

# Machine Intelligence on Sequence Data Paradigm

Kan Ren

Guest Talk at AI Class

2023.12.29

# Outlines

- Introduction to sequence data
- Sequential decision making
- Modeling sequence data

# About me

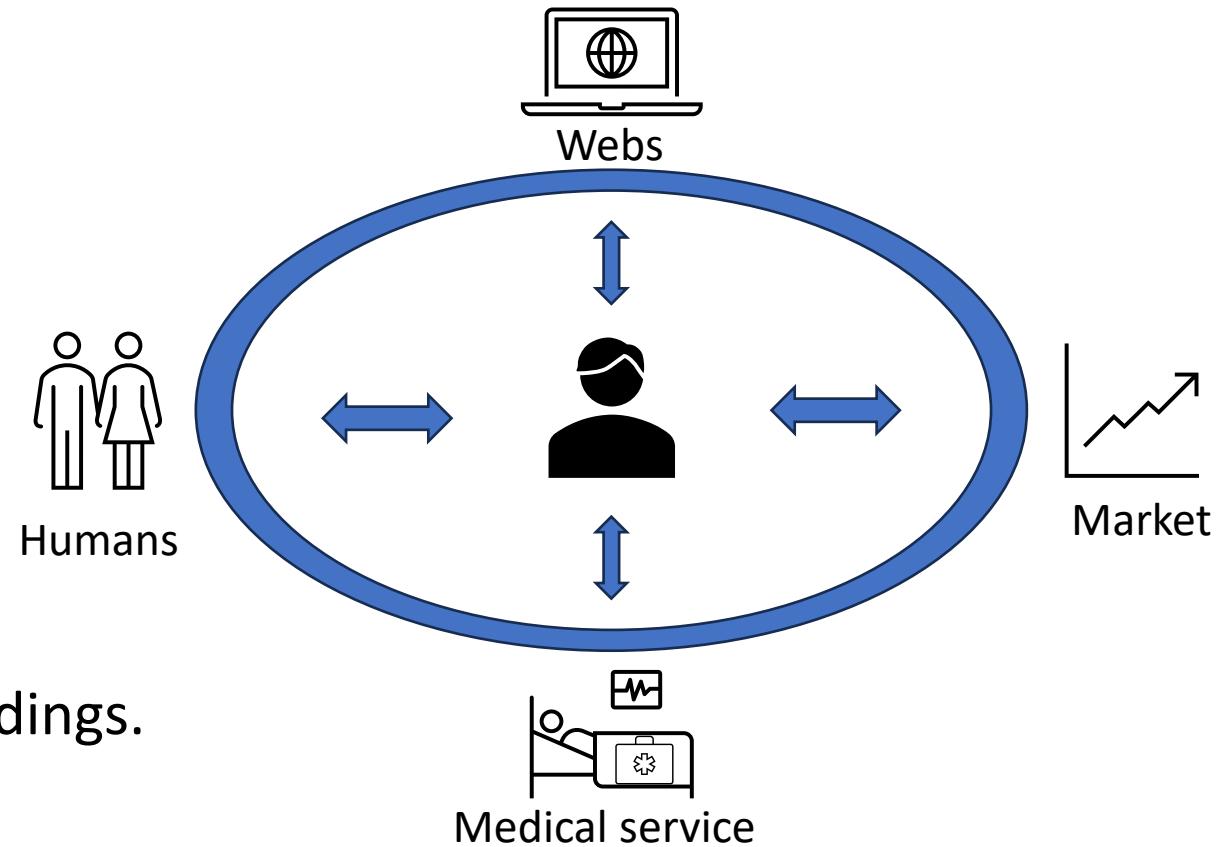
- Previously Senior Researcher in Microsoft Research Asia.
- Assistant Professor at ShanghaiTech University from Dec. 2023
- Machine learning for perception and decision making upon multi-modal big data.



<http://saying.ren/>

# Digitalization in human society

- Humans are *continuously*
  - Observing the surrounding envs.,
  - Processing the perceived data,
  - Memorizing information,
  - Make decisions influencing surroundings.



All the data are generated and collected along the time with temporal sequence, as defined as sequence data.

# Sequence data are everywhere

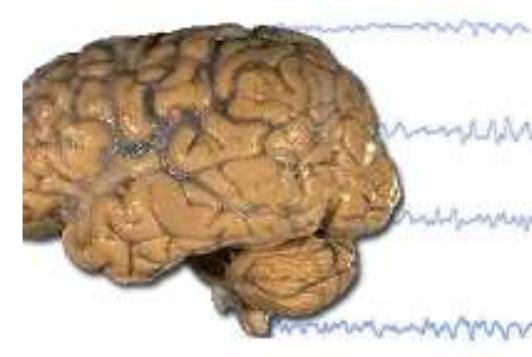
- Sequence data are the temporally generated and sensed data.



Finance



Urban



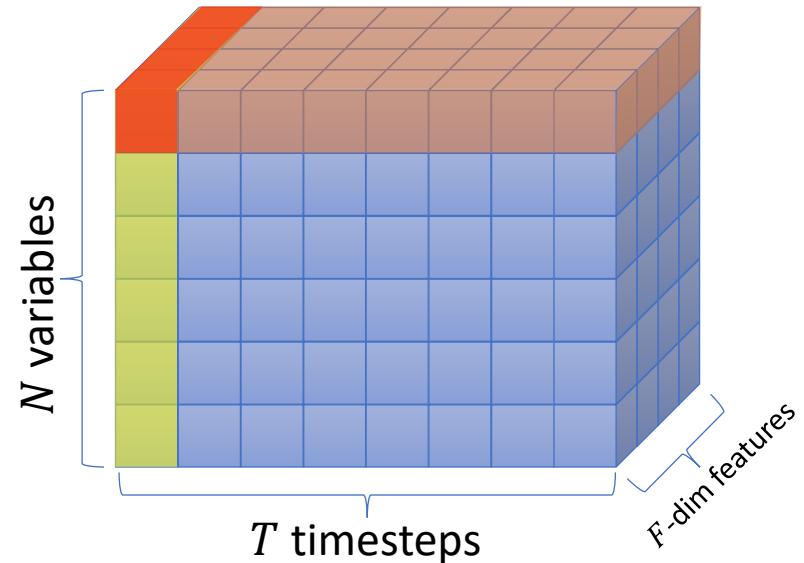
Healthcare



Marketing

# Formulation of sequence data

- Sequence data piece  $x \in \mathbb{R}^{N \times T \times F}$  is a three-dimensional tensor as



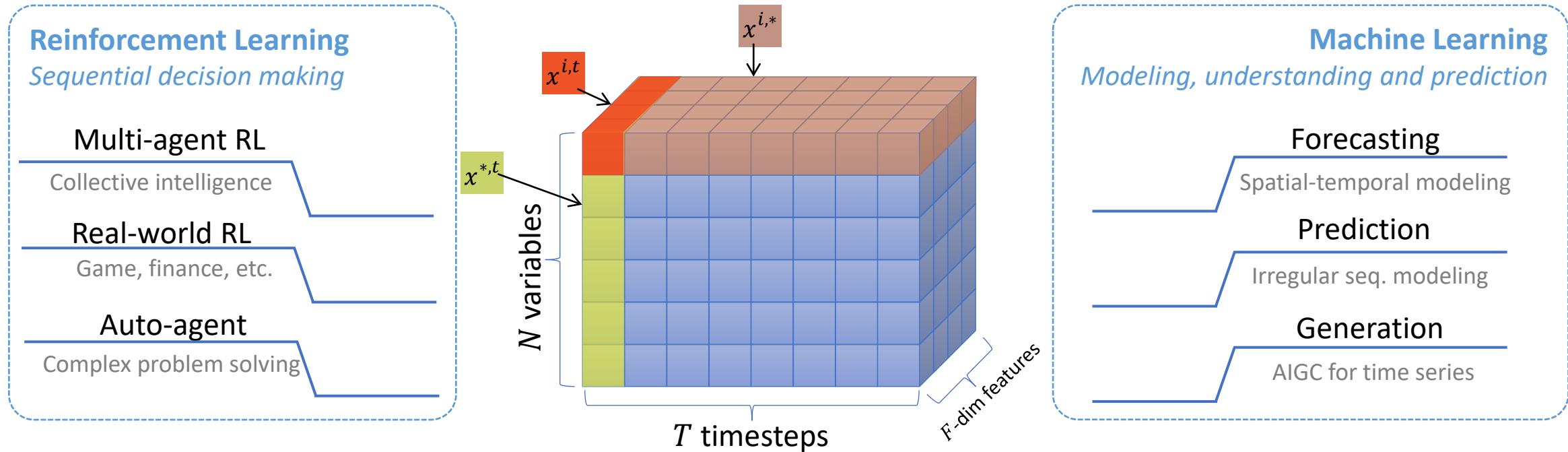
Domain	Variate	Feature	Timestep
Weather	PM2.5, temperature	(long, latitude, value)	Timestamp
Finance	Stock or instruments	(open, close, amount)	Timestamp
User behavior	Browsing, click, etc.	(Location, user id)	Temporal index
Natural language text	Character tokens	(token id)	Temporal index

# Machine intelligence on sequence data paradigm (SDP)

Learning to temporally *model, understand and interacting* with the sequence data resources.

# Machine intelligence on sequence data paradigm

- Process, understand, respond to temporally arriving sequence data with machine learning technology.

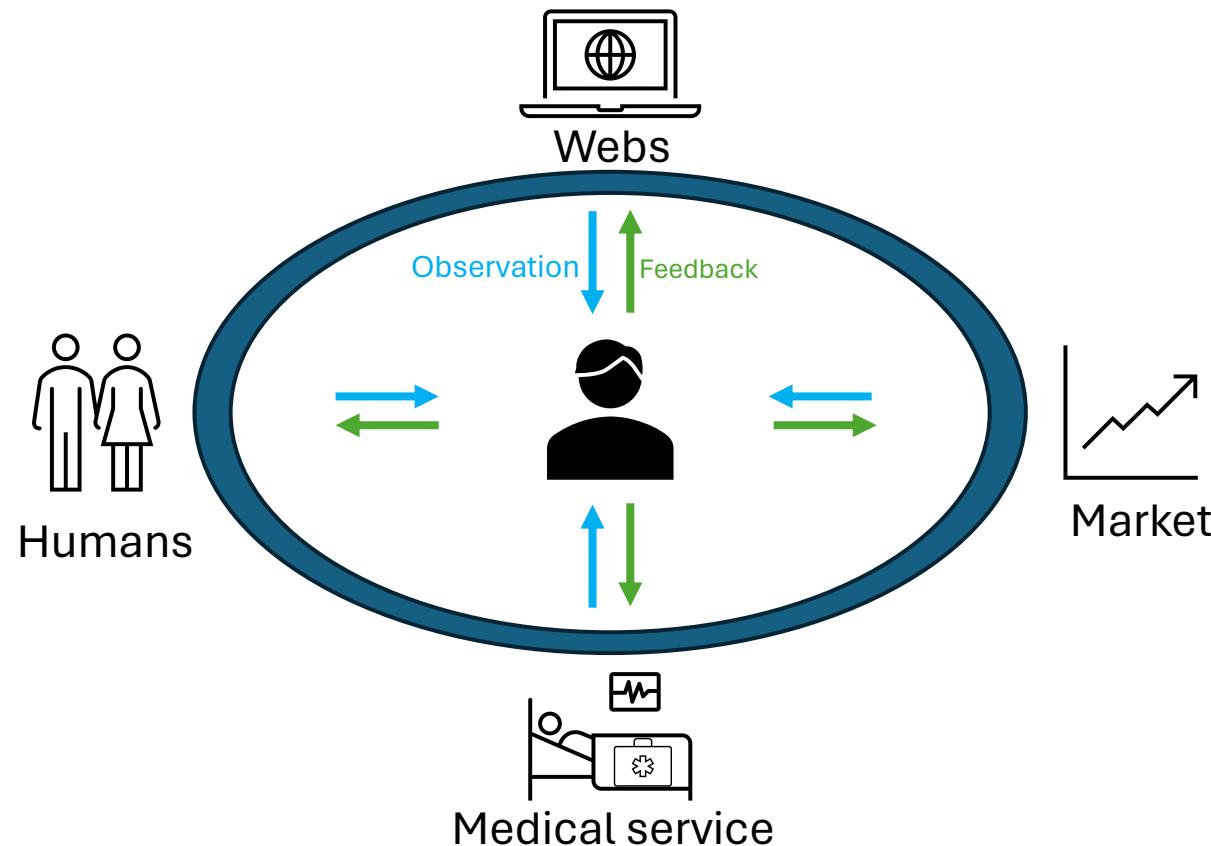


# Sequential decision making

- Decision making and reinforcement learning
- Game AI
- Real-world RL
- Autonomous agents

# Sequential decision making

- Humans are making decisions and providing feedbacks to the environment, which affects the environments and the corresponding future states of the env.



# Practical examples of sequential decision making



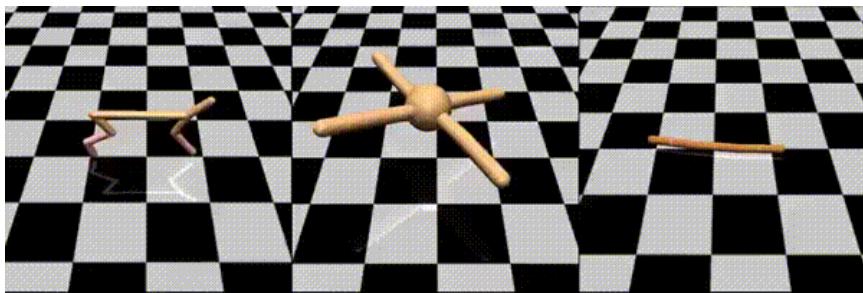
Logistics system



Video Games



Intelligent financial investment



MuJoCo Robot Control



Industrial production



Recommendation and ads

# Basis of reinforcement learning\*

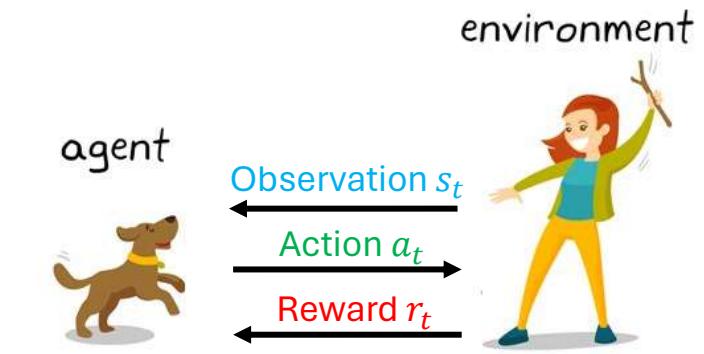
- Markov decision process (MDP) assumption
- At timestep  $t$ , agent following a policy  $\pi_\omega(s_t)$ ,
  - Obtains an **observation  $s_t$**  of the surrounding environment,
  - produces **action  $a_t$** ,
  - then, environment will transmit to  $s_{t+1}$ ,
  - and agent will receive **reward  $r_t$** ,
- The goal of the learning agent is to *maximize* the expected cumulative rewards as

$$R = \mathbb{E}_\pi \left[ \sum_{t=0}^{T-1} r_t(s_t, a_t) \right]$$

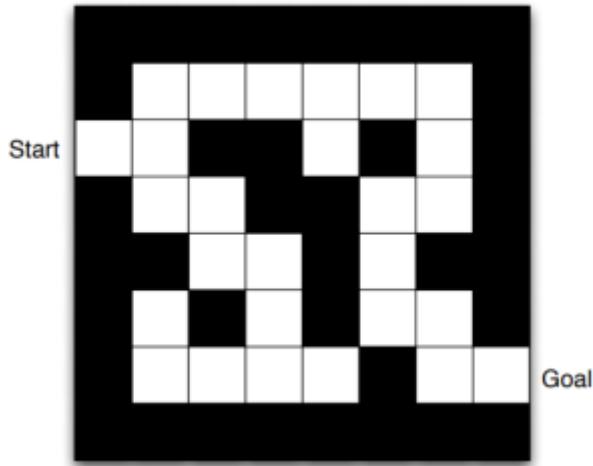
Interaction data stored in the transition dataset.

$\{s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_t, a_t, r_t, s_{t+1}, \dots, s_{T-1}, a_{T-1}, r_{T-1}, \}$

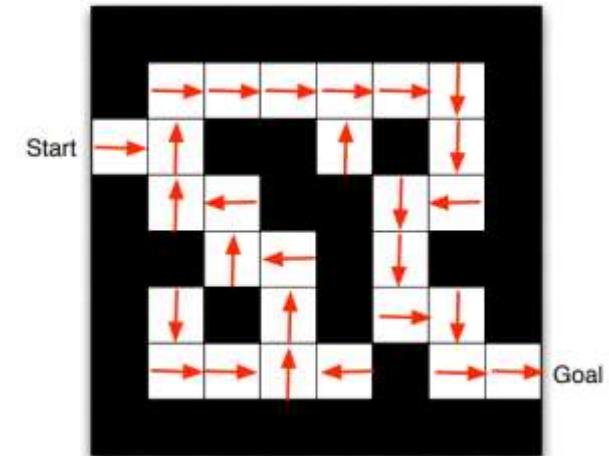
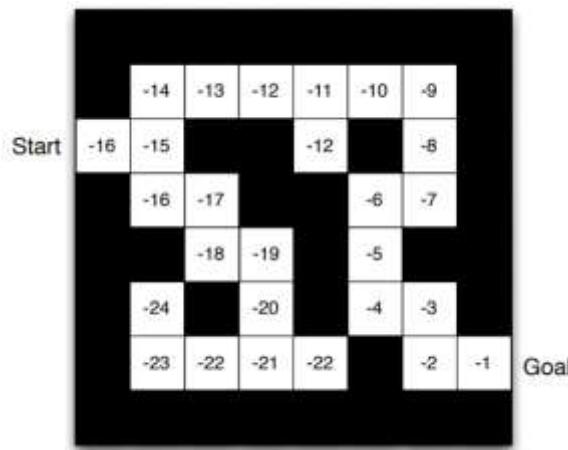
Transition



# Example of RL policy



- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

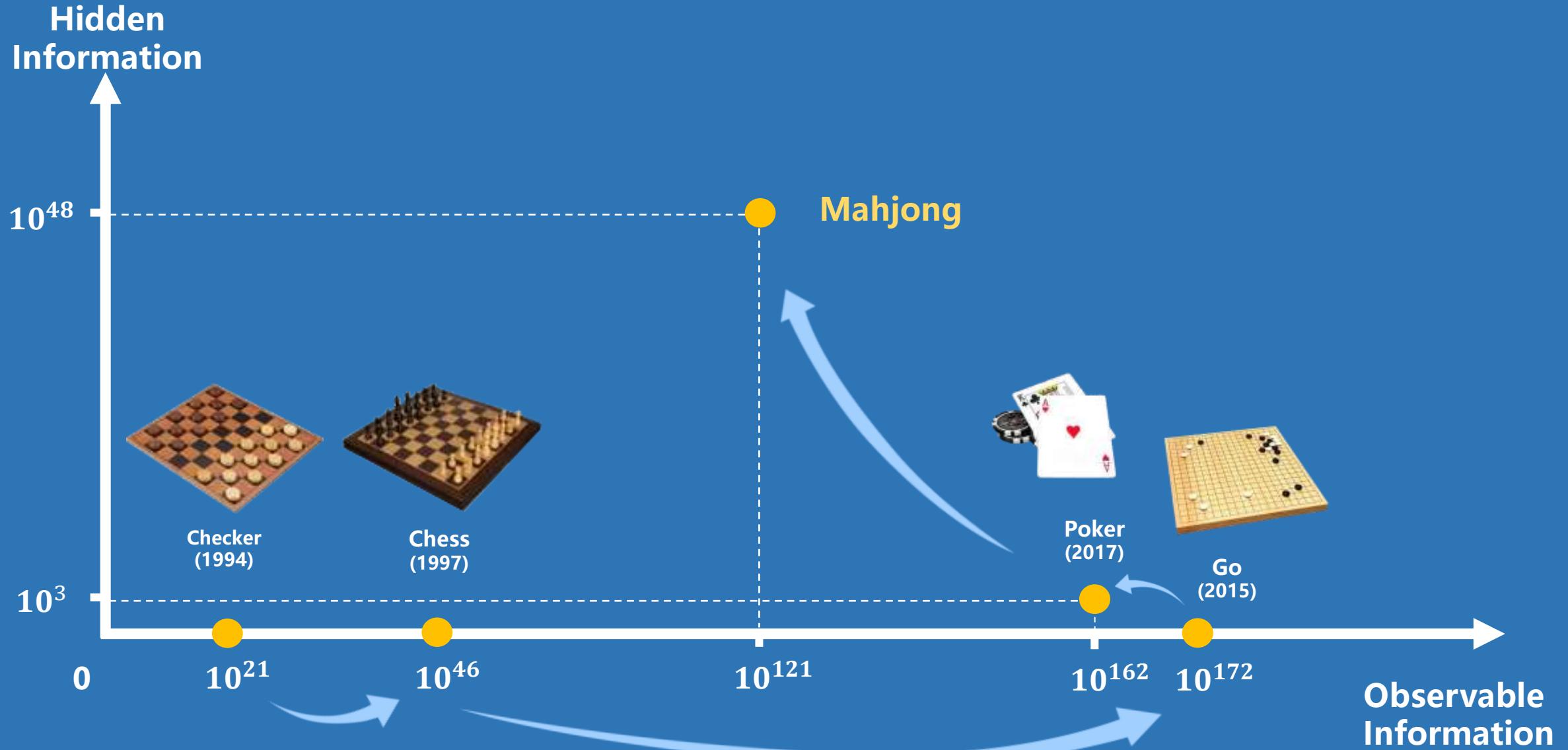


# Game AI in Mahjong

- Multi-player Mahjong game playing with unobservable information from other players.



# Breakthroughs of Game AI



# Challenges in Mahjong

- Complicated scoring rule
  - Ranking based long-term reward
  - Huge number of possible winning hands
- Large set of hidden information
  - More than  $10^{48}$  hidden states
- Playing rule is complex
  - Hard to build a tree for search

# Technologies to address the challenges

- Global reward predictor
  - Provide effective learning signal
  - Look ahead features to encode rich possible winning hands
- Parametric Monte-Carlo policy adaption
  - Modifies and adapts the offline-train policy to a specific round in the online playing stage
- Oracle guiding
  - An oracle agent that can see the perfect information
    - Including the private tiles of other players and wall tiles

# Online evaluation: highest ranking in the expert room

- Suphx is the first and only AI to achieve 10 DAN.
- Suphx is the highest ranking in the Expert room, the only room that AI is allowed to play in.

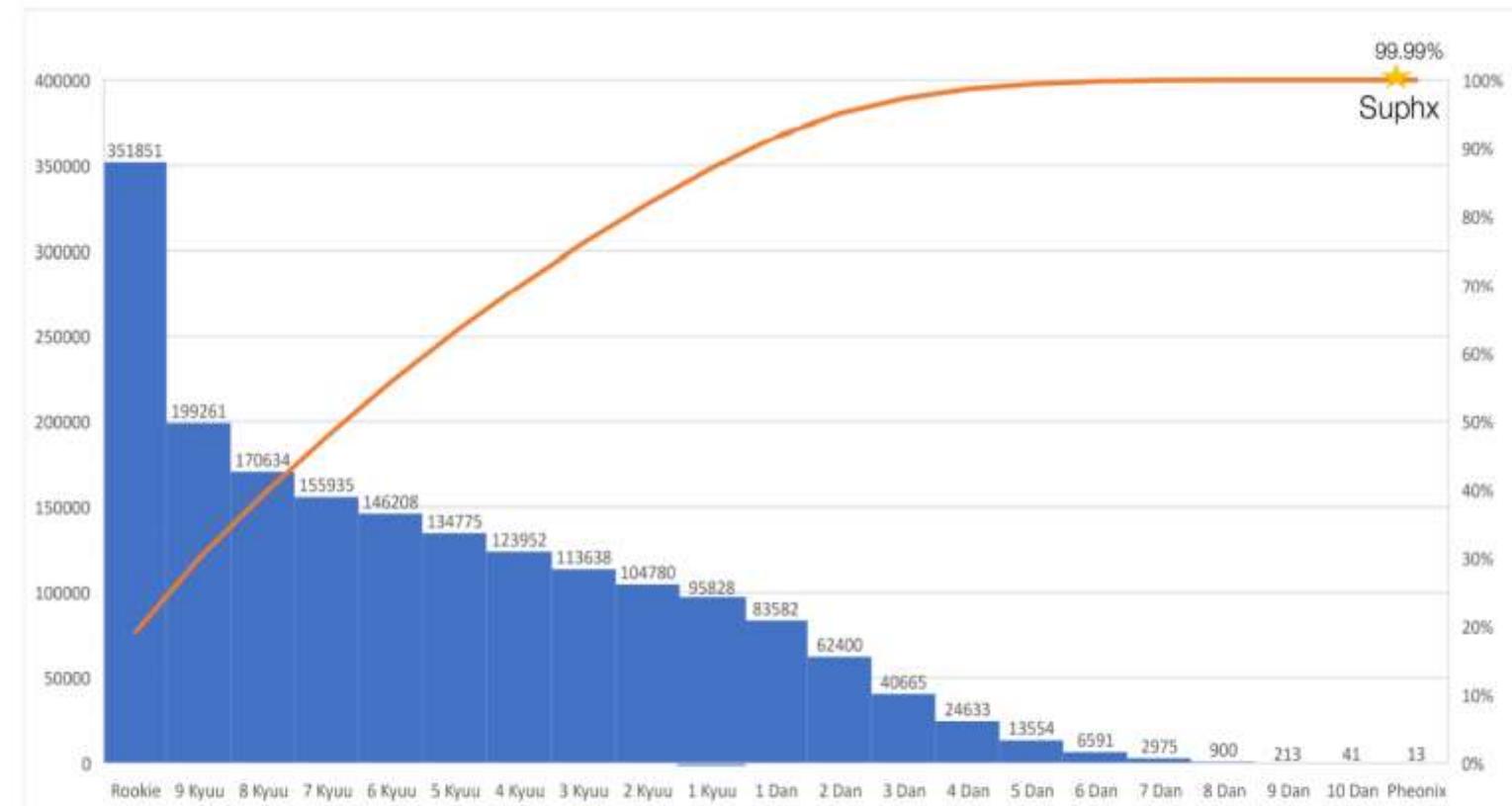
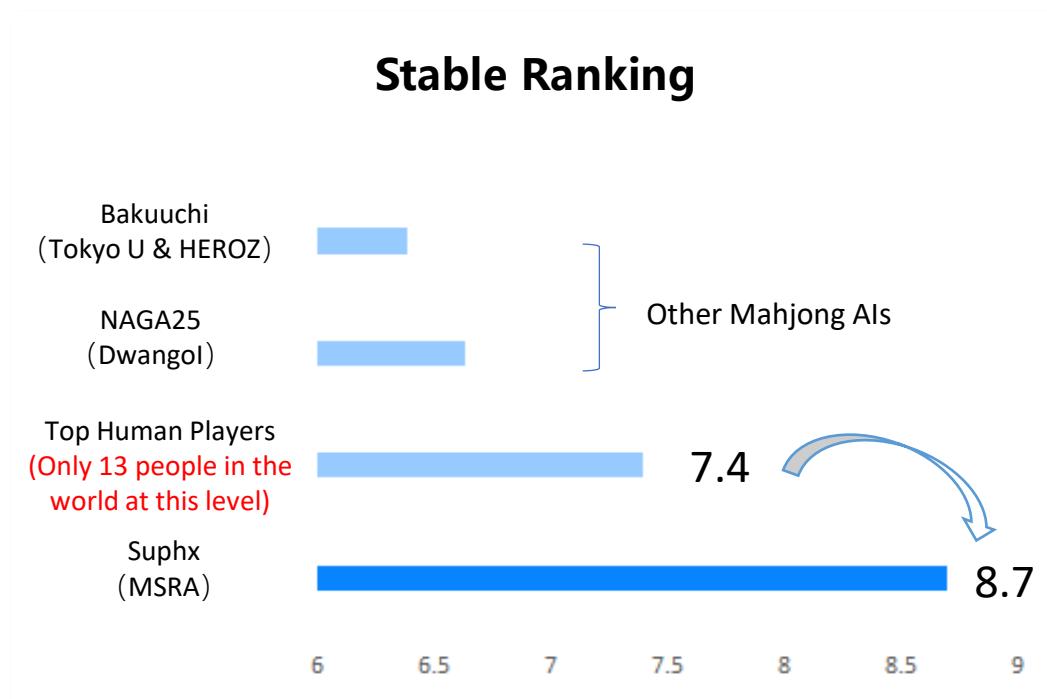


Figure 11: Distributions of record ranks of human players in Tenhou. Each bar indicates the number of human players above a certain level in Tenhou.

# Evaluation on Tenhou



- Played 5,000+ games in Expert room of Tenhou since March 2019 and achieved 10 DAN in June 2019.



I've watched Suphx's playing for quite a while. I feel like "it is even stronger than me."



**ASAPIN**

*The first player to achieve the highest DAN on 4-player Mahjong*



**Futokunaio**  
*The only player who achieved the highest DAN on both 3-player and 4-player Mahjong*

I've watched 300+ games of Suphx and it was really doing great. I am studying its style and I stopped watching human players' games.

# Game review by top human players

Some reviews are by highest ranking professional players in Japan.  
“best textbook”

Video Title	Duration	Views	Upload Date	Channel
最强日麻人工智能Suphx牌谱研究 03	49:04	3553	2019-06-30	夏之冰精
最强日麻人工智能Suphx牌谱研究 07	01:14:25	2447	2019-10-25	夏之冰精
最强日麻人工智能Suphx牌谱研究 05	49:10	3295	2019-10-18	夏之冰精
最强日麻人工智能Suphx牌谱研究 12	28:31	1766	2020-03-10	夏之冰精
最强日麻人工智能Suphx牌谱研究 06	26:27	2692	2019-10-21	夏之冰精
最强日麻人工智能Suphx牌谱研究 09	12:31	1456	2020-01-07	夏之冰精

天鳳十段を達成した麻雀AI「nSuphx (Super Phoenix)」の開発元 ...

<https://mj-news.net> ゲーム・アプリ, 天鳳 - Translate this page

 Aug 29, 2019 - Uploaded by 麻雀ウォッチ  
2019年6月22日天鳳十段に到達した麻雀AIの「nSuphx」(Twitter : @MSuphx)。これまでその開発者は明か ...

【麻雀】最強麻雀AIのSuphx(スーパーフェニックス)研究【1位牌譜】 ...

<https://www.youtube.com/watch> - Translate this page

 May 30, 2019 - Uploaded by うに丸ちゃんねる  
トッププレイヤークラスの成績を残している麻雀AI Suphx(スーパーフェニックス)の牌譜検討をします！普段は天鳳の鳳凰卓で東風、東 ...

【麻雀】最強麻雀AIのSuphx(スーパーフェニックス)研究【4位牌譜】 ...

<https://www.youtube.com/watch> - Translate this page

 May 25, 2019 - Uploaded by うに丸ちゃんねる  
Suphx(スーパーフェニックス)の牌譜検討をします！普段は天鳳の鳳凰卓で東風、東南、サンマの実況プレイ動画を中心に動 ...  
You've visited this page 3 times. Last visit: 8/3/19

【麻雀】最強の麻雀AIのSuphx(スーパーフェニックス)の牌譜研究を ...

<https://www.youtube.com/watch> - Translate this page

 May 20, 2019 - Uploaded by うに丸ちゃんねる  
Suphx(スーパーフェニックス)の牌譜検討をします！普段は天鳳の鳳凰卓で東風、東南、サンマの実況プレイ動画を中心に動 ...  
You've visited this page 3 times. Last visit: 8/3/19

Suphx観戦研究 - YouTube

<https://www.youtube.com/watch> - Translate this page

 May 27, 2019 - Uploaded by 遊鳥ちゅん  
解説したり雑談したりしながら天鳳打っていきます。Twitterもやってます → https://twitter.com/yutori\_style.

# Real-world reinforcement learning

- Sequential decision making in finance
  - Universal Trading for Order Execution with Oracle Policy Distillation (AAAI 2021)
  - Learning Multi-Agent Intention-Aware Communication for Optimal Multi-Order Execution in Finance (KDD 2023)

# Universal Trading for Order Execution with Oracle Policy Distillation

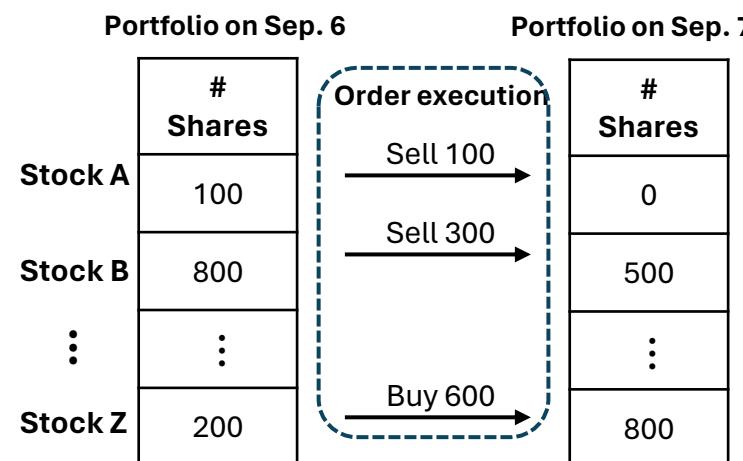
AAAI 2021

Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou,  
Weinan Zhang, Jiang Bian, Yong Yu, Tie-Yan Liu

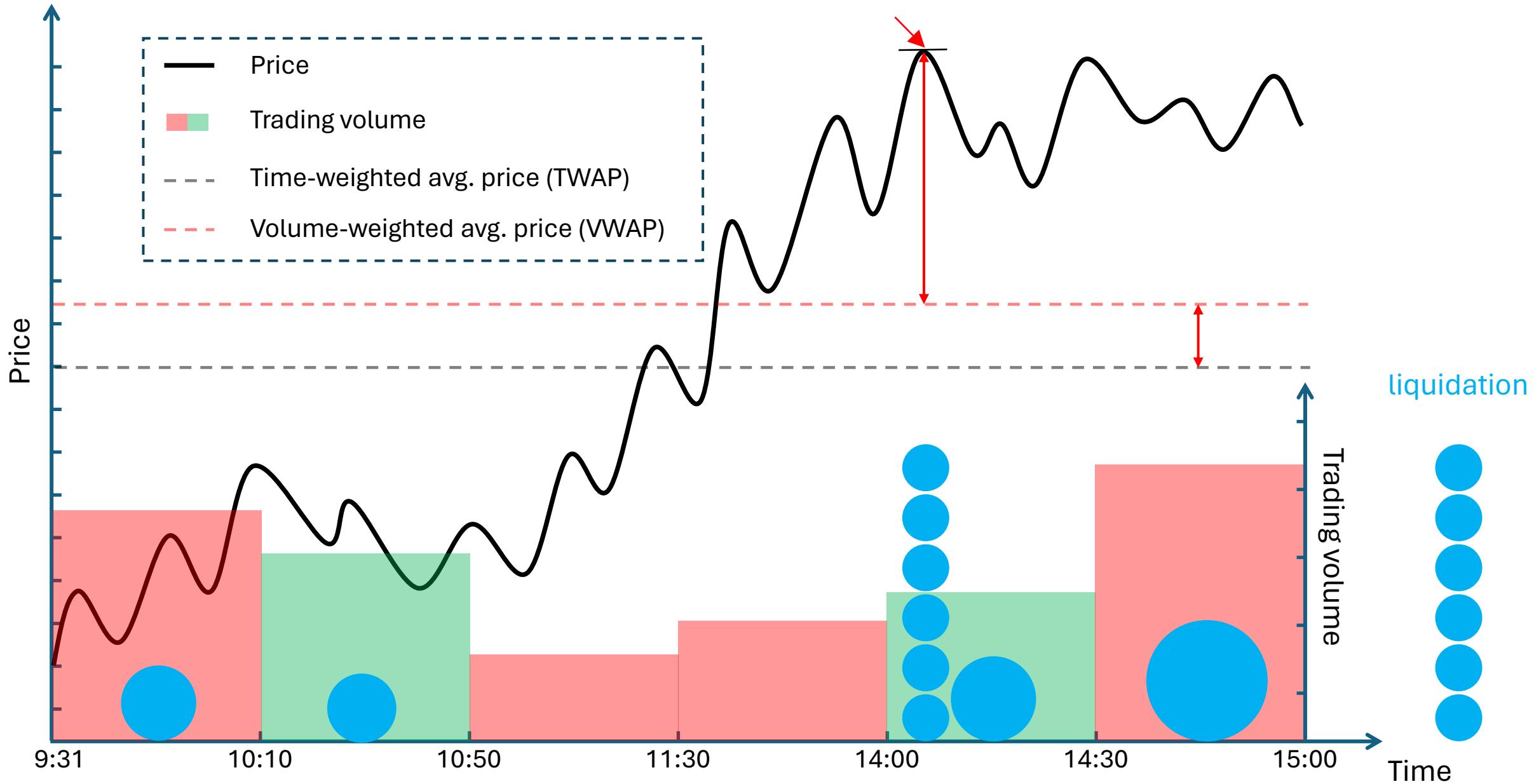
Shanghai Jiao Tong University, Microsoft Research

# Background

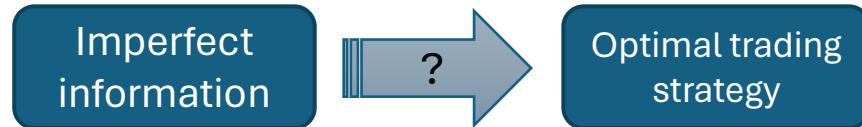
- Portfolio adjustment leads to order execution.



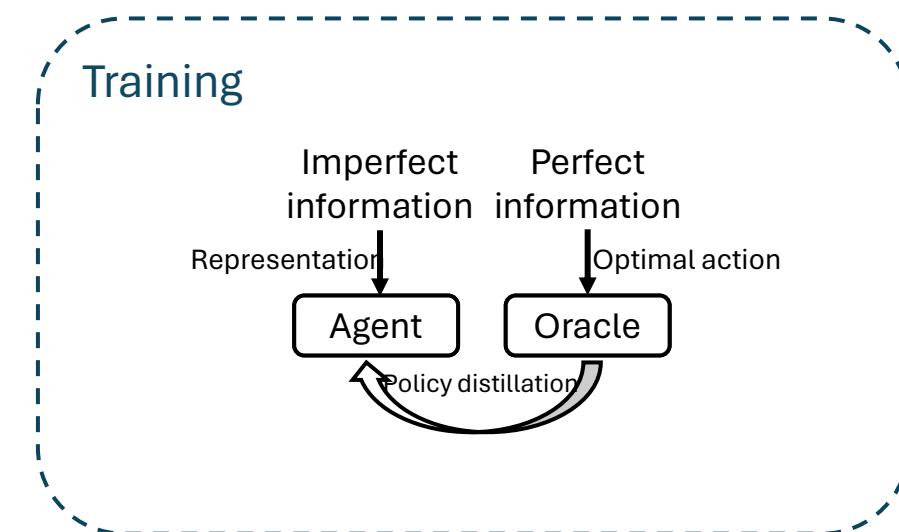
# Background of order execution



# Key challenge and our solution



- Oracle policy distillation
  - Bridge gap between imperfect information and optimal decision making
  - Stabilize policy training and derive more reasonable trading strategy



# Dataset

- China A-share stock market data w/ price-volume information
  - from 2017.1.1 – 2019.6.30
- Split training, validation and test datasets w.r.t. time.

	Training	Validation	Test
# instruments	3566	855 (CSI800)	855 (CSI800)
# order	1,654,385	35,543	33,176
Time period	1/2017 – 2/2019	3/2019 – 4/2019	5/2019 – 6/2019

# Evaluation

- Metric
  - **PA:** price advantage (%oo) to the average market price (TWAP)
  - **Reward** including negative market impacts

Category	Strategy	Reward( $\times 10^{-2}$ )	PA
financial model-based	TWAP (Bertsimas et al. 1998)	-0.42	0
	AC (Almgren et al. 2001)	-1.45	2.33
	VWAP (Kakade et al. 2004)	-0.30	0.32
learning-based	DDQN (Ning et al. 2018)	2.91	4.13
	PPO (Lin et al. 2020)	1.32	2.52
	OPD <sup>S</sup> (pure student) OPD (our proposed)	3.24 <b>3.36*</b>	( <b>5.19</b> ) ( <b>6.17*</b> )

PA	Annual return rate
1.0	0.55%
5.0	2.75%

Table 2: Performance comparison; the higher, the better.

A little bit higher than fixed term deposit. ☺

# Learning Multi-Agent Intention-Aware Communication for Optimal Multi-Order Execution in Finance

KDD 2023

Yuchen Fang, Zhenggang Tang, Kan Ren, Weiqing Liu,  
Li Zhao, Jiang Bian, Dongsheng Li, Weinan Zhang, Yong Yu, Tie-Yan Liu

Microsoft Research, Shanghai Jiao Tong University



# Multi-order execution, challenge 1: cash limit

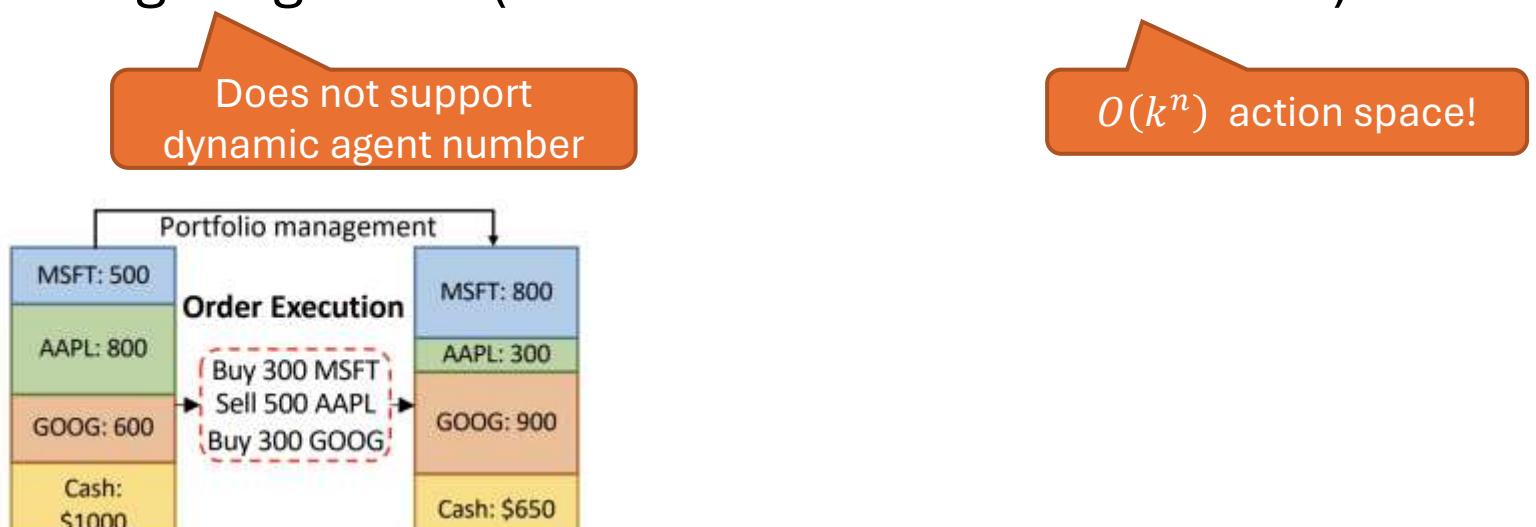


# Multi-order execution, challenge 1: cash limit



# Multi-order execution, challenge 2: order #

- How can multiple decision-making processes be conducted simultaneously?
  - Directly transfer the single-agent RL ( $k$ -discrete action at each time)?



(a) Illustration of financial investment as a combination of portfolio management and multi-order execution.

# Multi-order execution, challenge 3: intention unaware of trading agents



Conflict of action because of intention unawareness.

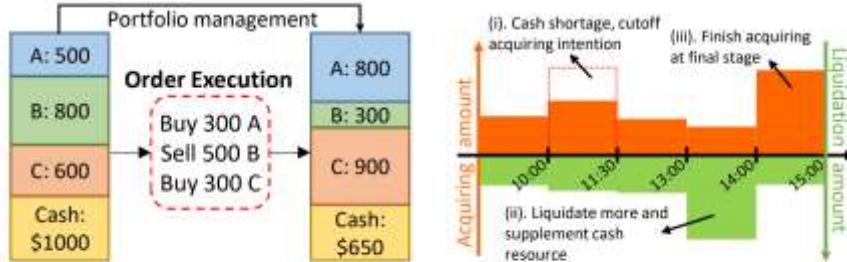
# Our proposal

- A novel method on communication-based MARL solution

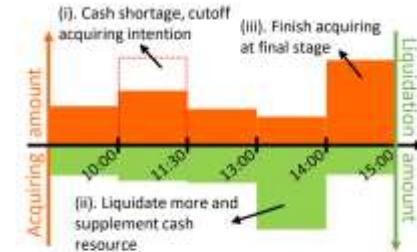


# Experiments

- Financial trading scenario
  - Multi-order execution task
- Datasets
  - China A-share stock market 2018 - 2021
  - US stock market 2017 – 2019
- Evaluation protocol
  - Temporal rolling windows
  - Train on training
  - Hyper-parameter selection on validation
  - Performance comparison on test



(a) Illustration of financial investment as a combination of portfolio management and multi-order execution.



(b) Imbalanced cash utilization requiring further coordination between acquirement and liquidation operations.

Figure 1: An example of multi-order execution and motivation of collaboration within it.

Market	Rolling window	Phase	# order	Time period
China A-share Stock Market	CHW1	Training	910,700	01/01/2018 - 31/08/2018
		Validation	19,740	01/09/2018 - 31/10/2018
		Test	22,480	01/11/2018 - 31/12/2018
	CHW2	Training	966,020	01/01/2019 - 31/08/2019
		Validation	22,820	01/09/2019 - 31/10/2019
		Test	25,220	01/11/2019 - 31/12/2019
	CHW3	Training	1,127,220	01/03/2020 - 31/10/2020
		Validation	26,420	01/11/2020 - 31/12/2020
		Test	20,260	01/01/2021 - 28/02/2021
US Stock Market	USW1	Training	1,000,020	01/01/2017 - 31/12/2017
		Validation	25,620	01/01/2018 - 31/03/2018
		Test	25,220	01/04/2018 - 30/06/2018
	USW2	Training	1,127,220	01/07/2018 - 31/06/2019
		Validation	25,540	01/07/2019 - 30/09/2019
		Test	25,820	01/10/2019 - 31/12/2019

Table 1: The dataset statistics on two real-world markets.

# Profit gain

- Effective annual return (additional to profits brought by portfolio management)
  - China market: 7% annual return, average 30% – 40% improvements
  - US market: 2% annual return, average 20% - 30% improvements

Method Group	Method	China A-share Market Window 1 (CHW1)				China A-share Market Window 2 (CHW2)				China A-share Market Window 3 (CHW3)						
		EG (%) ↑	ARR (%) ↑	POS ↑	GLR ↑	TOC (%) ↓	EG (%) ↑	ARR (%) ↑	POS ↑	GLR ↑	TOC (%) ↓	EG (%) ↑	ARR (%) ↑	POS ↑	GLR ↑	TOC (%) ↓
Financial model based (single-order optimization)	TWAP	0.00	0.00	0.50	1.00	0.00	0.00	0.50	1.00	0.00	0.00	0.50	0.00	1.00	0.00	
	AC	-3.26	-0.65	0.48	1.00	0.00	-1.25	-0.25	0.49	0.96	0.00	-6.14	-1.22	0.48	0.95	0.00
Single-agent RL (single-order optimization)	VWAP	-3.26	-0.65	0.49	1.01	0.00	-2.23	-0.45	0.48	0.92	0.00	-6.13	-1.22	0.48	0.95	0.00
	PPO	21.63±1.45	5.56±0.36	0.59±0.01	1.27±0.05	40.39±6.46	24.36±2.32	6.28±0.58	0.60±0.01	1.23±0.04	45.12±10.95	20.01 ± 1.11	5.13±0.28	0.58±0.01	1.26±0.03	30.24±5.37
Multi-agent RL (multi-order optimization)	DDQN	6.25±0.27	1.57±0.07	0.53±0.01	1.05±0.01	8.43±1.72	7.12±0.64	1.80±0.16	0.54±0.01	1.06±0.03	18.27±3.23	7.07±0.36	1.78±0.14	0.33±0.02	1.03±0.01	10.27±1.14
	CommNet	20.32±0.98	5.21±0.25	0.60±0.01	1.20±0.03	2.45±0.43	30.21±1.89	7.84±0.47	0.59±0.02	1.33±0.03	6.87±1.29	21.00±1.24	5.39±0.31	0.58±0.01	1.30±0.01	2.98±0.99
Multi-agent RL (multi-order optimization)	TarMAC	22.46±1.42	5.77±0.36	0.57±0.03	1.29±0.01	2.56±0.10	31.12±0.88	8.09±0.22	0.60±0.01	1.38±0.02	3.75±0.48	21.89±0.70	5.62±0.15	0.58±0.01	1.31±0.01	4.03±1.53
	IS	21.22±2.88	5.45±0.72	0.58±0.01	1.28±0.02	3.02±0.52	30.01±0.35	7.79±0.89	0.59±0.01	1.35±0.04	5.23±0.51	22.04±1.13	5.66±0.28	0.59±0.01	1.37±0.05	4.34±0.56
Multi-agent RL (multi-order optimization)	IaC <sup>C</sup>	<b>28.38±2.34*</b>	<b>7.35±0.39*</b>	<b>0.64±0.02*</b>	<b>1.33±0.01</b>	<b>1.56±0.05</b>	<b>32.28±0.21</b>	<b>8.40±0.05</b>	<b>0.60±0.01</b>	<b>1.38±0.01</b>	<b>1.03±0.33</b>	<b>24.78±1.02</b>	<b>6.39±0.26</b>	<b>0.60±0.01</b>	<b>1.39±0.01*</b>	<b>1.29±0.43*</b>
	IaC <sup>T</sup>	28.36±3.45	7.35±0.86	0.63±0.03	1.34±0.01*	1.23±0.36*	33.01±0.18*	8.60±0.05*	0.61±0.01*	1.41±0.01*	1.58±0.32*	25.45±1.22*	6.57±0.31*	0.60±0.09*	1.38±0.02	1.38±0.48

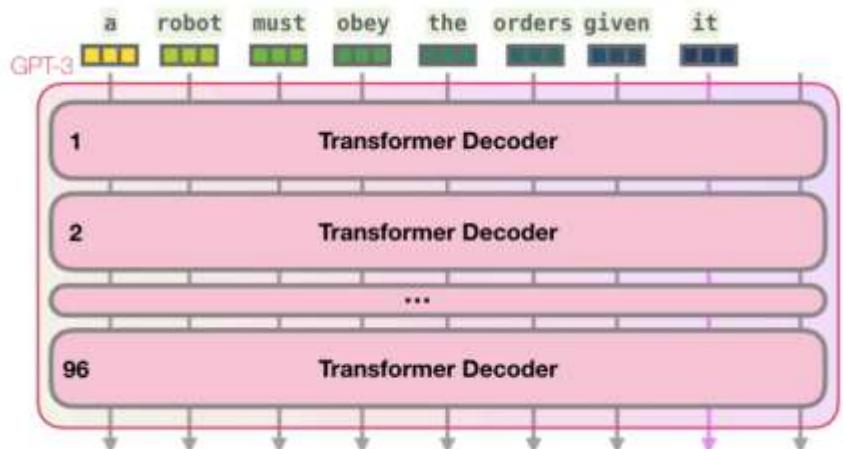
Method Group	Method	US Stock Market Window 1 (USW1)				US Stock Market Window 2 (USW2)					
		EG (%) ↑	ARR (%) ↑	POS ↑	GLR ↑	TOC (%) ↓	EG (%) ↑	ARR (%) ↑	POS ↑	GLR ↑	TOC (%) ↓
Financial model based (single-order optimization)	TWAP	0.00	0.00	0.50	1.00	0.00	0.00	0.00	0.50	1.00	0.00
	AC	1.25	0.25	0.51	1.01	0.00	0.34	0.07	0.50	1.01	0.00
Single-agent RL (single-order optimization)	VWAP	-2.56	-0.51	0.49	0.97	0.00	-1.25	-0.25	0.49	0.99	0.00
	PPO	4.45±0.56	1.12±0.14	0.53±0.01	1.02±0.05	13.25±2.78	4.32±1.13	1.09±0.28	0.53±0.01	0.98±0.05	10.25±1.57
Multi-agent RL (multi-order optimization)	DDQN	1.12±0.34	0.28±0.09	0.50±0.01	1.02±0.01	6.35±0.90	2.21±0.36	0.55±0.09	0.53±0.01	0.97±0.03	5.34±1.96
	CommNet	5.99±1.24	1.51±0.31	0.53±0.01	1.04±0.02	1.25±0.40	6.01±0.10	1.51±0.03	0.54±0.02	1.05±0.01	1.89±0.24
Multi-agent RL (multi-order optimization)	TarMAC	6.78±1.24	1.71±0.31	0.53±0.03	1.04±0.02	2.45±1.01	5.98±0.48	1.51±0.12	0.52±0.01	1.05±0.01	1.24±0.1
	IS	5.67±0.93	1.43±0.23	0.52±0.01	1.03±0.03	3.60±1.29	4.65±0.28	1.17±0.07	0.52±0.01	1.04±0.04	4.32±0.78
Multi-agent RL (multi-order optimization)	IaC <sup>C</sup>	7.78±0.16	1.96±0.04	0.54±0.01	<b>1.06±0.01*</b>	1.30±0.04	7.87±0.12	1.99±0.03	0.54±0.01	1.08±0.02	<b>1.02±0.15*</b>
	IaC <sup>T</sup>	8.02±0.54*	2.03±0.14*	0.54±0.00*	1.05±0.01	1.03±0.02*	7.99±0.34*	2.02±0.09*	0.55±0.01*	1.10±0.01*	1.14±0.22

Table 2: The results of all the compared methods on five rolling windows of two real-world markets. ↑ (↓) means the higher (lower) value is better. For learning-based methods, we report the mean value  $\pm$  standard deviation over six random seeds. The best results of learning-based methods are highlighted with bold format. \* indicates  $p\text{-value} < 10^{-6}$  in significance test [4].

# Sequential decision making in GPT models



{robot, it, a, given, obey, ..., human, ...}



RL concept	NLP concept
State space	The given sentence
Action space	The dictionary
Policy	GPT (language model)
Reward	Sentence quality

# Autonomous agents

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

ChatGPT

Model: Code interpreter ALPHA

Create a Shepard tone and save it as a wav.

A Shepard tone is an auditory illusion that creates the impression of a tone that continuously ascends or descends in pitch, but never seems to get any higher or lower. It consists of several sine waves, each an octave apart, that are overlapped and looped. The amplitude of each sine wave changes over time in a way that creates the illusion.

To create a Shepard tone, we will:

1. Define a range of frequencies, each an octave apart.
2. Generate sine waves for each frequency.
3. Modulate the amplitude of each sine wave to create the illusion of a continuous ascending or descending pitch.
4. Combine the sine waves to create the Shepard tone.
5. Save the tone as a .wav file.

Let's create a Shepard tone that gives the illusion of continuously ascending pitch. We'll start by defining some parameters and generating the sine waves.

Finished working Show work ↗

The Shepard tone has been successfully generated and saved as a WAV file. You can download the ascending Shepard tone from the following link: [ascending\\_shepard\\_tone.wav](#).



Model: Code interpreter ALPHA

I want you to OCR an image and give me a .txt file

Trigger OCR functions

Sure, I can help you with that. Please provide me with the image you'd like to OCR (optical character recognition) so that I can extract the text and create a .txt file for you.

Finished working Show work ↗

The text has been successfully extracted from the image and saved to a .txt file. You can download the .txt file from the following link: [ocr-output.txt](#).

Please let me know if there is anything else I can assist you with!

Creating sound

# Face finding with OpenCV



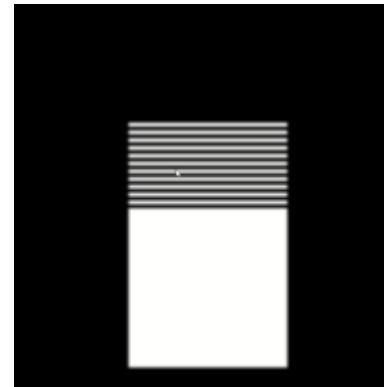
## Uploaded image



Converted to ASCII



## Generative Animation



# Creating machine learning models

Can we build our own machine learning model from scratch here?

Yes, we can build a simple machine learning model from scratch using Python. What kind of problem do you want to solve with the model?

Next word prediction

Next word prediction is a natural language processing task that can be tackled with a language model. One simple approach to building a language model is to use an n-gram model.

An n-gram model is a statistical language model that predicts the next word based on the previous n words. For example, if we choose n=2, a bigram model, the model would predict the next word based on the previous two words.

Leave some room to get you started with building a bigram model from scratch:

```
from collections import defaultdict, Counter

def bigram(LanguageModel):
    pass

# This is a bigram model on the given corpus.
# Args:
#     corpus: A string containing the text corpus for training.
# Returns:
#     A dictionary where keys are tuples of two words (k1, k2) and values
#     are integers representing the count of such a pair in the corpus.
model = bigram(Counter())

# Insertion into corpus (one word at a time)
words = corpus.split()

# Iterating over the words
for i in range(len(words) - 1):
    # Requirements response
    print(model[words[i], words[i + 1]])
```

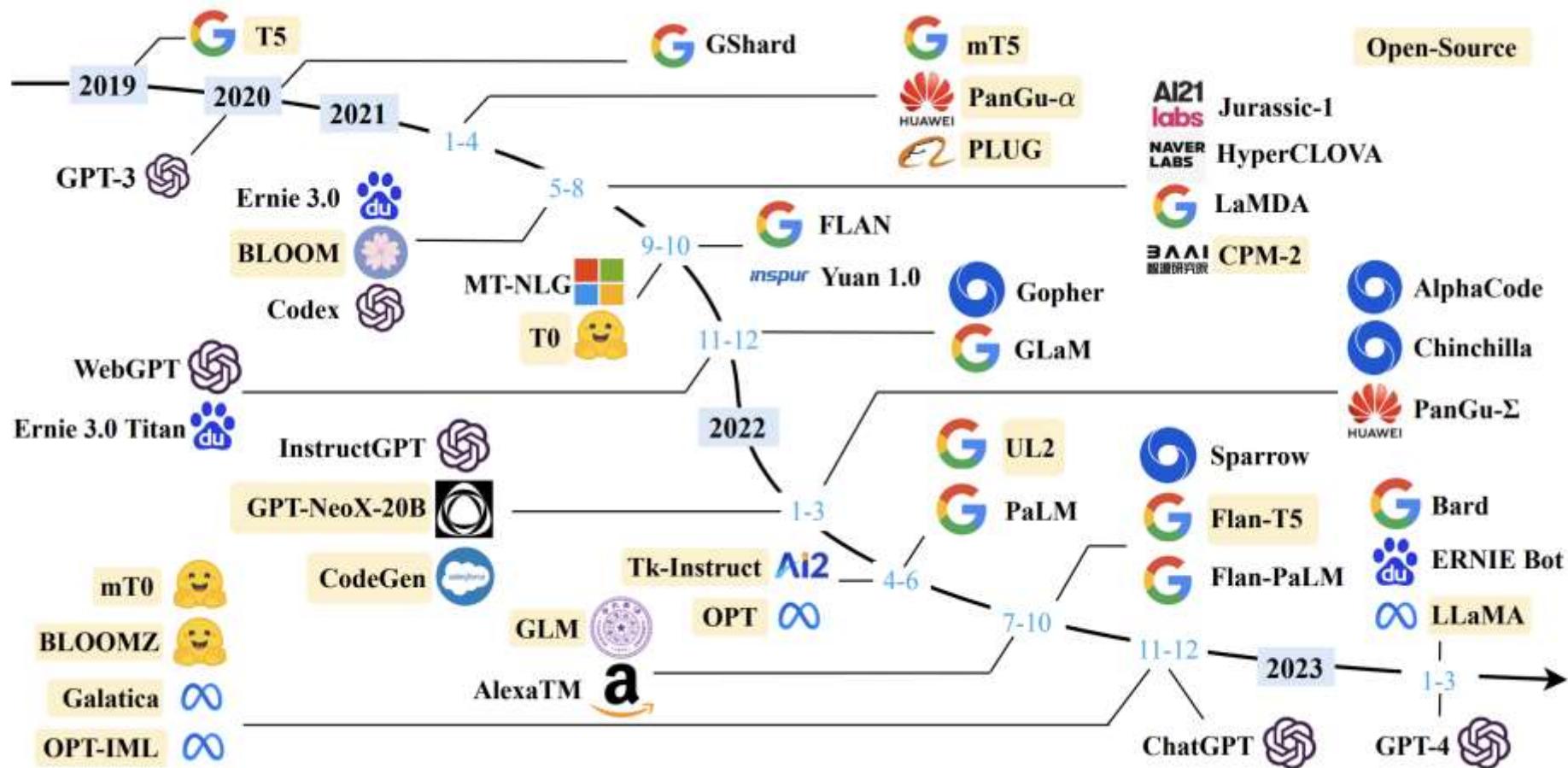
# **Solving AI Tasks with ChatGPT and Its Friends in Hugging Face**

Yongliang Shen, Kaitao Song, Xu Tan,  
Dongsheng Li, Weiming Lu, Yueling Zhuang  
Microsoft Research

NeurIPS 2023  
Paper: <https://arxiv.org/abs/2303.17580>  
Code: <https://github.com/microsoft/JARVIS/>

# Background

- The rise of LLMs



# Thinking

## LLMs

Stanford  
Alpaca



MOSS



Open Assistant

OpenAI  
ChatGPT

文心一言

## ML community / Tool Library / API



Azure



Hugging Face

GitHub

python

...

# Thinking

- **ChatGPT**
  - A ChatBot released by OpenAI, with powerful capability in generation, reasoning, understanding and other emergent ability.
- **Hugging Face**
  - The world's largest AI Model Community.
  - Hosting 175,195 AI models covering 36 tasks with different modalities.
  - Detailed model metadata and descriptions.

# ChatGPT + Hugging Face

- Pipeline
  - Task Planning
  - Model Selection
  - Task Execution
  - Response Generation

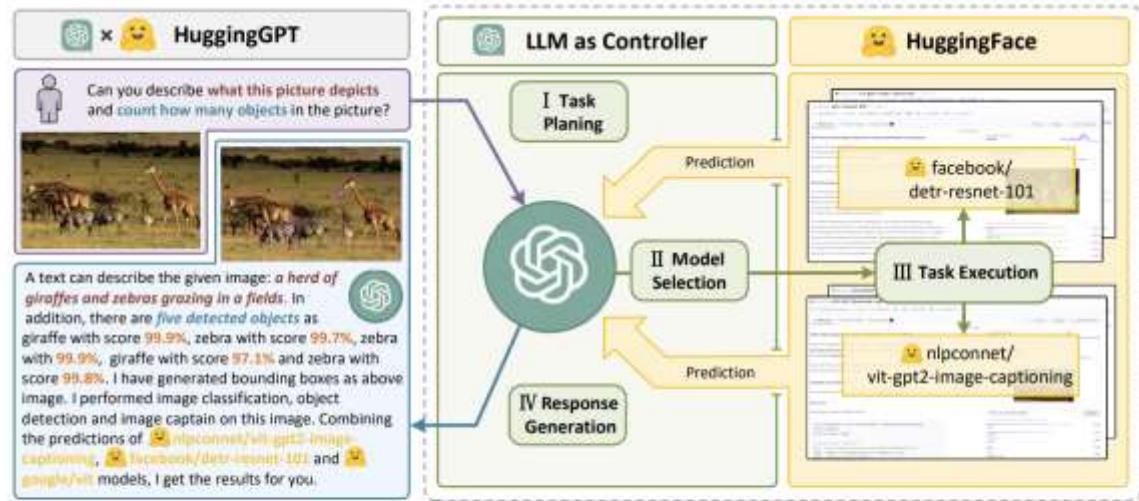


Figure 1: Language serves as an interface for LLMs (e.g., ChatGPT) to connect numerous AI models (e.g., those in Hugging Face) for solving complicated AI tasks. In this concept, an LLM acts as a controller, managing and organizing the cooperation of expert models. The LLM first plans a list of tasks based on the user request and then assigns expert models to each task. After the experts execute the tasks, the LLM collects the results and responds to the user.

# Method

- Implementation -> Prompt

Task Planning	Prompt	Model Selection	Prompt
	#1 Task Planning Stage - The AI assistant can parse user input to several tasks: <code>[{"task": task, "id": task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio": URL, "video": URL}]</code> . The "dep" field denotes the id of the previous task which generates a new resource that the current task relies on. A special tag " <code>&lt;resource&gt;-task_id</code> " refers to the generated text image, audio and video in the dependency task with id as task_id. The task MUST be selected from the following options: <code>{ Available Task List }</code> . There is a logical relationship between tasks, please note their order. If the user input can't be parsed, you need to reply empty JSON. Here are several cases for your reference: <code>{ Demonstrations }</code> . The chat history is recorded as <code>{ Chat Logs }</code> . From the chat logs, you can find the path of the user-mentioned resources for your task planning.		#2 Model Selection Stage - Given the user request and the call command, the AI assistant helps the user to select a suitable model from a list of models to process the user request. The AI assistant merely outputs the model id of the most appropriate model. The output must be in a strict JSON format: <code>"id": "id", "reason": "your detail reason for the choice"</code> . We have a list of models for you to choose from <code>{ Candidate Models }</code> . Please select one model from the list.
	Demonstrations		Candidate Models
	Look at /exp1.jpg, Can you tell me how many objects in the picture?		<code>{"model_id": model id #1, "metadata": meta-infos #1, "description": description of model #1}</code> <code>{"model_id": model id #2, "metadata": meta-infos #2, "description": description of model #2}</code> [ Select top-K candidate models with the same task type as the current task ] <code>{"model_id": model id #K, "metadata": meta-infos #K, "description": description of model #K}</code>
Response Generation	In /exp2.jpg, what's the animal and what's it doing?		Prompt
	Given an image /exp3.jpg, first generate a hed image, then based on the hed image and a prompt: a girl is reading a book, you need to reply with a new image.		#4 Response Generation Stage - With the input and the inference results, the AI assistant needs to describe the process and results. The previous stages can be formed as - User Input: <code>{ User Input }</code> , Task Planning: <code>{ Tasks }</code> , Model Selection: <code>{ Model Assignment }</code> , Task Execution: <code>{ Predictions }</code> . You must first answer the user's request in a straightforward manner. Then describe the task process and show your analysis and model inference results to the user in the first person. If inference results contain a file path, must tell the user the complete file path. If there is nothing in the results, please tell me you can't make it.

# Examples

- Simple Tasks

## NLP > Named Entity Recognition

**Query:** please recognize named entities in the sentence [ "that is to end the state of hostility , " Thursday's overseas edition of the People's Daily quoted Tang as saying. ]

**Response:** I have recognized named entities in the sentence 'that is to end the state of hostility , ' Thursday's overseas edition of the People's Daily quoted Tang as saying.' . The entities recognized are 'People's Daily' and 'Tang'. Both of them are organizations and people respectively. The model I used is ' mbert-base-uncased', which is a fine-tuned BERT model that is ready to recognize named entities in the sentence. The inference results are very accurate, with a score of 0.9992 for 'People's Daily' and 0.998 for 'Tang'.

## NLP > Text Generation

**Query:** please complete a sentence [ COVID-19 is transmitted via droplets, air-borne, ]

**Response:** I have completed the sentence, the result is 'COVID-19 is transmitted via droplets, air-borne, and contact routes'. I used a  microsoft/pegasus model, a domain-specific generative Transformer language model pre-trained on large-scale biomedical literature.

## CV > Depth Estimate

**Query:** please estimate the depth of ks4j.jpg



**Response:** I have estimated the depth of the image located at examples/ks4j.jpg using the  intel/dpt-large model. This model is specifically trained on 1.4 million images and is a Dense Prediction Transformer (DPT) model. The generated depth image can be found at ' j45s.jpg'.



## CV > Object Detection

**Query:** please detect all objects in examples.jpg

**Response:** I have detected all objects in the image located at  examples.jpg. I used the  facebook/detr-resnet-101 model for object detection. It is a DETR (End-to-End Object Detection) model with ResNet-101 backbone, which is known for its high accuracy in object detection tasks. I generated an image with predicted boxes and labels for each object. The complete path of the generated image is examples/d7da.jpg. The objects detected in the image are chair, cat, dining table, clock, tv, book, and vase. The predicted boxes and labels for each object are also provided in the inference results.



# Examples

- Complex Tasks

- REQUEST: Generate a new image with a similar pose as the left one and similar content as the right one.
- Task Planning:
  - Pose Detection
  - Image Caption
  - Pose Conditional Text-to-Image
- Model Selection:
  - Pose Detection => Openpose Detector
  - Image Caption => nlpconnect/vit-gpt2-image-captioning
  - Pose Conditional Text-to-Image => pose-sd-controlnet

HuggingGPT

Human :  
please based on the pose image of [Image] and the description of [Image], generate a new image



HuggingGPT :  
Based on the pose image of [Image] and the description of [Image], I have generated a new image for you. The process of generating this image involved using the openpose-control model to extract the pose information from the basketball image, and then using the image-to-text model to generate a description of the boy in the image. Finally, I used the openpose-text-to-image model to combine the pose information and the description to generate the final image. The complete path of the generated image is [Image]. If you have any other questions, please let me know.



# Examples

- Complex Tasks

- REQUEST: Is this a healthy meal or unhealthy meal?
- Task Planning:
  - Image Caption
  - QA
- Model Selection:
  - Image Caption => nlpconnect/vit-gpt2-image-captioning
  - QA => ChatGPT

HuggingGPT

Human : Look at the image [Image]. Is this a healthy meal or unhealthy meal?



HuggingGPT : This is a photograph of 2 bowls of food. One bowl contains chicken and broccoli while the other contains grapes. Based on my analysis, I believe this meal to be a healthy one as it includes protein in the form of chicken and vegetables in the form of broccoli. The model that I have used for this inference is 'Salesforce/blip-image-captioning-large' and the complete file path is [Image].



default ▾ Submit

HuggingGPT

Human : Look at the image [Image]. Is this a healthy meal or unhealthy meal?



HuggingGPT : This is a photograph of 2 bowls of food. One bowl contains chicken and broccoli while the other contains grapes. Based on my analysis, I believe this meal to be a healthy one as it includes protein in the form of chicken and vegetables in the form of broccoli. The model that I have used for this inference is 'Salesforce/blip-image-captioning-large' and the complete file path is [Image].



default ▾ Submit

# **MLCopilot: Unleashing the Power of Large Language Models in Solving Machine Learning Tasks**

Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, Yuqing Yang

Microsoft Research

Preprint work

Paper: <https://arxiv.org/abs/2304.14979>

Code: <https://github.com/microsoft/CoML/>

# Endless design choices in machine learning

Which one should I use?



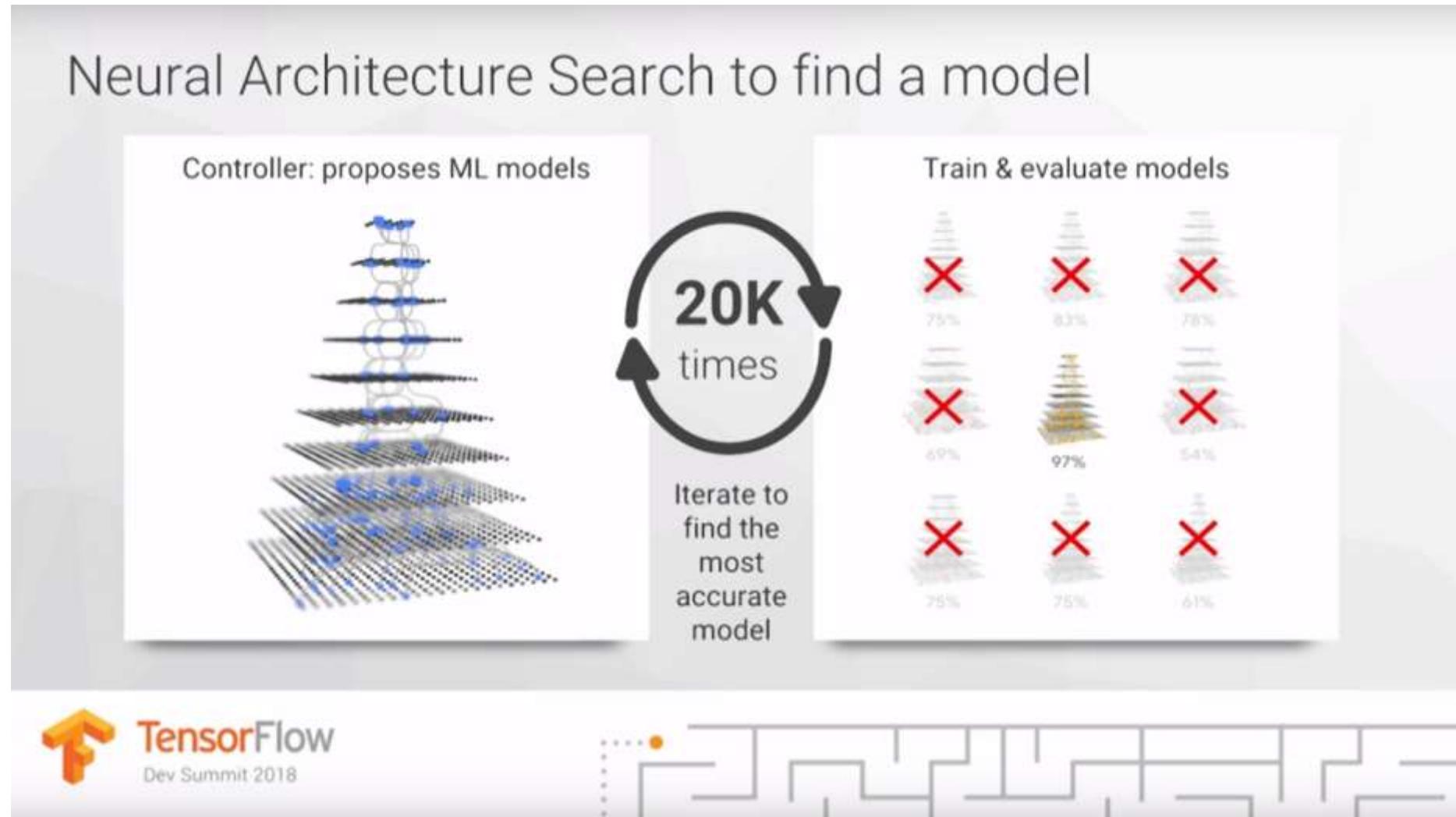
## sklearn.ensemble: Ensemble Methods

The `sklearn.ensemble` module includes ensemble-based methods for classification, regression and anomaly detection.

**User guide:** See the [Ensemble methods](#) section for further details.

<code>ensemble.AdaBoostClassifier([estimator, ...])</code>	An AdaBoost classifier.
<code>ensemble.AdaBoostRegressor([estimator, ...])</code>	An AdaBoost regressor.
<code>ensemble.BaggingClassifier([estimator, ...])</code>	A Bagging classifier.
<code>ensemble.BaggingRegressor([estimator, ...])</code>	A Bagging regressor.
<code>ensemble.ExtraTreesClassifier([...])</code>	An extra-trees classifier.
<code>ensemble.ExtraTreesRegressor([n_estimators, ...])</code>	An extra-trees regressor.
<code>ensemble.GradientBoostingClassifier(*[, ...])</code>	Gradient Boosting for classification.
<code>ensemble.GradientBoostingRegressor(*[, ...])</code>	Gradient Boosting for regression.
<code>ensemble.IsolationForest(*[, n_estimators, ...])</code>	Isolation Forest Algorithm.
<code>ensemble.RandomForestClassifier([...])</code>	A random forest classifier.
<code>ensemble.RandomForestRegressor([...])</code>	A random forest regressor.
<code>ensemble.RandomTreesEmbedding([...])</code>	An ensemble of totally random trees.
<code>ensemble.StackingClassifier(estimators[, ...])</code>	Stack of estimators with a final classifier.
<code>ensemble.StackingRegressor(estimators[, ...])</code>	Stack of estimators with a final regressor.
<code>ensemble.VotingClassifier(estimators, *[..., ...])</code>	Soft Voting/Majority Rule classifier for unfitted estimators.
<code>ensemble.VotingRegressor(estimators, *[..., ...])</code>	Prediction voting regressor for unfitted estimators.
<code>ensemble.HistGradientBoostingRegressor([...])</code>	Histogram-based Gradient Boosting Regression Tree.
<code>ensemble.HistGradientBoostingClassifier([...])</code>	Histogram-based Gradient Boosting Classification Tree.

# Searching for optimal ML solution is impractical



# ... especially with combinations are too many

```
net = torchvision.models.resnet50()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(lr=0.01, betas=(0.9, 0.999), eps=1e-08, params=net.parameters())
scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=10, gamma=0.1)
n_epochs = 10
for i in range(n_epochs):
    scheduler.step()
    print('Epoch: ', i)
```

Could be resnet101, efficientnetb0, mobilenetv3...

Could be SoftCrossEntropyLoss...

Could be SGD. Could be a different set of parameters.

Could be a another scheduler, or none.

Could use a different epoch number.

# GitHub Copilot: powered by Large Language Models



```
28  class Net(nn.Module):
29      def __init__(self):
30          super().__init__()
31          self.conv1 = nn.Conv2d(3, 6, 5)
32          self.pool = nn.MaxPool2d(2, 2)
33          self.conv2 = nn.Conv2d(6, 16, 5)
34          self.fc1 = nn.Linear(16 * 5 * 5, 120)
35          self.fc2 = nn.Linear(120, 84)
36          self.fc3 = nn.Linear(84, 10)
37
38      def forward(self, x):
39          x = self.pool(F.relu(self.conv1(x)))
40          x = self.pool(F.relu(self.conv2(x)))
41          x = torch.flatten(x, 1) # flatten all dimensions except batch
42          x = F.relu(self.fc1(x))
43          x = F.relu(self.fc2(x))
44          x = self.fc3(x)
45
46
47      net = Net()
48      import torch.optim as o
49
50      criterion = nn.CrossEnt
51
52      optimizer = optim.Adam(net.parameters(), lr=0.001)
```

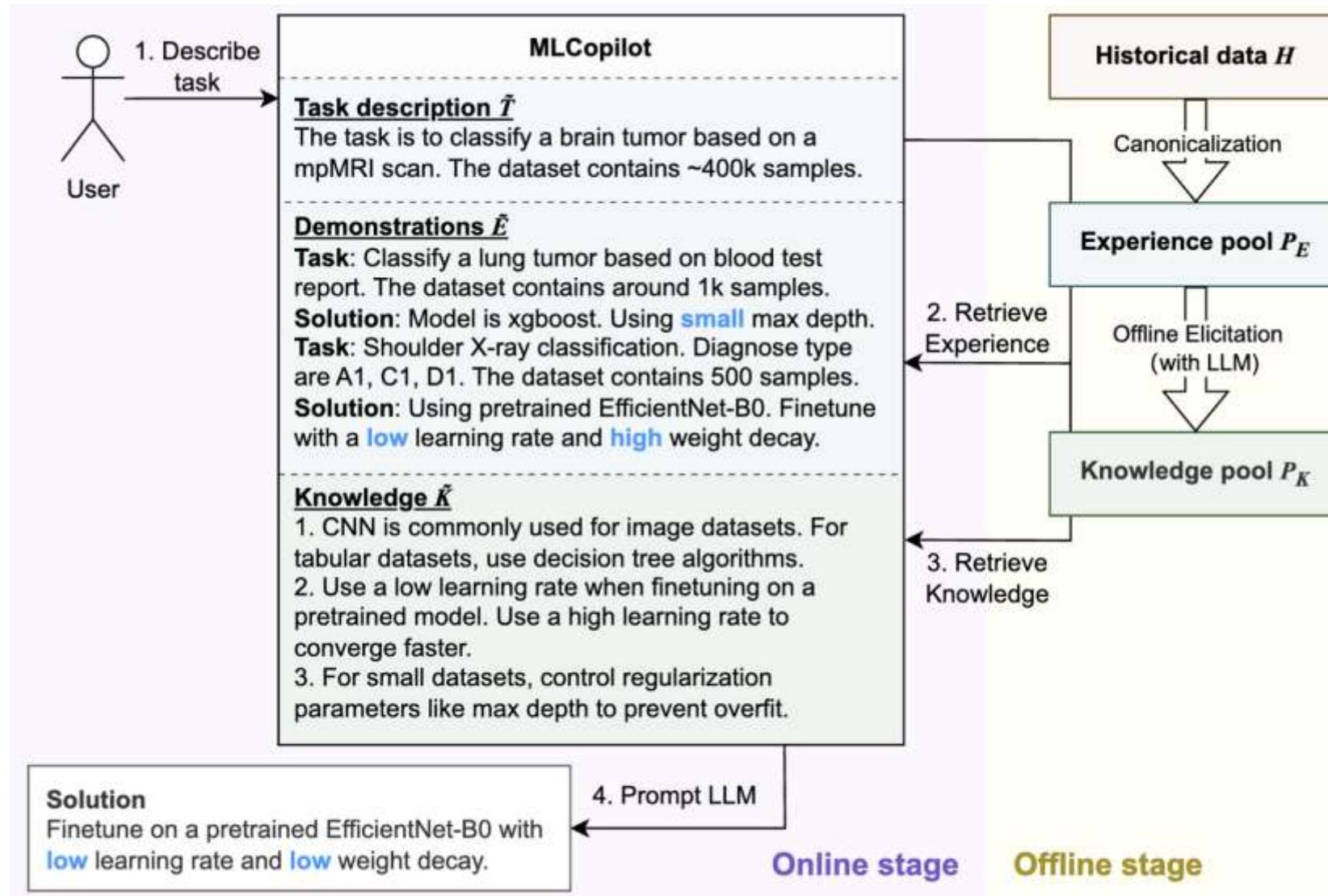
(params: `_params_t`, lr: float = ..., betas: Tuple[float, float] = ..., eps: float = ..., weight\_decay: float = ..., amsgrad: bool = ..., \*, foreach: bool | None = ..., maximize: bool = ..., capturable: bool = ..., differentiable: bool = ..., fused: bool = ...) -> None



**GitHub  
Copilot**

Not good at 调参!

# MLCopilot



# Essence of MLCopilot is a ML knowledge database

The database contains historical data and **auto-summarized** knowledge.

Space: 5971

1. Generally, larger datasets require higher nrounds and larger subsample values.
2. The majority class size and minority class size of the dataset can influence the configuration of alpha, booster, colsample bylevel, colsample bytree, eta, lambda, max depth, min child weight, nrounds, and subsample.
3. The number of numeric and categorical features in the dataset can determine the booster used.
4. The size of the dataset can influence the configuration of eta, lambda, max depth, min child weight, nrounds, and subsample.
5. The size of the minority class can determine the configuration of alpha, colsample bylevel, colsample bytree, eta, lambda, max depth, min child weight, nrounds, and subsample.

Test task: MAFA

1. Set the crop size and anchor matching IoU threshold according to the number of faces per image in the dataset: larger crop sizes and higher IoU thresholds for datasets with more faces per image, and smaller crop sizes and lower IoU thresholds for datasets with fewer faces per image.
2. Set the location loss weight according to the presence of facial landmarks in the dataset: higher weights for datasets with facial landmarks, and lower weights for datasets without facial landmarks.
3. Set the negative to positive ratio according to the difficulty of the dataset: higher ratios for datasets with more challenging scenarios (e.g. weather-based degradations, motion blur, focus blur).
4. Set the learning rate and optimizer according to the size of the dataset: lower learning rates and optimizers such as Adam or SGD for datasets with more images.

# Main results

- LLM-ZS: Zero-shot
  - i.e, GitHub Copilot
- LLM-FS: Few-shot

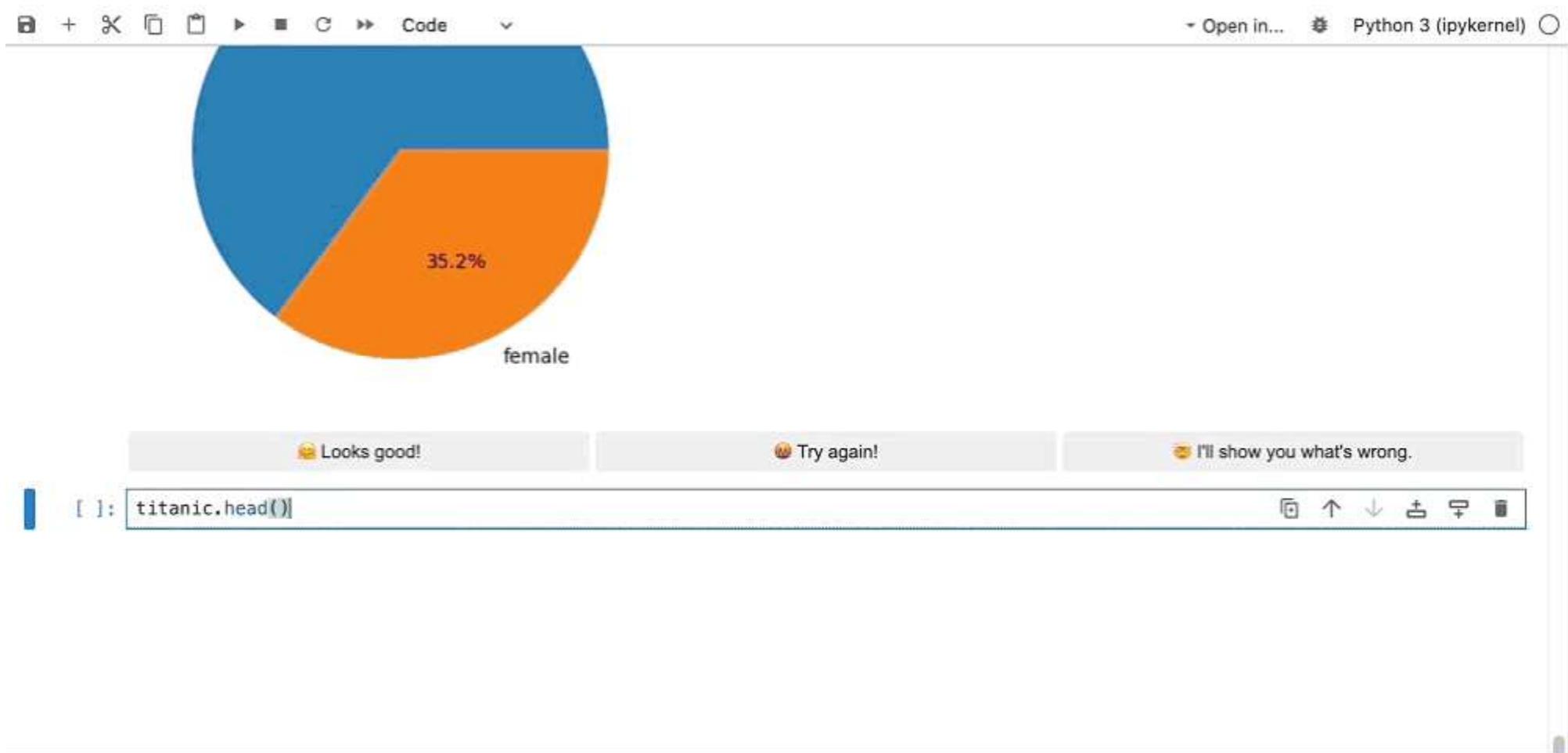
Method	nAcc@1	nAcc@2	nAcc@3	Method	nAcc@1	nAcc@2	nAcc@3
Random	54.70	60.70	64.80	Random	-0.86	-0.08	0.39
Constant	72.85	74.61	75.02	Constant	1.27	1.56	1.59
TST-M	72.73	74.44	74.56	TST-M	1.10	1.35	1.40
HyperSTAR	67.37	68.14	68.71	FLAML	1.28	1.31	1.58
ASKL	77.01	81.76	85.02	ASKL	1.26	1.29	1.44
FLAML	77.84	82.95	88.06	HyperSTAR	1.10	1.27	1.34
LLM-ZS	61.37	79.41	80.56	LLM-ZS	-1.03	1.25	1.26
LLM-FS	78.93	83.10	89.73	LLM-FS	0.43	0.57	0.62
MLCopilot	81.59	83.23	90.72	MLCopilot	1.48	1.54	1.62

Table 2: Results on HPO-B. The higher the better. Table 3: Results on PD1. The higher the better.

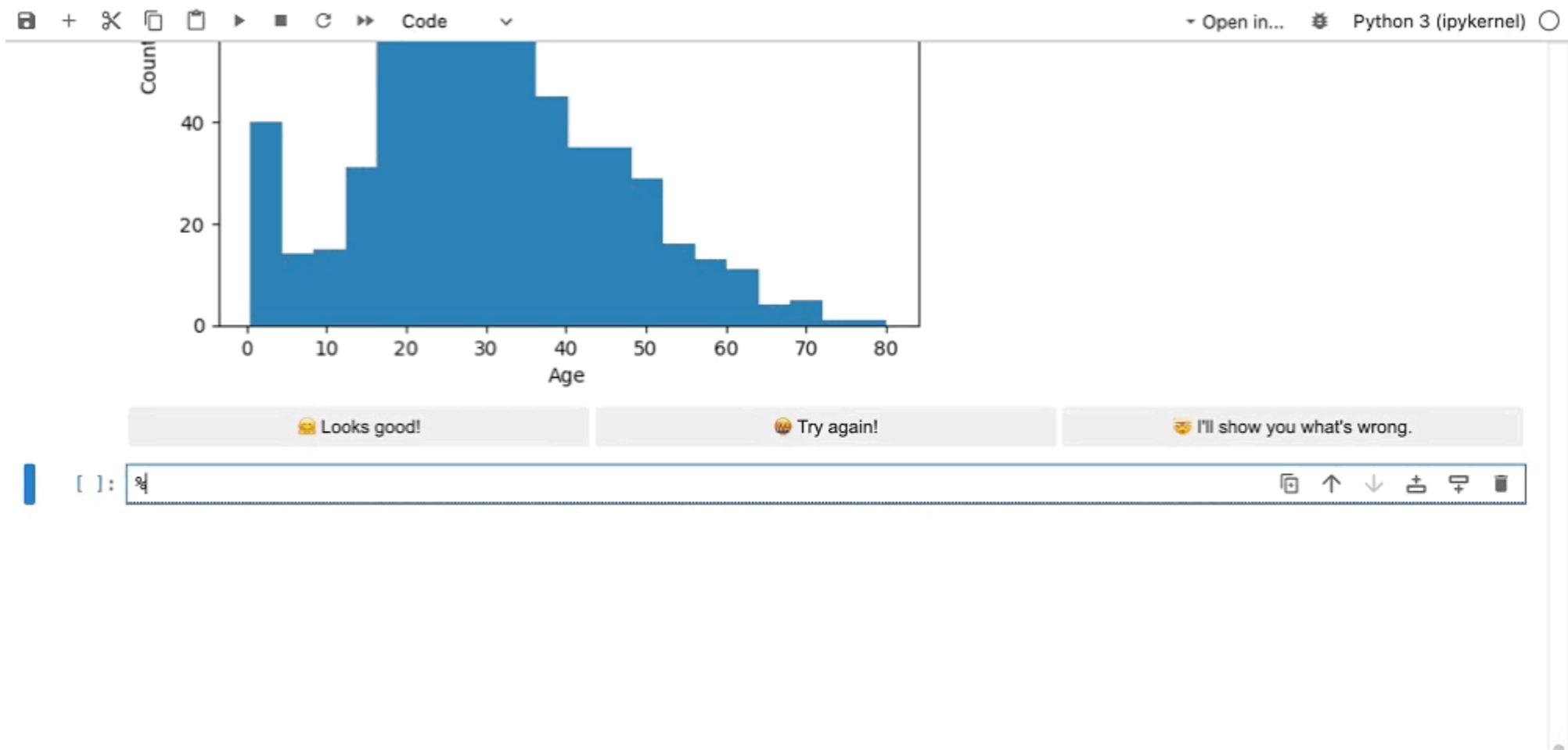
Method	Rank@1 ↓	Rank@2 ↓	Rank@3 ↓	AP@1 ↑	AP@2 ↑	AP@3 ↑
Random	109.55	73.16	54.79	90.76	91.20	91.38
Constant	78.00	54.33	49.25	91.13	91.23	91.28
TST-M	57.67	43.50	42.67	91.22	91.37	91.38
ASKL	92.58	64.67	51.25	90.93	91.15	91.34
HyperSTAR	97.75	72.97	52.03	90.78	91.05	91.25
FLAML	66.42	43.33	31.83	91.09	91.29	91.33
HyperFD	56.97	47.91	31.75	91.17	91.26	91.44
LLM-ZS	119.25	90.42	41.00	90.69	90.96	91.38
LLM-FS	66.69	52.43	40.98	91.26	91.40	91.48
MLCopilot	59.74	38.67	25.58	91.38	91.60	91.66

Table 4: Results on HyperFD.

# Data science problem solving



# Automatic debugging and fixing



# Modeling sequence data

- Prediction
- Generation
- Self-supervised learning

# Protecting the Future: Neonatal Seizure Detection with Spatial-Temporal Modeling

SMC 2023

Ziyue Li, Yuchen Fang, You Li, Kan Ren, Yansen Wang, Xufang Luo, Juanyong Duan, Congrui Huang,  
Dongsheng Li, Lili Qiu

Microsoft Research, ShanghaiTech University, Shanghai Jiao Tong University

# SIMPLE: Specialized Model-Sample Matching for Domain Generalization

ICLR 2023

Ziyue Li, Kan Ren, XINYANG JIANG, Yifei Shen, Haipeng Zhang, Dongsheng Li  
ShanghaiTech University, Microsoft Research, Shanghai Jiao Tong University

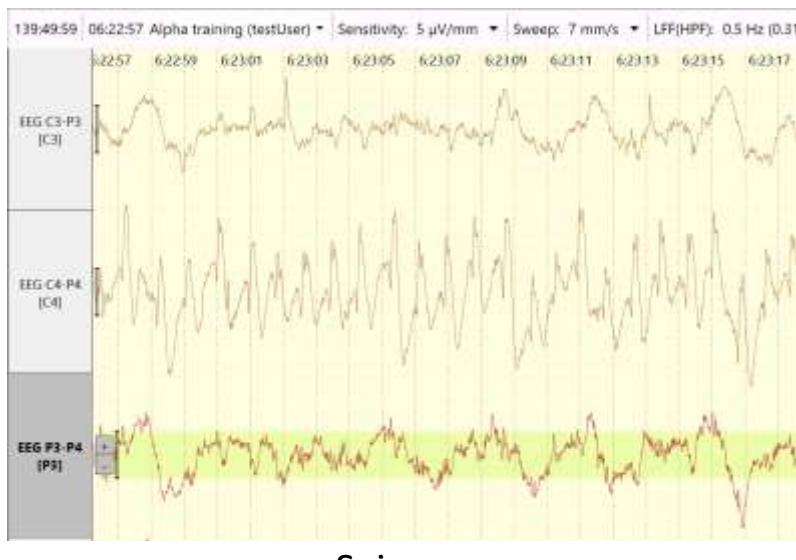


# Brain signal based seizure <sub>癫痫</sub> detection



# Brain signal based seizure 癫痫 detection

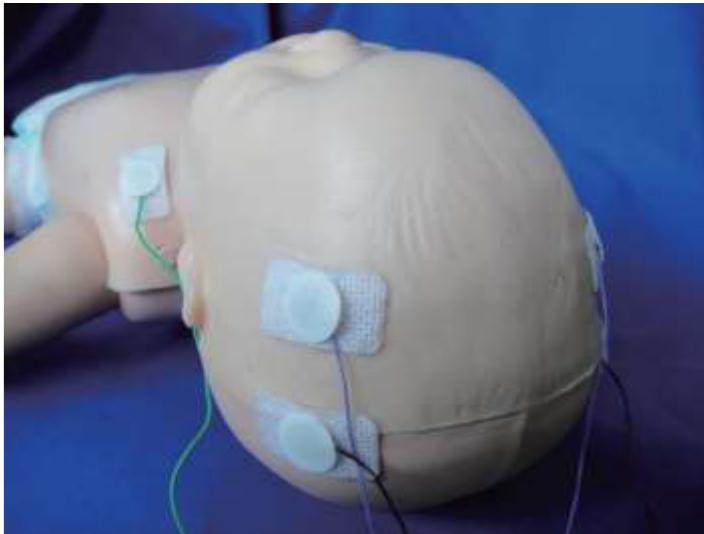
- Epilepsy (脑癫痫) and seizures.
  - Disorders of the brain.
- Electroencephalogram (EEG).
  - EEG monitoring requires experts and efforts.
- Seizure detection for neonates.
  - Fragmentary
  - Less sensors than adults



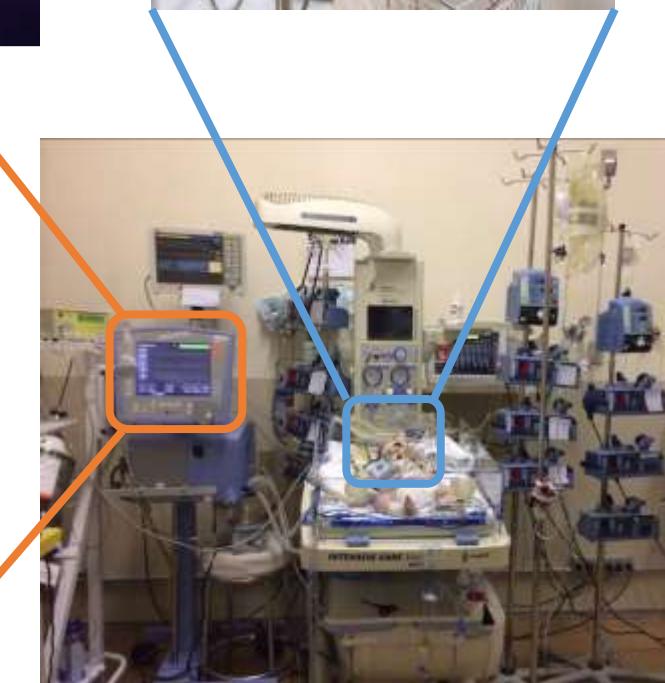
Seizure



Normal



Wearing the EEG device



# Brain signal based seizure detection

- Methodology
  - STATENet: Spatial-Temporal EEG Network
  - Mixture-of-expert ensemble learning
- Advantages
  - Superior performance (Table 1)
  - Transferability (Table 2)
  - (Aligned) anomaly localization (Fig. 3)

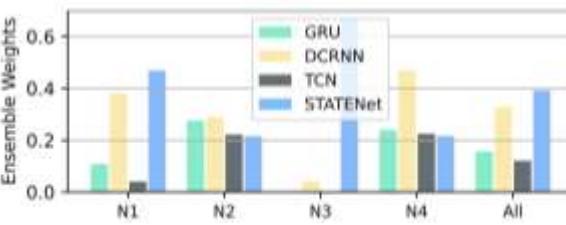
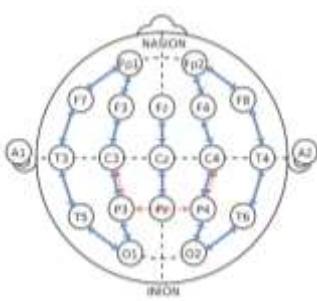


Fig. 4: Average ensemble weights assigned to models for four neonates (N1 to N4) and all neonates (All).

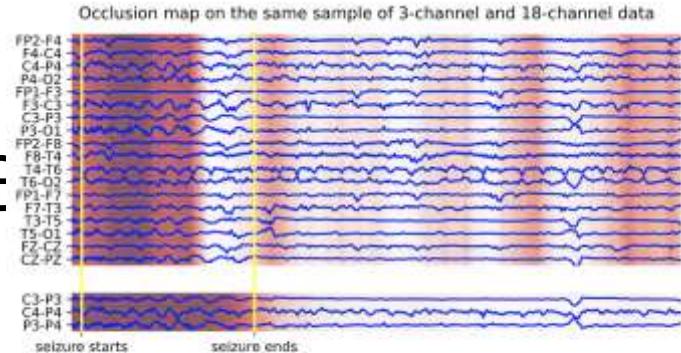


Fig. 3: The occlusion map of STATENet model on the same sample on 3-channel and 18-channel datasets. The deeper the color denotes more contribution to the positive prediction. The annotations of experts are shown with yellow lines.

Method	Metric	fold1		fold2		fold3		fold4		average
		3	18	3	18	3	18	3	18	
GBDT	AUROC	72.1	73.8	76.6	76.4	81.3	78.8	81.0	79.7	77.7 77.2
	AUPRC	37.4	36.2	45.5	46.2	56.5	54.5	57.9	47.4	49.4 46.1
ROCKET	AUROC	75.7	64.8	76.4	73.1	86.0	80.3	85.4	84.4	80.9 75.7
	AUPRC	49.2	37.7	43.1	32.4	60.5	59.4	59.9	50.4	53.2 45.0
GRU	AUROC	61.9	82.5	51.7	74.7	58.7	80.1	81.0	78.8	63.3 79.0
	AUPRC	30.2	59.5	13.4	51.9	27.4	65.2	55.0	45.4	31.5 55.5
TCN	AUROC	77.7	79.3	75.8	75.6	<b>90.2</b>	89.0	85.3	82.1	<u>82.3</u> 81.5
	AUPRC	54.1	54.5	52.9	54.1	76.2	76.1	63.3	53.1	61.6 59.5
MLSTM-FCN	AUROC	<u>78.3</u>	77.2	75.6	74.2	85.5	85.9	84.0	81.2	80.9 79.6
	AUPRC	51.2	48.2	48.7	49.0	61.4	71.1	61.4	51.6	55.7 55.0
InceptionTime	AUROC	72.6	74.0	75.2	73.1	72.6	85.6	70.9	81.2	72.8 78.5
	AUPRC	33.7	37.3	49.7	49.9	47.6	68.9	30.7	51.8	40.4 52.0
DCRNN	AUROC	73.3	79.5	<u>80.7</u>	<u>76.9</u>	84.5	87.0	85.3	81.9	81.0 81.3
	AUPRC	42.8	50.8	59.0	51.6	64.5	66.6	59.2	57.5	56.4 56.6
STATENet	AUROC	<b>87.1</b>	<b>91.5</b>	<u>85.2</u>	<u>87.7</u>	89.5	<u>93.4</u>	<u>88.2</u>	<u>91.2</u>	<u>87.5</u> <b>91.0</b>
	AUPRC	70.1	<b>78.0</b>	<u>71.5</u>	<b>74.8</b>	<b>78.8</b>	<u>85.3</u>	<u>74.1</u>	<u>77.4</u>	<u>73.6</u> <b>78.9</b>
MoE Ensemble	AUROC	89.4	92.7	88.6	87.3	91.1	96.1	88.9	91.3	89.5 91.8
	AUPRC	74.1	81.3	72.4	73.3	81.4	89.0	73.7	77.6	75.4 80.3

Table 1: The detailed experiment results. The higher metric value is better. The best and the second-placed results are formatted as bold font and underlined format, respectively.

Method	Metric	fold1		fold2		fold3		fold4		average
		18 to 3	3 to 18							
STATENet	AUROC	82.3	90.1	83.0	89.1	80.0	91.9	85.7	86.9	82.8 89.5
	AUPRC	58.3	75.5	66.3	75.7	68.4	76.8	71.4	65.4	66.1 73.4

Table 2: The results of transferred models. “3 to 18” means directly transferring the model trained on 3-channel data to 18-channel data, similarly for “18 to 3”.

# Learning Topology-Agnostic EEG Representations with Geometry-Aware Modeling

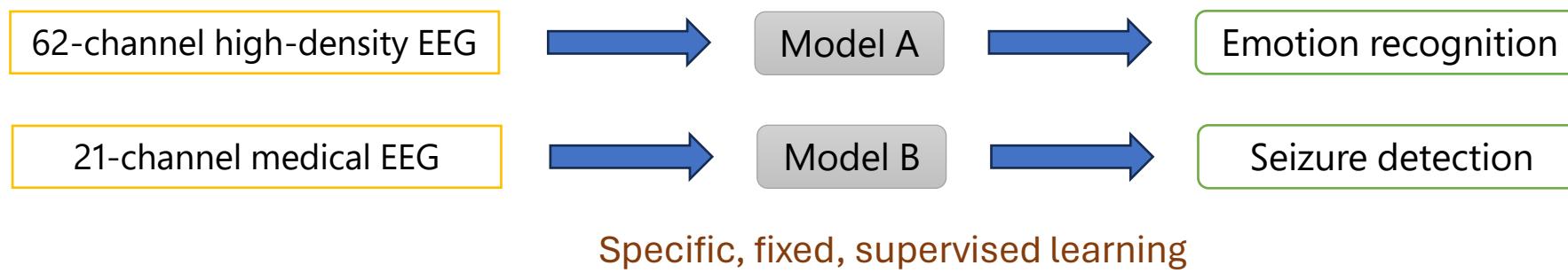
NeurIPS 2023

Ke Yi, Yansen Wang, Kan Ren, Dongsheng Li

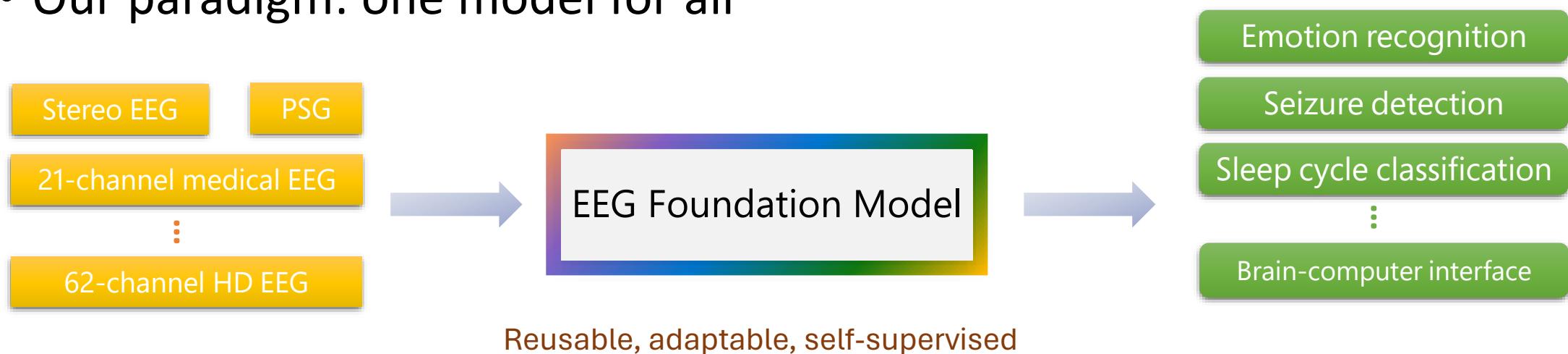


# Understanding Our Brain from EEG Signal with ML

- Existing paradigm: one model for one



- Our paradigm: one model for all



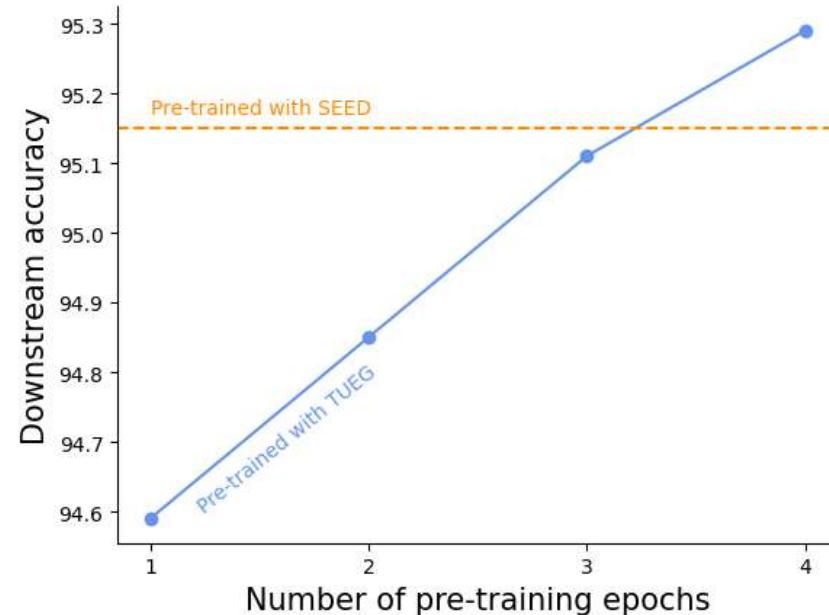
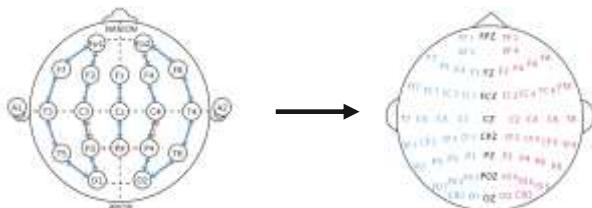
# EXPERIMENTS – EMOTION RECOGNITION

## Benchmark – SEED & SEED-IV

SEED: 15 subjects, 45 trials

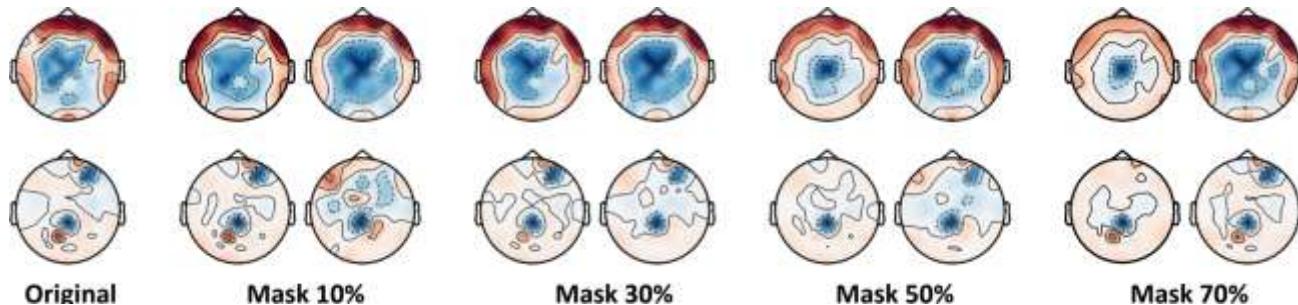
SEED-IV: 24 subjects, 72 trials

### Transfer from TUEG to SEED



### Reconstruction visualization

Proposed method brings  
robust understanding under high mask rate



# Towards Generating Real-World Time Series Data

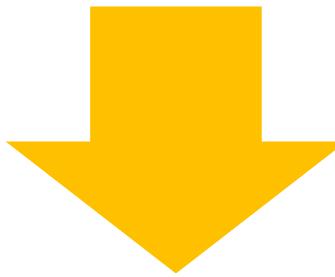
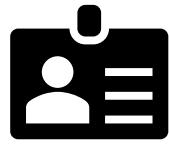
ICDM 2021

Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, Dongsheng Li

Microsoft Research

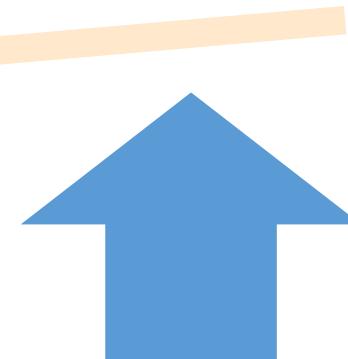


# Machine learning and data privacy

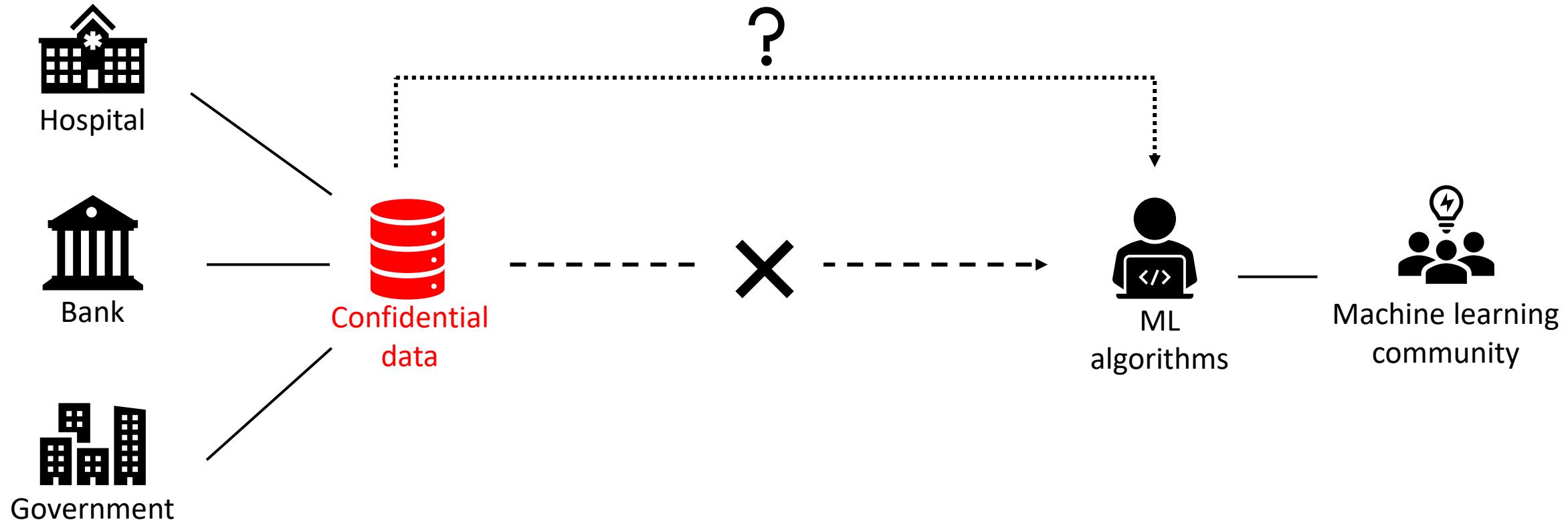


Machine learning  
model performance

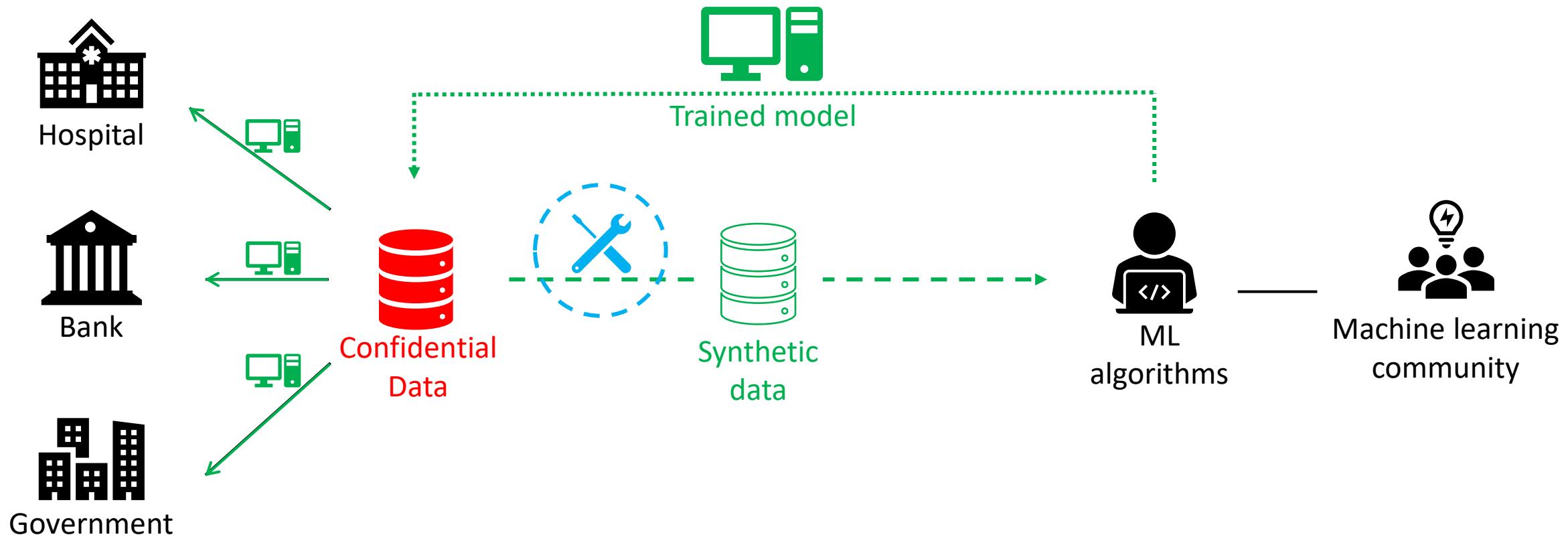
Data consumption  
and privacy issue



# How to leverage **data** legally and ethically?



# Overall framework



Which one is  
realistic/synthetic?

Group 1

	Hours	Heart Rate	Mean blood pressure	Oxygen saturation	Temperature	...
1A	0.181	112		94		...
	0.252		82			...
	0.416					synthetic
	0.554			94		...
	0.761	114	83	94	36.175	...
	1.188	113	81	95		...
	1.506					...
	1.911	113	79	95		...
	2.139		79			...
	...	...	...	...	...	...

	Hours	Heart Rate	Mean blood pressure	Oxygen saturation	Temperature	...
1B	0.084	96				...
	0.101					...
	0.118		88			...
	0.284	103	88		37.056	...
	0.401			89		...
	0.534	95	68	90		...
	0.784	83	117	96		...
	0.867					...
	1.034	85	70	100		...
	1.284	92	66	100		...
...						...

realistic

Which one is  
realistic/synthetic?

Group 2

2A

Hours	Heart Rate	Mean blood pressure	Oxygen saturation	Temperature	...
...	...	...	...	...	...
7.089	85	76	97	36.70211029	...
7.689	86	75	98		...
8.429	86	74	98		...
9.189	86	76	97		...
9.931	86	77	97	36.85286713	...
10.842	85	82	97		...
11.614	87	85	97		...
12.337	86	83	97		...
...	...	...	...	...	...

synthetic

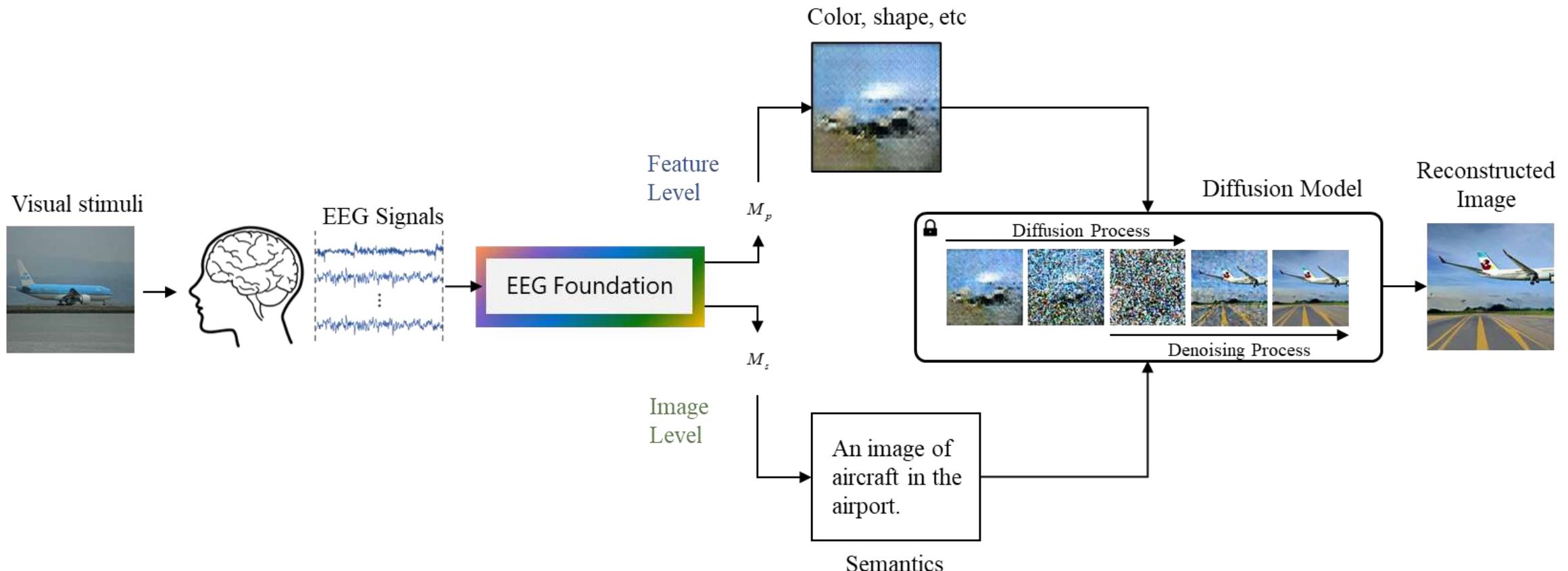
2B

Hours	Heart Rate	Mean blood pressure	Oxygen saturation	Temperature	...
...	...	...	...	...	...
2.639	89	100	107		...
3.089	91	99	118		...
3.639	96	99	129		...
4.639	99	98	115	38.333	...
5.639	97	99	99		...
6.639	94	99	96		...
7.639	94	98	90		...
8.473					...
8.639	91	97	88	38.333	...
...	...	...	...	...	...

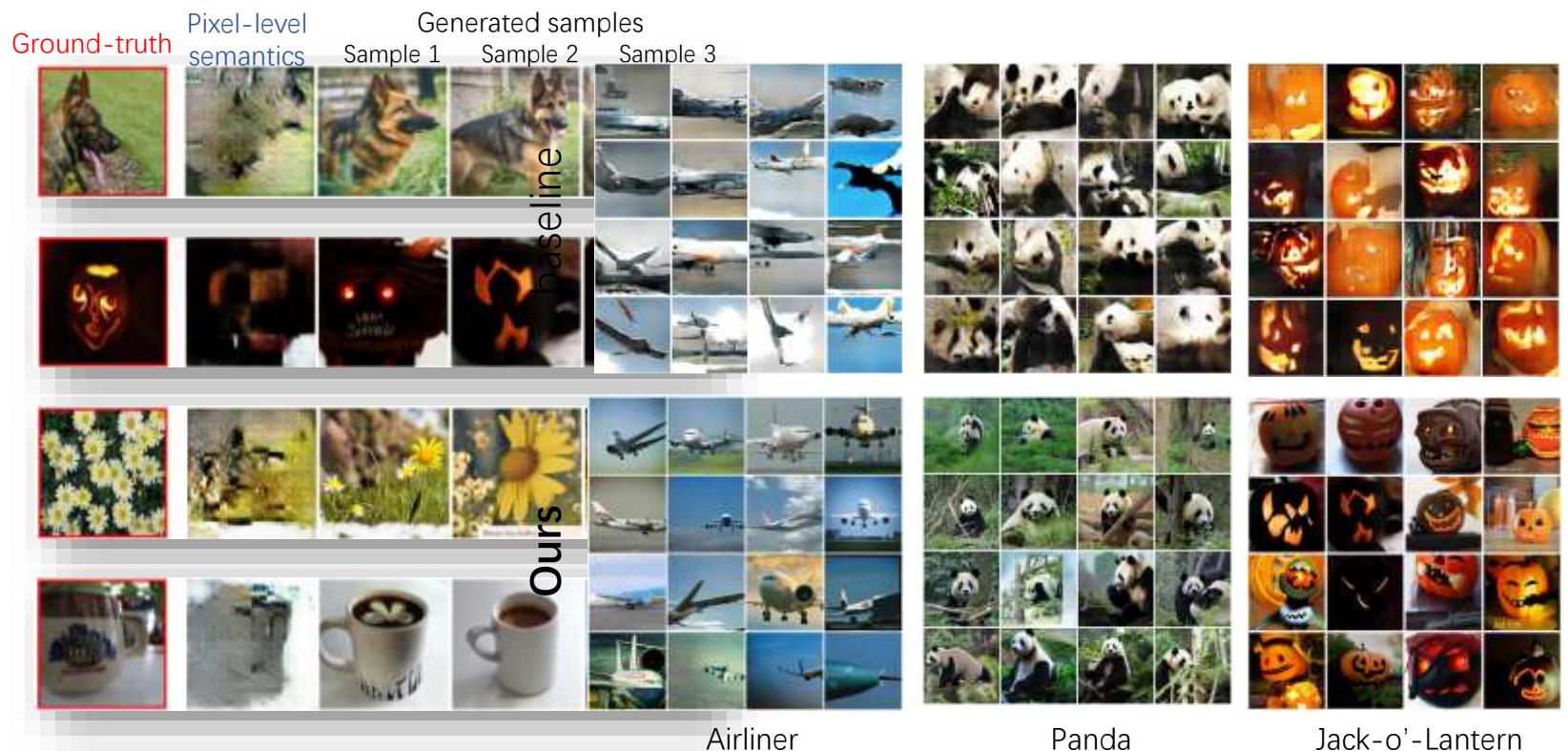
realistic

# NeuroImagen: A Method for Decoding Brain Signals

- Reconstruct human visual perception from EEG signals

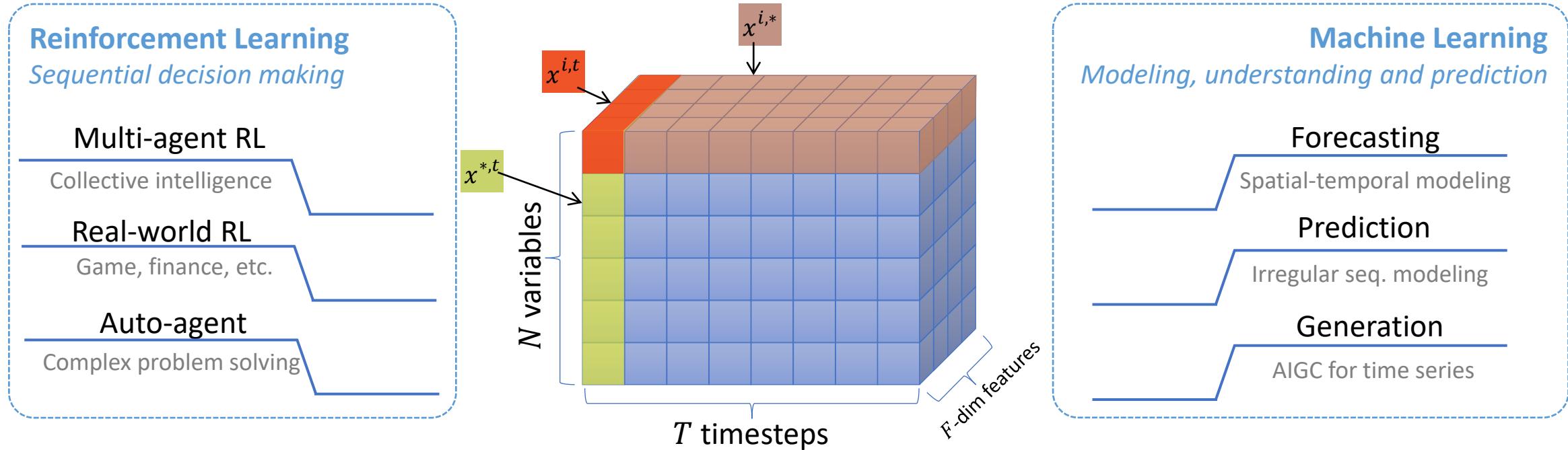


# Decoding what we see from EEG signals



# Machine intelligence on sequence data paradigm

- Process, understand, respond to temporally arriving sequence data with machine learning technology.



# Welcome to join our research group

- Machine learning for sequence data
  - Sequence model innovation
  - Time-series data mining
- Sequential decision making
  - Reinforcement learning algorithms
  - Large language model powered AI agents
- [renkan@shanghaitech.edu.cn](mailto:renkan@shanghaitech.edu.cn)
- meta\_learning (WeChat)

