭曲得太厉害。第二，硬件形态本身足以造成数量级巨大的能耗差异，这说明很多时候“算法效率”并不是全部，**真正的杠杆在于算法和硬件是否匹配。**

这篇论文最该被记住的，不是一个炫目的数字

如果只记住“省电很多”，你会错过这篇工作的真正价值。更重要的其实有三点。

第一，它把一个真实的药物研发子任务，完整地走通了从问题定义到硬件映射再到能耗估算的链条。很多硬件论文停在“这个任务理论上适合我们架构”；很多应用论文则停在“我们用了某个现成算法”。这篇工作的特别之处在于，它试图把两边接起来，让“适不适合”这件事有了具体的可计算、可比较的对象。

第二，它提醒我们，问题与硬件之间存在“自然契合度”。密码子优化之所以适合热力学采样，不是因为它特别“量子”或特别“高科技”，而是因为它本来就是一个离散能量最小化问题。**真正有前途的硬件，不一定是通用的，而是对某一类结构特别合拍的。**这和 GPU 最初在图形上占优、后来被机器学习接管的历史很像：硬件一旦和任务结构合拍，就会改变整个算法生态。

第三，它把“热噪声”从敌人变成了资源。传统计算机花大量功耗去维持确定性；而这类热力学的硬件的哲学，正好相反：既然热波动不可避免，那就把它纳入计算。这个转向有一点哲学味道，但并不空泛。它具体到一个很现实的问题：当你要的是概率分布，而不是一条完全确定的轨迹时，强行压制噪声有时就是浪费。

但别把演示当成终局

这篇论文也有它自己的边界，而且这些边界相当重要。

首先，作者选的 codon optimization 目标是一个简化版。生产级流程里还可能考虑 mRNA 二级结构、CpG 二核苷酸、UTR 影响以及更细的免疫原性或表达稳定性因素。论文自己也承认，这些项虽然可以继续加进局部或成对能量里，但问题规模和工程复杂度都会上升。也就是说，这个案例证明的是“可映射、可采样、可估能耗”，不是“药物设计的所有难题都被解决”。

其次，Potts 路径的硬件仍然是预测，不是已部署的成熟产品。论文把 Ising 路径放在更现实的位置，把 Potts 路径放在更长远的位置，这是诚实的，但也意味着我们不该把两者混为一谈。**实测结果、验证模型和外推估算，三者的证据强度完全不同。**

再者，codon optimization 也不是药物研发里最难的瓶颈。它重要，但它不是最难。论文自己也说得明白，这一案例的意义主要是证明一个方法论：问题可以如何被重新表述，硬件如何参与，估算如何建立。它更像一扇门，而不是终点。

最后，能耗比较永远和实现细节纠缠在一起。GPU 的“实际消耗”会受利用率、内核启动开销、批量大小和实现语言影响；热力学芯片的能耗也会受更新周期、噪声相关时间、通信和编程方式影响。作者已经尽量用上下界去夹住这个不确定性，但这并不意味着比较完全无争议。它只是比很多泛泛而谈的“省电”说法严谨得多。

更大的问题：我们到底该把计算放在哪里？

如果把这篇论文抽象一层，它讨论的就不只是密码子优化，而是一个更大的问题：**对于不同类型的任务，我们是否应该继续让同一种通用硬件承担一切？**

这正是“硬件彩票”这类讨论背后的核心焦虑。今天很多机器学习和优化流程跑在 GPU 上，并不意味着 GPU 天生是最优解，而可能只是历史路径最顺手的结果。若是更早期的硬件环境不同，算法演化出来的样子也许会不同。热力学计算的激进之处，不在于它宣称要替代所有芯片，而在于它逼我们重新想：**当问题本身是概率性的、组合性的、能量地形式的，为什么硬件就不能也长得更像这个问题？**

这也是为什么这篇论文会比一般的“又一个 benchmark”更有意思。它没有把结论写成某种霸道的统一理论，而是给了一个足够具体的例子：一个真实的生物任务，经过明确的数学映射，落到一种新硬件上，最后得到可比较的能耗估算。这个链条不短，但每一步都必要。少了其中任何一步，故事都很容易变成口号。

如果说传统计算的直觉是“把噪声压低，然后算出唯一答案”，那么这篇论文提供的直觉是另一种：**在某些问题上，答案不是被“算出来”的，而是被系统“冷却”出来的。**这不是修辞，而是工程哲学的变化。它暗示着未来的计算版图，也许不会围绕“更快的通用处理器”单线前进，而会出现越来越多针对特定问题结构而生的物理计算器件。

结尾：边界线本身，就是答案的一部分

在域壁编码里，状态不是一个孤立点，而是两个区域之间的边界。这个比喻其实很适合用来理解整篇论文：真正重要的，也不是某一个状态本身，而是**问题如何从一个领域的表述，跨过边界进入另一个领域的可执行形式**。密码子优化原本属于分子生物学和生物工程；经过能量函数化、Potts 建模、域壁编译和热力学采样，它变成了一个可以被物理硬件直接“沉下去”的问题。

这篇工作没有终结任何争论，也没有证明热力学硬件会立刻改写药物研发工业。但它做成了一件更扎实的事：它把“未来可能更省电”的口号，变成了一个可以核算、可以比较、也可以质疑的具体案例。**而在计算领域，能被认真质疑的演示，往往比一百句空泛的预测更有价值。**

如果下一代硬件真的会改变算法的形状，那么像密码子优化这样的问题，也许会成为最早的信号之一。它们不一定最耀眼，却足够清楚地告诉我们：当我们不再把热噪声当作敌人，计算的边界线本身，就会开始移动。

参考资料

1. [Energy-efficient codon optimization on thermodynamic hardware](#)
2. [Thermodynamic Computing](#)
3. [Codon Bias as a Means to Fine-Tune Gene Expression](#)
4. [Codon bias and heterologous protein expression](#)
5. [Domain wall encoding of discrete variables for quantum annealing and QAOA](#)
6. [Understanding domain-wall encoding theoretically and experimentally](#)
7. [mRNA codon optimization with quantum computers](#)
8. [CMOS-compatible Ising and Potts annealing using single-photon avalanche diodes](#)
9. [Electricity 2024 - Analysis - IEA](#)
10. [The hardware lottery](#)