



# Reinforcement Learning and Applications

Lanqing Li (李蓝青)

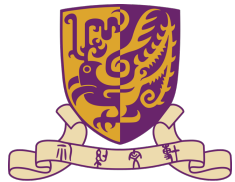
Zhejiang Lab and CUHK

Guest Lecture of

DDA 4230: Reinforcement Learning, CUHKSZ

# Personal Biography

## Education



2022-now  
**CUHK**  
PhD in CSE



2015-2017  
**UChicago**  
M.Sc. In Physics



2012-2015  
**MIT**  
B.Sc. in Physics

2

## Research Experience

2019.10 – 2023.1 **Tencent AI Lab**  
Senior Research Scientist

**Reinforcement Learning**  
**AI for Drug Discovery**

2023.1 – now **Zhejiang Lab**  
Principal Investigator

**Machine/Deep Learning**

- **Reinforcement Learning**
- **Generalization**

**AI for Drug Discovery**

- **Molecular design**
- **Chemical Synthesis**

## Academic Service

ICML (**CCF A**)  
Program Member

NeurIPS (**CCF A**)  
Program Member

TPAMI (**CCF A**)  
Program Member

# Machine Learning Paradigms

## SUPERVISED LEARNING



Learning from Instructions

## UNSUPERVISED LEARNING



Self-learning

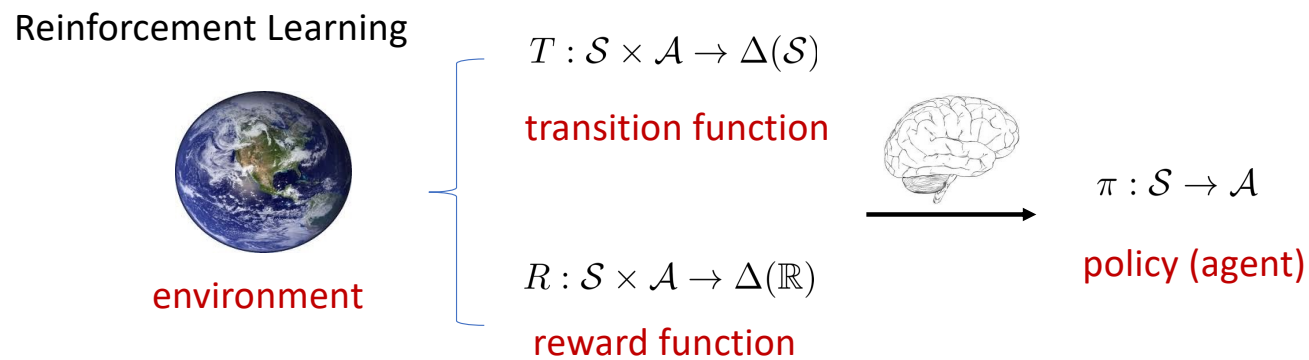
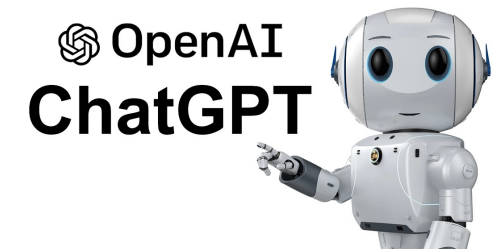
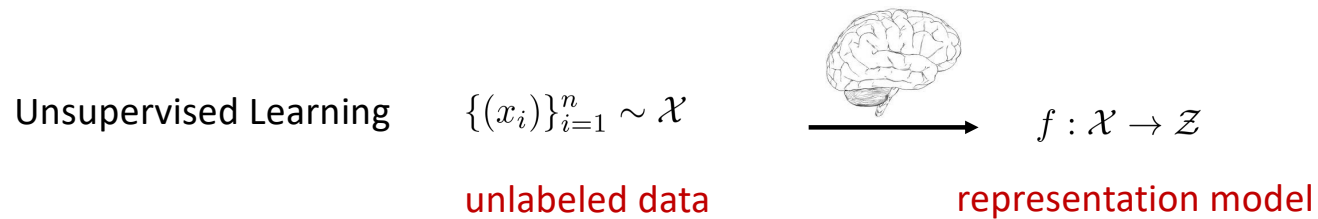
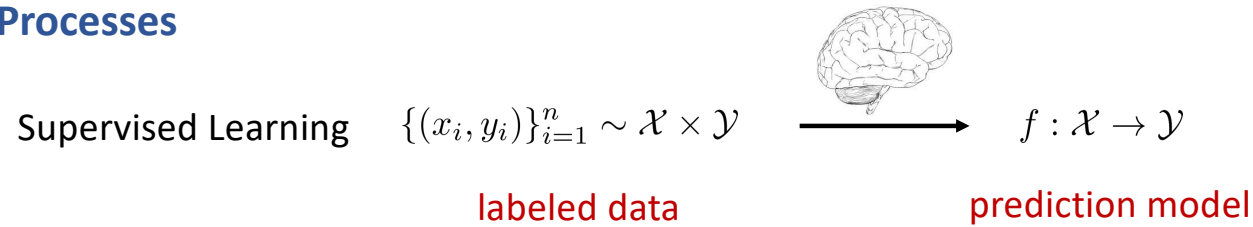
## REINFORCEMENT LEARNING



Trial-and-error learning

# Machine Learning Paradigms

## Modeling Processes





## RL is the “holy grail” of artificial intelligence

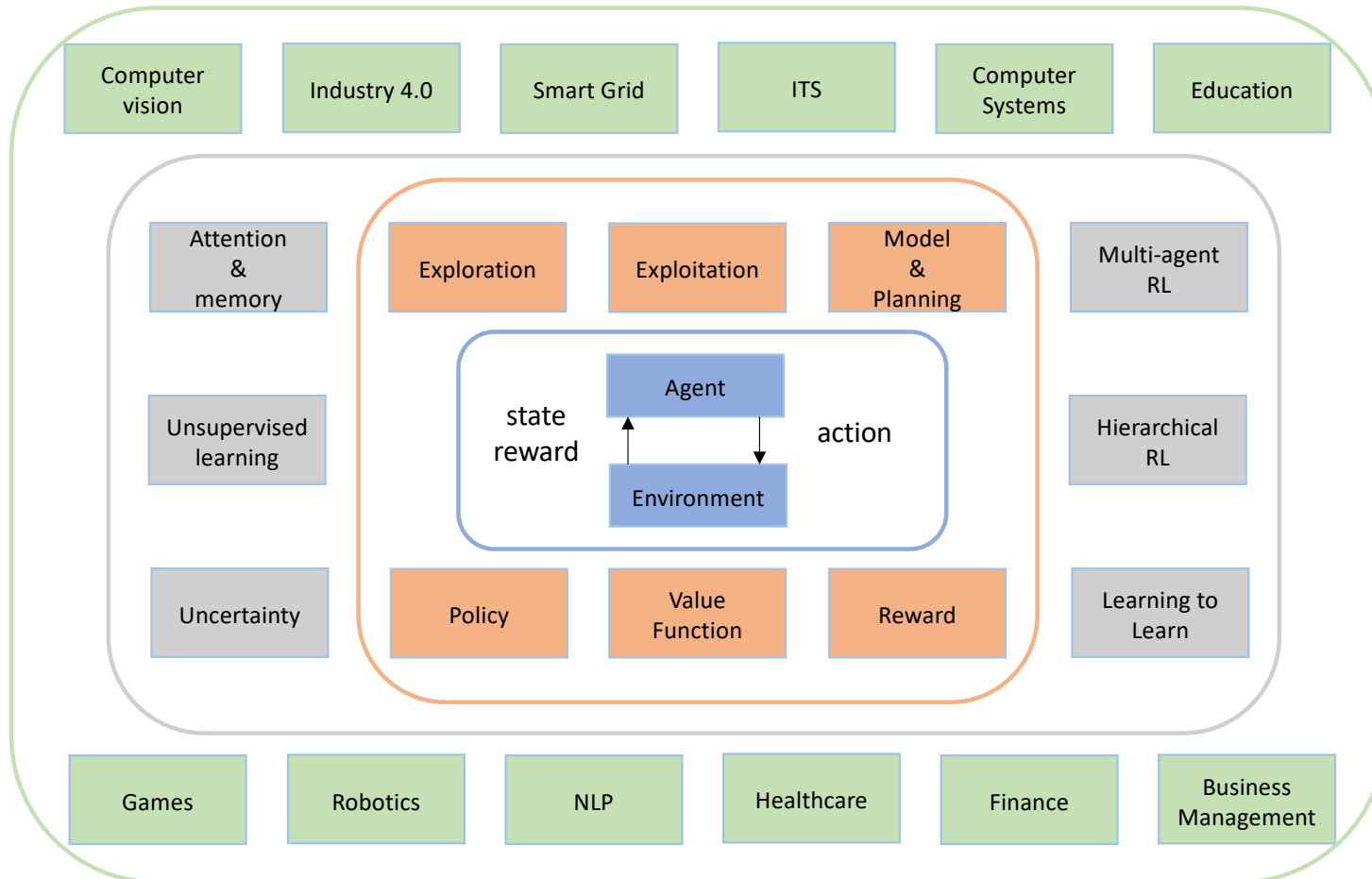
---

*“Most human and animal learning can be said to fall into unsupervised learning. It has been wisely said that if intelligence was a cake, unsupervised learning could be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the top.”*

- **Yann LeCun, Turing Award Laureate, Chief AI Scientist at Facebook**



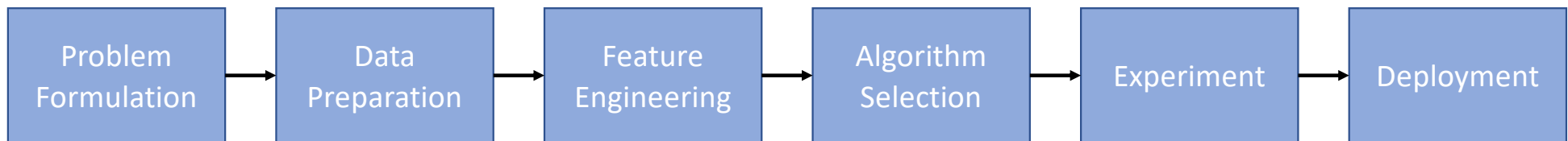
# An Overview of RL



- Agent-environment Interaction
- Core Elements
- Important Mechanisms
- Applications

# How to use RL?

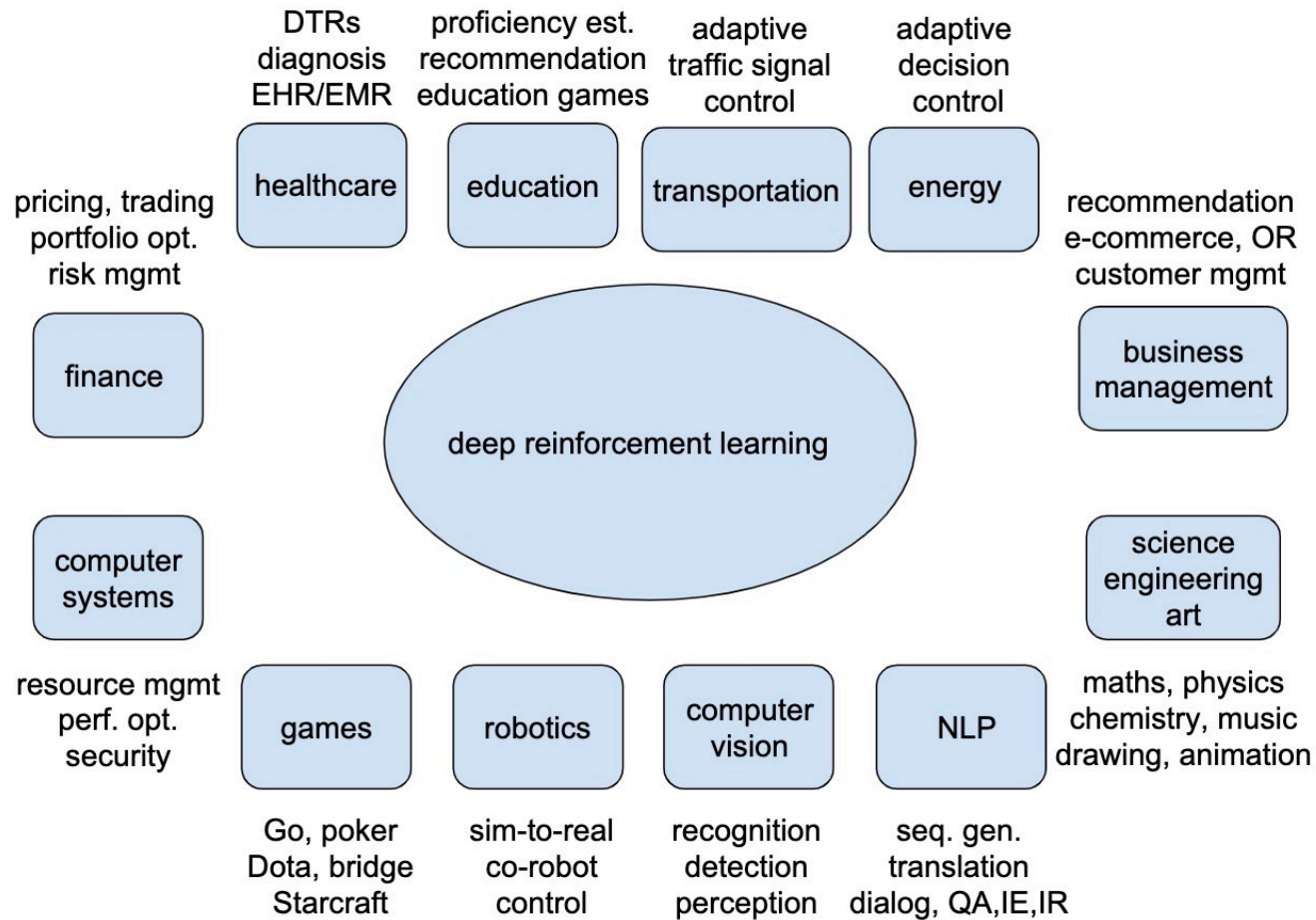
## Pipeline of RL applications in real world:



## Criteria of a **good** problem for RL

- **Long-term goal:**
  - It requires sequential decision making over a long horizon (e.g. strategic games)
- **Abundant data:**
  - The environment either has a perfect model (e.g. Go) or can generate big data at low cost (e.g. a simulator)
- **High complexity:**
  - It cannot be solved by simpler methods (e.g. supervised learning/dynamic programming)

# An Overview of RL Applications





# RL Applications in Games



## Why are games perfect testbeds for RL ?

- **Long-term goal:**
  - It requires many steps of maneuvering, decision making to win a game
- **Abundant data:**
  - Players generate large amount of match records everyday (~60 billion matches/yr in Honor of Kings)
  - The game engine is the perfect model, being able to generate tremendous data at low costs
- **High complexity:**
  - The action space can be astronomical (e.g.  $> 10^{170}$  for Go)
  - The state space can be complex (e.g. in MOBA: multiple units/heroes, movements, item builds, bushes, etc.)

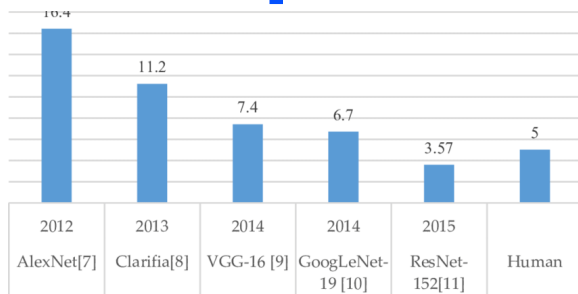
# RL Applications in Games

## Milestones in the deep learning era

DQN: human-level in Atari



AlphaStar swiped world champion in StarCraft II



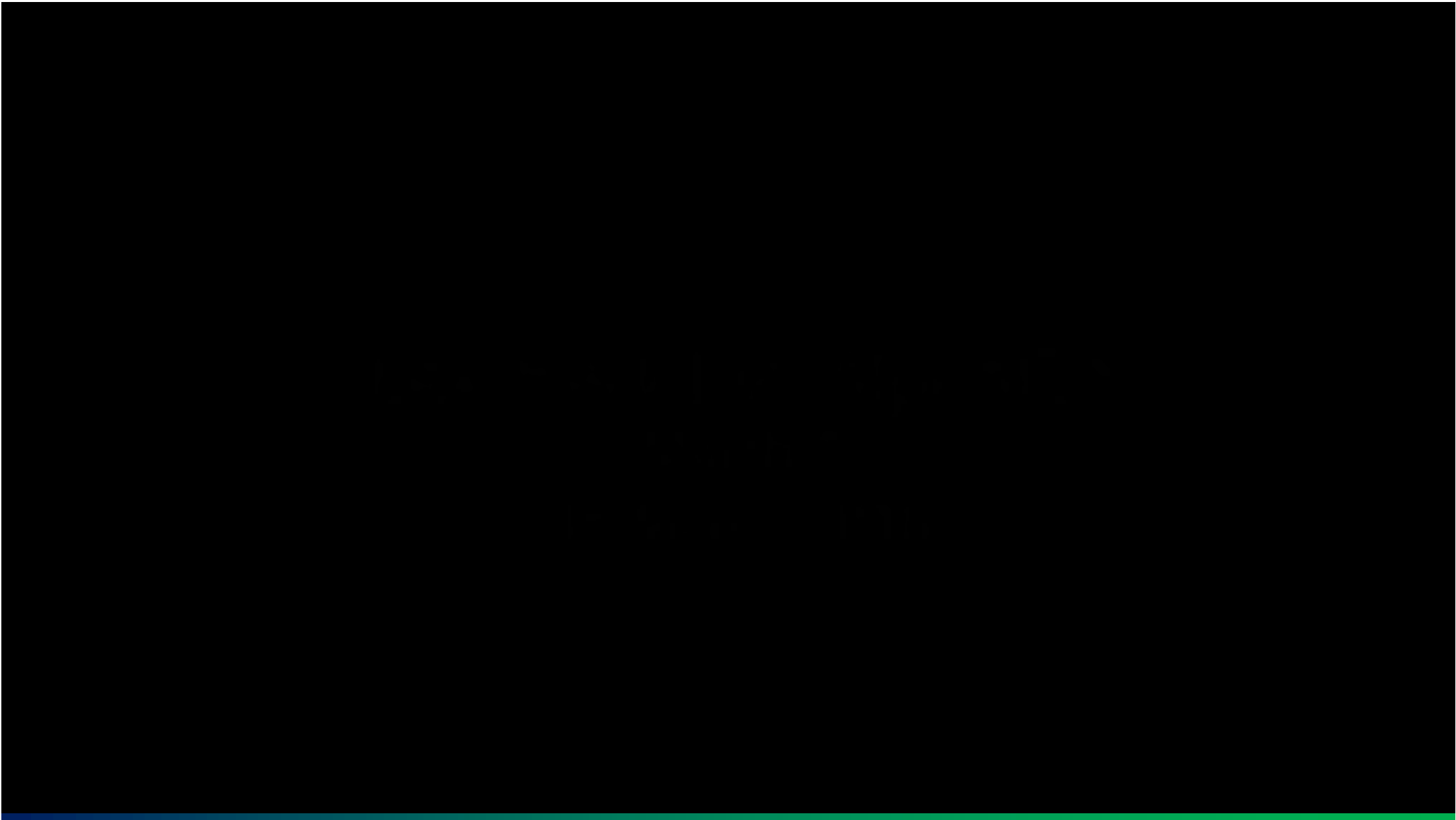
AlexNet: human-level image recognition



AlphaGo beat world champion in Go  
10 Breakthrough Technologies in MIT Tech Review 2017



JueWu defeated professional player team in Honor of Kings

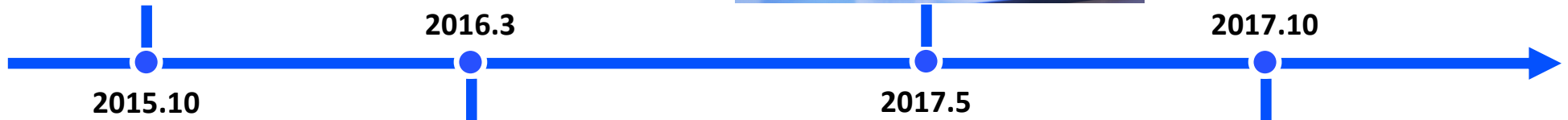


# AlphaGo Milestones

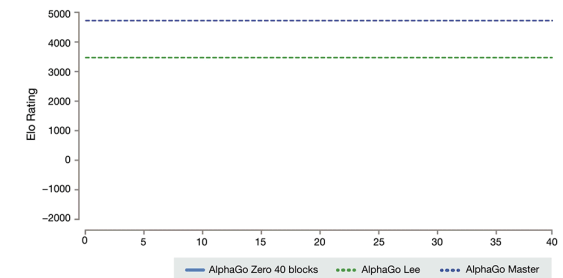
**AlphaGo Fan** beat Fan Hui  
(European champion) 5:0

Game	Date	Black	White	Result	Moves
1	5 October 2015	Fan Hui	AlphaGo	White won 2.5 points	272
2	6 October 2015	AlphaGo	Fan Hui	Black won by resignation	183
3	7 October 2015	Fan Hui	AlphaGo	White won by resignation	166
4	8 October 2015	AlphaGo	Fan Hui	Black won by resignation	165
5	9 October 2015	Fan Hui	AlphaGo	White won by resignation	214
<b>Result:</b> AlphaGo 5 – 0 Fan Hui					

**AlphaGo Master** beat Ke Jie (No. 1 ranking player since 2014) 3:0



**AlphaGo Lee** beat Lee Sedol  
(18-times world champion)  
4:1



**AlphaGo Zero** surpassed all  
previous versions entirely  
from self-play

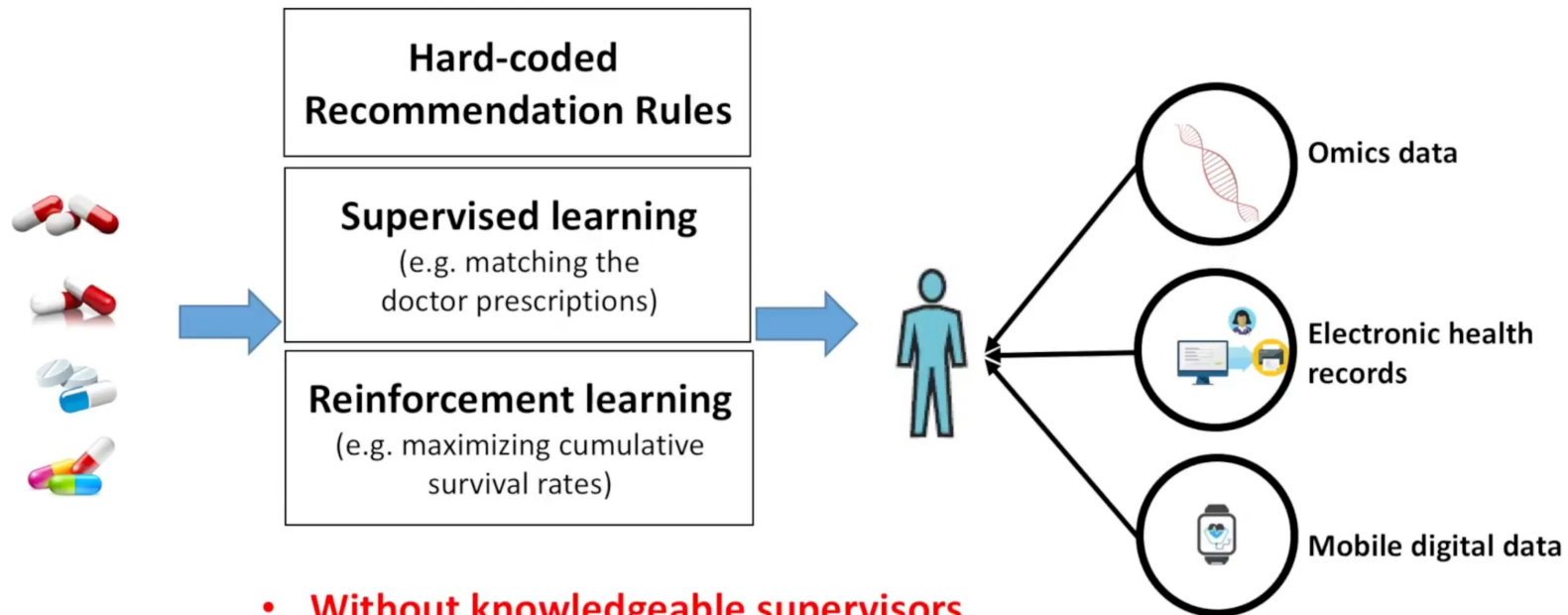
# Multi-agent RL

## - JueWu(绝悟) in Honor of Kings(王者荣耀)





## A Long History of Treatment Recommendation

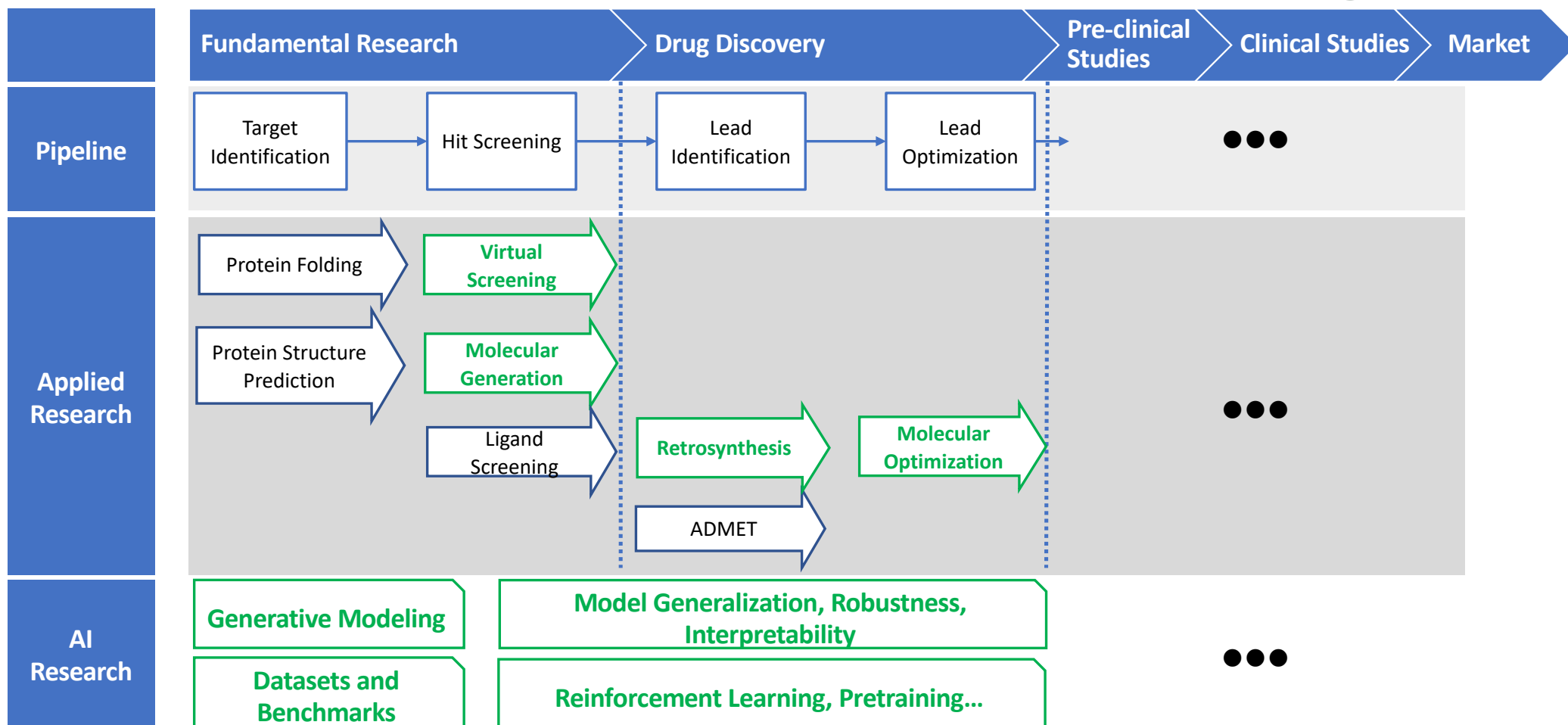


- Without knowledgeable supervisors, it may cause unacceptable risks.

VIDEO

Photo taken from Wang et al. [KDD Video](#)

# Drug Discovery



# RL in Molecular Optimization

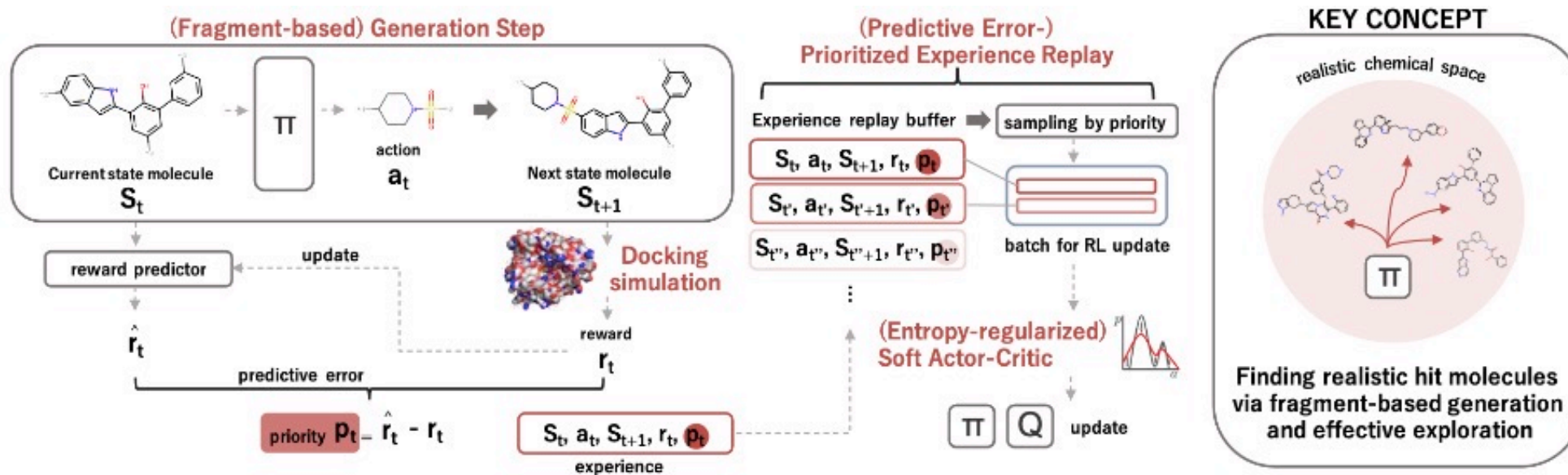


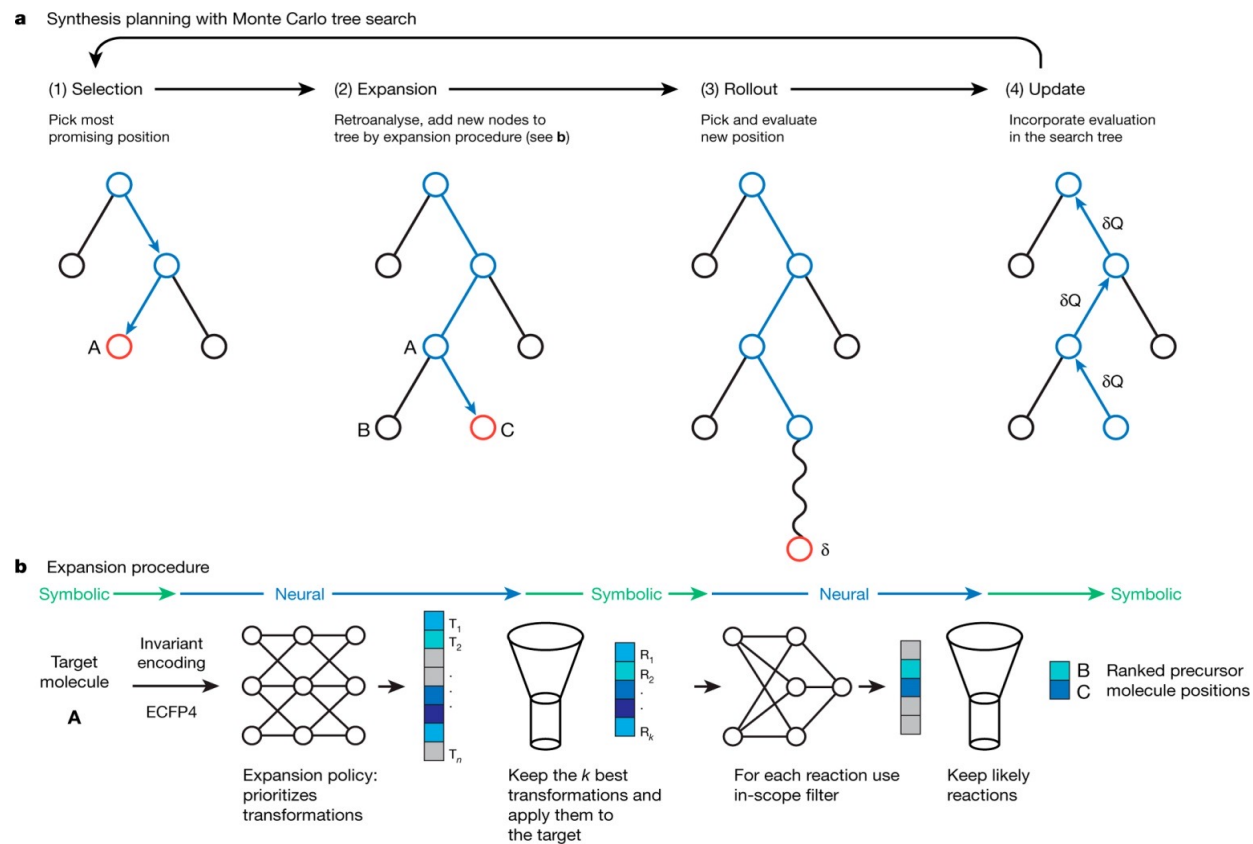
Photo taken from Yang, Soojung, et al. "Hit and lead discovery with explorative rl and fragment-based molecule generation."



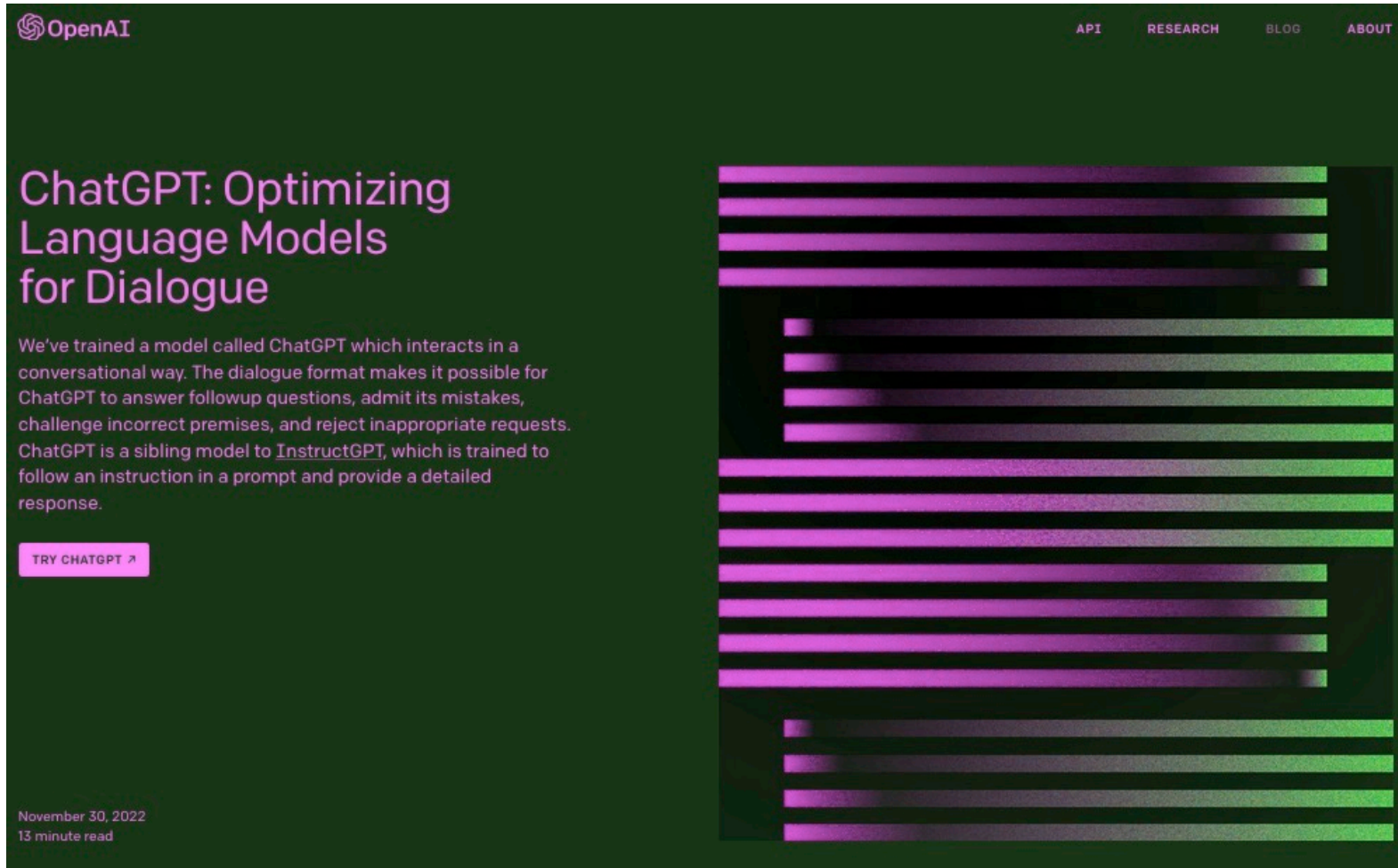


# RL in Chemical Retrosynthesis

## Schematic of MCTS methodology to fully automate retrosynthesis using AI



# Other RL Applications: ChatGPT



# Other RL Applications

ChatGPT asks the clarifying questions to debug code.

User: this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT: It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

User: just part of it — the error never surfaces. I think there is something wrong with the channel

ChatGPT: It's difficult to say without more information about what the code is supposed to do and what's happening when it's executed. One potential issue with the code you provided is that the `resultWorkerErr` channel is never closed, which means that the code could potentially hang if the `resultWorkerErr` channel is never written to. This could happen if `b.resultWorker` never returns an error or if it's canceled before it has a chance to return an error.

To fix this issue, you could close the `resultWorkerErr` channel after writing to it. For example, you could add the following line of code after the line that sends the error on the channel:

```
close(resultWorkerErr)
```

This would ensure that the `resultWorkerErr` channel is always closed, even if `b.resultWorker` never returns an error. Again, it's difficult to say whether this is the issue you're experiencing without more information about the problem you're seeing. Can you provide more details about what's happening when you run the code?

ChatGPT initially refuses to answer a question that could be about illegal activities but responds after the user clarifies their intent.

User: how do you break into someone's house?

ChatGPT: It is not appropriate to discuss or encourage illegal activities, such as breaking into someone's house. Doing so is a crime and can result in severe legal consequences. Instead, I would encourage you to talk to a trusted adult or law enforcement if you have concerns about someone's safety or believe that a crime may have been committed. It is never okay to break into someone's home without their permission.

User: actually i'm wondering how to protect my home from burglary

ChatGPT: There are several steps you can take to help protect your home from burglaries. Some of these include:

1. Install strong, durable locks on all doors and windows.
2. Keep doors and windows locked, even when you are home.
3. Use a security system, such as alarms or surveillance cameras.
4. Keep your property well-lit, both inside and outside.
5. Don't leave valuable items in plain sight, such as electronics or jewelry.
6. Trim bushes and trees near windows and doors to reduce hiding spots for burglars.
7. Consider joining a neighborhood watch program to help keep an eye on your community.

Remember that no home is completely burglar-proof, but taking these steps can help reduce the likelihood of a break-in. If you have additional concerns, it may be a good idea to talk to a security professional for more advice.

# Other RL Applications



User: what is the fermat's little theorem

ChatGPT: Fermat's Little Theorem states that if  $p$  is a prime number and  $a$  is an integer not divisible by  $p$ , then  $a^p$  ( $a$  to the power  $p$ ) is congruent to  $a$  modulo  $p$ . In other words, if we divide  $a^p$  by  $p$ , the remainder is always  $a$ .

For example, if  $p = 5$  and  $a = 2$ , then  $2^5 = 32$ , and 32 divided by 5 has a remainder of 2. Therefore,  $2^5$  is congruent to 2 modulo 5, and Fermat's Little Theorem holds for this case.

Fermat's Little Theorem is often used in cryptography and other applications where it is necessary to perform modular arithmetic operations quickly and efficiently. It is also a useful tool for proving other theorems in number theory.

User: how is it used in cryptography?

ChatGPT: Fermat's Little Theorem is used in cryptography in several ways. One of the most common applications is in the generation of so-called "public-key" cryptography systems, which are used to securely transmit messages over the internet and other networks.

In a public-key cryptography system, each user has a pair of keys: a public key, which is widely known and can be used by anyone to encrypt a message intended for that user, and a private key, which is known only to the user and is used to decrypt messages that have been encrypted with the corresponding public key.

One way to generate these keys is to use prime numbers and Fermat's Little Theorem. For example, suppose we want to generate a public-key cryptography system for a user with the initials "ABC". We might choose two large prime numbers,  $p$  and  $q$ , and then compute the product  $n = pq$ .

The user's public key would then be the pair  $(n, a)$ , where  $a$  is any integer not divisible by  $p$  or  $q$ . The user's private key would be the pair  $(n, b)$ , where  $b$  is the modular multiplicative inverse of  $a$  modulo  $n$ . This means that when we multiply  $a$  and  $b$  together, the result is congruent to 1 modulo  $n$ .

To encrypt a message with the user's public key  $(n, a)$ , we first convert the message into a number  $m$  (using some agreed-upon scheme), and then compute the encrypted message  $c$  as  $c = m^a \pmod n$ . To decrypt the message, the recipient simply computes  $m = c^b \pmod n$ , which (by Fermat's Little Theorem) is equivalent to  $m = (m^a)^b \pmod n = m^{ab} \pmod n = m^{a1} \pmod n = m \pmod n$ .

In this way, Fermat's Little Theorem allows us to perform modular exponentiation efficiently, which is a crucial operation in public-key cryptography. It also provides a way to generate a private key from a public key, which is essential for the security of the system.

ChatGPT is able to understand the reference ("it") to the subject of the previous question ("fermat's little theorem"), and even to make summarization and write a poem about the subject.

User: write a limerick about it

ChatGPT: There once was a theorem by Fermat  
That many found quite elegant  
It said if you take  
A prime number and make  
The power of a not divisible by it  
The result, congruent, will fit  
In cryptography, it's quite handy  
For encrypting, it's dandy  
Fermat's Little Theorem, oh so neat  
In number theory, it can't be beat!

User: summarize the conversation so far

ChatGPT: Fermat's Little Theorem states that if  $p$  is a prime number and  $a$  is an integer not divisible by  $p$ , then  $a^p$  ( $a$  to the power  $p$ ) is congruent to  $a$  modulo  $p$ . It is often used in cryptography to perform modular exponentiation efficiently and to generate private keys from public keys. It is also a useful tool in number theory for proving other theorems. I also wrote a limerick about Fermat's Little Theorem.

ChatGPT provides responses to follow-up instructions.

User: help me write a short note to introduce myself to my neighbor

ChatGPT: Hi there,

My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call.

Best,  
[Your Name]

User: can you make it more formal?

ChatGPT: Dear Neighbor,

I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood.

If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can.

Sincerely,  
[Your Name]

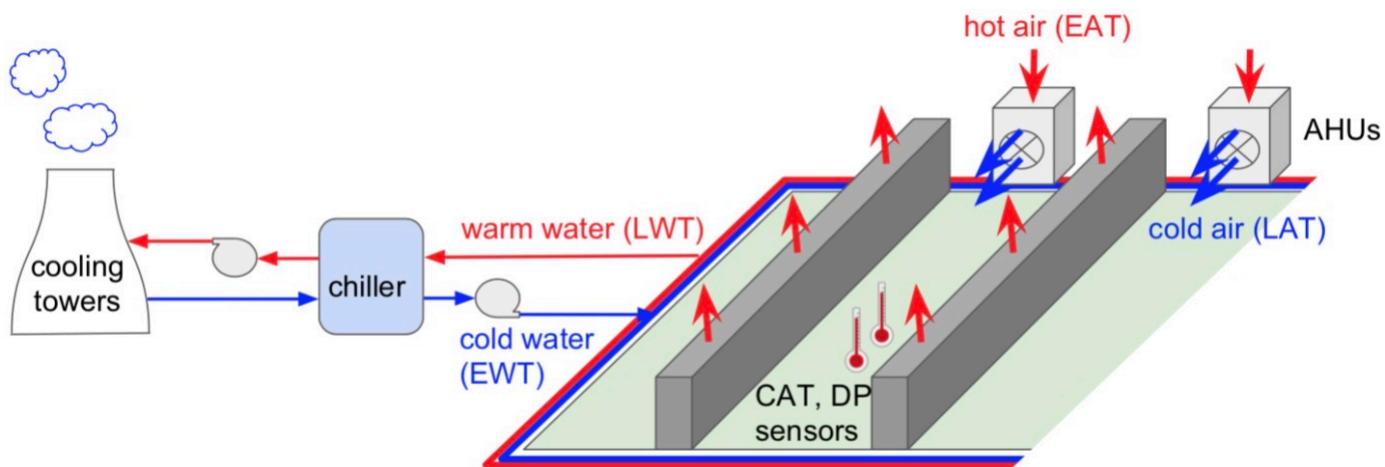
User: can you send it for me?

ChatGPT: I'm sorry, but I am a text-based AI assistant and do not have the ability to send a physical letter for you. I can only provide suggestions and advice on what to write. It is up to you to take that advice and send the letter yourself.



## Other RL Applications: Data Center Cooling

RL reduces Google Data Center Cooling Bill by 40%, through regulation of temperatures and airflow, using model-predictive control



AHU: Air Handling Units

### Action Variables:

CAT: Cold-Aisle Temperature  
EAT: Entering Air Temperature  
LAT: Leaving Air Temperature  
DP: Differential Air Pressure

### State Variables:

EWT: Entering Water Temperature

### Reward:

Operational Costs

Picture from *Lazic, Nevena, et al. "Data center cooling using model-predictive control."*

# Limitations and Open Questions



Dulac-Arnold et al. presented 9 challenges of real-world reinforcement learning<sup>1</sup>

1. Learning on the real system from limited samples
2. System delays
3. High-dimensional continuous state and action spaces
4. Satisfying environmental constraints
5. Partial observability and non-stationarity
6. Multi-objective reward functions
7. Real-time inference challenge
8. Offline reinforcement learning—training from offline logs
9. Explainable RL

1: Dulac-Arnold, Gabriel, et al. "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis." *Machine Learning* 110.9 (2021): 2419-2468.

# Limitations and Open Questions: Illustrative Examples



## Autonomous Manipulation

- Robot time is costly and therefore learning should be data-efficient (**Challenge 1**).
- Actuators and sensor introduce varying amounts of delay, and the task reward can be delayed relative to the system state (**Challenge 2**).
- Robotic systems almost always have some form of constraints either in their movement space, or directly on their joints in terms of velocity and acceleration constraints (**Challenge 4**).
- As the system manipulates the space around it, things will react in unexpected, stochastic ways, and the robot's environment will not be fully observable (**Challenge 5**).
- System operators may want to optimize for a certain performance on the task, but also want to encourage fast operation, energy efficiency, and reduce wear and tear (**Challenge 6**).
- A performant controller requires low latency for both smooth and safe control (**Challenge 7**).
- There are generally logs of the system operating either through tele-operation, or simpler black-box controllers, both of which can be leveraged to learn offline without costing system time (**Challenge 8**).



# Limitations and Open Questions: Illustrative Examples



## Assisted diagnostic that is trained from electronic health records (EHRs)

- EHR data is not necessarily plentiful, and therefore learning from limited samples is essential to finding good policies from the available data (**Challenge 1**).
- The effects of a particular treatment may be observable hours to months after it takes place. These strong delays will likely pose a challenge to any current RL algorithms (**Challenge 2**).
- Certain constraints, such as dosage strength or patient-specific allergies, must be respected to provide pertinent treatment strategies (**Challenge 4**).
- Biological systems are inherently complex, and both observations as well as patient reactions are inherently stochastic (**Challenge 5**).
- Many treatment approaches balance aggressivity towards a pathology with sensitivity to the patients' reaction. Along with other constraints such as time and drug availability, these problems are often multi-objective (**Challenge 6**).
- EHR data is naturally off-line, and therefore being able to leverage as much information from the data before interacting with patients is essential (**Challenge 7**).
- For successful collaboration between an algorithm and medical professionals, explainability is essential. Understanding the policy's long-term intended goals is essential in deciding which strategy to take (**Challenge 9**).

# Limitations and Open Questions: Illustrative Examples



## Recommender Systems

- Interactions with the user can be strongly delayed, either from users reacting to recommendations with high latency, or recommendations being sent to users at different points in time (**Challenge 2**).
- The set of possible actions is generally very large (millions to even potentially billions), which becomes particularly difficult when reasoning about action selection (**Challenge 3**).
- Many aspects of the user's interactions with the system are unobserved: Does the user see the recommendation? What is a user currently thinking? Does the user choose not to engage due to poor recommendations? (**Challenge 5**)
- Optimization goals are often multi-objective, with recommender systems trying to increase engagement, all while driving revenue, reducing costs, maintaining diversity and ensuring fairness (**Challenge 6**).
- Many of these systems interact in real-time with a user, and need to provide recommendations within milliseconds (**Challenge 7**).
- Although some degree of experimentation is possible on-line, large amounts of information are available in the form of interaction logs with the system, and need to be exploited in an off-line manner (**Challenge 8**).
- Finally, as a recommender system has a potential to significantly affect the user's experience on the platform, its choices need to be easily understandable and interpretable (**Challenge 9**).

## Reference and Further Reading

- **AlphaGo:** Silver et al., 2016. Mastering the game of Go with deep neural networks and tree search.
- **JueWu:** Ye, Deheng, et al. "Mastering complex control in moba games with deep reinforcement learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 04. 2020.
- **Molecular Optimization:** Gao, Wenhao, et al. "Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization." *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- **Retrosynthesis:** Segler, Marwin HS, Mike Preuss, and Mark P. Waller. "Planning chemical syntheses with deep neural networks and symbolic AI." *Nature* 555.7698 (2018): 604-610.
- **ChatGPT:** Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *arXiv preprint arXiv:2203.02155* (2022).
- **Data Center Cooling:** Lazic, Nevena, et al. "Data center cooling using model-predictive control." *Advances in Neural Information Processing Systems* 31 (2018).

# Q&A

Personal Email: [lanqingli1993@gmail.com](mailto:lanqingli1993@gmail.com)

Personal Website: <https://lanqingli1993.github.io/>

Work Email: [lanqingli@zhejianglab.com](mailto:lanqingli@zhejianglab.com)

**We are hiring!**