大/小/微模型赋能先进制造：实践与思考

Large/Small/Micro AI Models for Manufacturing (AI4M): Applications and Insights

宋学官

大连理工大学 机械工程学院

一、AI4M的背景意义

二、AI4M的基础知识

三、AI4M的研究进展

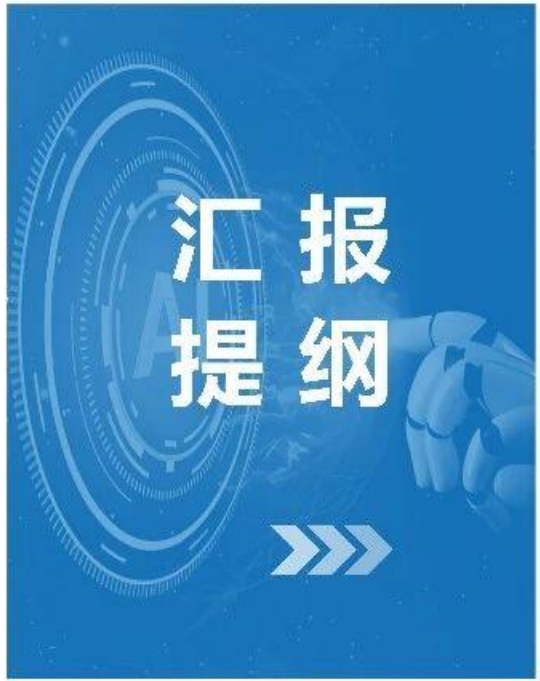
四、AI4M的案例展示

五、AI4M的瓶颈所在

六、AI4M的科学问题

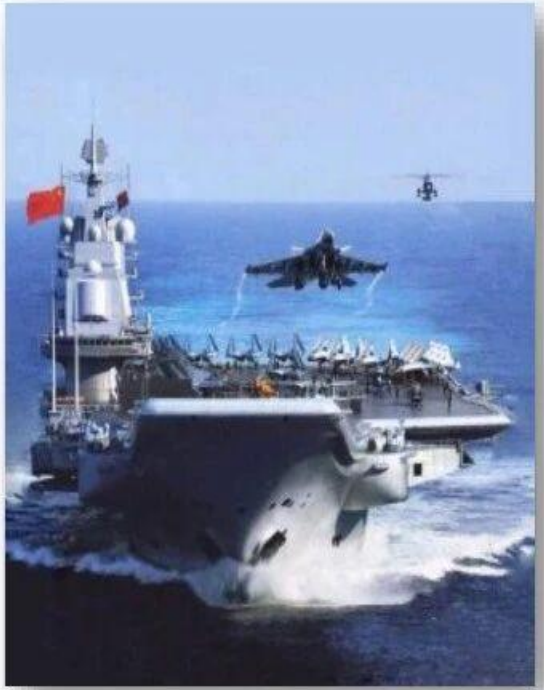
七、AI4M的发展方向

八、思考与总结

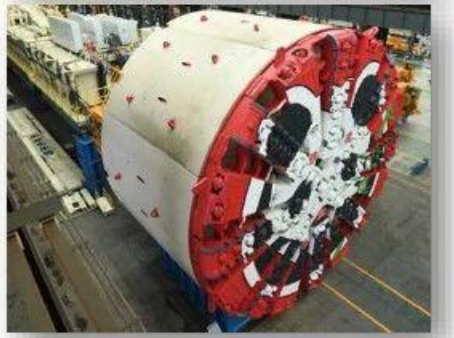


AI4M的背景意义

- **先进制造**是指采用高新技术和先进设备来改善制造业过程和生产效率的统称，是衡量一个国家科技发展水平的重要标志，关乎国民经济发展和国防安全建设。

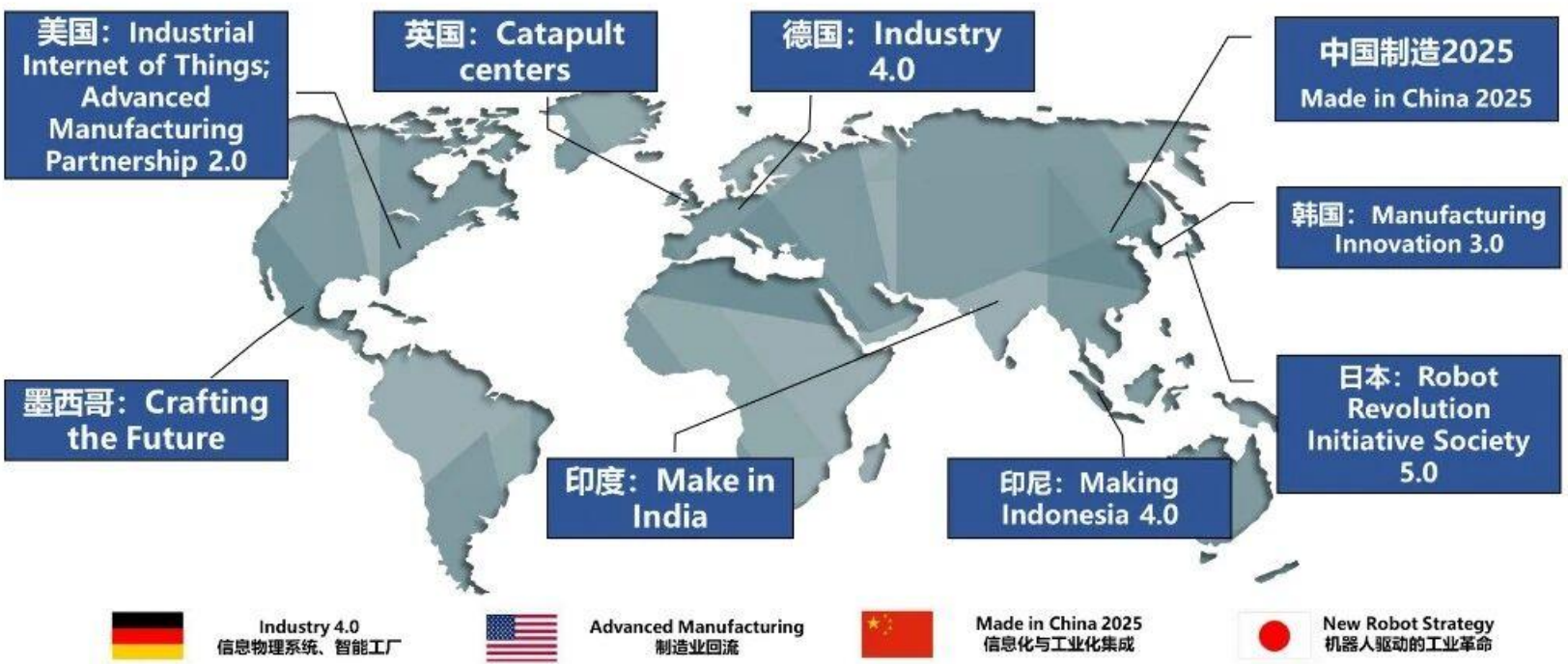




AI4M的背景意义

- 《中国制造2025》：加快推进制造业转型升级，**到2035年整体达到世界制造强国中等水平**
- 2022年10月，美国发布《国家先进制造业战略》，**先进制造业是美国经济和国家安全引擎**



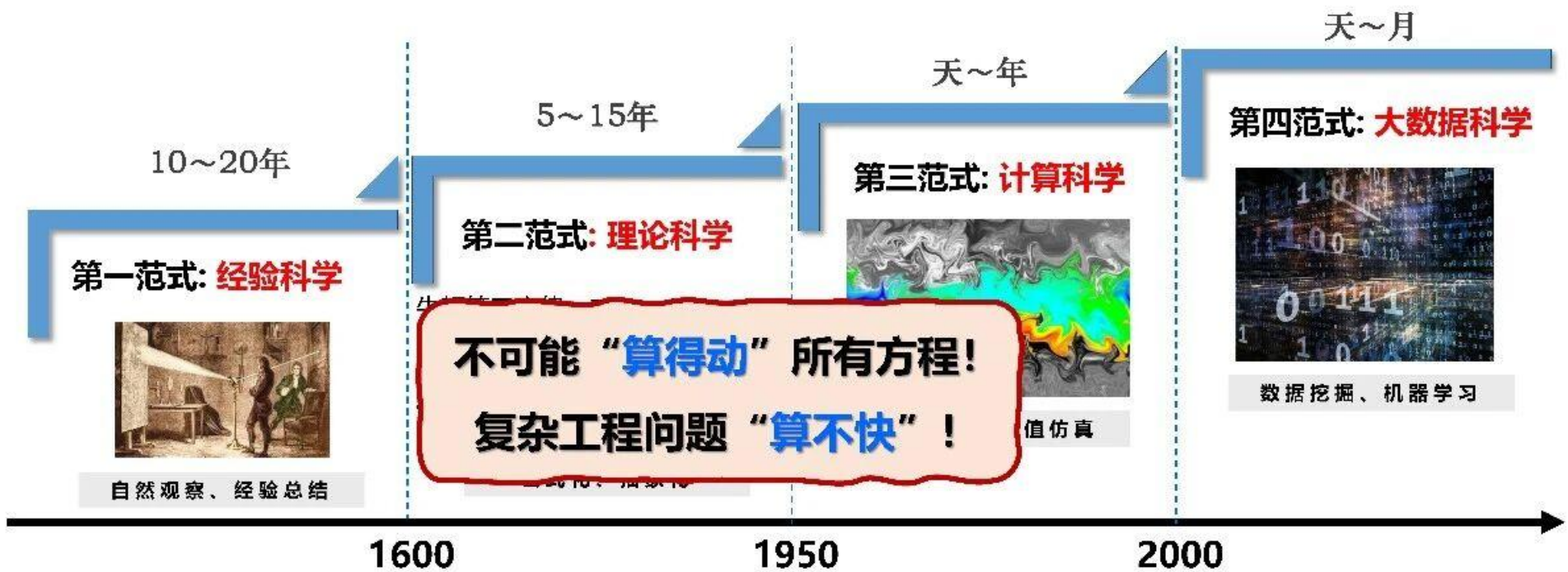
AI4M的背景意义

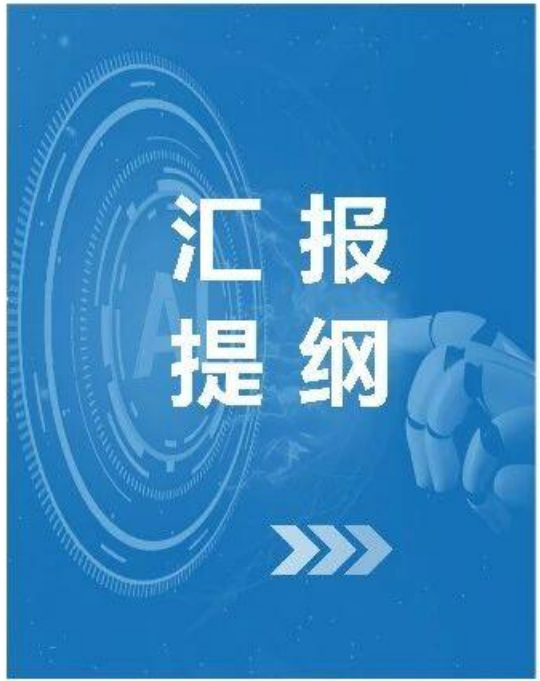
“AI4M”已成为工业4.0的核心技术之一，世界主要工业强国的重点发展布局方向

年份	国家	重大计划或布局名称	目标
2023	美国	《美国国家人工智能研究和发展战略计划》	明确把AI作为国家优先事项，持续推动通用AI在包括智能设计在内的多个关键领域发展。
2018	德国	《联邦政府人工智能战略要点》	形成“人工智能德国制造”产业品牌，积极推动AI知识与技术向中小企业转移，强调通过智能设计提升创新活力。
2021	法国	《国家人工智能第二阶段发展战略（2021-2025年）》	将“推动法国成为嵌入式AI和可信AI领域领导者”作为三大目标之一。通过设立跨学科研究中心，统筹推进智能设计与智能制造发展。
2020	日本	《制造业基础技术的振兴政策》	强调数字化、智能化转型是“日本制造”的关键，将AI技术作为推进产品设计制造环节应用的核心。
2021	中国	《“十四五”智能制造发展规划》	将设计仿真、混合建模、多目标协同优化等作为设计领域的聚焦重点，将AI、大数据等在工业领域内的适用性技术作为攻关核心之一。

AI4M的背景意义

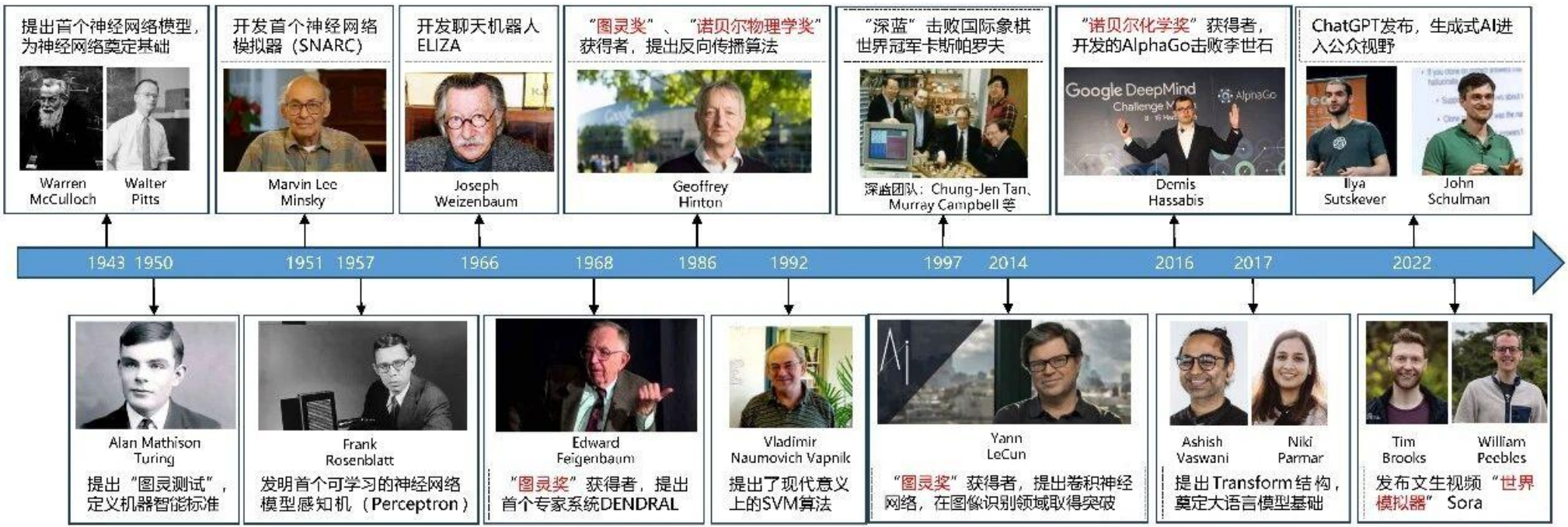
□ 人工智能(AI)：科学研究的第四范式，正深刻重塑先进制造全生命周期技术体系





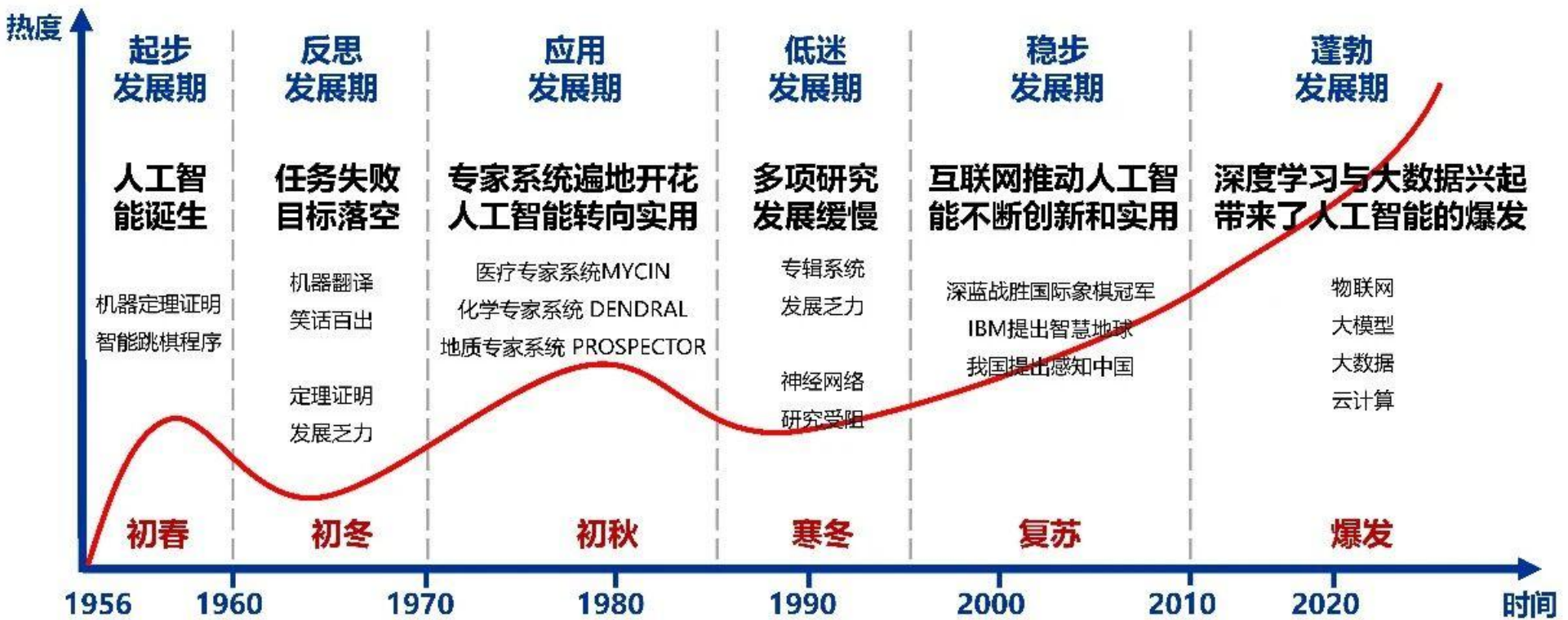
AI的发展历程

□ AI经过数十年发展才实现规模应用，在制造业的落地始终伴随着传感器和物联网技术的迭代升级。



AI的发展历程

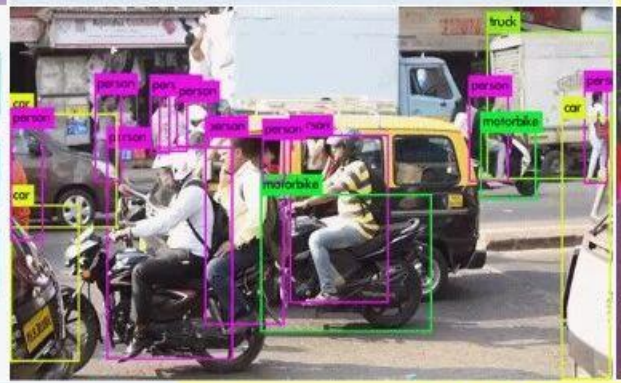
□ 人工智能经历三次热潮与两次寒冬，技术在挫折中不断演进，当前正迈向以大模型为核心的新阶段。



AI的不同分类方式

人脸检测识别

目标检测

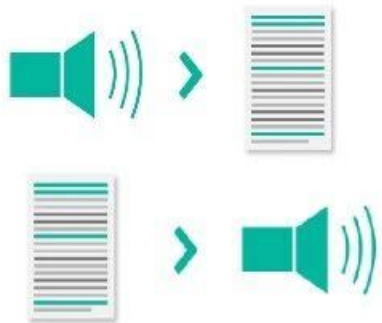


图像分割

图像生成

音乐合成

文本声音互转



AI的不同分类方式

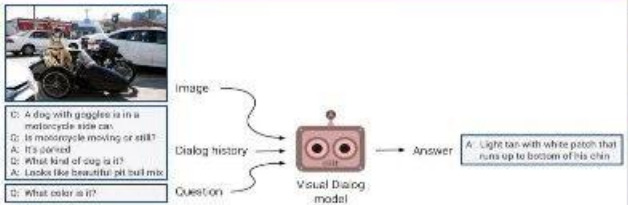
语言描述

Describes without errors

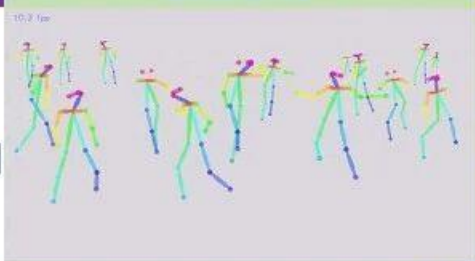
A person riding a motorcycle on a dirt road.

A group of young people playing a game of frisbee.

人机对话

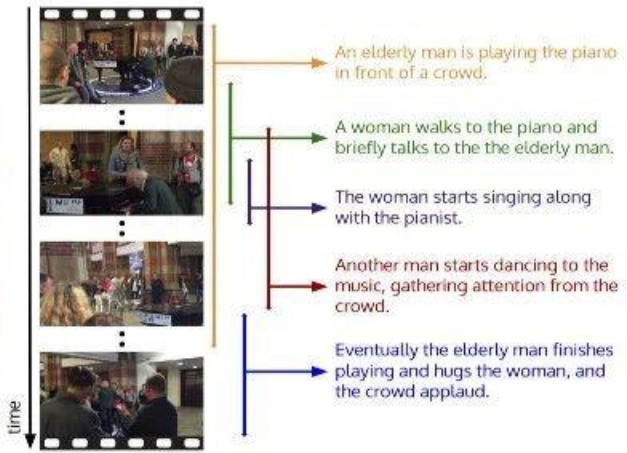


姿态估计



视频生成

视频解析



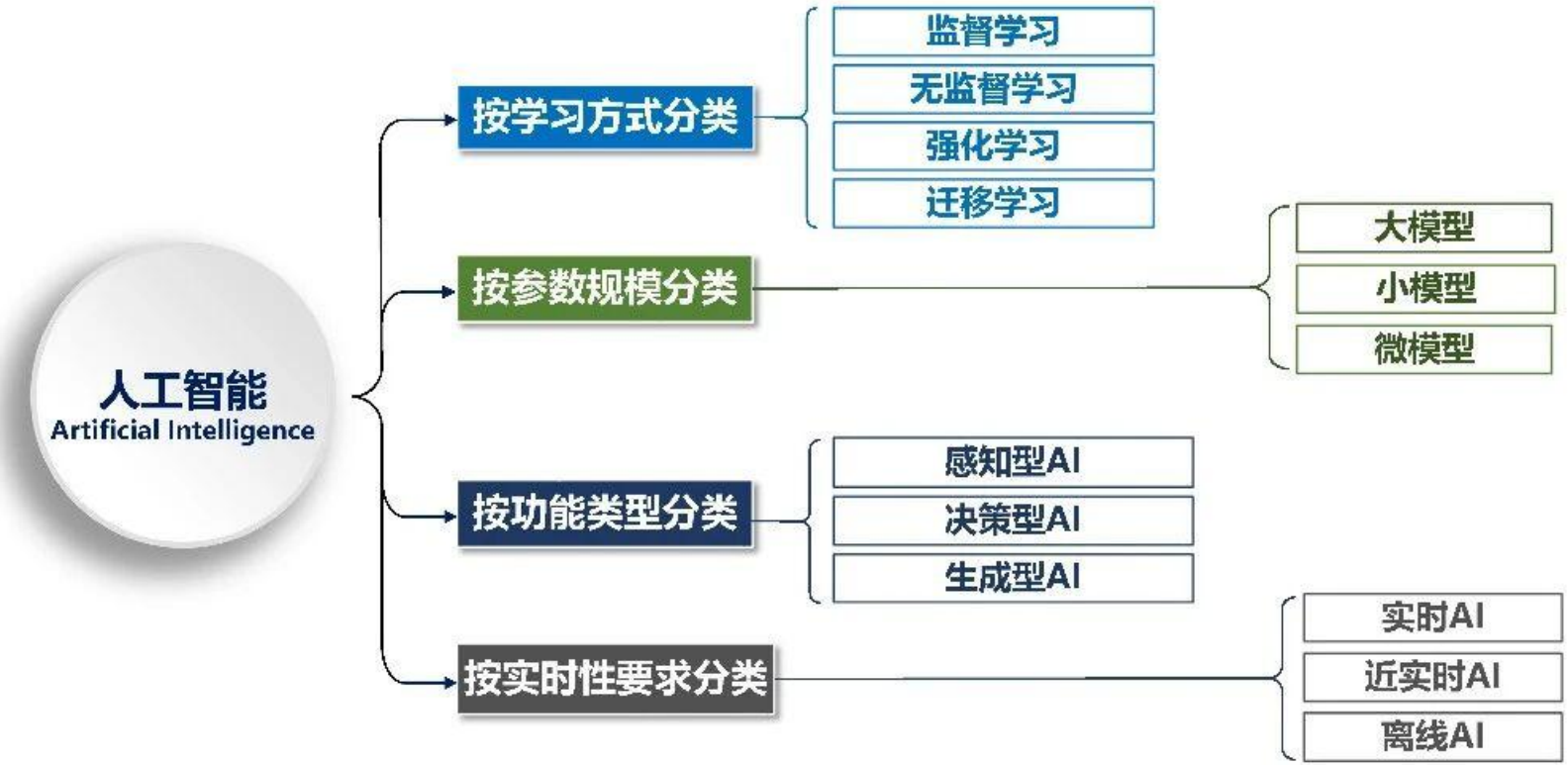
内容创作

deepseek

ChatGPT

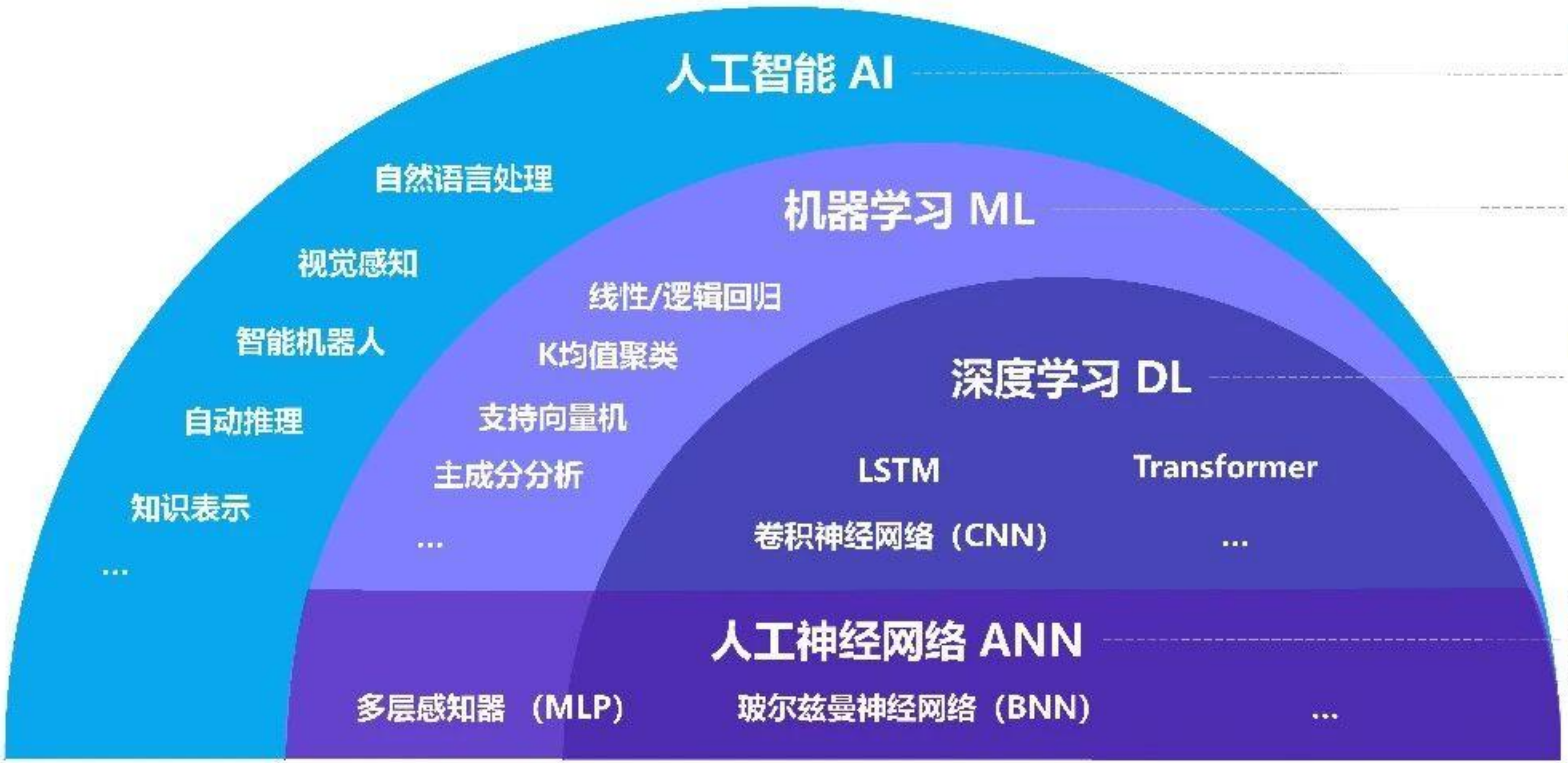
AI的不同分类方式

□ AI可以根据学习方式、参数规模、功能类型和实时性要求等不同维度进行分类



AI的不同分类方式

□ AI可以根据模型结构复杂度和智能实现方式的差异进行系统性划分。



AI≈建造智能机器
作为整体领域，包含所有模拟人类智能的技术

ML≈让机器自己学
人工智能的一个重要分支，无需手动编程，机器通过数据训练，总结规律，自动改进性能

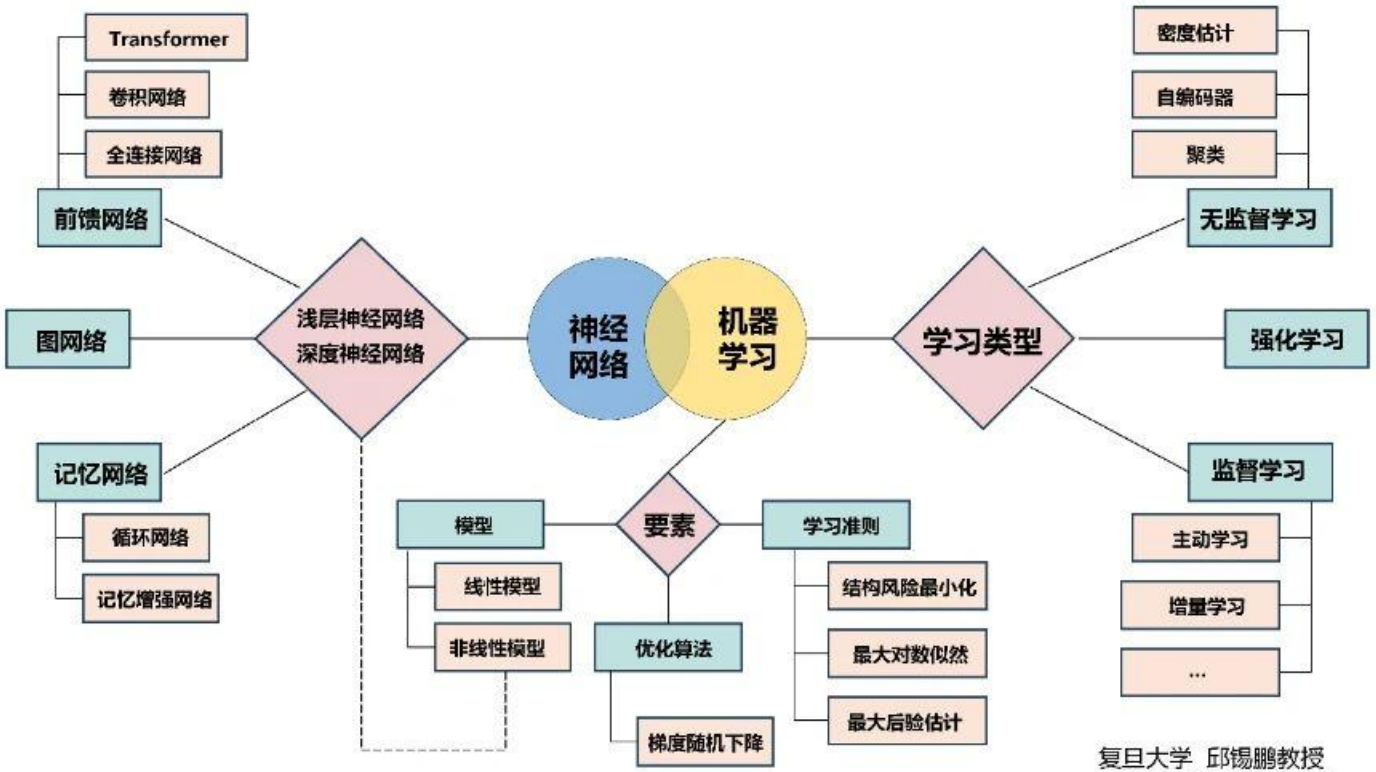
DL≈用复杂模型学
机器学习的重要分支，使用多层神经网络处理复杂数据，自动提取高层次特征。“深度”即：在网络中使用多层

神经网络≈大脑的简化版计算单元
又称类神经网络/神经网络，是一种通过模拟人脑神经元的计算模型，由多层互联节点（神经元）组成

网址：https://zhuanlan.zhihu.com/p/1889735973048395146

AI的不同分类方式

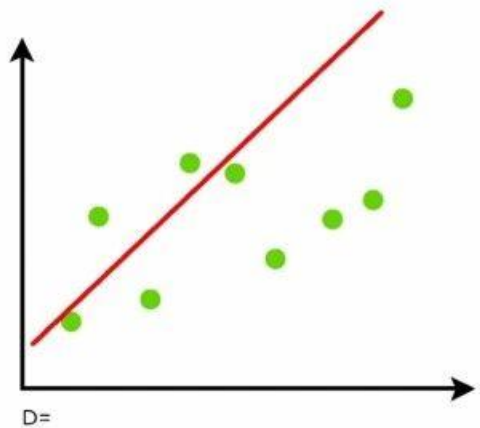
□ AI可以根据模型深度、学习策略和学习方式的不同进行系统性划分。



复旦大学 邱锡鹏教授
网址：https://cloud.tencent.cn/developer/article/1630825

最小二乘法

□ 一种简单有效的非线性拟合方法，适合低维、平滑、连续且样本适中的数据



D=

问题描述：

给定一组数据点$(x_i, y_i), i=1,2...n$，我们希望找到一个m次多项式：

$$y(x) = w_0 + w_1 x + w_2 x^2 \dots + w_m x^m$$

使得误差平方和最小：$E = \sum_{i=1}^{n} \left(y_i - P(x_i)\right)^2$

构建误差函数：

将多项式带入误差函数：

$$E(w_0, w_1, \dots, w_m) = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{m} w_j x_i^j\right)^2$$

最小化误差：

为了最小化E，对每个系数w_k求偏导并令其为零：

$$\frac{\partial E}{\partial w_k} = 0, \quad k = 0,1,\dots m$$

计算偏导数：

$$\frac{\partial E}{\partial w_k} = -2\sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{m} w_j x_i^j\right) x_i^k = 0$$

$$\sum_{i=1}^{n} y_i x_i^k = \sum_{j=0}^{m} w_j \sum_{i=1}^{n} x_i^{j+k}, \quad k = 0,1,\dots m$$

方程组：

将上述方程写成矩阵形式（正规方程组）：

$$\begin{cases} w_0 + w_1 \sum x_i + w_2 \sum x_i^2 + \cdots + w_m \sum x_i^m = \sum y_i \\ w_0 \sum x_i + w_1 \sum x_i^2 + w_2 \sum x_i^3 \dots + w_m \sum x_i^{m+1} = \sum y_i x_i \\ \cdots \\ w_0 \sum x_i^m + w_1 \sum x_i^{m+1} + w_2 \sum x_i^{m+2} \dots + w_m \sum x_i^{2m} = \sum y_i x_i^m \end{cases}$$

用矩阵表示：

$$X^T X w = X^T y$$

其中：

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_m \end{bmatrix}, \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_n \end{bmatrix}$$

求解系数：

解方程组：

$$w = (X^T X)^{-1} X^T y$$

关键点：

当$m < n-1$时，通常有唯一解。

矩阵$X^T X$是对称且正定的（如果X列满秩）。

示例：

$$\begin{cases} w_0 n + w_1 \sum x_i = \sum y_i \\ w_0 \sum x_i + w_1 \sum x_i^2 = \sum y_i x_i \end{cases}$$

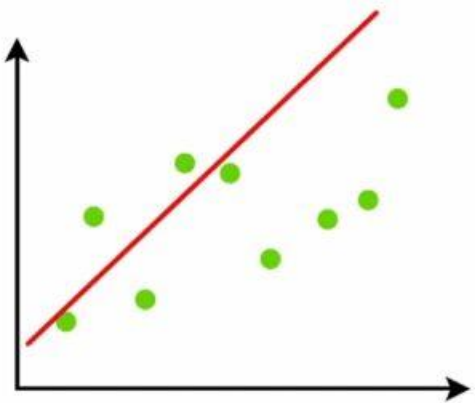
解得：

$$w_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}, \quad w_0 = \frac{\sum y_i - w_1 \sum x_i}{n}$$

模型求解过程

多项式拟合

□ 一种简单有效的非线性拟合方法，适合低维、平滑、连续且样本适中的数据



D=

构造多项式特征：

模型： 二阶多项式模型

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2$$

数值案例： 输入x=$[1,2]^T$，目标输出：y=10

初始参数：$w(0) = [0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1]^T$

学习率： $\eta = 0.01$

模型架构

输入向量扩展为多项式特征向量：

$$\phi(x) = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1x_2 \quad x_2^2]^T = [1 \quad 1 \quad 2 \quad 1 \quad 2 \quad 4]^T$$

计算初始预测值：

$$\hat{y}^{(0)} = w^{(0)} \cdot \phi(x) = 0.1 \times 1 + 0.1 \times 1 + 0.1 \times 2 + 0.1 \times 1 + 0.1 \times 2 + 0.1 \times 4 = 1.1$$

损失函数：

计算损失函数，使用均方误差（MSE）：

$$L = (\hat{y} - y)^2 = (1.1 - 10)^2 = 79.21$$

反向传播：

梯度计算：

损失函数对权重的梯度：

$$\frac{\partial L}{\partial w_i} = (\hat{y} - y) \cdot \phi(i)$$

所以每一项的梯度：

$$\frac{\partial L}{\partial w} = (1.1 - 10) \cdot \begin{bmatrix}1\\1\\2\\1\\2\\4\end{bmatrix} = \begin{bmatrix}-8.9\\-8.9\\-17.8\\-8.9\\-17.8\\-35.6\end{bmatrix}$$

参数更新：

$$w^{(1)} \leftarrow w^{(0)} - \eta \cdot \nabla L$$

$$= \begin{bmatrix}1\\1\\1\\1\\1\\1\end{bmatrix} - 0.1 \cdot \begin{bmatrix}-8.9\\-8.9\\-17.8\\-8.9\\-17.8\\-35.6\end{bmatrix} = \begin{bmatrix}1.89\\1.89\\2.78\\1.89\\2.78\\4.56\end{bmatrix}$$

计算结果：

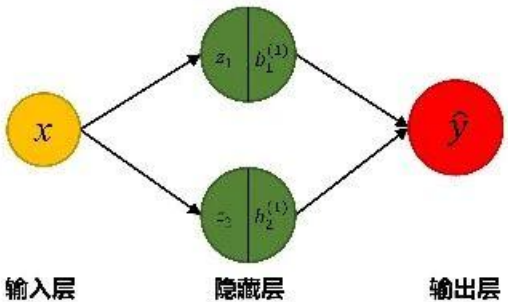
对上述过程重复100次，输出$\hat{y}$和MSE

迭代次数	输出$\hat{y}$	MSE
1	1.10	79.21
2	3.48	42.50
...	...	...
100	≈1	$<10^5$

模型求解过程

浅层神经网络

□ 模型结构简单，训练速度快，适用于小数据集，适合低维数据分类/回归。



模型架构： 输入层 （1个神经元）
隐藏层 （2个神经元，激活函数为ReLU）
输出层 （1个神经元，线性激活）

数值案例： 输入$x=[1,2]^T$，输出：$y=10$

初始参数： $W_1=\begin{bmatrix}0.1 & 0.2\\ 0.3 & 0.4\end{bmatrix}$，$b_1=[0.0 \quad 0.0]^T$

$W_2=[0.5 \quad 0.6]$，$b_2=0.0$

学习率： $\eta=0.01$

模型架构

前向传播：

隐藏层输入：$z_1=W_1x+b_1=\begin{bmatrix}0.1 & 0.2\\ 0.3 & 0.4\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}0.1+0.4\\0.3+0.8\end{bmatrix}=\begin{bmatrix}0.5\\1.1\end{bmatrix}$

隐藏层输出：$a_1=ReLU(z_1)=\begin{bmatrix}0.5\\1.1\end{bmatrix}$，激活函数$ReLU=\begin{cases}0, x<0\\ x, x\geq 0\end{cases}$

输出层输入：$z_2=W_2a_1=[0.5 \quad 0.6]\cdot\begin{bmatrix}0.5\\1.1\end{bmatrix}=0.25+0.66=0.91$

网络输入：$\hat{y}=z_2=0.91$

损失函数：

$L=\frac{1}{2}(y-\hat{y})^2=\frac{1}{2}(10-0.91)^2=\frac{1}{2}\cdot 82.81\approx 41.40$

反向传播：

输出层误差：$\delta_2=\hat{y}-y=0.91-10=-9.09$

输出层梯度：$\frac{\partial L}{\partial W_2}=\delta_2\cdot a_1^T=-9.09\cdot[0.5 \quad 1.1]=[-4.545 \quad -9.999]$

$\frac{\partial L}{\partial b_2}=\delta_2=-9.09$

隐藏层误差：

$\delta_1=(W_2^T\cdot\delta_2)\cdot ReLU'(z_1)\rightarrow\begin{bmatrix}0.5\\0.6\end{bmatrix}\cdot(-9.09)=\begin{bmatrix}-4.545\\-5.454\end{bmatrix}=\delta_1$

隐藏层梯度：$\frac{\partial L}{\partial W_1}=\delta_1\cdot x^T=\begin{bmatrix}-4.545\\-5.454\end{bmatrix}[1 \quad 2]=\begin{bmatrix}-4.545 & -9.090\\-5.454 & -10.908\end{bmatrix}$

$\frac{\partial L}{\partial b_1}=\delta_1=\begin{bmatrix}-4.545\\-5.454\end{bmatrix}$

参数更新：

$W_2\leftarrow W_2-\eta\frac{\partial L}{\partial W_2}$

$=[0.5 \quad 0.6]-0.01\cdot[-4.545 \quad -9.999]$

$=[0.5454 \quad -0.6999]$

$b_2\leftarrow b_2-\eta\cdot(-9.09)=0+0.0909=0.0909$

$W_1\leftarrow W_1-\eta\cdot\frac{\partial L}{\partial W_1}=\begin{bmatrix}0.14545 & 0.29090\\0.35454 & 0.50908\end{bmatrix}$

$b_1\leftarrow b_1-\eta\cdot\delta_1=\begin{bmatrix}0.04545\\0.05454\end{bmatrix}$

计算结果：

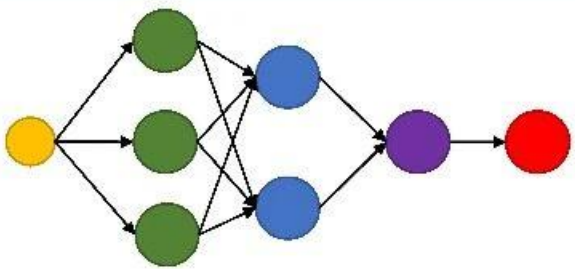
对上述过程重复100次，输出$\hat{y}$和MSE

迭代次数	输出$\hat{y}$	MSE
1	0.91	82.80
2	1.83	66.60
…	…	…
100	9.998	$<10^{-5}$

模型求解过程

深度学习神经网络（DNN）

□ 模型结构层次深，特征提取能力强，适用于**大规模数据集**，专为图像识别等数据设计。



模型架构： 输入层 （1个神经元）
隐藏层1（3个神经元，激活函数为ReLU）
隐藏层2（2个神经元，激活函数为ReLU）
隐藏层3（1个神经元，激活函数为ReLU）
输出层 （1个神经元，线性激活）

数值案例： 输入$x=[1,2]^T$，目标输出：y=10

初始参数： $W_1=\begin{bmatrix}0.1 & 0.2\\ 0.3 & 0.4\\ 0.5 & 0.6\end{bmatrix}$，$b_1=[0 \quad 0 \quad 0]^T$

$W_2=\begin{bmatrix}0.1 & 0.2 & 0.3\\ 0.4 & 0.5 & 0.6\end{bmatrix}$，$b_2=[0 \quad 0]^T$

$W_3=[0.1 \quad 0.2]$，$b_3=0$

$W_4=[0.3]$，$b_4=0$

学习率： $\eta=0.01$

模型架构

前向传播：

隐藏层1输入：$z_1=W_1x+b_1=\begin{bmatrix}0.1 & 0.2\\ 0.3 & 0.4\\ 0.5 & 0.6\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}+\begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}0.5\\1.1\\1.7\end{bmatrix}$

隐藏层1输出：$a_1=ReLU(z_1)=\begin{bmatrix}0.5\\1.1\\1.7\end{bmatrix}$

隐藏层2输入：$z_2=W_2x+b_2=\begin{bmatrix}0.1 & 0.2 & 0.3\\ 0.4 & 0.5 & 0.6\end{bmatrix}\begin{bmatrix}0.5\\1.1\\1.7\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}0.78\\1.77\end{bmatrix}$

隐藏层2输出：$a_2=ReLU(z_2)=\begin{bmatrix}0.78\\1.77\end{bmatrix}$

隐藏层3输入：$z_3=W_3x+b_3=[0.1 \quad 0.2]\begin{bmatrix}0.78\\1.77\end{bmatrix}+0=0.432$

隐藏层3输出：$a_3=ReLU(z_3)=0.432$

输出层输入：$z_4=W_4a_3=0.3\times0.432+0=0.1296$

网络输入：$\hat{y}=z_4=0.1296$

损失函数：

$L=\frac{1}{2}(y-\hat{y})^2=\frac{1}{2}(10-0.1296)^2=\frac{1}{2}\times9.8704^2\approx48.71$

反向传播：

输出层误差：$\delta_4=\frac{\partial L}{\partial z_4}=\hat{y}-y=0.1296-10=-9.8704$

隐藏层3误差：$\delta_3=(W_4\cdot\delta_4)\odot ReLU'(z_3)$
$=0.3\times(-9.8704)\times1=-2.9611$

隐藏层2误差：$\delta_2=(W_3\cdot\delta_3)\odot ReLU'(z_2)$
$=\begin{bmatrix}0.1\\0.2\end{bmatrix}\times(-2.9611)\odot\begin{bmatrix}1\\1\end{bmatrix}$
$=\begin{bmatrix}-0.2961\\-0.5922\end{bmatrix}$

隐藏层1误差：$\delta_1=(W_2\cdot\delta_2)\odot ReLU'(z_1)$
$=\left(\begin{bmatrix}0.1 & 0.4\\ 0.2 & 0.5\\ 0.3 & 0.6\end{bmatrix}\begin{bmatrix}-0.2961\\-0.5922\end{bmatrix}\right)\odot ReLU'\left(\begin{bmatrix}0.5\\1.1\\1.7\end{bmatrix}\right)$
$=\begin{bmatrix}-0.26649\\-0.35532\\-0.44415\end{bmatrix}$

输出层梯度：$\frac{\partial L}{\partial W_4}=\delta_4\cdot a_3=0.432\times(-9.8704)$
$=-4.263$

$\frac{\partial L}{\partial b_4}=\delta_4=-9.8704$

隐藏层3梯度：$\frac{\partial L}{\partial W_3}=\delta_3\cdot a_2=-2.9611\times\begin{bmatrix}0.78\\1.77\end{bmatrix}$
$=\begin{bmatrix}-2.3096\\-5.2433\end{bmatrix}$

模型求解过程

深度学习神经网络（DNN）

□ 模型结构层次深，特征提取能力强，适用于大规模数据集，专为图像识别等数据设计。

$$\frac{\partial L}{\partial b_3} = \delta_3 = -2.9611$$

隐藏层2梯度：$\frac{\partial L}{\partial W_2} = a_1 \cdot \delta_2^T = \begin{bmatrix}0.5\\1.1\\1.7\end{bmatrix} \times [-0.2961 \quad -0.5922] = \begin{bmatrix}-0.1481 & -0.2961\\-0.3257 & -0.6514\\-0.5034 & -1.0067\end{bmatrix}$

$$\frac{\partial L}{\partial b_2} = \delta_2 = \begin{bmatrix}-0.2961\\-0.5922\end{bmatrix}$$

隐藏层1梯度：$\frac{\partial L}{\partial W_1} = x \cdot \delta_1^T = \begin{bmatrix}1\\2\end{bmatrix} \times [-0.26649 \quad -0.35532 \quad -0.44415]$

$$= \begin{bmatrix}-0.26649 & -0.35532 & -0.44415\\-0.53298 & -0.71064 & -0.88830\end{bmatrix}$$

$$\frac{\partial L}{\partial b_1} = \delta_1 = \begin{bmatrix}-0.26649\\-0.35532\\-0.44415\end{bmatrix}$$

参数更新：

$$W_l = W_l - \eta \cdot \frac{\partial L}{\partial W_l}$$

$$b_l = b_l - \eta \cdot \frac{\partial L}{\partial b_l}$$

输出层更新：$W_4 = 0.3 - 0.01 \times (-4.263) = 0.3426$

$b_4 = 0 - 0.01 \times (-9.8704) = 0.0987$

隐藏层3更新：$W_3 = \begin{bmatrix}0.1\\0.2\end{bmatrix} - 0.01 \times \begin{bmatrix}-2.3096\\-5.2433\end{bmatrix} = \begin{bmatrix}0.1231\\0.2524\end{bmatrix}$

$b_3 = 0 - 0.01 \times (-2.9611) = 0.0296$

隐藏层2更新：$W_2 = \begin{bmatrix}0.1 & 0.4\\0.2 & 0.5\\0.3 & 0.6\end{bmatrix} - 0.01 \times \begin{bmatrix}-0.1481 & -0.2961\\-0.3257 & -0.6514\\-0.5034 & -1.0067\end{bmatrix} = \begin{bmatrix}0.1015 & 0.4029\\0.2033 & 0.5065\\0.3050 & 0.6101\end{bmatrix}$

$$b_2 = \begin{bmatrix}0\\0\end{bmatrix} - 0.01 \times \begin{bmatrix}-0.2961\\-0.5922\end{bmatrix} = \begin{bmatrix}0.00296\\0.00592\end{bmatrix}$$

隐藏层1更新：$W_1 = \begin{bmatrix}0.1 & 0.3 & 0.5\\0.2 & 0.4 & 0.6\end{bmatrix} - 0.01 \times \begin{bmatrix}-0.26649 & -0.35532 & -0.44415\\-0.53298 & -0.71064 & -0.88830\end{bmatrix}$

$$= \begin{bmatrix}0.10266 & 0.30355 & 0.50444\\0.20533 & 0.40711 & 0.60888\end{bmatrix}$$

$$b_1 = \begin{bmatrix}0\\0\\0\end{bmatrix} - 0.01 \times \begin{bmatrix}-0.26649\\-0.35532\\-0.44415\end{bmatrix} = \begin{bmatrix}0.00266\\0.00355\\0.00444\end{bmatrix}$$

计算结果：

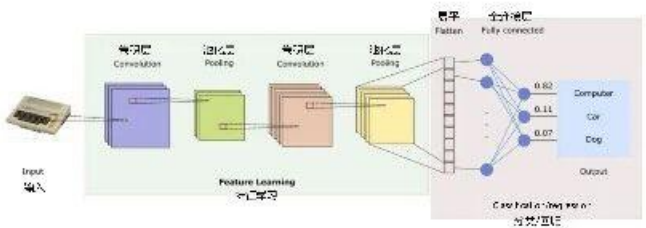
对上述过程重复100次，输出$\hat{y}$和MSE

迭代次数	输出$\hat{y}$	MSE
1	0.1296	96.84
2	0.3448	93.24
…	…	…
100	10.6241	0.3895

模型求解过程

卷积神经网络（CNN）

□ 模型通过局部连接和权值共享机制，能够高效提取图像局部特征，适用于机械制造中的视觉检测任务，如零件表面划痕检测、尺寸精度测量和装配完整性检查等。



模型架构： 输入层：

卷积层：1个3*3卷积核，步长1，输出大小3*3（ReLU激活函数）

池化层：2*2最大池化，步长2，输出1*1

全连接层：1个隐藏神经元，ReLU

输出层：1个神经元

数值案例： 输入$X=\begin{bmatrix}1&2&3&0&1\\0&1&2&3&1\\1&0&1&2&0\\2&1&0&1&3\\1&2&3&0&2\end{bmatrix}$，输出：y=10

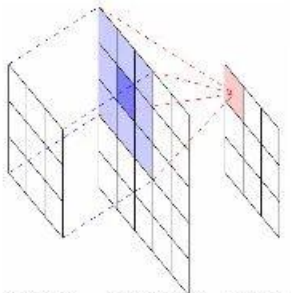
学习率： $\eta=0.01$

卷积层：

权重和偏置：

卷积核：$K=\begin{bmatrix}0.1&0.2&0.1\\0.0&0.1&0.2\\0.2&0.1&0.0\end{bmatrix}$

$b_2=0$



前向传播：

以左上角3*3区域为例

对应元素乘积后求和：

$$Z_{0,0}=\sum_{m=0}^{2}\sum_{n=0}^{2}K_{m,n}\cdot K_{i+m,j+n}+b$$
$$=(1\times0.1+2\times0.2+3\times0.1)$$
$$+(0\times0.0+1\times0.1+2\times0.2)$$
$$+(1\times0.2+0\times0.1+1\times0.0)$$
$$=(0.1+0.4+0.3)+(0+0.1+0.4)+(0.2+0+0)$$
$$=1.5$$

以上述方式逐个计算区域与卷积核的乘积之和，

卷积层输出：

$$Z_{conv}=\begin{bmatrix}1.5&1.7&1.3\\1.1&1.5&1.2\\0.7&1.3&1.8\end{bmatrix}$$

激活函数（ReLU）：$A_{conv}=max(0,Z_{conv})=Z_{conv}$

池化层：

取前2*2区域：

$$pooling\ regin$$
$$=max\begin{Bmatrix}Z_{2i,2j}&Z_{2i,2j+1}\\Z_{2i+1,2j}&Z_{2i+1,2j+1}\end{Bmatrix}$$
$$=max\begin{bmatrix}1.5&\mathbf{1.7}\\1.1&1.5\end{bmatrix}$$
$$\rightarrow A_{pool}=1.7$$

全连接层（ReLU）：

权重：$W_{fc}=0.5$

偏置：$b_{fc}=0.5$

$$Z_{fc}=W_{fc}\cdot A_{pool}+b_{fc}=0.5\times1.7+0.5=1.35$$
$$A_{fc}=max(0,Z_{fc})=1.35$$

输出层：

权重：$W_{out}=0.3$

偏置：$b_{out}=0$

$$Z_{out}=W_{out}\cdot A_{fc}+b_{out}=0.3\times1.35+0=0.405$$
$$\hat{y}=Z_{out}=0.405$$

卷积神经网络（CNN）

□ 模型通过局部连接和权值共享机制，能够高效提取图像局部特征，适用于机械制造中的视觉检测任务，如零件表面划痕检测、尺寸精度测量和装配完整性检查等。

损失函数：

$$L = (y - \hat{y})^2 = (10 - 0.405)^2 \approx 92.064$$

反向传播：

输出层：$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) = -19.19$

$$\frac{\partial L}{\partial W_{out}} = \frac{\partial L}{\partial \hat{y}} \cdot A_{fc} = -19.19 \times 0.85 = -16.3115$$

全连接层：$\frac{\partial L}{\partial Z_{fc}} = \frac{\partial L}{\partial A_{fc}} \cdot W_{out} = -19.19 \times 0.3 = -5.757$

$$\frac{\partial L}{\partial W_{fc}} = -5.757 \cdot A_{pool} = -5.757 \times 1.7 = -9.7869$$

回传至池化层：$\frac{\partial L}{\partial A_{conv[0,1]}} = -5.757 \times 0.5 = -2.8785$

回传至卷积核：$\frac{\partial L}{\partial K} = -2.8785 \cdot \begin{bmatrix} 2 & 3 & 0 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix}$

$$= \begin{bmatrix} -5.757 & -8.6355 & 0 \\ -2.8785 & -5.757 & -8.6355 \\ 0 & -2.8785 & -5.757 \end{bmatrix}$$

参数更新：

$$W_{out} = W_{out} - \eta \cdot \frac{\partial L}{\partial W_{out}} \leftarrow 0.3 - 0.01 \times (-16.5665) = 0.465665$$

$$W_{fc} = W_{fc} - \eta \cdot \frac{\partial L}{\partial W_{fc}} \leftarrow 0.5 - 0.01 \times (-9.9399) = 0.599399$$

$$K_{new} = \begin{bmatrix} 0.1 & 0.2 & 0.1 \\ 0.0 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.0 \end{bmatrix} - 0.01 \cdot \begin{bmatrix} -5.757 & -8.6355 & 0 \\ -2.8785 & -5.757 & -8.6355 \\ 0 & -2.8785 & -5.757 \end{bmatrix}$$

$$= \begin{bmatrix} 0.15757 & 0.286355 & 0.1 \\ 0.028785 & 0.15757 & 0.286355 \\ 0.2 & 0.129785 & 0.05757 \end{bmatrix}$$

计算结果：

对上述过程重复100次，输出$\hat{y}$和MSE

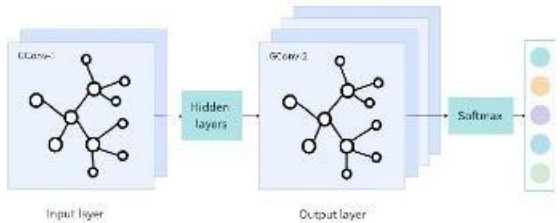
迭代次数	输出$\hat{y}$	MSE
1	0.405	94.965
2	0.3448	93.24
...	...	...
100	9.547	0.103

图神经网络（GNN）

□ 图神经网络（Graph Neural Network）突破传统CNN对规则数据的限制，通过图结构直接建模机械系统的复杂物理关系，成为智能机械研究的新范式。

图神经网络（GNN）

□ 模型擅长处理拓扑关系数据，适用于机械系统故障传播分析等复杂系统分析。



模型架构：输入为图$G=(V,E)$，共3个节点，每个节点输入为一个2维向量
一层图卷积层，一层全连接输出层

节点特征输入：$X=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1&2\\0&1\\3&1\end{bmatrix}$

输出：y=10

邻接矩阵：$A=\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}$

GCN结构：$H=AXW_1+b_1$

初始参数：图卷积权重$W_1=\begin{bmatrix}0.1&0.2\\0.3&0.4\end{bmatrix}$，
偏置$b_1=[0.0\quad 0.0]^T$
回归层权重$W_{fc}=[0.5\quad 0.6]$，偏置$b_{fc}=0$

学习率：$\eta=0.01$

模型架构

前向传播（以节点2为例）：

节点2的邻居是$\{v_1\quad v_3\}$，对应的特征是$\{x_1\quad x_3\}$，

计算邻接传播：$AX=\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}\begin{bmatrix}1&2\\0&1\\3&1\end{bmatrix}=\begin{bmatrix}0&1\\4&3\\0&1\end{bmatrix}$

图卷积输出：$H=AXW_1+b_1=\begin{bmatrix}0&1\\4&3\\0&1\end{bmatrix}\begin{bmatrix}0.1&0.2\\0.3&0.4\end{bmatrix}=\begin{bmatrix}0.3&0.4\\1.3&2.0\\0.3&0.4\end{bmatrix}$

全图平均池化（取所有节点的平均）：

$$\bar{H}=\frac{1}{3}\sum_{i=1}^{3}H_i=\frac{1}{3}[0.3+1.3+0.3\quad 0.4+2.0+0.4]=[0.6333\quad 0.9333]$$

输出层：$\hat{y}=W_{fc}\cdot\bar{H}+b_{fc}=[0.5\quad 0.6]\cdot\begin{bmatrix}0.6333\\0.9333\end{bmatrix}=0.8767$

损失函数：

$L=(y-\hat{y})^2=(10-0.8767)^2=83.007$

反向传播：

输出层一阶导数：$\frac{\partial L}{\partial\hat{y}}=\hat{y}-y=0.8767-10=-9.1233$

输出层权重梯度：$\frac{\partial L}{\partial w_{fc}}=\frac{\partial L}{\partial\hat{y}}\cdot\frac{\partial\hat{y}}{\partial w_{fc}}=-9.1233\cdot\bar{H}$

$=-9.1233\cdot[0.6333\quad 0.9333]=[-5.7787\quad -8.5145]$

模型求解过程

图神经网络（GNN）

□ 模型擅长处理拓扑关系数据，适用于机械系统故障传播分析等复杂系统分析。

回归层偏置梯度：

$$\frac{\partial L}{\partial b_{fc}} = \frac{\partial L}{\partial \hat{y}} = -9.1233$$

更新回归层参数：

$$W_{fc} \leftarrow W_{fc} - \eta \cdot \frac{\partial L}{\partial W_{fc}}$$

$$= [0.5 \quad 0.6] - 0.01 \cdot [-5.7787 \quad -8.5145]$$

$$= [0.5806 \quad 0.7934]$$

$$b_{fc} \leftarrow b_{fc} - \eta \cdot \frac{\partial L}{\partial b_{fc}}$$

$$= 0 - 0.01 \times (-9.1233) = 0.0912$$

回归层输出：

$$\hat{y} = W_{fc} \cdot \bar{H} + b_{fc}$$

$$b_{fc} = 0$$

$$\rightarrow \frac{\partial L}{\partial \bar{H}} = \frac{\partial L}{\partial \hat{y}} \cdot W_{fc} = -9.1233 \times [0.5 \quad 0.6] = [-4.5617 \quad -5.4740]$$

平均池化是对3个节点取平均：

$$\frac{\partial L}{\partial H_i} = \frac{1}{3} \cdot \frac{\partial L}{\partial \bar{H}} = \frac{1}{3} \cdot [-4.5617 \quad -5.4740] = [-1.5206 \quad -1.8247]$$

参数更新：

令损失对图卷积权重W_1的梯度为：

$$\frac{\partial L}{\partial W_1} = (AX)^T \cdot \frac{\partial L}{\partial H} = \begin{bmatrix} 0 & 4 & 0 \\ 1 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1.5206 & -1.8247 \\ -1.5206 & -1.8247 \\ -1.5206 & -1.8247 \end{bmatrix} = \begin{bmatrix} -6.0825 & -7.2988 \\ -6.0825 & -9.1235 \end{bmatrix}$$

更新图卷积权重

$$W_1 \leftarrow W_1 - \eta \cdot \frac{\partial L}{\partial W_1}$$

$$= \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} + 0.01 \times \begin{bmatrix} 6.0825 & 7.2988 \\ 6.0825 & 9.1235 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1608 & 0.2729 \\ 0.3608 & 0.4912 \end{bmatrix}$$

计算结果：

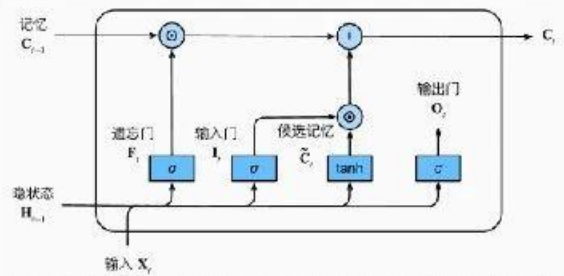
对上述过程重复100次，输出$\hat{y}$和MSE

迭代次数	输出$\hat{y}$	MSE
1	0.8767	83.007
2	2.5347	55.773
…	…	…
100	≈ 10.0	$<10^{-5}$

模型求解过程

长短期记忆网络(LSTM)

□ 模型擅长处理时序依赖，适用于信号分析、寿命预测等时间序列数据处理任务。



模型架构： 输入：2个神经元（向量$x_t \in R^2$）
隐状态维度：1（即$h_t \in R$）
仅1个LSTM单元（单时间步）
输出层：1个神经元，线性输出
激活函数：$sigmoid: \sigma(z) = \frac{1}{1+e^{-z}}$

$$sigmoid: \sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

卷积层：1个3*3卷积核，步长1，
输出大小3*3（ReLU激活函数）
池化层：2*2最大池化，步长2，输出1*1
全连接层：1个隐藏神经元，ReLU
输出层：1个神经元

数值案例： 输入x = $\begin{bmatrix}1.0\\2.0\end{bmatrix}$，目标输出：y=10

学习率： $\eta = 0.01$

初始参数：

遗忘门	W_f	[0.1 0.2]	U_f	[0.3]	b_f	0.1
输入门	W_i	[0.2 0.3]	U_i	[0.1]	b_i	0.1
候选状态	W_c	[0.1 0.2]	U_c	[0.0]	b_c	0.0
输出门	W_0	[0.2 0.1]	U_0	[0.2]	b_0	0.1
输出层	W_y	[0.5]		b_y	0	

前向传播：

遗忘门：$W_f x_t = 0.1 \times 1 + 0.2 \times 2 = 0.5$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$= \sigma(0.5 + 0 + 0.1) = \sigma(0.6) \approx 0.645$$

输入门：$W_i x_t = 0.2 \times 1 + 0.3 \times 2 = 0.8$

$$U_i h_{t-1} = 0$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$= \sigma(0.8 + 0 + 0.1) = \sigma(0.9) \approx 0.711$$

候选记忆：$W_c x_t = 0.1 \times 1 + 0.2 \times 2 = 0.5$

$$U_c h_{t-1} = 0$$
$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$= tanh(0.5) \approx 0.462$$

更新细胞状态：$C_t = f_t \cdot C_{t-1} = 0.645 \times 0 + 0.711 \times 0.462 \approx 0.329$

输出门：$W_o x_t = 0.2 \times 1 + 0.1 \times 2 = 0.4$

$$U_o h_{t-1} = 0$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) = \sigma(0.5) \approx 0.622$$

隐藏状态：$tanh(C_t) = tanh(0.329) \approx 0.318$

$$h_t = o_t \cdot tanh(C_t) = 0.622 \times 0.318 = 0.198$$

隐藏状态：$\hat{y}_t = W_o x_t + b_y = 0.5 \times 0.198 + 0 = 0.099$

损失函数：

遗忘门：$L = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(10 - 0.099)^2 \approx 49.01$

反向传播：

输出层误差：$\delta_y = y - \hat{y} = 0.099 - 10 = -9.901$

输出层权重梯度：$\frac{\partial L}{\partial w_y} = \delta_y \cdot h_t = -0.099 \times 0.198 = -0.0196$

$$\frac{\partial L}{\partial b_y} = \delta_y = -9.901$$

误差回传到隐藏层状态：$\delta_h = \delta_y \cdot W_y = -9.901 \times 0.5 = -4.95$

输出门梯度：$\frac{\partial o_t}{\partial z} = o_t \cdot (1 - o_t) = 0.622 \times (1 - 0.622) = 0.235$

$$\delta_o = \delta_h \cdot tanh(C_t) \cdot \frac{\partial o_t}{\partial z} = -4.95 \times 0.318 \times 0.235 = -0.3699$$

长短期记忆网络(LSTM)

□ 模型擅长处理时序依赖，适用于信号分析、寿命预测等时间序列数据处理任务。

记忆状态C_t的梯度：

$h_t = o_t \cdot tanh(C_t)$

$\frac{\partial h_t}{\partial C_t} = o_t \cdot \left(1 - tanh^2(C_t)\right) = 0.622 \times (1 - 0.318^2) = 0.5591$

$\delta_{C_t} = \delta_h \cdot \frac{\partial h_t}{\partial C_t} = -4.95 \times 0.5591 = -2.77$

输入门i_t和候选记忆$\tilde{C}_t$梯度：

$C_t = i_t \cdot \tilde{C}_t + f_t \cdot C_{t-1}$

$\frac{\partial C_t}{\partial i_t} = \tilde{C}_t = 0.4621$

$\frac{\partial L}{\partial i_t} = \delta_{C_t} \cdot \tilde{C}_t = -2.77 \times 0.4621 = -1.28$

sigmoid：$\frac{\partial i_t}{\partial z_i} = i_t \cdot (1 - i_t) = 0.711 \times (1 - 0.711) = 0.205$

$\delta_{z_i} = -1.28 \times 0.205 = -0.2628$

候选记忆：$\frac{\partial C_t}{\partial \tilde{C}_t} = i_t = 0.711$

$\frac{\partial L}{\partial \tilde{C}_t} = \delta_{C_t} \cdot i_t = -2.77 \times 0.711 = -1.97$

$\frac{\partial \tilde{C}_t}{\partial z_{\tilde{C}}} = 1 - tanh^2(z_{\tilde{C}}) = 1 - tanh^2(0.5) = 0.7864$

$\delta_{z_{\tilde{C}}} = -1.97 \times 0.7864 = 1.549$

遗忘门f_t：$\frac{\partial C_t}{\partial f_t} = C_{t-1} = 0 \rightarrow \delta_{z_f} = 0$

参数更新：$\frac{\partial L}{\partial W_0} = \delta_{z_0} \cdot x = [-0.3688 \quad -0.7376]$

$\frac{\partial L}{\partial U_0} = \delta_{z_0} \cdot h_{t-1} = 0 \qquad \frac{\partial L}{\partial b_0} = \delta_{z_0} = -0.3688$

$W_0^{(new)} = W_0 - \eta \cdot \frac{\partial L}{\partial W_0} = [0.2 \quad 0.1] + 0.01 \times [-0.3688 \quad -0.7376] = [0.2037 \quad 0.1074]$

$U_0^{(new)} = U_0 - 0 = [0.1]$

$b_0^{(new)} = 0 + 0.01 \times 0.3688 = 0.0037$

输入门权重更新：$\frac{\partial L}{\partial W_i} = \delta_{z_i} \cdot x = [-0.2628 \quad -0.5256]$

$W_i^{(new)} = [0.3 \quad 0.4] + 0.01 \times [0.2628 \quad 0.5256] \qquad b_i^{(new)} = 0 + 0.01 \times 0.2628 = 0.0026$

候选记忆门权重更新：$\frac{\partial L}{\partial W_C} = \delta_{z_C} \cdot x = [-1.5499 \quad -3.0998]$

$W_C^{(new)} = [0.2 \quad 0.1] + 0.01 \times [1.5499 \quad 3.0998] \quad b_C^{(new)} = 0 + 0.01 \times 1.5499 = 0.0155$

遗忘门梯度为0：$\delta_{z_f} = 0$，所有的权重保持不变

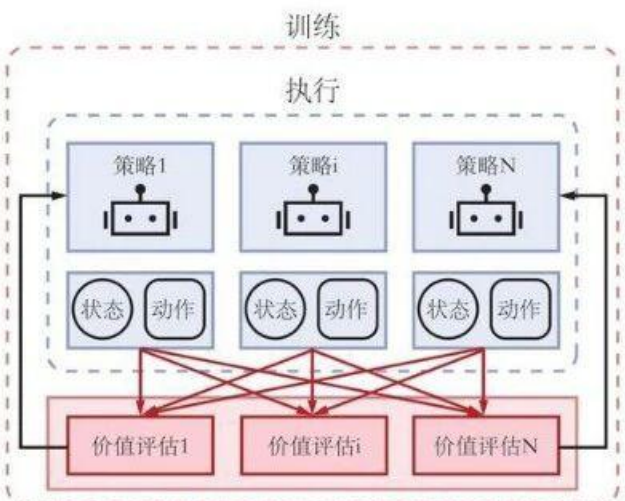
计算结果：

对上述过程重复100次，输出$\hat{y}$和MSE

迭代次数	输出$\hat{y}$	MSE
1	0.099	98.02
2	0.14	96.02
...	...	...
100	9.5	0.25

强化学习（RL）

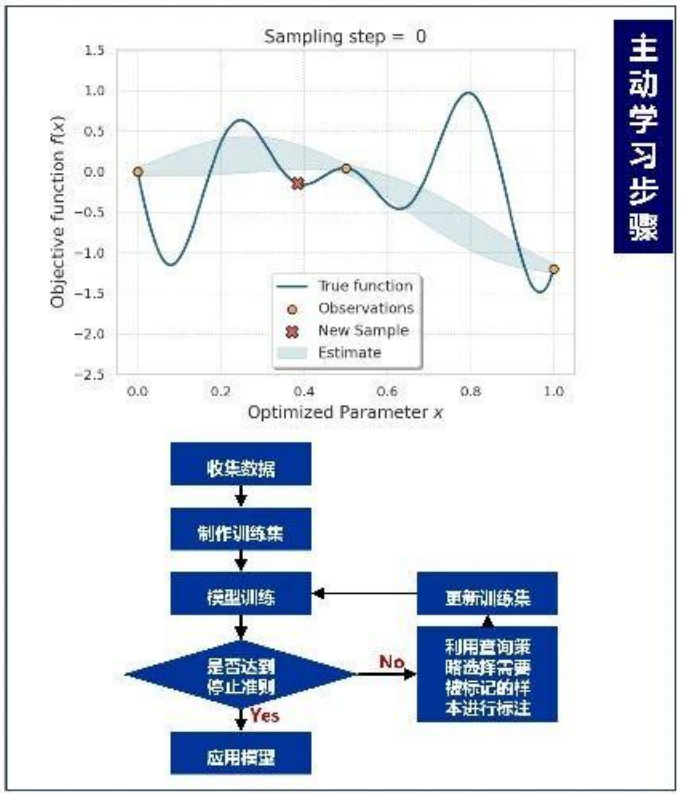
- 用于智能体在与环境不断交互的过程中，通过持续学习和优化决策策略，实现长期累积回报最大化或完成特定复杂任务目标的一类动态、反馈驱动的决策优化问题。



	强化学习	神经网络（深/浅）
目标函数	最大化长期累计回报	最小化监督损失（如 MSE）
决策单位	决策动作（从策略中采样或确定性选择）	对输入进行映射预测（如分类/回归）
学习对象	学习策略（policy）或价值函数	学习一个静态函数映射关系
反馈类型	动态奖励信号（reward）	静态标签（ground truth）
与环境的交互	与环境持续交互产生数据	无交互，单次前向预测
损失函数来源	来自时序差分误差或策略梯度	由监督信号直接定义
采样机制	主动选择动作以影响环境状态	被动接收输入样本

主动学习（AL）

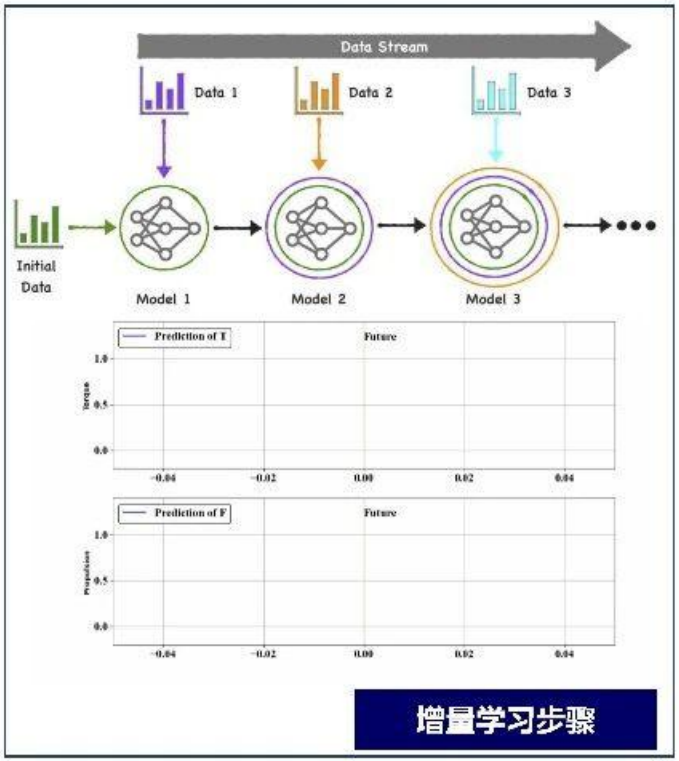
□ 智能选择高价值样本进行标注，适用于标注成本高的场景，显著提升数据利用效率。



	主动学习	神经网络（深/浅）
学习目标定位	学习“选择哪些样本进行学习”以最大化收益	学习“如何从已知样本中拟合输入输出映射”
是否主动选择样本	是（如不确定性采样、熵采样等）	否（训练集固定）
训练流程结构	迭代性地构建训练集，每轮训练后动态选样	一次性训练已有数据
样本利用效率目标	用尽可能少的样本达到尽可能好结果	倾向于用尽可能多的样本获得泛化性能
损失函数构造	在主动选择前使用损失或不确定度指标（如熵、Margin）评估样本	直接最小化训练集的均方误差/交叉熵等
是否需要人参与标注	是，主动学习需人提供选择样本的标签（专家在环）	否，训练数据预先准备好
模型结构依赖性	可以搭配任意模型（神经网络、SVM、树等）	是具体结构（感知器、CNN、LSTM等）
对数据池的假设	假设存在大量未标注数据池供选择	假设训练集已标注完备

增量学习（IL）

□ 模型支持在线更新，无需全量数据重训练，适用于数据流场景，动态适应新数据分布。



增量学习步骤

	增量学习	神经网络（深/浅）
目标函数	连续学习，保持性能与稳定性	一次性训练，最小化损失
决策单位	不断更新模型同时保留旧知识	批量训练后做预测
学习对象	流式数据或阶段性任务	静态数据上的单次学习
反馈类型	每次增量接收一批标签或少量反馈	统一训练集标签
与环境的交互	不断接收新数据，需适应分布漂移	一次性训练，无需适应
损失函数来源	与历史保持一致性的正则项	来自批量训练数据
采样机制	训练顺序重要（可致灾难遗忘）	随机采样批次训练

计算复杂度

□ 衡量算法的计算效率，帮助评估模型在数据规模增大时的可扩展性，指导资源分配，优化模型设计（如层数、参数量），并在实际部署中平衡速度与精度。

模型	计算复杂度公式	说明
多项式拟合（最小二乘）	$O(nd^2 + d^3)$	n 为样本数，d为特征数
浅层神经网络	$O(Tn(dh + hk))$	输入维度d，隐藏神经元数h，输出维度k
深度学习神经网络	$O\left(Tn\sum_{l=1}^{L} h_{l-1}h_l\right)$	L层网络，第l层神经元数h_l，n为样本数，T是迭代步
卷积神经网络	$O\left(Tn\sum_{l=1}^{L} H_l' W_l' C_l K_l^2 M_l\right)$	$H \times W \times C$输出尺寸(高×宽×通道数)，卷积核尺寸K × K，数量M，L层I迭代步
图神经网络	$O(TnL(Eh + Nh^2))$	N为节点数，E为边数，节点特征维度d，隐层维度h，L层T迭代步
LSTM	$O(T_{iter}Tnh(d + h))$	输入维度d，隐藏状态维度h，n样本，T_{iter}迭代步

AI的分类方式：以参数规模为例

□ AI模型可以根据模型的参数规模进行分类，以互联网模型和数据为例：

大模型

- **参数量：**十亿~万亿级
- **灵活性：**通用性强
- **训练成本：**极高
- **推理速度：**慢（需云端计算）
- **典型应用：**多模态生成任务
  通用智能任务

小模型

- **参数量：**百万~数亿级
- **灵活性：**专用性强
- **训练成本：**低
- **推理速度：**快（可本地部署）
- **典型应用：**垂直领域优化任务
  专用系统

微模型

- **参数量：**百万以下
- **灵活性：**高度定制化
- **训练成本：**中等（需领域数据）
- **推理速度：**取决于基础模型
- **典型应用：**边缘侧实时控制
  低功耗嵌入式系统

Bilibili每天产生的数据量

数据量：约7.8 PB
视频播放：51 TB
视频投稿：7.17 PB
互动数据：4.5 TB
用户行为日志：10.4 TB

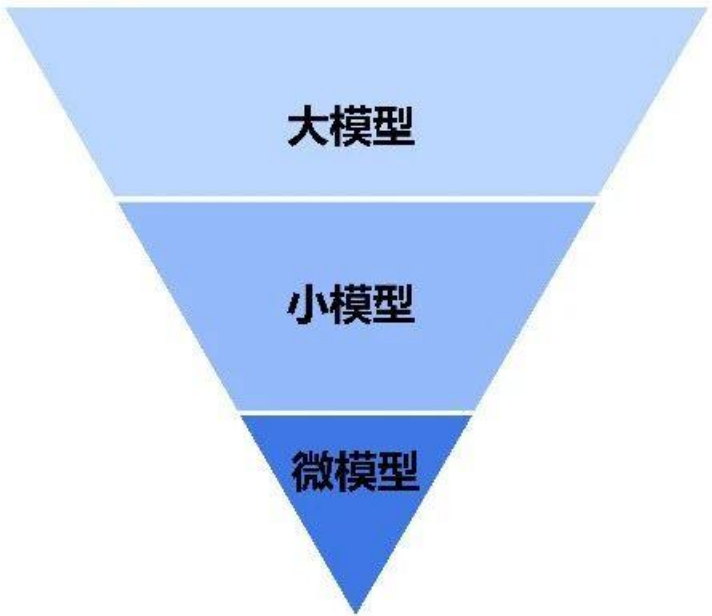
智能汽车工厂每天产生的数据量

数据量：约2.0 TB
生产数据：750 GB(传感器数据、生产流数据)
供应链数据：200 GB
能源数据：50 GB
AI检测数据：1 TB

互联网平台与智能工厂的数据量级比约为 10^3~10^4 : 1，呈现显著的数量级差

AI4M：大模型、小模型、微模型

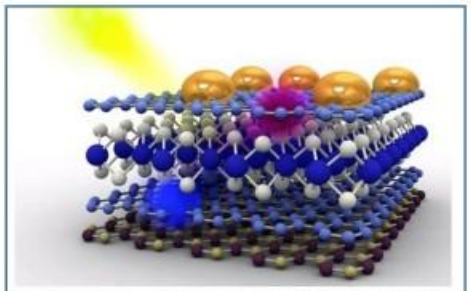
□ 根据AI模型的参数量级，可以将先进制造中的模型分为：大、小、微



模型类型	参数量级	应用场景	特点
大模型	>10M	智能工厂、多机协同...	泛化强、资源需求高
小模型	10k~10M	装备监测、数字孪生...	响应快、适配性好
微模型	<10k	优化设计、性能分析...	极低延迟、轻量部署

AI4M：材料、设计、制造、运维

工程材料、设计优化、加工装配、控制运维是重大装备先进制造的四个主要阶段



纳米材料/复合材料

材料

工业设计/优化设计

设计

精密制造/3D打印

制造

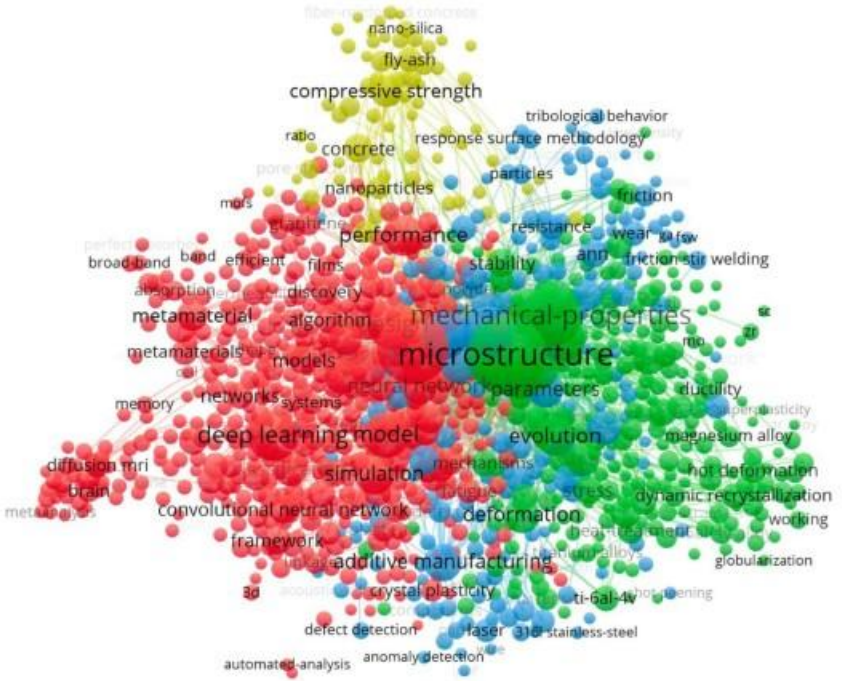
设备监控/故障诊断

运维

AI4M统计：工程材料

□ 对“AI+工程材料”相关关键词进行检索，并绘制统计图和关键词共现图谱

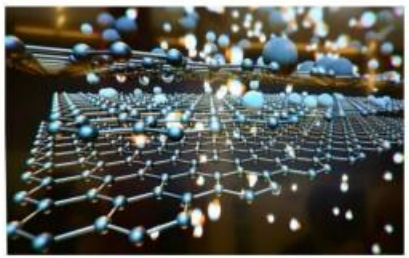
检索式： TS=（“artificial intelligence” OR “machine learning” OR “deep learning” OR “neural network” OR “data-driven”）AND TS=("material design" OR "composite material" OR "material discovery" OR "microstructure"OR"metamaterial"OR"material property")

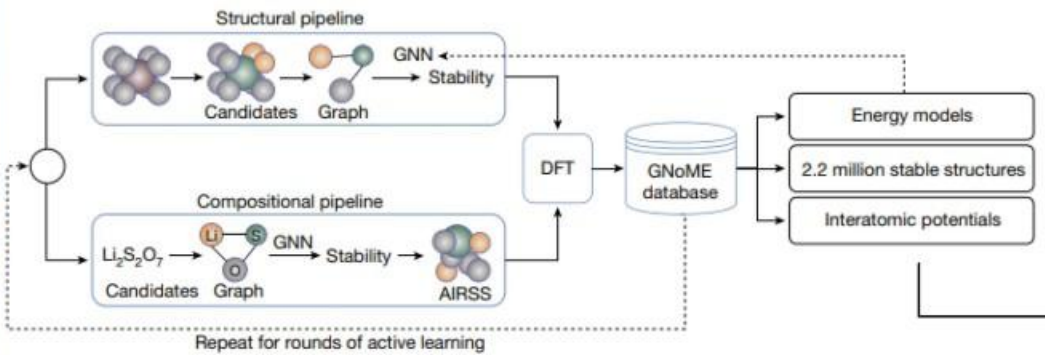


关键词共现图谱

AI4M案例：工程材料

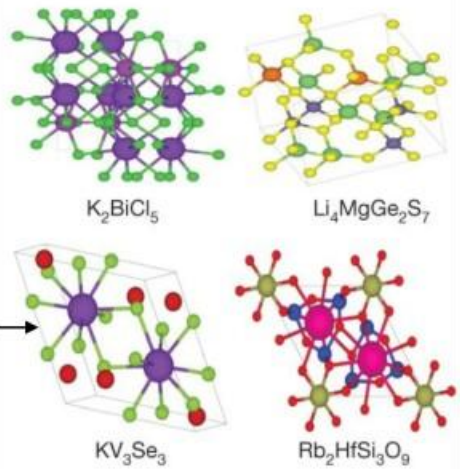
□ **大模型应用于新材料发现：DeepMind团队提出的GNOME图神经网络，快速发现220万个新的材料晶体结构，将材料发现的效率提高了一个数量级。**





模型：GNOME图神经网络　参数量：15M
效果：将材料发现的效率提高了一个数量级

大规模图网络



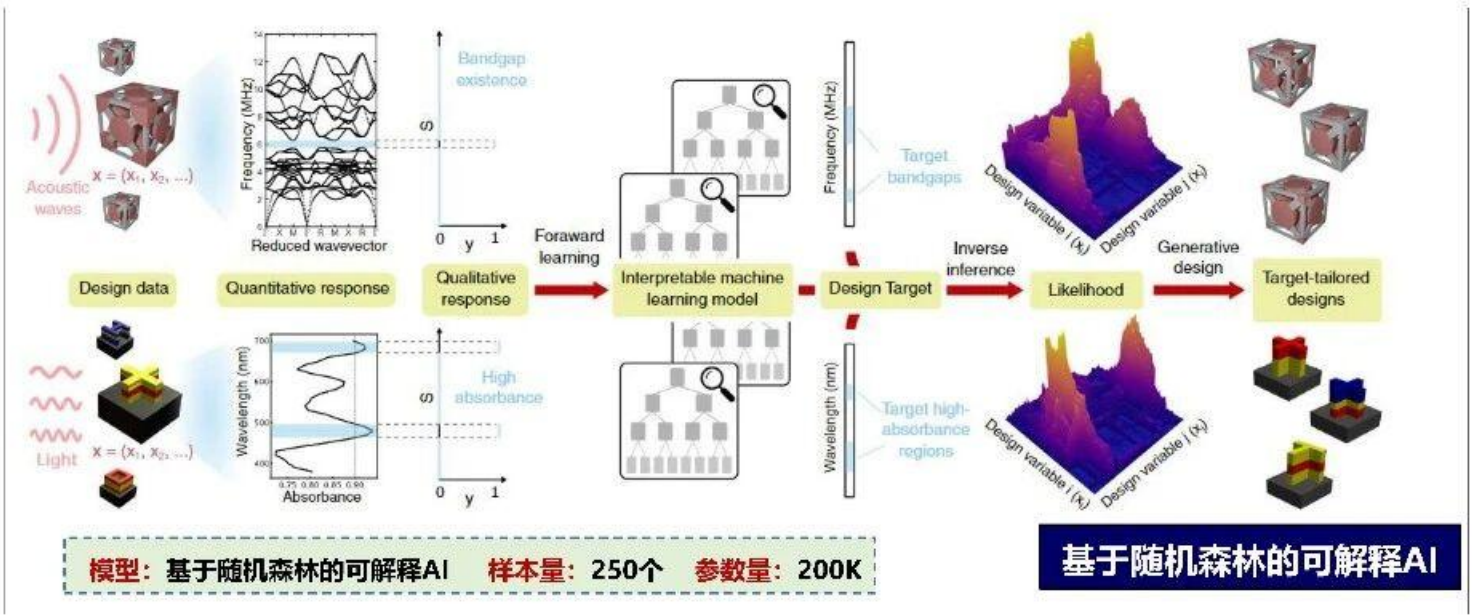
模型预测的新材料结构

Merchant A, et al. Scaling deep learning for materials discovery. **Nature**,2023, 624(7990): 80-85.

AI4M案例：工程材料

□ **小模型应用于超材料设计：通过可解释的随机森林微模型，实现超材料的快速设计，发现更多可行解和可行结构，大幅缩短工程材料设计的研发周期。**

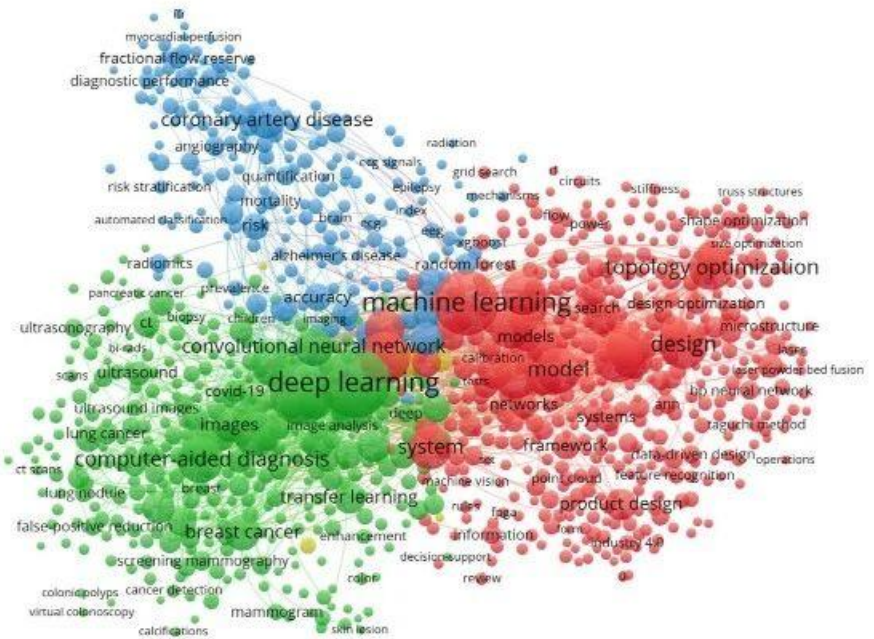
美国双院院士
Wei Chen



Chen W, et al. Generative inverse design of metamaterials with functional responses by interpretable learning. Advanced Intelligent Systems,2024: 2400611.

AI4M统计：设计优化

□ 对“AI+设计优化”相关关键词进行检索，并绘制统计图和网络共现图谱

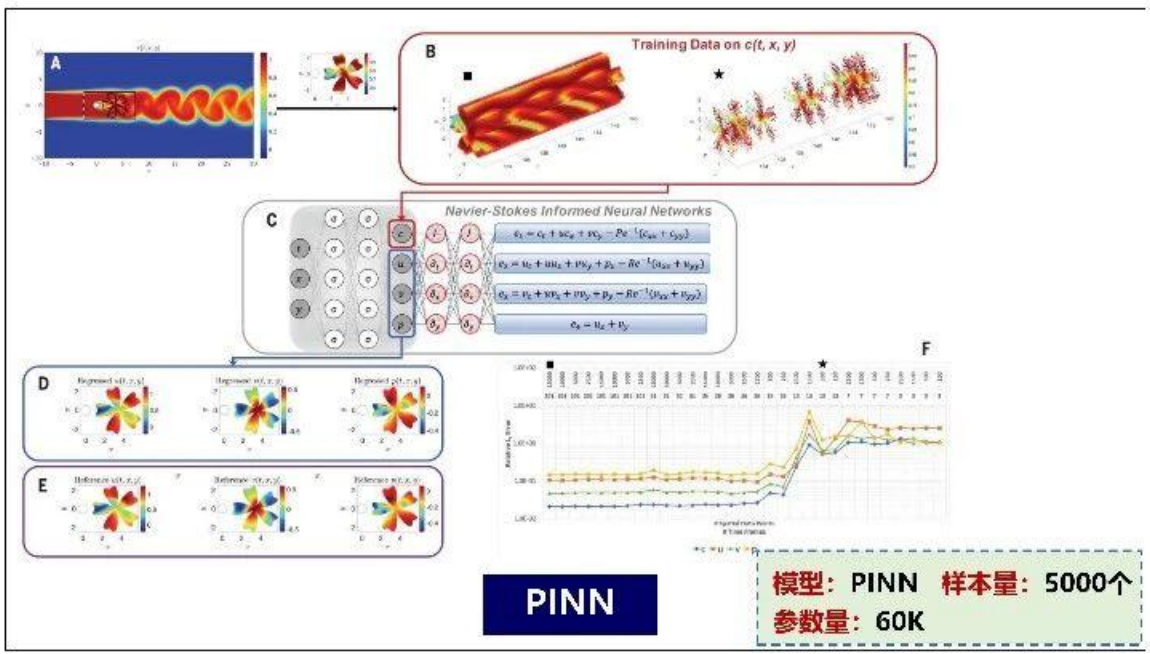


关键词共现图谱

检索式： TS=("artificial intelligence" OR "machine learning" OR "deep learning" OR "neural network"OR"data-driven") AND TS=("mechanical design" OR "topology optimization" OR "CAD" OR "structural optimization"OR"parameter optimization"OR"product design")

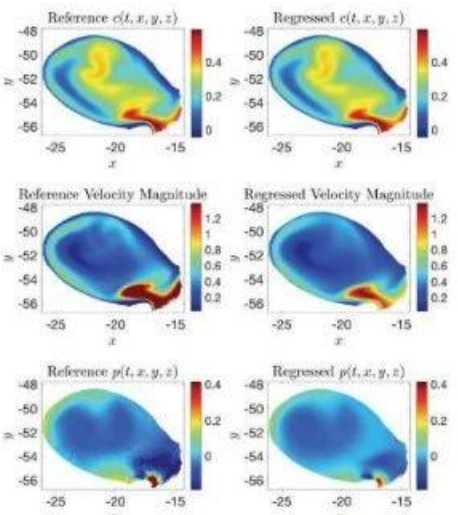
AI4M案例：设计优化

□ **小模型应用于流场重构：结合流场可视化数据和Navier-Stokes方程，通过PINN小模型从5000个仿真云图中直接推断出流体速度场和压力场。**



PINN

模型：PINN 样本量：5000个
参数量：60K



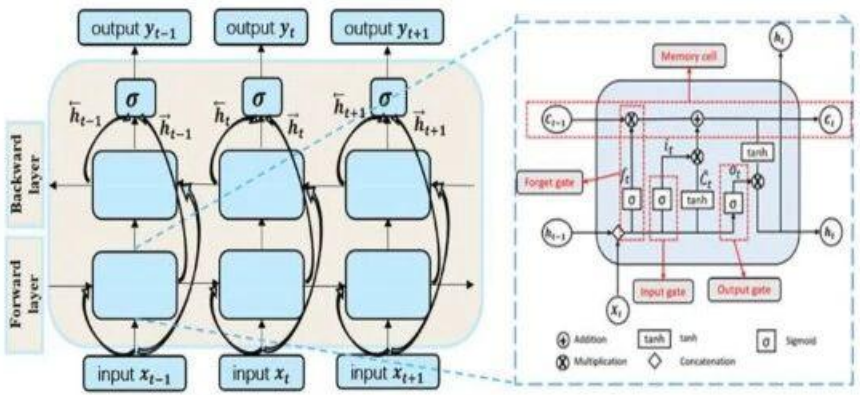
预测结果

Raissi M, et al. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. **Science**,2020, 367(6481): 1026-1030.

AI4M案例：设计优化

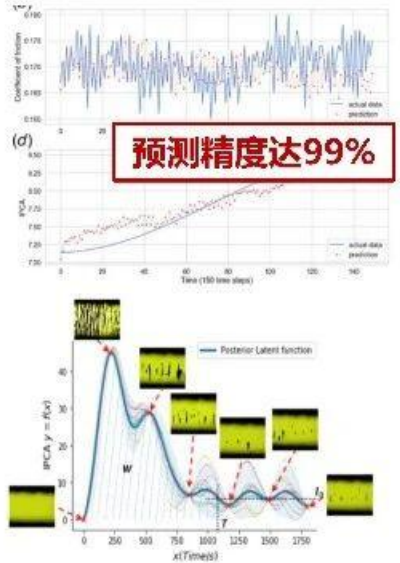
□ 小模型用于摩擦与润滑预测：针对在线铁谱监测磨损存在延迟、预测精度低问题，基于CSTS数据集构建方法及Bi-LSTM模型，实现摩擦系数及IPCA预测。

数据采集及数据集构建



模型：Bi-LSTM；　参数量：2M
算法效果：较传统方法（如MLP、Dilated CNN等）的均方误差降低50%以上，有效捕捉磨合期的高波动磨损特征

训练深度学习模型Bi-LSTM

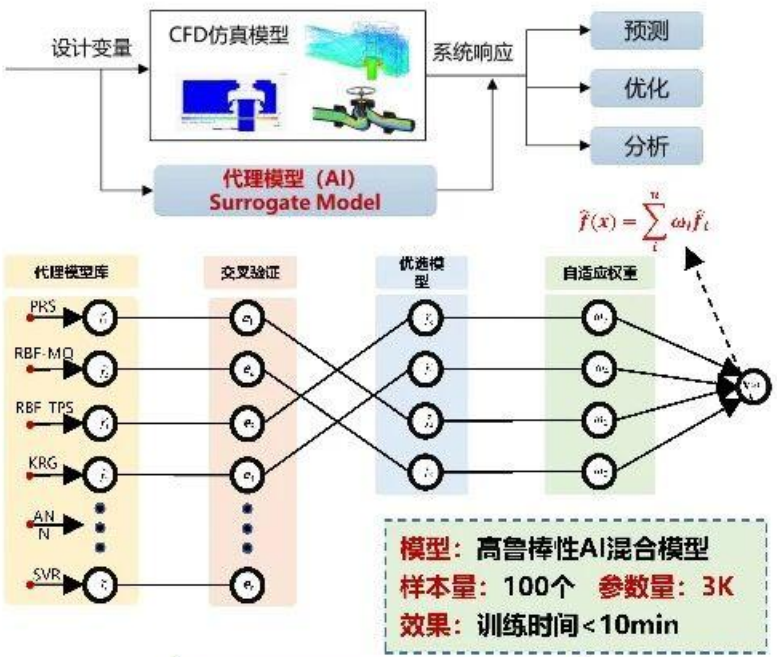


摩擦系数及IPCA实时预测

Zhang Zhinan*, et al. Short-Term Cross-Sectional Time-Series Wear Prediction by Deep Learning Approaches. Journal of Computing and Information Science in Engineering, 2022, 23(021007).

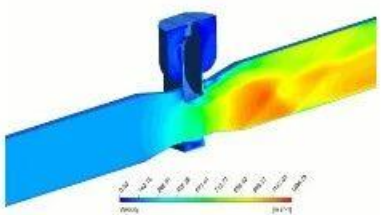
AI4M案例：设计优化

□ **微模型用于核电阀门的性能预测：基于逐点概率分布的高鲁棒性AI混合模型技术，对核电阀门关键性能预测精度＞95%，可以替代初始CFD模型分析。**



高鲁棒性AI混合模型

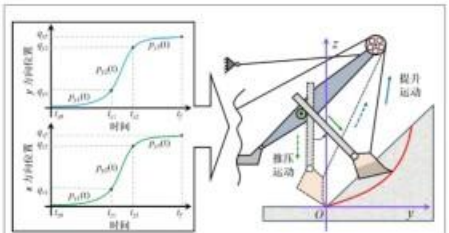
阀门优化设计



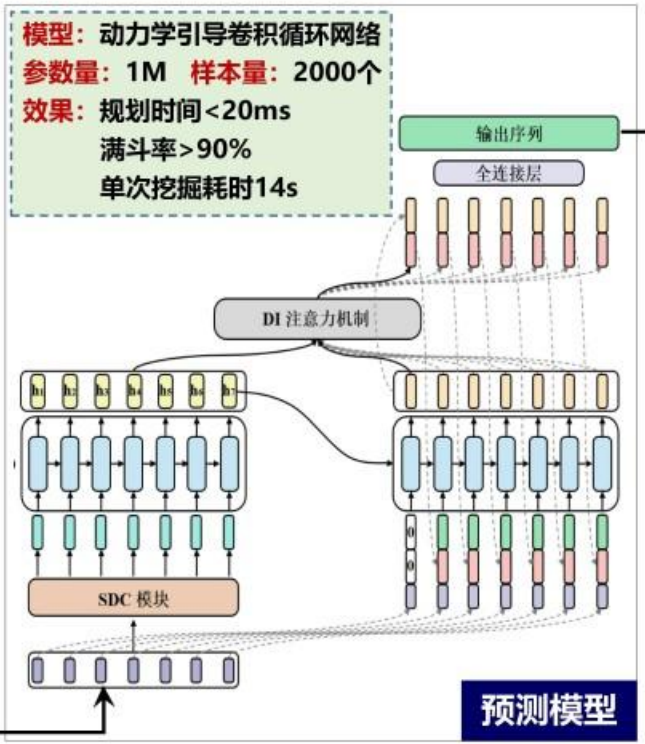
阀门动态结果

AI4M案例：设计优化

□ 小模型用于大型工程机械轨迹优化： AI小模型精度超过离散元方法（DEM），耗时仅为其1%，基于此优化了全球最大的矿用电铲的挖掘轨迹，单斗能力≥150吨。



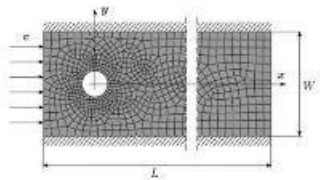
输入数据

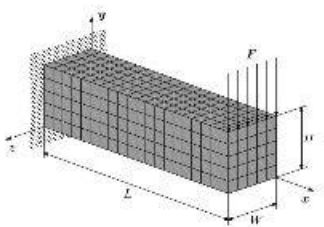


预测模型

AI4M案例：设计优化

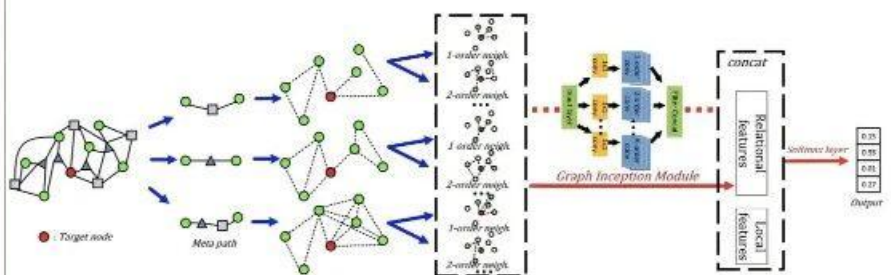
□ **小模型应用于结构应力场和流场重构：** 通过全局信息引导的图神经网络重构物理场信息，实现结构应力场或流场特征的快速重构，精度≥ 99%。



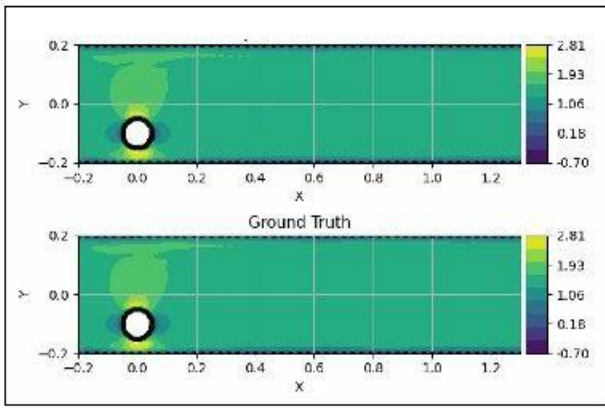


输入数据

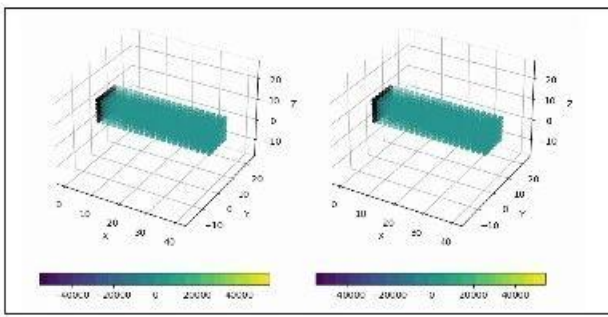
模型：图神经网络模型
样本量：500个　　参数量：100K
效果：耗时约为传统方法的3%；精度达到99%以上



图神经网络模型



流场重构

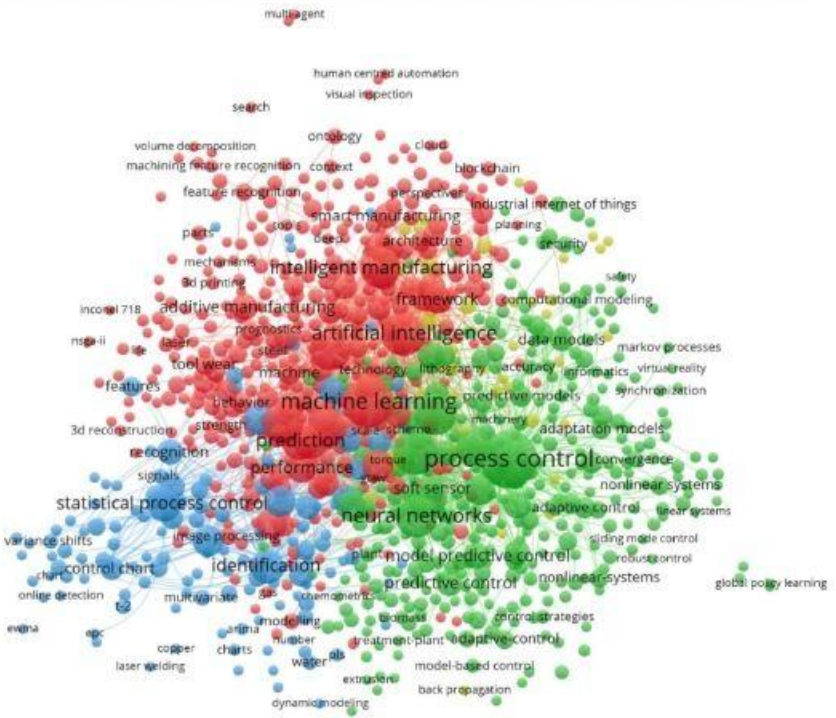


应力重构

AI4M统计：加工装配

□ 对“AI+加工装配”相关关键词进行检索，并绘制统计图和关键词共现图谱

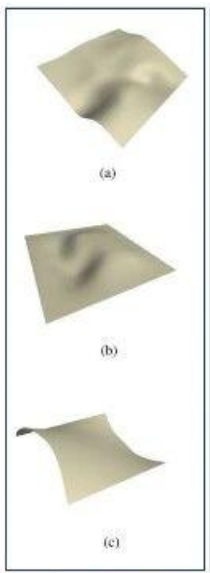
检索式： TS=("artificial intelligence" OR "machine learning" OR "deep learning" OR "neural network"OR"data-driven") AND TS=(“mechanical manufacturing” OR “process planning” OR “process control” OR “machining parameter” OR “intelligent manufacturing” OR “manufacturing process”)

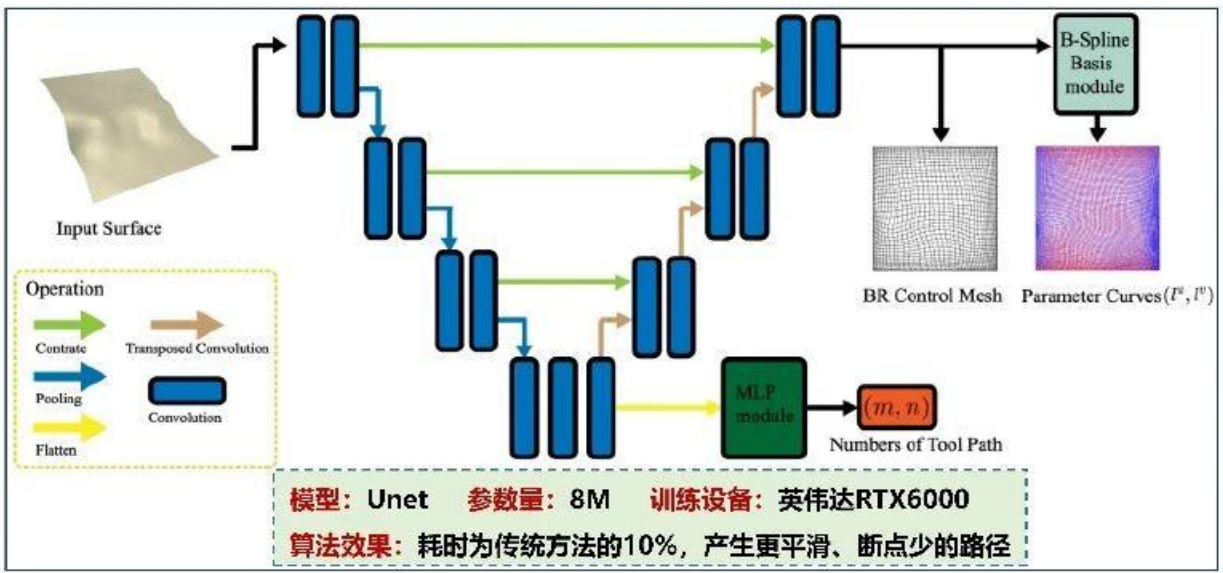


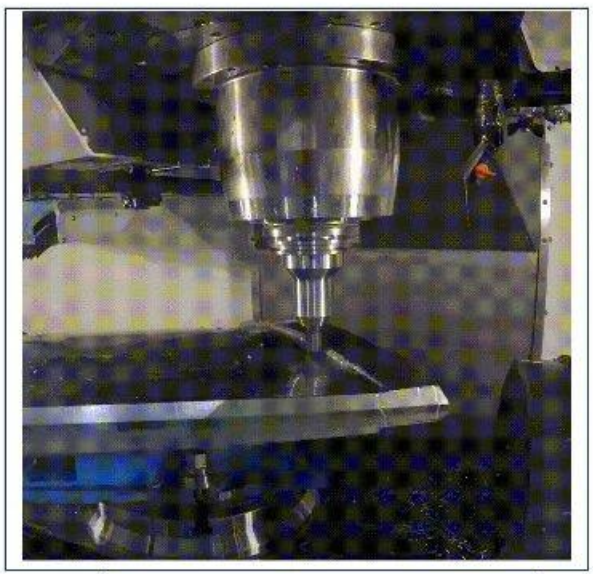
关键词共现图谱

AI4M案例：加工装配

□ **小模型应用于加工制造过程CNC刀具路径规划：基于BRNet神经网络生成刀具最优路径，实现毫秒级的B样条曲面减材铣削实时路径规划。**





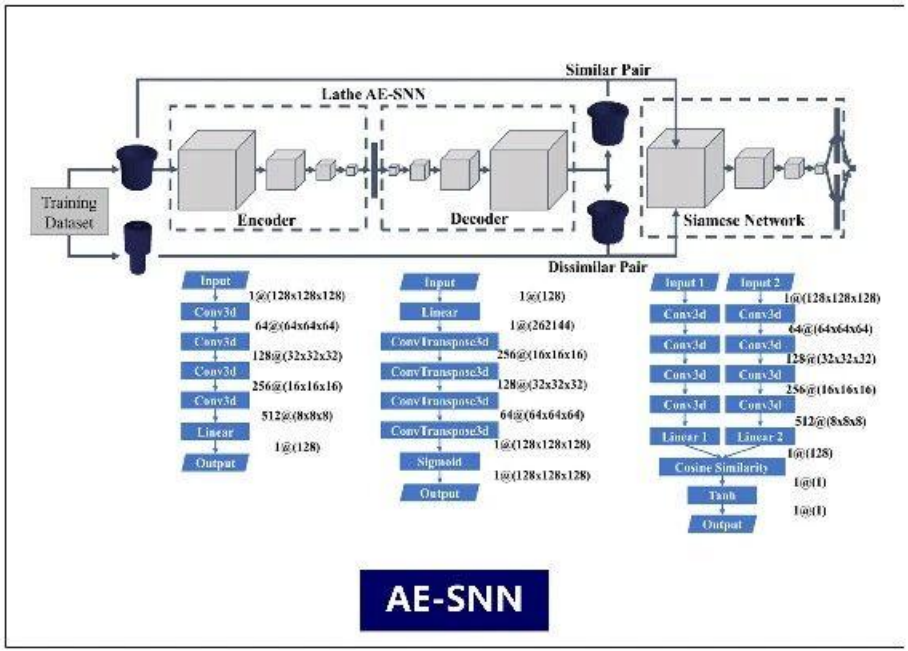


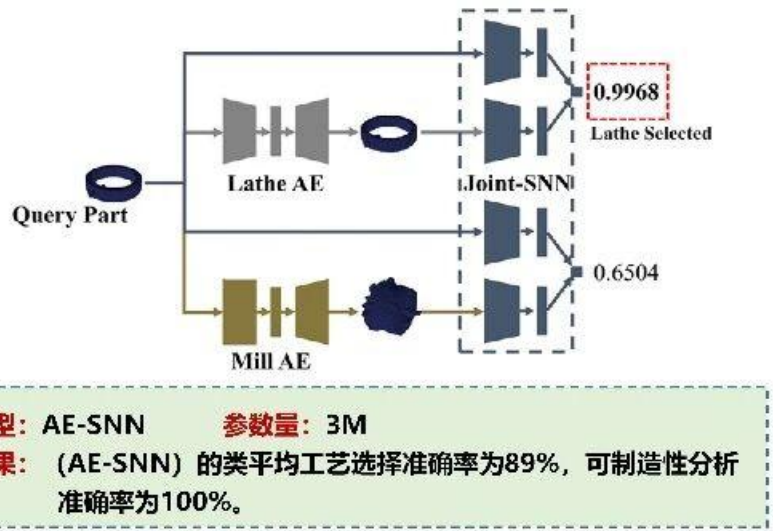
建立数据集　　训练深度学习模型BRNet　　AI实时生成CNC刀具路径

Feng Y F et al. Real-Time Tool-Path Planning Using Deep Learning for Subtractive Manufacturing. IEEE Transactions on Industrial Informatics,2024,20(4):5979-5988.

AI4M案例：加工装配

□ **小模型应用于可制造性分析与加工工艺选择：孪生神经网络与基于 Autoen-coder 的深度生成加工作模型集成，实现查询零件形状与采样输出的自动比较。**

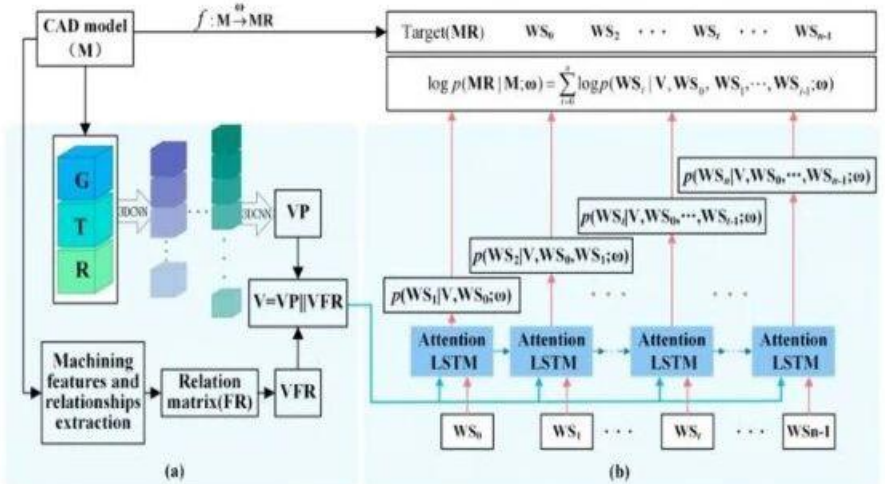




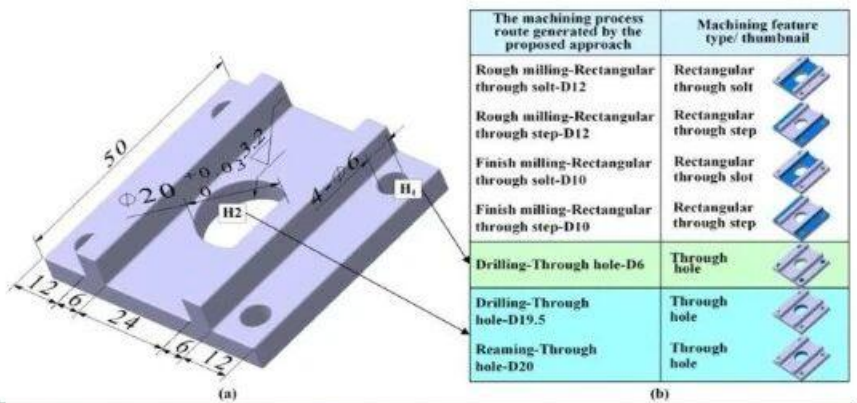
Melkote S, et al. Automated manufacturability analysis and machining process selection using deep generative model and Siamese neural networks . **Journal of Manufacturing Processes**,2023, 67: 57-67.

AI4M案例：加工装配

□ **小模型应用于智能制造刀具工作路径的规划：构建了3DCNN将结构几何特征、技术需求和加工特征转化为高阶向量，基于该向量利用LSTM预测加工路径。**



3DCNN-LSTM



模型： 3DCNN-LSTM **参数量：** 23k

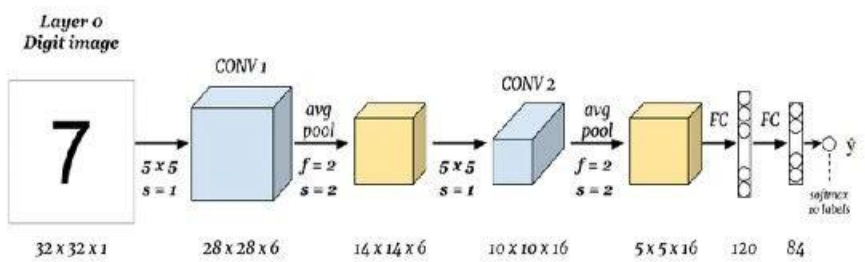
效果： 通过使用3DCNN-LSTM网络实现了不同需求下刀具加工路径的预测，预测结果精度达到94%。

预测结果

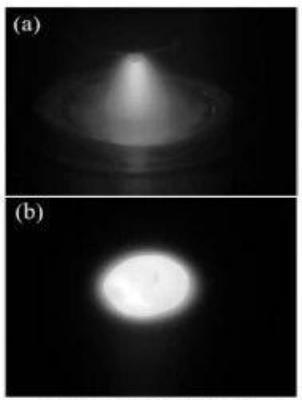
Yang Y, et al. A deep learning-based approach for machining process route generation. **The International Journal of Advanced Manufacturing Technology**, 2021, 115: 3493-3511.

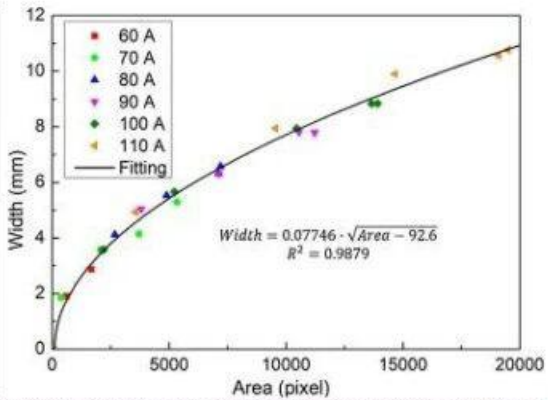
AI4M案例：加工装配

□ **小模型应用于智能制造焊接质量的预测：依据焊接过程中正面的图像参数，采用CNN对获取的图像进行特征提取以预测焊接背面的质量，并验证其性能。**



CNN





模型： CNN **参数量：** 53k

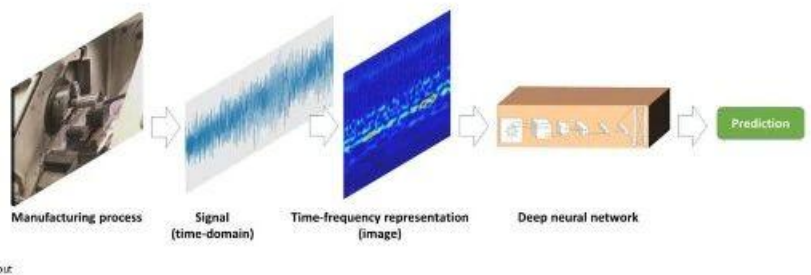
效果： 通过使用CNN能够快速预测不同工况下焊接部件背面的质量，预测结果对应的R^2达到了0.9879。

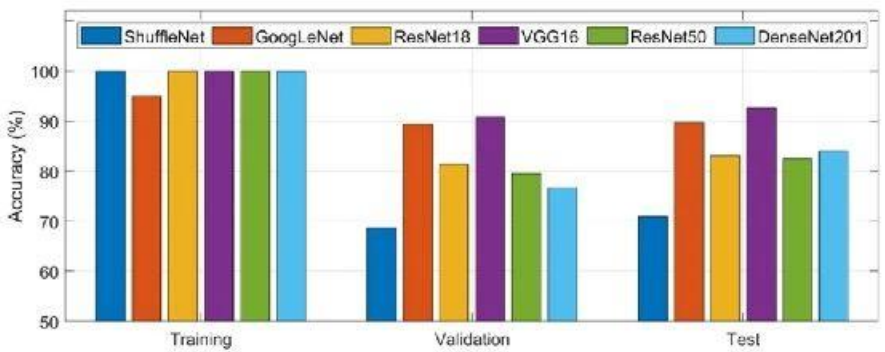
预测结果

Wang Q, et al. A tutorial on deep learning-based data analytics in manufacturing through a welding case study. **Journal of Manufacutring Processes**, 2021, 63: 2-13.

AI4M案例：加工装配

□ **大模型应用于智能制造过程监控：将时频分析与深度神经网络相融合，提出了一种制造过程监控方法，应用于机床车削操作，并对其性能进行了详细的评价。**





模型： VGG CNN **参数量：** 138M

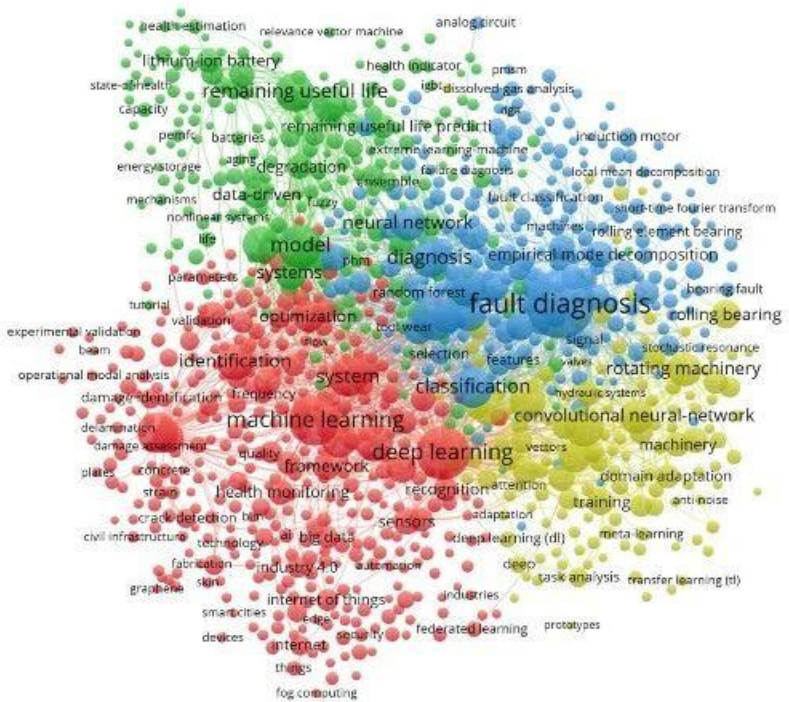
效果： 通过使用从VGG-16网络修改的较浅网络来缓解过拟合问题，改善了分类准确性，实现了95.58%的分类准确性

预测结果

LIAO Y, et al. Manufacturing process monitoring using time-frequency representation and transfer learning of deep neural networks . **Journal of Manufacturing Processes**,2021, 68: 231-248.

AI4M统计：控制运维

□ 对“AI+控制运维”相关关键词进行检索，并绘制统计图和关键词共现图谱

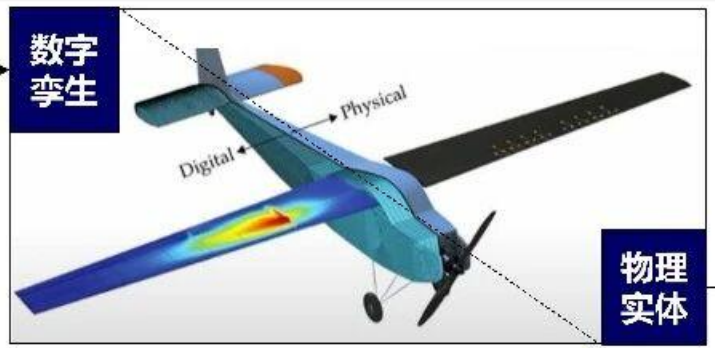


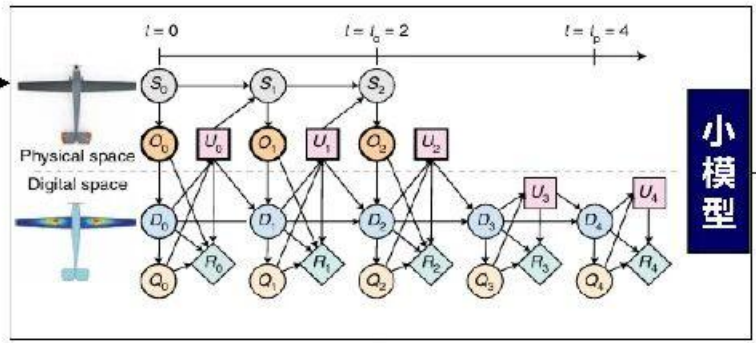
关键词共现图谱

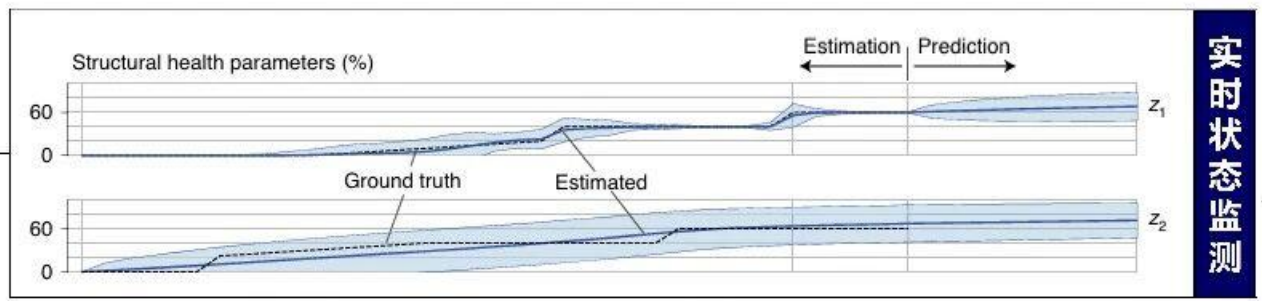
检索式： TS=("artificial intelligence" OR "machine learning" OR "deep learning" OR "neural network"OR"data-driven") AND TS=("predictive maintenance" OR "operation and maintenance" OR "fault diagnosis" OR "health monitoring"OR"equipment health management"OR"remaining useful life")

AI4M案例：控制运维

□ **小模型应用于装备故障诊断与健康监测：基于概率图模型构建无人机的结构数字孪生模型，并使用传感器数据进行动态更新，从而实现无人机状态实时监测。**





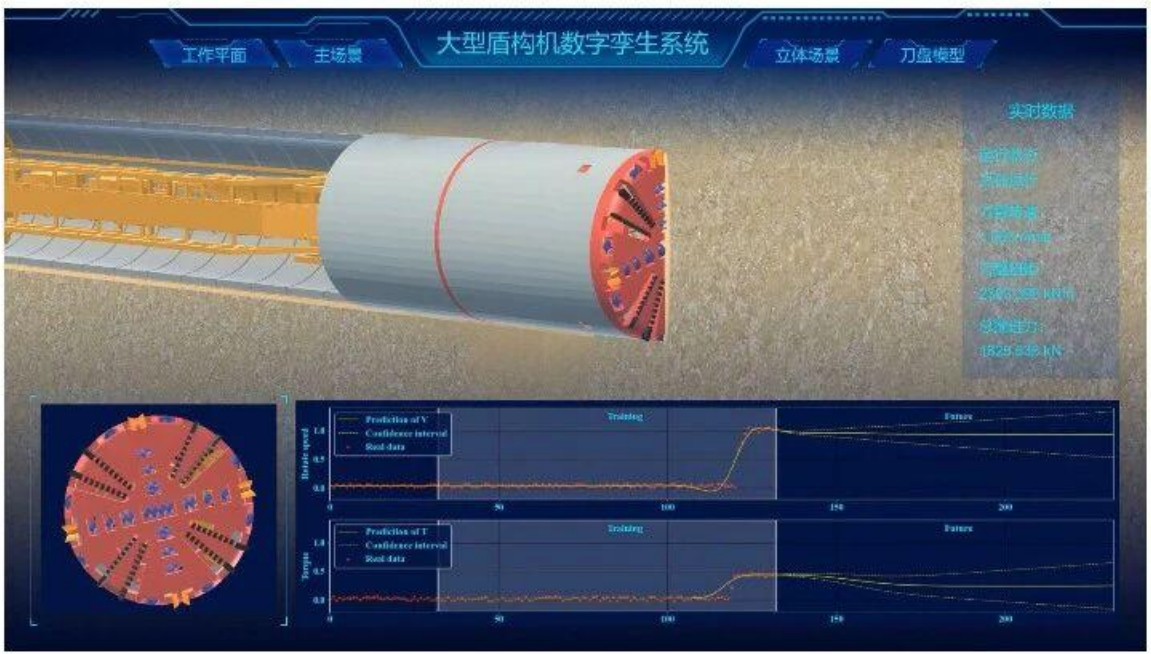


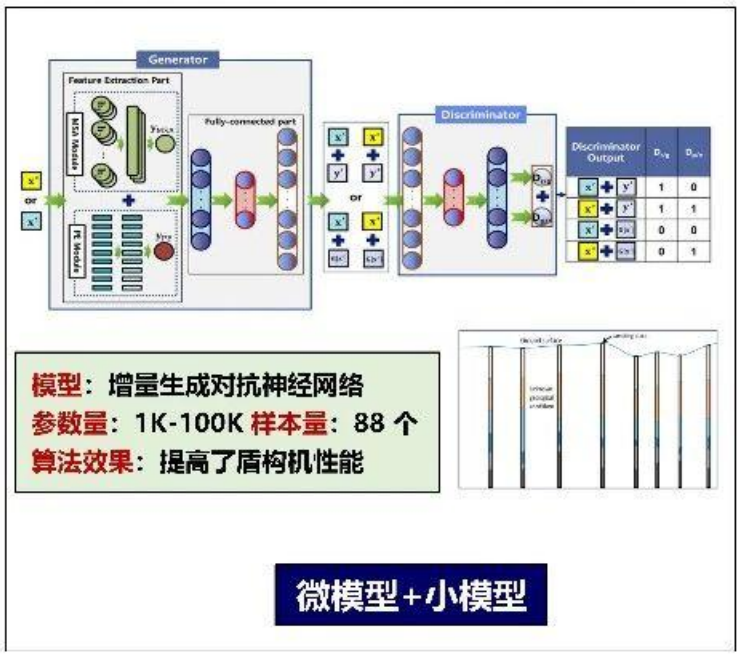
模型：概率图模型
参数量：10K-100K
算法效果：两个时间步内对结构健康变化做出响应

Kapteyn M G et al. A probabilistic graphical model foundation for enabling predictive digital twins at scale. Nature Computational Science,2021,1(5):337-347.

AI4M案例：控制运维

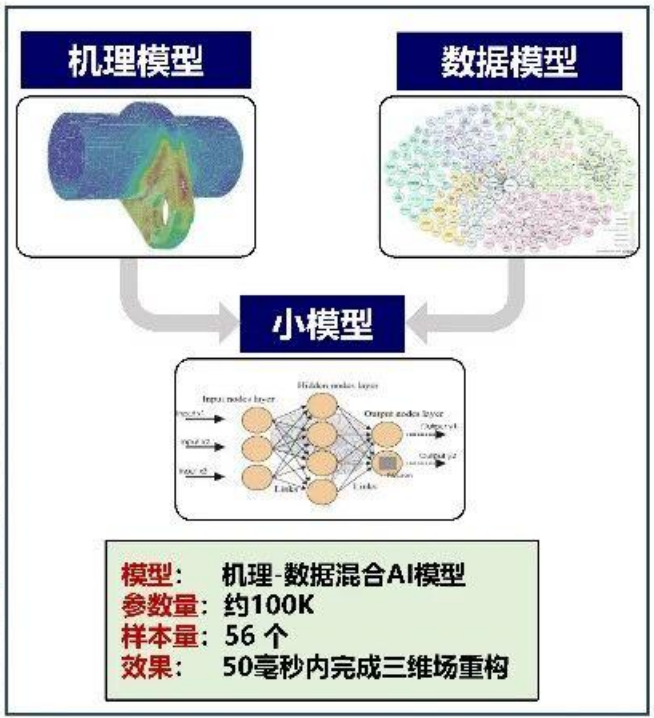
□ **微模型+小模型应用于装备运行数据处理：** 通过时序数据聚类、稀疏数据划分以及基于小模型参数预测等方式，实现装备运行数据的清洗、分析与预测。





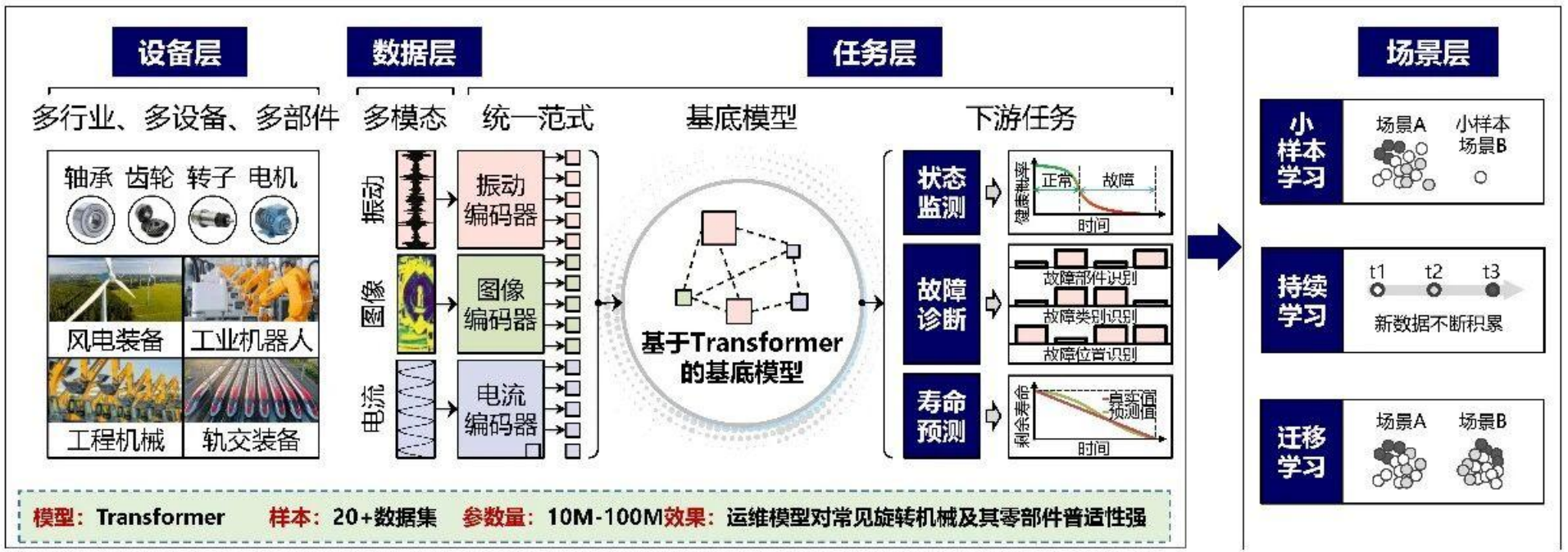
AI4M案例：控制运维

□ **小模型应用于装备结构健康监测：** 通过小样本数据模型驱动的形-性一体化数字孪生系统，实现结构应力场等三维全场信息实时检测与计算。



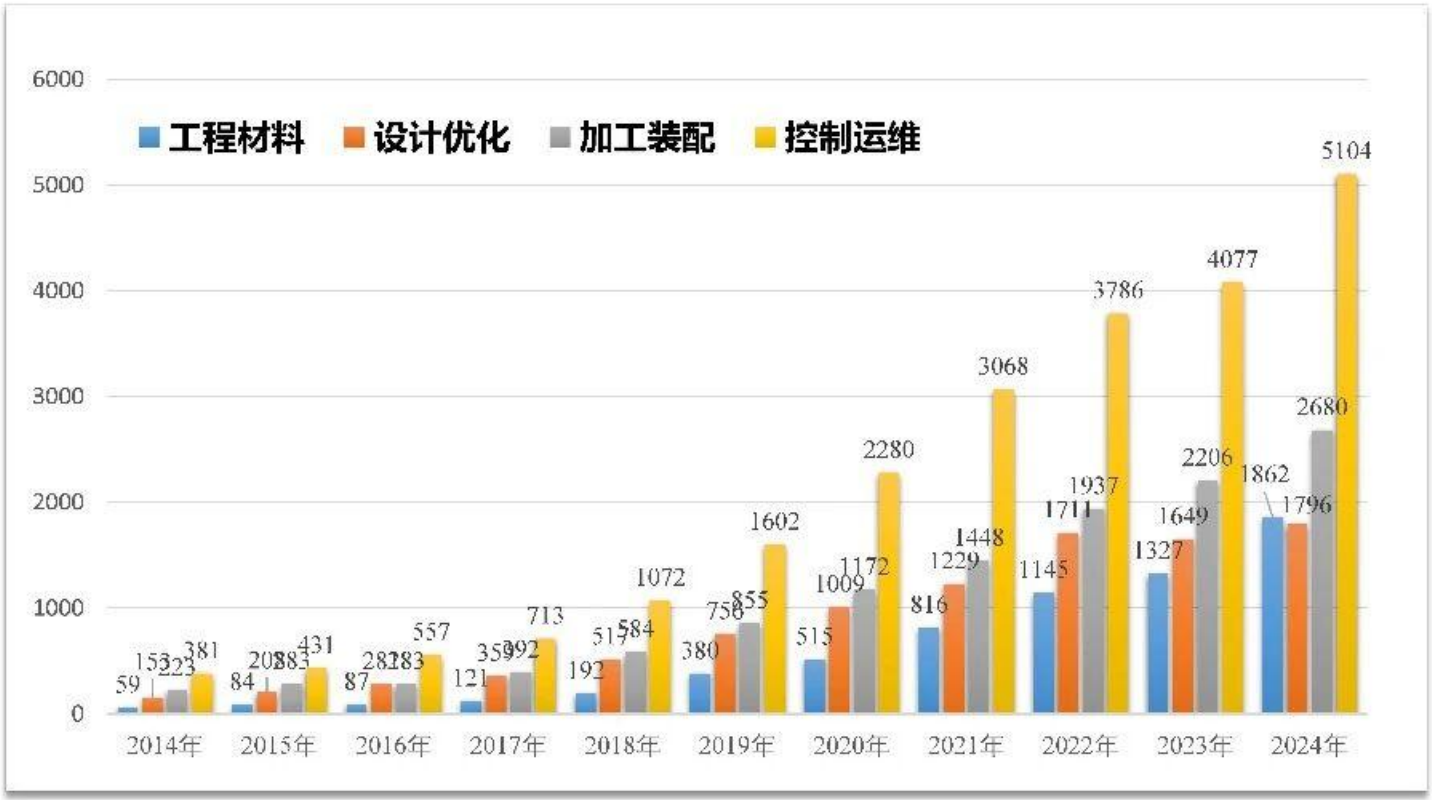
AI4M案例：控制运维

□ **大模型应用于装备智能运维：提出“基底模型预训练+适配微调”的通用智能运维基础模型，建立了“对象-数据-任务-场景”多层次贯通的智能运维新模式。**



雷亚国 等. 面向机械设备通用健康管理的智能运维大模型. 机械工程学报, 2025, 61(6): 1-13.

AI4M的数据归纳



- AI+论文数快速上升，2024年数量是2014年的20倍；
- 运维是论文数量最多的领域，加工制造紧随其后，设计最少，这与领域知识有直接关系；
- 绝大部分研究以小模型和微模型为主，大模型研究较少，应用为主。

一、AI4M的背景意义

二、AI4M的基础知识

三、AI4M的研究进展

四、AI4M的案例展示（大连理工）

五、AI4M的瓶颈所在

六、AI4M的科学问题

七、AI4M的发展方向

八、思考与总结

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：选取六种主流大语言模型，兼顾代表性、可获取性与运行效率，确保评估结果具备可复现性，反映当前主流模型在工程任务中的真实表现。

模型名称	研发公司	版本	核心特点
DeepSeek	深度求索	DeepSeek R1	➢ 基于强化学习和监督微调，推理能力强 ➢ 成本效益高，运行成本低 ➢ 性能接近GPT-4o，但训练成本更低
豆包	字节跳动	Doubao-1.5-pro	➢ 最新版本，性能稳定，适合多种任务处理 ➢ MoE架构：高效模型结构，降低推理成本
Gemini	Google	Gemini 2.0Flash	➢ 多模态能力强大，支持文本、图像、视频和音频输入 ➢ 性能优Gemini 1.5 Pro，响应速度更快
Grok	xAI	Grok2	➢ 集成于X平台，注册即送免费使用 ➢ 支持多模态输入，适合复杂任务处理
Kimi	月之暗面	Kimi-latest	➢ Kimi智能助手产品使用最新的Kimi 大模型版本 ➢ 支持多模态能力
ChatGPT	OpenAI	GPT-4o	➢ 最新版本，具有强大的世界知识和语料库，支持多模态能力（如图像处理） ➢ 响应速度快，API成本低 ➢ 支持桌面版和轻量化体验

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：预测任务一维、二维、十维测试函数。

维度	名称	表达式
一维	测试函数1	$f(x) = 10.3x - 5.2$
	测试函数2	$f(x) = 1.2x^2 - 3.5x + 5$ 定义域：[-5，5]
	测试函数3	$f(x) = (6x-2)^2 \cdot \sin(12x-4)$ 定义域：[0，1]
二维	测试函数4	$f(x) = x_1 + 2x_2 + 1$ 定义域：[-5,5],[-5,5]
	测试函数5	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$ 定义域：[-4.5,5.5], [-4.5,5.5]
	测试函数6	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$ 定义域：$[-100,100]\times[-100,100]$
十维	测试函数7	$f(x) = \sum_{i=1}^{10}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ 定义域：$x_i \in [-1,1]$
	测试函数8	$f(x) = \sum_{i=1}^{9}\left((x_{i+1}^2 - x_i)^2 + (x_i - 1)^2\right)$ 定义域：$x_i \in [-3,3]$
	测试函数9	$c_1 = -6.089, c_2 = -17.164, c_3 = -34.054, c_4 = -5.914, c_5 = -24.721,$ $c_6 = -14.986, c_7 = -24.100, c_8 = -10.708, c_9 = -26.662, c_{10} = -22.179, s = \sum_{i=1}^{10} e^{x_i}$ $f(x) = \sum_{j=1}^{10}(e^{x_j}(c_j + x_j - \log s))$ 定义域：$x_i \in [-0.1,10]$

AI4M案例-1 ： 设计优化

□ 大语言模型应用于优化设计：优化任务单目标无约束/带约束测试函数。

模型名称	版本	表达式	真实解
单目标无约束	测试函数1	$f(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cos(x_1)+10$ 定义域：$x_1\in[-5,10]$，$x_2\in[0,15]$	$f(x)=0.3979$
	测试函数2	$f(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$ 定义域：$x_1\in[-5,5]$，$x_2\in[-5,5]$	$f(x)=0$
	测试函数3	$f(x)-10d+\sum_{i=1}^{10}[x_i^2-10cos(2\pi x_i)]$ $x_i\in[-5.12,5.12]$，$i-1,2,\ldots d,d-10$	$f(x)=0$
单目标带约束	测试函数4	$f(x)-5\sum_{i=1}^{4}x_i-5\sum_{i=1}^{4}x_i^2-\sum_{i=5}^{13}x_i$ $g_1(x)-2x_1+2x_2+x_{10}+x_{11}-10<0$ $g_2(x)-2x_1+2x_3+x_{10}+x_{12}-10<0$ $g_3(x)-2x_2+2x_3+x_{11}+x_{12}\quad 10\le 0$ $g_4(x)-\quad 8x_2+x_{10}\le 0$ $g_5(x)-\quad 8x_2+x_{11}\le 0$ $g_6(x)=\quad 8x_3+x_{12}<0$ $g_7(x)=\quad 2x_4\quad x_5+x_{10}<0$ $g_8(x)=\quad 2x_6\quad x_7+x_{11}<0$ $g_9(x)=\quad 2x_8\quad x_9+x_{12}<0$	$f(x)=\quad 15$
	测试函数5	$f(x)=\left\lvert\frac{\sum_{i=1}^{n}cos^4(x_i)-2\prod_{i=1}^{n}cos^2(x_i)}{\sqrt{\sum_{t=1}^{n}ix_t^2}}\right\rvert$, $n-20$ $g_1(x)-7.5\quad\prod_{t-1}^{n}x_i\le 0$ $g_2(x)-\sum_{i=1}^{n}x_i\quad 7.5n\le 0$	$x_i\subset(0,10],(i=1,\ldots,n)$ $f(x)=-0.8036$ x=(3.1625,3.1283,3.0948,3.0615,3.0279,2.9939,2.9587,2.9218,0.4948,0.4884,0.4823,0.4766,0.4713,0.4662,0.4614,0.4568,0.4525,0.4483,0.4443,0.4404)
	测试函数6	$f(x)-\quad-(\sqrt{n})^n\prod_{i=1}^{n}x_i\le 0$ $g_1(x)-\sum_{i=1}^{n}x_i^2-1-0$	$x_i\in(0,10],(i-1,\ldots,n)$ $f(x)-\quad-1.0005$ x=(3.1625,3.1283,3.0948,3.0615,3.0279,2.9939,2.9587,2.9218,0.1918,0.4884,0.4823,0.4766,0.4713,0.4662,0.4614,0.4568,0.4525,0.44830.4443,0.4404)

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：优化任务多目标优化和多峰全局优化测试函数。

模型名称	版本	表达式	真实解
多目标优化	测试函数7	$f(x)=1+9(\sum_{i=2}^{n}x_i)/(n-1)$ $g_1(x)=x_1, g_2(x)=f(x)\left[1-\sqrt{x_1/f(x)}\right]$ 定义域：$x\in[0,1]$	$x_1\in[0,1]$ $x_i=0,\quad i=2,\ldots,n$
	测试函数8	$f(x)=1+9\left(\sum_{i=2}^{n}x_i\right)/(n-1)$ $g_1(x)=x_1, g_2(x)=f(x)\left[1-\left(x_1/f(x)\right)^2\right]$ 定义域：$x\in[0,1]$	$x_1\in[0,1]$ $x_i=0,\quad i=2,\ldots,n$
	测试函数9	$f(x)=1+9\left(\sum_{i=2}^{n}x_i\right)/(n-1)$ $g_1(x)=x_1$ $g_2(x)=f(x)\left[1-\sqrt{x_1/f(x)}-x_1/f(x)\sin(10\pi x_1)\right]$ 定义域：$x\in[0,1]$	$x_1\in[0,1]$ $x_i=0,\quad i=2,\ldots,n$
多峰全局优化	测试函数10	$f(x)=\left(4-2.1x_1^2+\frac{1}{3}x_1^4\right)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$ $x_1\in[-3,3], x_2\in[-2,2]$	(0.0898, -0.7126), (-0.0898, 0.7126), $f(x)$=-1.0316258
	测试函数11	$f(x)=\frac{\pi}{n}\left\{10\sin^2(\pi x_1)+\sum_{i=1}^{n-1}[(x_i-1)^2(1+10\sin^2(\pi x_{i+1}))]+(x_n-1)^2\right\}-10\le x_i\le 10; i=1\ to\ 5$	x_i=1 $f(x)$–0
	测试函数12	$f(x)=\sum_{i=1}^{11}\left[a_i-x_1\frac{b_i^2-b_ix_2}{b_i^2+b_ix_3+x_4}\right]\quad -2\le x_i\le 2;\ i=1\ to\ 4$ (a_i, b_i) (i − 1 to 11) : (0.1975, 4), (0.1947, 2), (0.1735, 1), (0.16, 0.5), (0.0844, 0.25),(0.0627, 0.1667), (0.0456, 0.125), (0.0342, 0.1), (0.0323, 0.0833), (0.0235, 0.0714),(0.0246, 0.0625)	(0.195, -0.179, 0.130, 0.130) $f(x)$−3.1302 × 10 $^{-4}$

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：预测任务底层实际采用的计算方法

函数	样本量	DeepSeek	豆包	Gemini	Grok	Kimi	ChatGPT
测试函数1	10n	Linear Regression	Linear Regression	Polynomial Regression (degree 2)	Linear Regression	Linear Regression	Polynomial Regression (degree 2)
	20n	Linear Regression	Linear Regression	Linear Regression	Linear Regression	Linear Regression	Linear Regression
	50n	Linear Regression	Linear Regression	Polynomial Regression (degree 2)	Linear Regression	Linear Regression	Linear Regression
测试函数2	10n	Polynomial Regression (degree 2)	Linear Regression	Polynomial Regression (degree 1~5)	Linear Regression	Linear Regression	Polynomial Regression (degree 2)
	20n	Polynomial Regression (degree 2)	Polynomial Regression (degree 2)	Polynomial Regression (degree 2)	Polynomial Regression (degree 2)	Linear Regression	Polynomial Regression (degree 2)
	50n	Polynomial Regression (degree 2)	Polynomial Regression (degree 2)	Linear Regression	Polynomial Regression (degree 2)	Linear Regression	Linear Regression
测试函数3	10n	cubic spline interpolation	Linear Regression	Polynomial Regression (degree 3)	Polynomial Regression (degree 3)	Linear Regression	Polynomial Regression (degree 3)
	20n	Gaussian Process Regression (GPR) with RBF kernel	Linear Regression	Polynomial Regression (degree 3)	Linear Regression	Linear Regression	Polynomial Regression (degree 3)
	50n	Cubic Spline Interpolation	Linear Regression	Linear Regression	Linear Regression	Linear Regression	Linear Regression
测试函数4	10n	Linear Regression	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Linear Regression	Linear Regression	Random Forest Regressor
	20n	Polynomial Regression (degree 2)	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Linear Regression	Linear Regression	Random Forest Regressor
	50n	Random Forest Regressor	Linear Regression	Polynomial Regression (degree 2)	Linear Regression	Linear Regression	Linear Regression

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：预测任务底层实际采用的计算方法

函数	样本量	DeepSeek	豆包	Gemini	Grok	Kimi	ChatGPT
测试函数5	10n	Random Forest Regressor	Linear Regression	Polynomial Regression (degree 3)	Polynomial Regression (degree 3)	Linear Regression	Linear Regression
	20n	Random Forest Regressor	Linear Regression	Random Forest Regressor	Polynomial Regression (degree 2)	Linear Regression	Linear Regression
	50n	Random Forest Regressor	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Polynomial Regression (degree 2)	Linear Regression	Linear Regression
测试函数6	10n	Random Forest Regressor	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Linear Regression	Linear Regression	Linear Regression
	20n	Random Forest Regressor	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Polynomial Regression (degree 2)	Linear Regression	Linear Regression
	50n	Random Forest Regressor	Linear Regression	Gaussian Process Regression (GPR) with RBF kernel	Polynomial Regression (degree 3)	Linear Regression	Linear Regression
测试函数7	10n	Random Forest Regressor	Linear Regression/ Decision Tree Regressor	Linear Regression	Linear Regression	Random Forest Regressor	Linear Regression
	20n	Random Forest Regressor	Linear Regression	Random Forest Regressor	Random Forest Regressor	Linear Regression	Random Forest Regressor
	50n	数据量过大，输出异常	数据量过大，输出异常	Polynomial Regression (degree 2)	数据量过大，输出异常	数据量过大，输出异常	Linear Regression
测试函数8	10n	Random Forest Regressor	Decision Tree Regressor	Linear Regression	Random Forest Regressor	Linear Regression	分批输入，遗忘上面训练集，未给出结果
	20n	Random Forest Regressor	Linear Regression	Linear Regression	Polynomial Regression (degree 2)	Linear Regression	Linear Regression
测试函数9	10n	Random Forest Regressor	Linear Regression	SVR with RBF kernel	Linear Regression	Random Forest Regressor	Random Forest Regressor
	20n	Random Forest Regressor	Linear Regression	Random Forest Regressor	Random Forest Regressor	Linear Regression	Random Forest Regressor

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：优化任务底层实际采用的计算方法

函数	DeepSeek	豆包	Gemini	Grok	Kimi	ChatGPT
测试函数1	解析法	BFGS (Broyden - Fletcher - Goldfarb - Shanno)	SLSQP	解析法	解析法	SLSQP
测试函数2	解析法	解析法	解析法	解析法	解析法	解析法
测试函数3	解析法	解析法	解析法	解析法	解析法	Differential Evolution
测试函数4	SLSQP	CVXPY	SLSQP	SLSQP	解析法	SLSQP
测试函数5	SLSQP	SLSQP	trust-region	SLSQP	SLSQP	SLSQP
测试函数6	解析法	Lagrange multipliers	解析法	Large Margin Cosine Loss	Lagrange multipliers	Lagrange multipliers
测试函数7	解析法	解析法	NSGA-II (未给出结果)	Evolutionary Algorithms	分析法/试错法	解析法
测试函数8	解析法	解析法	NSGA-II (未给出结果)	MOPSO	分析法/试错法	解析法
测试函数9	解析法	解析法	NSGA-II	Evolutionary Algorithms	解析法	解析法
测试函数10	解析法	L-BFGS-B	L-BFGS-B	L-BFGS-B	解析法	解析法
测试函数11	Evolutionary Algorithms	L-BFGS-B	Evolutionary Algorithms	SHGO	SLSQP	L-BFGS-B
测试函12	Evolutionary Algorithms	L-BFGS-B	SLSQP	SLSQP	SLSQP	trust-region

AI4M案例-1：设计优化

□ 大语言模型应用于优化设计：不同预测任务与优化任务的模型性能对比结果。

函数	样本量	DeepSeek	ChatGPT	豆包	Gemini	Grok	Kimi
测试函数 1	10n	1	1	0.9985	0.9987	0.9433	0.9293
	50n	1	1	0.9992	0.9974	0.9943	1
	200n	1	0.9625	0.9994	0.9997	0.4095	0.9840
测试函数 2	10n	0.9871	1	0.9932	0.8168	-0.4793	-0.6877
	50n	1	1	0.9980	-0.3371	0.9983	0.9092
	200n	1	-0.5609	0.9993	0.9949	0.9981	-0.3804
测试函数 3	10n	0.6717	0.0625	-0.3636	0.4050	0.4699	\
	50n	0.9775	0.0360	0.8904	0.9580	0.9381	0.0293
	200n	0.9472	/	0.8803	/	-0.6022	0.0293
测试函数 4	10n	0.9412	-0.0494	0.1284	0.8368	-0.4475	-23.1230
	50n	0.9103	-0.2623	0.4551	0.2137	-0.5670	0.3544
	200n	0.5926	0.0051	0.4227	0.8629	/	0.1225
测试函数 5	10n	-0.1583	-0.2503	-0.3329	-1.8038	-2.5573	-0.1871
	50n	0.3613	0.0558	0.1025	0.0301	-0.9876	-0.1851
	200n	-1.7013	-0.2210	-0.0309	0.1253	-0.1874	-0.2141
测试函数 6	10n	0.2610	0.0455	-0.4429	0.1306	-0.6968	-2.7484
	50n	0.6136	-0.2726	-0.6571	0.3358	-0.3161	-1.2533
	200n	0.1293	/	0.4934	0.3782	-0.1442	-0.1320
测试函数 7	10n	-0.1697	/	/	0.0271	-0.0012	0.0257
	50n	-0.3100	-17.9687	/	-0.1627	0.3316	-0.1491
	200n	/	-20.3344	/	-0.9765	/	/
测试函数 8	10n	-0.0727	/	0.1082	-0.0963	-0.4775	-0.6877
	50n	-0.1139	-3.9387	-0.3280	-0.0809	-0.1018	-0.2029
测试函数 9	10n	-1.0879	/	-0.4591	-0.0810	-1.4716	/
	50n	-0.9205	-0.3502	/	-1.0289	-0.2972	-0.3223

不同维度函数下的模型性能对比（R2）

函数	DeepSeek	豆包	Gemini	Grok	Kimi	ChatGPT
测试函数 1 单目标无约束	0.3979	0	0	0.3979	0.3979	0.3979
测试函数 2 单目标无约束	0	-0.9219	-1.14	0	0	0
测试函数 3 单目标无约束	0	0	0	0	0	0
测试函数 4 单目标带约束	-15	/	/	-13.75	0	-21
测试函数 5 单目标带约束	-0.4333	/	/	/	-0.0027	-124211.80
测试函数 6 单目标带约束	-1	-1	-1	-100000	-1	-100000
测试函数 7 多目标优化	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	/	$F_2 = 1 - \sqrt{F_1}$ $0 \le F_1 \le 1$	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$F_2 = 1 - \sqrt{F_1}$ $0 \le F_1 \le 1$
测试函数 8 多目标优化	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$F_2 = \sqrt{(10 - F_1)(F_1)}$ $0 \le F_1 \le 10$	/	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$F_2 = 1 - F_1^2$ $F_1 \in [0,1]$
测试函数 9 多目标优化	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2, \ldots, n$	/	$F_1 = x_1$ $F_2 = 1 - \sqrt{x_1}$ $x_1 \sin(10\pi x_1)$	/	$F_1 = x,$ $g(x) = 1$ $F_2 = 1 - \sqrt{x_1}$ $-x_1 \sin(10\pi x_1)$
测试函数 10 多峰全局优化	[-0.0898,0.7126]/ [0.0898,-0.7126] -1.0316	[0, 0] 0	[0, 0] 0	[0, 0] 0	[0, 0] 0	[0, 0] 0
测试函数 11 多峰全局优化	$x_1 = 1$ $f=0$	[0.001,0.010, 0.010,0.010, 0.010,3.110] $f=1$	$x_1 = 1$ $f=0$	$x_1 = 1$ $f=0$	$x_1 = 1$ $f=0$	$x_1 = 1$ $f=0$
测试函数 12 多峰全局优化	(0.194,0.181, 0.230,0.210) $f=0$	(2,0,0, 0.005) $f=879897.86$	(0.543, 0,0,0) $f=-4$	(1.656,1.195, 1.738, 0.261) $f=-423.096$	(0.195, - 0.179, 0.130, 0.120) $f=0$	(0.543, 0,0,0) $f=-4$

不同优化函数下的模型输出结果

LLM在工程优化设计领域中展现出日益显著的应用潜力。在处理得当的情况下，面对包含复杂目标函数与约束条件的工程问题时，体现出其在工程场景中作为智能调度器与方法集成器的独特价值。

AI4M案例-2：设计优化

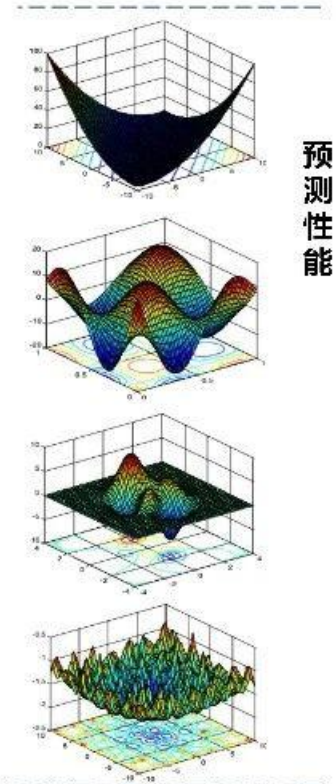
□ 微模型应用于优化设计：选取9种常用的小模型，兼顾代表性、可用性与计算效率，确保评估结果具备可复现性，反映小模型在工程任务中的真实表现。

模型名称	核心特点
PRS（多项式拟合）	计算效率高，系数具有物理可解释性，鲁棒性强，适用于小样本、无梯度问题，精度有限，具有过拟合风险
RBF（径向基函数）	计算成本低，参数含义清晰，局部响应强，鲁棒性低，平滑性较好，连续可微，扩展能力有限
KRG（克里金）	适用于小样本建模，可解释性强，计算精度高，对噪声和稀疏数据更稳健，非线性适应性高
SVR（支持向量回归）	支持非线性回归，具有全局优化特点，抗噪声能力强，非线性适应度高，参数调节复杂
MLS（移动最小二乘）	适用于散乱数据点的拟合与平滑，具有高的光滑性和良好的局部特性，可扩展性强，对噪声具有一定鲁棒性
RF（随机森林）	可有效降低过拟合风险，对噪声和缺失值鲁棒性强，训练过程可并行，可处理高维特征，较少的超参数调节
Adboost	无需特征筛选，对噪声和异常值敏感，适用于二分类与多分类问题，无法并行训练
DCNN（深度卷积神经网络）	训练成本高，具有强大的特征提取能力，可实现端到端的学习，泛化能力强，可迁移性强，易于扩展
ELM（单/多隐藏层前向神经网络）	训练速度快，无需迭代调参，单隐含层结构，泛化能力强，适用于回归与分类，易于扩展

AI4M案例-2 ： 设计优化

□ 微模型应用于优化设计：采用 38 个测试函数对常见的代理模型方法进行系统对比。

序号	维度	标准测试函数	定义域	非线性程度
1	1	$y=(6x-2)^2\sin\left[2(6x-2)\right]$	$[0;1]^D$	高
2	1	$y=\frac{\sin(10\pi x)}{2x}+(x-1)^4$	$[0.5;2.5]^D$	低
3	2	$y=\left(x_2-\frac{5.1x_1^2}{4\pi^2}+\frac{5x_1}{\pi}-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$	$[-5.0;10,15]^D$	高
4	2	$y=10^5x_1^2+x_2^2-(x_1^2+x_2^2)^2+10^5(x_1^2+x_2^2)^4$	$[-20;20]^D$	高
5	2	$y=[1-2x_2+0.05\sin(4\pi x_2-x_1)]^2+[x_2-0.5\sin(2\pi x_1)]^2$	$[-5;10]^D$	高
6	2	$y=2x_1^2-1.05x_1^4+\frac{1}{6}x_1^6+x_1x_2+x_2^2$	$[-5;5]^D$	高
7	2	$y=(4-2.1x_1^2+\frac{1}{3}x_1^4)x_1^2+x_1x_2+x_2x_2+(-4+4x_2^2)x_2^2$	$[-3,-2;3,2]^D$	高
8	2	$y=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2-6x_1x_2+3x_2^2)][30-(2x_1-3x_2)^2(18-32x+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$[-2,2]^D$	高
9	2	$y=x_1\exp(-x_1^2-x_2^2)$	$[-2;2]^D$	高
10	2	$y=[10+x_1\sin(x_1)][4+\exp(-x_2)^2]$	$[-2;2]^D$	高
11	2	$y=(4-2.1x_1^2+\frac{1}{3}x_1^4)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	$[-3,-2;3,2]^D$	低
12	3	$y=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}A_{ij}(x_j-P_{ij})^2\right)$	$[0;1]^D$	低
13	3	$y=-x_1x_2x_3$	$[0.0,0;20.11.42]^D$	低
14	4	$y=100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)+10.1\left[(x_2-1)^2+(x_4-1)^2\right]+19.8(x_2-1)(x_4-1)$	$[-10;10]^D$	高
15	4	$y=10\sin\left[2(x_1-0.6\pi)\right]+x_2+x_3+x_4+x_1x_2+x_3x_4+x_1^2+x_4^2$	$[0;1]^D$	低
16	4	$y=\frac{1}{2\exp}\left[1.1-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{4}A_{ij}(x_j-P_{ij})^2\right)\right]$	$[0;1]^D$	高
17	4	$y=\sum_{i=1}^{5}\left[\left(\sum_{j=1}^{5}x_j\right)-b_j\right]^2$	$[0;4]^D$	低
18	4	$y=1+\exp[-2[(x_1-1)^2+x_2^2]-0.5(x_1^2+x_2^2)]+\exp[-2[x_1^2+(x_2-1)^2]-0.5(x_1^2+x_2^2)]$	$[0;1]^D$	低
19	4	$y=24.55x_1+26.75x_2+39x_3-40.5x_4$	$[0;1]^D$	低



预测性能

非线性度量

$$y=L(x)=Ax+b$$

非线性度计算

$$M=\sqrt{\min_{A,b}E[\|L(x)-f(x)\|_2^2]}$$

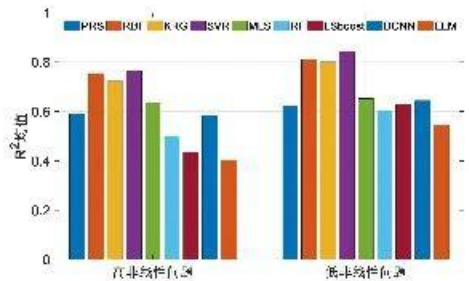
测试问题

低维：1-7维　　数量：26个
高维：8-16维　　数量：12个
低非线性度：19个
高非线性度：19个

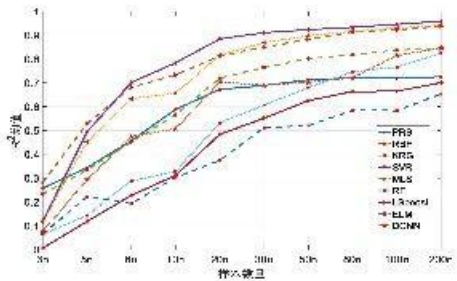
AI4M案例-2：设计优化

□ 微模型应用于优化设计：SVR对不同非线性程度问题的预测具有较高的精度

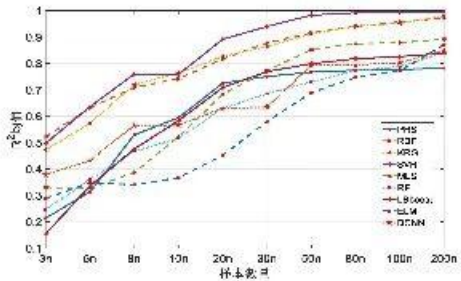
PRS → RBF → KRG → SVR → MLS → RF → Adboost → DCNN → ELM



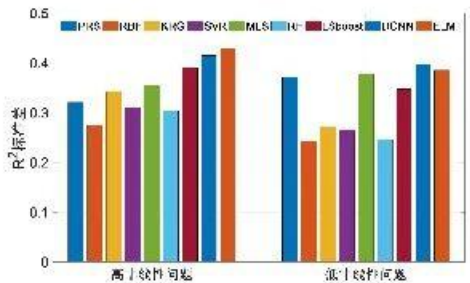
不同模型的R^2均值比较



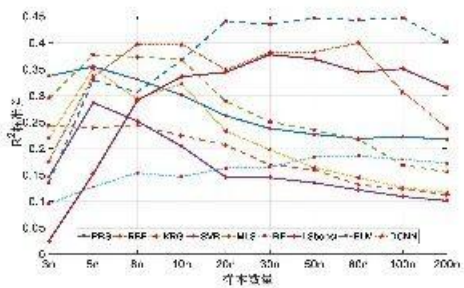
高非线性函数R^2均值



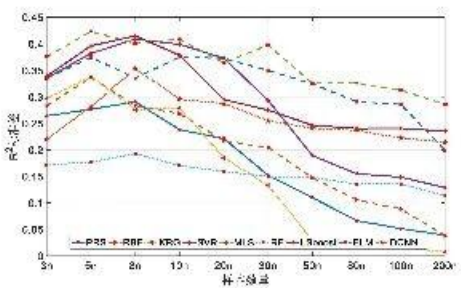
低非线性函数R^2均值



不同模型的R^2标准差比较



高非线性函数R^2标准差

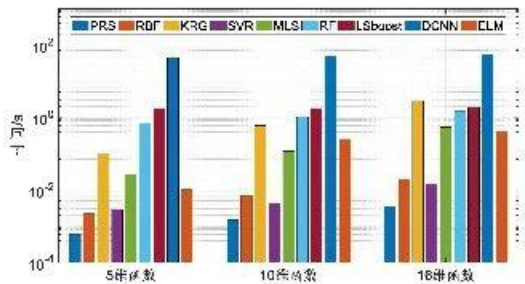


低非线性函数R^2标准差

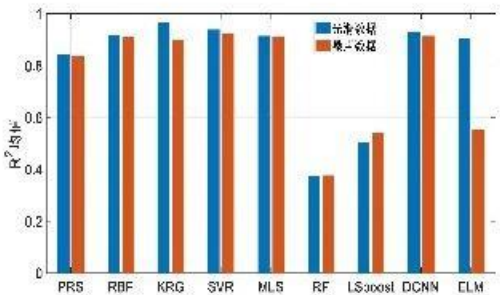
AI4M案例-2：设计优化

□ 微模型应用于优化设计：传统代理模型训练时间少、预测速度快、在低维具有高精度。

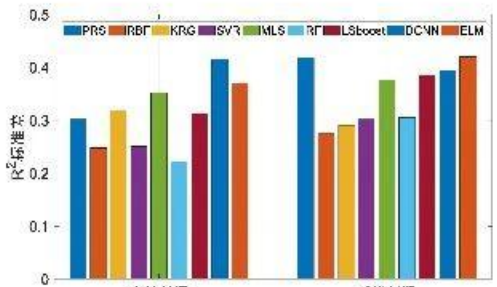
PRS → RBF → KRG → SVR → MLS → RF → Adboost → DCNN → ELM



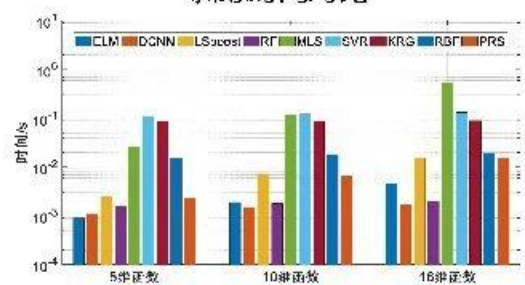
训练时间对比



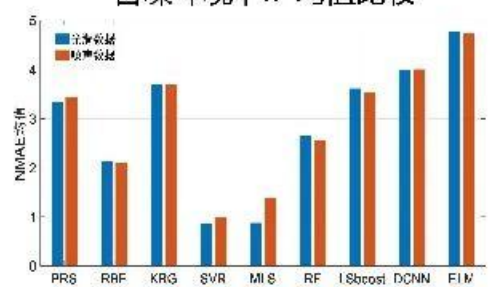
含噪环境下R^2均值比较



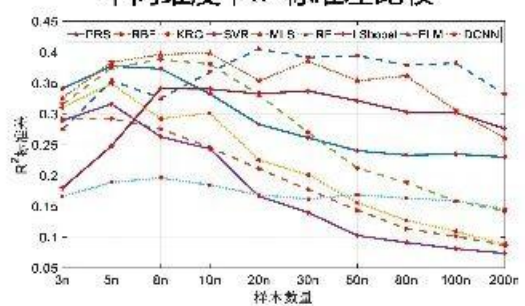
不同维度下R^2标准差比较



预测时间对比



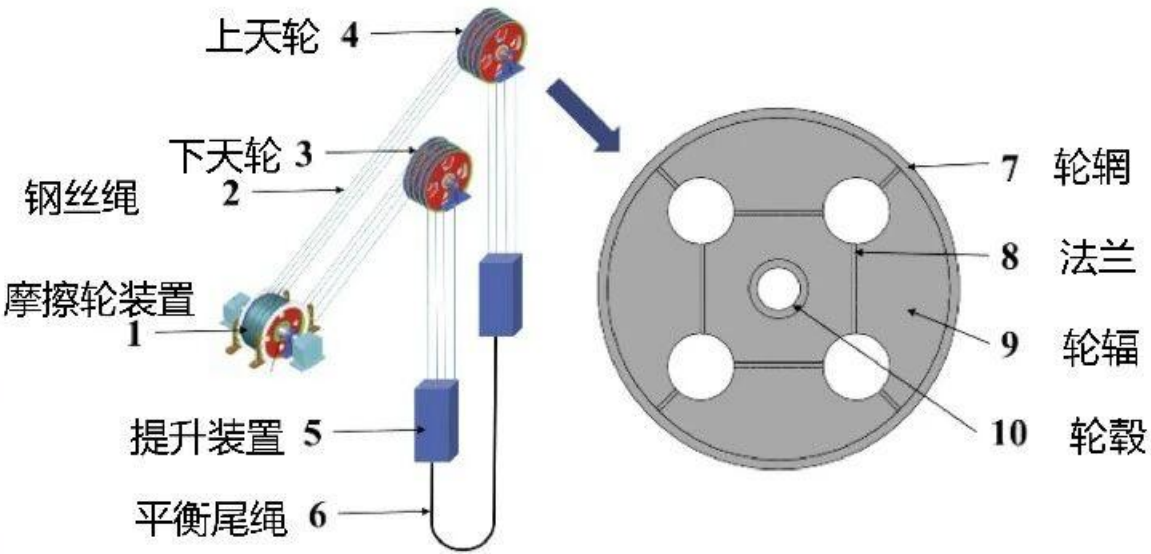
含噪环境下NMAE均值比较



不同样本数量下R^2标准差比较

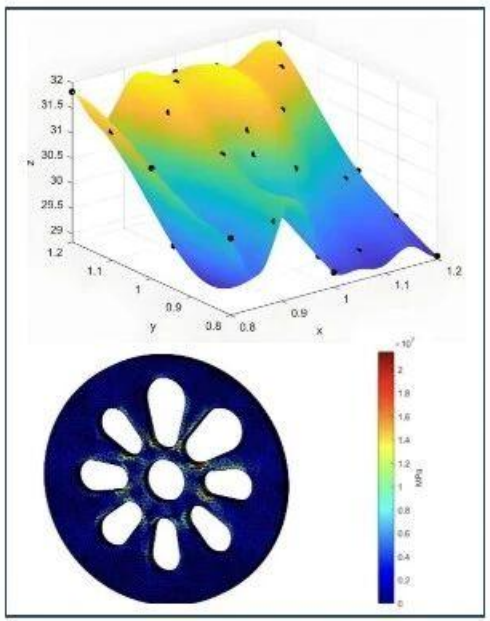
AI4M案例-3：设计优化

- **小模型应用于优化设计：天轮是矿井提升系统中最重要的部件之一，承担着矿井内物料的升降运输任务。其结构的稳定性是保证运输设备的安全及经济效益的重要因素。**

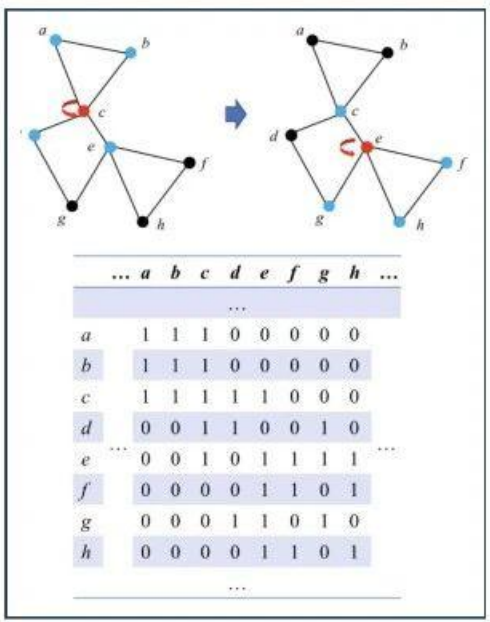


AI4M案例-3：设计优化

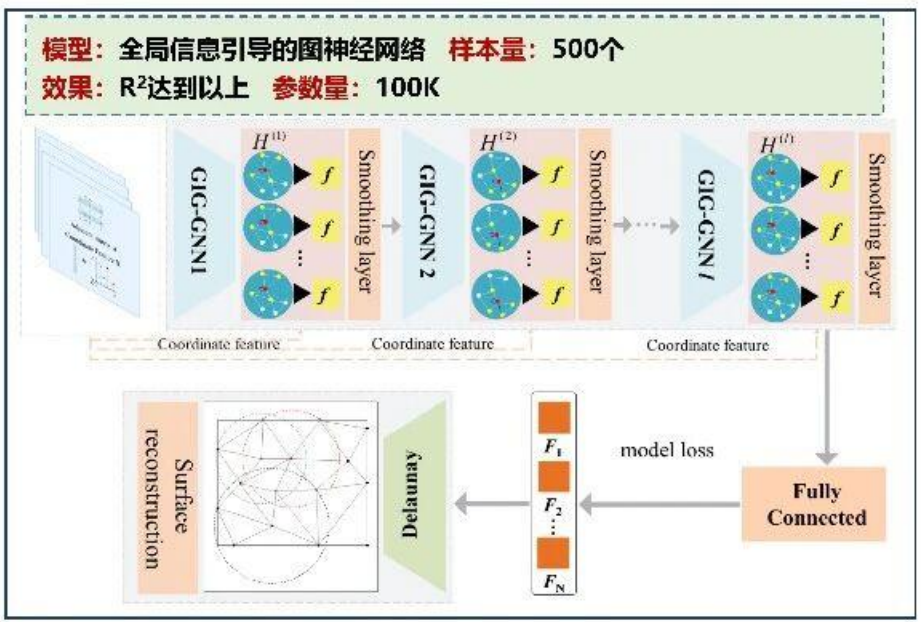
□ **小模型应用于优化设计：** 基于拉丁超立方采样构建天轮设计空间，通过FEM获取应力数据，并转换为图结构，最终利用图神经网络实现应力场的快速高精度预测。



构建样本集



数据结构转化



图神经网络

AI4M案例-3：设计优化

小模型应用于优化设计：构建样本集，计算各节点间的欧氏距离并排列，选取每个节点的若干最近邻节点，组成邻接关系集合，实现原始数据到图结构的映射与构建。

构建样本集——采样+仿真

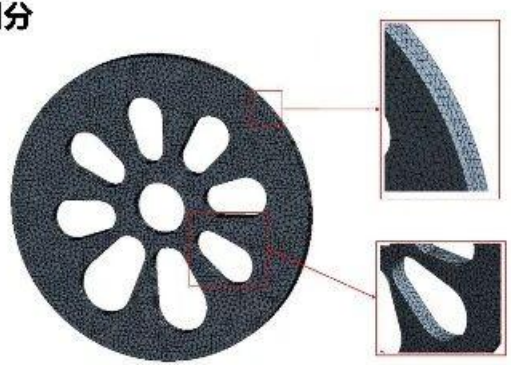
- **采样设计**

  样本数：500组

  应力：-1×10^{6} N to 1×10^{5} N

  扭矩：1×10^{6} N·m to 2.2×10^{6} N·m

- **网格划分**



- 仿真计算

  生成样本集

数据转化——构建邻接矩阵

节点间的欧式距离D_{ij}被定义为：

$$D_{ij} = \|v_i - v_j\|_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

其中，x, y, z表示节点的空间坐标

对距离进行升序排列，寻找每个节点的最邻近节点，可以被表示为

$$\mathcal{N}(i) = argmin^{k} D_{ij}$$

其中$argmin^{k}$ 表示k个最小值索引，根据$\mathcal{N}(i)$构建节点关系矩阵A。可以表示为：

$$A_{ij} = \begin{cases} 1, & \text{if } D_{ij} \le D_k \\ 0, & \text{otherwise} \end{cases}$$

AI4M案例-3：设计优化

小模型应用于优化设计：图神经网络融合全局与局部特征，通过投影矩阵分布全局信息，并利用归一化拉普拉斯矩阵聚合节点特征，实现应力场精准预测。

建立图神经网络模型

- **构建节点特征矩阵X**

$$X = [x_1, x_2, \ldots, x_N]^T, X \in R^{N \times d}$$

其中，x_i表示每个节点v_i的节点特征，d表示特征维度，N表示节点总数。

- **定义图神经网络**

定义一个图$\mathcal{G}$，图神经网络的函数映射关系可以被写为：

$$Y_G = \mathcal{G}_{GIG}(X, A, G)$$

全局特征信息G包括应力F和力矩M，Y_G表示天轮的应力场。

通过一个投影矩阵将全局信息G融入每个节点，从而实现全局特征的分布，即：

$$G_{node} = GW^{(G)} \in R^{N \times d}$$

其中$W^{(G)} \in R^{N \times d}$表示投影矩阵

为了提高感知效果并实现局部特征的聚合，采用对称归一化拉普拉斯矩阵，数学表达式为：

$$L_{sym} = \hat{D}^{-\frac{1}{2}}(A + I)\hat{D}^{-\frac{1}{2}}$$

$$\hat{D} = \sum_{ij}(A + I)$$

其中，L_{sym}表示拉普拉斯矩阵，$\hat{D}$是包含自环的度矩阵，I用于保留每个节点的自身特征。

第l+1层由非线性函数定义为：

$$H^{(l+1)} = \sigma(H^{(l)}, A, G_{node}),\ H^{(0)} = X$$

其中$\sigma(\cdot)$表示激活函数

AI4M案例-3：设计优化

□ 小模型应用于优化设计：训练采用MAE损失与Adam优化器，设置学习率衰减及数据标准化，有效提升模型收敛速度、稳定性与预测精度。

建立图神经网络模型

通过与拉普拉斯矩阵相乘体现图的结构以及节点之间的关系，因此，第$l+1$层可以进一步表示为：

$$H^{(l+1)} = \sigma(L_{sym}H^{(l)}W^{(l)} + G_{node}W^{(G)})$$

$$\sigma(\varphi) = \frac{1}{1+e^{-\varphi}}$$

□ 训练策略

在训练中，选择MAE损失函数来进行评估，损失函数可以表示为：

$$L(f,\hat{f}) = \frac{1}{N}\sum_{i=1}^{n}|f_i - \hat{f}_i|$$

f_i表示第i个节点的预测值。

采用Adam优化器执行优化任务，epoch设置为1000，batch设置为64，在训练过程中，学习率遵循衰减方案，可表示为：

$$l_r^e = \frac{l_r^0}{1+\gamma e}$$

l_r^e表示每个epoch对应的学习率，γe表示衰减率，l_r^0为初始学习率，设置为0.002。

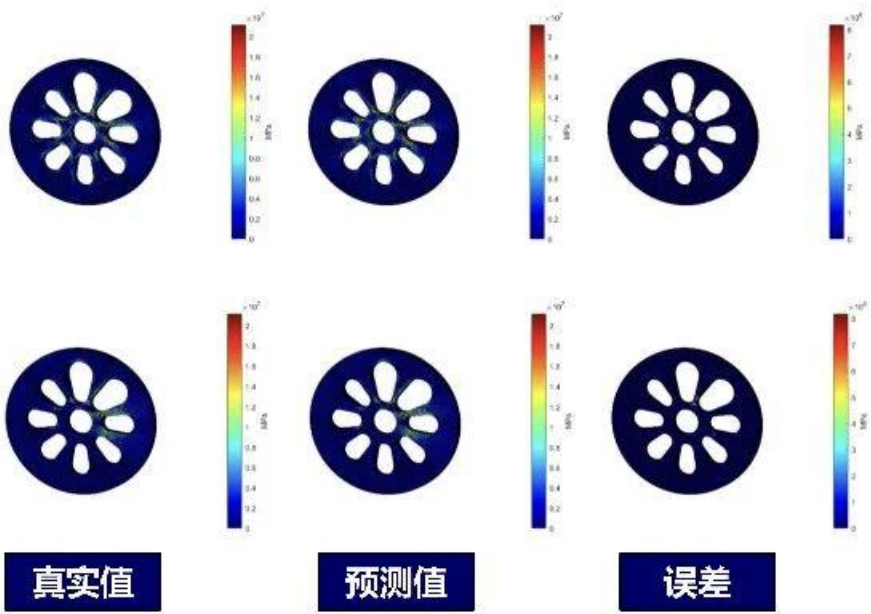
为了防止具有极大差异的特征在训练中占据主导地位，对数据进行标准化处理

$$X_{norm}^d = \frac{X_{row}^d - \mu_d}{\sigma_d}$$

μ_d和σ_d分别表示每个特征的均值和标准差。

AI4M案例-3：设计优化

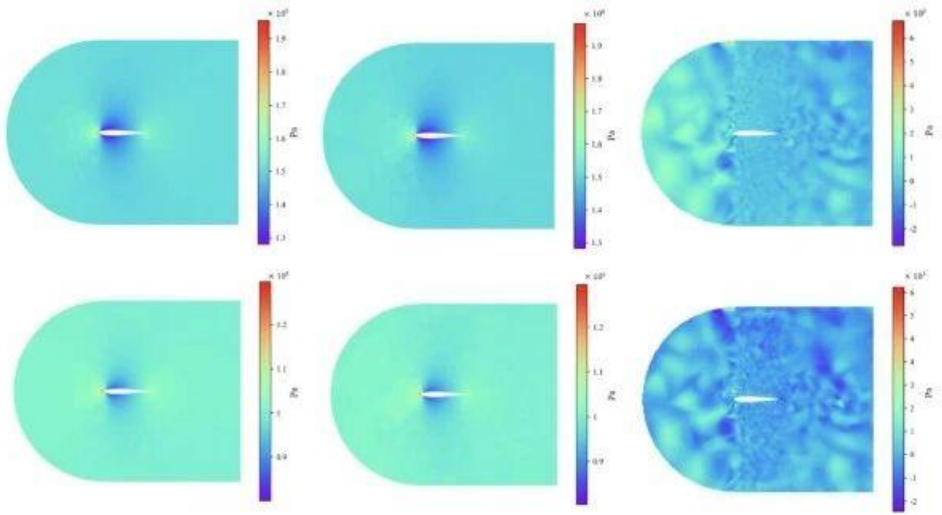
□ **小模型应用于优化设计：基于图神经网络的天轮应力场预测方法，实现了预测结果与仿真结果的高度吻合，R^2达到0.99，兼具工程实用精度与计算效率优势。**



模型	MAE	MSE	RMSE	R^2
GIG-GNN	3.0e-03	2.4e-05	0.0047	0.9974
PointNet	3.9e-03	2.5e-05	0.0050	0.9499
GCN	7.3e-03	1.3e-04	0.0110	0.9708

AI4M案例-3：设计优化

□ **小模型应用于优化设计：基于翼型流场数据的验证表明，所提图神经网络框架预测精度稳定达到$R^2>0.99$，验证了模型在复杂物理场中的强泛化能力与高可靠性。**

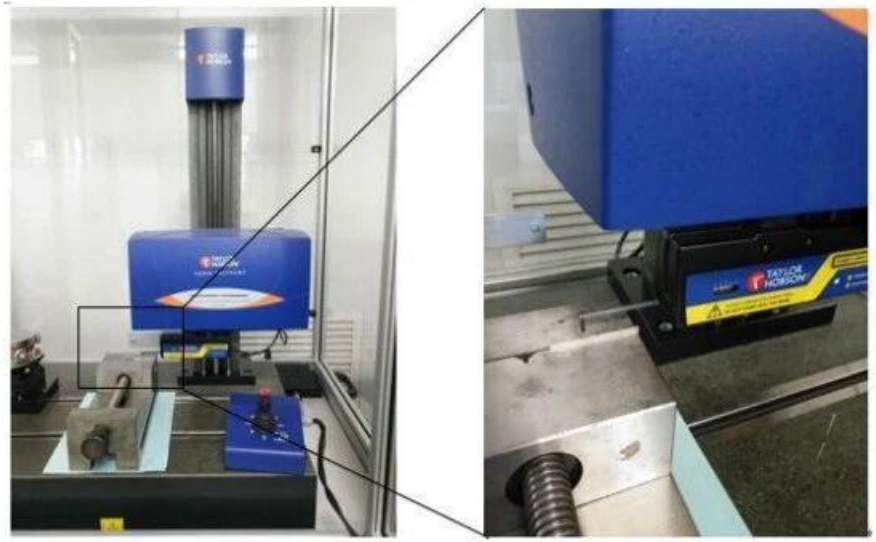


真实值　　预测值　　误差

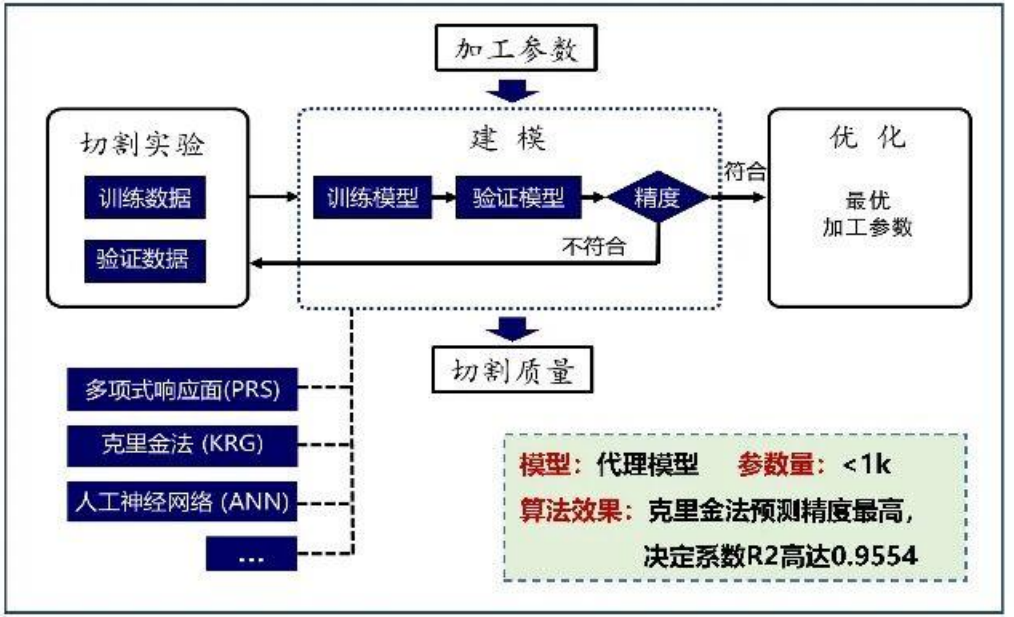
模型	MAE	MSE	RMSE	R^2
GIG-GNN	8.5e-04	1.5e-06	0.0012	0.9968
PointNet	3.7e-03	1.8e-05	0.0042	0.9748
GCN	1.3e-02	3.0e-03	0.0172	0.7348

AI4M案例-4：加工装配

□ **微模型应用于激光切割工艺优化：** 使用多种代理模型建立了激光切割质量预测模型，结合遗传算法，优化了激光功率、切割速率和辅助气体压力等主要加工参数。



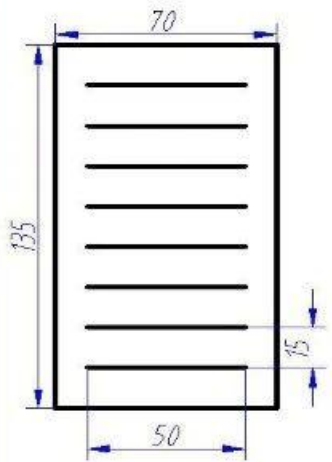
光纤激光切割不锈钢薄板实验



技术路线

AI4M案例-4：加工装配

□ **微模型应用于激光切割工艺优化：使用多种代理模型建立了激光切割质量预测模型，结合遗传算法，优化了激光功率、切割速率和辅助气体压力等主要加工参数。**



优化变量

激光功率 (550W ~ 750W)

切割速度 (0.5m/min ~ 4m/min)

辅助气体压力 (3bar ~ 11bar)

元素	C	Si	Mn	P	S	Ni	Cr
质量分数(%)	0.08	1.00	2.00	0.045	0.03	8.00~11.00	18.00~20.00

试验材料

训练

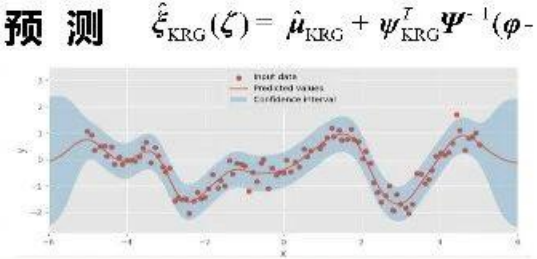
$$L(\varphi_1,\cdots,\varphi_n \mid \hat{\mu}_{KRG},\sigma^2_{KRG}) = \frac{1}{(2\pi\sigma^2_{KRG})^{n/2}|\Psi|^{1/2}}\exp\left(\frac{(\varphi-1\hat{\mu}_{KRG})^T\Psi^{-1}(\varphi-1\hat{\mu}_{KRG})}{2\sigma^2_{KRG}}\right)$$

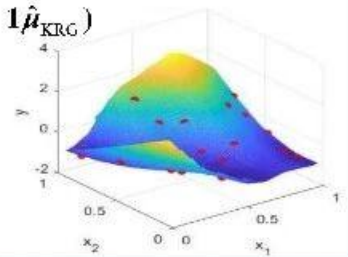
$$\frac{\partial L}{\partial \mu}=0 \quad \Rightarrow \quad \hat{\mu}_{KRG}=\frac{1^T\Psi^{-1}\varphi}{1^T\Psi^{-1}1}$$

$$\frac{\partial L}{\partial \sigma^2}=0 \quad \Rightarrow \quad \sigma^2_{KRG}=\frac{1}{m}(1\hat{\mu}_{KRG} \quad \varphi)^T\Psi^{-1}(\varphi \quad 1\hat{\mu}_{KRG})$$

预测

$$\hat{\xi}_{KRG}(\zeta)=\hat{\mu}_{KRG}+\psi^T_{KRG}\Psi^{-1}(\varphi- \quad 1\hat{\mu}_{KRG})$$

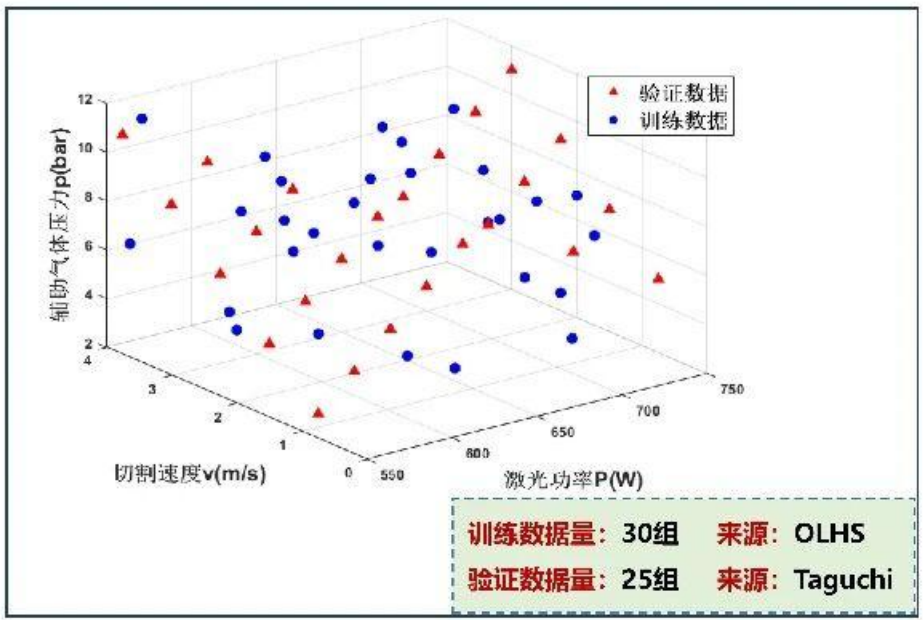




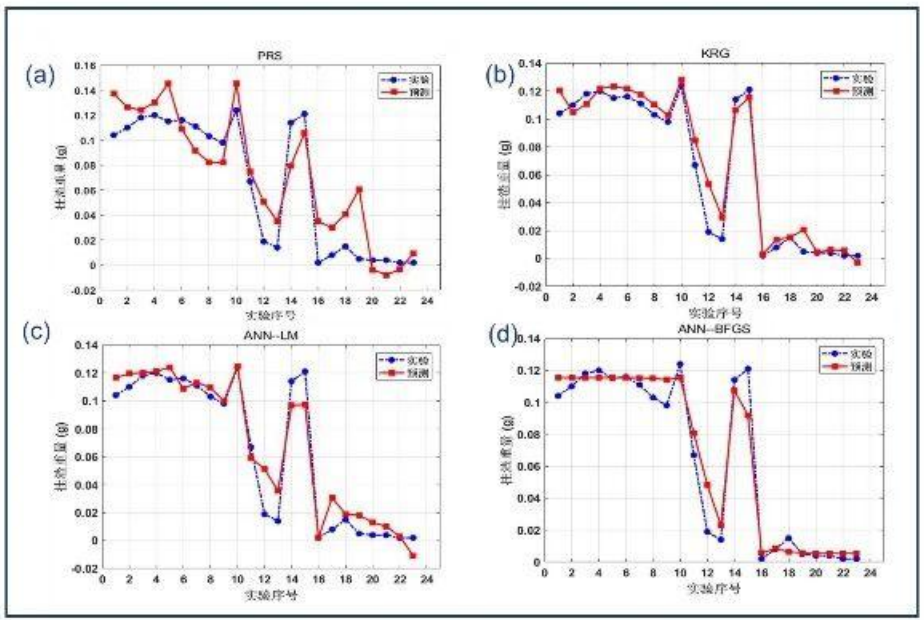
克里金法（KRG）

AI4M案例-4：加工装配

□ **微模型应用于激光切割工艺优化：** 验证样本与训练样本之比高达76%，说明克里金方法建立的挂渣量预测模型稳定可靠，可以成功用于激光切割挂渣量的预测中。



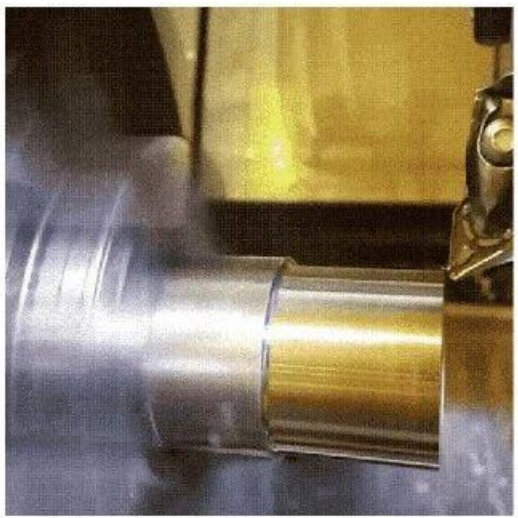
试验设计



预测值与实测值比较

AI4M案例-5：加工装配

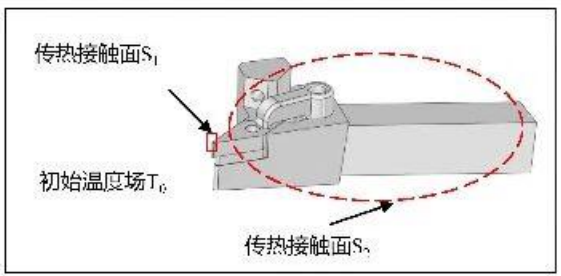
- **小模型应用于刀具温度场预测：**难加工材料在切削过程中在切削区域内极易产生强时变的局部高温和大温度梯度，造成刀具磨损和工件加工变形。



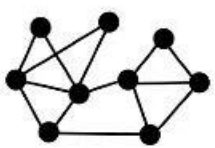
切削过程中切削区域内部温度场的精确、实时感知对刀具磨损预测、控制零件加工精度和表面质量以及切削模型精化等具有重要意义，是实现高品质复杂零件加工制造的关键。

AI4M案例-5 ： 加工装配

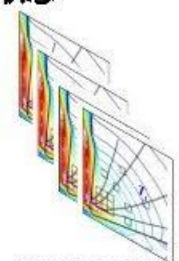
□ **小模型应用于刀具温度场预测：在图神经网络中融合热力学先验知识，实现刀具温度分布的高效预测与精准重构，为热损伤评估与加工优化提供理论与数据支撑。**



刀具温度场初始状态

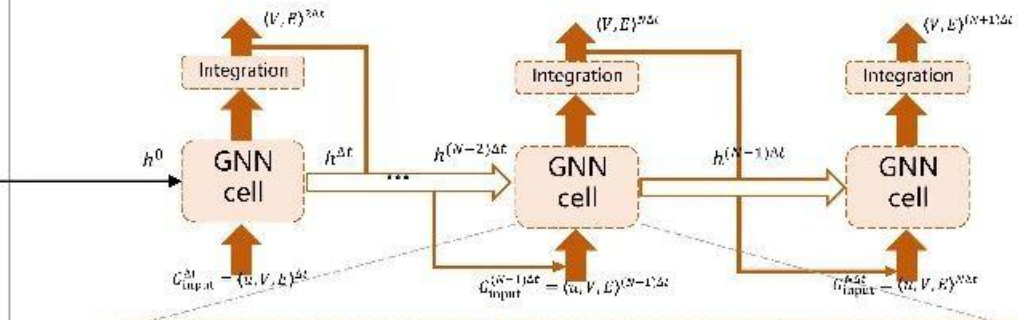


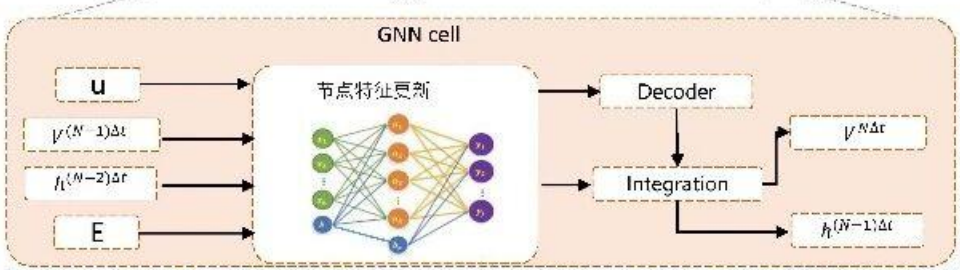
邻接矩阵



时间序列样本

输入数据





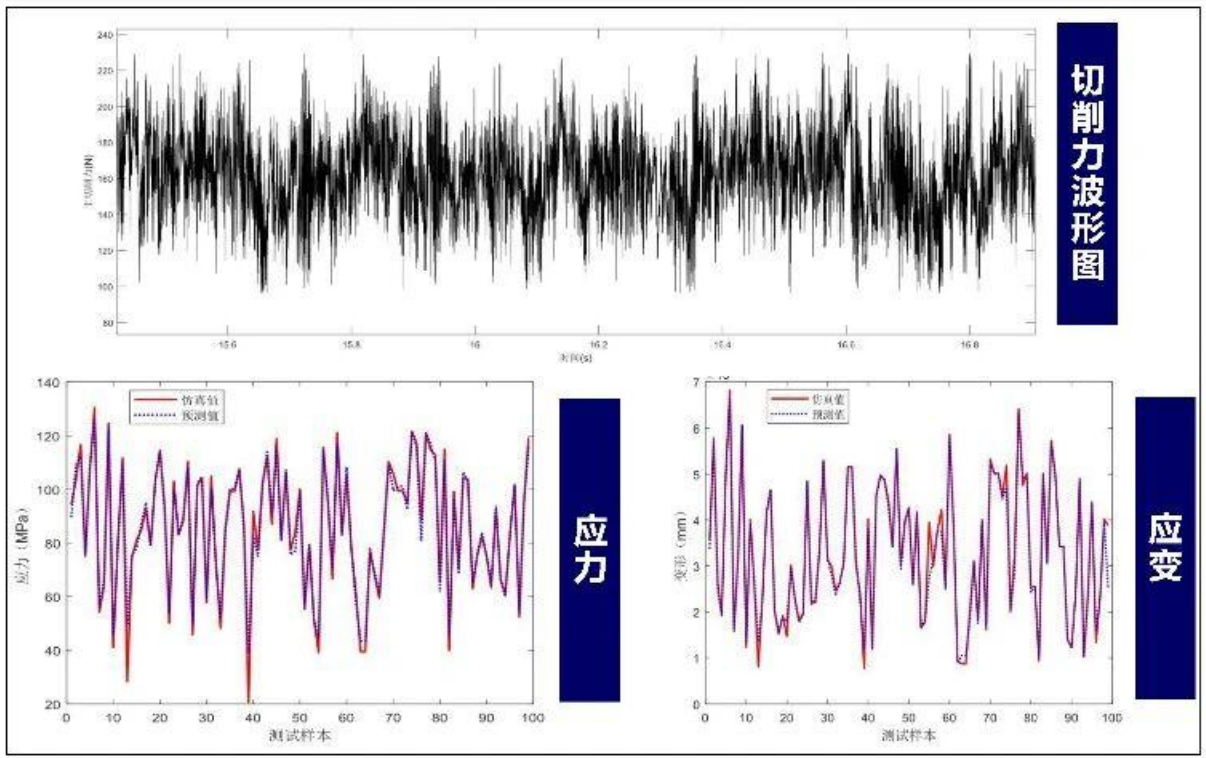
预测模型

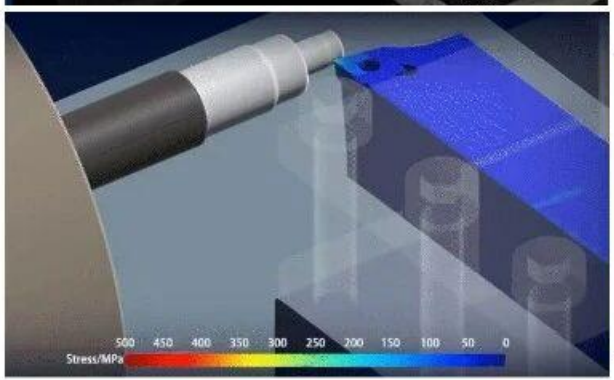
模型： 热力学规律引导的循环图神经网络 **样本：** 100个 **参数量：** 700K

效果： 耗时为传统方法的5%，精度达到90%以上

AI4M案例-5 ： 加工装配

- **小模型应用于刀具温度场预测：** 热力学信息引导的循环图神经网络实现对温度场特征的快速预测，预测精度超过90%，兼具计算效率与结果可靠性。

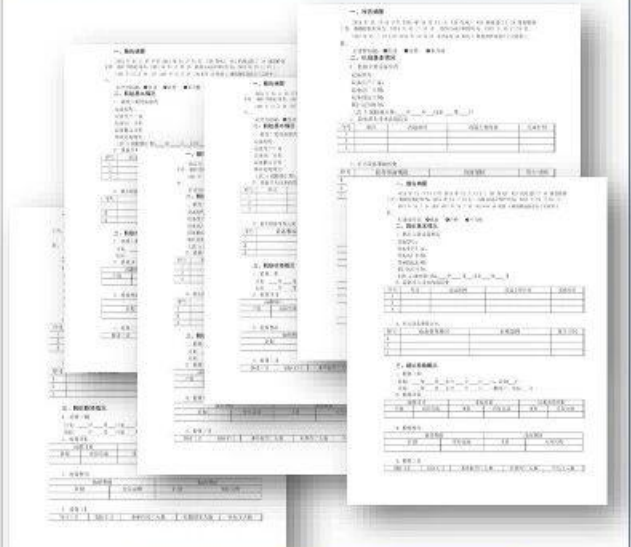




AI4M案例-6：控制运维

□ **大模型应用于核电站维修报告检索系统：采用本地大模型部署、模型局部微调和人工反馈强化学习等技术，实现核电站维修报告的理解、评分与信息推送。**

红沿河核电站

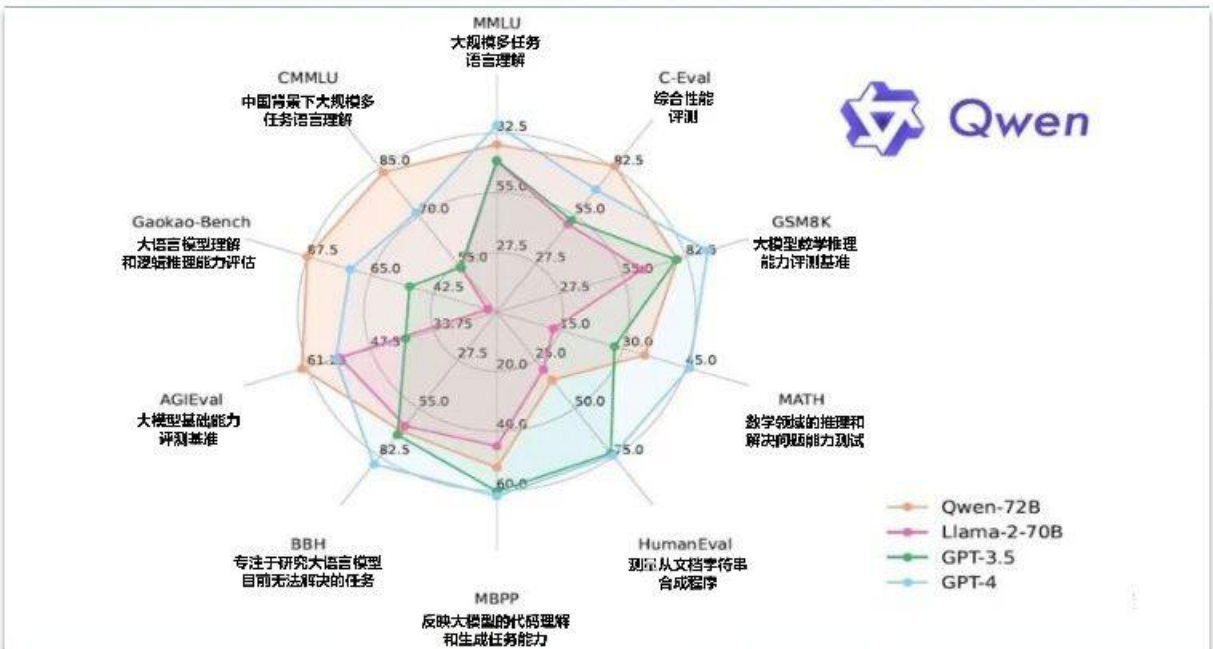


核电站维修报告

红沿河核电数据对安全性有着极高要求，但核电报告样本数量稀少

AI4M案例-6： 控制运维

□ **大模型应用于核电站维修报告检索系统：综合性能、适配性等多方面因素，选定通义千问Qwen2-72B大语言模型作为核心语言助手。**



AI大模型选型依据

模型：Qwen2-72B-Instruct-GPTQ-Int4

参数量：720B

样本量：50000个

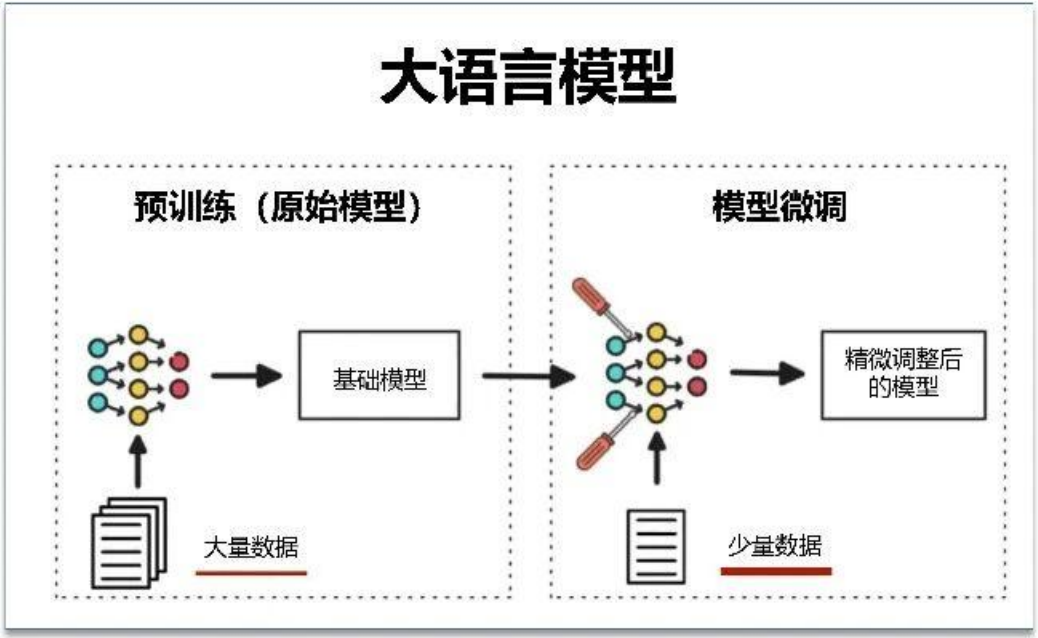
训练时长：350h

算法效果：比传统方法提高30%

通义千问

AI4M案例-6： 控制运维

- **大模型应用于核电站维修报告检索系统：运用LORA（Low - Rank Adaptation）技术对模型进行微调，让模型能更好地适配报告评估打分任务，提升其对报告指标打分的理解与推理能力。**



大模型微调技术

- 使用LoRA微调，模型原有参数冻结
- 秩（Pank）：8
- 缩放因子（Alpha）：8
- 丢弃率（Dropout）：0.1
- 学习率（Learning Rate）：1e-4
- 批大小（Batch Size）：8
- 新增训练组对：1000

参数设置

AI4M案例-6： 控制运维

- **大模型应用于核电站维修报告检索系统：RAG技术通过检索增强生成，提升模型对复杂问题的理解和回答能力，为核电报告评分提供精准的知识支持。**

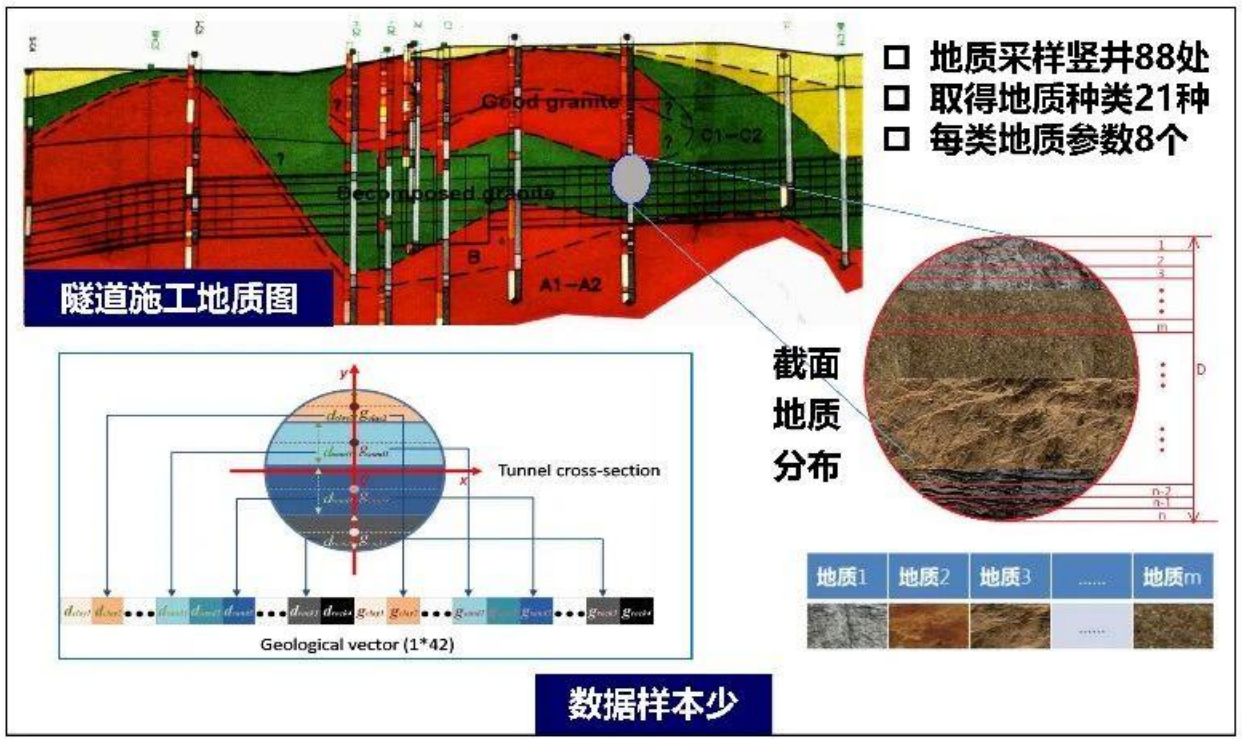
- 可快速响应批量任务，评估一份含30+指标的报告仅需十分钟左右
- 精准把握指标细节，评估准确率超90%，减少人为误差，避免主观偏见与疲劳疏忽
- 提供专业可行建议，建议正确率超90%，促进知识传承学习，为新手及学习者提供专业指导。
- Qwen2:72b通用性与可扩展性强，能适应多种类型报告的评估任务 。
- 语义理解技术精准把握意图、过滤冗余，挖掘关联知识，整合多源数据，为多领域决策提供参考。
- 知识库构建后，知识检索从小时级提速至秒级，助力企业高效获取知识，大幅提升运维决策效率与科学性，为企业稳定运行奠基。

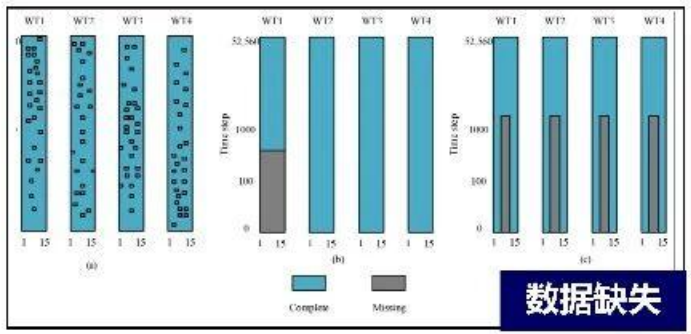
AI4M的六大挑战

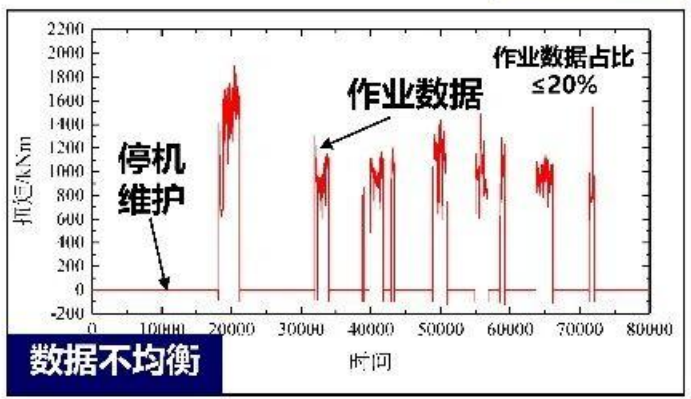

1. 实测数据少
样本数据不足
样本数据缺失
样本数据不均衡
2. 多源数据杂
传感时序数据
仿真性能数据
文本记录数据
难点与挑战
3. 物理规律缺
违背物理常识
模型外推性差
模型难以解释
4. 训练成本高
数据获取成本
数据标记成本
模型训练成本
5. 泛化能力低
过拟合、欠拟合
模型跨领域问题
长尾分布问题
6. 预测精度低
模型选择不当
参数调优不足
数据质量不佳

AI4M挑战一：实测数据少

- 受传感器安装成本、布置空间、装备工作环境、监测技术等因素的限制，实测数据呈现出样本数量少、关键数据缺失、数据不均衡等特点。



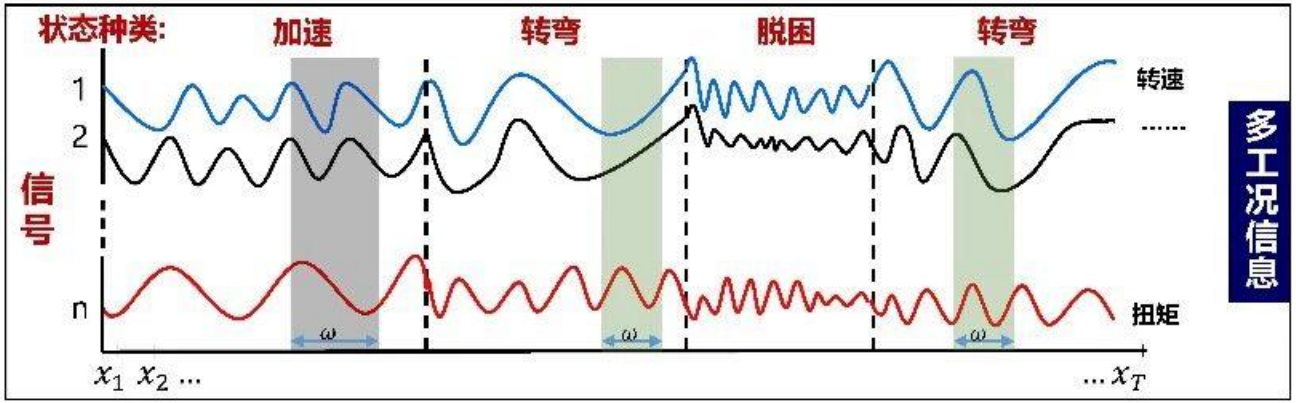




AI4M挑战二：多源数据杂

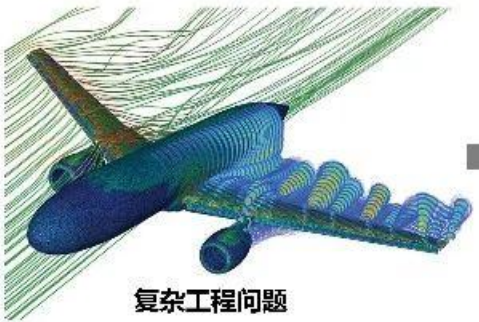
□ 设备零部件的多样性、工作场景的多样化、传感器的差异等特征，导致实测多源数据在时间、空间、性能等多维度相互交叠且属性各异。

- 运行参数
  - 土仓压力
  - 推进速度
  - …
  - 电流电压
- 地质参数
  - 岩石硬度
  - 岩石密度
- 结构参数
  - …
  - 质量、惯性
- …

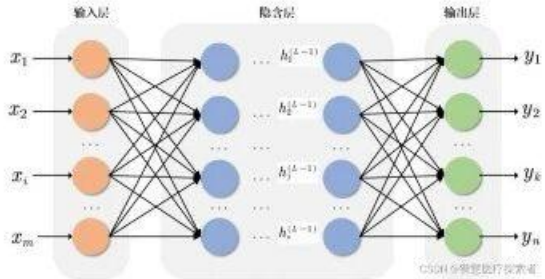


AI4M挑战三：物理规律缺

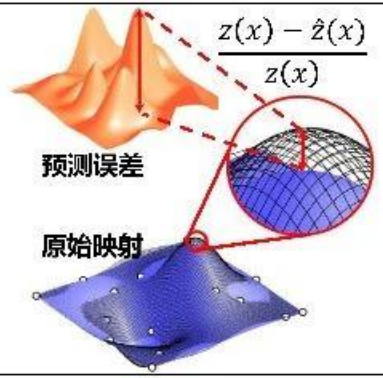
□ AI模型因完全依赖数据驱动而缺乏物理规律约束，导致其预测结果常出现较大偏差，甚至违背基本科学原理和工程常识，难以满足工程实际需求。



复杂工程问题



纯数据驱动AI模型



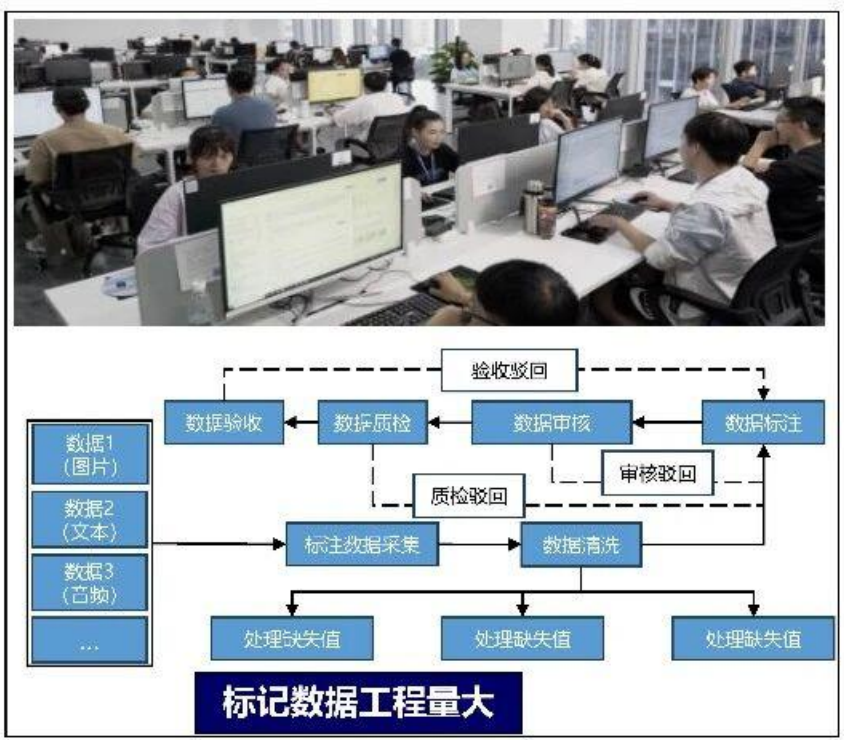
AI预测误差大

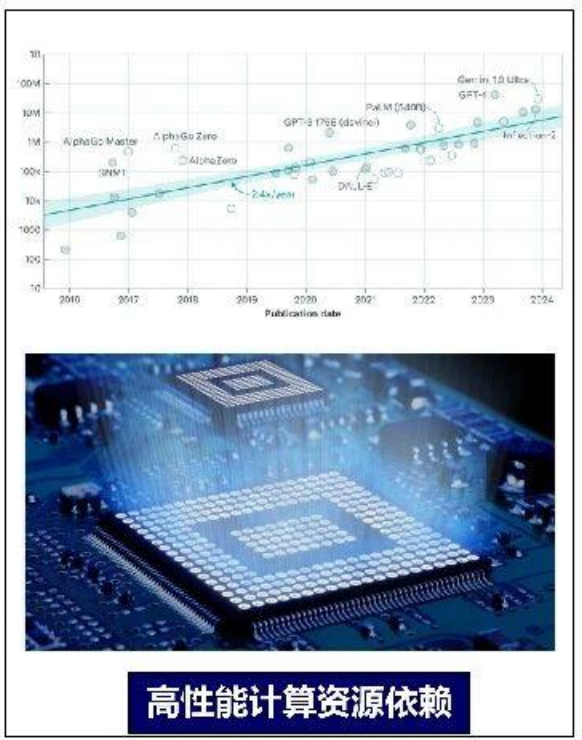
AI违背科学原理

OpenAI
SORA

AI4M挑战四：训练成本高

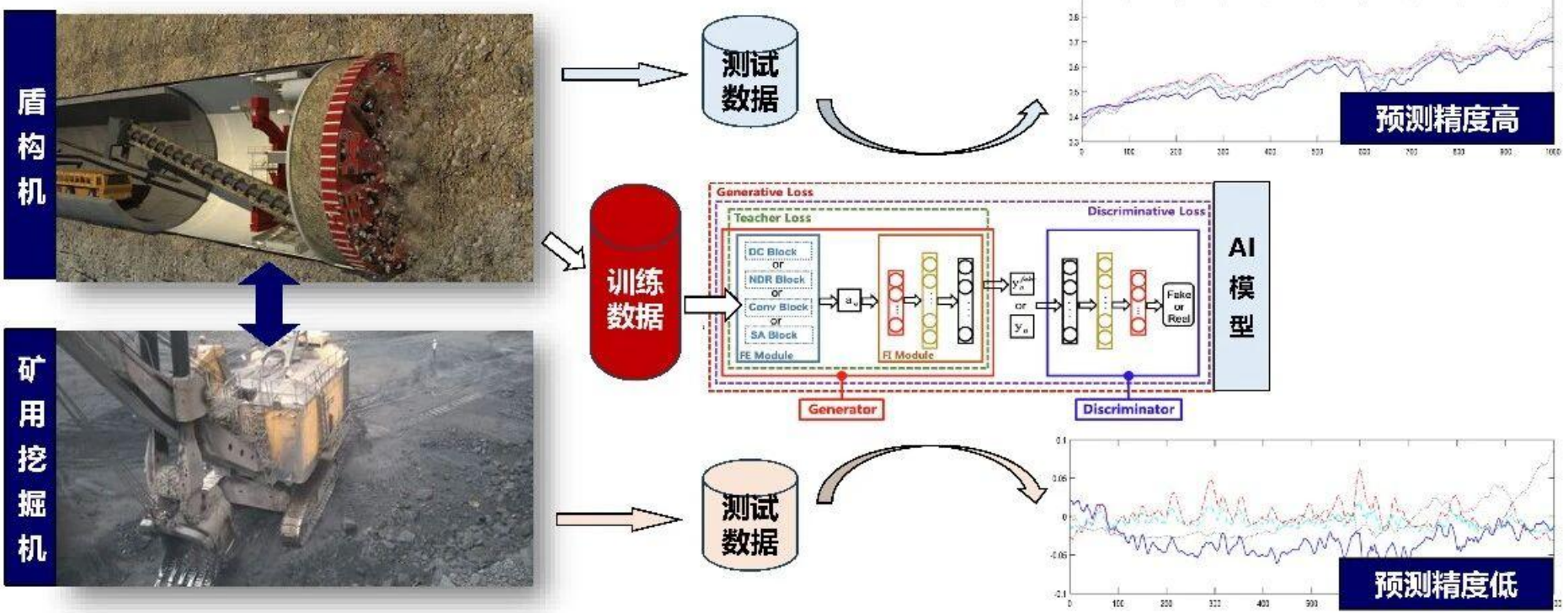
□ 工业数据获取成本高昂、数据标记工程量大、依赖高性能计算资源的三大特性，导致模型训练成本在数据获取、标记和算力适配等环节呈现指数级增长。





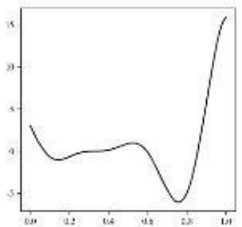
AI4M挑战五：泛化能力低

□ AI4M通常针对特定工程问题训练专用模型，模型高度依赖训练数据的工况范围和装备类型，导致单一模型在不同场景下的通用性较差、泛化能力低。



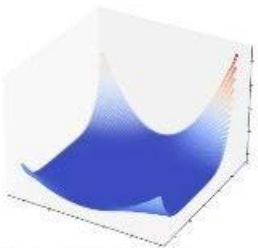
AI4M挑战六：预测精度低

□ 在面对具有高维度、强非线性等复杂问题时预测误差较大，无法运用于工程场景中

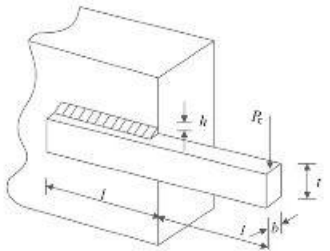


测试函数三

$f(x) = (6x-2)^2 \cdot \sin(12x-4)$

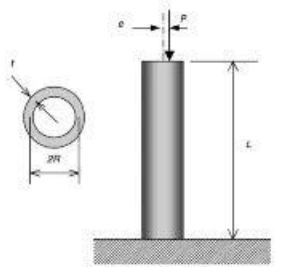


测试函数五 $f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$



优化问题一

以制造成本最小为目标的焊接梁设计

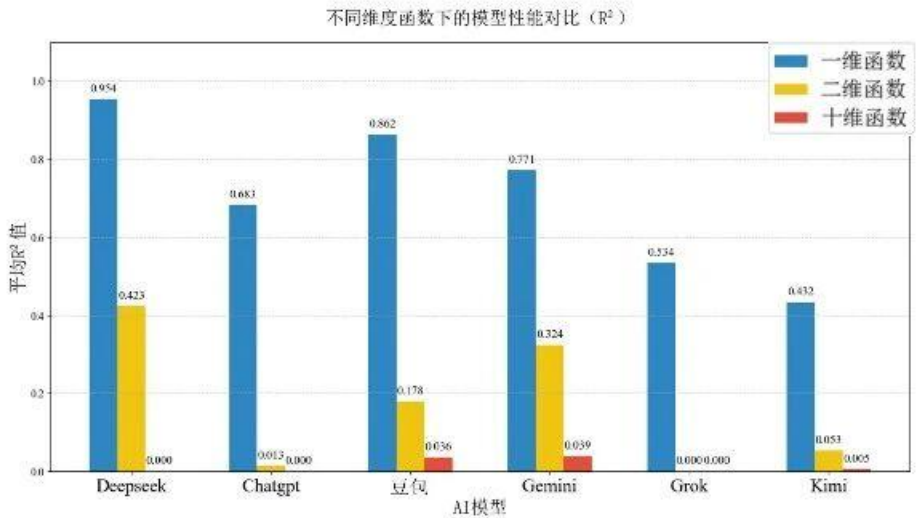


优化问题二

以质量最小为目标的立柱设计

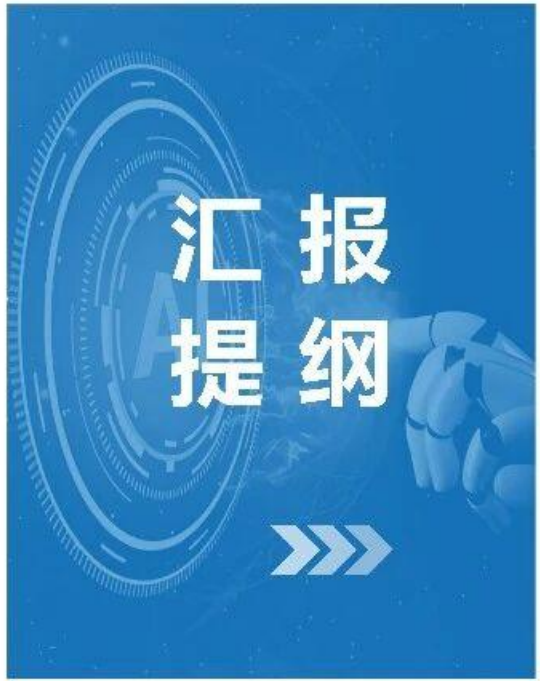
函数	样本量	DeepSeek	ChatGPT	豆包	Gemini	Grok	Kimi
测试函数1	10*n	1	1	0.9985	0.9987	0.9433	0.9293
	50*n	1	1	0.9992	0.9974	0.9943	1
	200*n	1	0.9625	0.9994	0.9997	0.4095	0.9840
测试函数2	10*n	0.9871	1	0.9932	0.8168	-0.4793	-0.6877
	50*n	1	1	0.9980	-0.3373	0.9953	0.9892
	200*n	1	-0.3659	0.9995	0.9949	0.9981	-0.3804
测试函数3	10*n	0.6717	0.0625	-0.3636	0.4050	0.4699	/
	50*n	0.9775	0.0360	0.8904	0.9980	0.9181	0.0293
	200*n	0.9472	/	0.8800	/	-0.6023	0.0293
测试函数4	10*n	0.9412	-0.0494	0.1234	0.8168	-0.4475	-21.1230
	50*n	0.9103	-0.2623	0.4551	0.2157	-0.5670	0.3544
	200*n	0.5926	0.0051	0.4227	0.8629	/	0.1225
测试函数5	10*n	-0.1583	-0.2503	-0.3329	-1.8038	-2.3373	-0.1871
	50*n	0.3613	0.0558	0.1025	0.0301	-0.9876	-0.1851
	200*n	-1.7013	-0.2210	-0.0509	0.1253	-0.1874	-0.2141
测试函数6	10*n	0.2810	0.0435	-0.4429	0.1306	-0.6965	-2.7484
	50*n	0.6136	-0.2726	-0.6511	0.3358	-0.3161	-1.2553
	200*n	0.1293	/	0.4034	0.3782	-0.1442	-0.1320
测试函数7	10*n	-0.1697	/	/	0.0271	-0.0012	0.0257
	50*n	-0.3100	-17.9687	/	-0.1627	-0.3316	-0.1491
	200*n	/	-20.3344	/	0.9765	/	/
测试函数8	10*n	-0.0727	/	0.1082	-0.0963	-0.4775	-0.6877
	50*n	-0.1139	-3.9337	-0.3250	-0.0809	-0.1018	-0.2026
测试函数9	10*n	-1.0879	/	-0.4591	-0.0810	-1.4716	/
	50*n	-0.9705	-0.3502	/	-1.0289	-0.2972	-0.3223

$R^2>0.8$ $R^2<0.5$或无法预测



- **大语言模型对一维测试函数的预测性能较好**
- **对二维、十维测试函数预测时表现出较差的性能**
- **对于面向实际场景的复杂约束优化问题表现出明显的性能局限**

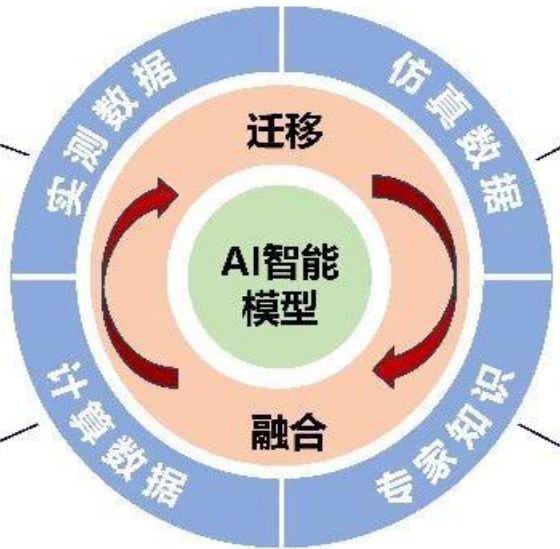
*大语言模型赋能优化设计初探，机械工程学报（在投）

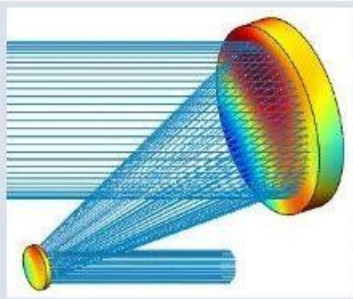


“测-算”多源域数据的内在关联与迁移机制

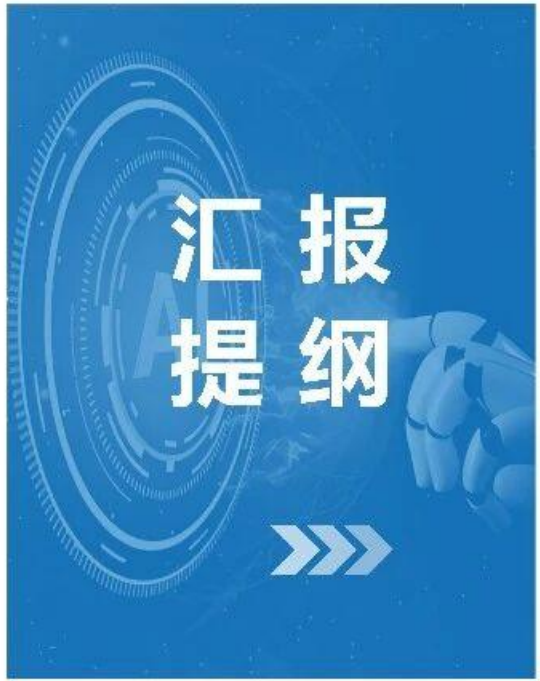
- 如何揭示现场实测与计算仿真等多源域数据的内在关联关系，构建物理信息引导的多保真度、可迁移AI智能代理模型，实现重大装备多源域数据的融合与关键性能的高效、精准预测。

$$D(r)\nabla^2\nabla^2 w(r,\theta)+D'(r)\left[2\frac{\partial^3 w(r,\theta)}{\partial r^3}+\frac{2}{r}\frac{1}{\mu}\frac{\partial^2 w(r,\theta)}{\partial r^2}-\frac{1}{r^2}\frac{\partial w(r,\theta)}{\partial r}+\frac{2}{r^2}\frac{\partial^3 w(r,\theta)}{\partial r\partial\theta^2}-\frac{3}{r^3}\frac{\partial^2 w(r,\theta)}{\partial\theta^2}\right]$$
$$-D''(r)\left[\frac{\partial^2 w(r,\theta)}{\partial r^2}+\frac{\mu}{r}\frac{\partial w(r,\theta)}{\partial r}+\frac{\mu}{r^2}\frac{\partial^2 w(r,\theta)}{\partial\theta^2}\right]=q(r,\theta)$$





“测-算”多源数据融合的优势：优势互补，突破“算不快、测不准”的难题



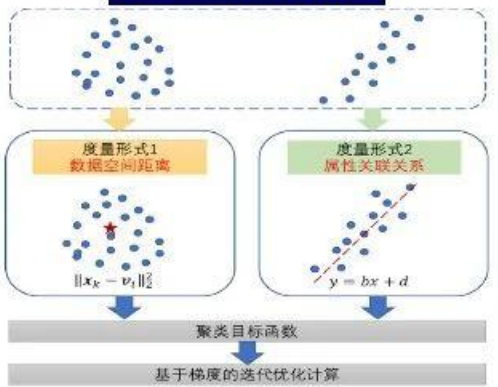
AI4M方向一：实测数据处理技术

□ 受工作环境、人为操作、传感器材料等外部和内部因素的影响，实测数据包含大量噪音，需对数据进行高效处理以提取蕴含的关键特征。

传感器最优布局技术

缺失值填补技术

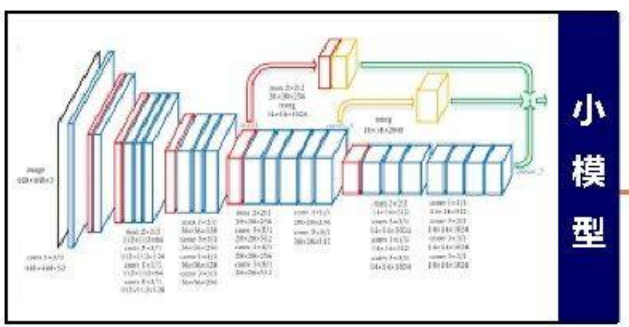
数据聚类技术

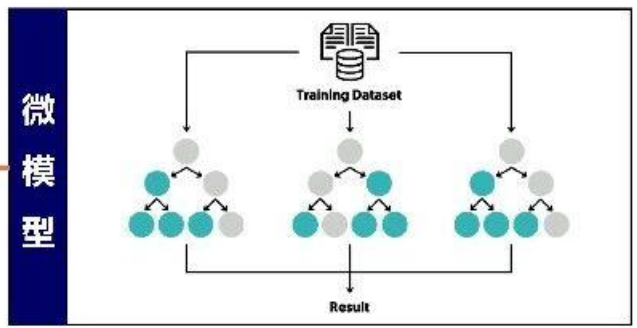


- 多类型传感器最优布局技术
- 基于最优布局的数据完整获取
- 低时延数据在线特征提取与离线数据的精准表征

AI4M方向二：大小微模型的融合技术

- 大小微模型存在各自的优势，结合大模型的泛化推力、小模型的领域适应和微模型的快速响应能力，可实现模型性能与计算效率在复杂场景下的平衡。

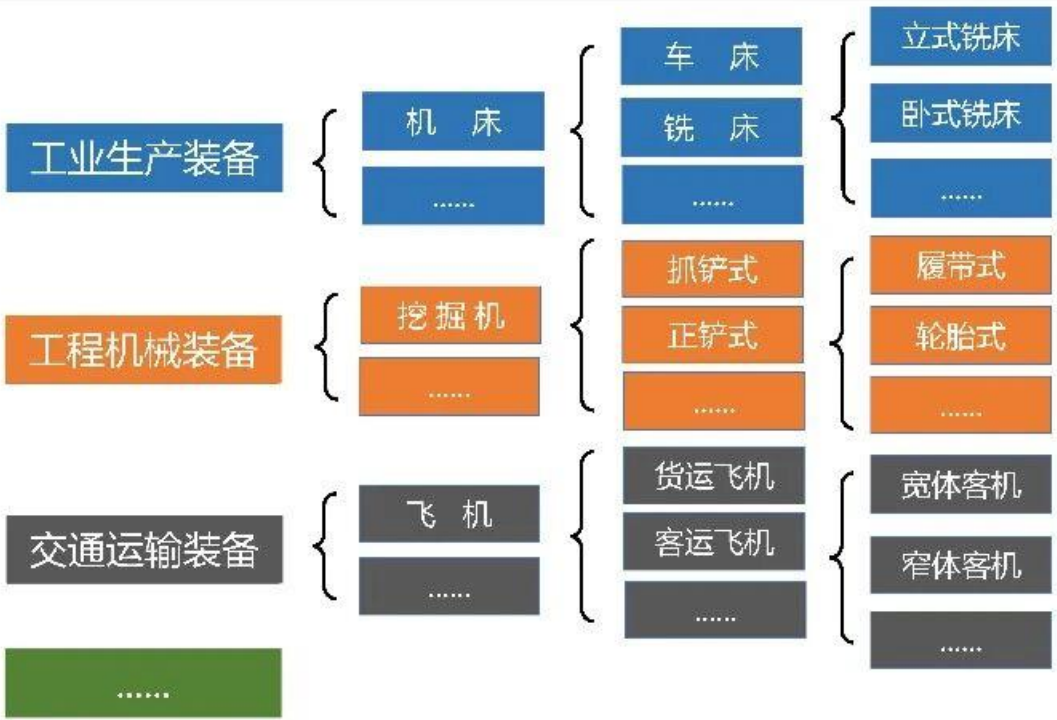




- 工业变保真模型融合技术
- 工业知识蒸馏与特征迁移
- 多种类工业模型分层协作

AI4M方向三：领域专业小模型开发

- 针对特定场景、特定任务、特定对象，开发结构简洁、计算高效、具备专业知识的小模型仍然是“当务之急”，才能实现系统从边缘端开始的智能化。




立式铣床


卧式铣床

...


履带正铲式挖掘机


轮胎正铲式挖掘机

...


宽体客运飞机

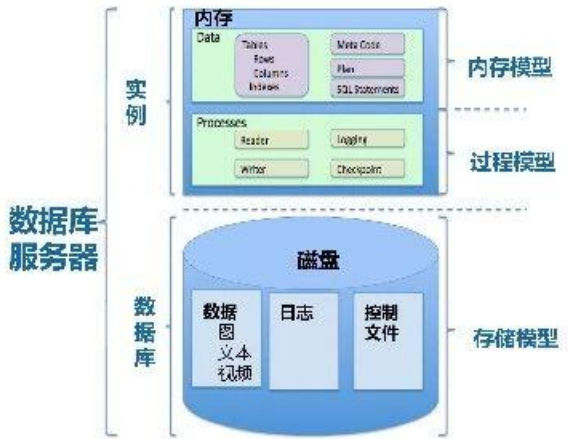

窄体客运飞机

...

- 专业领域的物理知识引导建模（PINN）
- 端到端的小、微模型部署与进一步优化

AI4M方向四：领域有效数据库集成

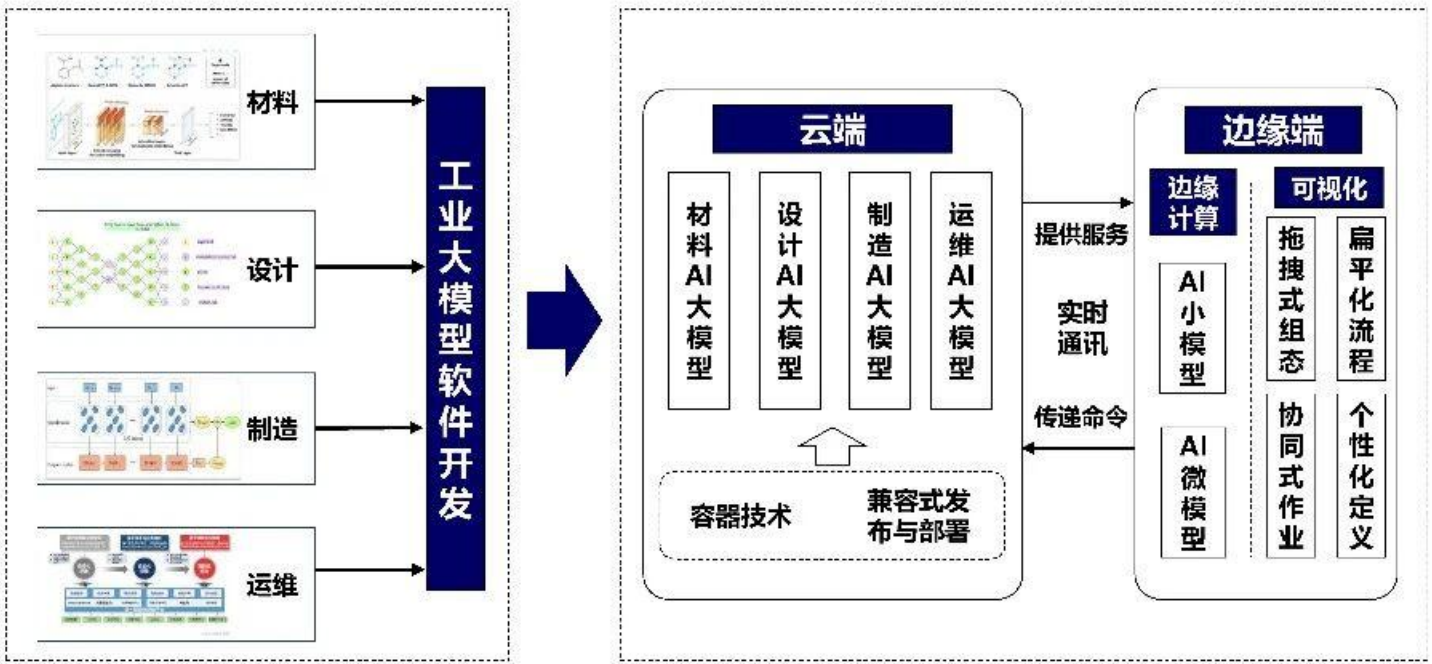
□ 装备多源数据的集成在于对设计、制造、运维等数据的整合、关联、标准化管理，打破信息孤岛，形成智能协同，支持跨领域协作以应对复杂场景需求。



- 材料-设计-制造-运维全生命周期数据库
- 数据库共享与加密方法与平台

AI4M方向五：云边结合的大小微模型软件开发

□ 软件开发是装备智能化升级的核心架构之一，云边结合的软件开发通过优化模型部署架构和计算资源分配，实现智能系统的“大脑+神经末梢”协同机制。



- AI模型在云端与边缘端的自适应切分策略
- 专用边缘计算的AI小/微模型建模技术

AI4M方向五：云边结合的大小微模型软件开发

□ 自主开发了云边结合、数据驱动的优化设计软件DADOS (以微模型为基础)

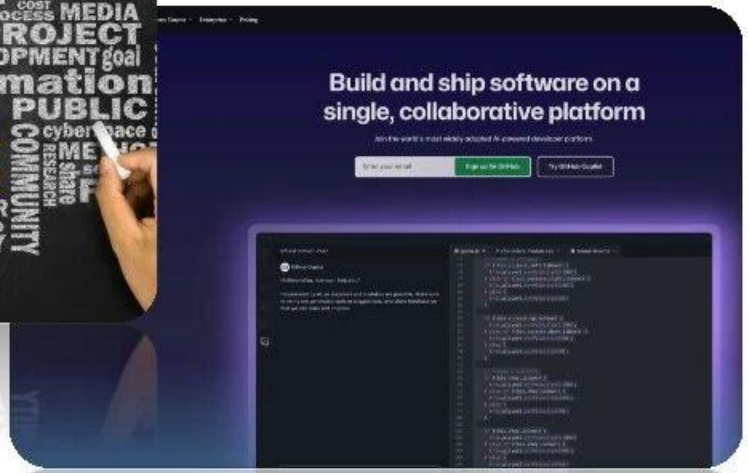
DADOS

A Cloud-based Data-driven Design Optimization System

网址：www.dados.com

AI4M：共享与开放

- 源代码的开放共享，不仅推动模型技术快速演进，也能激发协同创新与产业生态的正向循环，助力先进制造智能化走深走实，同时确保研究成果的真实可信。



AI4M政策：约束与务实

□ 强调求真务实的科研作风，充分尊重技术型人才的创新自主权，切实推动研究落地生根、开花结果，促进科研从“纸面”走向“地面”，打造真正的科研强国。

明确系统功能边界和安全响应措施，**不得进行夸大和虚假宣传**

deepseek

xiaomi

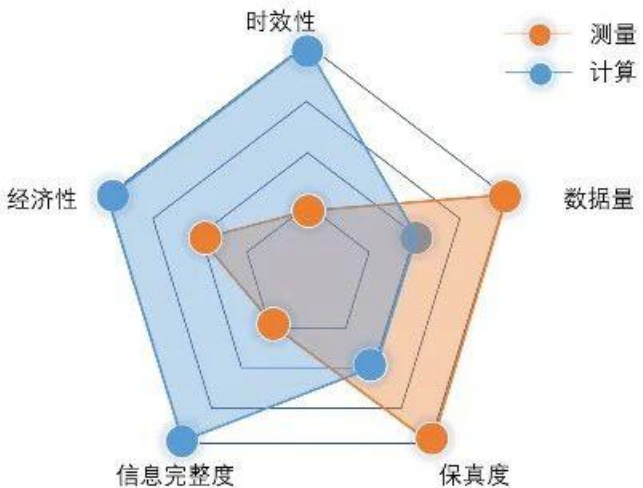
HUAWEI

UNITREE宇树

大疆创新

AI4M：算-测融合

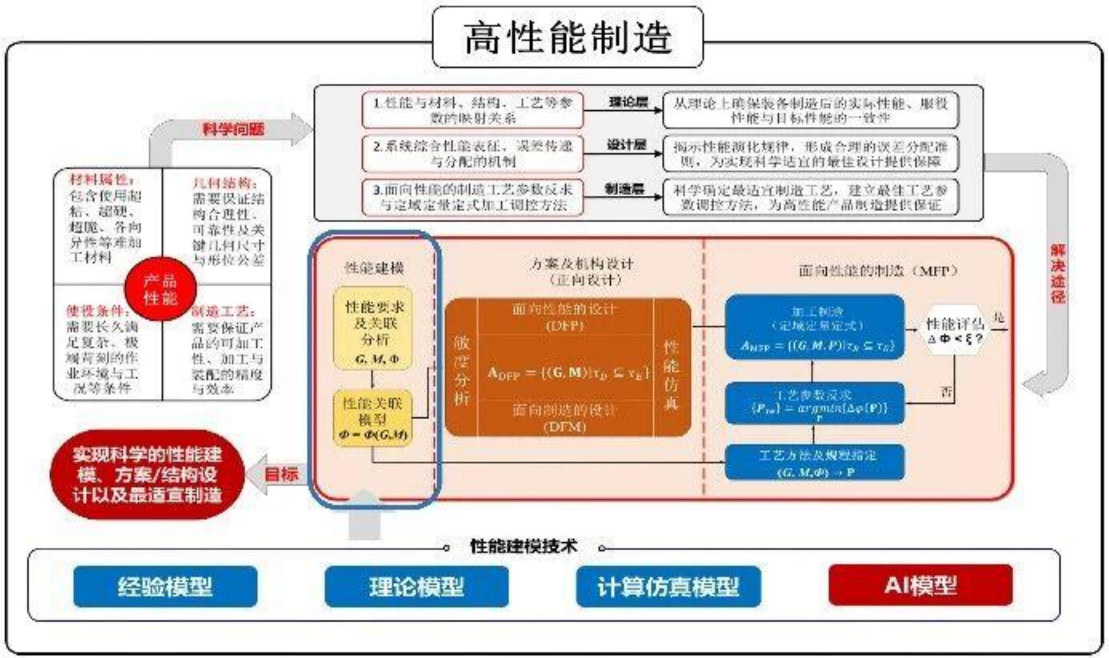
□ “算”与“测”的深度融合，机理与数据的协同，如同人类双腿需协调迈步，唯有双轮驱动、优势互补，方能突破技术瓶颈，实现制造技术的跨越式发展。



物理测试与计算科学同进步

AI4M：性能建模

□ AI模型是性能建模重要手段之一，有效弥补机理模型的不足，实现数-模联动。



郭东明：高性能制造

The sciences do not try to explain, they hardly even try to interpret, **they mainly make models**. The justification of such a mathematical construct is solely and precisely that it is expected to work.
——John von Neumann