

机器学习、代理模型与 天体物理应用

Lile Wang 王力乐
KIAA/PKU



Machine Learning in Astrophysics

有了 AI/机器学习方法，能做什么？

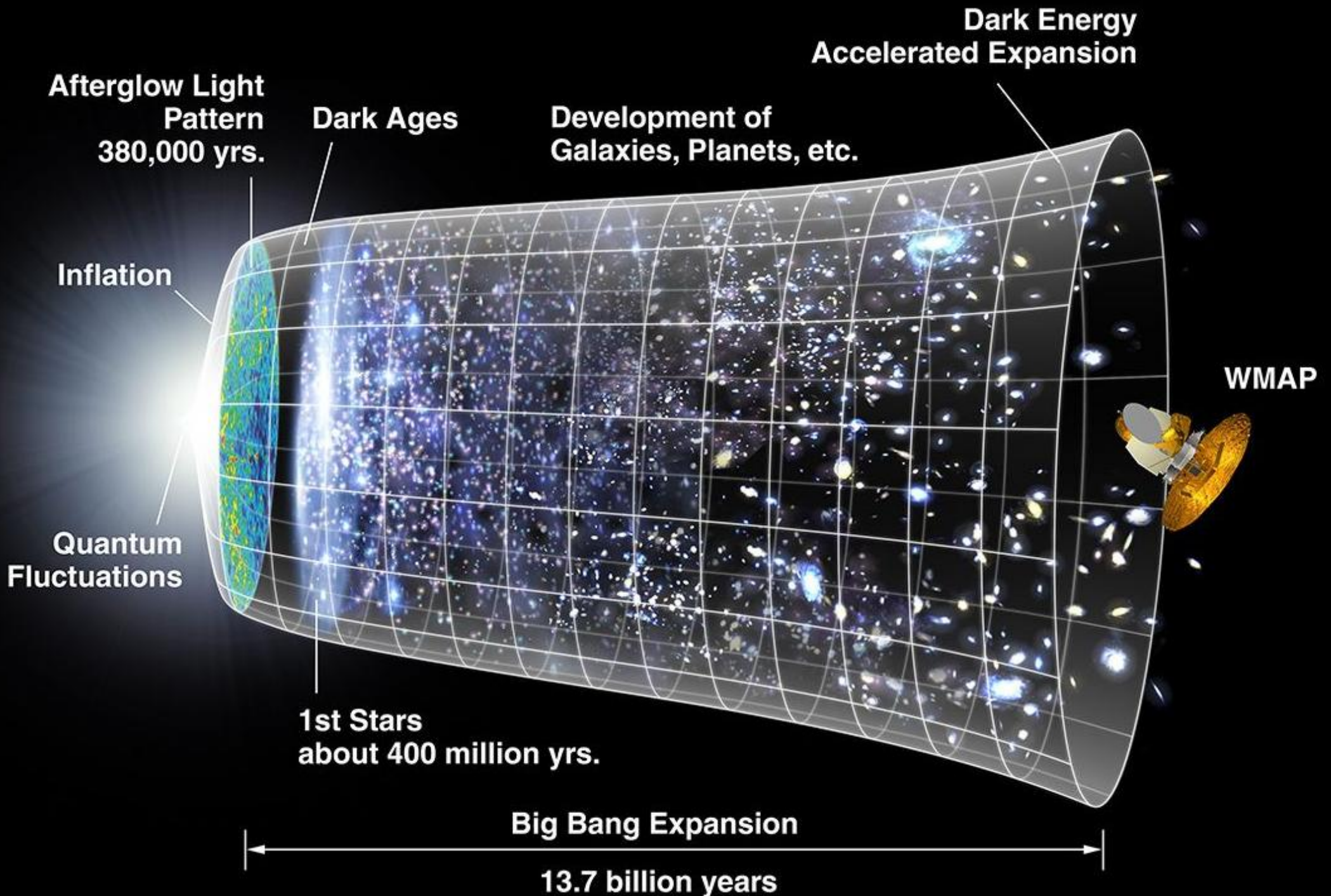
别被人讥讽——

拿着一把锤子，看谁都是钉子

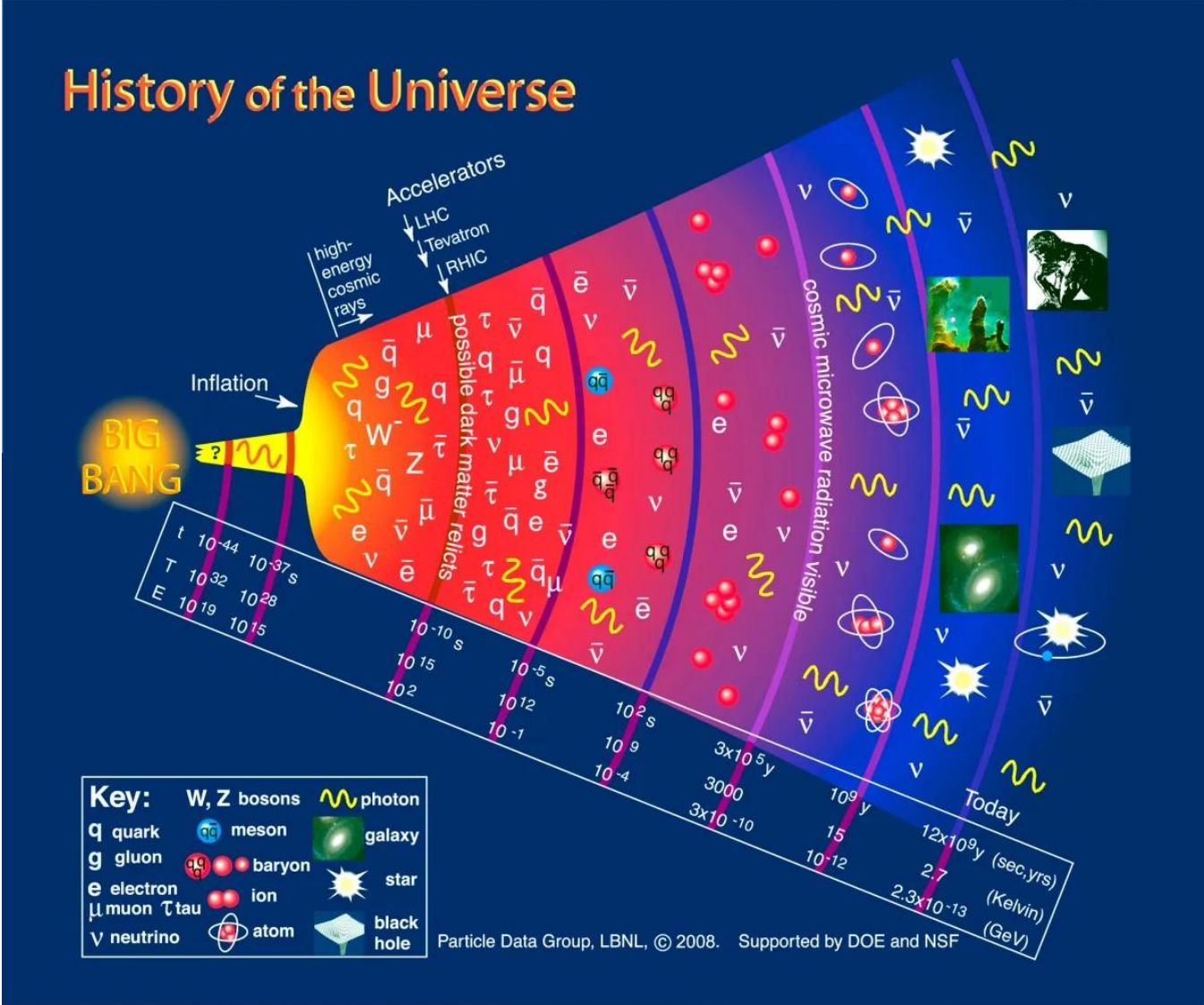
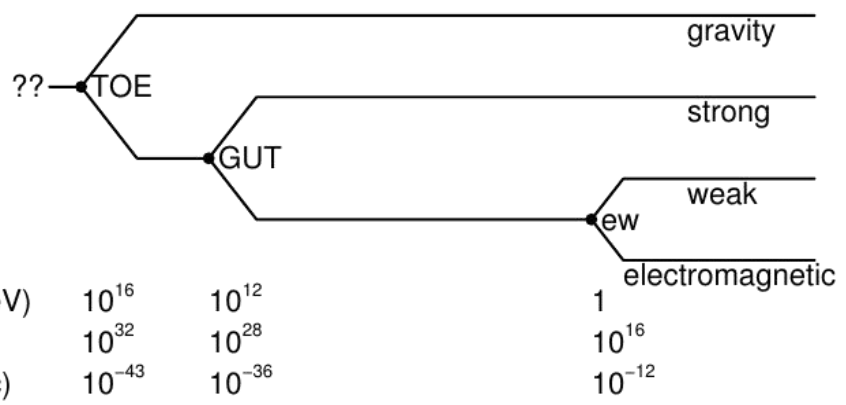
科研方法的使用理应是自限、自洽的

核心问题是：我们要敲哪个钉子？



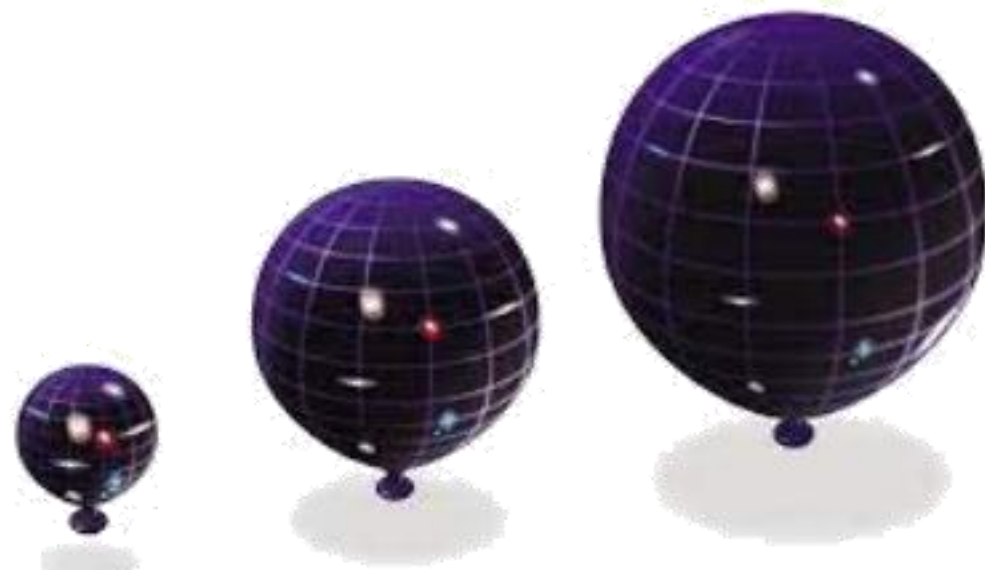


- “大爆炸”一词最早是小黑子，现在只是习惯性称谓，切忌望文生义
- 极早期的宇宙温度极高，实现了物理学家们渴求的条件
- 随着温度（能量标度）的一步步降低，相互作用一步步从统一模型中剥离
- 原初核合成：最早被研究的大爆炸遗迹
- 微波背景辐射（CMB）：1978、2006年诺贝尔奖

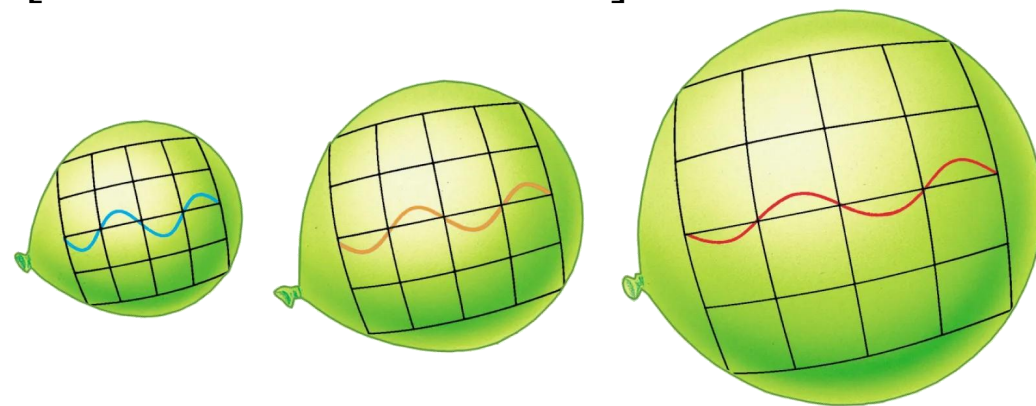
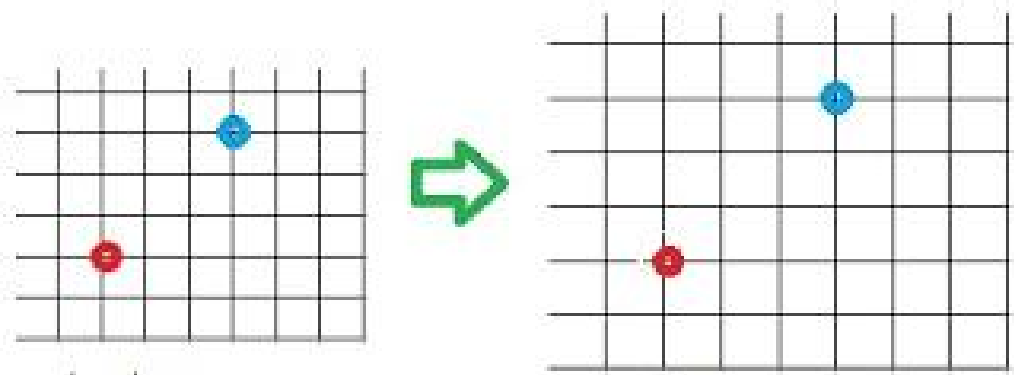


背景知识 0：哈勃定律与宇宙膨胀

- 运动的来源是时空背景，而非星系本身
 - 为什么要说“星系”？
- 理解历史，定义超光速
 - 表观超光速比比皆是，但这是另一种
 - 广义相对论只禁止“局部”的超光速（物体不能走出类空间隔）
 - 两个蜗牛爬得慢，但是气球吹得快
- 宇宙学红移不是多普勒效应！
- 共动坐标系的引入： x/a

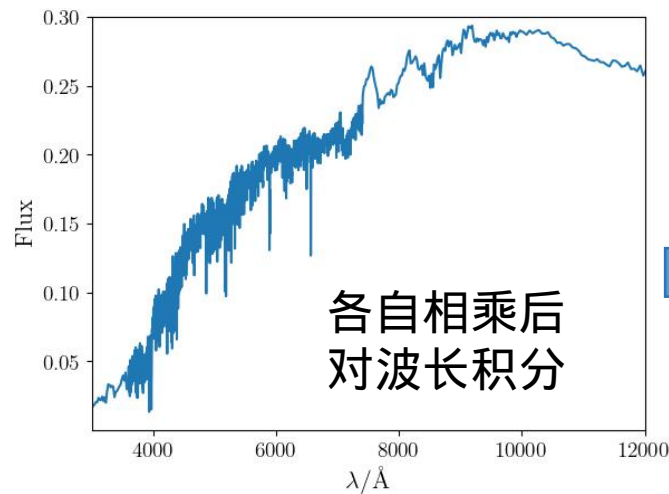


$$-d\tau^2 = -dt^2 + a^2 \left[\frac{dr^2}{1 - kr^2} + r^2(d\theta^2 + \sin^2\theta d\varphi^2) \right] \rightarrow \frac{dt}{a} = \frac{dr}{1 - kr^2}$$

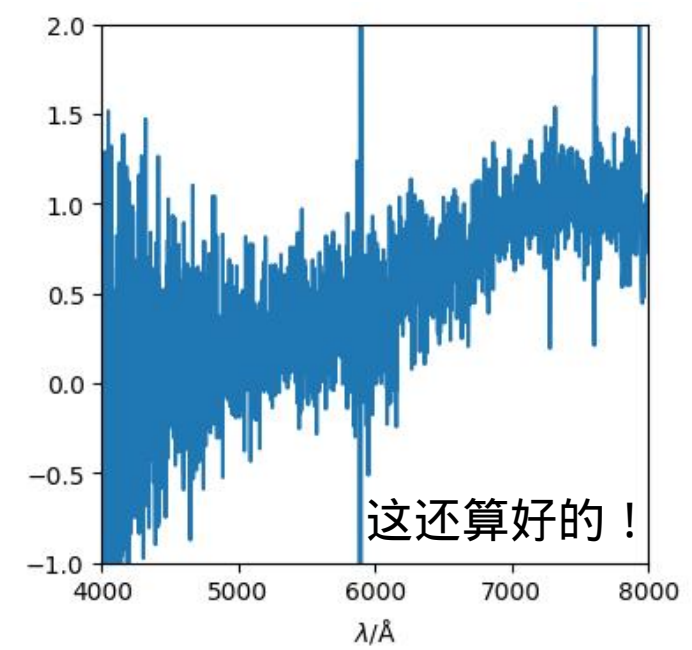
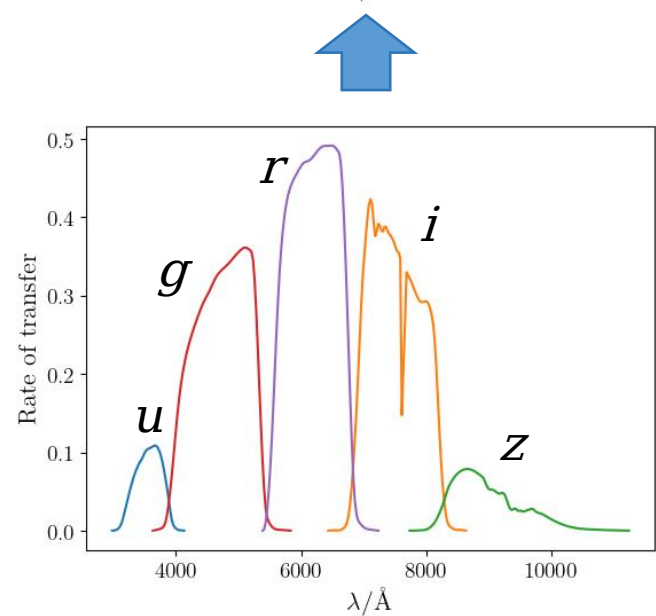


背景知识 0：哈勃定律与宇宙膨胀

- 光谱各处的信噪比正比于该波长范围内、曝光时间段内接收到的光子数平方根 \sqrt{N} (泊松过程)
- 对同一天体使用同一望远镜进行 t 时长曝光，量子效率不变：
- 设不进行分光时，信噪比为 \sqrt{N}
- 使用 $R = 3000$ 的光谱仪 (假设各个波长的谱分布函数基本同数量级)，信噪比变为多少？
- 若要让 $R = 3000$ 的光谱测量具有相同的信噪比，需要多少曝光时间？
- 对大多数被观测的星系，天文学家们被迫采用“超廉价”版光谱
- 测光：用滤色片就行，但请做好标定

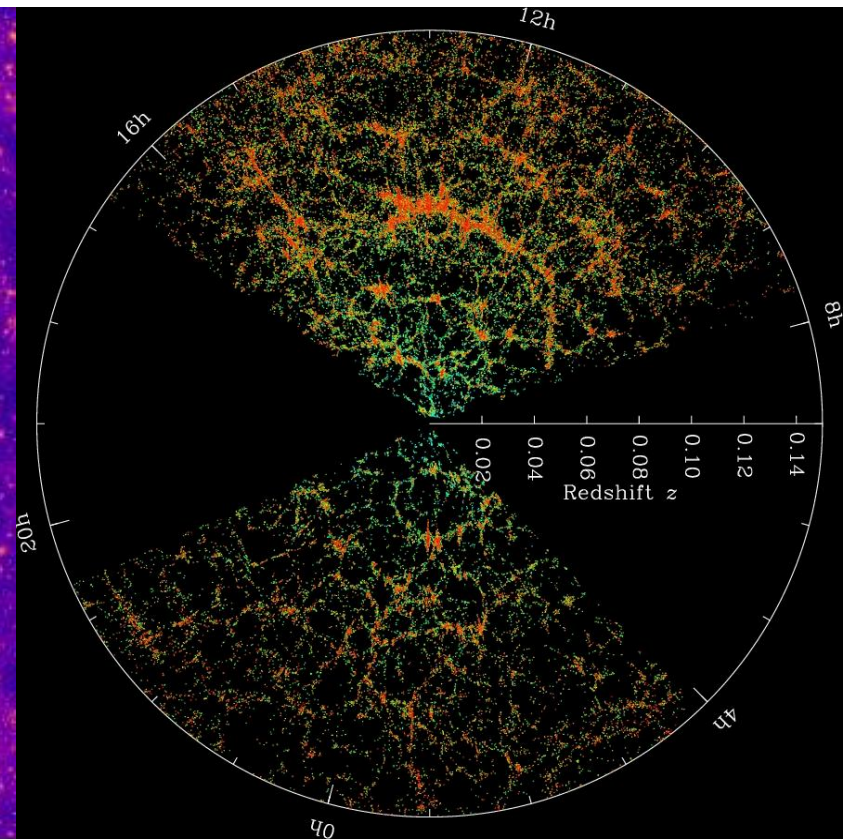
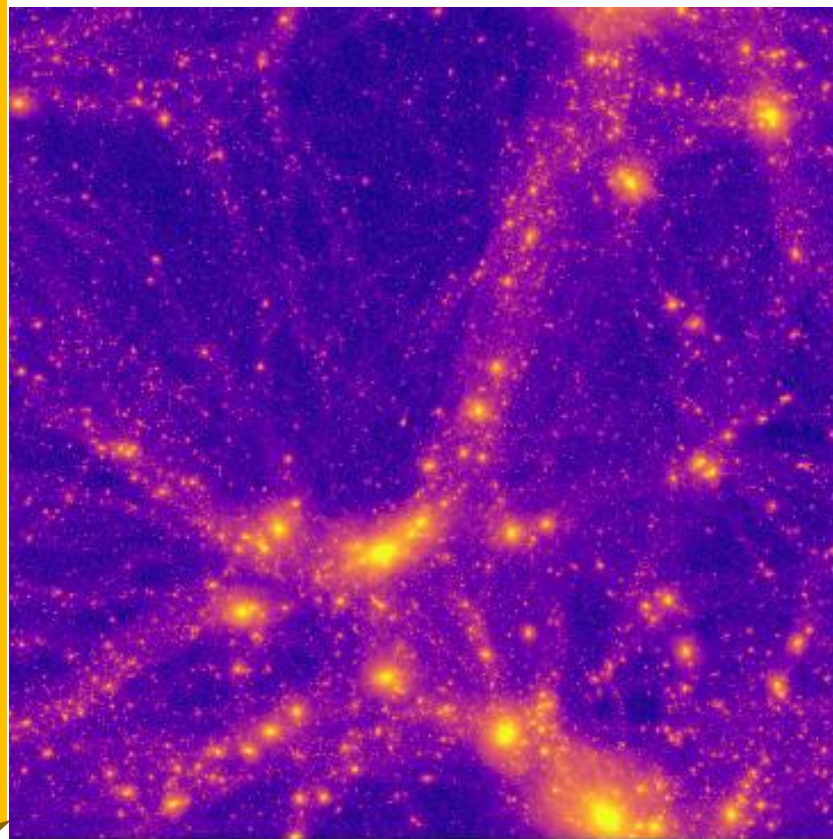


对 r 波段归一化：
 $u = 0.036,$
 $g = 0.657,$
 $r = 1,$
 $i = 0.784,$
 $z = 0.154.$



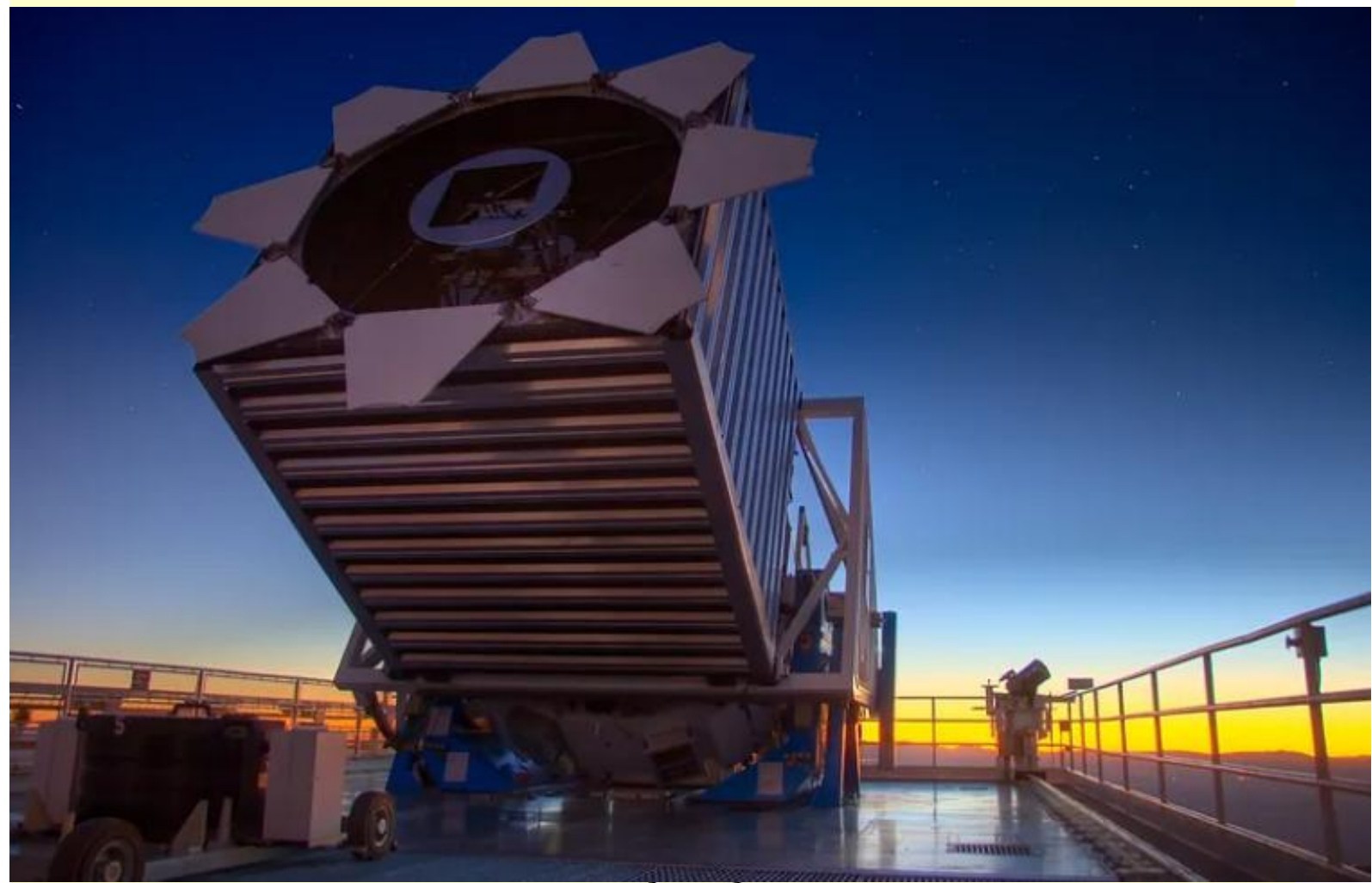
“昂贵”的光谱与“廉价”的测光

- 红移意味着“星系发光时的宇宙尺度”
- 星系的红移数据可以用来：
 - 研究宇宙大尺度结构（结合宇宙学模型）
 - 研究宇宙不同阶段星系的特征（星系形成及演化）
- 红移数据原则上需要光谱数据……
- 有无可能通过测光数据给出大致的红移？
- 好消息：某些光谱具有特征的星系（如所谓 LRG）原则上测光红移能给得相对准确
- 坏消息 0：具有特定特征的星系并不多，且越早期数量越少
- 坏消息 1：识别这些特征并不简单
- 看上去就很适合机器学习……
- 事实上也是机器学习在天文中最早有效应用（SDSS）



“昂贵”的光谱与“廉价”的测光

- 红移意味着“星系发光时的宇宙尺度”
- 星系的红移数据可以用来：
 - 研究宇宙大尺度结构（结合宇宙学模型）
 - 研究宇宙不同阶段星系的特征（星系形成及演化）
- 红移数据原则上需要光谱数据……
- 有无可能通过测光数据给出大致的红移？
- 好消息：某些光谱具有特征的星系（如所谓 LRG）原则上测光红移能给得相对准确
- 坏消息 0：具有特定特征的星系并不多，且越早期数量越少
- 坏消息 1：识别这些特征并不简单
- 看上去就很适合机器学习……
- 事实上也是机器学习在天文中最早有效应用（SDSS）



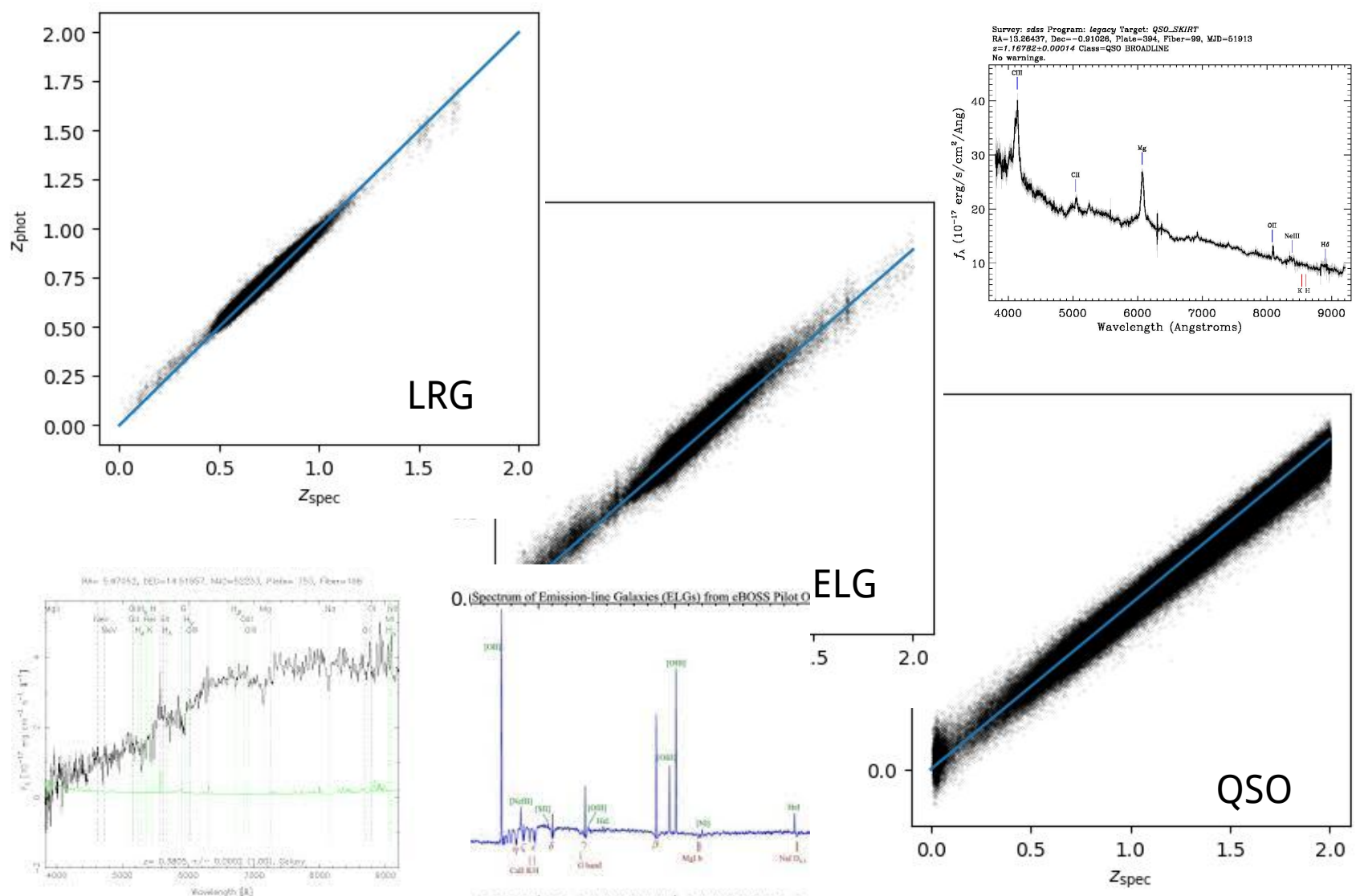
4000 Å

wavelength

9000 Å

“昂贵”的光谱与“廉价”的测光

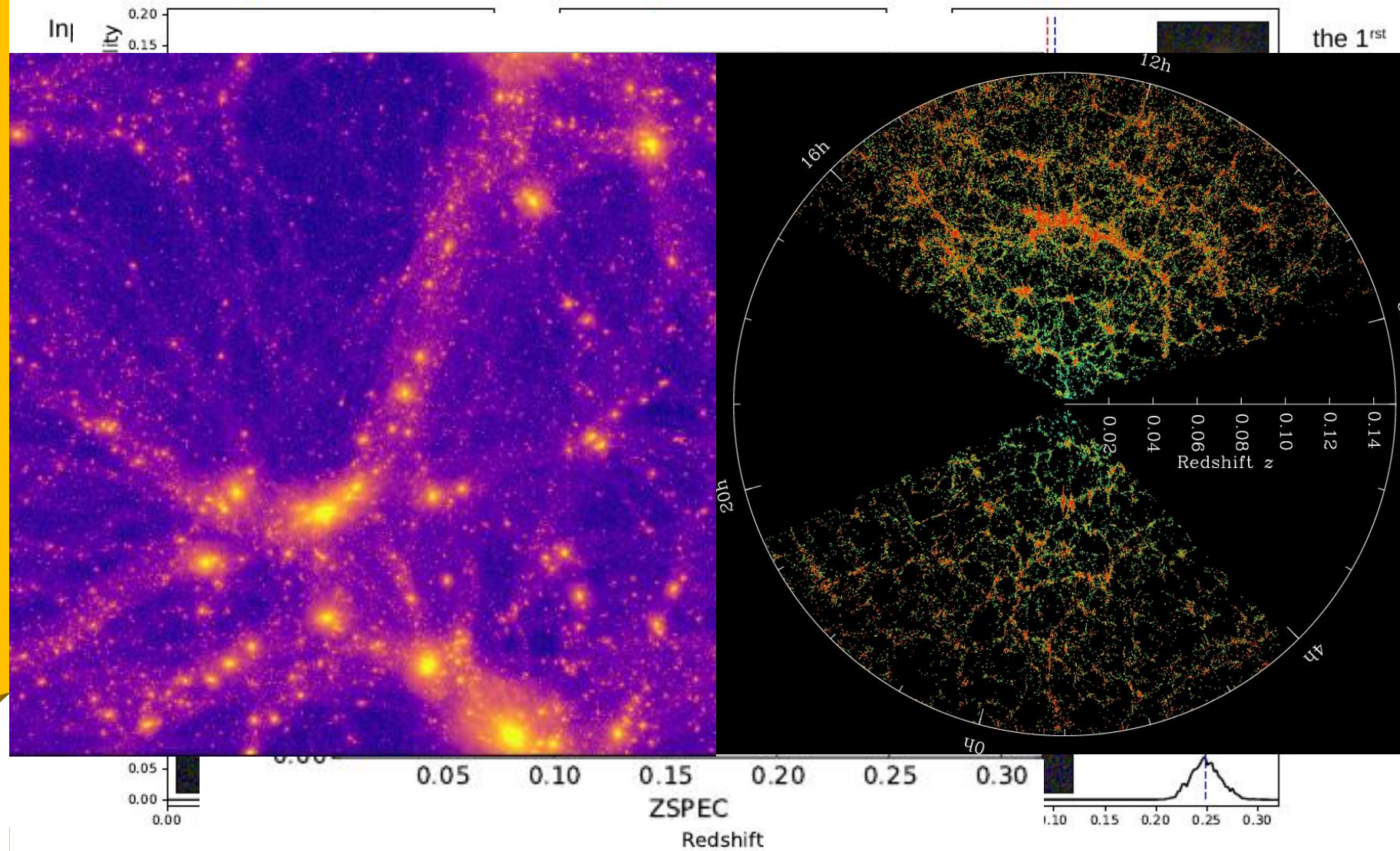
- 训练数据要求不算高
 - 有实锤的红移，也有测光数据
 - 思考：有必要分别下载吗？
 - 应当如何最大化利用现有数据，生成训练集？
- SDSS “钦定” 的 “古典” 方法：随机决策森林
- 就这么埋头炼丹？要不要 PI 一下？
- 某些星系的光谱特性有利于结果，其他类型则不尽然……
- 新的好消息：星系类型会反映在形态上，测光和成像数据往往能同时具备！
- 我们不必事先建立完备的星系演化理论（否则还做个啥星系研究）可以 “推卸责任” ……



测光红移：从简单做起……

我自个儿瞎做的
(自从我还是
博士生助教时)

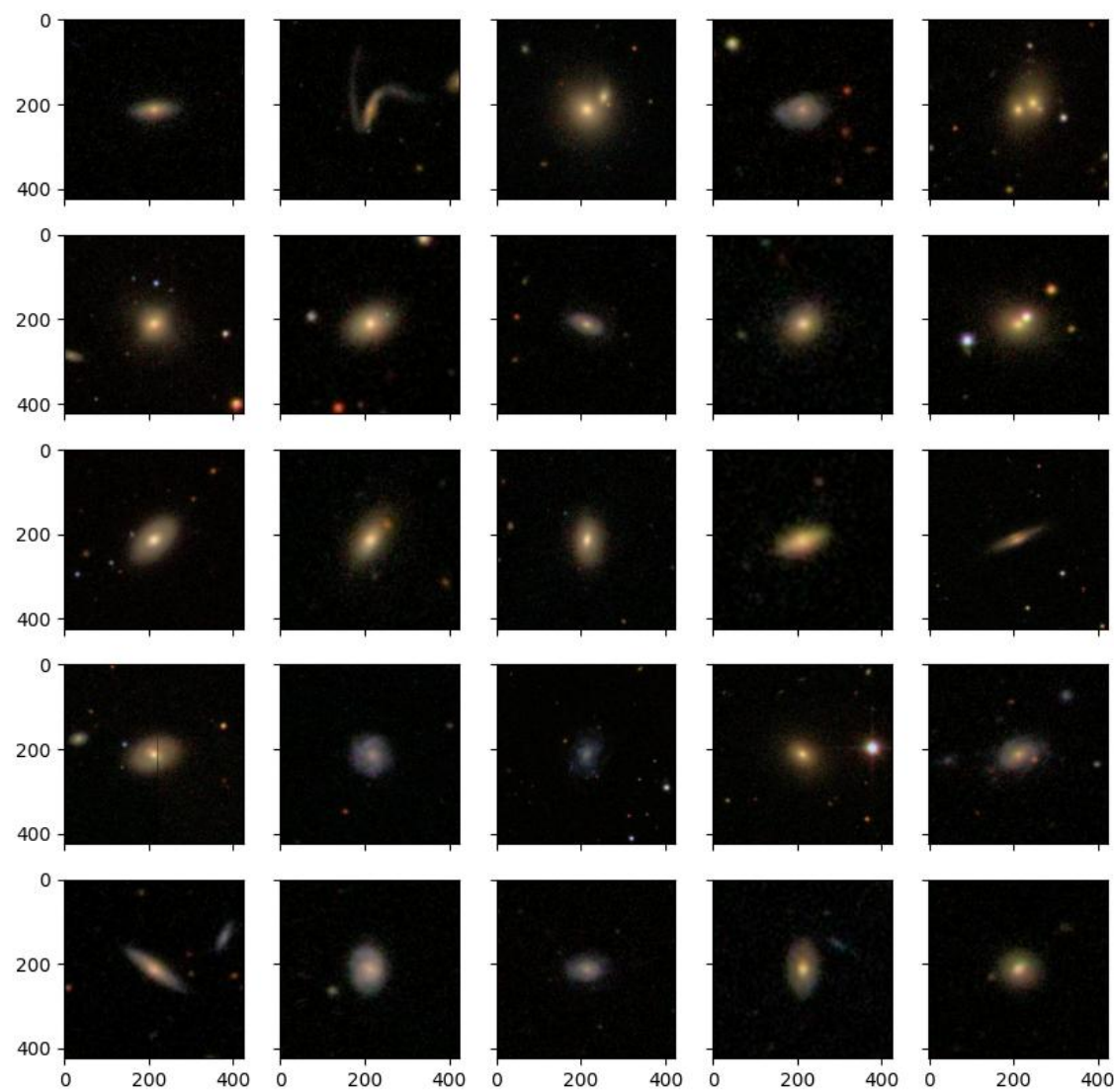
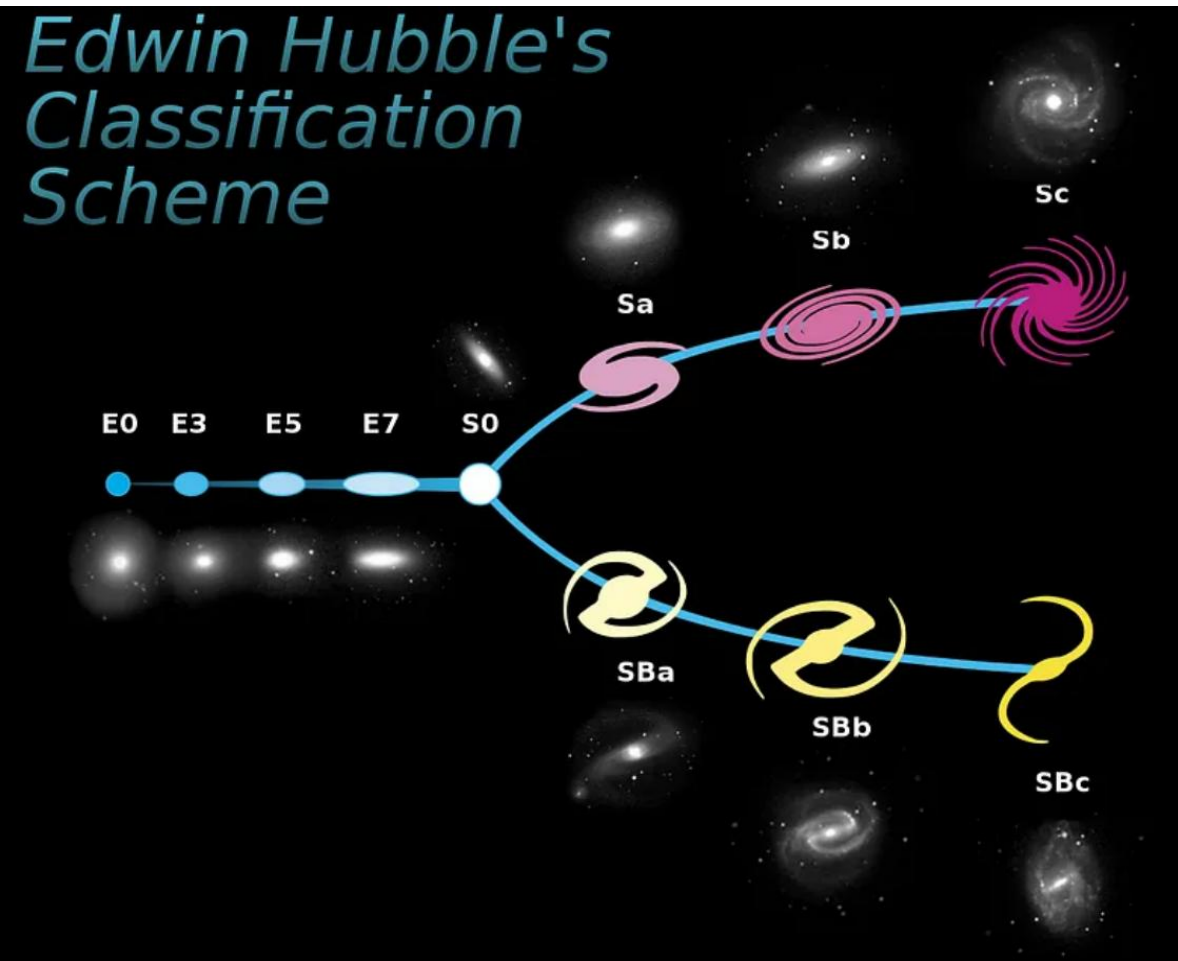
- 来一点儿神经网络……
- 貌似没有改善多少
- 机器学习是 GIGO (吃啥吐啥)
- 测光红移的最大问题是简并 (相同输入, 不同输出)
- 构型数据能帮忙消除一部分简并, 但不可能完美……
- 为什么? 来点儿最简单的几何与最简单的星系动力学
- 但我们也可以帮它……
- 或者引入更强的假设 (宇宙大尺度结构简介)
- 至少物理上明晰一些?



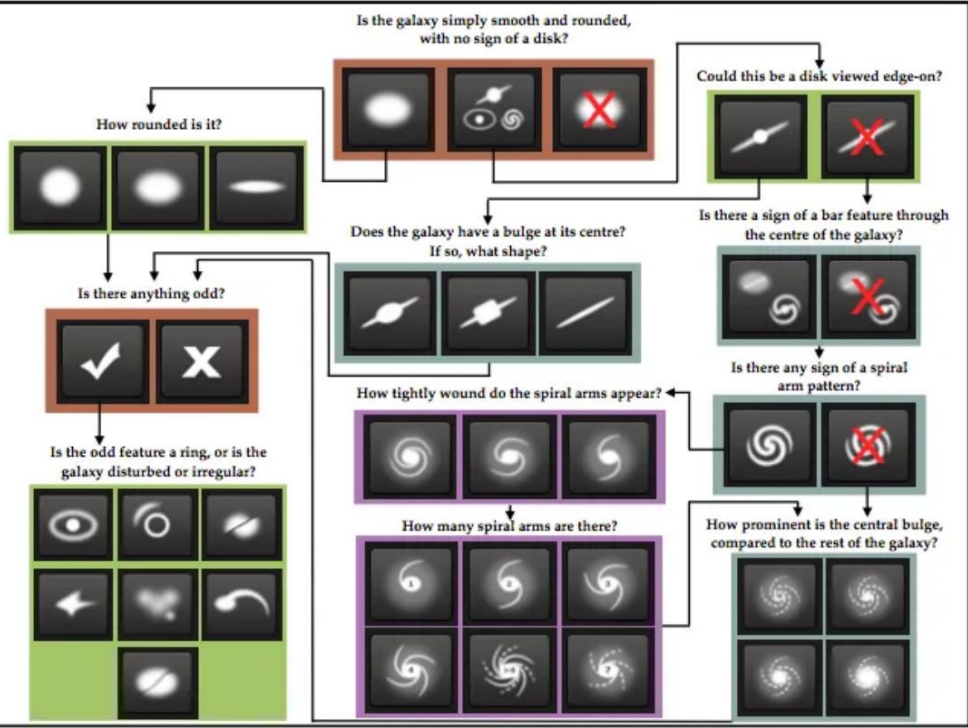
测光红移：……但做不了太完美

Pasquet et al.,
2019 A&A, 621,
A26

Edwin Hubble's Classification Scheme



星系分类：另一个经典例子



versus

Image
RGB channels
52x52

Convolution & Pooling
16 channels
24x24

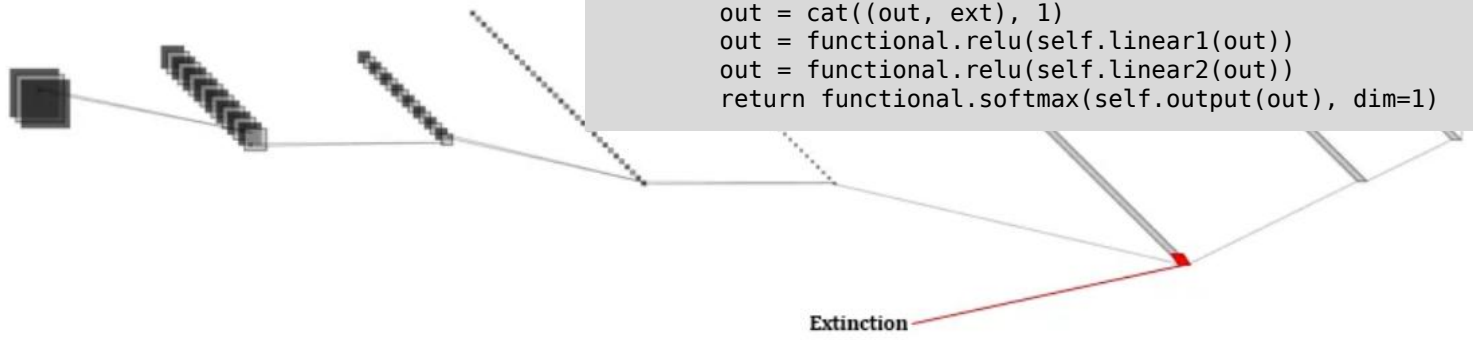
Convolution & Pooling
32 channels
12x12

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.conv_1 = self.conv_layer(3, 16, 3)
        self.max_pool1 = nn.MaxPool2d(3, 3)
        self.conv_2 = self.conv_layer(16, 32, 3)
        self.max_pool2 = nn.MaxPool2d(3, 3)
        self.flatten = nn.Flatten()
        self.linear1 = nn.Linear(513, 128)
        self.linear2 = nn.Linear(128, 32)
        self.output = nn.Linear(32, 3)

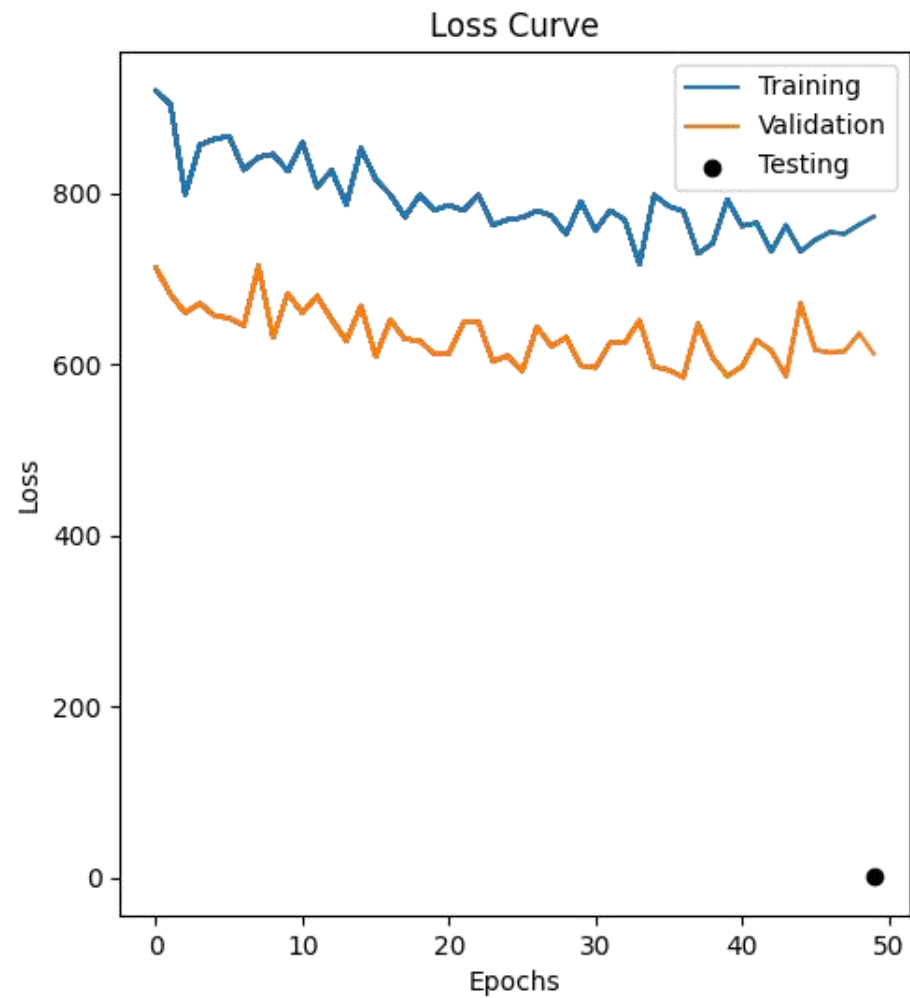
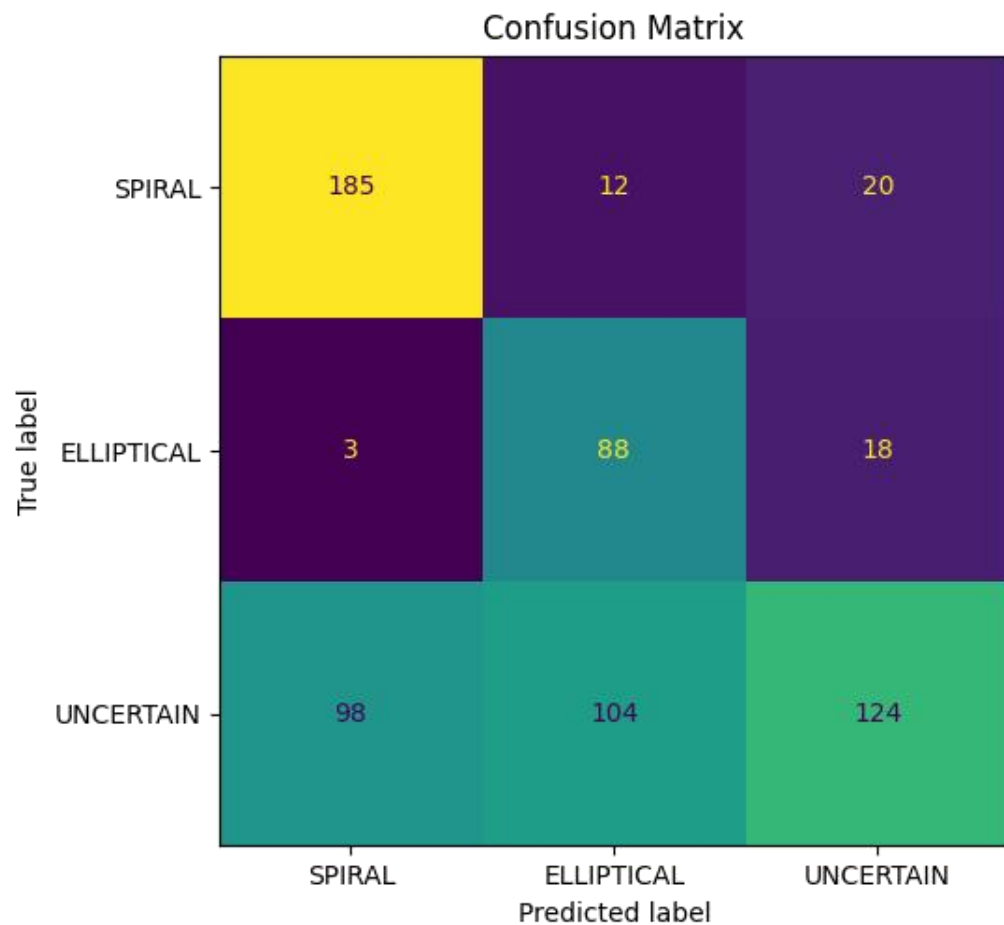
    @staticmethod
    def conv_layer(in_c, out_c, kernel):
        conv_layer = nn.Conv2d(in_c, out_c,
                                kernel_size=(kernel, kernel), padding=(0, 0))
        return conv_layer

    def forward(self, x, ext):
        out = functional.relu(self.conv_1(x))
        out = self.max_pool1(out)
        out = functional.relu(self.conv_2(out))
        out = self.max_pool2(out)
        out = out.view(out.size(0), -1)
        out = cat((out, ext), 1)
        out = functional.relu(self.linear1(out))
        out = functional.relu(self.linear2(out))
        return functional.softmax(self.output(out), dim=1)
```

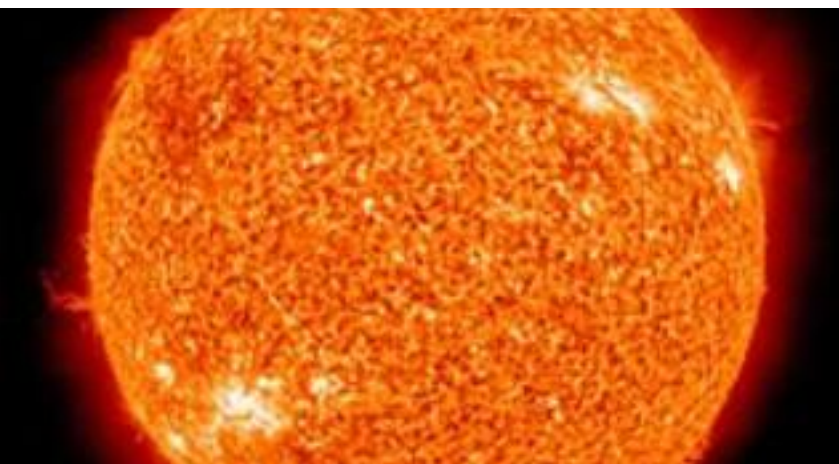
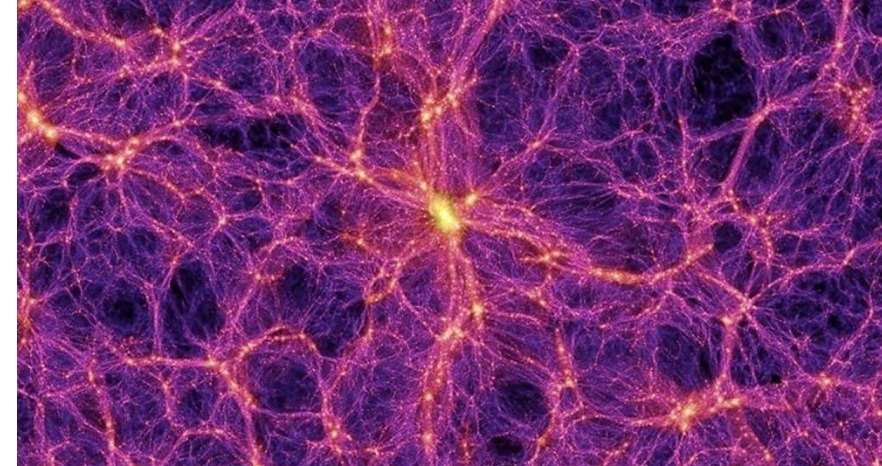
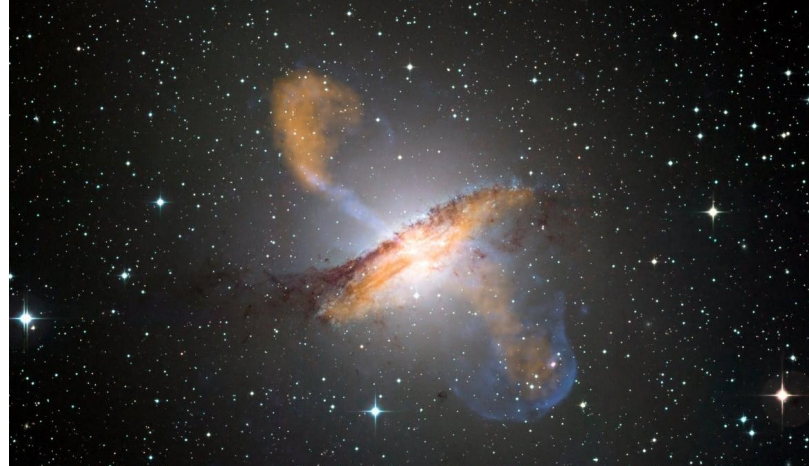
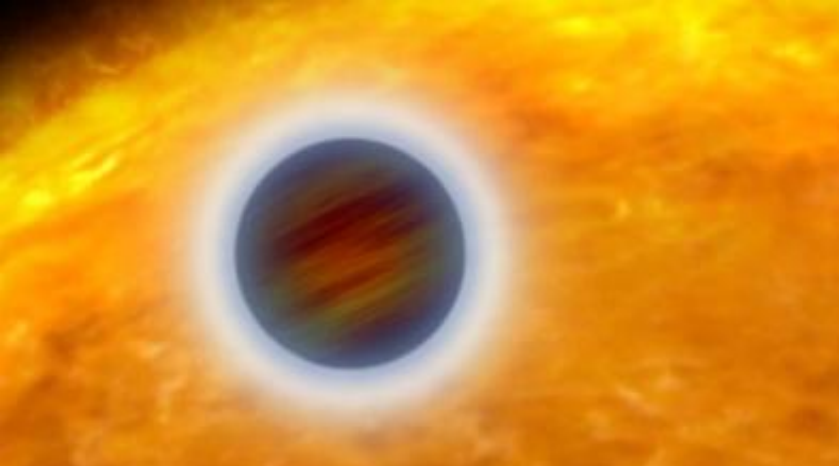


星系分类：经典例子

Learning Metrics

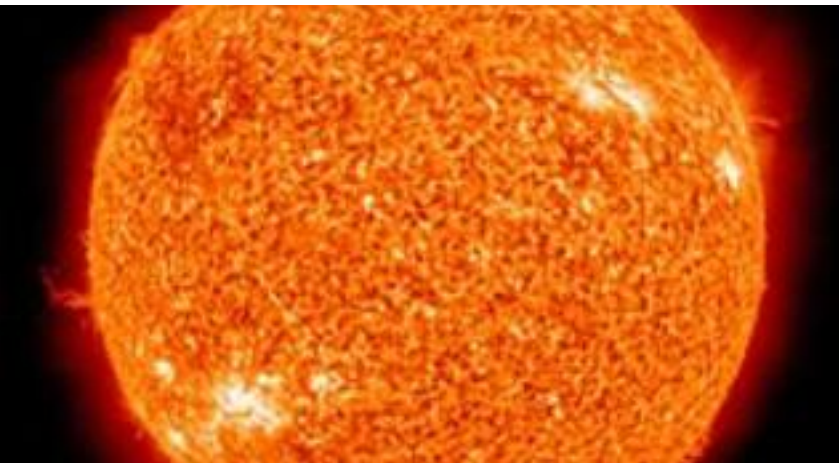
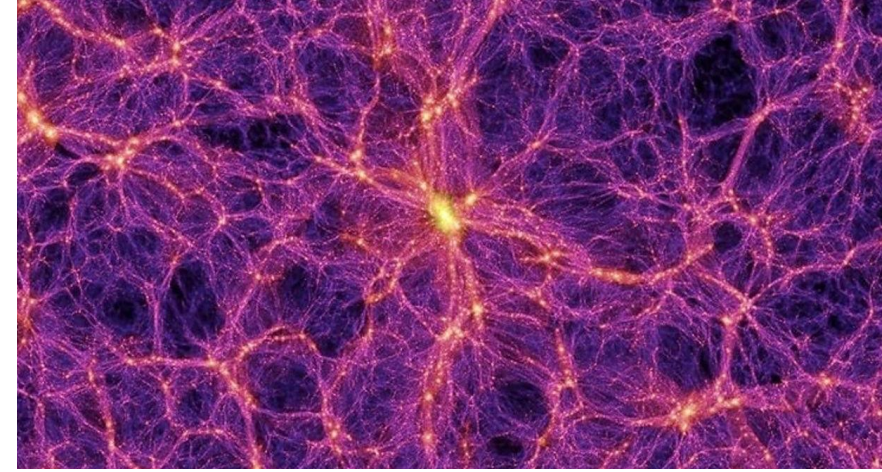
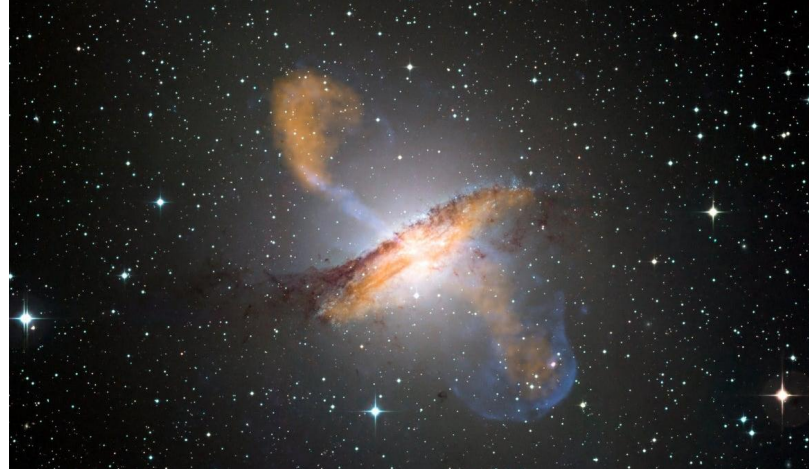
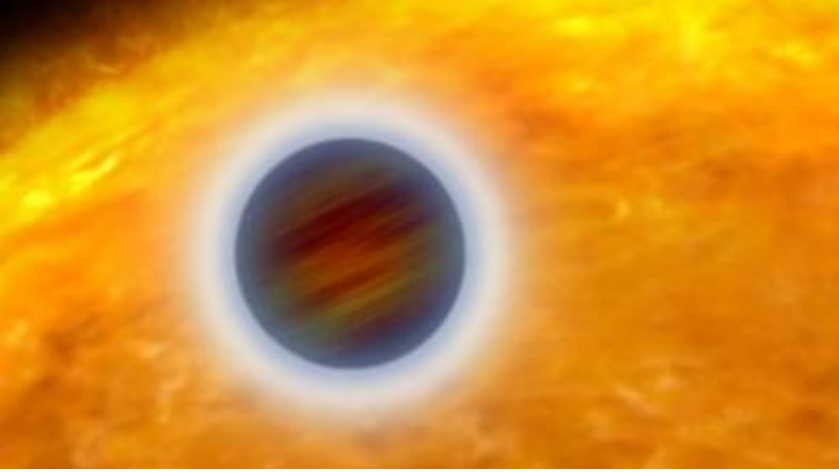


星系分类：经典例子



天体物理依靠数值模拟，因为——

- (1) “看天吃饭”（字面意思 + 引申意思皆然）
- (2) 问题太复杂，解析方法投降



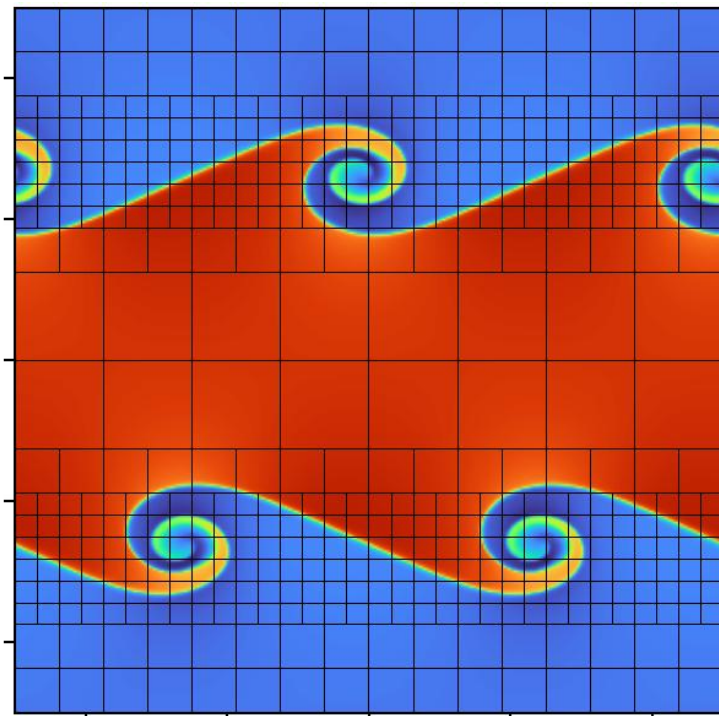
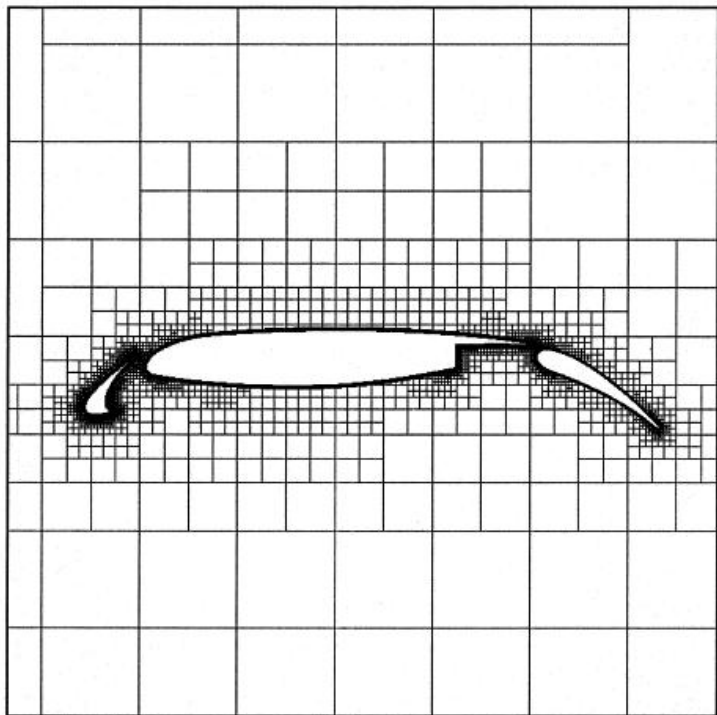
天体物理数值模拟的一般逻辑（不是所有）？

- 主干：（磁）流体动力学或粒子动力学

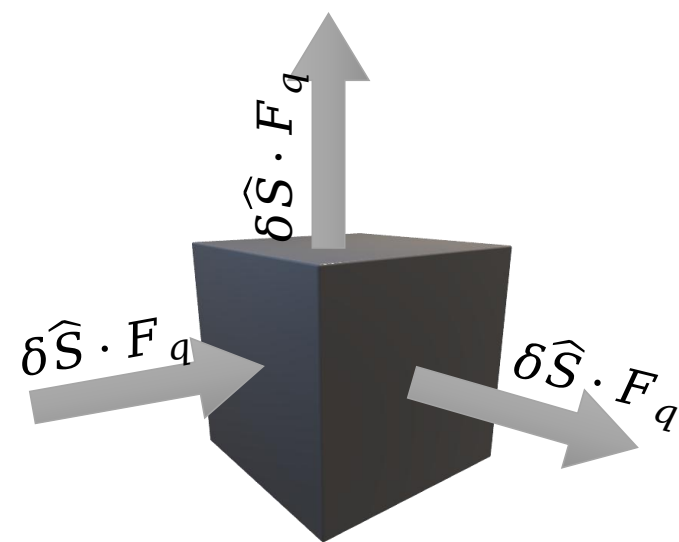
- 前台：微观机制

+ 细节 1：辐射传输问题

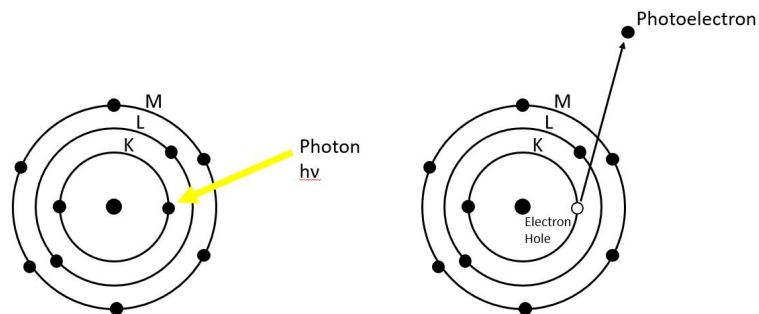
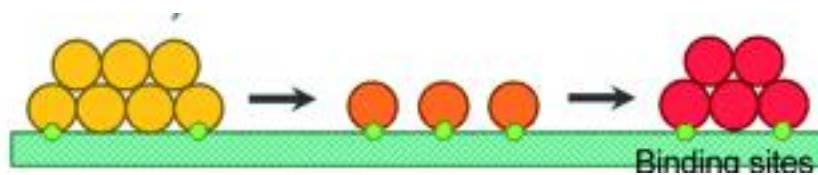
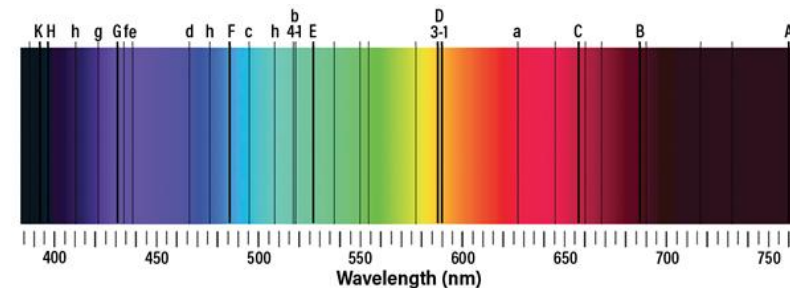
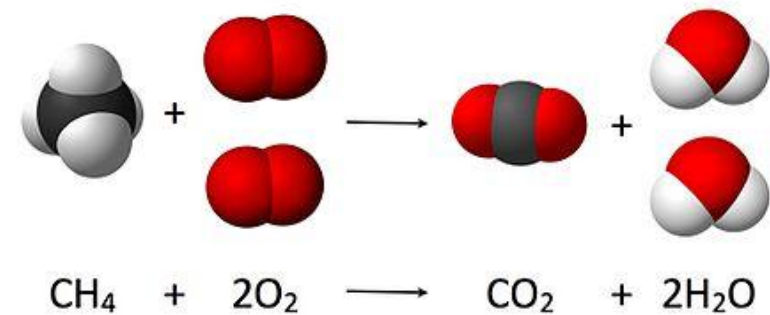
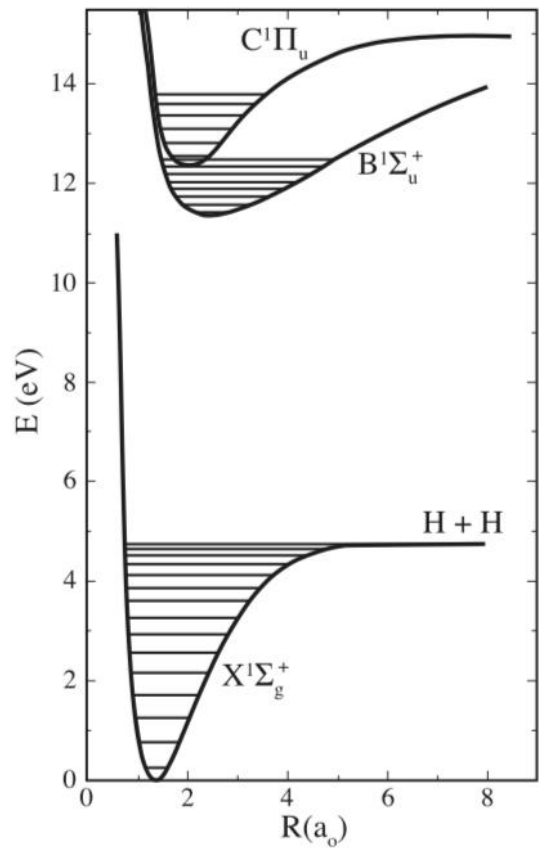
+ 细节 2：化学反应或核反应



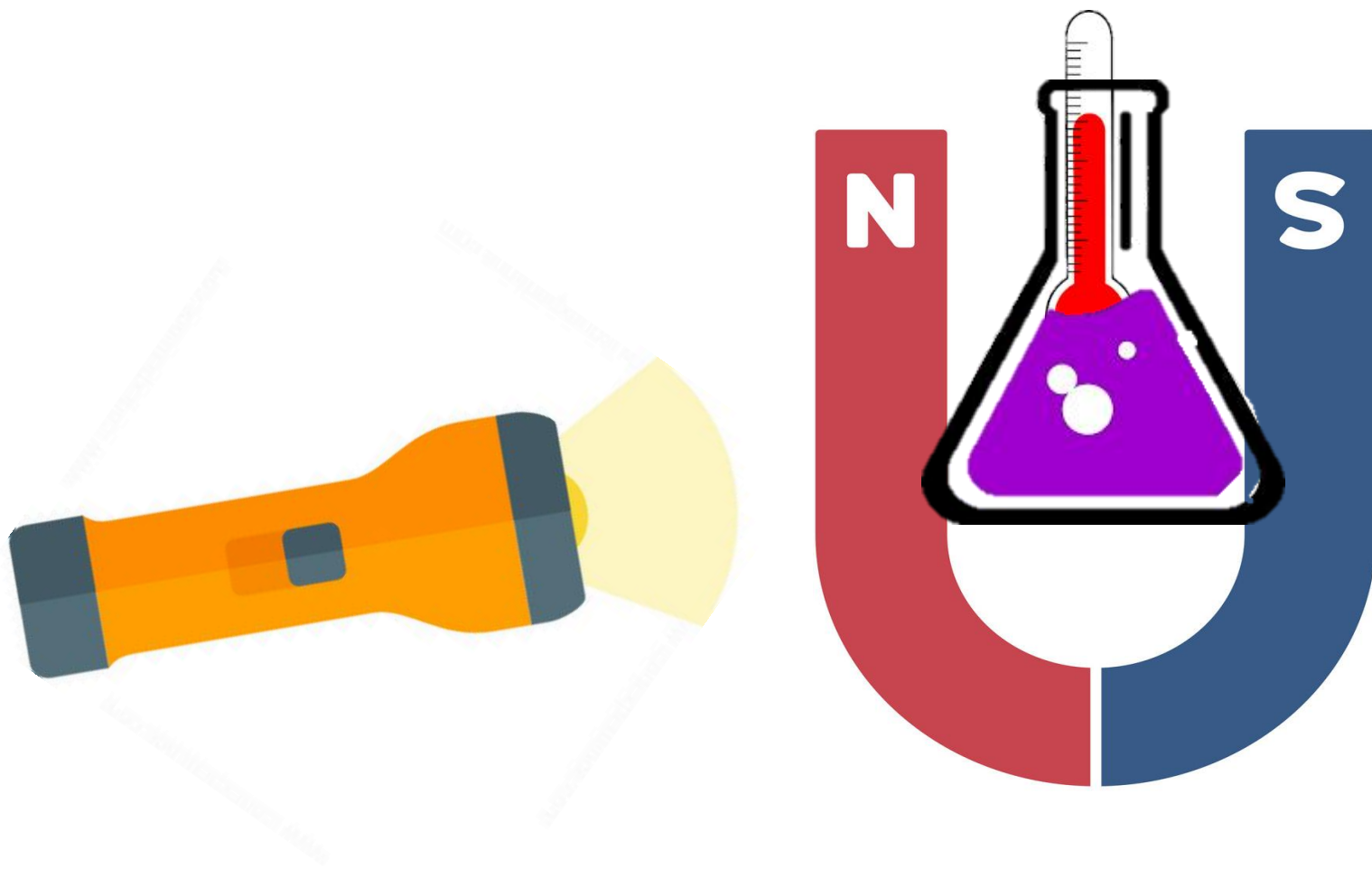
$$\Delta \int q \, dV = \sum_{\text{surface}} \int dt \int d\hat{S} \cdot F_q$$



流体力学的基本要义——守恒律



但同时还有微观物理.....



反应流体力学：不可或缺的鲁布·哥德堡机器

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0;$$

$$\partial_t (\rho \mathbf{v}) + \nabla \cdot \left(\rho \mathbf{v} \mathbf{v} - \frac{\mathbf{B} \mathbf{B}}{4\pi} + P_{\text{tot}} \mathbf{I} \right) = -\nabla \Phi;$$

$$\partial_t \mathbf{B} = \nabla \times (\mathbf{v} \times \mathbf{B} - c \mathbf{E}');$$

$$\partial_t \epsilon + \nabla \cdot \left[(\epsilon + P_{\text{tot}}) \mathbf{v} - \frac{(\mathbf{B} \cdot \mathbf{v}) \mathbf{B}}{4\pi} + \mathbf{S}' \right] = 0,$$

$$\mathbf{E}' = \frac{4\pi}{c^2} (\eta_{\text{O}} \mathbf{J} + \eta_{\text{H}} \mathbf{J} \times \mathbf{b} + \eta_{\text{A}} \mathbf{J}_{\perp}),$$



$$\frac{dn^i}{dt} = \mathcal{A}^i_{jk} n^j n^k + \mathcal{B}^i_j n^j;$$

$$\frac{d\epsilon}{dt} = \Gamma - \Lambda;$$

反应流体力学：不可或缺的鲁布·哥德堡机器

```

cmz_colab@kiaagpu2401 ~/scratch/cosmic
% mpiexec -bind-to user:0,1,2,3,64,65,66,67 -np 8 ./kratos cosmic_test_large.par
Reading input parameters from ascii : cosmic_test_large.par
Loaded metallicity data
Loaded cooling data from cooling.txt
Loaded heating data from ion_heat.txt
Reading particles... 134217728 particles read.
Init field... MG with 6 iters, defect = 9.39762e-05
Done.
=====
a = 0.02 dt_c = 0.548463
MG with 2 iters, defect = 5.46753e-05
cycle = 0, t = 5.484630e-01, dt = 5.484630e-01, Speed = 0.000000e+00
=====
a = 2.060000e-02 dt_c = 5.403723e-01
MG with 2 iters, defect = 8.930718e-05
cycle = 1, t = 1.088835e+00, dt = 5.403723e-01, Speed = 1.297068e+09
=====
a = 2.121800e-02 dt_c = 5.324619e-01
MG with 2 iters, defect = 6.847148e-05

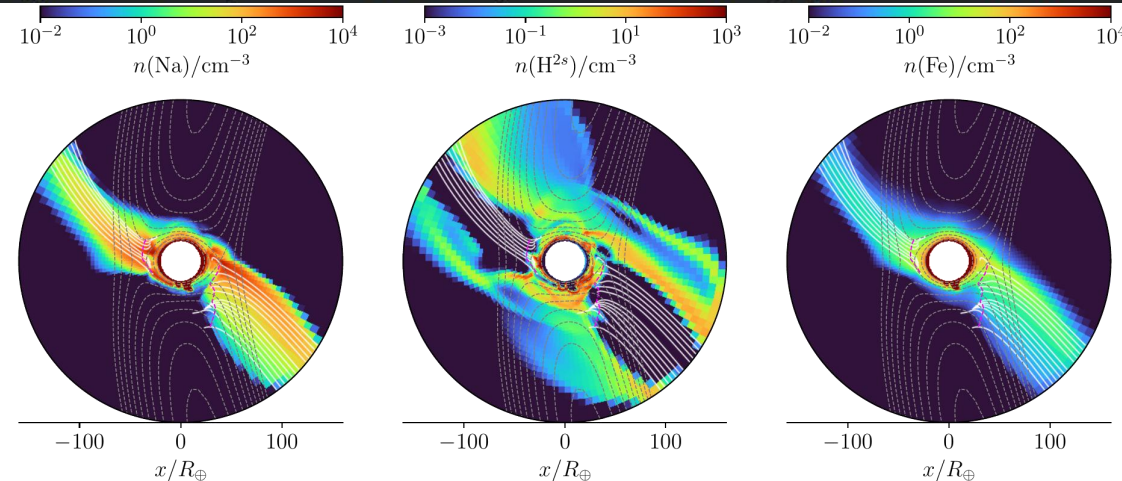
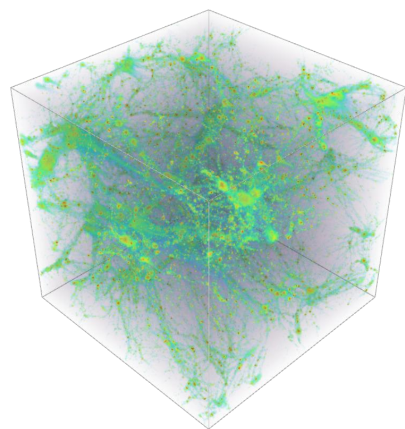
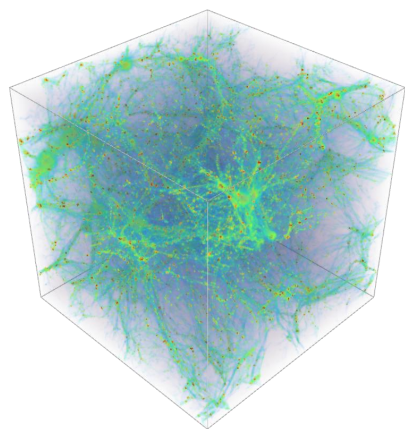
```

```

cycle = 231700, t = 7.731265e+01, dt = 3.462804e-04, Speed = 3.183928e+06
cycle = 231710, t = 7.731611e+01, dt = 3.462704e-04, Speed = 3.178442e+06
cycle = 231720, t = 7.731957e+01, dt = 3.462732e-04, Speed = 3.190248e+06
cycle = 231730, t = 7.732303e+01, dt = 3.462735e-04, Speed = 3.180139e+06
cycle = 231740, t = 7.732650e+01, dt = 3.462642e-04, Speed = 3.182938e+06
cycle = 231750, t = 7.732996e+01, dt = 3.462596e-04, Speed = 3.184222e+06
cycle = 231760, t = 7.733342e+01, dt = 3.462193e-04, Speed = 3.175107e+06
cycle = 231770, t = 7.733688e+01, dt = 3.461238e-04, Speed = 3.171489e+06
cycle = 231780, t = 7.734034e+01, dt = 3.463324e-04, Speed = 3.179861e+06
cycle = 231790, t = 7.734381e+01, dt = 3.472235e-04, Speed = 3.184096e+06
cycle = 231800, t = 7.734729e+01, dt = 3.483252e-04, Speed = 3.179090e+06
cycle = 231810, t = 7.735066e+01, dt = 3.487364e-04, Speed = 3.182025e+06
cycle = 231820, t = 7.735415e+01, dt = 3.481309e-04, Speed = 3.180104e+06
cycle = 231830, t = 7.735762e+01, dt = 3.472681e-04, Speed = 3.181164e+06
cycle = 231840, t = 7.736109e+01, dt = 3.467191e-04, Speed = 3.182695e+06
cycle = 231850, t = 7.736428e+01, dt = 7.225673e-05, Speed = 3.204879e+06
cycle = 231860, t = 7.736488e+01, dt = 6.931734e-05, Speed = 3.542428e+06
cycle = 231870, t = 7.736563e+01, dt = 7.922358e-05, Speed = 3.515064e+06
cycle = 231880, t = 7.736641e+01, dt = 7.569162e-05, Speed = 3.503148e+06
cycle = 231890, t = 7.736724e+01, dt = 8.901633e-05, Speed = 3.500003e+06

```

a = 1.00, z = -0.00



有了微观机制，事情变得复杂起来

Wang 2025a
ApJS, 227, 63
Wang 2025b
arXiv:2504.04941

```

cmz_colab@kiaagpu2401 ~/scratch/cosmic
% mpiexec -bind-to user:0,1,2,3,64,65,66,67 -np 8 ./kratos cosmic_test_large.par
Reading input parameters from ascii : cosmic_test_large.par
Loaded metallicity data
Loaded cooling data from cooling.txt
Loaded heating data from ion_heat.txt
Reading particles... 134217728 particles read.
Init field... MG with 6 iters, defect = 9.39762e-05
Done.
=====
a = 0.02 dt_c = 0.548463
MG with 2 iters, defect = 5.46753e-05
cycle = 0, t = 5.484630e-01, dt = 5.484630e-01, Speed = 0.000000e+00
=====
a = 2.060000e-02 dt_c = 5.403723e-01
MG with 2 iters, defect = 8.930718e-05
cycle = 1, t = 1.088835e+00, dt = 5.403723e-01, Speed = 1.297068e+09
=====
a = 2.121800e-02 dt_c = 5.324619e-01
MG with 2 iters, defect = 6.847148e-05

```

```

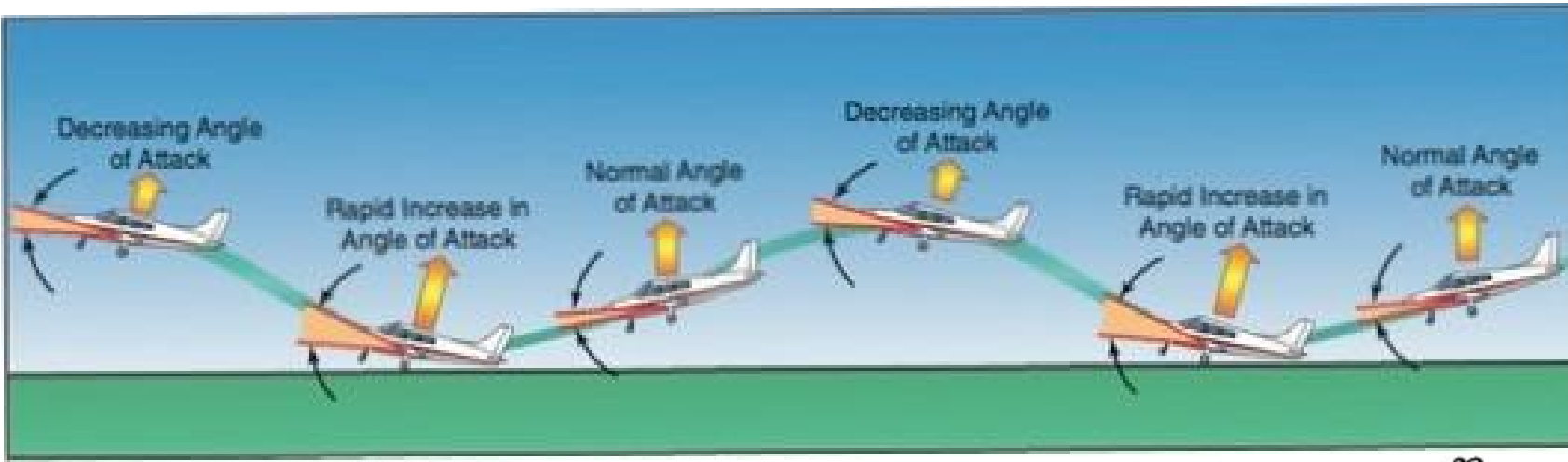
cycle = 231700, t = 7.731265e+01, dt = 3.462804e-04, Speed = 3.183928e+06
cycle = 231710, t = 7.731611e+01, dt = 3.462704e-04, Speed = 3.178442e+06
cycle = 231720, t = 7.731957e+01, dt = 3.462732e-04, Speed = 3.190248e+06
cycle = 231730, t = 7.732303e+01, dt = 3.462735e-04, Speed = 3.180139e+06
cycle = 231740, t = 7.732650e+01, dt = 3.462642e-04, Speed = 3.182938e+06
cycle = 231750, t = 7.732996e+01, dt = 3.462596e-04, Speed = 3.184222e+06
cycle = 231760, t = 7.733342e+01, dt = 3.462193e-04, Speed = 3.175107e+06
cycle = 231770, t = 7.733688e+01, dt = 3.461238e-04, Speed = 3.171489e+06
cycle = 231780, t = 7.734034e+01, dt = 3.463324e-04, Speed = 3.179861e+06
cycle = 231790, t = 7.734381e+01, dt = 3.472235e-04, Speed = 3.184096e+06
cycle = 231800, t = 7.734729e+01, dt = 3.483252e-04, Speed = 3.179090e+06
cycle = 231810, t = 7.735066e+01, dt = 3.487364e-04, Speed = 3.182025e+06
cycle = 231820, t = 7.735415e+01, dt = 3.481309e-04, Speed = 3.180104e+06
cycle = 231830, t = 7.735762e+01, dt = 3.472681e-04, Speed = 3.181164e+06
cycle = 231840, t = 7.736109e+01, dt = 3.467191e-04, Speed = 3.182695e+06
cycle = 231850, t = 7.736428e+01, dt = 7.225673e-05, Speed = 3.204879e+06
cycle = 231860, t = 7.736488e+01, dt = 6.931734e-05, Speed = 3.542428e+06
cycle = 231870, t = 7.736563e+01, dt = 7.922358e-05, Speed = 3.515064e+06
cycle = 231880, t = 7.736641e+01, dt = 7.569162e-05, Speed = 3.503148e+06
cycle = 231890, t = 7.736724e+01, dt = 8.901633e-05, Speed = 3.500003e+06

```

流体动力学	当然	必须的
多重网格自引力	有的，有的	不需要，行星自引力即可
粒子模块	不然呢？暗物质怎么办？	也有的，不然辐射怎么办？
反应（化学/核）	……别哪壶不开提哪壶	这是我的优势，也是我的短板

为什么那么慢（两个数量级）？！

Wang 2025a
ApJS, 227, 63
Wang 2025b
arXiv:2504.04941



$$\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{f}(x) \longrightarrow \mathbf{y}(x) = \sum_{t=1}^n \kappa_t e^{\lambda_t x} \mathbf{c}_t + \mathbf{g}(x)$$

- 刚性比例：最快和最慢的模式之比，也即是矩阵的条件数
- 非线性——雅可比矩阵 $J_{ij}|_{x_0} \equiv \left[\frac{\partial}{\partial y_j} \left(\frac{dy_i}{dx} \right) \right]$

什么是刚性？为何要用隐式方法？



驾驶员/飞行员诱发振荡 (PIO)



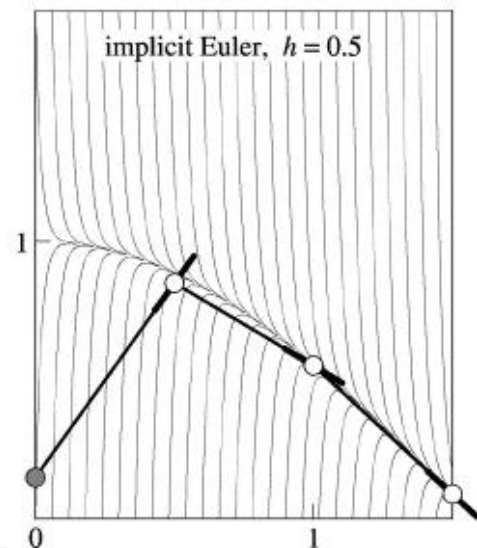
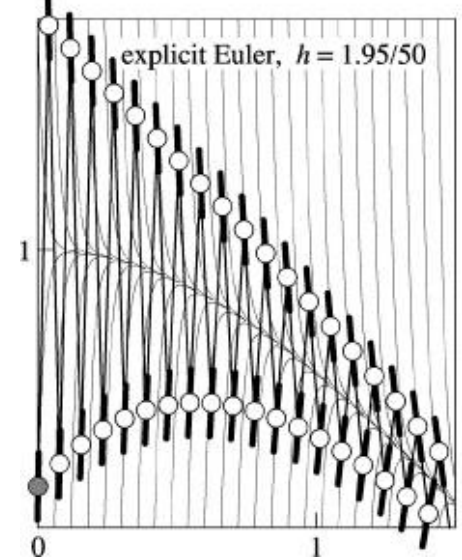
驾驶员/飞行员诱发振荡 (PIO)

- 刚性方程导致求解失败：冲过头（overshoot）
- 现实生活中的解决：事先建立预期，“先看终点再起步”
- 如果令到达点而非出发点的导数等同于割线斜率：

$$y_i|_{x=x_1} = y_i|_{x=x_0} + (x_1 - x_0) \left[\frac{dy_i}{dx} \right]_{x=x_1}$$

- 上式是 $y_{x=x_1}$ 的隐函数，此类方法称为“隐式方法”
- 解析求 $y_{x=x_1}$ 不现实，可通过一步牛顿法估计（半隐式）：

$$y_i|_{x=x_1} \simeq y_i|_{x=x_0} + \Delta x (\delta_{ij} - J_{ij}|_{x_0} \Delta x)^{-1} \left[\frac{dy_j}{dx} \right]_{x=x_0}$$



刚性与隐式方法

- 解析求 $y_{x=x_1}$ 不现实，可通过一步牛顿法估计（半隐式）：

$$y_i|_{x=x_1} \simeq y_i|_{x=x_0} + \Delta x (\delta_{ij} - J_{ij}|_{x_0} \Delta x)^{-1} \left[\frac{dy_j}{dx} \right]_{x=x_0}$$

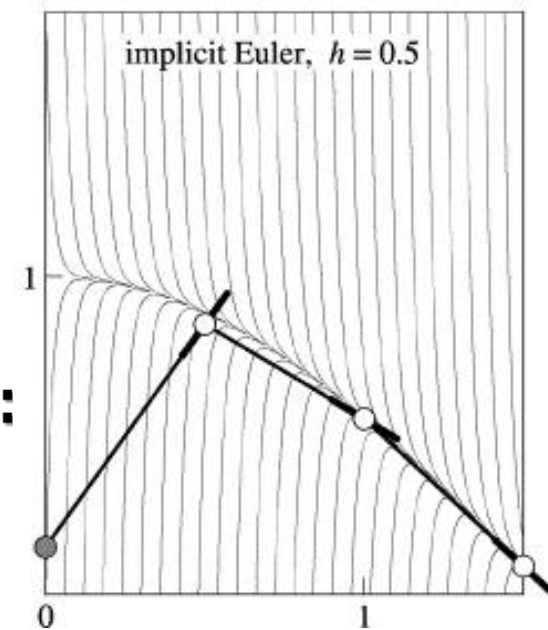
- 这一求解方式的原理为何？
- 为了使 $y_{x=x_1} \equiv y_1$ 处的切线斜率给出割线斜率（单步隐式法）：

$$F(y_1) = \frac{y_1 - y_0}{\Delta x} - f(y_1; x_1)$$

- 以牛顿法求解上述方程， y_1 的迭代求解的递推式为：

$$y_1' = y_1 - \left[\frac{\partial F}{\partial y_1} \right]^{-1} F(y_1) = y_1 - \left(\frac{I}{\Delta x} - J \right)^{-1} \left[\frac{y_1 - y_0}{\Delta x} - f(y_1; x_1) \right]$$

- 令 y_1 的首次猜测为 $y_0 \equiv y_{x=x_0}$ ，即有原式。



刚性与隐式方法

- 反应方程组大多具有**强刚性 (stiff)**

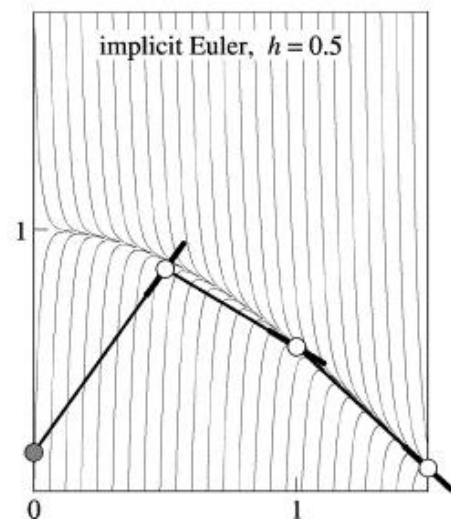
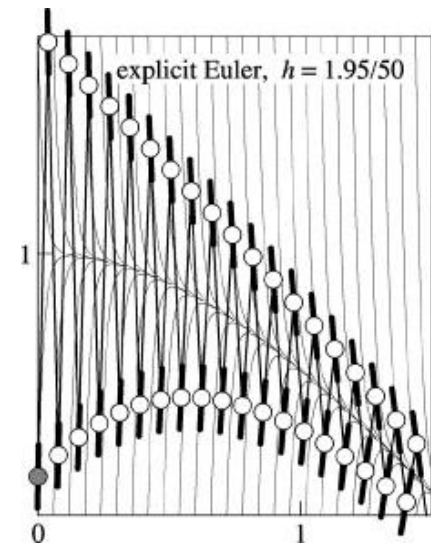
- **刚性**问题需要使用**隐式方法**

- **隐式方法**需要矩阵求解

$$y_i|_{x=x_1} \simeq y_i|_{x=x_0} + \Delta x (\delta_{ij} - J_{ij}|_{x_0} \Delta x)^{-1} \left[\frac{dy_j}{dx} \right]_{x=x_0}$$

- 矩阵求解复杂度为 $O(N^3)$ (N 为有效组分数)

- $O(N^3)$ 算法打死也快不起来，哪怕你用上了GPU...



微分方程组刚性带来的困难

Algorithm 1: Parallel LU decomposition of matrices on multi-thread devices

Data : Matrix to be decomposed a_{ij} (size $N \times N$)

Result: LU-decomposed matrix stored in a_{ij} ;

Permutation vector p_i .

```

1  $i \leftarrow$  thread index ( $0 \leq i \leq N - 1$ ) ;
2  $v_i \leftarrow$  maximum absolute value in the  $i$ th row of  $a_{ij}$ 
3 for  $j \leftarrow 0$  to  $N - 1$  do
  /* Compute U and unrescaled L          */
4  for  $k \leftarrow 0$  to  $\min\{i - 1, j - 1\}$  do
     $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$ ;
  /* Pivoting: Let diagonals hold greatest
    possible absolute value          */
5  synchronize threads;
6  if  $i \geq j$  then  $t_i \leftarrow |a_{ij}|/v_i$ ;
7   $i_{\max} \leftarrow$  index for the maximum  $t_i$  with  $i \geq j$ ;
8  synchronize threads;
9  if  $j \neq i_{\max}$  then swap( $a_{i_{\max},i}, a_{ji}$ );
10 if  $i$  is 0 then swap( $v_{i_{\max}}, v_j$ ),  $p_i \leftarrow i_{\max}$ ;
  /* Rescale L                          */
11 synchronize threads;
12 if  $j < i$  then  $a_{ij} \leftarrow a_{ij}/a_{jj}$  ;
13 end

```

$$y_i|_{x=x_1} \simeq y_i|_{x=x_0} + \Delta x (\delta_{ij} - J_{ij}|_{x_0} \Delta x)^{-1} \left[\frac{dy_j}{dx} \right]_{x=x_0}$$

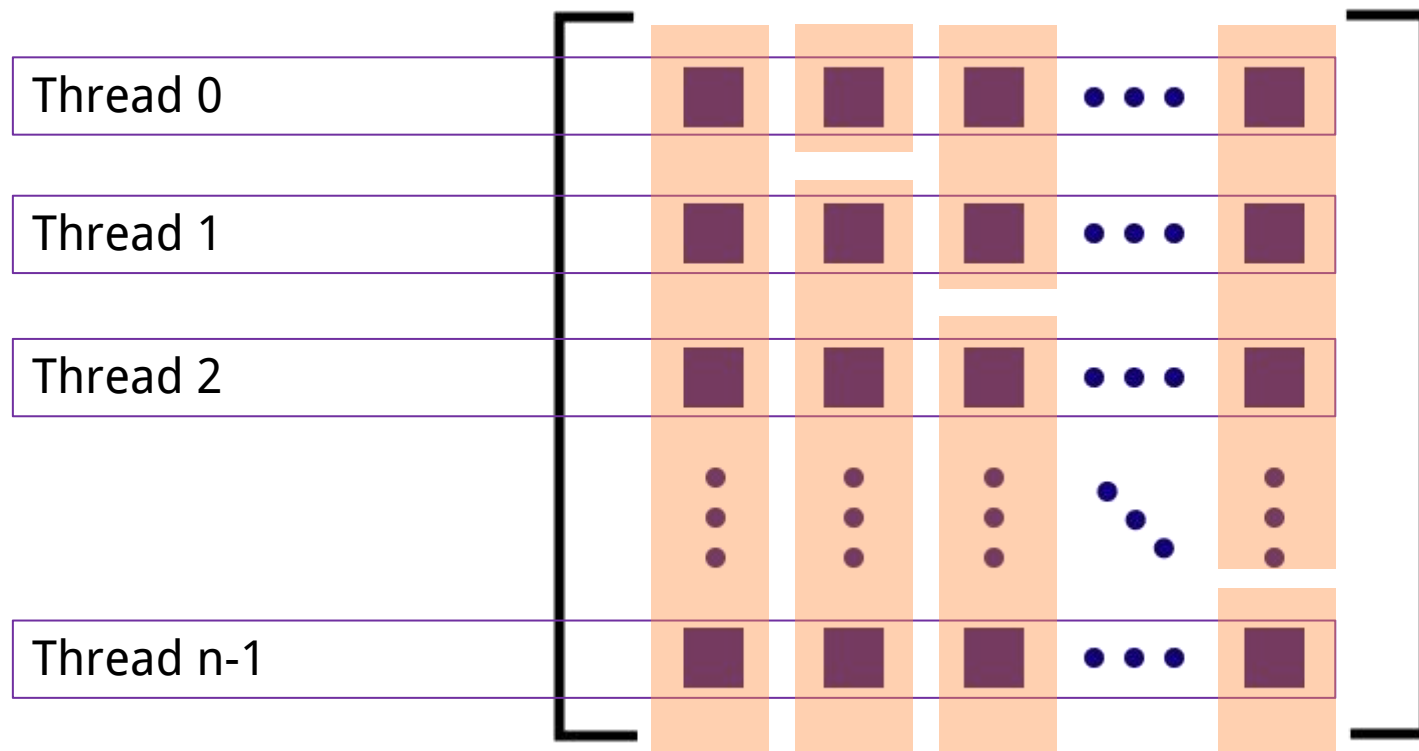
for-loop

0

1

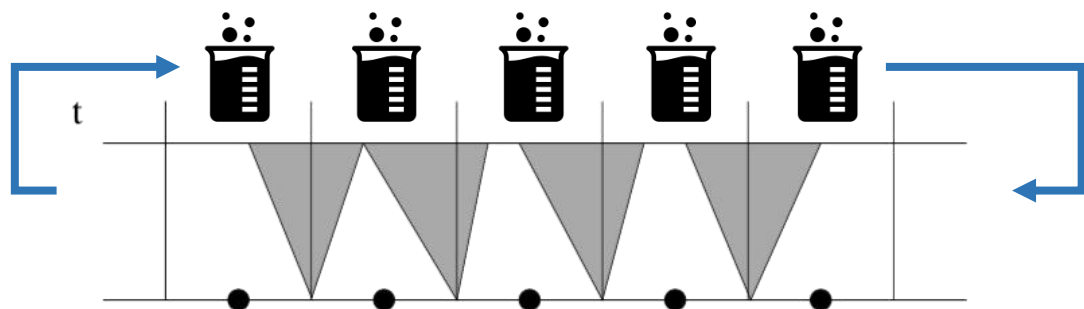
2

n-1

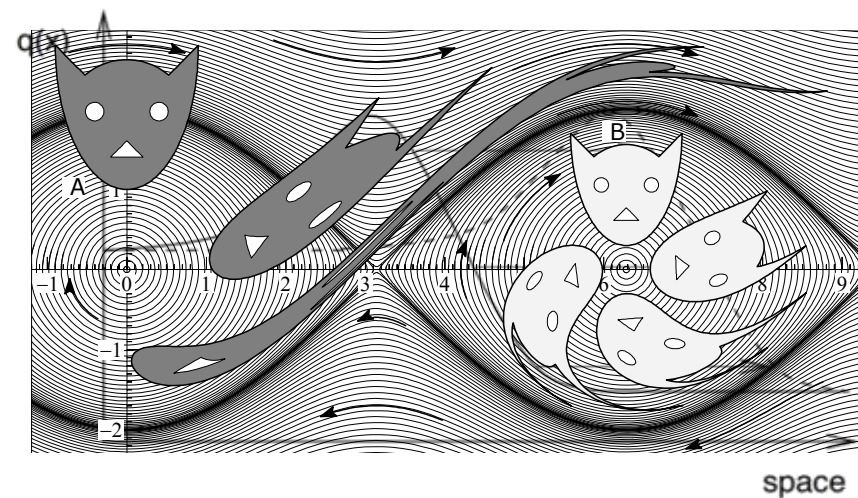
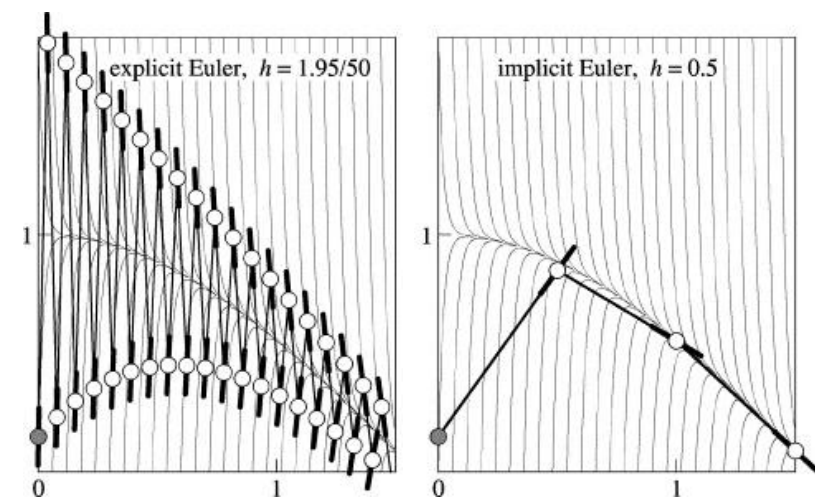


附录：GPU 大并发小矩阵的 LU 分解

- 针对反应方程组的加速显然具有更高优先级
- 反应动力学方程组（符合质量作用定律）
通常具有类抛物形式
 - 初始误差总会消散（反应通常渐趋平衡）



- 相反地，流体动力学是双曲方程，保持守恒性
 - 任何误差皆会保留、累计



谁的困难，谁的机遇——为什么？

- 将圆锥曲线中的坐标的乘幂换为对坐标的导数

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G$$

- 双曲：时间和空间的二阶导数项符号相反

- 符号相反意味着可以实数域内因式分解：

$$(\partial_t + \partial_x)(\partial_t - \partial_x)u = 0$$

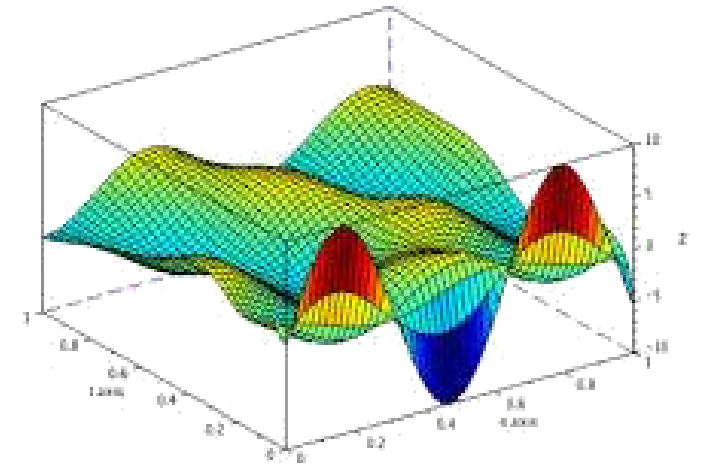
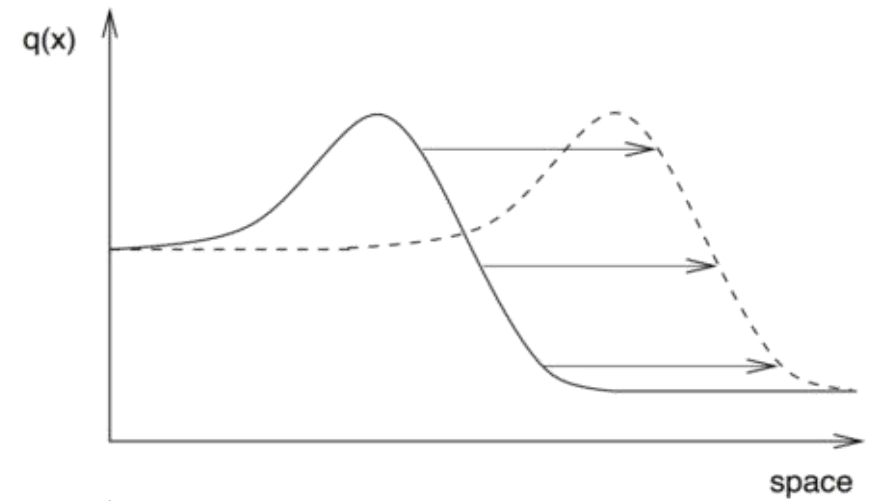
- 因式分解降阶显示了守恒律：

$$u(t; x) = u_0(x - vt)$$

- 椭圆：没有时间，空间的二阶导数项符号相同

- 亥姆霍茨定理：被边界条件唯一给定（例外？）

- 抛物：数值上等同于椭圆（椭圆用抛物求解）



微分方程的大致分类

- 考虑一个简单的二阶微分方程：

$$\frac{d^2 u}{dx^2} = -v(x), \quad u|_{x=-1} = u|_{x=1} = 0, \quad v(x) = \Theta(0.25 - |x|)$$

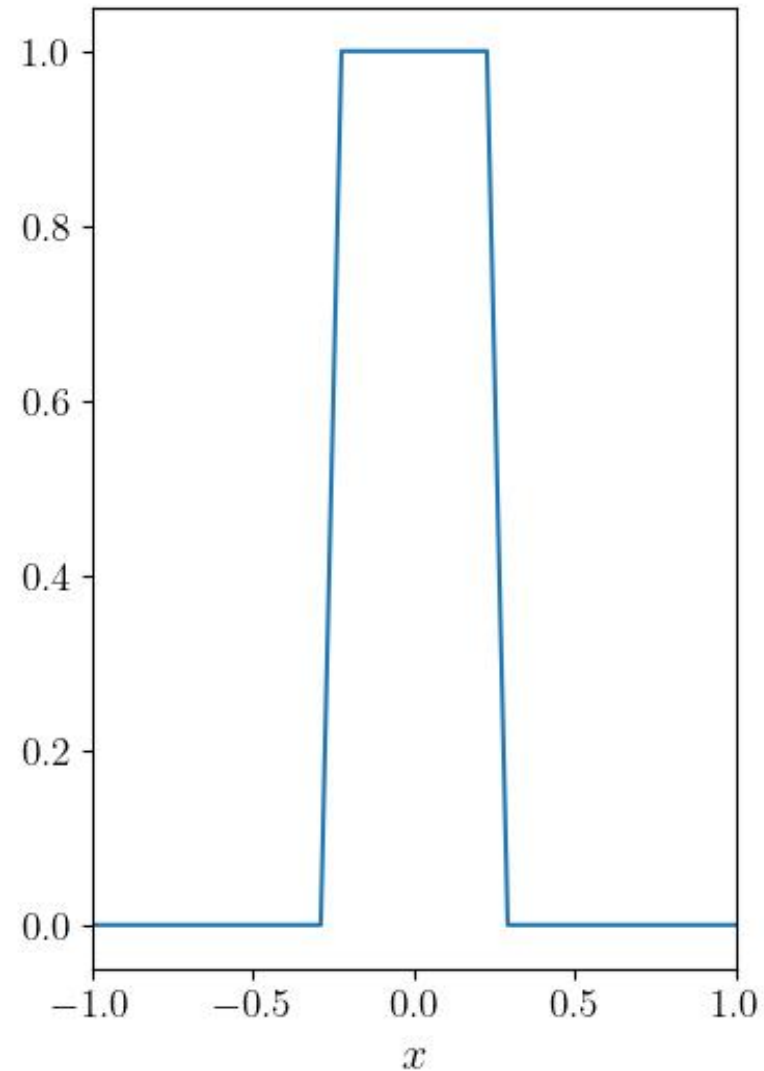
- 二阶导数的离散化表达？

$$\frac{2u_n - u_{n+1} - u_{n-1}}{\Delta x^2} = v(x_n)$$

- 进行简单变形

$$u'_n = \frac{u_{n+1} + u_{n-1} + v(x_n)\Delta x^2}{2}$$

- 如果 $\{u_n\}$ 本身构成方程的解，则上式应该总是成立
- 但若 $\{u_n\}$ 与解有差距，为何仍然成立？
- 最粗浅简单的理解：函数空间中的不动点迭代



先来看看椭圆问题的松弛方法

- 最粗浅简单的理解：不动点迭代

$$u'_n = \frac{u_{n+1} + u_{n-1} + v(x_n)\Delta x^2}{2}$$

- 或将方程视为矩阵乘向量的形式 ($b_n = v(x_n)\Delta x^2$) :

$$\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_0 (= 0) \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N (= 0) \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{N-1} \\ b_N \end{bmatrix}$$

- 简记为 $Au = b$ ，并将 A 做额外分解.....

椭圆问题的松弛方法

- 对迭代 $u' = Ru + D^{-1}b$, 令“真值”为 \tilde{u} 、当前解与真值的“误差” $e \equiv u - \tilde{u}$
- 将迭代式左右皆减去 \tilde{u} , 因 $\tilde{u} = R\tilde{u} + D^{-1}b$ 总是成立, 得到: $e' = Re$
- 不断迭代, 就是往残差 e 上“不断乘 R ”
- 对 R 做本征值分解: $R \equiv \Lambda^{-1} \text{diag}(\lambda_0, \dots, \lambda_N)$. 容易得到: $R^n \equiv \Lambda^{-1} \text{diag}(\lambda_0^n, \dots, \lambda_N^n) \Lambda$
- 只要 $|\lambda_i| < 1$ 就行, 而 $\lambda_i = \cos(\quad / \quad)$
- 此方法称为雅可比迭代, 更好的高斯—赛德尔迭代同理

$$u'_n = \frac{u_{n+1} + u_{n-1} + v(x_n)\Delta x^2}{2}$$

$$\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_0 (= 0) \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N (= 0) \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{N-1} \\ b_N \end{bmatrix}$$

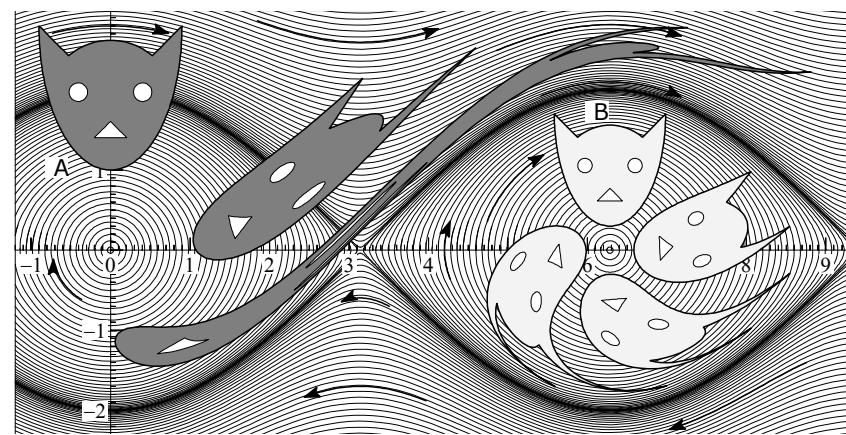
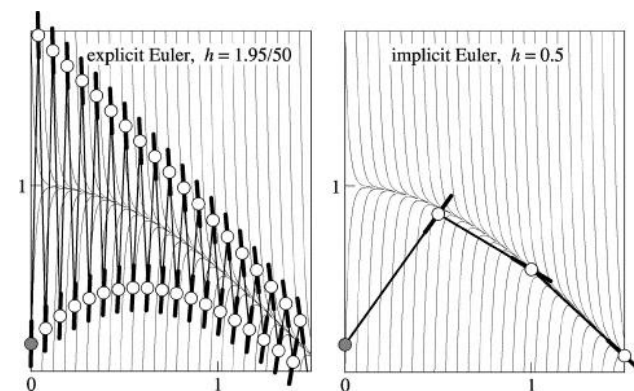
$$U \equiv \begin{bmatrix} 0 & -1 & & & \\ & 0 & -1 & & \\ & & 0 & -1 & \\ & & & \ddots & \\ & & & & 0 & -1 \\ & & & & & 0 \end{bmatrix}, \quad L \equiv U^T, \quad D = 2I$$

$$R \equiv -D^{-1}(U + L)$$

椭圆问题的松弛方法

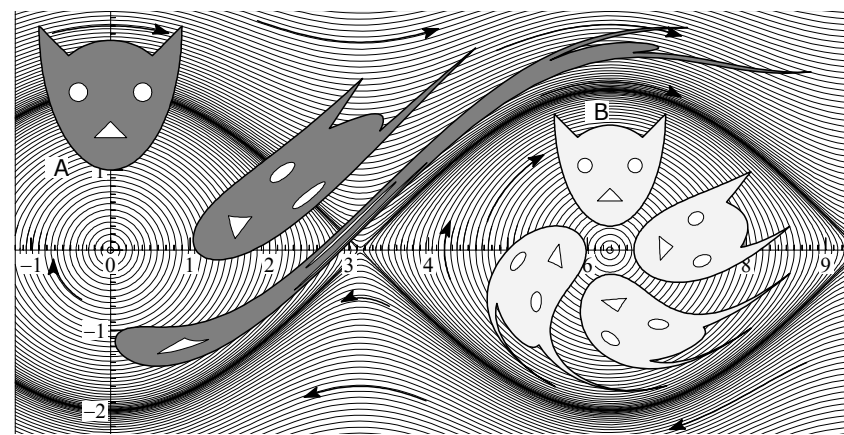
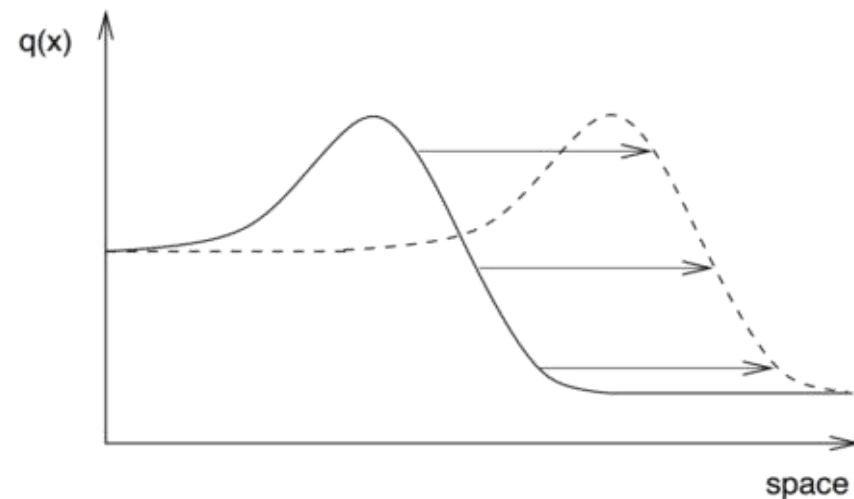
- 椭圆问题最终是会收敛到“目标轨道”上去的
 - $R^n \equiv \Lambda^{-1} \text{diag}(\lambda_0^n, \dots, \lambda_N^n) \Lambda$
 - 只要所有的本征值都满足 $|\lambda_i| < 1$ ，迭代就收敛
 - 初始时刻对 \tilde{u} 的偏差到最后会被 λ_i^n “压掉”
 - 压得越快，传统方法反而越麻烦（刚性）
 - 更合适采用代理模型（压得越快越好用）
- 双曲问题最终是没有“最终轨道”可供收敛的
 - $u(t; x) = u_0(x - vt)$ 意味着任意扰动都会保持
 - 传统算法优势很大，暂时不必采用代理模型

$$u'_n = \frac{u_{n+1} + u_{n-1} + v(x_n) \Delta x^2}{2}$$



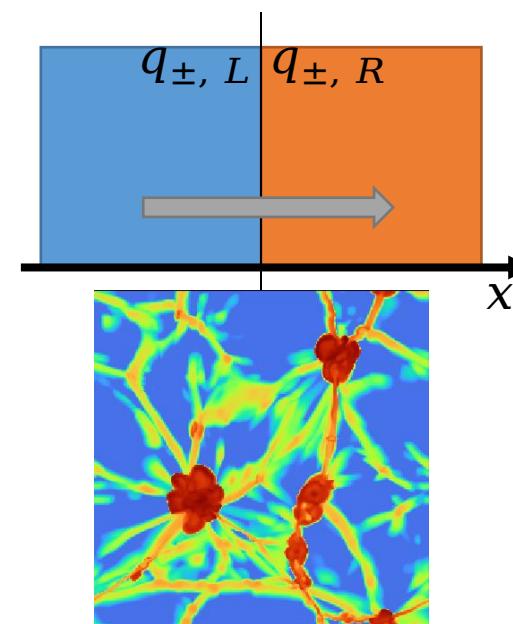
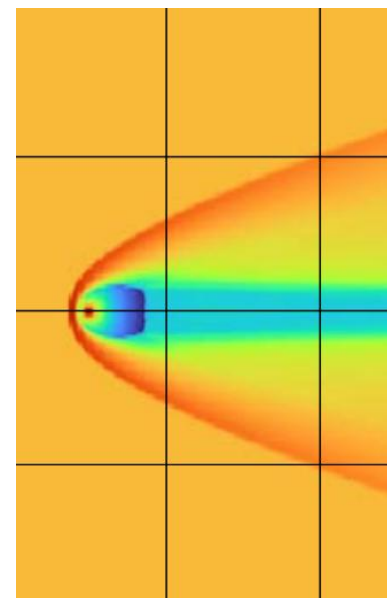
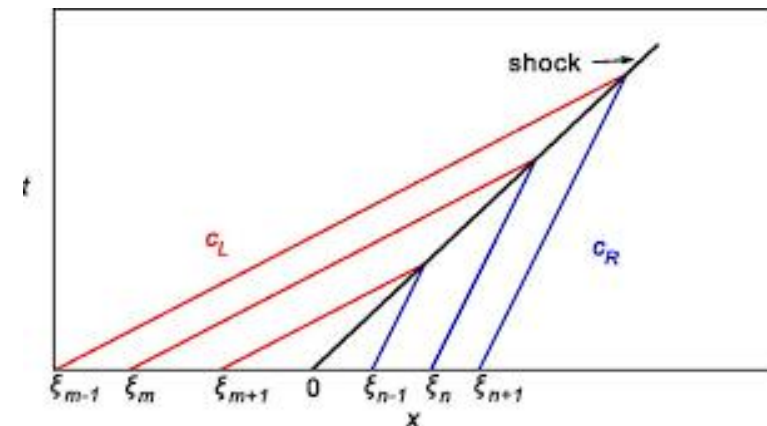
铁路警察，各管一段

- 双曲问题最终是没有“最终轨道”可供收敛的
 - $u(t; x) = u_0(x - vt)$ 意味着任意扰动都会保持
 - 传统算法优势很大，暂时不必采用代理模型
- 非线性双曲问题存在间断性
 - 神经网络表示定理仅对光滑函数成立，针对间断是非一致收敛的
 - 间断性是流体力学问题中不得不品尝的一部分
 - 激波：快的追慢的，但又不能（显著）超车
 - 接触间断面：压强和速度连续，温度和密度不连续



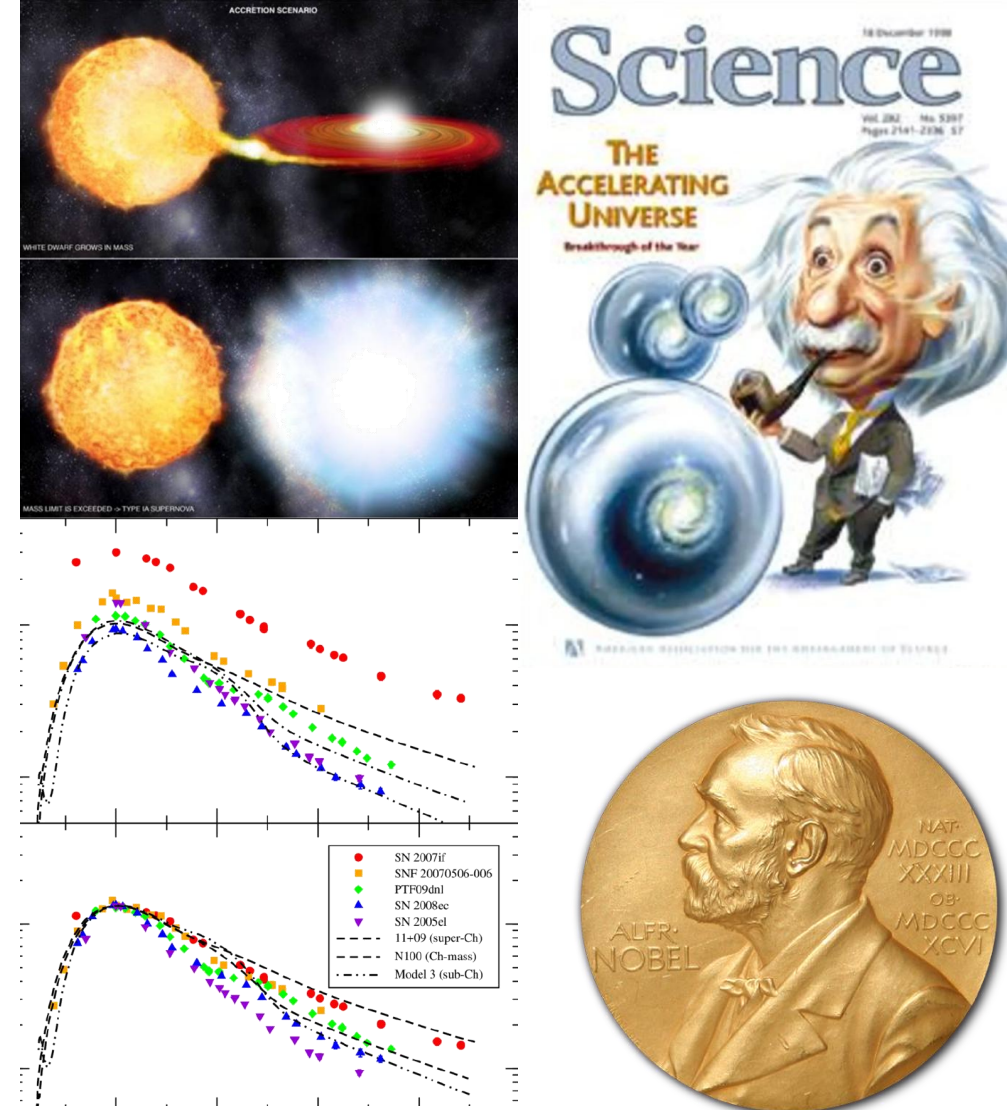
铁路警察，各管一段

- 非线性双曲问题所能存在的间断性
 - 切向间断性：容易理解（可能须寄生于接触间断性）
 - 激波：“气流撞墙”及其参考系变换
 - 接触间断性：压强和速度连续，温度和密度不连续
- 许多天体物理现象都植根于间断性
 - 超新星的点燃与爆发
 - 太阳风与周边星际介质
 - 吸积激波——从行星籽到整个星系
- 神经网络的表示定理具有局限性，暂勿揠苗助长



非线性双曲问题的间断性

- 2011 年诺贝尔物理学奖给了超新星宇宙学
- Ia 型超新星被认为是其中的关键
 - 白矮星超过钱德拉塞卡质量上限后，相对论性简并电子气导致流体静力学稳态消失
 - 简并物质的燃烧是不稳定的（同一原理：氦闪）
 - 看上去爆发条件一致，绝对光度理应一致
- 问题真的那么简单吗？
 - 结合其他观测手段，Ia 超新星的光度没那么一致
 - 人们还发现了反复的爆发……
 - 值得进行深入细致的研究——做数值模拟！



具体问题简介：天文学图像

Zhang et al. 2025
ApJ, 990, 105
arXiv: 2504.14180

- 控制方程：流体力学 + 核反应

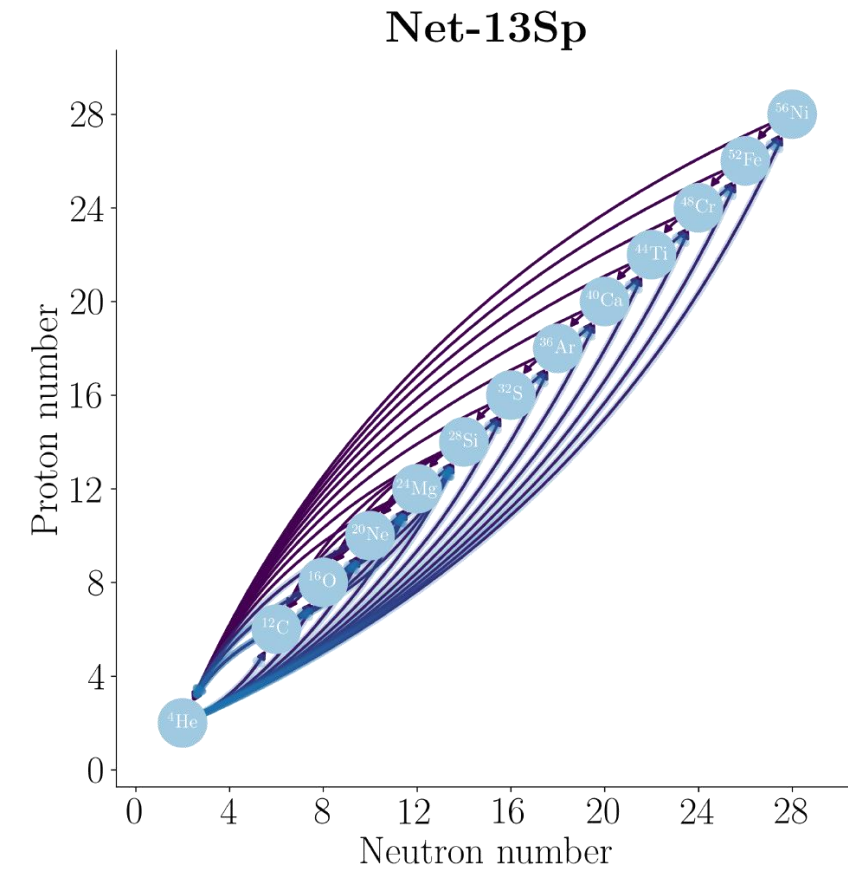
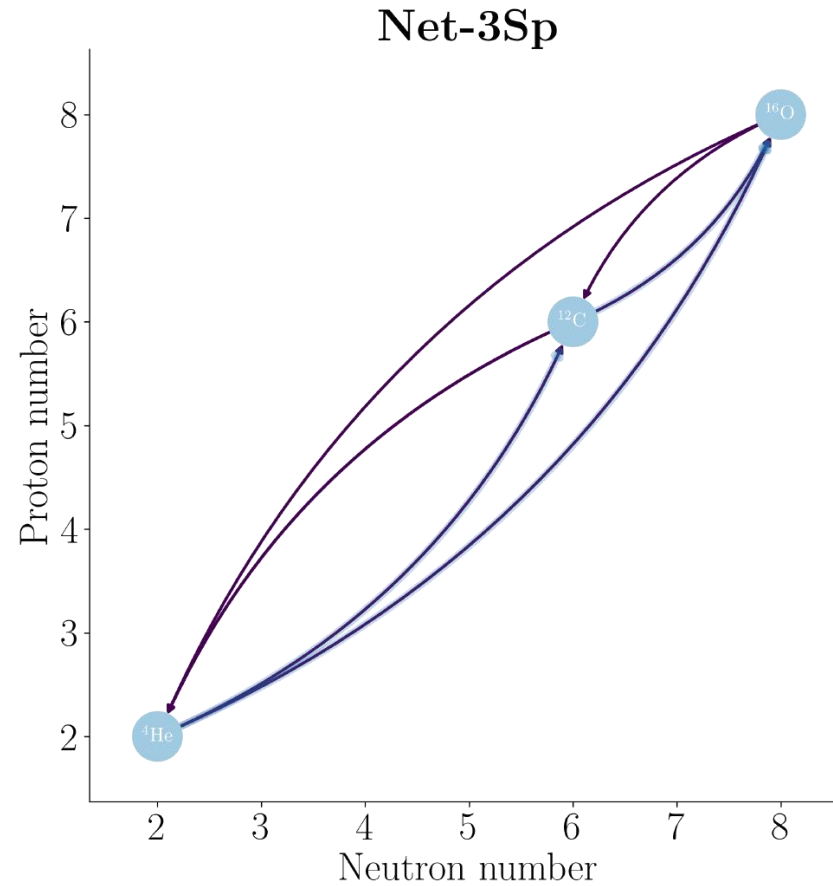
$$\frac{\partial(\rho X_k)}{\partial t} + \nabla \cdot (\rho X_k \mathbf{v}) = \rho \dot{\omega}_k,$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + p \mathbf{I}) = 0,$$

$$\frac{\partial(\rho \epsilon)}{\partial t} + \nabla \cdot [\mathbf{v}(\rho \epsilon + p)] = \rho \dot{S}.$$

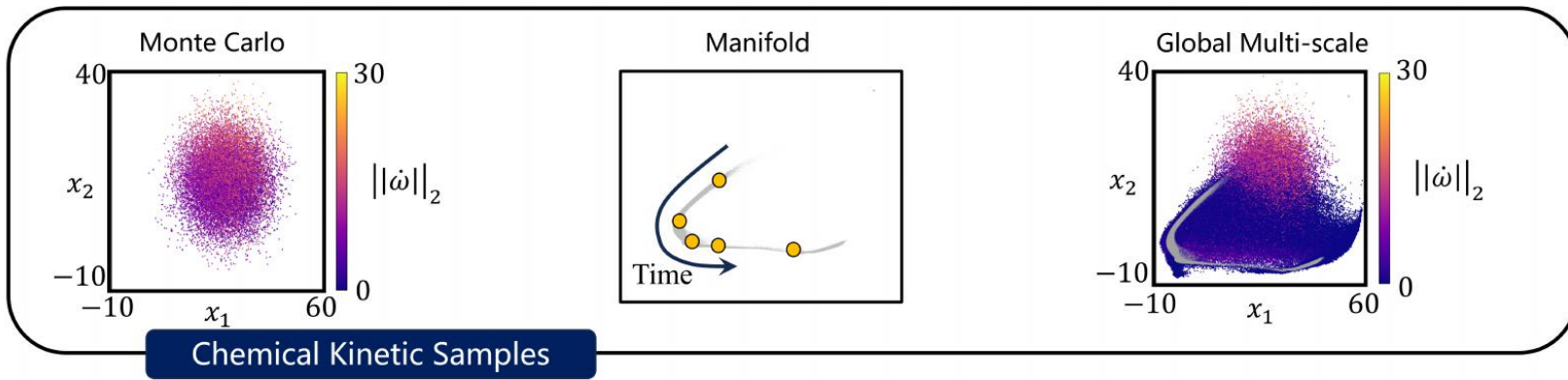
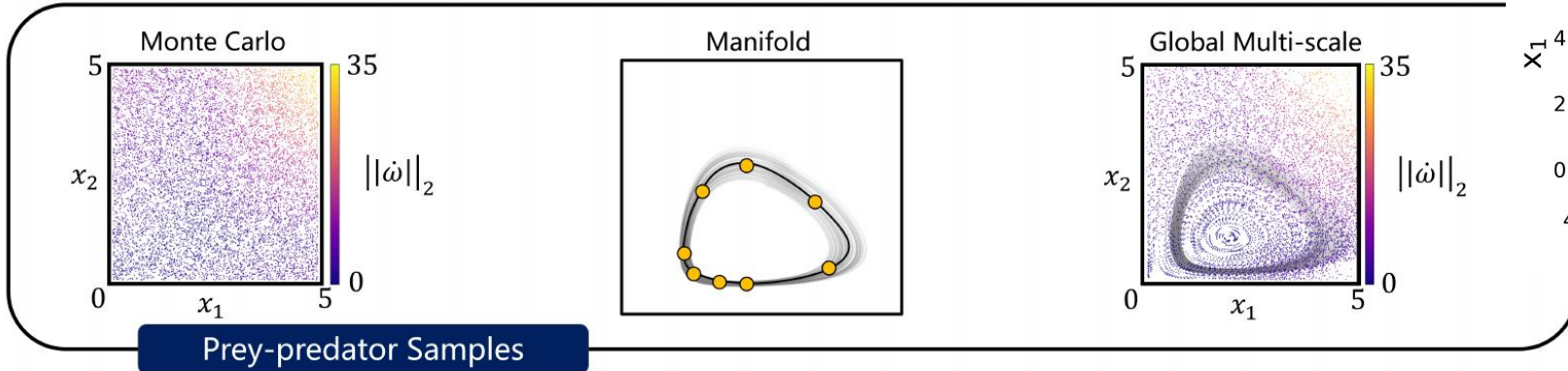
$$\begin{bmatrix} \mathbf{X}(t + \Delta t) \\ \epsilon(t + \Delta t) \end{bmatrix} := \mathcal{F}(T(\epsilon(t), \rho, \mathbf{X}(t)), \rho, \mathbf{X}(t))$$

$$= \begin{bmatrix} \mathbf{X}(t) + \int_t^{t+\Delta t} \dot{\omega} dt \\ \epsilon(t) + \int_t^{t+\Delta t} \dot{S} dt \end{bmatrix},$$

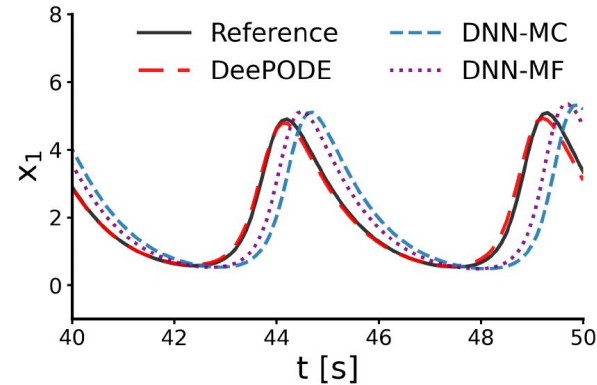


问题简介：控制方程

$$\frac{dx_1}{dt} = -x_1 + x_1 x_2, \quad \frac{dx_2}{dt}$$



$$\frac{dn^i}{dt} = A^i + B_j^i n^j + C_{jk}^i n^j n^k + D^i(n^j).$$



络，
太复杂
过高，
理采样

- 一个或许不错的策略：
 - 首先，遍历相对合理的初始状态空间，确定相空间轨道大致形态范围
 - 其次，蒙特卡洛采样，从随机初态出发，依据上一步的相轨道形态及范围确定是否保留
 - 最后，把上一步决定保留的数据点继续演化，直至到达合适的相空间覆盖率

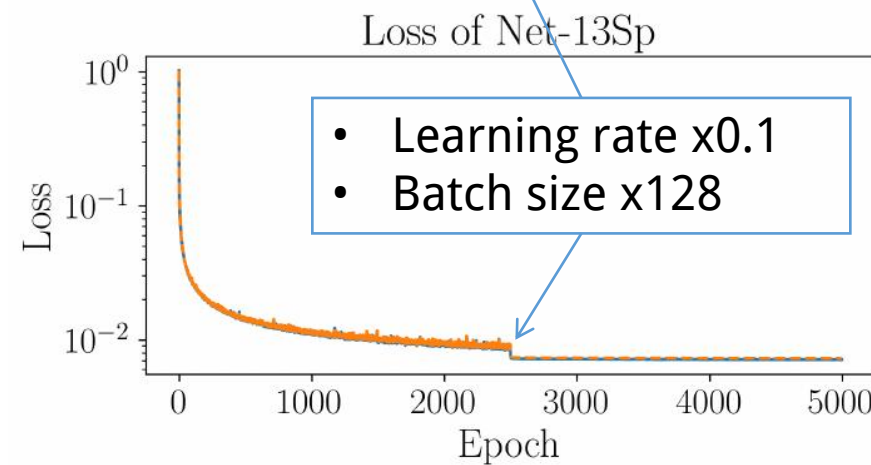
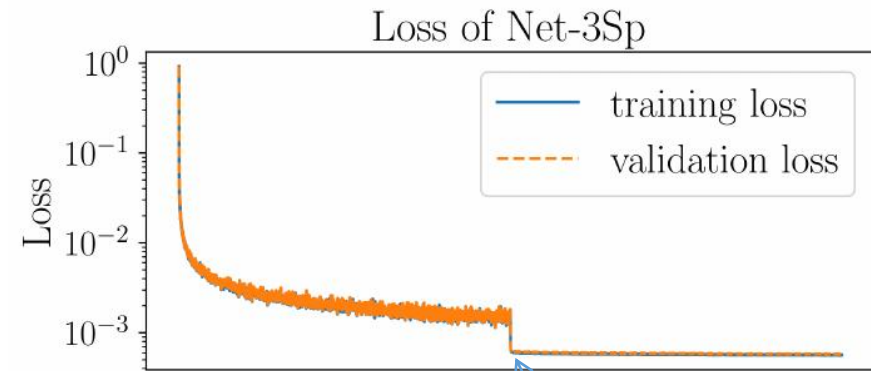
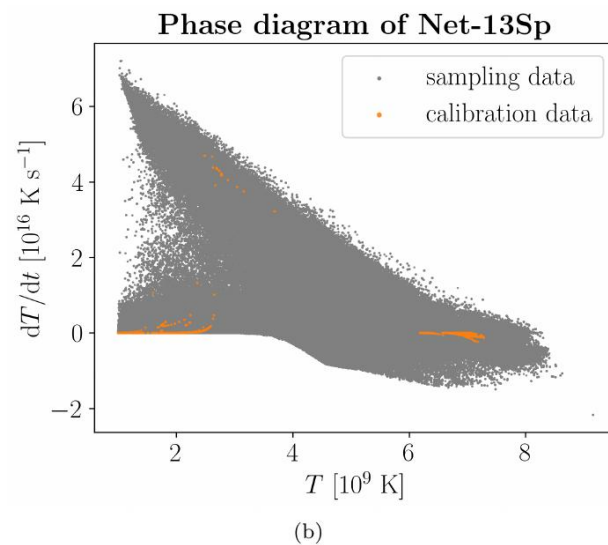
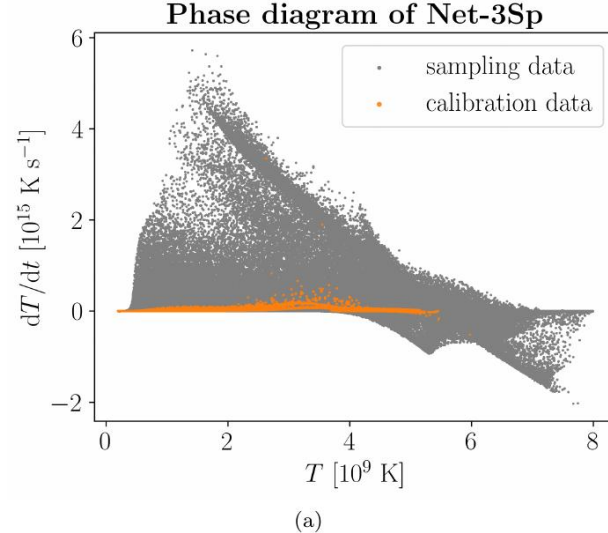
采样训练策略：基于相空间梯度的蒙卡采样

- 核反应的反应通道较为简单：
 - 通常没有化学反应的多向联通
 - 总体可概括为“自己烧自己”
- 损失函数和收敛评估：

$$L_{\text{data}} = \frac{1}{N} \sum_{i=1}^N \| \mathbf{u}_i - \tilde{\mathbf{u}}_i \|_{L_1}$$

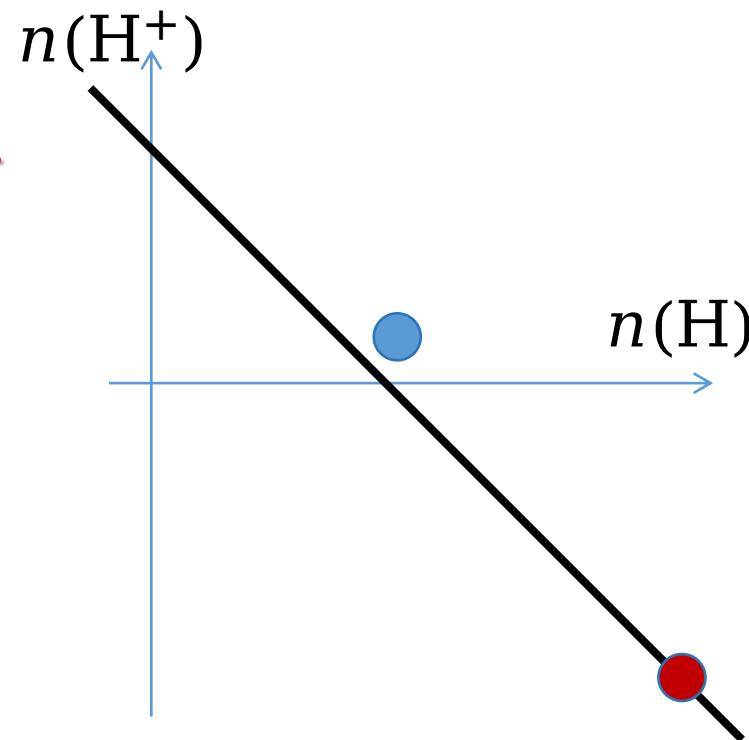
$$L_{\text{sum}} = \frac{1}{N} \sum_{i=1}^N \left\| \sum_k^{N_s} \tilde{X}_k^i - 1 \right\|_{L_1}$$

单纯的损失函数并不足够，
 还需额外约束（质能守恒）



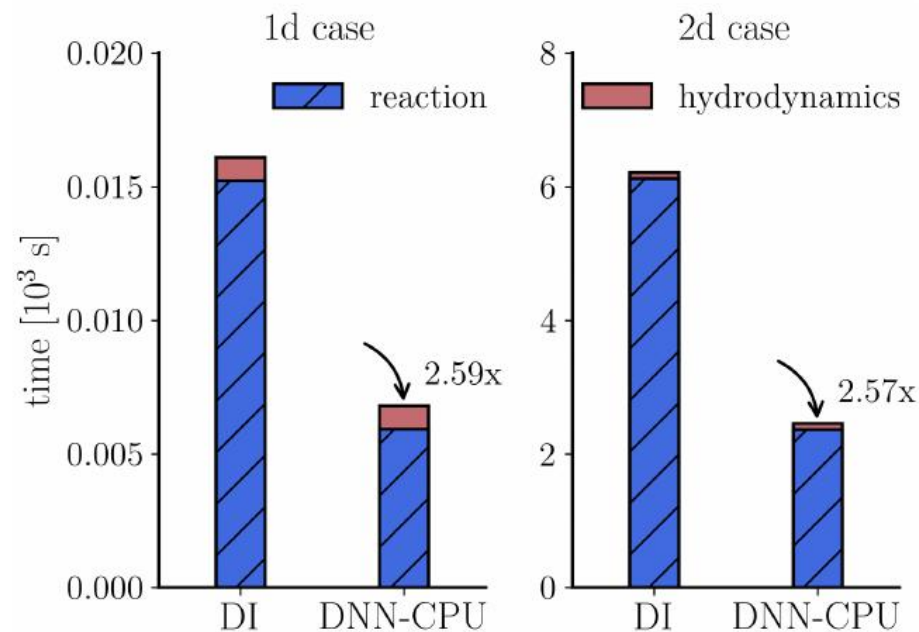
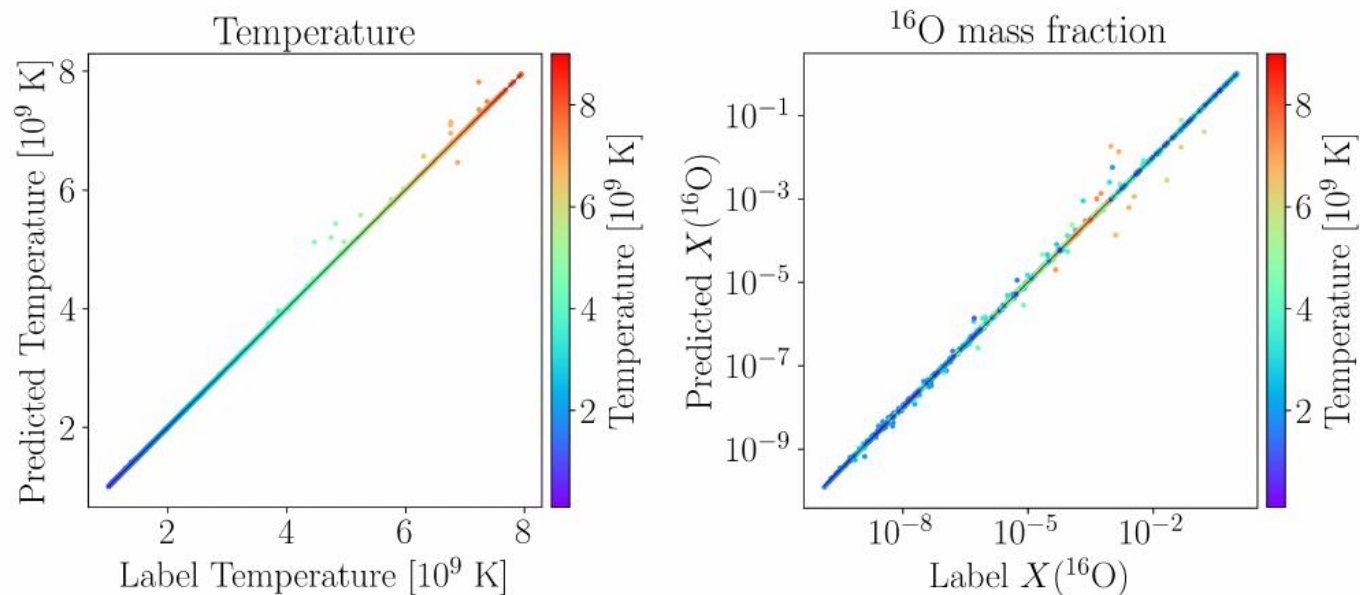
柿子要捡软的捏（以及谁是软柿子）？

- 一个大麻烦：高阶精度算法通常破坏元素守恒，AI 代理模型亦然
 - 例子： $H^+ + e^- \rightarrow H$ ，假定反应充分进行
 - 准确但不守恒： $H^+ + e^- \rightarrow 0.999 H$
 - 守恒但不准确： $H^+ + e^- \rightarrow 2 H + (-1)H^+ + (-1)e^-$
- 柿子捡软的捏：化学比核反应更麻烦
 - 化学反应必须满足高维的元素守恒（化学计量空间约束）
核反应只须处理质能守恒
 - 化学反应发生的条件及结果复杂，通常构成网络
核反应常为“燃烧核燃料”，初末条件相对清晰



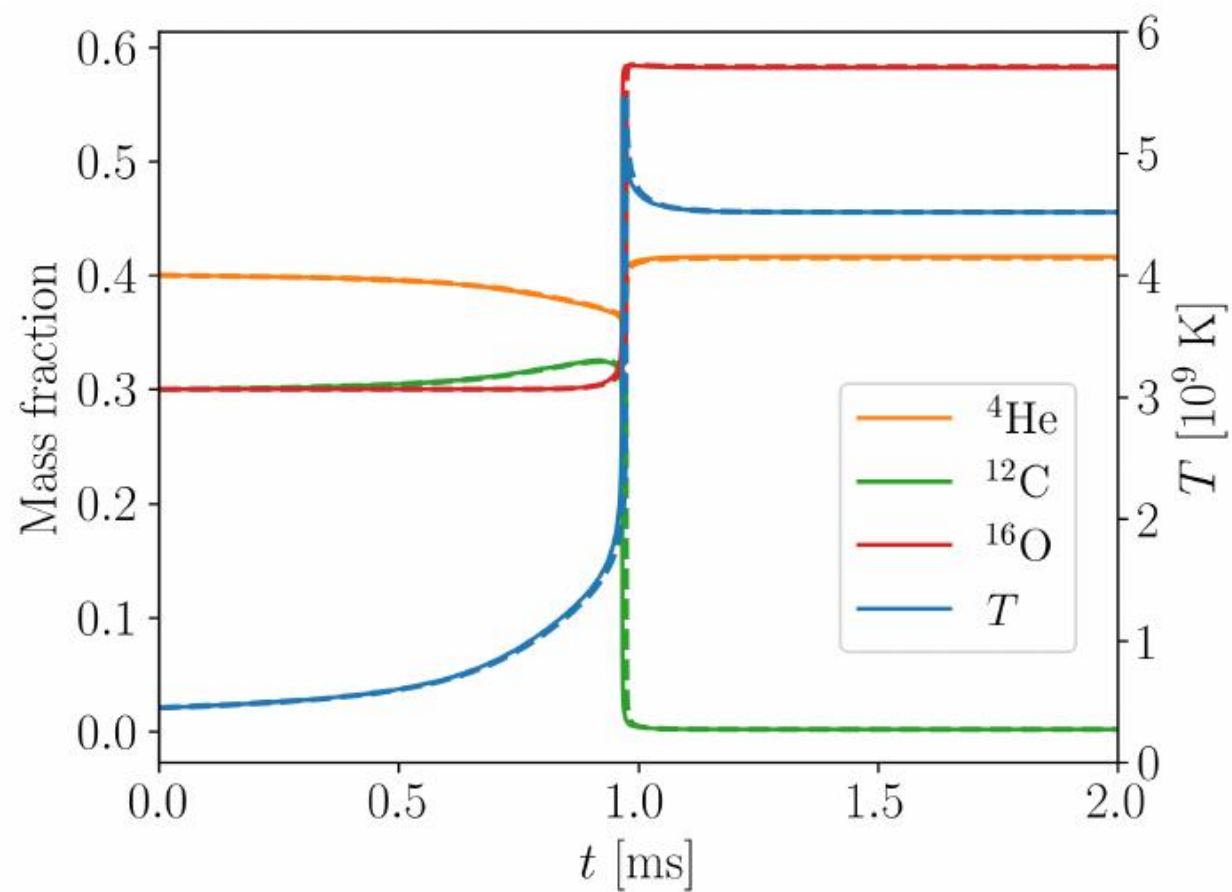
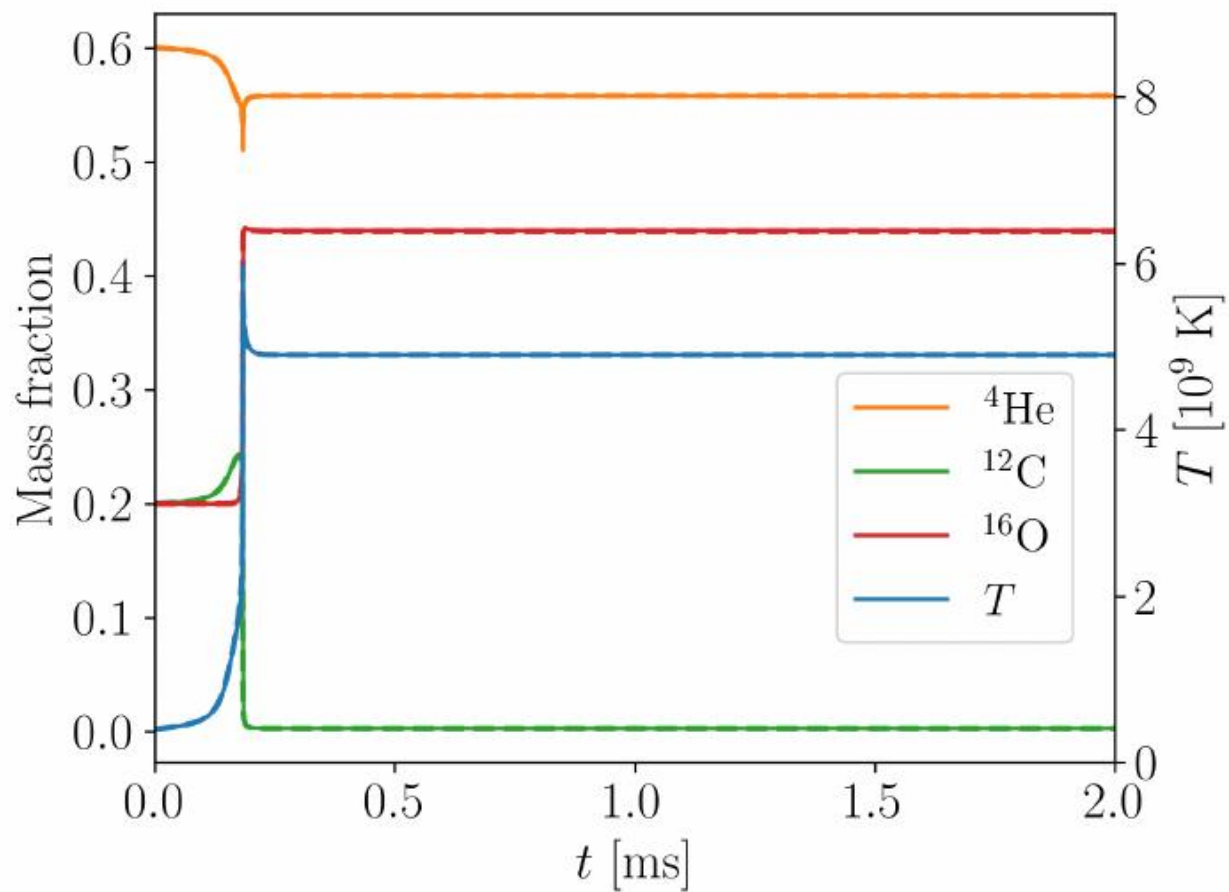
柿子要捡软的捏（以及谁是软柿子）？

- 占了便宜，又快又准
- 问题：久期演化仍然糟糕（采样时难以解决久期演化）
- 既是段子，也是智慧：铁路警察，各管一段



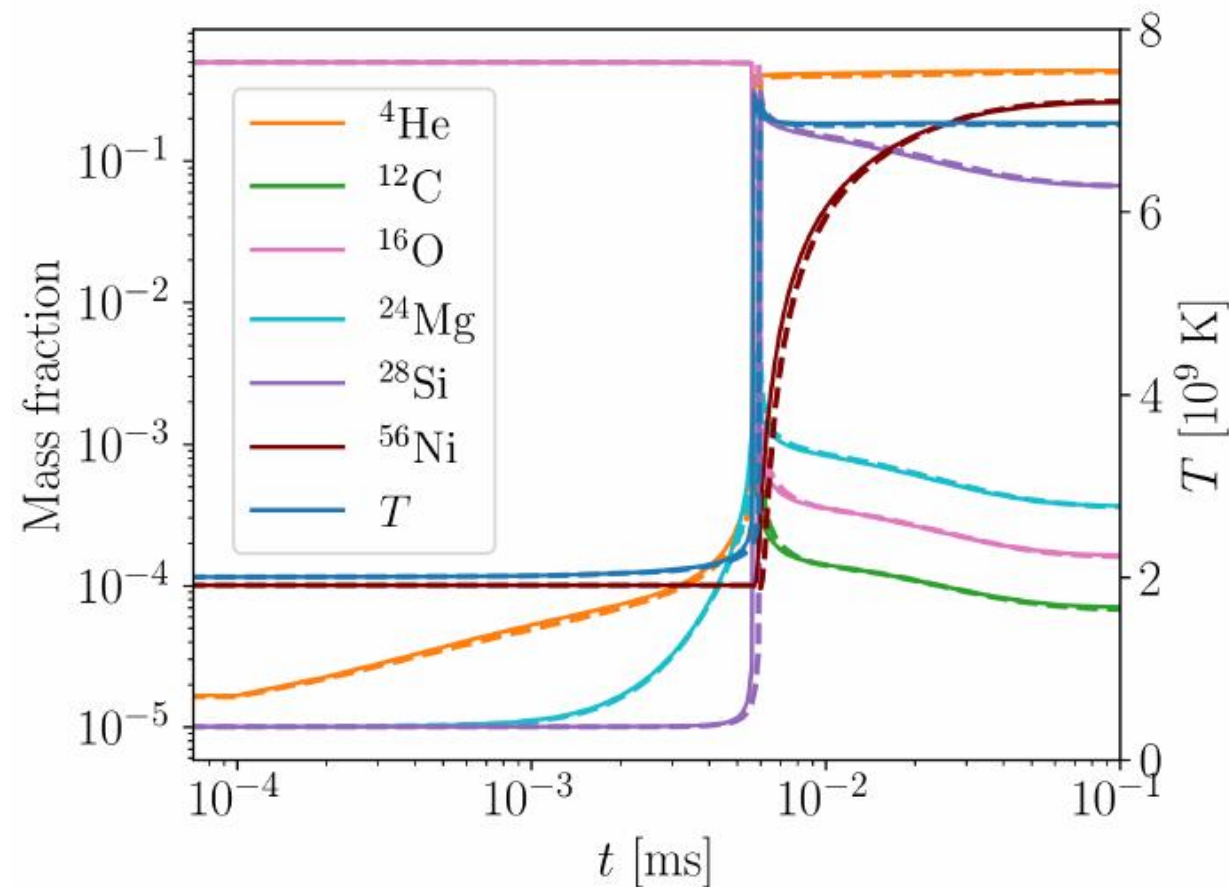
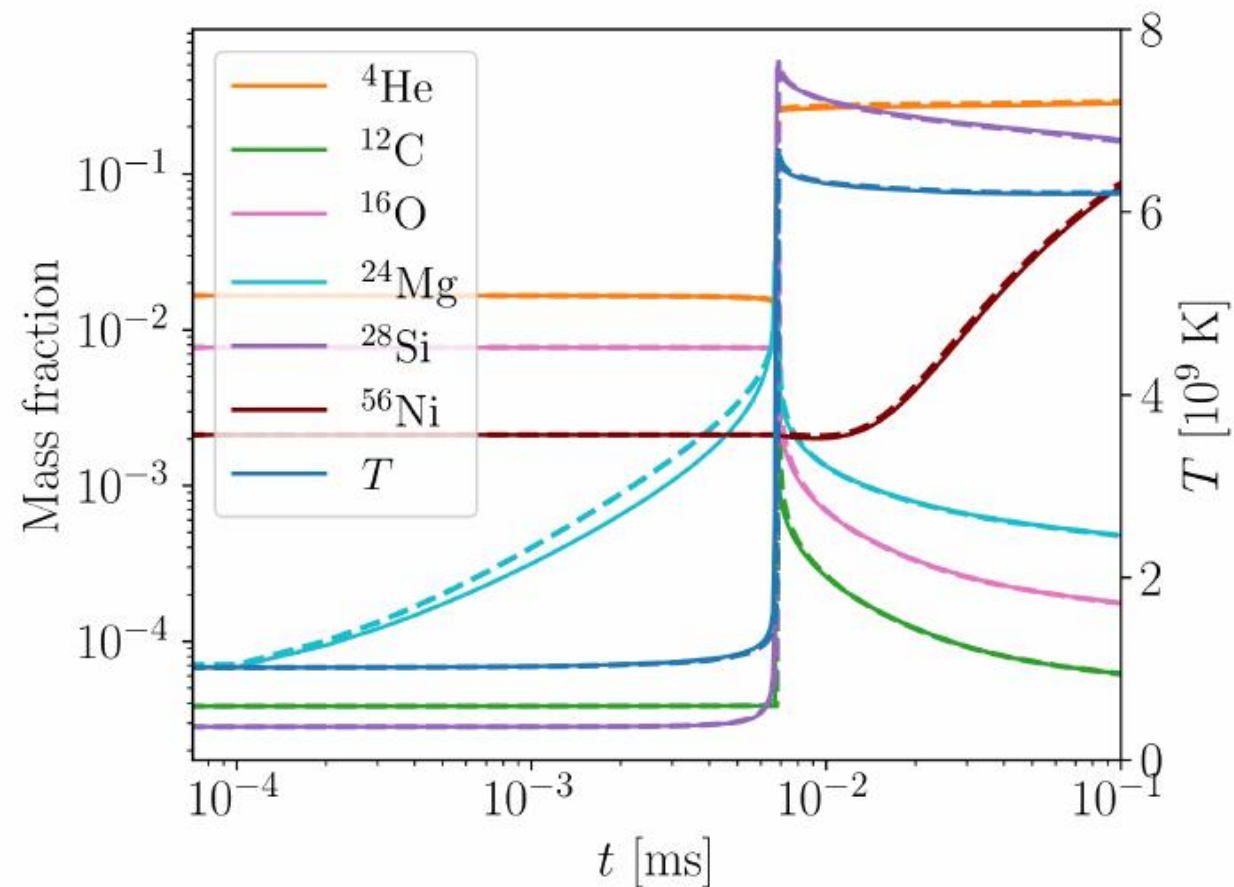
效果如何？

- 损失函数中的残差的量级基本与实践中的误差匹配



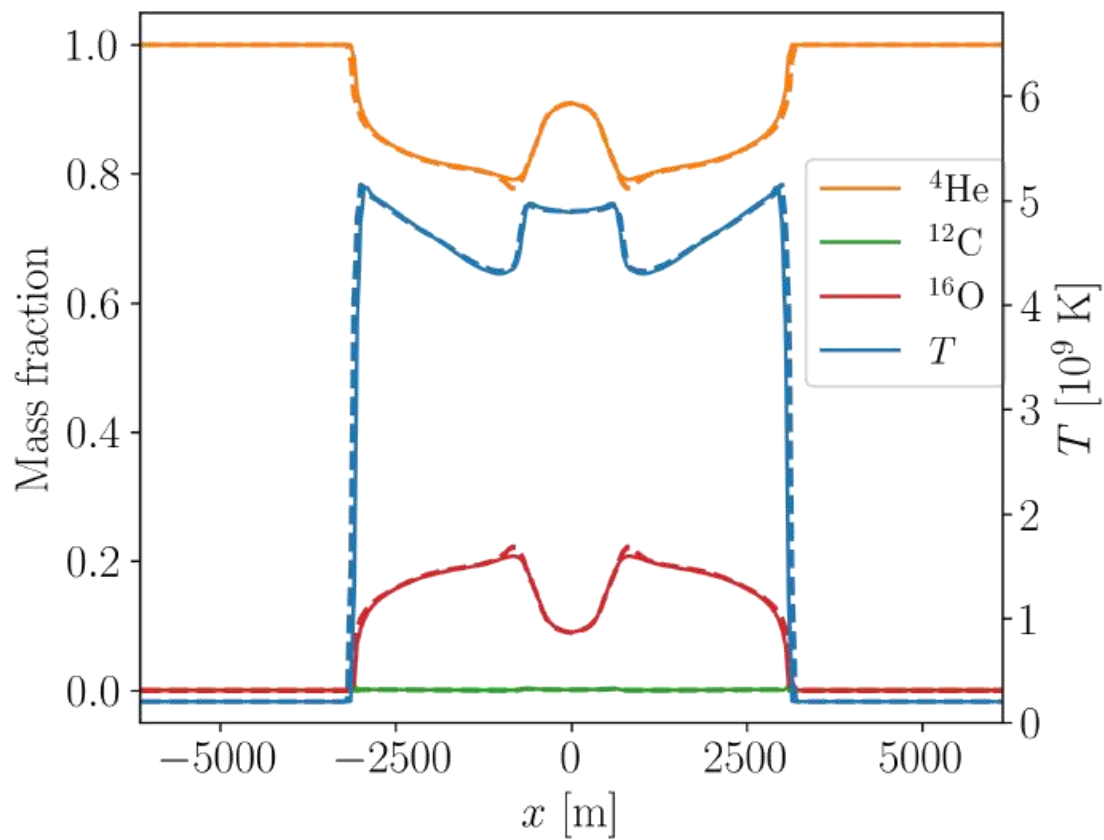
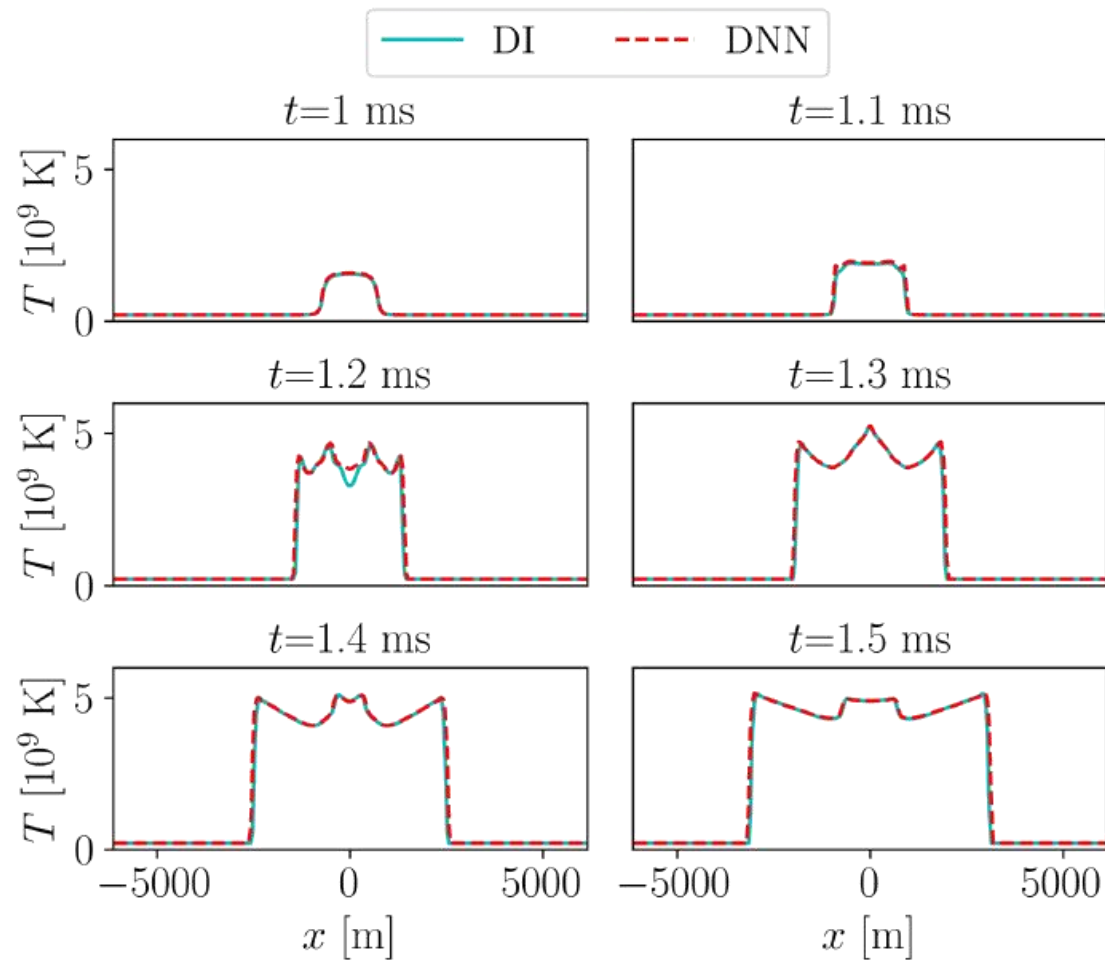
效果如何（三组分，单点）？

- 损失函数中的残差的量级基本与实践中的误差匹配



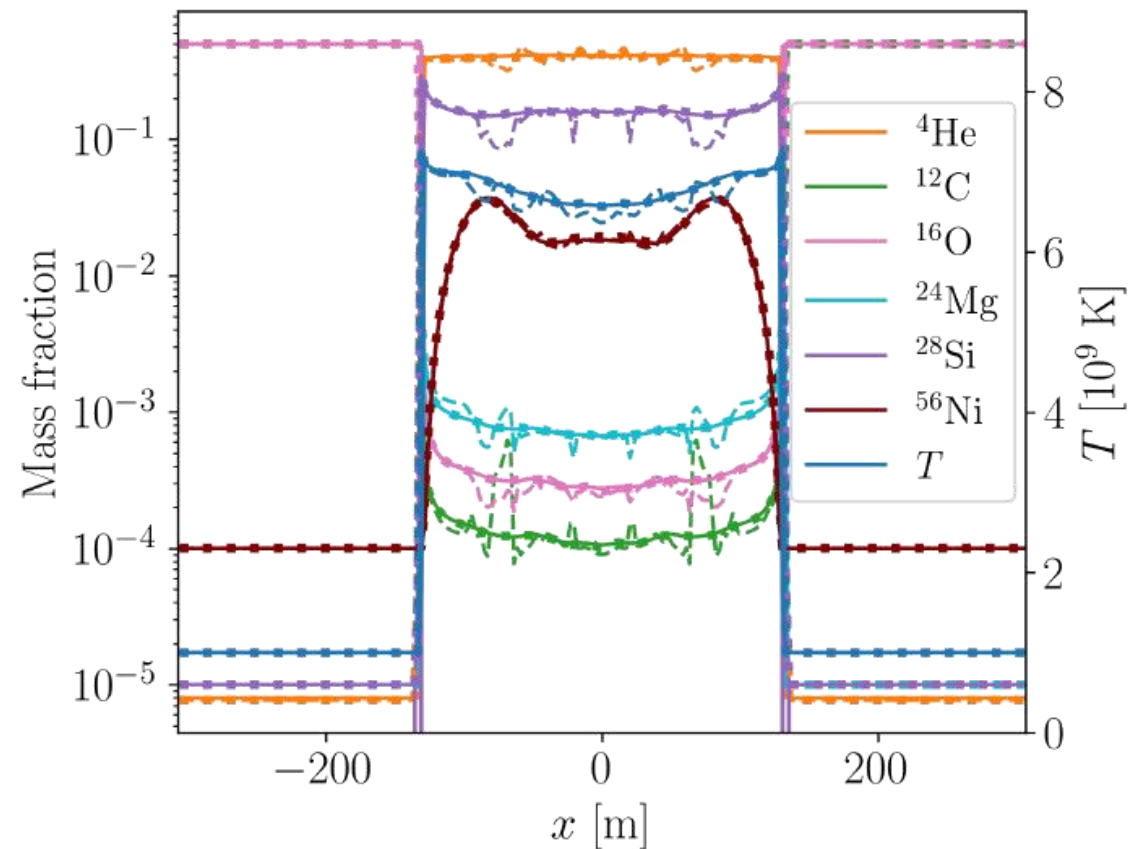
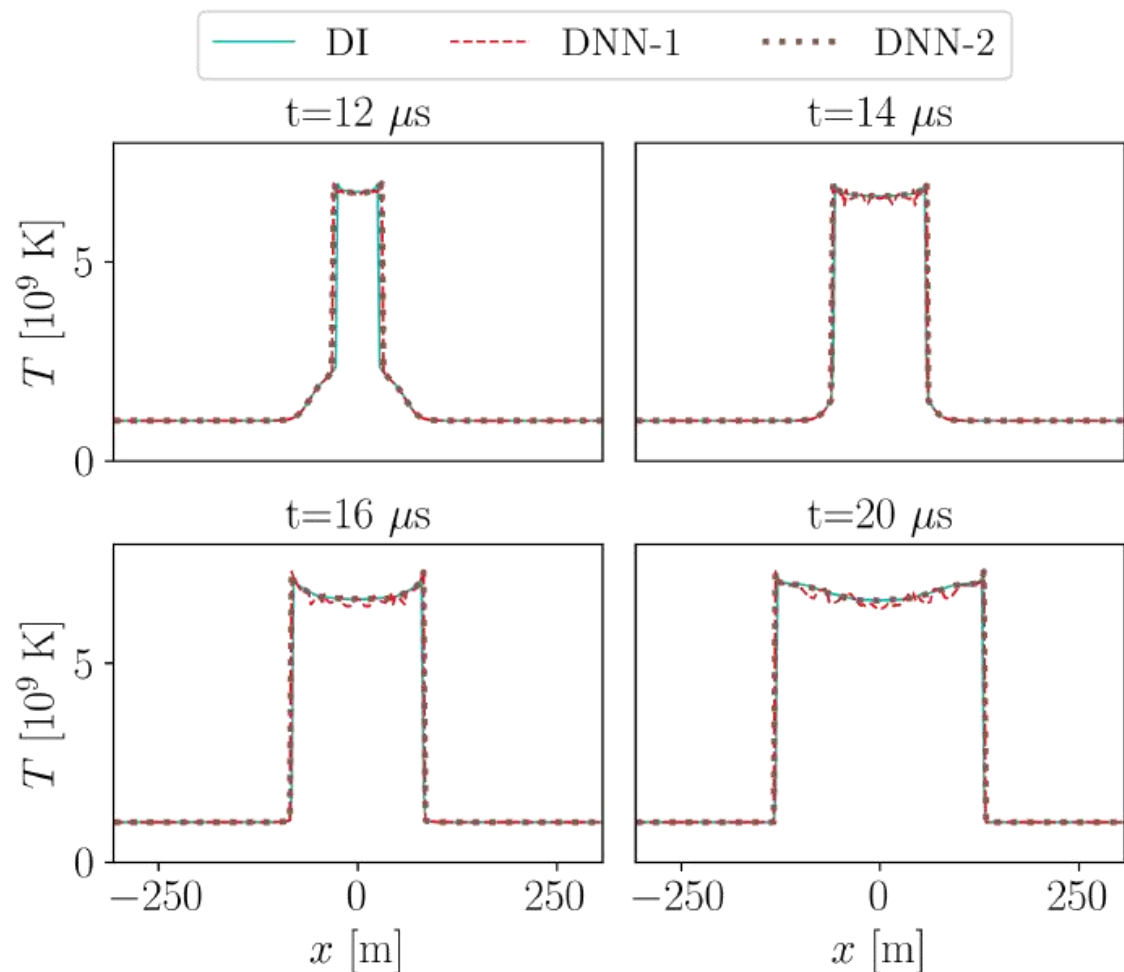
效果如何（13 组分，单点）？

- 因为某种程度上的负反馈机制，接入流体力学后性能似乎继续提升



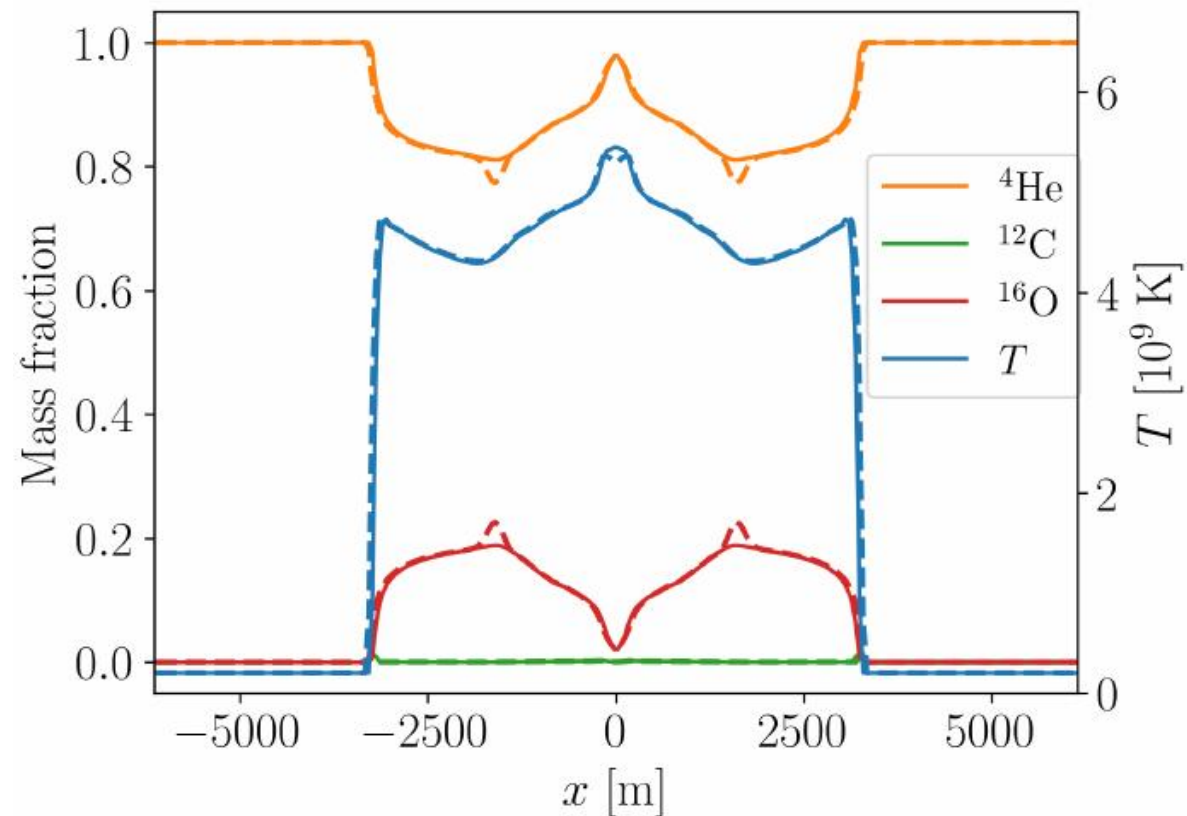
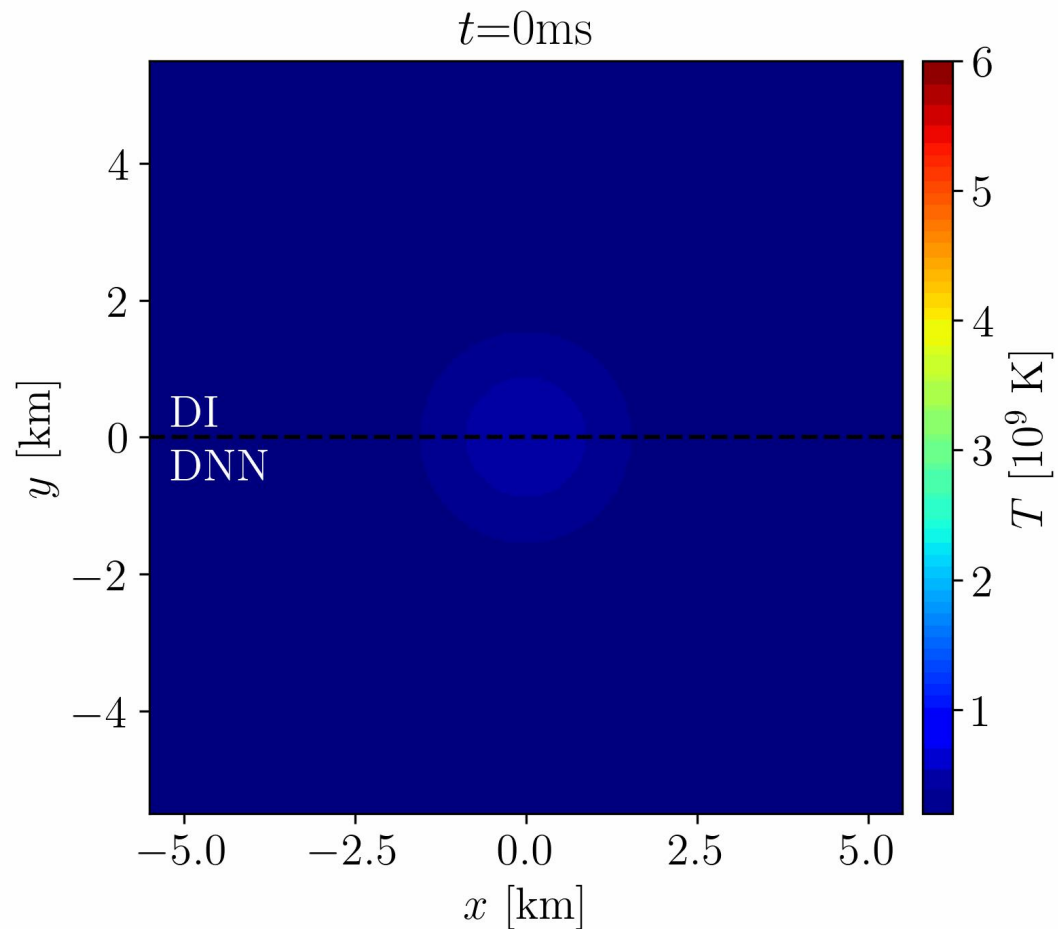
效果如何（3组分，一维）？

- DNN-2: 别再对灰烬继续使用神经网络了 ($T > 6.5 \times 10^9$ K)



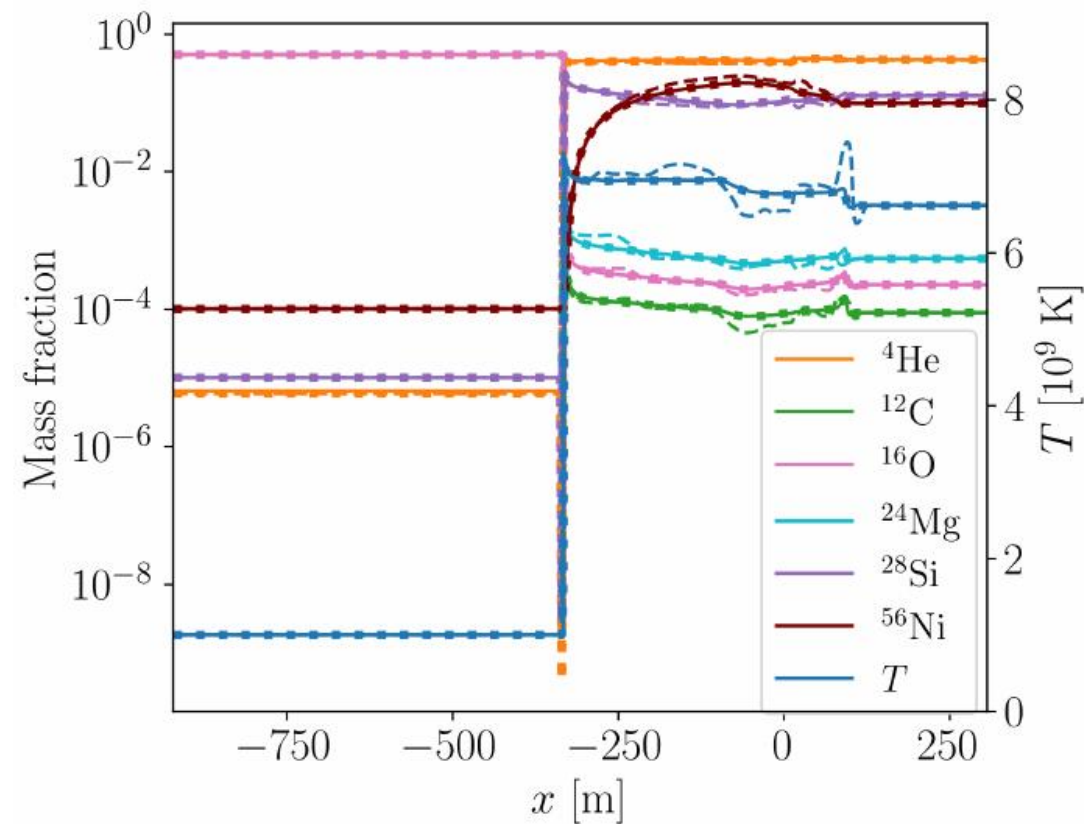
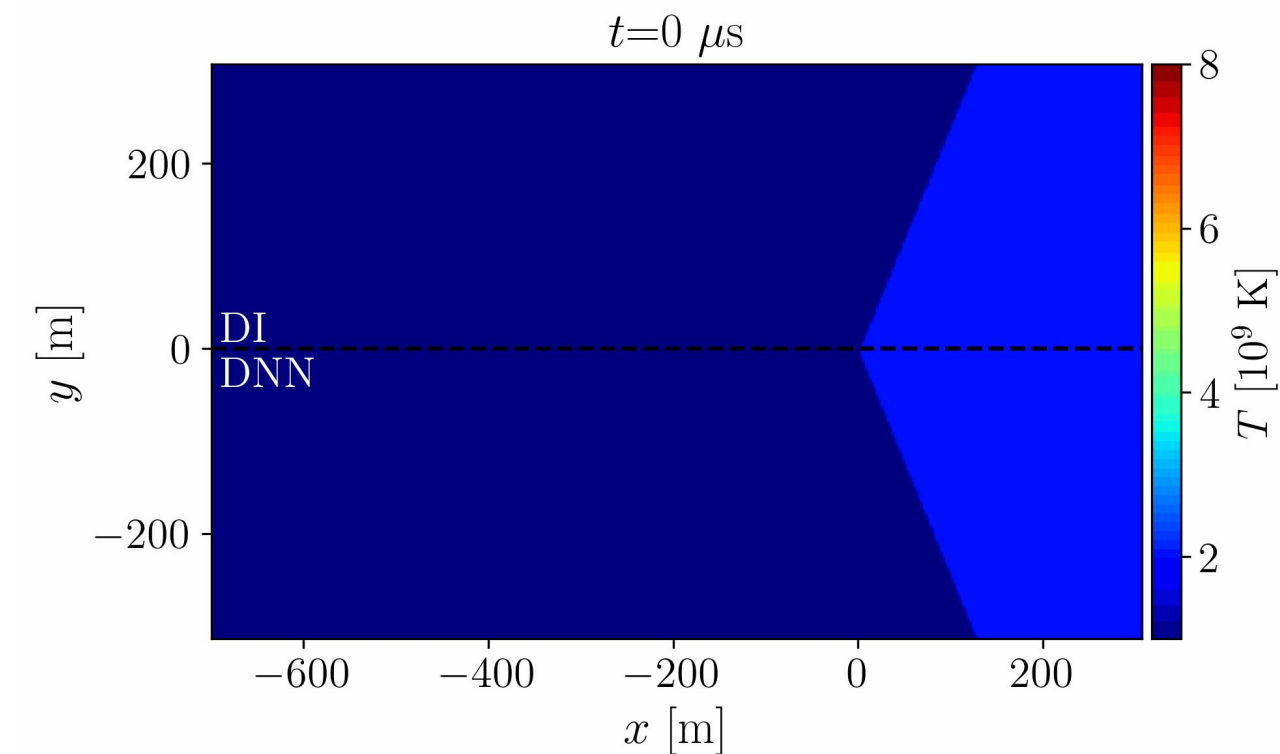
效果如何 (13 组分, 一维) ?

- 超声速爆轰，右侧曲线： $t = 1.5 \text{ ms}$



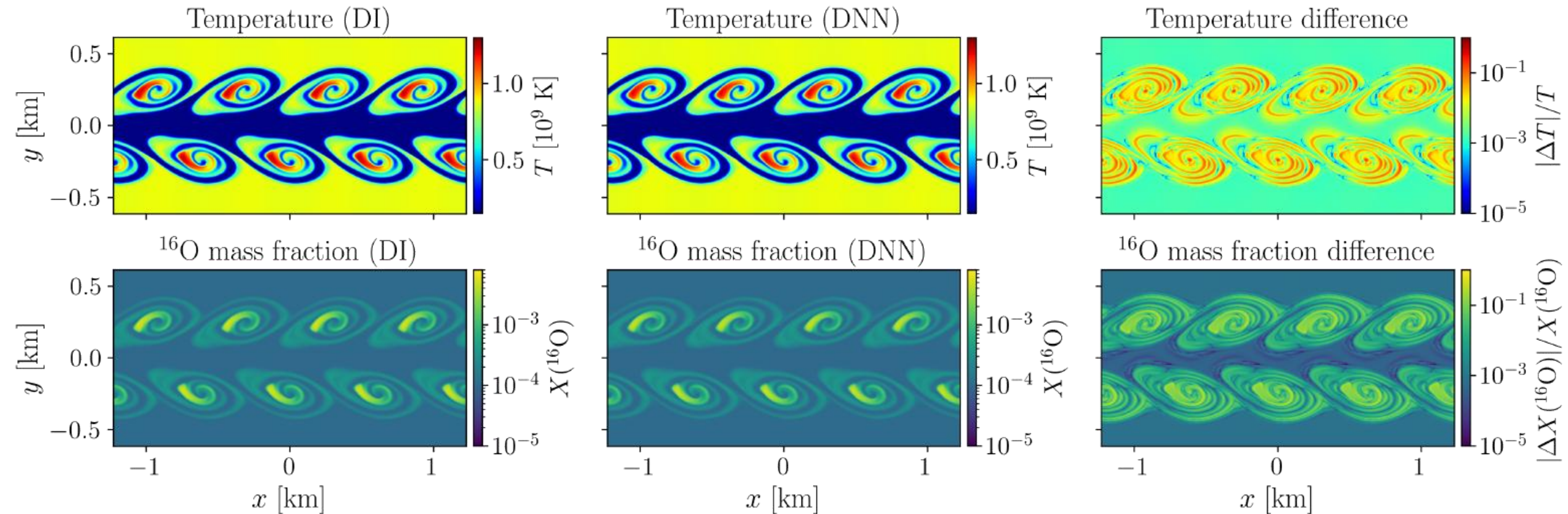
效果如何（3组分，二维）？

- 超声速爆轰，右侧曲线： $t = 40 \mu\text{s}$

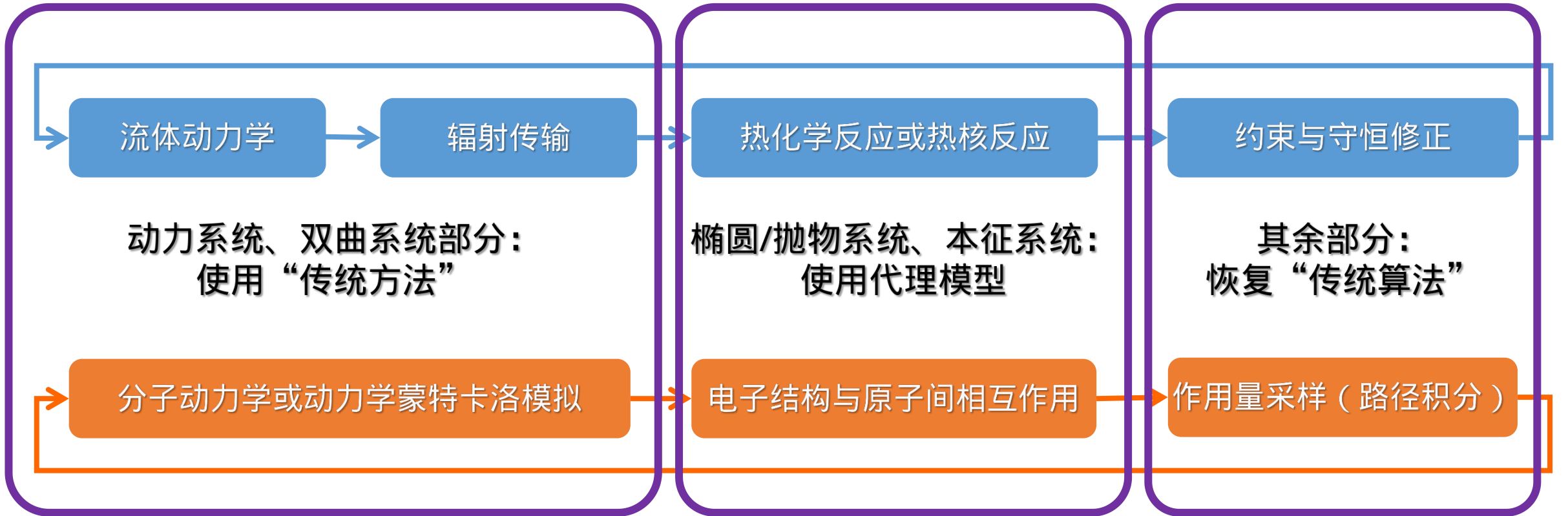


效果如何（13 组分，二维）？

- 亚声速（开尔文——亥姆霍茨不稳定性），13 组分 ($t = 6.5 \text{ ms}$)



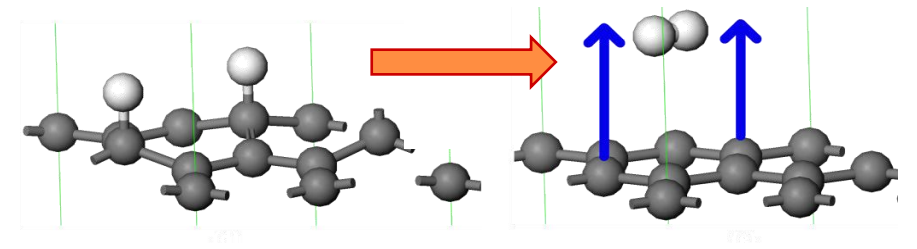
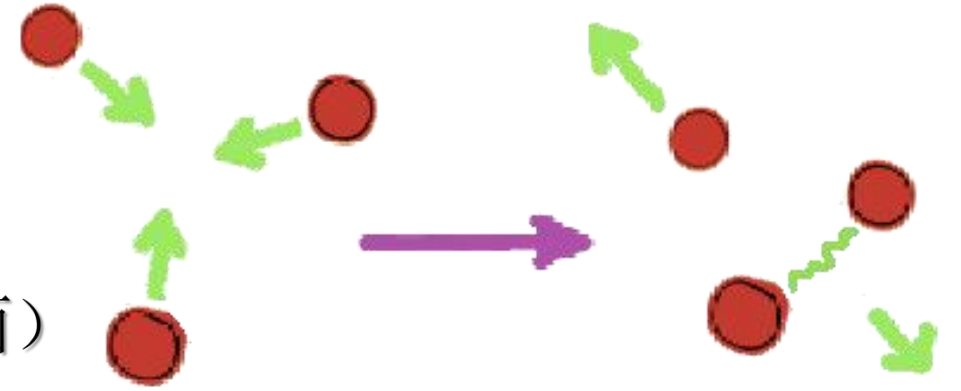
效果如何（13 组分，二维）？



这一逻辑的推广——何处应该采用 AI 代理模型？

- 最低多极矩缺失：难以实现动量与能量的高效耗散，几乎都是弹性碰撞
- 无辅助的效率系数： $\sim 10^{-29} \text{ cm}^3 \text{ s}^{-1}$
- 最常见的 H_2 形成机制：借助他物带走能量
 1. 三体反应（需要 $n(\text{H}) > 10^6 \text{ cm}^{-3}$ ）
 2. H^- 机制（通过“排挤” e^- ；最常见：太阳表面）
 3. 尘埃催化机制

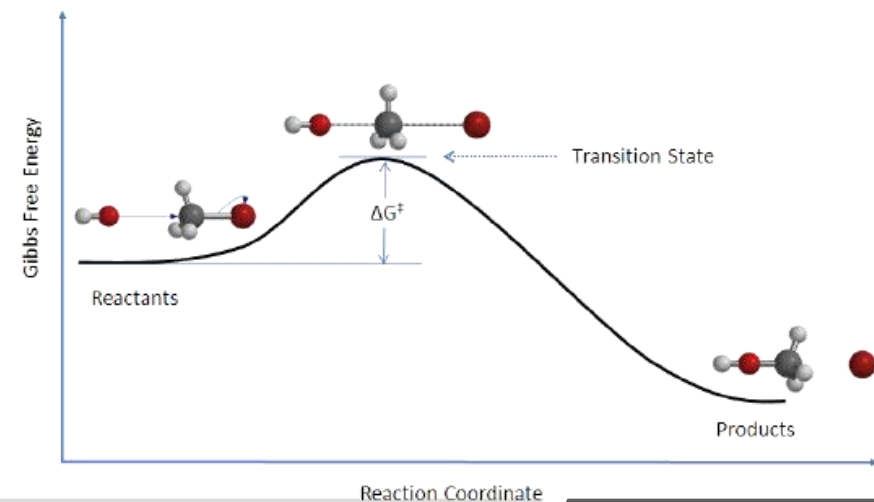
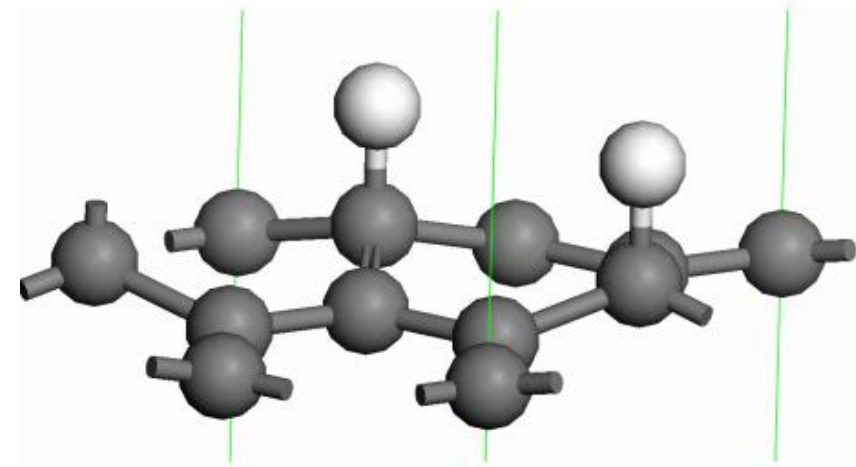
通过晶格振动（声子）等内部自由度进行耗散
- 分子态星际气体中，1 与 3 更为重要
 - 机制 1 需要湍流；机制 2 需要电离
 - 机制 3 在英文里叫做“Godilocks soup”，不能太冷、不能太热



同一逻辑的天体化学运用——氢分子合成

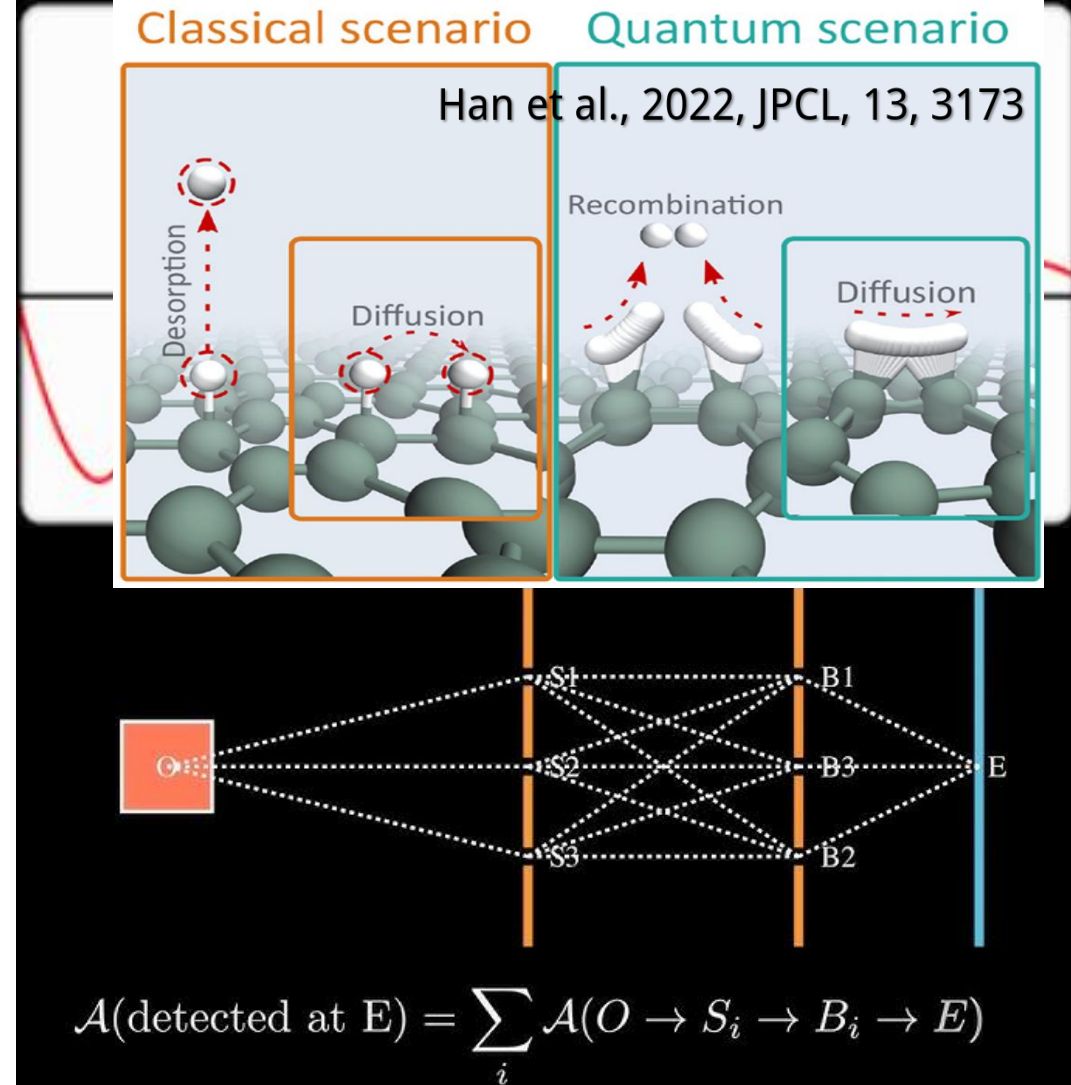


- 先来看看键长
 - 石墨烯中的 C-C 键：0.142 nm
 - 氢分子中的 H-H 键：0.074 nm
- 典型键能：似乎是个很好的放能反应
 - H-H 键能：~4.5 eV
 - 石墨烯表面的 C-H 键能：> 0.5 eV
- 麻烦来了： $\exp(-0.5 \text{ eV}/k_B 20 \text{ K}) \sim 10^{-108}$
- 人们通常认为需要物理吸附（不成键），然而……
 - 不能太热，否则未及形成分子即脱离吸附
 - 不能太冷，否则被玻尔兹曼因子“压死”



天文环境的氢分子合成依赖尘埃……吗？

- 穿深估计，给人以很大希望： $P \sim \exp(-2l/\lambda)$
其中 $\lambda \sim h(2m_p \Delta E_A)^{-1/2} \sim 0.04 \text{ nm}$
- 然而，当前的“量子化学计算”
事实上通常采取玻恩——奥本海默近似
 - 电子结构：薛定谔方程或 Kohn-Sham 方程
 - 离子/原子核（此处的研究核心）仍是经典的
- “救命啊，费曼先生！”
 - 经典力学：最小作用量原理
 - 量子力学：作用量是相位，最终是诸路径之和



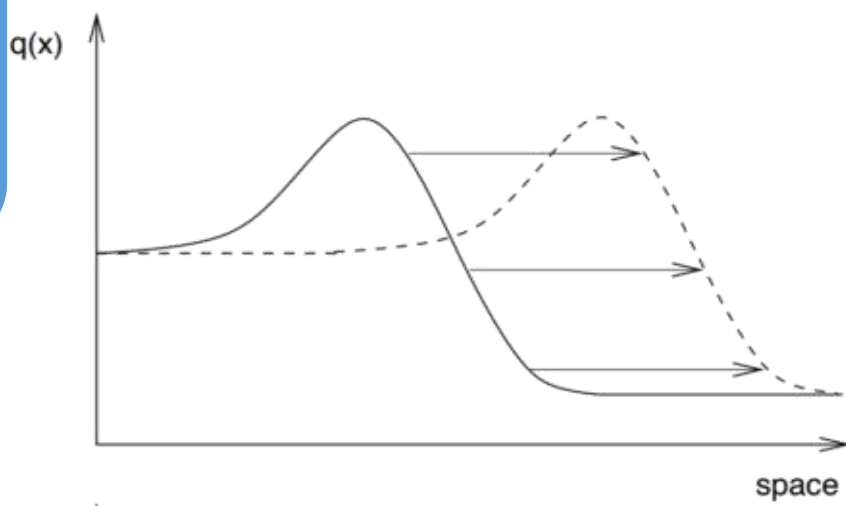
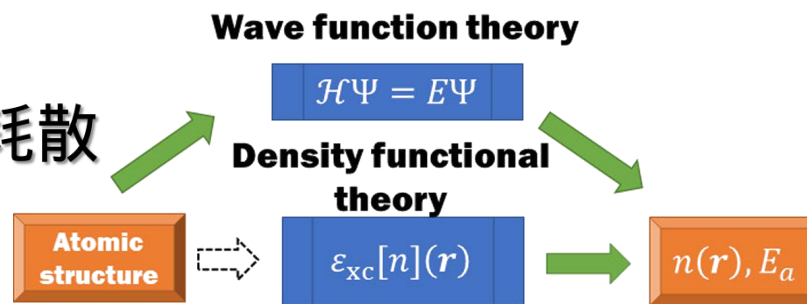
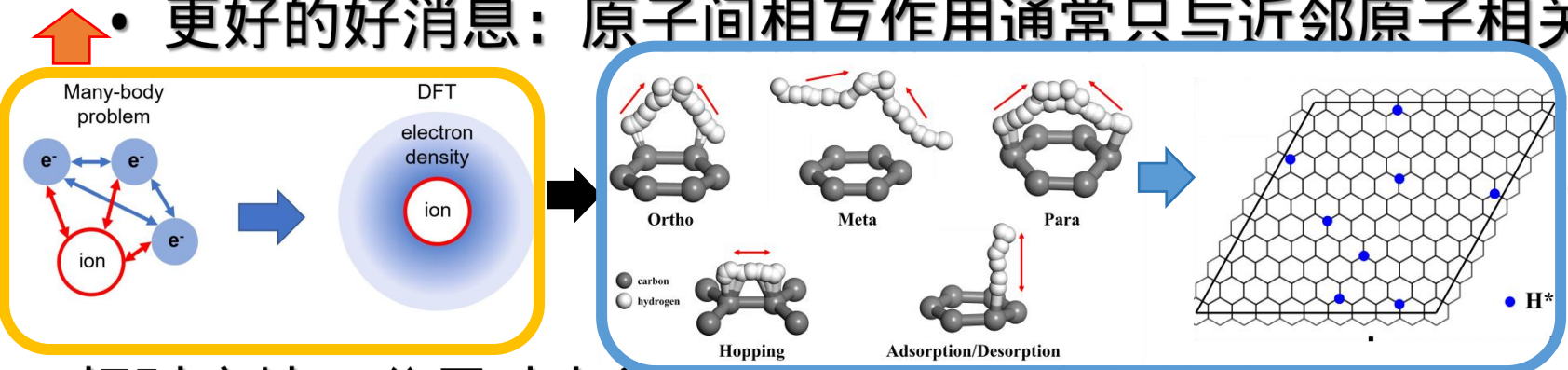
如何定量研究隧穿？

• 相同的段子（智慧）：**铁路警察，各管一段，分而治之**
Railway securitymen only serve their own sectors

• 电子结构求解（即便使用密度泛函方法）是本征系统求解

• 边界条件和相互作用项（离子势）**状态函数**，误差终将耗散

• 更好的好消息：原子间相互作用通常只与近邻原子相关

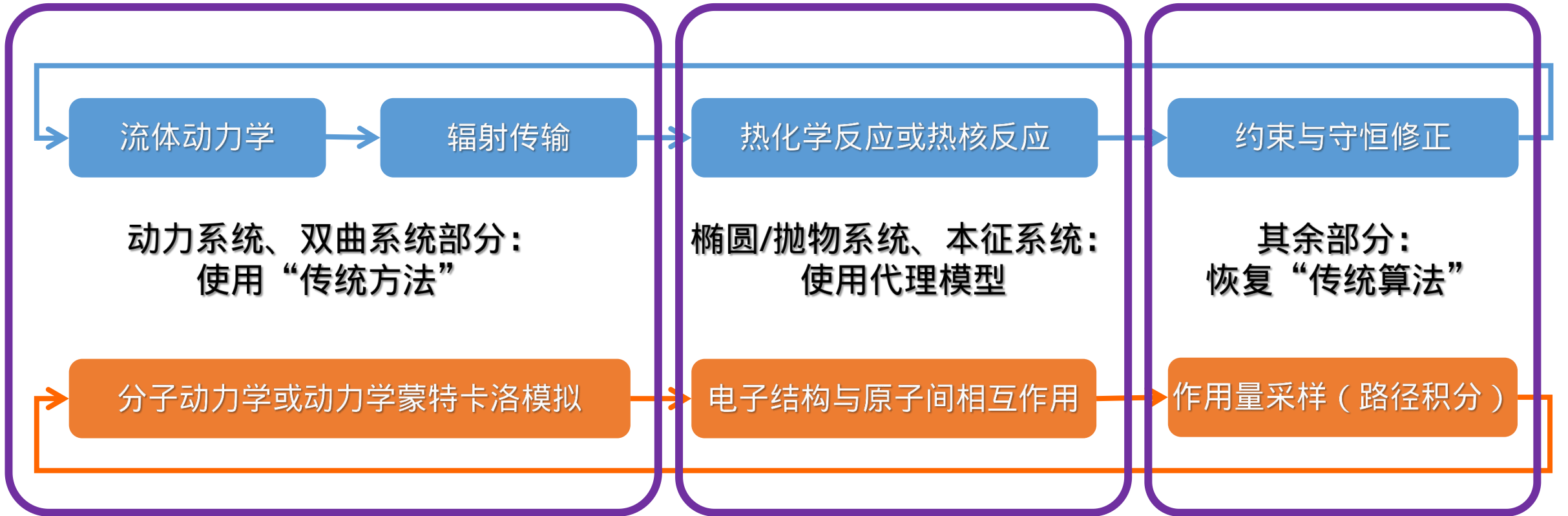


• 相对应地，分子动力学系统须以守恒律为基础

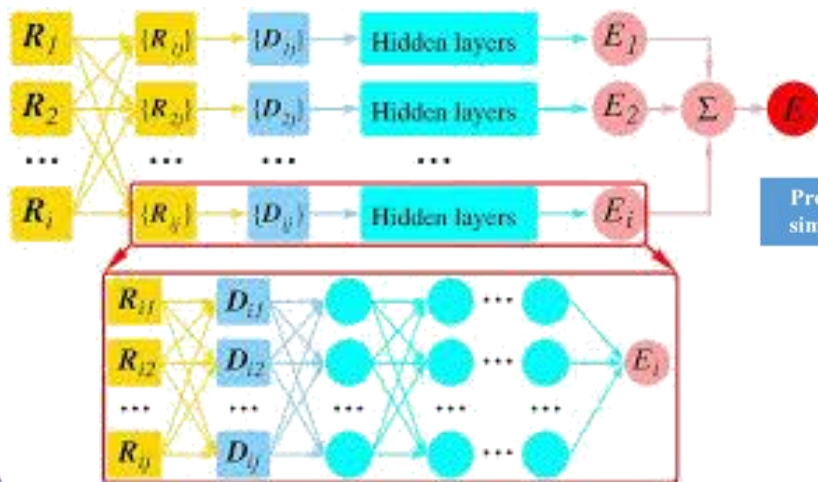
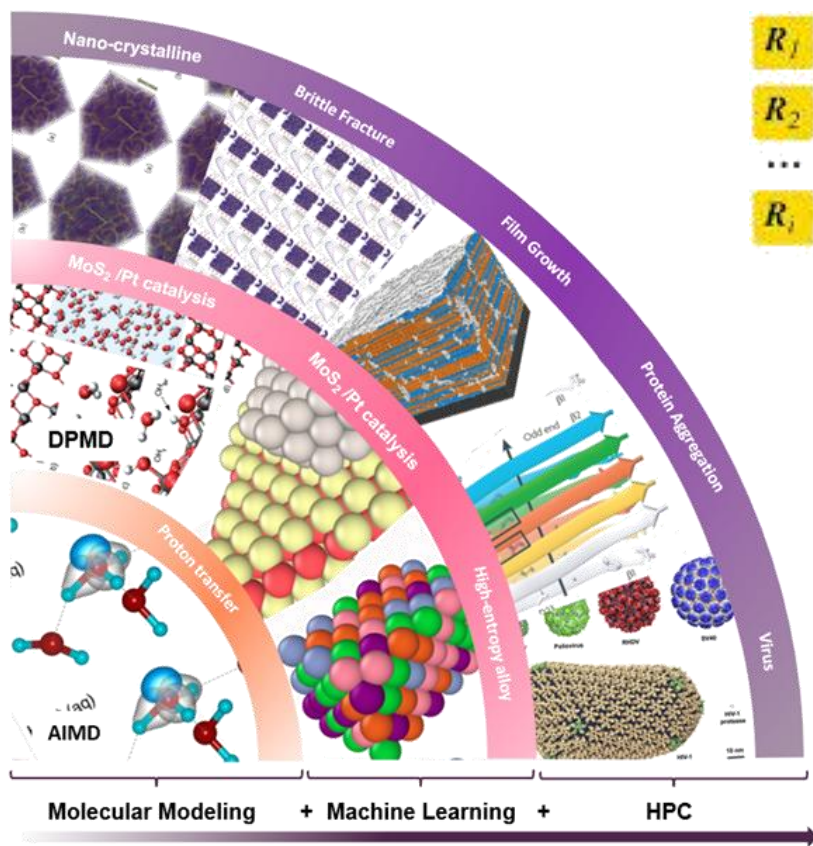
• 通常是**路径函数**

• 代理模型的误差将无可避免地累积、发散……

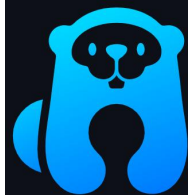
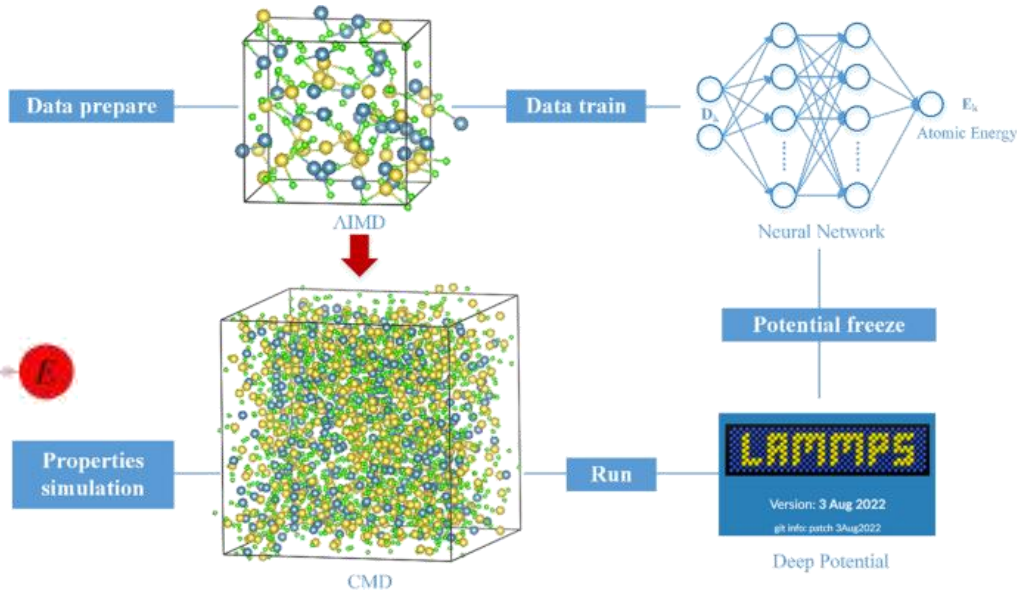
问题：哪一部分该用上代理模型？



不同的问题，相同的逻辑——何处采用 AI 代理模型？



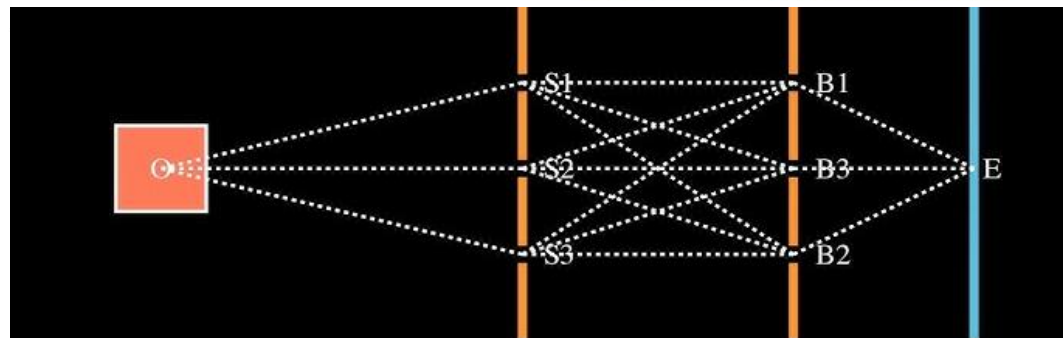
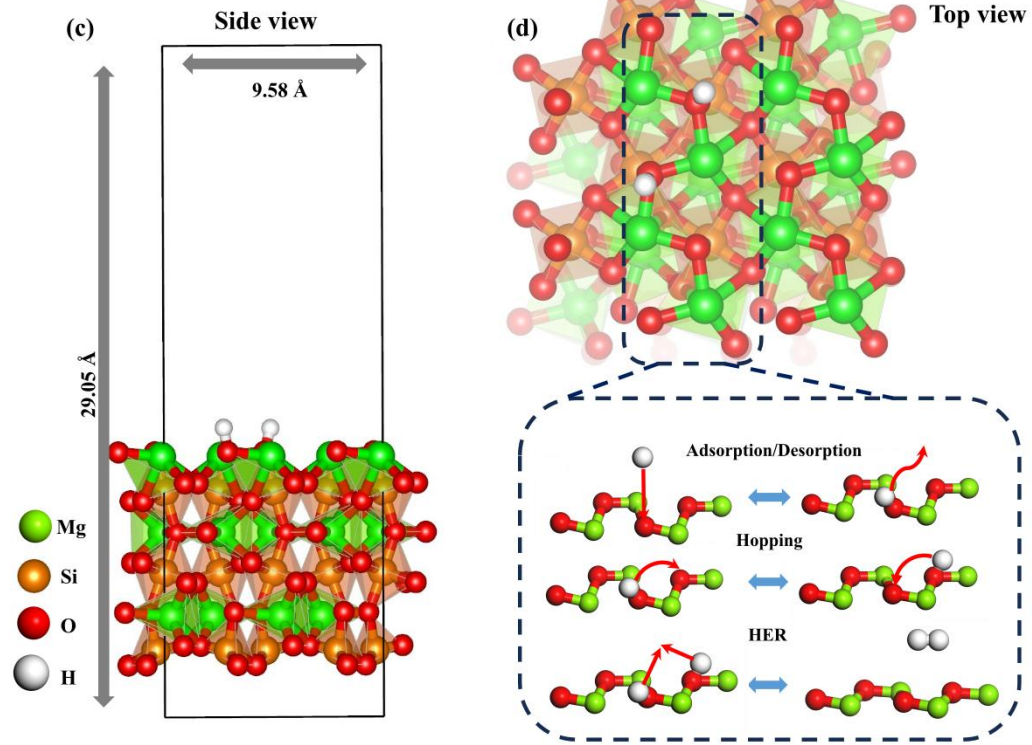
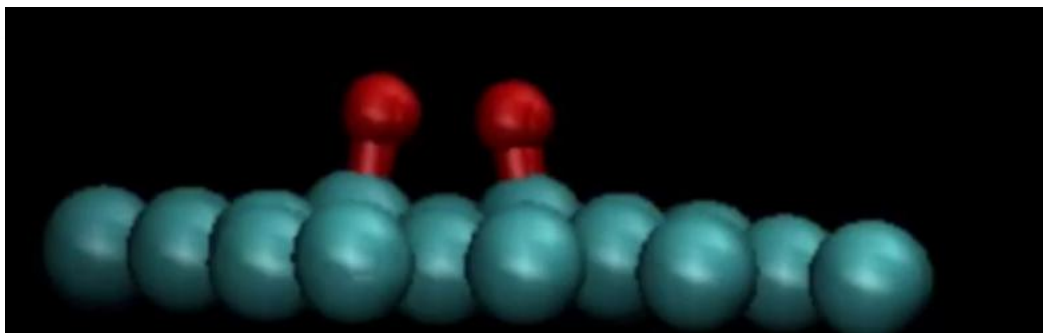
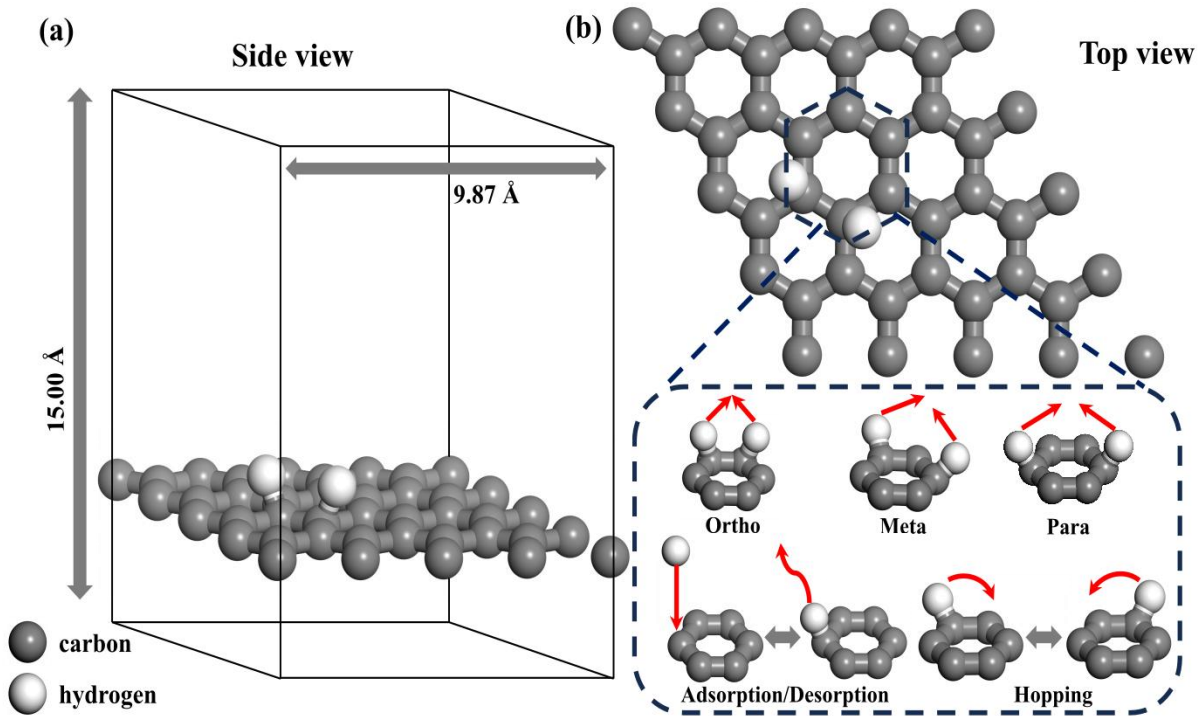
Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics
 Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, Weinan E
 Phys. Rev. Lett., 2018, 120, 143001.



DeePMD-kit

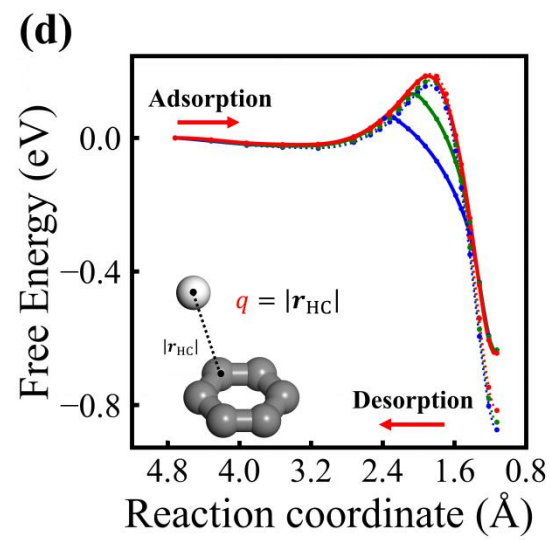
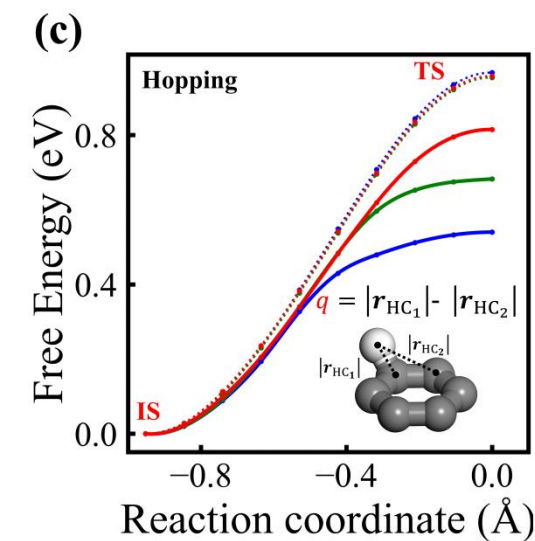
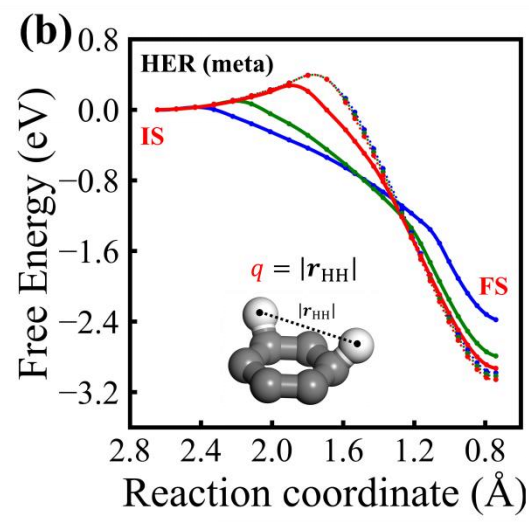
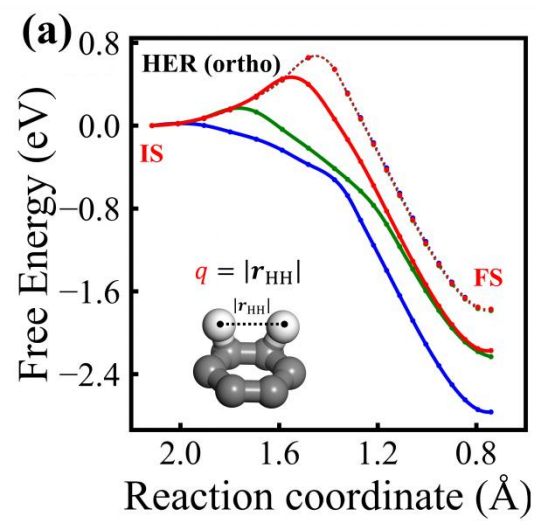
Wang, H., Zhang, L. F., Han, J. Q. & E, W. N. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Comput. Phys. Commun.* 228, 178-184 (2018).

深度势能的基本概念



如何进行核量子效应的定量计算？

Yang, Wang,
 Li & Xu, 2025,
 arXiv:2509.25070



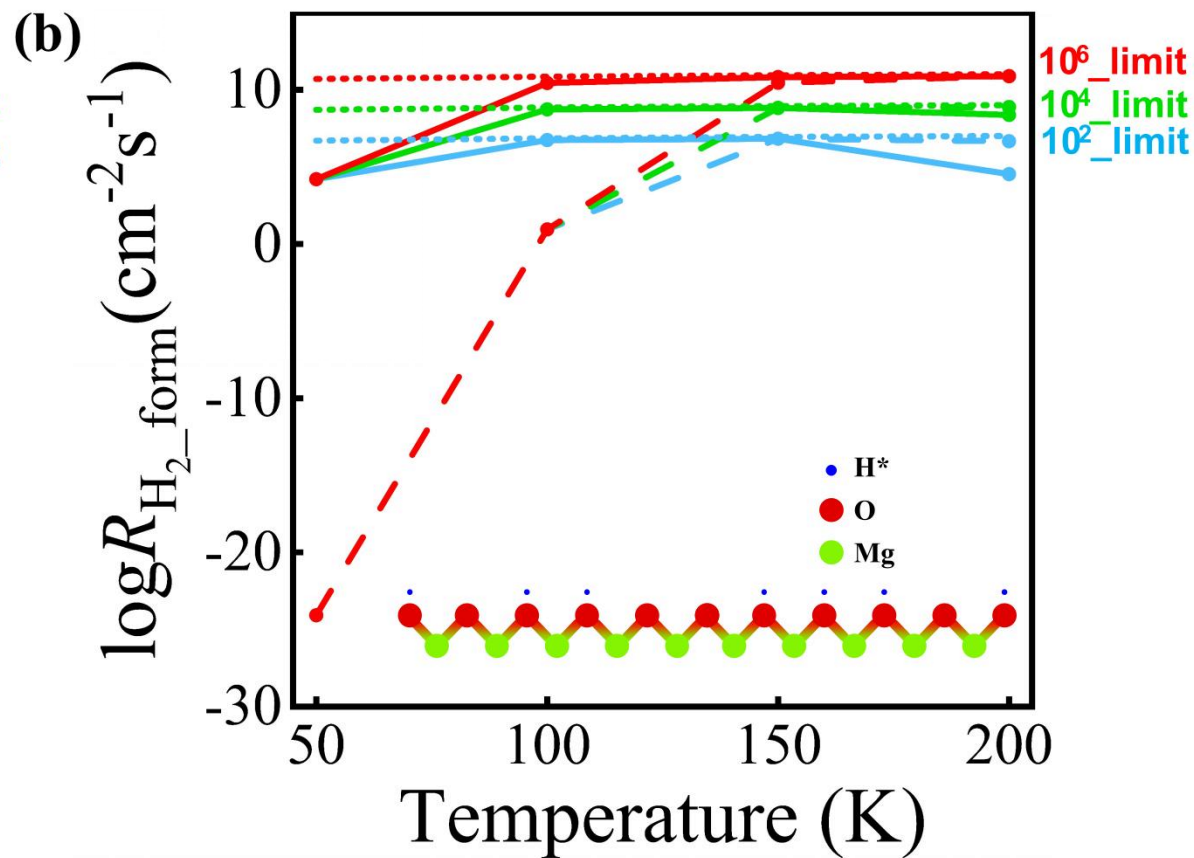
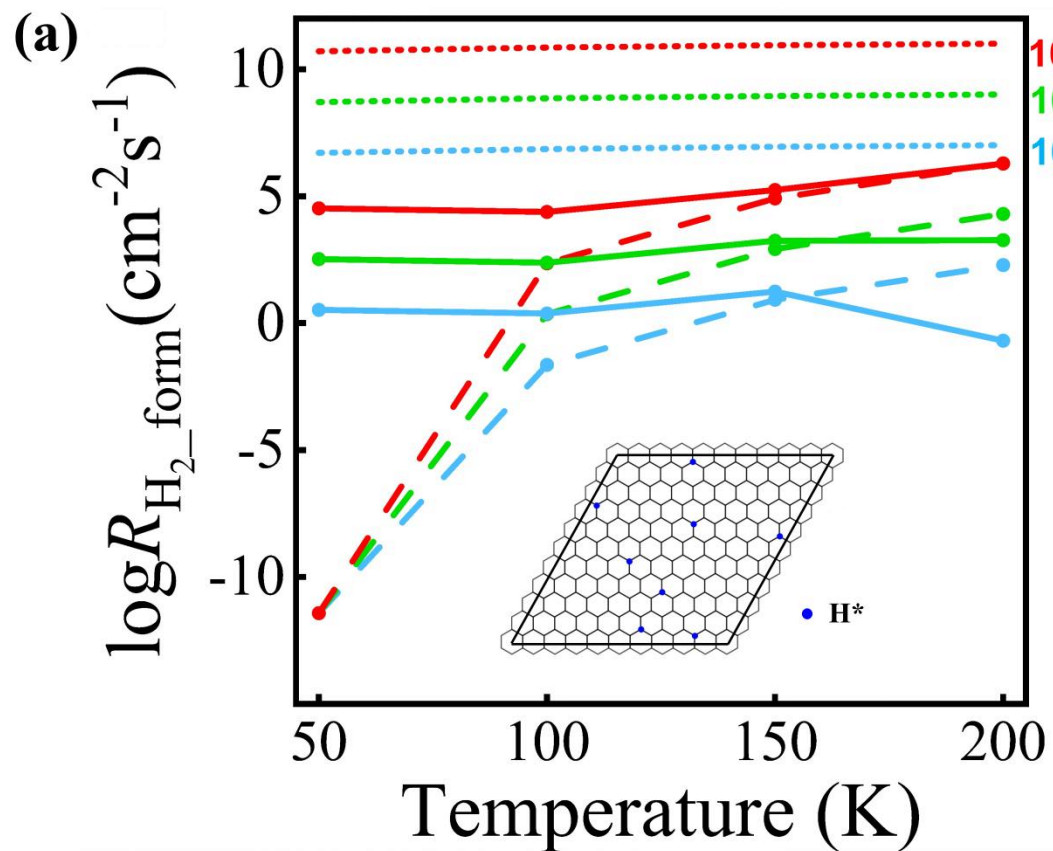
- 核量子效应给出更低的等效活化能 $\Delta E_{a,eff}$
 - 邻位: ~ 0.7 eV to ~ 0.03 eV
 - 间位: ~ 0.6 eV to ~ 0.02 eV
- 温度越低, 等效活化能 $\Delta E_{a,eff}$ 越低
 - $\lambda_{th,H} \sim 2.4 \text{ nm} \times (T/20 \text{ K})^{-1/2}$
 - 更长的热波长 λ_{th} , 带来更多可能路径
 - 路径更多, 找到更低活化能 ΔE_a 能力越强
- 物理吸附势阱太浅了, 未及形成分子先脱吸附
- 化学吸附 + 核量子效应才是关键

核量子效应确为关键

• KMC 结果给出：即便 $T < 50$ K，形成效率也有“兜底”

$$R = \frac{1}{2} n_H v_H n_d \sigma_d \eta S_H$$

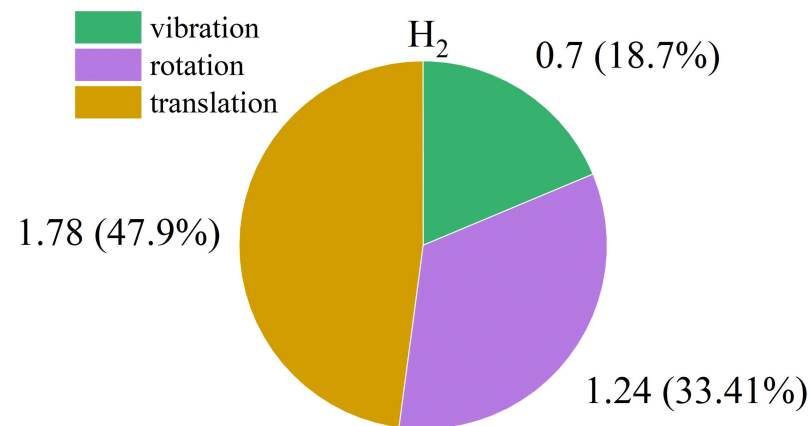
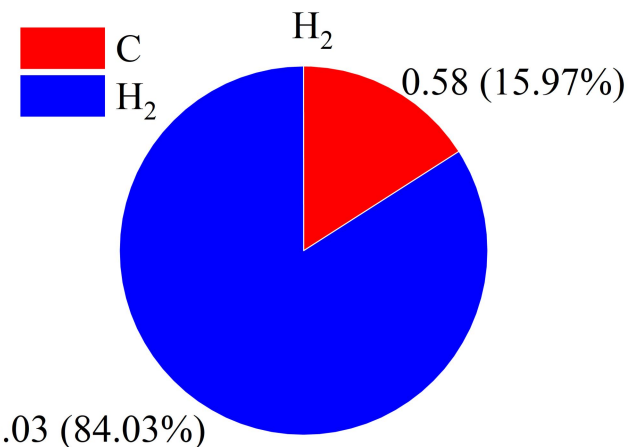
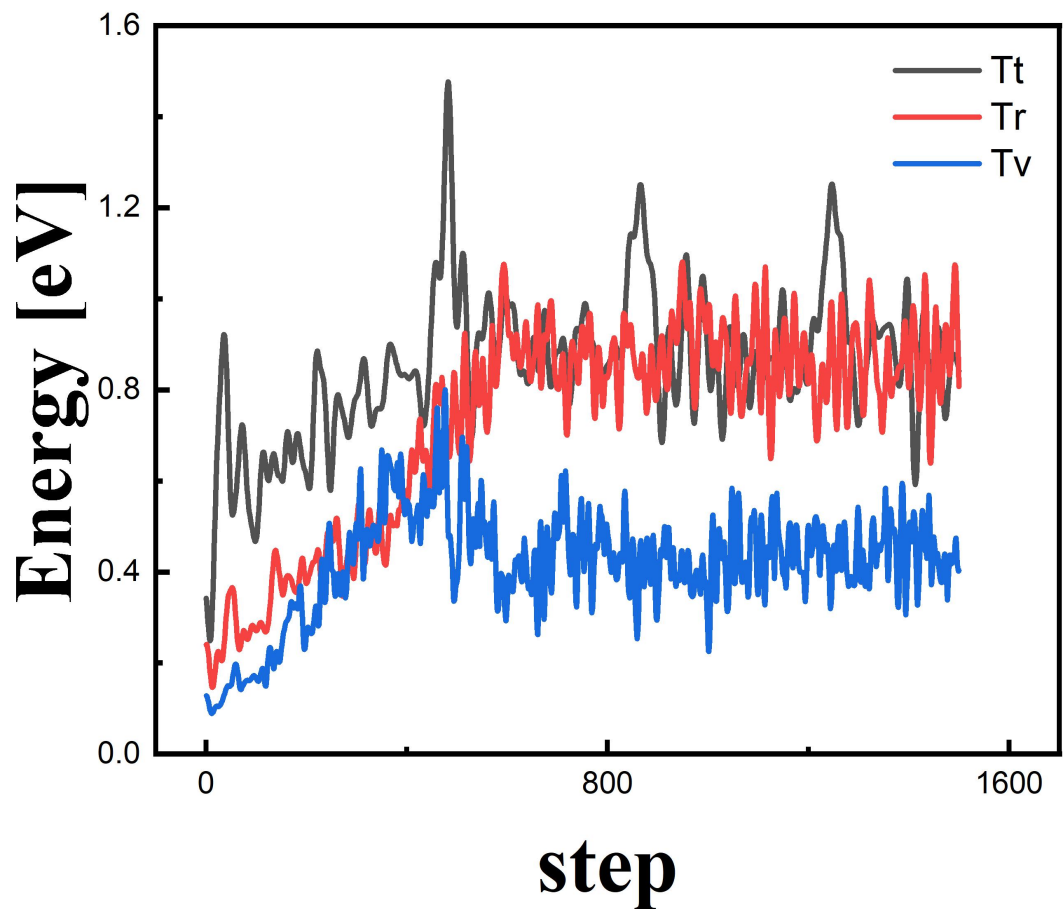
• 与此同时， $T > 150$ K 的高温段效率跌落但仍未为零（由于高温下较强的脱吸附）



核量子效应的结果：合理解释更高形成效率

Yang, Wang,
Li & Xu, 2025,
arXiv:2509.25070

50 K, 26 条反应轨道, 1500 步, 0.2 fs



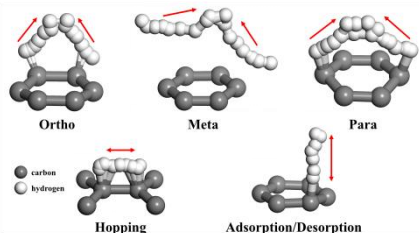
天文观测改进：通过非平衡能量配分反推

WORK
IN PROGRESS

微观
物理过程

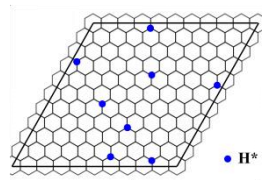


宏观天体
物理过程

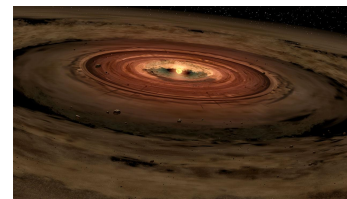


微观部分：
使用 DPMD 代理
昂贵的电子结构

“介观”部分：
使用动力学蒙卡



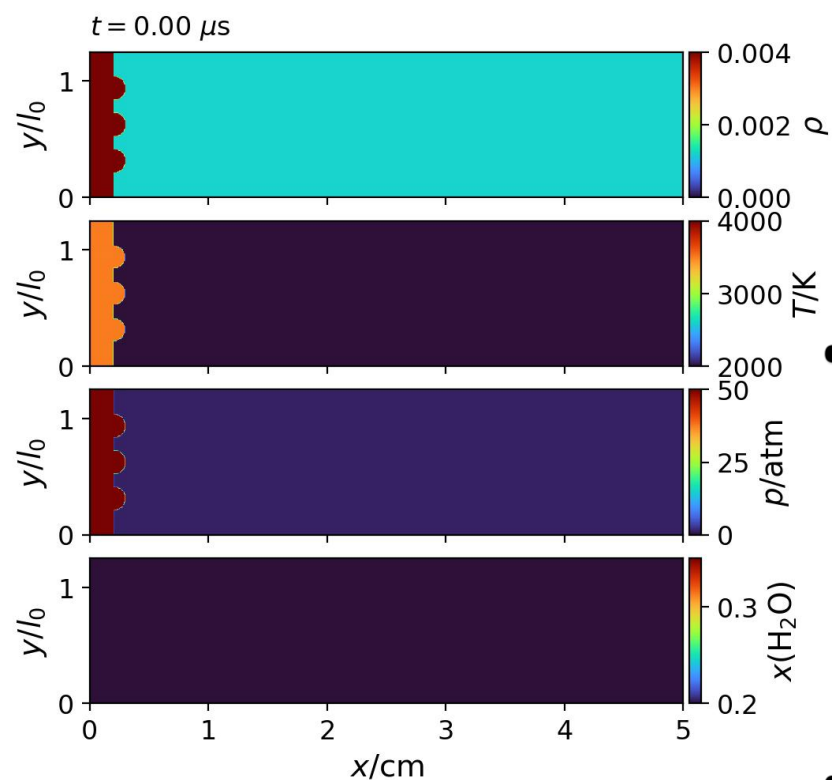
天文学可观测量：
用多物理场模拟
解决



总体而言：用 AI “代理” 最麻烦的微观部分

总结

- 天体物理高度依赖数值模拟
- 先进数值模拟可以拆成两部分
 - (通常) 高效、稳定的流体力学运算
 - 复杂、昂贵、困难的微观物理计算 (高维、强刚性)
- 动力学运算目前似乎仍应采用“传统算法”
 - 关键原因 1: 双曲系统的本质是守恒律, 导致误差积累
 - 关键原因 2: 非线性双曲系统存在间断性 (天文上重要)
 - 现实原因: 传统算法效率尚可, 代理后反而代价剧增
- 微观机制运算可以采用代理模型
 - 关键原因: 微观机制多体现为椭圆/抛物/本征系统, 对误差积累不敏感, 但使传统算法遭遇极大困难
 - 背后机制: 微观过程时标短、ML 对维度不太敏感
- 方法本身应有自限性和自查能力: **分而治之, 各管一段**

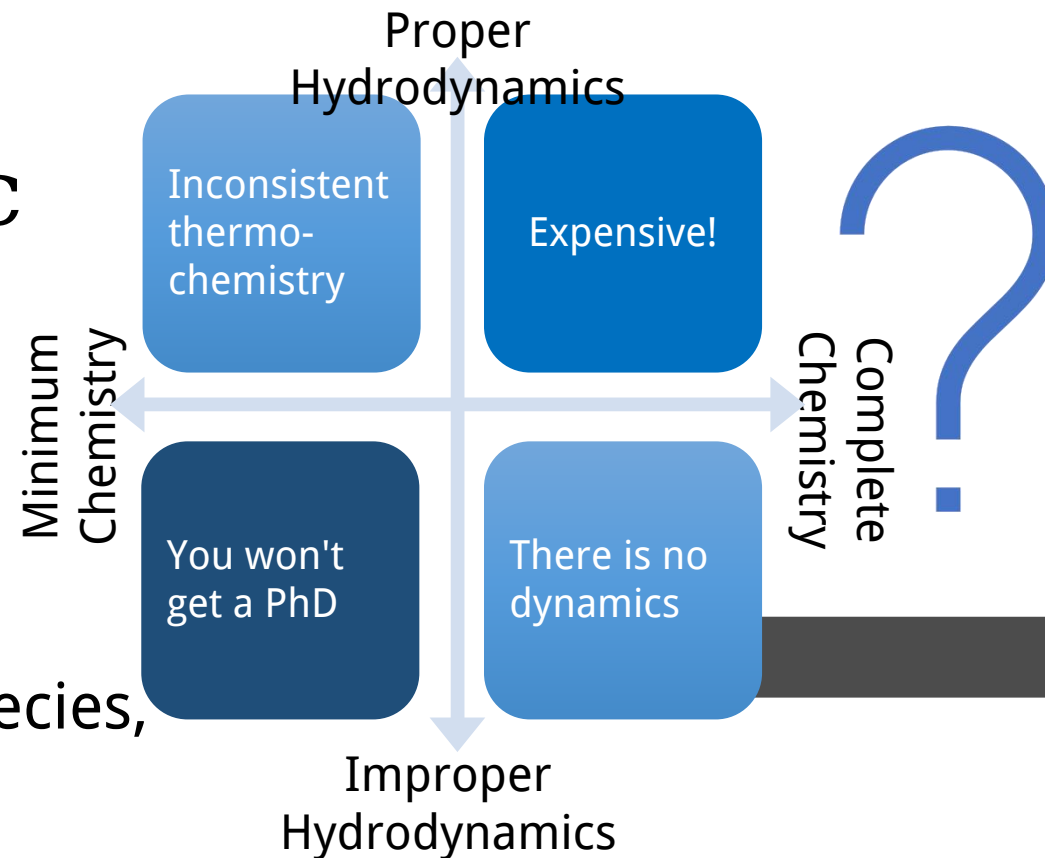


Why we need it —or, why not equilibrium?

Because sometimes we have

$$\tau_{\text{dyn}} \approx \tau_{\text{reac}}$$

Which occurs quite often



>30 Species >130 Reactions
Real-time Non-equilibrium
Cooling/Heating: Species by species,
process by process