Frontiers in Computer Science and Technology

计算机科学与技术前沿

Spring 2024

Tailin Wu, Westlake University

Website: ai4s.lab.westlake.edu.cn/course

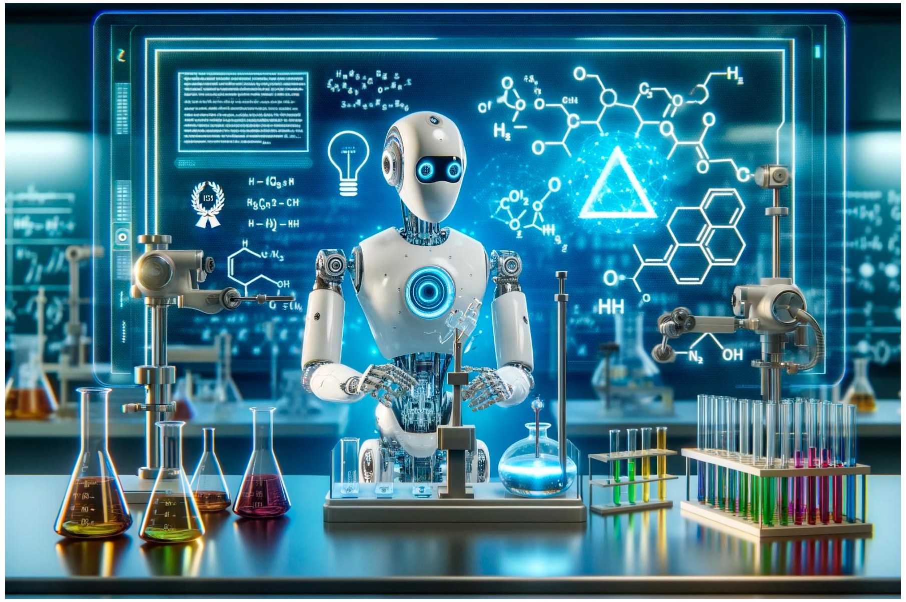


Image from: SciTechDaily

Outline

- **Course logistics**
- Why study this course?
- Course introduction
    - Tasks
    - Neural Architecture
    - Learning paradigm
    - Application in AI and Science

Course logistics

We are going to learn important branches of **AI techniques** and their **application in science**, both the **fundamentals** and **the frontiers**:

AI techniques:

- Deep Learning
- Generative models
- Foundation models
- Reinforcement learning
- Robotics
- Computer vision and autonomous driving
- Graph Neural Network
- Evolutionary Machine Learning and Multi-objective Optimization

AI for Science:

- AI + PDEs/scientific computing
- AI + life science
- AI + materials science

Course timeline

All teachers are PIs in the AI direction in Westlake University

#	Topic	Date	Teacher
1	Course introduction	Mon, 2/26	Tailin Wu
2	Frontiers in Deep Learning	Mon, 3/04	Tailin Wu, Tao Lin
3	Frontiers in Generative Modeling	Mon, 3/11	Tailin Wu
4	Foundation models	Mon, 3/18	Zhengzhong Lan
5	Introduction to Reinforcement Learning	Mon, 3/25	Tailin Wu
6	From AlphaGo to Robotics: Reinforcement Learning Applications	Mon, 4/01	Donglin Wang
7	Computer Vision and Autonomous Driving	Mon, 4/08	Kaicheng Yu
10	Frontiers in Graph Neural Networks	Mon, 4/29	Tailin Wu
11	AI + Life Sciences	Mon, 5/06	Ziqing Li
12	Evolutionary Machine Learning and Multi-objective optimization	Mon, 5/13	Yaochu Jin
13	AI + PDE/Scientific computing	Mon, 5/20	Tailin Wu
14	AI + Materials science	Mon, 5/27	Tailin Wu

*For project timeline see the slides later.

Course arrangement

Time: Every Monday 8:50am – 10:35am

Typical time split:

- 8:50 – 9:35am (45min): First part
- 9:35 – 9:40am (5min): Discussion
- 9:40 – 10:25am (45min): Second part
- 10:25 – 10:35am (10min): Discussion

No prerequisite for the course

But would be good to have a basic understanding of neural network, its training, and write basic neural network with PyTorch

- If you don't have any of the above background, that is fine. You only need **half a day** to master them, using the materials provided in this lecture.

Project guidance

Choose a **problem related to your research**, and **use AI to solve it.**

Team size: 1-3 people, encouraging **interdisciplinary** collaboration

Course project design @ midterm:

- Give a presentation (10min) that formulates the problem for the 5 questions, each with one slide:
  1. What is the problem?
  2. Why is it important
  3. Why is it hard?
  4. What is the limitation of the prior method?
  5. What is the main component of the proposed method?

  Then detail the proposed method (3-4 slides) that uses an AI technique to solve the problem

#	Topic	Date	Teacher
8	Course project design (1)	Mon, 4/15	Tailin Wu
9	Course project design (2)	Mon, 4/22	Tailin Wu

Project guidance

Course project report and discussion @ final:

Give a presentation (15min) that

- Formulates the problem in terms of the 5 questions before, each with one slide.
- Then detail the proposed method (3-4 slides) that uses an AI technique to solve the problem.
- Then report the main experiment results (3-4 slides)

Submit a project report that summarize the project.

#	Topic	Date	Teacher
15	Course Project Reporting and Discussion (1)	Mon, 6/03	Tailin Wu
16	Course Project Reporting and Discussion (2)	Tue, 6/11	Tailin Wu

Course grading

Assessment Criteria	Percentage
Attendance	5%
Project proposal and discussion	30%
Project conclusion presentation	30%
Project conclusion report	35%

Grade	Assessment Standard
A	90-100 points
B	80-89 points
C	70-79 points
D	60-69 points
F	Below 60 points

The grading will be **generous**. The important thing is that you learn **useful AI techniques** that you can use for your own **research**.

Other logistics

Email: wutailin@westlake.edu.cn

Website: https://ai4s.lab.westlake.edu.cn/course

Questions?

Outline

- Course logistics
- **Why study this course?**
- Course introduction
  - Tasks
  - Neural Architecture
  - Learning paradigm
  - Application in AI and Science

Why study this course?

Self-introduction:

Department & lab

Research interest

Familiarity (1-10) with deep learning

Why study this course?

If you are from science/engineering background, not so familiar with AI, you will:

- Know major AI techniques, their application areas, and limitations
- Able to use state-of-the-art AI techniques for your own research

If you are from AI background, you will:

- Learn state-of-the-art AI techniques in different subfields
- Know the open research problems for each subfield
- Collaborate and explore interdisciplinary research

Outline

- Course logistics
- Why study this course?
- **Course introduction**
  - Tasks
  - Neural Architecture
  - Learning paradigm
  - Application in AI and Science

Course introduction

Tasks

- Classification/ regression
- Simulation
- Inverse design/ inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

Learning paradigm

- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Course introduction: tasks

Tasks

- Classification/ regression
- Simulation
- Inverse design/ inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

Learning paradigm

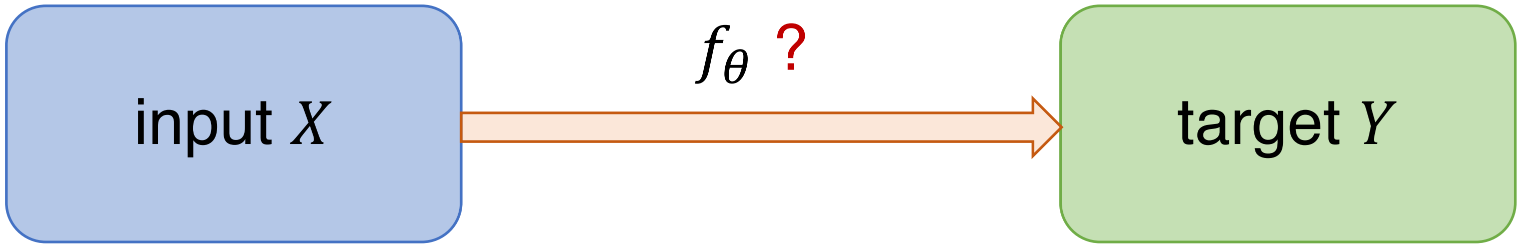
- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Task 1: Classification & regression

- image
- video
- graph
- time series
- natural language
- …



classification

- label (discrete)
- scalar/tensor (continuous)

regression

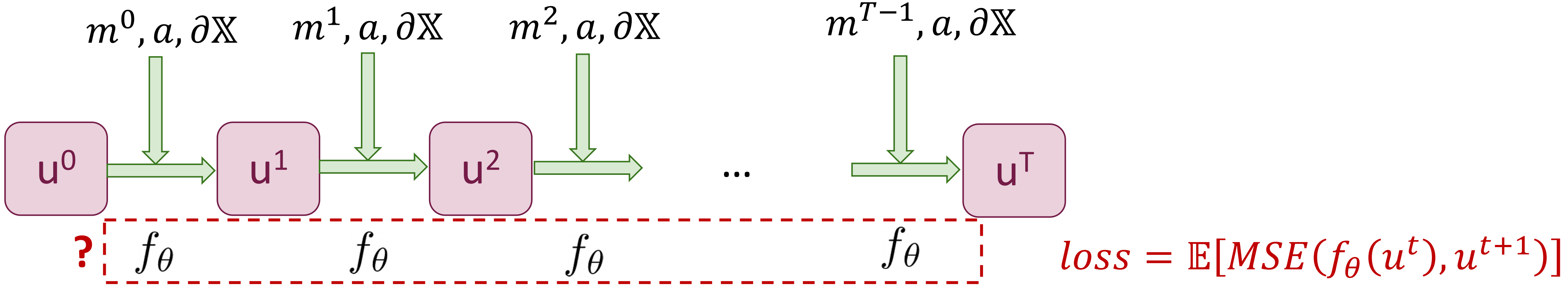
Given many examples of (X, Y) pairs, learn a neural network (NN) f_θ that minimizes the prediction loss:

$$\theta^* = \mathrm{argmin}_\theta \, \mathbb{E}_{(X,Y)\sim P(X,Y)}[\ell(f_\theta(X), Y)]$$

f_θ: neural network to be learned
ℓ: loss function

Task 2: (Learning) simulation

Goal: learn the mapping f_θ from u^t to u^{t+1}:



u^t: original **state**（状态）of the system. Can be a graph (e.g., mesh, particle-based systems, molecules), a tensor, or an infinite-dimensional function $u(t, x)$ as solution to a PDE

f_θ: **neural surrogate models**（神经网络代理模型）

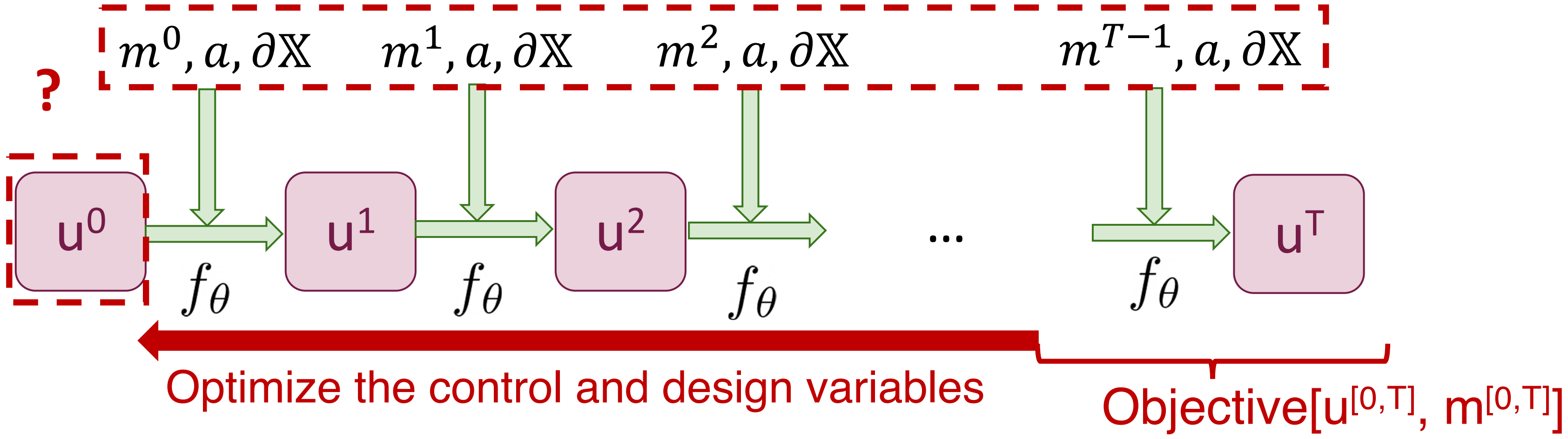
m^t: **external control**（外界控制）

a: **static parameters**（静态参数） of the system that does not change with time (e.g. parameters of PDE, spatially varying diffusion coefficient)

$\partial\mathbb{X}$: **boundary condition**（边界条件） of the system

PDE: partial differential equation
ODE: ordinary differential equation

Tasks 3 & 4: Inverse design, inverse problem, and control



u^t: original **state** of the system. Can be an infinite-dimensional function $u(t, x)$ as solution to a PDE, or a graph (e.g., mesh, particle-based systems, molecules)

f_θ: neural surrogate models

m^t: external **control（外界控制）**

a: **static parameters（静态参数）** of the system that does not change with time (e.g. parameters of PDE, spatially varying diffusion coefficient)

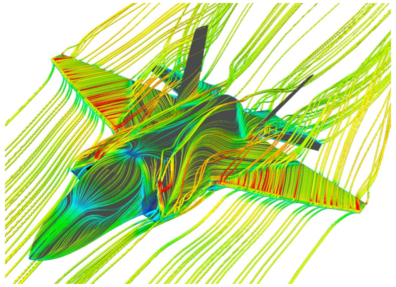
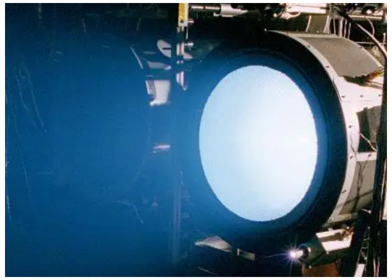
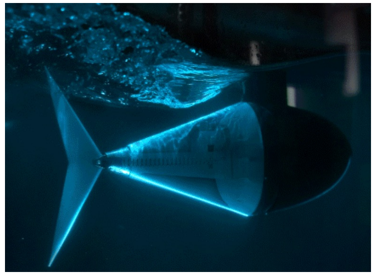
$\partial\mathbb{X}$: **boundary condition（边界条件）** of the system

} control（控制）

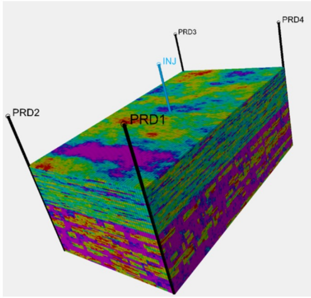
} inverse design（反向设计）

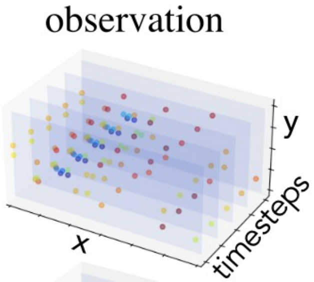
Tasks 3 & 4: Inverse design, inverse problem, and control

- **Inverse design:** boundary $\partial\mathbb{X}$, initial condition u^0, parameter a to **optimize design objective**: plane design, rocket shape, underwater robot shape

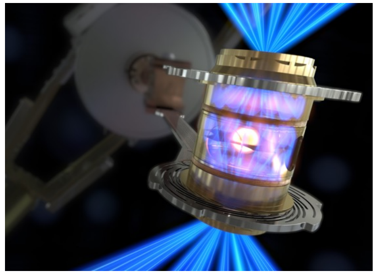




- **Inverse problem : infer initial condition** u^0 or parameter a to **match prediction with observation**





- **Control: optimize control** m^t to **optimize control objectives**: controlled nuclear fusion, robotics




Tasks 2 & 3: Steady-state simulation and inverse design

Simulation:

$$a, \partial\mathbb{X} \xrightarrow{f_\theta\ ?} u$$

parameter and boundary

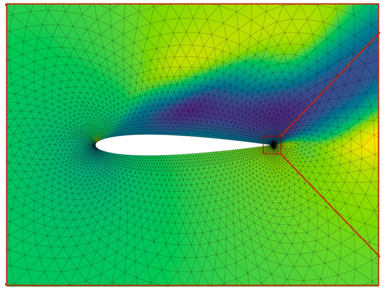
steady state of the system

Inverse design/inverse problem:

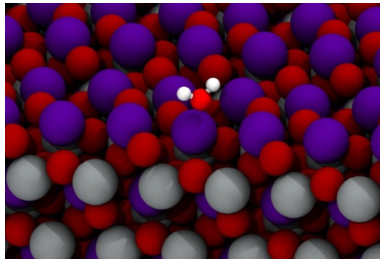
$$?\ a, \partial\mathbb{X} \xrightarrow{f} u$$

parameter and boundary

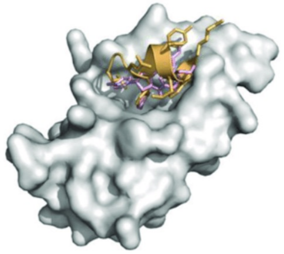
steady state of the system



Aerodynamics simulation



Materials design



Protein design

Course introduction: Neural architecture

Tasks

- Classification/ regression
- Simulation
- Inverse design/ inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

Learning paradigm

- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Neural architecture: overview

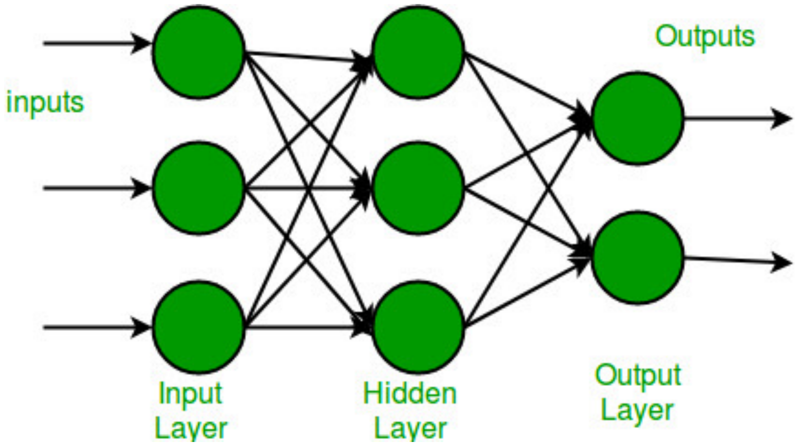
The choice of neural architecture depend on the data structure:

Data structure	Examples	Suitable neural architecture
Vector	simple vectors	Multilayer Perceptron (MLP)
Graph	molecules, irregular mesh	Graph Neural Network (GNN)
Grid	image, videos	Convolutional Neural Network (CNN)
Sequence	time series, natural language	Transformer

Neural architecture 1: Multilayer Perceptron (MLP)

f_θ

input $x \in R^d$
(vector)

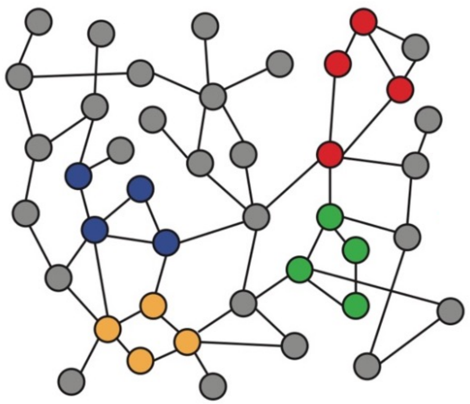


prediction $\hat{y}$

An MLP f_θ with n layers: $\hat{y} = W_n\sigma(\ldots\sigma(W_2\sigma(W_1x + b_1) + b_2) + b_n$

W_i: weight matrix to be learned
b_i: bias vector to be learned
σ: (nonlinear) activation function, e.g., ReLU, softplus, ELU

Neural architecture 2: Graph Neural Networks (GNN)

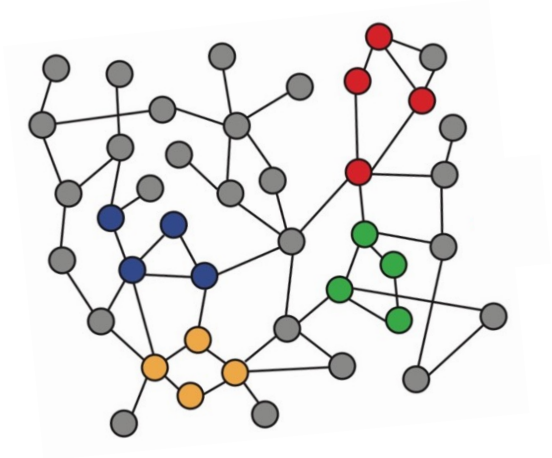


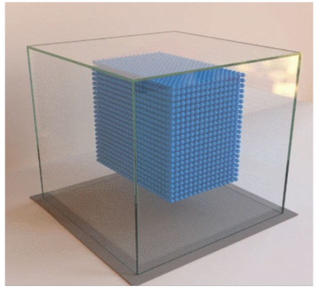
input graph $G = (V, E)$

GNN f_θ

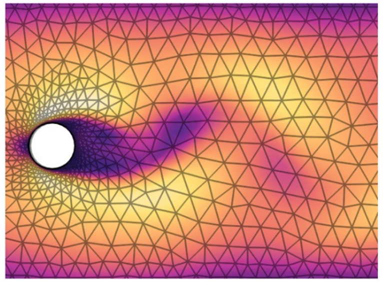
V: set of nodes with node features
E: set of edges with edge features

Predictions on the node/edge with updated features

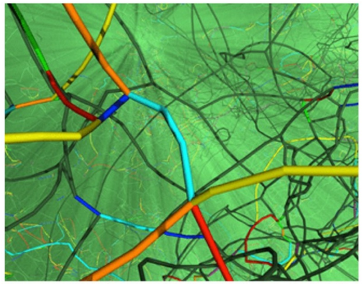




Fluid dynamics, computer graphics



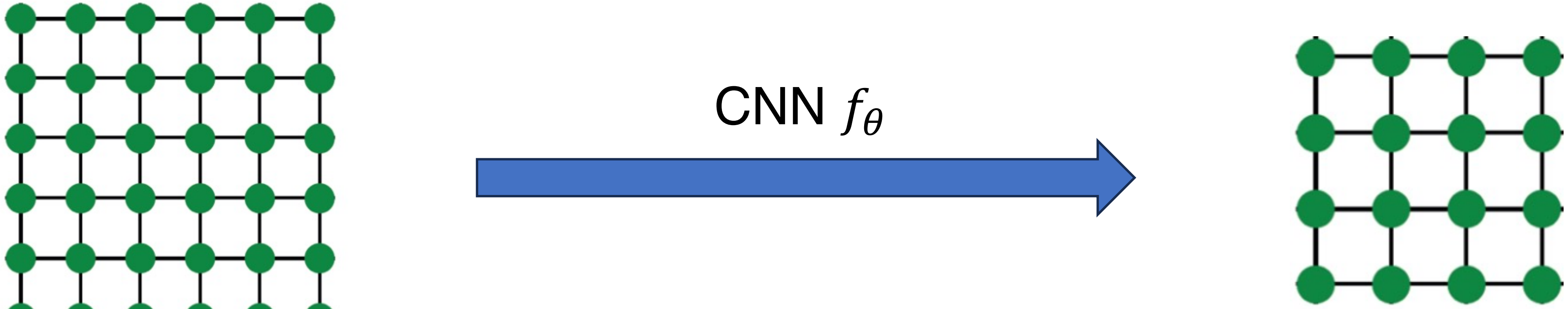
Mesh-based simulation for PDEs



Dislocation in materials

Proteins and small molecules

Neural architecture 3: Convolutional Neural Networks (CNN)

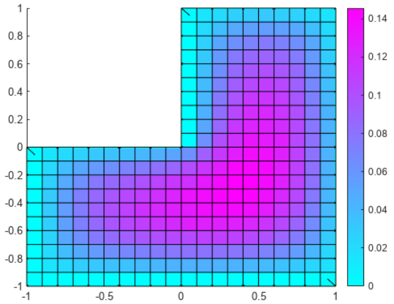


input tensor $X \in R^{d_1 \times d_2 \times \cdots \times d_n}$

output tensor $\hat{Y} \in R^{d_1' \times d_2' \times \cdots \times d_n'}$

image

video



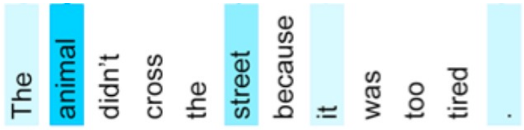
PDE discretized on a regular grid

Neural architecture 4: Transformer

Transformer f_θ

input sequence X

output sequence $\hat{Y}$



Natural language

DNA sequence

Protein sequence

Neural architecture: Summary

Data structure	Suitable neural architecture	Course #
Vector	Multilayer Perceptron (MLP)	2 (Tailin Wu & Tao Lin)
Graph	Graph Neural Network (GNN)	10 (Tailin Wu)
Grid	Convolutional Neural Network (CNN)	7 (Kaicheng Yu)
Sequence	Transformer	4 (Zhenzhong Lan)

For each neural architecture (same goes for topics in the course), we will introduce its:

- Motivation
- Architecture
- Typical tasks
- Research frontiers

Course introduction: Learning paradigm

Tasks

- Classification/ regression
- Simulation
- Inverse design/ inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

Learning paradigm

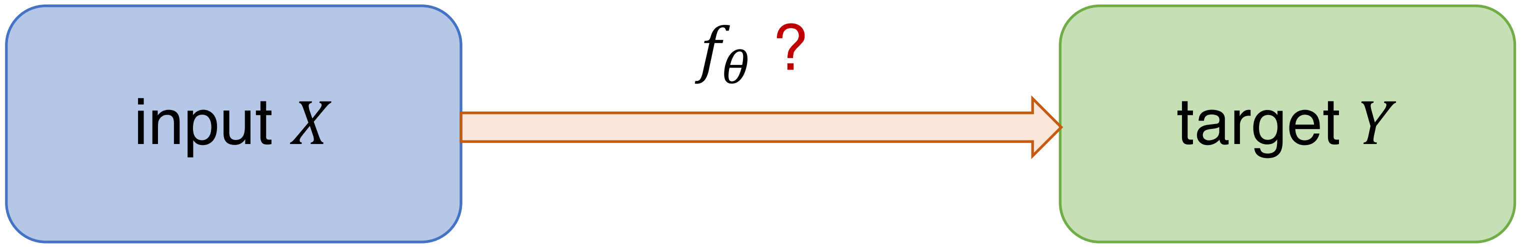
- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Learning paradigm 1: Supervised learning

- image
- video
- graph
- time series
- natural language
- …



classification

- label (discrete)
- scalar/tensor (continuous)

regression

Given many examples of $(X, Y) = \{(x_i, y_i)\}_{i=1}^{N}$ pairs, learn a neural network (NN) f_θ that minimizes the prediction loss:

$$\theta^* = \mathrm{argmin}_\theta \, \mathbb{E}_{(X,Y)\sim P(X,Y)}[\ell(f_\theta(X), Y)]$$

f_θ: neural network to be learned
ℓ: loss function

Learning paradigm 2: Generative modeling

Images and shapes generated by diffusion models:

By DallE 2

By MeshDiffusion [1]

[1] Liu, Zhen, et al. "Meshdiffusion: Score-based generative 3d mesh modeling." *ICLR 2023*

Learning paradigm 2: Generative modeling

Robotic policy by diffusion models [1]

Text to video generation by Sora [2]

[1] Fu, Zipeng, Tony Z. Zhao, and Chelsea Finn. "Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation." *arXiv preprint arXiv:2401.02117* (2024).
[2] OpenAI team. “Video generation models as world simulators”, 2024

Learning paradigm 2: Generative modeling

- image
- video
- graph
- time series
- natural language
- …

input X $\Rightarrow$ $p_\theta(X)$?

Probability model

Given many examples of the input X, learn a probability model $p_\theta(X)$ that can **sample** new instances of X that conform to the data distribution

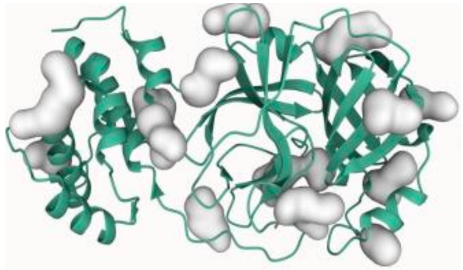
Major generative models:

- Diffusion models
- Flow
- Generative adversarial network (GAN)
- Variational autoencoder (VAE)

Learning paradigm 3: Foundation models

chatGPT

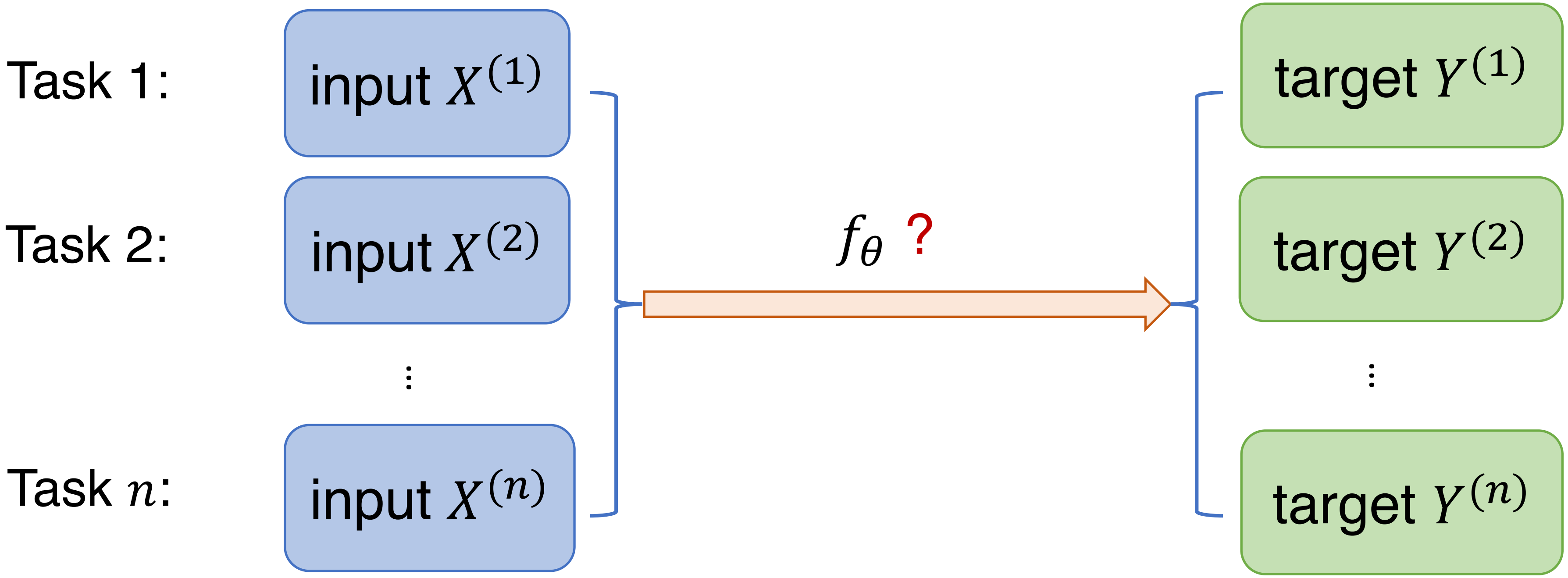
Sora [1]



uniMol [2]

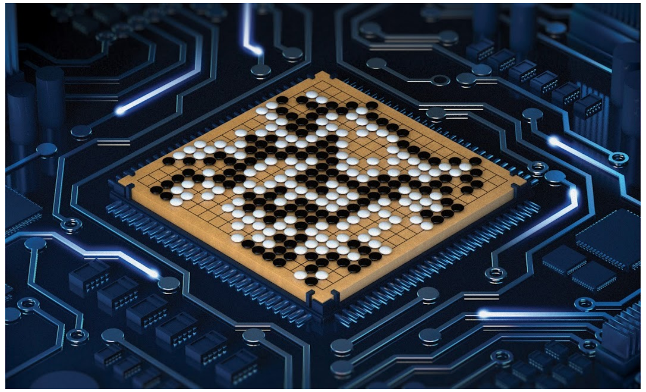
[1] OpenAI team. “Video generation models as world simulators”, 2024
[2] Zhou, Gengmo, et al. "Uni-Mol: a universal 3D molecular representation learning framework." ICLR 2023

Learning paradigm 3: Foundation models

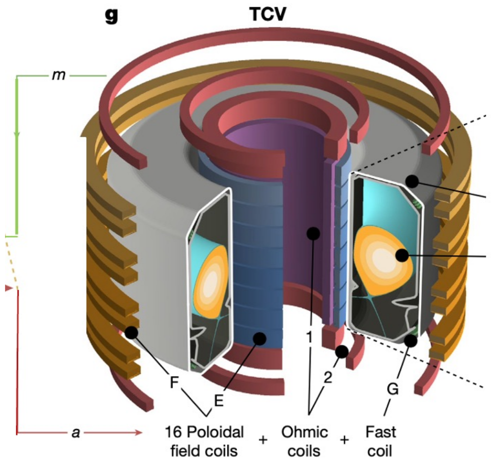


Given many diverse tasks, each consists of its massive number of examples $\left(X^{(n)}, Y^{(n)}\right) = \left\{\left(x_i^{(n)}, y_i^{(n)}\right)\right\}_{i=1}^{N^{(n)}}$, learn a single foundation model f_θ that can faithfully predict the target from the input.

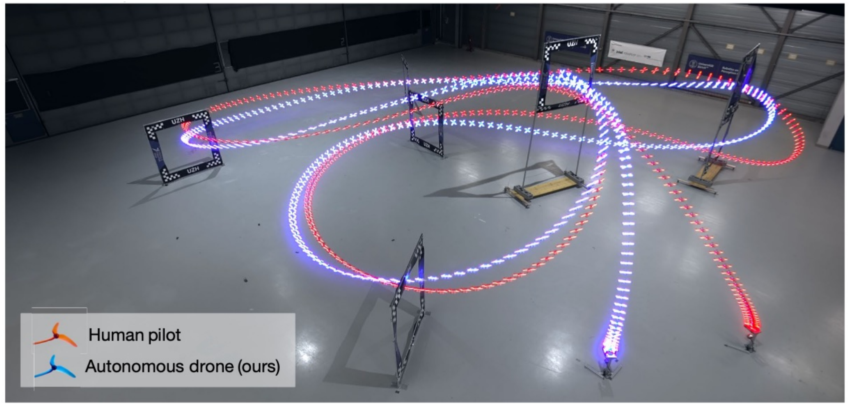
Learning paradigm 4: Reinforcement learning



AlphaGo [1]



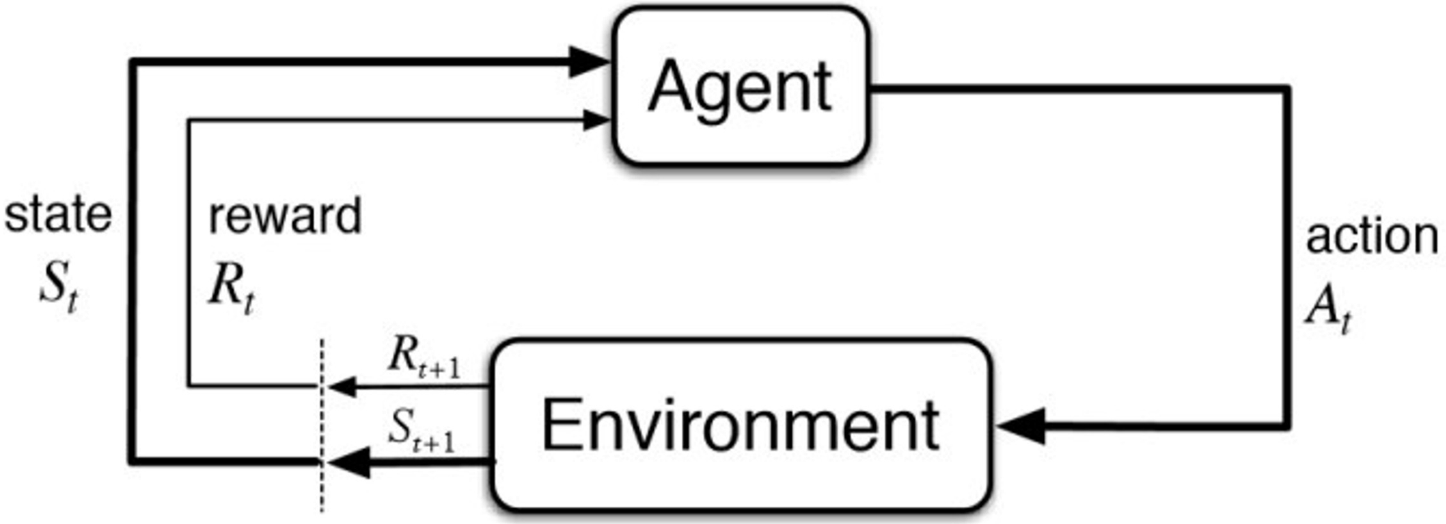
Controlled nuclear fusion [2]



Drone racing [3]

[1] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.
[2] Degrave, Jonas, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning." *Nature* 602.7897 (2022): 414-419.
[3] Kaufmann, Elia, et al. "Champion-level drone racing using deep reinforcement learning." *Nature* 620.7976 (2023): 982-987.

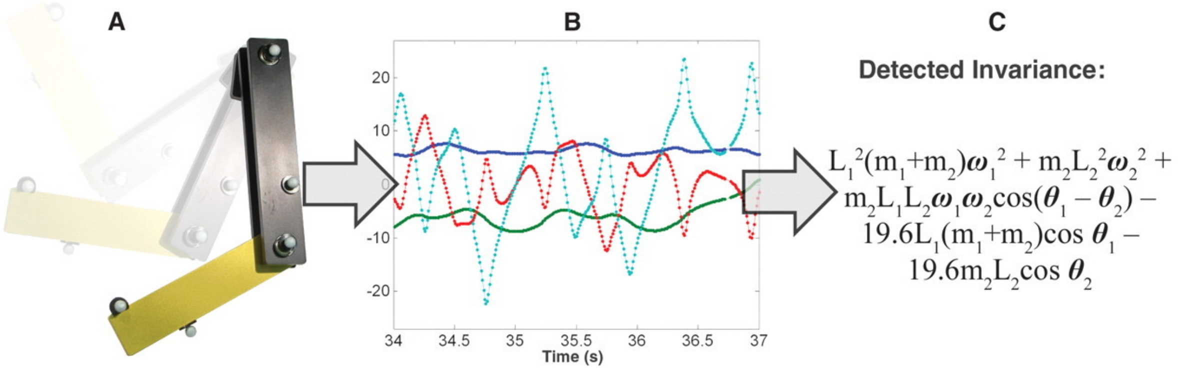
Learning paradigm 4: Reinforcement learning



Goal: maximize the long-term expected reward w.r.t. to the policy $\pi(A_t|S_t)$

$$\max_{\pi(A_t|S_t)} \mathbb{E}_t[R_t]$$

Learning paradigm 5: Evolutionary and multi-objective learning

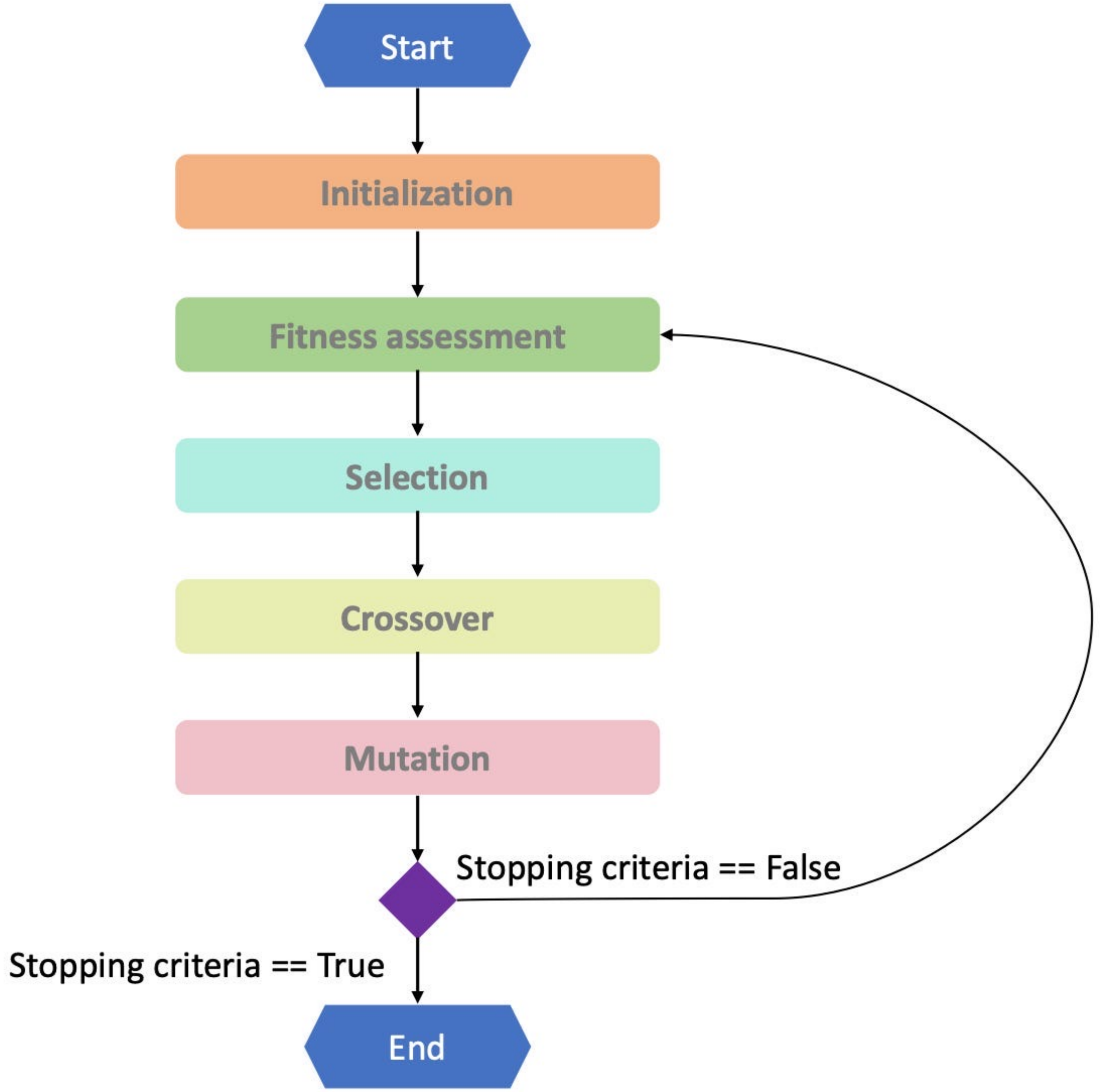


Discovering equations from data [1]

Optimization in engineering

[1] Schmidt, Michael, and Hod Lipson. "Distilling free-form natural laws from experimental data." *science* 324.5923 (2009): 81-85.

Learning paradigm 5: Evolutionary and multi-objective learning



Learning paradigm: Summary

Learning paradigm	Suitable scenarios	Course #
Supervised learning	Most standard	2 (Tao Lin, Tailin Wu)
Generative modeling	High-dimensional data, can also be used in any tasks in regression	3 (Tailin Wu)
Foundation models	Large diverse tasks	4 (Zhenzhong Lan)
Reinforcement learning	Agent interacting with environment, cannot pass gradient through	5 (Tailin Wu)
Evolutionary and multi-objective learning	Gradient-free, discrete optimization	12 (Yaochu Jin)

Course introduction: Application in AI and Science

Tasks

- Classification/ regression
- Simulation
- Inverse design/ inverse problem
- Control/planning

×

Neural architecture

- Multilayer perceptron
- Graph Neural Networks
- Convolutional Neural Networks
- Transformers

×

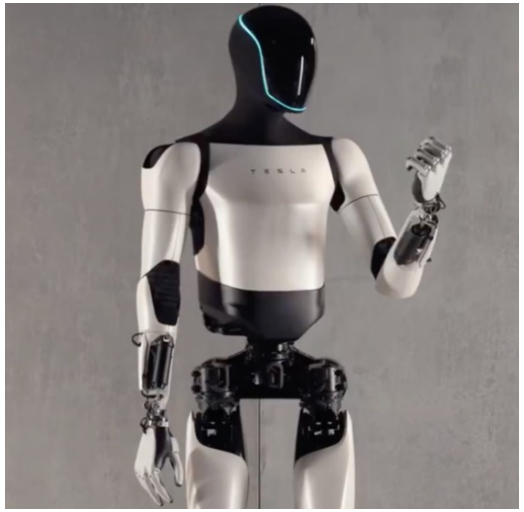
Learning paradigm

- Supervised learning
- Generative modeling
- Foundation models
- Reinforcement learning
- Evolutionary and multi-objective optimization

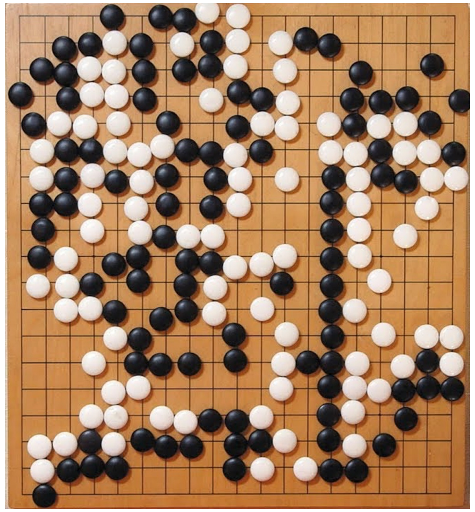
Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Application in AI



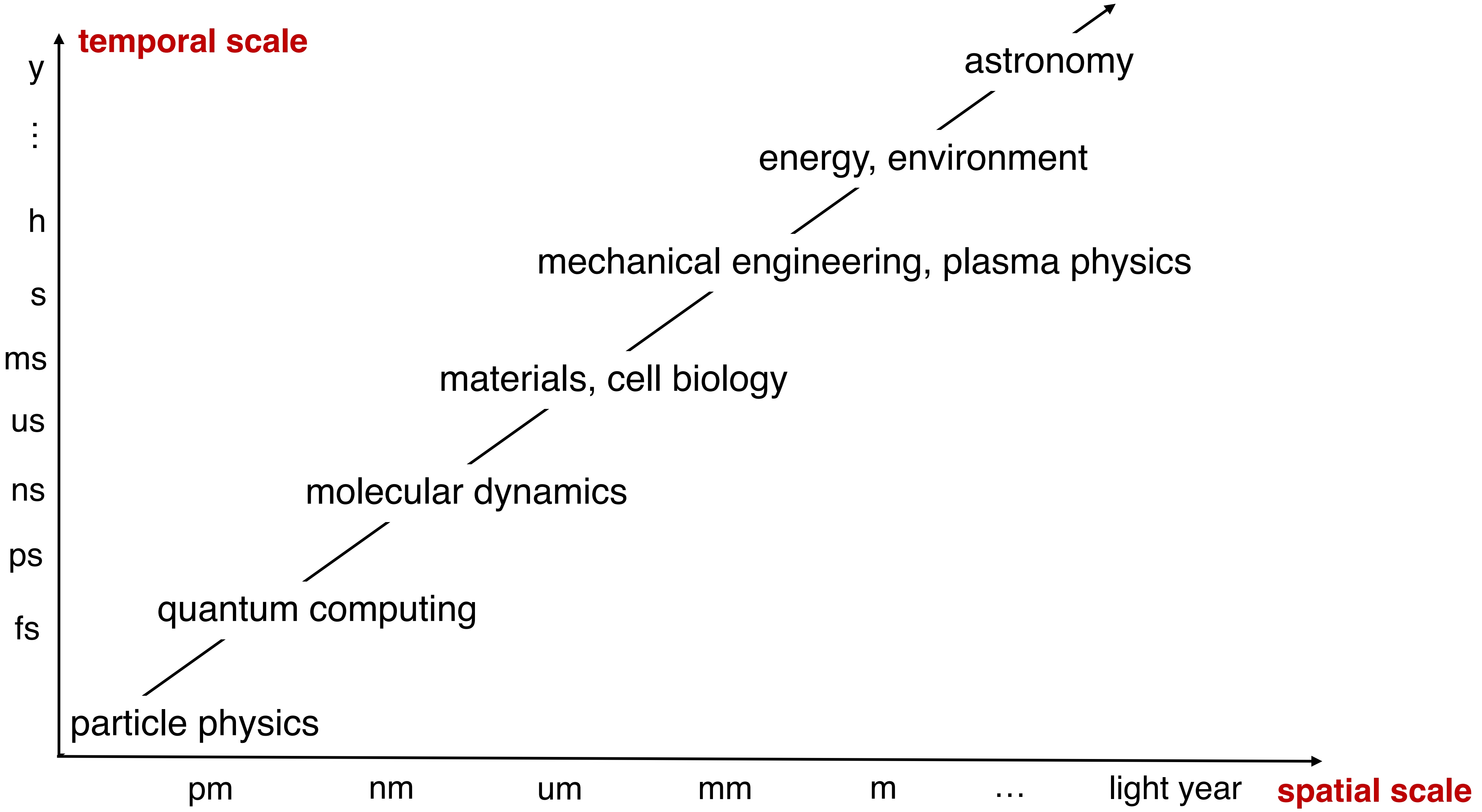
robotics



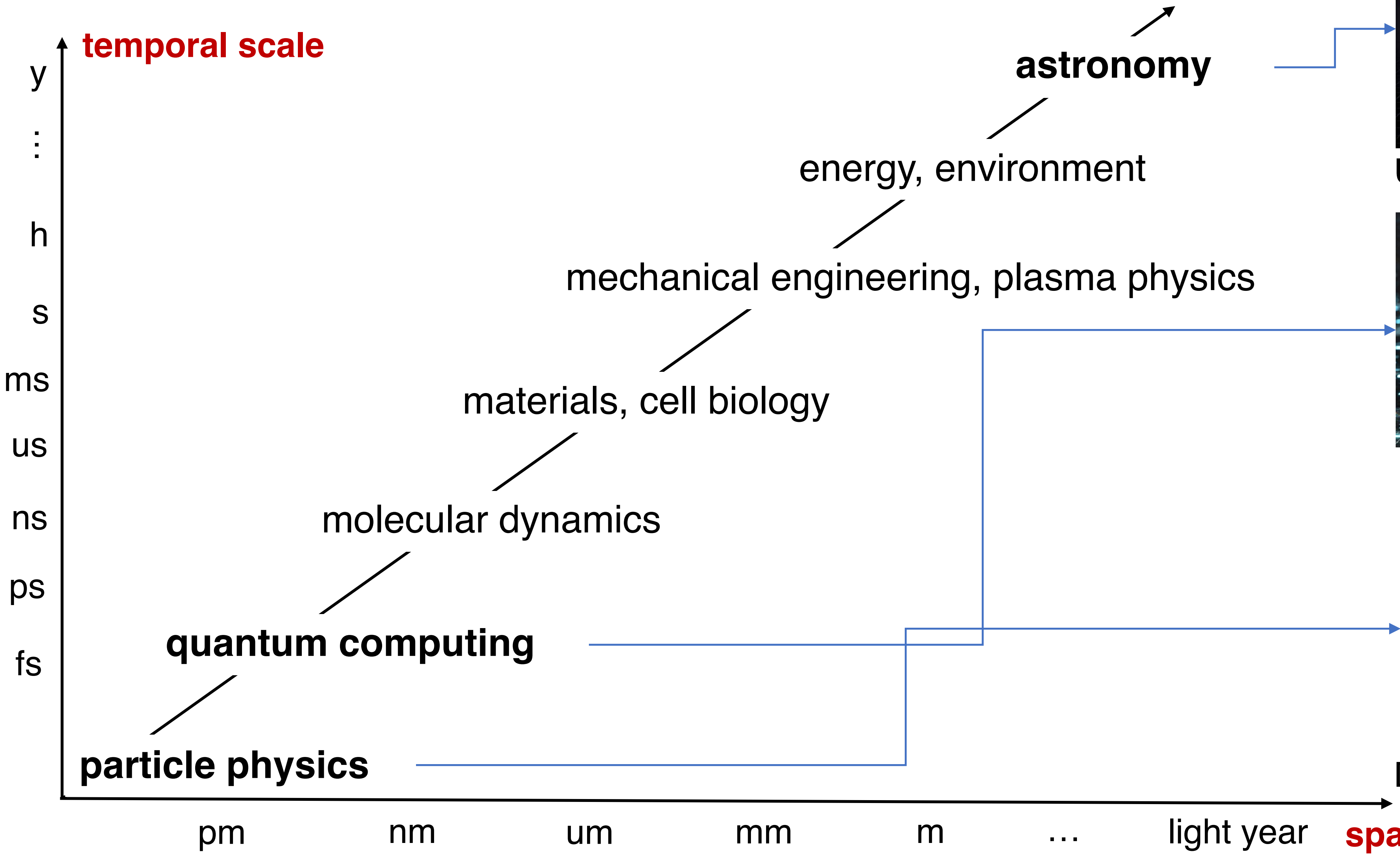
games

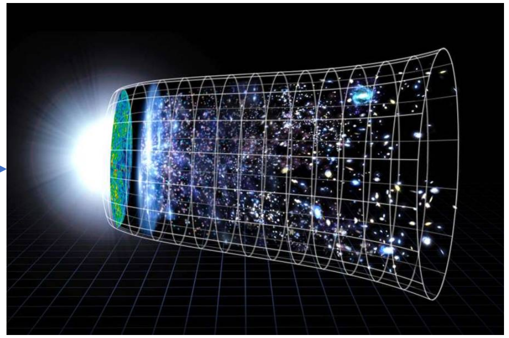
self-driving

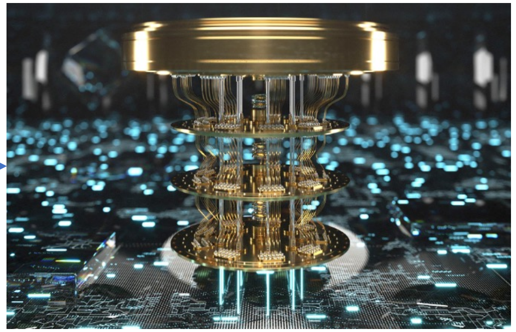
Application in AI for Science: from microscopic to macroscopic

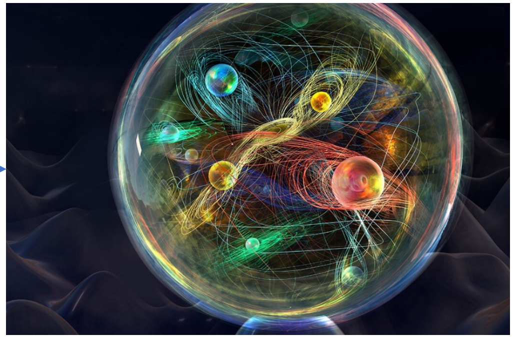


AI for Science: from microscopic to macroscopic

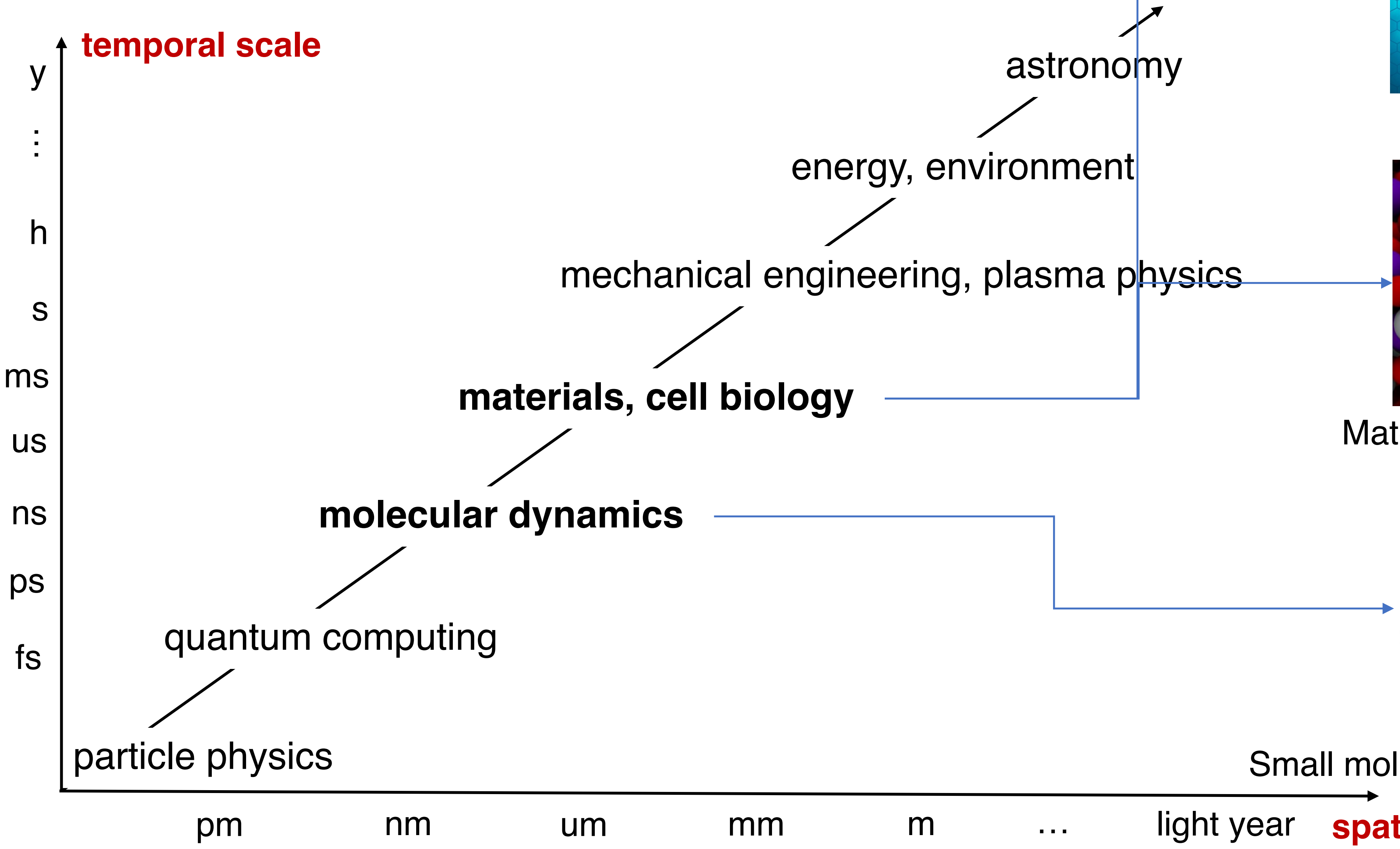



Understand our universe

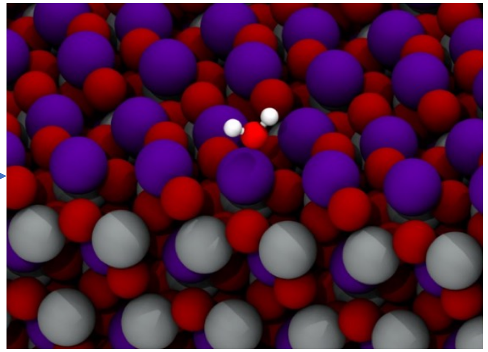

Speed up computation

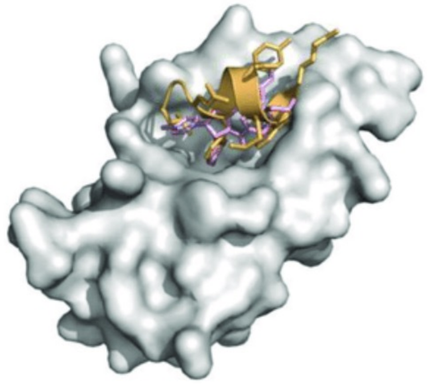

Discovering new physics

AI for Science: from microscopic to macroscopic

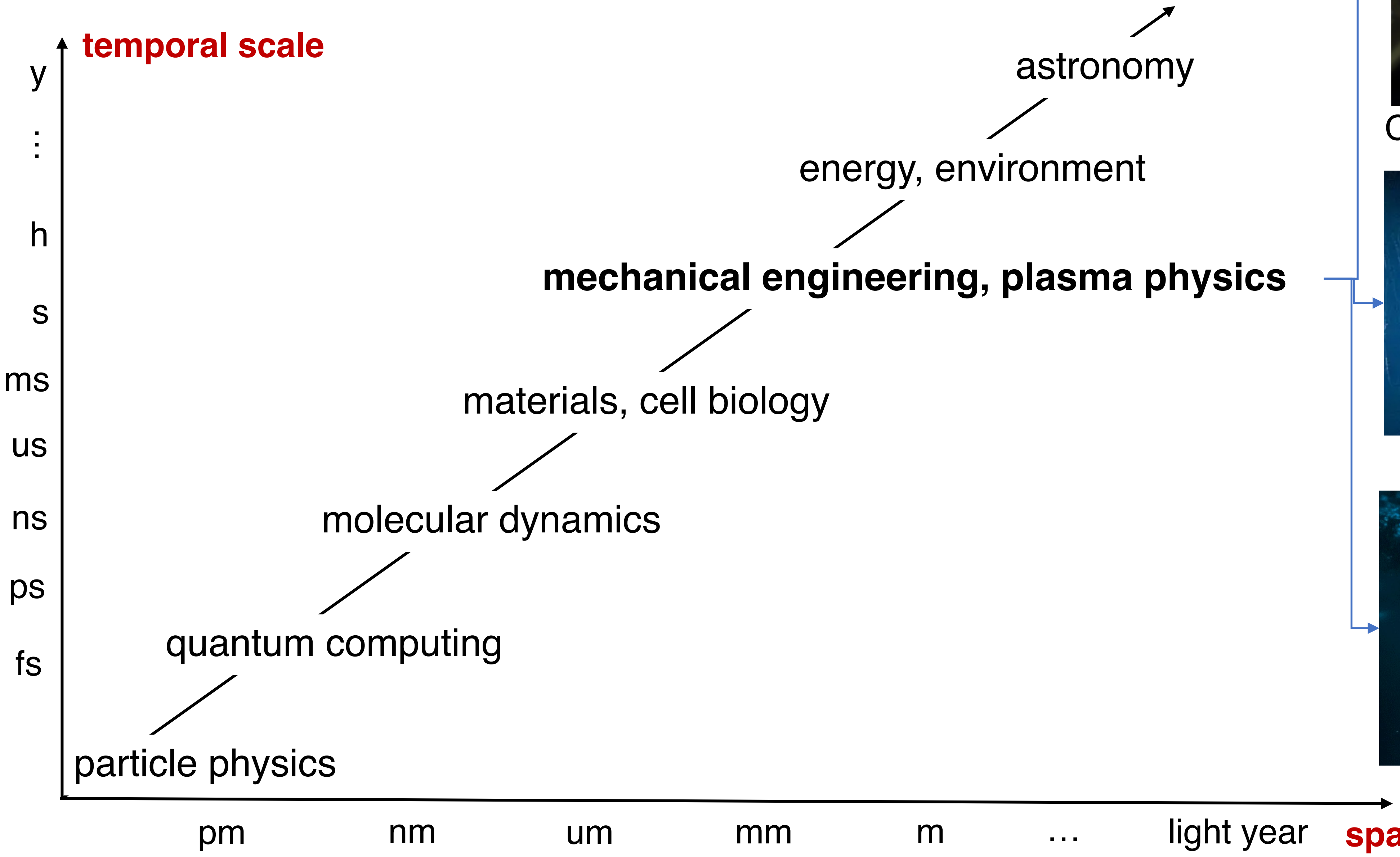


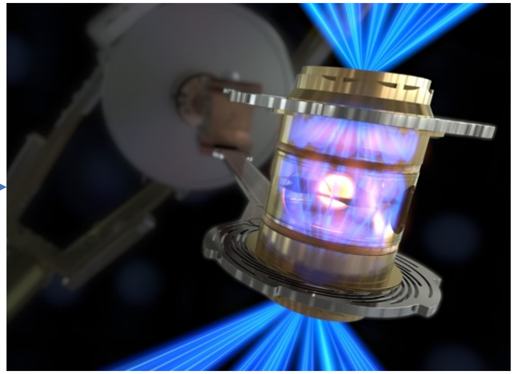

AI for genomics


Material simulation & design

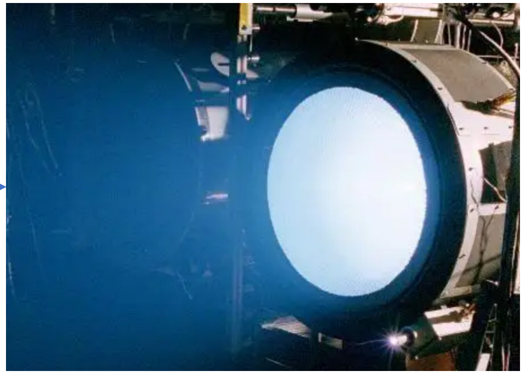

Small molecules & protein simulation

AI for Science: from microscopic to macroscopic

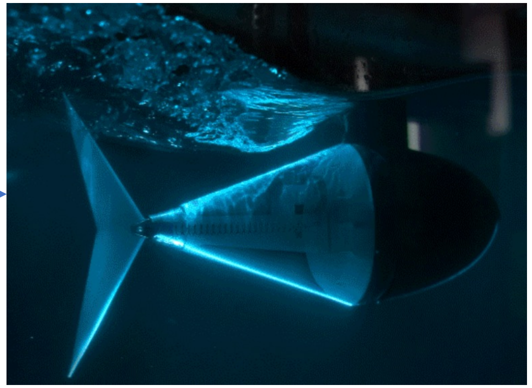




Controlled nuclear fusion



Ion thruster in space



Underwater robots

AI for Science: from microscopic to macroscopic

temporal scale

y
⋮
h
s
ms
us
ns
ps
fs

astronomy

energy, environment

mechanical engineering, plasma physics

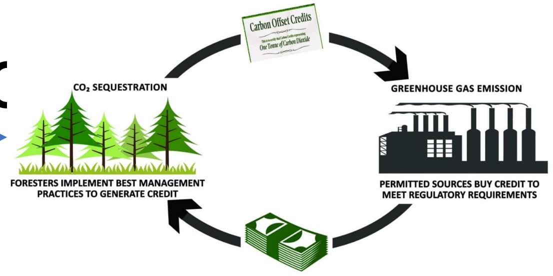
materials, cell biology

molecular dynamics

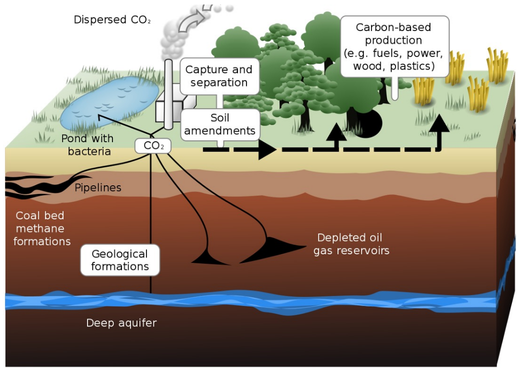
quantum computing

particle physics

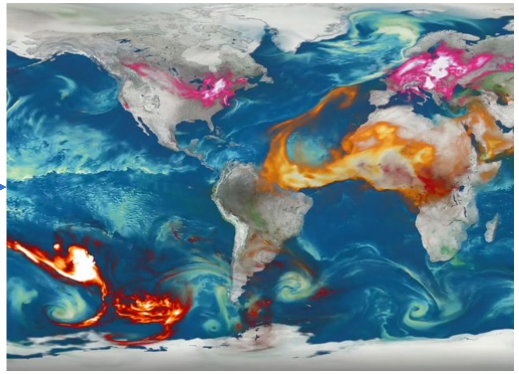
pm nm um mm m … light year **spatial scale**



Policy design for carbon credit

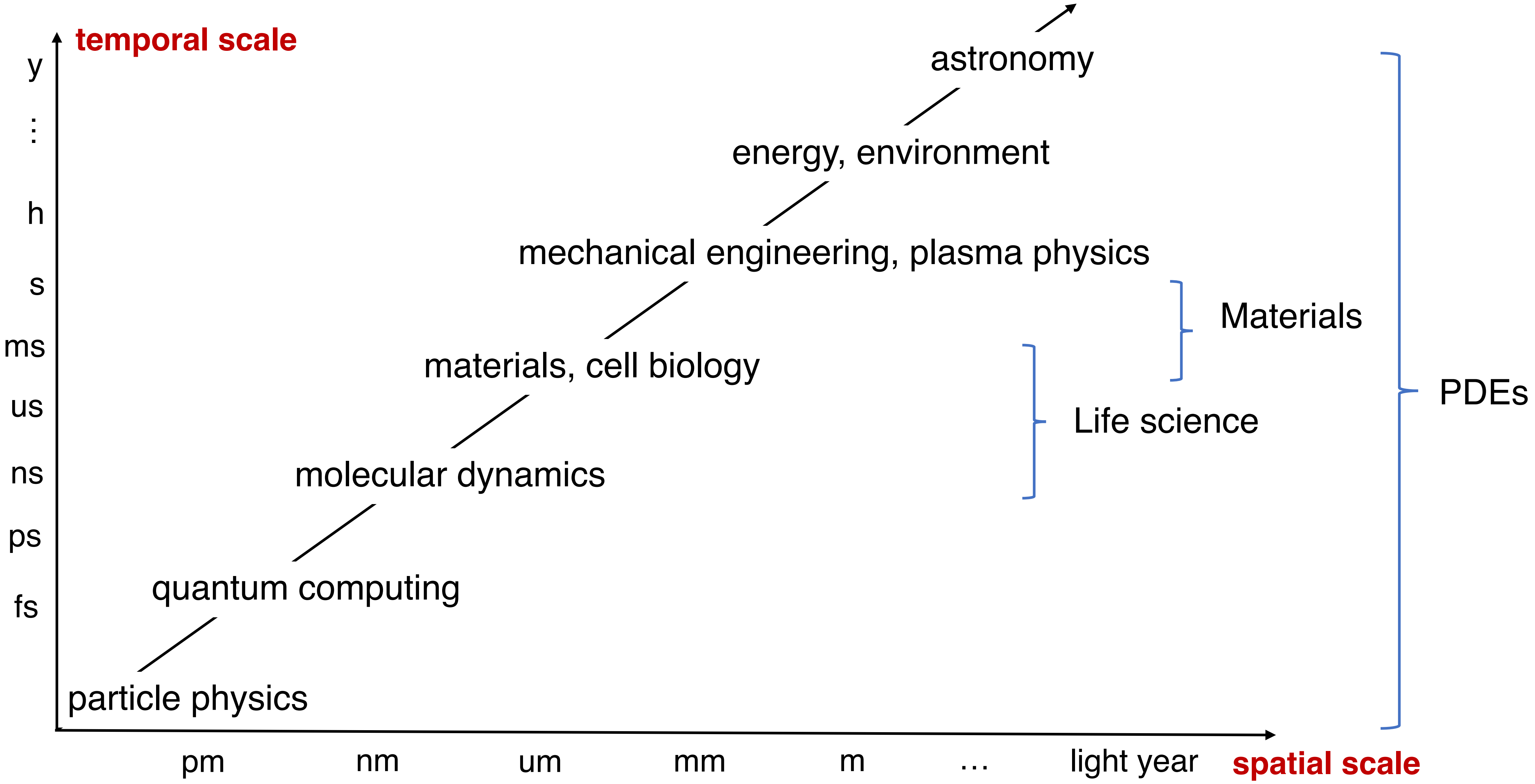


Carbon capture

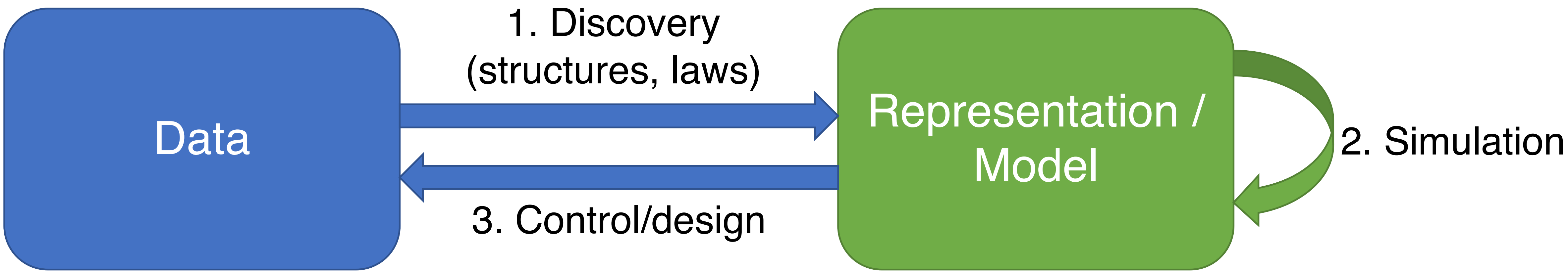


Weather forecasting

Application in AI for Science: from microscopic to macroscopic



AI for Science: universal tasks



These three tasks are fundamental in **science** and **engineering**

These three tasks are equally fundamental in **machine learning**

Application: Summary

Application	Area	Course #
Robotics	AI	6 (Donglin Wang)
Games	AI	6 (Donglin Wang)
Self-driving	AI	7 (Kaicheng Yu)
AI + PDEs/scientific computing	AI for science	13 (Tailin Wu)
AI + Life sciences	AI for science	11 (Ziqing Li)
AI + Materials science	AI for science	14 (Tailin Wu)

Course introduction: Summary

Tasks		Neural architecture		Learning paradigm
• Classification/ regression • Simulation • Inverse design/ inverse problem • Control/planning	×	• Multilayer perceptron • Graph Neural Networks • Convolutional Neural Networks • Transformers	×	• Supervised learning • Generative modeling • Foundation models • Reinforcement learning • Evolutionary and multi-objective optimization

Application (AI & Science)

- Robotics
- Games (e.g., Go, atari)
- Autonomous Driving
- PDEs
- Life science
- Materials science

Useful materials to get started in deep learning

- Learn PyTorch in 1h: https://pytorch.org/tutorials/beginner/introyt/introyt1_tutorial.html
- Book: Deep Learning book (https://www.deeplearningbook.org/)

Hope you all learn useful techniques to help your research!

Questions?