
麦克斯韦妖佯谬分析

钱学森学院 少年班 2401
西安交通大学

or

摘要

麦克斯韦妖是热力学第二定律历史当中最为著名的思想实验之一：一个能够去观测分子速率并且开展选择性操控分子工作的“妖精”，似乎可以在不进行做功的情况下，促使孤立系统的熵产生自发的减少。鉴于数学推导以及三维分子动力学数值模拟这两个角度，对这一佯谬开展了系统的分析工作。在理论方面，依靠理想气体熵函数的严格推导，证明了妖的操作会导致气体熵发生严格的减小，同时结合了 Szilard 单分子引擎以及 Landauer 信息擦除原理，展示了信息熵对热力学熵所具备的补偿机制。在数值层面，构建了基于硬球碰撞模型来开展的 1000 粒子氩气分子动力学系统，直观地再现了由妖所诱导的温度分化、粒子数失衡以及非平衡稳态。仿真结果对 Sagawa-Ueda 涨落定理进行了定量的验证：妖所积累的 Shannon 信息熵始终远大于物理熵减的绝对值，它的信息利用效率约为 $\eta \approx 0.32$ 。研究所表明的情況是，在把妖的记忆存储以及擦除纳入到系统边界之后，系统的总熵依然契合 $\Delta S_{\text{total}} \geq 0$ ，热力学第二定律在信息物理框架之下依然得以成立。

1 引言

1867 年，詹姆斯·克拉克·麦克斯韦即 James Clerk Maxwell 提出了一个极其著名的思想实验，主要目的是去挑战热力学第二定律。热力学第二定律的克劳修斯表述以及数学表达指出，对于任何的孤立系统而言，它的总熵 S 永远都不会发生减少：

$$\Delta S_{\text{total}} \geq 0 \quad (1)$$

麦克斯韦在《热理论》最后一章当中，设想了这样的一种思想实验：

"But if we conceive a being whose faculties are so sharpened that he can follow every molecule in its course, such a being, whose attributes are still as essentially finite as our own, would be able to do what is at present impossible to us. For we have seen that the molecules in a vessel full of air at uniform temperature are moving with velocities by no means uniform, though the mean velocity of any great number of them, arbitrarily selected, is almost exactly uniform. Now let us suppose that such a vessel is divided into two portions, A and B, by a division in which there is a small hole, and that a being, who can see the individual molecules, opens and closes this hole, so as to allow only the swifter molecules to pass from A to B, and only the slower ones to pass from B to A. He will thus, without expenditure of work, raise the temperature of B and lower that of A, in contradiction to the second law of thermodynamics" [5].

赵凯华先生在《新概念物理教程 热学》里进行了如此的描述：

“麦克斯韦曾给热力学第二定律提出一个难题。他设想有一个能观察到所有分子轨迹和速度的‘小精灵’，把守着气体容器内隔板上一个小孔的闸门。当小精灵见到这边来了高速运动的分子，就打开闸门让它去到那边；见到那边来了低速运动的分子，就打开闸门让它来到这边。假设闸门完全没有摩擦，那么小精灵无需做功就可以使隔板两侧的气体一边越来越冷，另一边越来越热。这样一来，系统的熵降低了，热力学第二定律受到了挑战。人们将这个‘小精灵’称为麦克斯韦妖（Maxwell demon）。”¹⁰

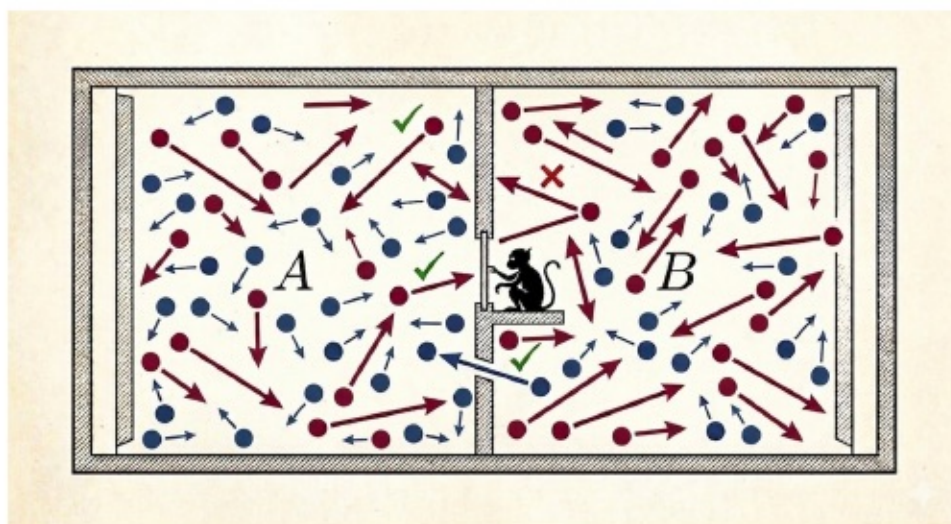


Figure 1: 麦克斯韦妖示意图

麦克斯韦妖借助在微观尺度上对分子开展选择性操作工作，似乎在不消耗宏观机械功的前提下，促使孤立系统的自发熵减得以实现。

2 数学表述

假设存在着一个绝热的刚性容器，被中间带有无质量小门的隔板划分成为体积相等的 A 以及 B 两个腔室，在初始状态之下，这两个腔室充满处于平衡态的同种理想气体，温度均处于 T 。

麦克斯韦妖负责去控制这个小门。它的操作逻辑如下所示：

1. 要是 A 室当中速率高于平均值的分子飞向小门，那么妖就会打开门来把它放入到 B 室当中；
2. 如果 B 室当中速率低于平均值的分子飞向小门，妖就会打开门把它放进 A 室里。

在经过足够长的时间之后，快分子会聚集于 B 室当中，慢分子则聚集在 A 室。这个时候 B 室的温度会升高至 T_H ，A 室的温度会降低到 T_C ，并且契合 $T_H > T_C$ 的条件。假设这个过程所交换的热量为 Q 也就是等效于从 A 室流向 B 室，并且小门不存在摩擦，妖也没有对系统开展宏观机械功的输出即 $W = 0$ 。

此时，气体系统的宏观热力学熵变 ΔS_{gas} 可以被近似地表示为：

$$\Delta S_{gas} = \oint \frac{dQ}{T} \approx -\frac{Q}{T_C} + \frac{Q}{T_H}$$

¹事实上这一部分推导并不细致，也并不严谨。完整的推导工作可见附录

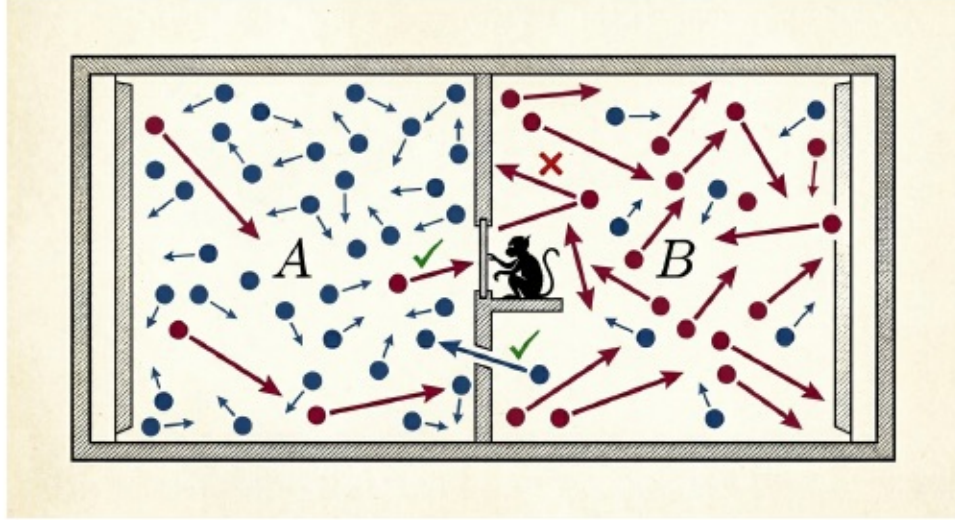


Figure 2: 一段时间后的系统情况示意图

鉴于 $T_H > T_C$ ，很显然会有：

$$\Delta S_{gas} < 0 \quad (2)$$

在假设妖并没有对系统产生其他影响的前提之下，孤立系统的总熵变 $\Delta S_{total} = \Delta S_{gas} < 0$ 。这个情况直接违背了公式 (1)，并且构成了麦克斯韦妖悖论的核心数学矛盾。关于麦克斯韦妖的物理学研究综述方面，可以去参见 Maruyama 等人 [4]。

3 信息熵和兰道尔原理

妖的分选操作工作依赖于信息的获取。在统计力学当中玻尔兹曼熵被定义为 $S = k_B \ln \Omega$ ，在信息论当中香农熵被定义为 $H = -\sum_i p_i \log_2 p_i$ [7]。如果系统微观状态处于等概率分布的时候，两者之间就存在着等价的关系：

$$S = (k_B \ln 2) \cdot H \quad (3)$$

齐拉特也就是 Szilard [8] 把妖的操作量化成为单分子引擎：拥有体积 V 以及温度 T ，插入隔板之后分为 $V/2$ 两个腔室。妖会去测量分子位置即获取 1 bit 信息，随后连接重物来让气体发生等温膨胀直到 V 。开展做功的情况为：

$$W = \int_{V/2}^V P dV = \int_{V/2}^V \frac{k_B T}{V} dV = k_B T \ln 2 \quad (4)$$

在内能不变的情况下，吸热 $Q = W$ ，热浴所产生的熵变就是：

$$\Delta S_{bath} = -\frac{Q}{T} = -k_B \ln 2 \quad (5)$$

到了这一步，系统就完成了做功工作，并且热力学的总熵依然产生了 $k_B \ln 2$ 的减少。

在 1961 年，罗尔夫·兰道尔即 Rolf Landauer [3] 指出：去获取信息以及存储信息在理论方面可以是可逆的也就是不会产生熵增，但是因为妖的存储器即大脑的容量是有限的，为了去处理下一个循环的工作，它就必须把上一个分子的信息擦除掉。这就是一个不可逆的物理过程。

依据兰道尔原理，如果擦除 1 bit 的信息，就会不可避免地向环境耗散热量 Q_{erase} ，同时会导致环境产生最小的熵增 ΔS_{erase} ，其数值为：

$$\Delta S_{\text{erase}} \geq k_B \ln 2 \quad (6)$$

如果把气体、妖以及环境当作封闭系统来使用，那么一个完整循环的总熵变就变成了：

$$\Delta S_{\text{total}} = \Delta S_{\text{bath}} + \Delta S_{\text{erase}} \geq -k_B \ln 2 + k_B \ln 2 = 0 \quad (7)$$

熵减会被信息擦除所带来的熵增所抵消，从而使得热力学第二定律依然保持成立的状态。

4 数值实验与验证

为了直观并且定量地去验证上述的信息热力学理论，本研究构建了基于三维硬球碰撞来开展的分子动力学数值实验。模型选定了氩气即 Ar 的真实物理参数为 $m = 6.63 \times 10^{-26} \text{ kg}$ ，并且把初始温度设定为 300 K。在系统内部包含了相互之间发生三维绝对弹性碰撞的粒子群体，从而用来真实地模拟气体内部试图去抹平非平衡态的热化过程。麦克斯韦妖把初始的均方根速率 v_{rms} 当作阈值来运用，对跨越了绝热隔板当中孔洞的分子去开展无机功的信息分选工作。

4.1 仿真方法与参数设置

模拟用的容器是边长 $L = 1 \mu\text{m}$ 的立方体，借助 $x = 0$ 平面的绝热隔板将其等分成为了 A 即左以及 B 即右这两个腔室，同时在隔板的中央开设有直径为 $d = 0.4L$ 的圆孔。系统当中含有 $N = 1000$ 个氩原子，其质量 $m = 6.63 \times 10^{-26} \text{ kg}$ 且硬球半径 $r = 0.015L$ 。初始温度 $T_0 = 300 \text{ K}$ ，初始的速度是按照麦克斯韦分布来进行随机抽样的工作。时间步长被设定为 $\Delta t = 0.015 \times (r/v_{\text{rms}})$ ，一共开展了 4×10^5 步的模拟。

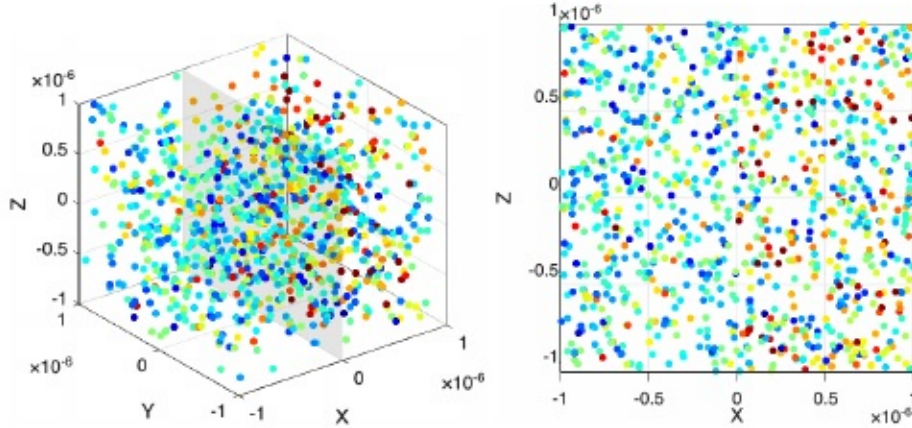


Figure 3: 系统终态模拟示意图

粒子以及容器的壁面之间会发生完全弹性的反射；粒子之间的碰撞检测工作每 2 步就会执行一次，要是两个粒子的球心距离小于 $2r$ 并且处于相互靠近的状态，就会沿着球心的连线去交换速度分量，从而模拟出三维弹性的硬球碰撞过程。麦克斯韦妖会把初始的均方根速率 v_{rms} 当作判别阈值：在每一个时间步当中，都会去检测跨越了 $x = 0$ 平面的分子，如果它位于开孔的区域也就是满足 $|y| < d/2$ 以及 $|z| < d/2$ ，就会把 A 室当中速率高于 v_{rms} 的分子放行进入 B 室，并且把 B 室当中速率低于 v_{rms} 的分子放行进入 A 室；如果不契合这个条件，跨越的分子就会被弹性反射回到原来的腔室当中。每当放行了一个分子，妖的存储器就会产生 1 bit 的累加。

每隔 100 步就会去记录系统的状态。温度是借助能量均分定理 $T = \sum m v_i^2 / (3Nk_B)$ 来进行计算，气体熵运用 Sackur-Tetrode 公式来近似得出 $S = Nk_B \ln(T^{3/2}V/N)$ ，妖所累积的信息熵则是依靠兰道尔对应关系 $S_{\text{info}} = N_{\text{bits}} \cdot k_B \ln 2$ 来进行换算的。

数值实验得出的结论能够较好地反映出麦克斯韦妖的运行逻辑以及相关结论。也就是说麦克斯韦妖的确是在把 A 室的动能转移到 B 室当中，从而促使两个腔室之中的温度向着两个不同的方向发生演化。

就像图 3 所示，在系统演化了相当长的一段时期以后，两个腔室的粒子速率就已经展现出了明显的差别。在系统的初始状态当中，两个腔室的粒子分布十分均匀，同时速率也都契合了麦克斯韦分布。而在系统的最终状态下，两个腔室的粒子速率分布情况可以参见图 4 所展示的内容。

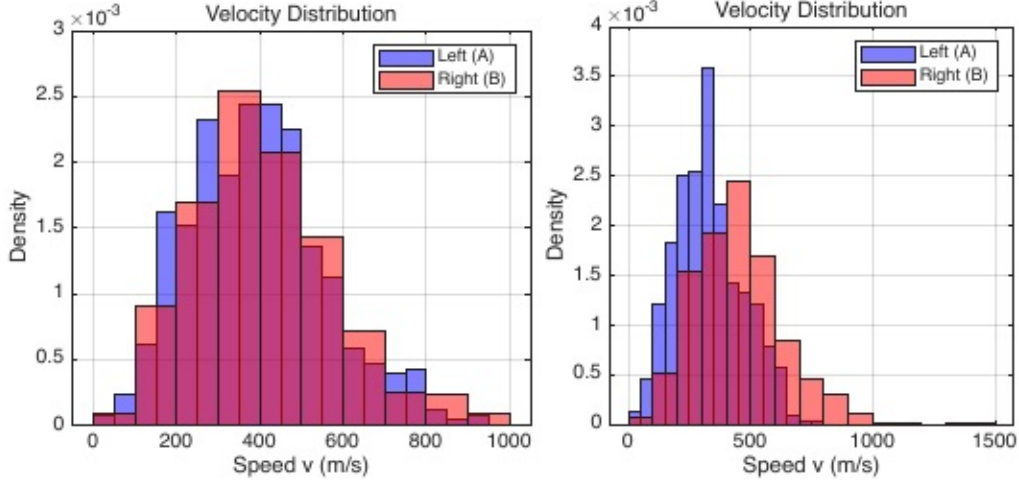


Figure 4: 系统初末态的速率分布对比图

在妖的持续干预之下，两个腔室的粒子速率分布情况发生了显著的左右频移以及撕裂现象：左室即 A 的蓝峰开始向低速区域集中，而右室即 B 的红峰则向高速区域进行了展宽。

这样的现象证实了麦克斯韦妖在微观相空间当中所具备的分类能力。在三维分子动力学模型里面，高频的粒子间弹性碰撞可谓是无时无刻不在促使着气体去向单一的平衡态发生弛豫。然而直方图的明确分裂情况表明，只要妖去开展信息测定以及门控工作的频率足够高，它就能够有效地去对抗系统内部存在的热力学耗散，从而强行地维持住一种高度有序的非平衡稳态分布特性。

图 5 直观地呈现出了麦克斯韦妖佯谬当中最为经典的宏观表象。在不引入任何外部机械功也就是 $W = 0$ 并且系统总动能严格守恒的前提下，右室的温度即 T_B 红线从初始的 300 K 持续地攀升到了接近 386 K 的程度；而左室的温度即 T_A 蓝线则对应地下降到了 205 K 左右。

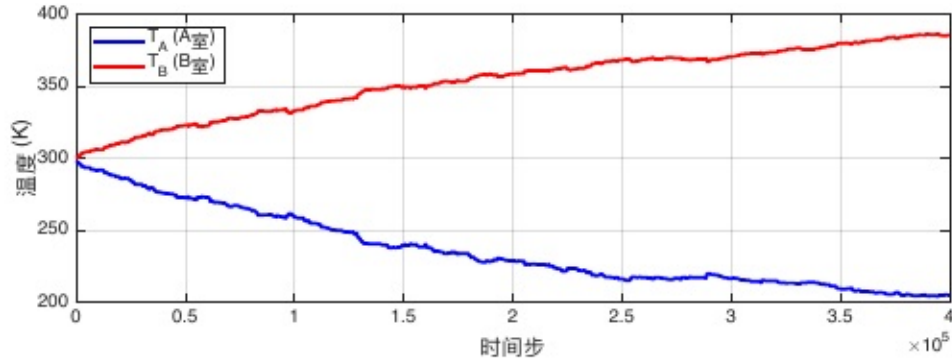


Figure 5: 系统数值实验的温度演化图

这种宏观温度持续分化的情况，违背了克劳修斯关于“热量不能自发从低温物体转移到高温物体”的经典表述。曲线的斜率在演化的后期阶段逐渐地变缓，这也就表明了妖所构建

出来的温差最终达到了一种动态平衡的状态。这证明了妖确实能够借助微观层面的信息获取以及路径选择工作，在宏观的尺度上驱动着能量实现定向的转移。

图 6 记录了两个腔室的粒子数 N_A 以及 N_B 随着时间推移的动态演化过程。尽管妖的门控逻辑看起来似乎是对称的即去放行 A 中的快分子以及 B 中的慢分子，但是仿真的结果显示，系统在演化的初期阶段就迅速地打破了粒子数的平衡状态，随后在存在着明显密度差的状态之中产生了动态的涨落。

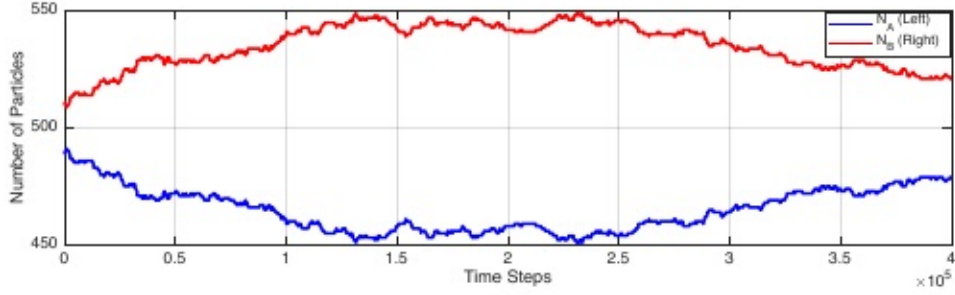


Figure 6: 系统数值实验的腔室粒子数演化图

这种自发产生的宏观密度失衡情况，主要原因囊括了麦克斯韦分布具备的统计非对称性以及温度与压强之间的强耦合效应。首先，在单一的阈值 v_{rms} 之下，分布曲线左右两侧的粒子积分通量并不处于严格相等的状态；其次，伴随着 A 室温度不断降低，绝大多数的分子都会变成“慢分子”，这就会使得 A 向 B 的跨越率发生骤降，同时 B 室的高温会促使大量的分子重新具备越过能量壁垒也就是门控阈值的能力，从而形成反向的回流。这也就表明，基于动能阈值来开展的信息分选工作，不仅会在系统当中诱发出逆向的热流，不可避免地还会引起粒子的质量流，进而自发地建立起宏观的密度梯度以及压强差。

图 7 属于本数值实验当中最为核心的结果。图里的绿线代表了妖在进行测定分子状态的时候所累积起来的香农信息熵即 S_{info} ；而黑线则代表了物理系统实际减少的热力学熵的绝对值也就是 $|\Delta S_{gas}|$ 。

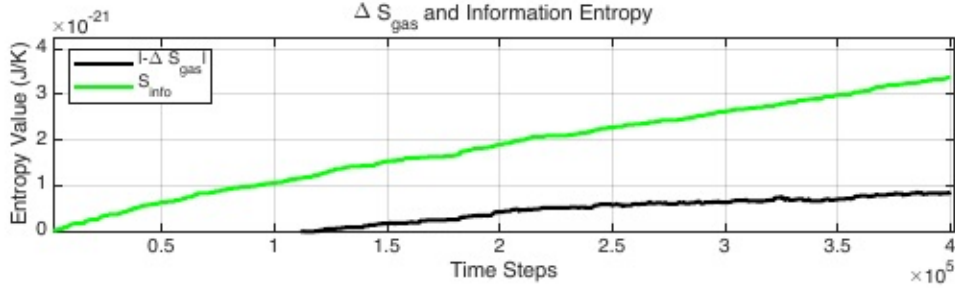


Figure 7: 系统数值实验的熵演化图

依据现代信息热力学的 Sagawa-Ueda 涨落定理^[6]，在存在信息反馈的非平衡过程当中，系统的自由能变化以及妖所获取的互信息量 I 之间满足严格的不等式 $\langle \Delta F \rangle \geq -k_B T \cdot I$ 。在本次仿真里面，每放行一个分子，妖就会获取 1 bit 的位置-速率信息即互信息，所对应的信息熵增量为 $k_B \ln 2$ 。这个不等式等价于 $S_{info} \geq |\Delta S_{gas}|$ ——即妖所累积的信息熵并不会低于它所诱导出来的物理熵减绝对值。图 7 的仿真数据十分清晰地印证了这个不等式：绿色曲线始终都位于黑色曲线的上方。

如果从定量的角度看，定义妖的信息利用效率为 $\eta = |\Delta S_{gas}|/S_{info}$ ，即单位信息消耗所换取的物理熵减。至仿真末态，系统累计放行 358 个分子（即 358 bit）， $\eta \approx 0.32$ ，信息冗余比 $S_{info}/|\Delta S_{gas}| \approx 3.2$ ——妖每消耗约 3 bit 的存储空间便可换取相当于 1 bit 热力学熵的物理减熵。这一效率并非如直觉预期的极低值，其原因在于：妖的固定阈值 v_{rms} 分选策略具有较高的选择性，只有约 10%-15% 的隔板穿越事件被放行，其余分子均被弹性反射，因此总信息累积量仅 358 bit 而非与总穿越次数成正比的庞大开销。

两条曲线之间的缺口表明，妖的信息积累始终高于其诱导的物理熵减。依据兰道尔原理 [3]，在妖最终擦除记忆位以释放存储空间时，每一个被擦除的比特至少向环境耗散 $k_B T \ln 2$ 的热量 [1]。仿真末态 $S_{\text{info}} \approx 3.4 \times 10^{-21} \text{ J/K}$ ，而 $|\Delta S_{\text{gas}}| \approx 1.1 \times 10^{-21} \text{ J/K}$ ，信息擦除产生的熵增约为物理熵减的 3 倍。仿真的结果以量化的方式确证了：即便妖在测量阶段以较高效率运行，信息擦除阶段依然会产生不可逆熵增，系统的总熵仍然增加，热力学第二定律在信息物理学的框架之下依然成立。

5 结论

依靠对麦克斯韦妖佯谬去开展详细的数学推导工作，能够得出结论：麦克斯韦妖并没有使得封闭系统的总熵发生减少。事实上对于麦克斯韦妖所观测的系统而言，它的一大部分熵会被麦克斯韦妖以观测的形式转化为信息熵，并且经过麦克斯韦妖被存储下来。如果把麦克斯韦妖以及被观测的系统当作一个封闭系统来使用，那么这个系统仍然契合熵不减的规律。或者认为麦克斯韦妖存在着有限的记忆容量，必须要去把旧的信息擦除掉才能够进行下一次的测量。在进行了记忆的擦除工作之后，这个被观测系统的熵同样不减 [1]。热力学第二定律依然保持成立的状态。在近些年里，信息热力学的理论预言已经在纳米尺度上获得了精密的实验验证。Toyabe 等人 [9] 借助高速反馈的电场来控制旋转的纳米珠，在实验当中促使信息到自由能的转换得以实现，从而定量地验证了广义的 Jarzynski 等式。Bérut 等人 [2] 运用双阱光势当中的胶体粒子，直接去测量了擦除 1 bit 信息所需要的最小耗散热即 $k_B T \ln 2$ ，这与兰道尔原理的预言实现了精确的吻合。这些实验充分地表明，麦克斯韦妖佯谬的解决并不仅仅是停留在纸面上的数学自洽，并且还得到了物理实验的直接支持。

致谢

本项研究工作的完成，不仅是对几组数学方程所开展的推演，更是一场向跨越了 150 年、并且连结着蒸汽时代以及信息时代的科学史诗的致敬。在此处，怀揣着深深的敬畏之情，向那些身处人类认知边界的物理学先驱们致以最崇高的敬意。

首先，需要向詹姆斯·克拉克·麦克斯韦即 James Clerk Maxwell 致以敬意。在 1867 年，正是他那无比深邃的想象力，释放出了这只游荡在热力学第二定律上空的“妖精”。在那个原子论尚未被大众所完全接受的年代当中，麦克斯韦凭借着惊人的洞察力，率先撕开了宏观绝对确定性的一角，从而迫使人类去直面微观世界的概率以及涨落。

同样地，也需要去铭记路德维希·玻尔兹曼即 Ludwig Boltzmann。他以孤掌难鸣的孤勇姿态，运用概率以及统计的手段为热力学铸造了不朽的灵魂即 $S = k_B \ln W$ 。尽管他最终以悲剧结束了生命，但是他为系统微观状态以及宏观熵之间建立的坚实桥梁，成为了后世去降伏这只妖精的通途大道。

伴随着历史的车轮驶入到 20 世纪当中，必须去特别感谢利奥·齐拉特即 Leo Szilard。他在 1929 年那篇具有划时代意义的论文里面，首次跨越了物质以及信息之间的鸿沟，并指出“信息同样是接受热力学约束的物理存在”。

在这场无比漫长的科学远征当中，同样也不能够忘记克劳德·香农即 Claude Shannon 以及约翰·冯·诺依曼即 John von Neumann。前者赋予了“信息”纯粹的数学骨架，而后者则极其敏锐地察觉到了信息熵以及热力学熵之间所具备的深刻同源性，从而让这两个看似完全不同的学科产生了命运的交汇。

最后，需要向罗尔夫·兰道尔也就是 Rolf Landauer 以及查尔斯·本内特即 Charles H. Bennett 致以最为深切的感谢。正是他们为麦克斯韦妖佯谬赋予了致命的最后一击。

从锅炉发出的轰鸣再到永不休止的演算，从蒸汽机里的卡诺循环直到量子比特的信息存储，这只“妖精”始终指引着人类跨越了物理、数学、计算机以及信息论的重重天堑。十分感谢这些伟大的灵魂，他们凭借凡人的身躯，丈量了整个世界的法则；他们对于真理的纯粹求索，将会永远地照亮物理学的星辰大海。

参考文献

- [1] Charles H. Bennett. The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.
- [2] Antoine Bérut, Artak Arakelyan, Artyom Petrosyan, Sergio Ciliberto, Raoul Dillenschneider, and Eric Lutz. Experimental verification of landauer's principle linking information and thermodynamics. *Nature*, 483(7388):187–189, 2012.
- [3] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.
- [4] Koji Maruyama, Franco Nori, and Vlatko Vedral. Colloquium: The physics of maxwell's demon and information. *Reviews of Modern Physics*, 81(1):1–23, 2009.
- [5] J. Clerk Maxwell. *Theory of Heat*. Longmans, Green, and Co., London, 3 edition, 1872.
- [6] Takahiro Sagawa and Masahito Ueda. Fluctuation theorem with information exchange: Role of correlations in stochastic thermodynamics. *Physical Review Letters*, 109(18):180602, 2012.
- [7] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [8] Leo Szilard. Über die entropieverminderung in einem thermodynamischen system bei eingriffen intelligenter wesen. *Zeitschrift für Physik*, 53(11–12):840–856, 1929.
- [9] Shoichi Toyabe, Takahiro Sagawa, Masahito Ueda, Eiro Muneyuki, and Masaki Sano. Experimental demonstration of information-to-energy conversion and validation of the generalized jarzynski equality. *Nature Physics*, 6(12):988–992, 2010.
- [10] 赵凯华 and 罗蔚茵. 新概念物理教程·热学. 高等教育出版社, 北京, 2 edition, 2005. ISBN 7-04-017680-7.

附录

A 附录：麦克斯韦妖操作下气体系统熵减的推导

在正文的第二节当中，推导过程运用了克劳修斯等式的近似表达。为了从根本上去展现出系统在麦克斯韦妖的作用之下所发生的宏观热力学异变，本附录借助理想气体熵的具体数学解析式，给出了系统熵变严格小于 0 的证明。

假设容器的总容积为 $2V_0$ ，并且 A 室以及 B 室的体积保持恒等即 $V_A = V_B = V_0$ 。在初始的状态之下，A 以及 B 两个腔室分别含有 n 摩尔的单原子理想气体，同时系统的初始温度处于 T_0 。依据定容热容的定义，初始系统的总内能就是：

$$U_{\text{initial}} = U_{A0} + U_{B0} = nC_V T_0 + nC_V T_0 = 2nC_V T_0 \quad (8)$$

在经过麦克斯韦妖的分选工作之后，分子会发生等量的交换，粒子数依旧保持 n 不变，并且系统处于绝热状态、总体积不发生改变、未曾对外开展做功。如果设稳态之后 A 室的温度下降到了 T_C ，B 室的温度上升到了 T_H 即 $T_H > T_0 > T_C$ 。依靠热力学第一定律也就是系统的能量守恒法则，最终态的总内能必定会等于初始的内能：

$$nC_V T_C + nC_V T_H = 2nC_V T_0 \implies T_C + T_H = 2T_0 \quad (9)$$

对于单原子的理想气体而言，它的摩尔熵函数可以被写成温度以及体积的函数：

$$S(T, V) = nC_V \ln T + nR \ln V + S_0 \quad (10)$$

在整个过程当中，鉴于 V_A 以及 V_B 均保持着恒定即 $= V_0$ ，体积项在去计算差值的时候将会相互发生抵消。系统的初始总熵即为：

$$S_{initial} = 2(nC_V \ln T_0 + nR \ln V_0 + S_0) \quad (11)$$

系统的末态总熵就是：

$$S_{final} = (nC_V \ln T_C + nR \ln V_0 + S_0) + (nC_V \ln T_H + nR \ln V_0 + S_0) \quad (12)$$

因为上述的缘故，理想气体系统的总宏观熵变就变成了：

$$\Delta S_{gas} = S_{final} - S_{initial} = nC_V \ln T_C + nC_V \ln T_H - 2nC_V \ln T_0 \quad (13)$$

在提取了公因式并且合并了对数项之后，就得到了严密的熵变表达式：

$$\Delta S_{gas} = nC_V \ln \left(\frac{T_C \cdot T_H}{T_0^2} \right) \quad (14)$$

依据数学当中的均值不等式即 AM-GM Inequality，对于两个并不相等的正数 T_C 以及 T_H 会有：

$$\sqrt{T_C \cdot T_H} < \frac{T_C + T_H}{2} \quad (15)$$

如果把能量守恒等式 (9) 代入到不等式的右边，就可以得到：

$$\sqrt{T_C \cdot T_H} < \frac{2T_0}{2} = T_0 \implies T_C \cdot T_H < T_0^2 \quad (16)$$

把这个不等式的关系代入到方程 (14) 之中，鉴于真数部分满足 $\frac{T_C \cdot T_H}{T_0^2} < 1$ ，很显然它的对数值必定会小于 0：

$$\Delta S_{gas} = nC_V \ln \left(\frac{T_C \cdot T_H}{T_0^2} \right) < 0 \quad (17)$$

到了这一步，依靠理想气体的严密数学方程彻底地证明了，如果在绝热的边界内部麦克斯韦妖不去付出额外的信息物理代价的话，它的行为确实导致了孤立气体的熵发生了严格的减小。这就正是需要去引入兰道尔原理的缘故。

B 附录：数值实验 Matlab 代码

```
1 clear; clc; close all;
2 kB = 1.380649e-23;
3 m_Ar = 6.6335209e-26;
4 T0 = 300;
5 N = 1000;
6 L = 1e-6;
7 Lx = L; Ly = L; Lz = L;
8 hole_d = L * 0.4;
9 R_particle = L * 0.015;
10 v_rms_initial = sqrt(3 * kB * T0 / m_Ar);
11 dt = (R_particle / v_rms_initial) * 0.015;
12 steps = 400000;
13 plot_interval = 100;
14
15 x = (rand(N, 1) * 2 - 1) * Lx;
16 y = (rand(N, 1) * 2 - 1) * Ly;
17 z = (rand(N, 1) * 2 - 1) * Lz;
```

```

18
19 sigma_v = sqrt(kB * T0 / m_Ar);
20 vx = sigma_v * randn(N, 1);
21 vy = sigma_v * randn(N, 1);
22 vz = sigma_v * randn(N, 1);
23 v_rms_threshold = v_rms_initial;
24 log_len = floor(steps/plot_interval);
25 time_arr = zeros(1, log_len);
26 TA_arr = zeros(1, log_len); TB_arr = zeros(1, log_len);
27 NA_arr = zeros(1, log_len); NB_arr = zeros(1, log_len);
28 E_total_arr = zeros(1, log_len);
29 dS_gas_arr = zeros(1, log_len); S_info_arr = zeros(1, log_len);
30 eta_arr = zeros(1, log_len);
31 total_bits = 0; log_idx = 1;
32 Vol_half = Lx * 2 * Ly * 2 * Lz;
33 S0_A = (N/2)*kB*log(T0^1.5 * Vol_half/(N/2));
34 S_total_0 = S0_A * 2;
35 initial_E_tot = sum(0.5 * m_Ar * (vx.^2 + vy.^2 + vz.^2));
36
37 fig = figure('Name', 'Maxwell Demon Simulation', 'Position', [50, 50,
    1600, 900]);
38 colormap('jet');
39 ax1 = subplot(2, 3, 1); hold on; box on; view(3); grid on;
40 xlim([-Lx, Lx]); ylim([-Ly, Ly]); zlim([-Lz, Lz]);
41 patch([0 0 0 0], [-Ly Ly Ly -Ly], [-Lz -Lz Lz Lz], 'k', 'FaceAlpha', 0.1,
    'EdgeColor', 'none');
42 scatter_particles = scatter3(NaN, NaN, NaN, 20, NaN, 'filled');
43 caxis([0, v_rms_initial*2]);
44 title(''); xlabel('X'); ylabel('Y'); zlabel('Z');
45 ax2 = subplot(2, 3, 2); hold on; box on; grid on;
46 title('Velocity Distribution'); xlabel('Speed v (m/s)'); ylabel('Density'
    );
47 h_hist_A = histogram(NaN, 'Normalization', 'pdf', 'FaceColor', 'b', '
    FaceAlpha', 0.5);
48 h_hist_B = histogram(NaN, 'Normalization', 'pdf', 'FaceColor', 'r', '
    FaceAlpha', 0.5);
49 legend('Left (A)', 'Right (B)');
50 ax3 = subplot(2, 3, 3); hold on; box on; grid on;
51 h_NA = plot(NaN, NaN, 'b-', 'LineWidth', 2);
52 h_NB = plot(NaN, NaN, 'r-', 'LineWidth', 2);
53 title('Particle Density Evolution'); xlabel('Time Steps'); ylabel('Number
    of Particles');
54 legend('N_A (Left)', 'N_B (Right)', 'Location', 'northeast');
55 ax4 = subplot(2, 3, 4); hold on; box on; grid on;
56 h_TA = plot(NaN, NaN, 'b-', 'LineWidth', 2);
57 h_TB = plot(NaN, NaN, 'r-', 'LineWidth', 2);
58 title('Temperature Separation'); xlabel('Time Steps'); ylabel('
    Temperature (K)');
59 legend('Left (A)', 'Right (B)', 'Location', 'northwest');
60 ax5 = subplot(2, 3, 5); hold on; box on; grid on;
61 h_dS = plot(NaN, NaN, 'k-', 'LineWidth', 2);
62 h_Sinfo = plot(NaN, NaN, 'g-', 'LineWidth', 2);
63 title('\Delta S_{gas} and Information Entropy');
64 xlabel('Time Steps'); ylabel('Entropy Value (J/K)');
65 legend('|\Delta S_{gas}|', 'S_{info}', 'Location', 'northwest');
66 ax6 = subplot(2, 3, 6); hold on; box on; grid on;
67 h_Etot = plot(NaN, NaN, 'm-', 'LineWidth', 2);
68 title('Total Kinetic Energy');
69 xlabel('Time Steps'); ylabel('Total Energy (J)');

```

```

70 ylim([initial_E_tot*0.99, initial_E_tot*1.01]);
71
72 for step = 1:steps
73     x_old = x;
74     x = x + vx * dt; y = y + vy * dt; z = z + vz * dt;
75     hit = x < -Lx; x(hit) = -Lx + abs(x(hit)+Lx); vx(hit) = -vx(hit);
76     hit = x > Lx; x(hit) = Lx - abs(x(hit)-Lx); vx(hit) = -vx(hit);
77     hit = y < -Ly; y(hit) = -Ly + abs(y(hit)+Ly); vy(hit) = -vy(hit);
78     hit = y > Ly; y(hit) = Ly - abs(y(hit)-Ly); vy(hit) = -vy(hit);
79     hit = z < -Lz; z(hit) = -Lz + abs(z(hit)+Lz); vz(hit) = -vz(hit);
80     hit = z > Lz; z(hit) = Lz - abs(z(hit)-Lz); vz(hit) = -vz(hit);
81     v_mag = sqrt(vx.^2 + vy.^2 + vz.^2);
82     at_hole = (abs(y) < hole_d/2) & (abs(z) < hole_d/2);
83     cross_L2R = (x_old <= 0) & (x > 0);
84     cross_R2L = (x_old >= 0) & (x < 0);
85     pass_L2R = cross_L2R & at_hole & (v_mag > v_rms_threshold);
86     pass_R2L = cross_R2L & at_hole & (v_mag < v_rms_threshold);
87     block_L2R = cross_L2R & ~pass_L2R;
88     block_R2L = cross_R2L & ~pass_R2L;
89     reflect = block_L2R | block_R2L;
90     vx(reflect) = -vx(reflect);
91     x(reflect) = x_old(reflect);
92     total_bits = total_bits + sum(pass_L2R) + sum(pass_R2L);
93     if mod(step, 2) == 0
94         pos = [x, y, z]; vel = [vx, vy, vz];
95         dist_mat = pdist2(pos, pos, 'squaredeuclidean');
96         [pair_i, pair_j] = find(triu(dist_mat < (2*R_particle)^2, 1));
97         for k = 1:length(pair_i)
98             pi = pair_i(k); pj = pair_j(k);
99             dr = pos(pi,:) - pos(pj,:);
100             dv = vel(pi,:) - vel(pj,:);
101             if dot(dv, dr) < 0
102                 factor = dot(dv, dr) / norm(dr)^2;
103                 vel(pi,:) = vel(pi,:) - factor * dr;
104                 vel(pj,:) = vel(pj,:) + factor * dr;
105             end
106         end
107         vx = vel(:,1); vy = vel(:,2); vz = vel(:,3);
108     end
109     if mod(step, plot_interval) == 0
110         idx_A = x <= 0; idx_B = x > 0;
111         N_A = sum(idx_A); N_B = sum(idx_B);
112         v_mag = sqrt(vx.^2 + vy.^2 + vz.^2);
113         T_A = sum(m_Ar * v_mag(idx_A).^2) / (3 * max(N_A, 1) * kB);
114         T_B = sum(m_Ar * v_mag(idx_B).^2) / (3 * max(N_B, 1) * kB);
115         E_tot = sum(0.5 * m_Ar * v_mag.^2);
116         S_A = N_A*kB*log(max(T_A,eps)^1.5 * Vol_half/max(N_A,eps));
117         S_B = N_B*kB*log(max(T_B,eps)^1.5 * Vol_half/max(N_B,eps));
118         dS_gas = (S_A + S_B) - S_total_0;
119         S_info = total_bits * kB * log(2);
120         time_arr(log_idx) = step;
121         TA_arr(log_idx) = T_A; TB_arr(log_idx) = T_B;
122         NA_arr(log_idx) = N_A; NB_arr(log_idx) = N_B;
123         E_total_arr(log_idx) = E_tot;
124         dS_gas_arr(log_idx) = dS_gas; S_info_arr(log_idx) = S_info;
125         eta_arr(log_idx) = -dS_gas / max(S_info, 1e-300);
126         set(scatter_particles, 'XData', x, 'YData', y, 'ZData', z, 'CData',
127             v_mag);
128         h_hist_A.Data = v_mag(idx_A);

```

```

128     h_hist_B.Data = v_mag(idx_B);
129     set(h_NA, 'XData', time_arr(1:log_idx), 'YData', NA_arr(1:log_idx
130         ));
130     set(h_NB, 'XData', time_arr(1:log_idx), 'YData', NB_arr(1:log_idx
131         ));
131     set(h_TA, 'XData', time_arr(1:log_idx), 'YData', TA_arr(1:log_idx
132         ));
132     set(h_TB, 'XData', time_arr(1:log_idx), 'YData', TB_arr(1:log_idx
133         ));
133     set(h_dS, 'XData', time_arr(1:log_idx), 'YData', -dS_gas_arr(1:
134         log_idx));
134     set(h_Sinfo, 'XData', time_arr(1:log_idx), 'YData', S_info_arr(1:
135         log_idx));
135     set(h_Etot, 'XData', time_arr(1:log_idx), 'YData', E_total_arr(1:
136         log_idx));
136     drawnow;
137     log_idx = log_idx + 1;
138     end
139 end
140 fprintf('仿真结束\n');
141 fprintf('最终温度: T_A = %.1f K, T_B = %.1f K\n', TA_arr(end), TB_arr(end
142     ));
142 fprintf('最终粒子数: N_A = %d, N_B = %d\n', NA_arr(end), NB_arr(end));
143 fprintf('总信息比特数: %d bits\n', total_bits);
144 fprintf('信息熵 S_info = %.3e J/K\n', S_info_arr(end));
145 fprintf('物理熵变 |ΔS_gas| = %.3e J/K\n', -dS_gas_arr(end));
146 fprintf('信息效率 = |-ΔS_gas| / S_info = %.3f\n', -dS_gas_arr(log_idx
147     -1) / max(S_info_arr(log_idx-1), 1e-300));
147 fprintf('信息冗余比 S_info / |-ΔS_gas| = %.1f\n', S_info_arr(log_idx-1) /
    max(-dS_gas_arr(log_idx-1), 1e-300));

```