Reinforcement Learning China Summer School

RLChina 2020

Imitation Learning

Yang Yu

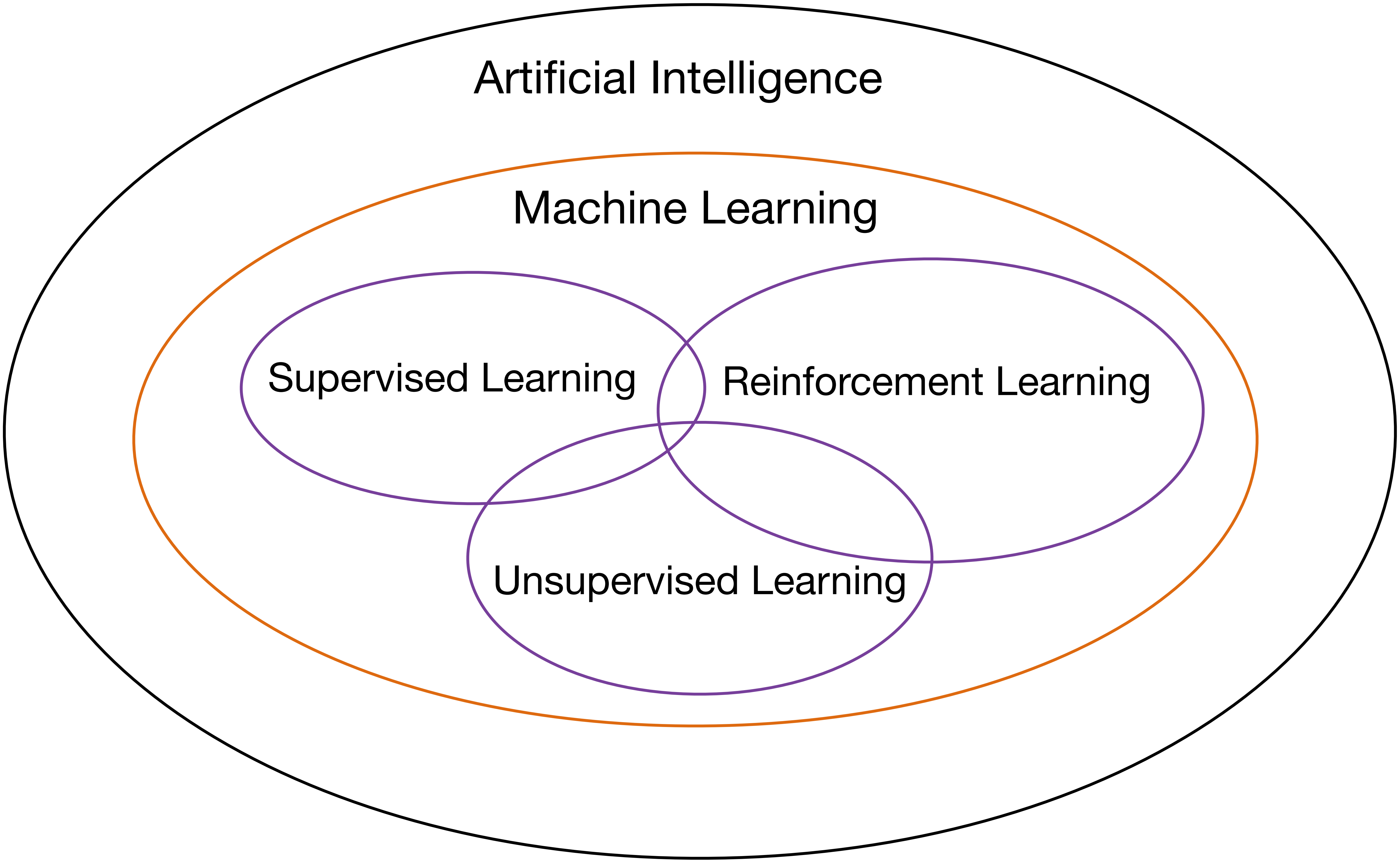
Nanjing Univeristy

Aug. 1, 2020

Previously

- Value-based Reinforcement Learning
- Policy-based RL and RL Theory
- Optimisation in Learning
- Model-based Reinforcement Learning
- Control as Inference

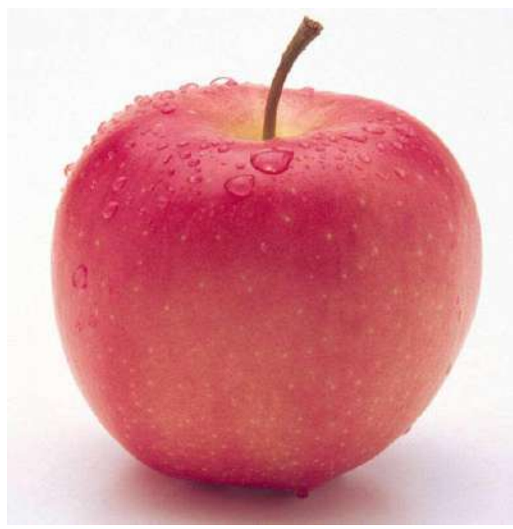
Position of RL in AI



Supervised learning

Instance

Label



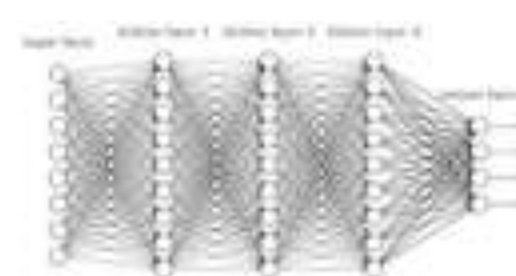
apple

x

y

Learn a model to fit the data $f(x) = y$

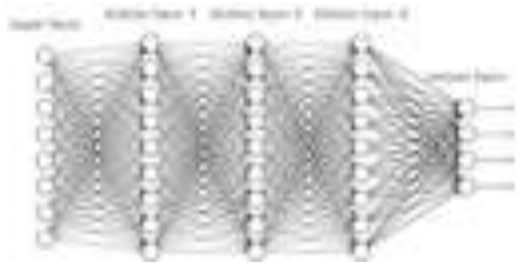
function model



Supervised learning

Learn a model to fit the data $f(x) = y$

function model



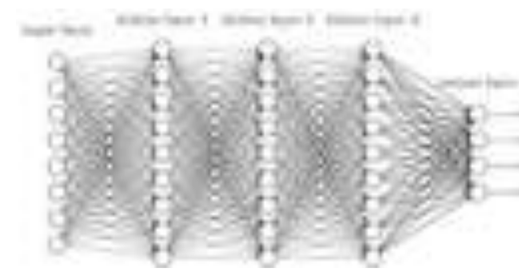
Find the model parameters:

$$\theta^* = \arg\min_{\theta} \sum_i \|f(x_i|\theta) - y_i\| + \|\theta\|$$

Supervised learning

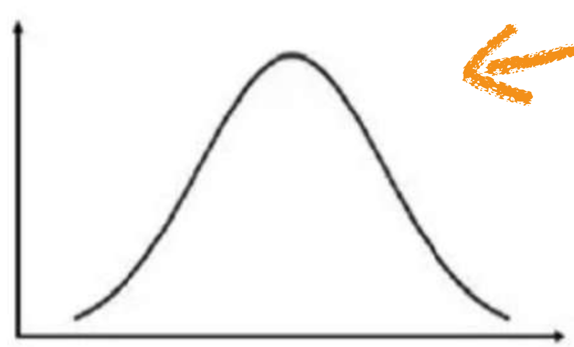
Learn a model to fit the data $f(x) = y$

function model

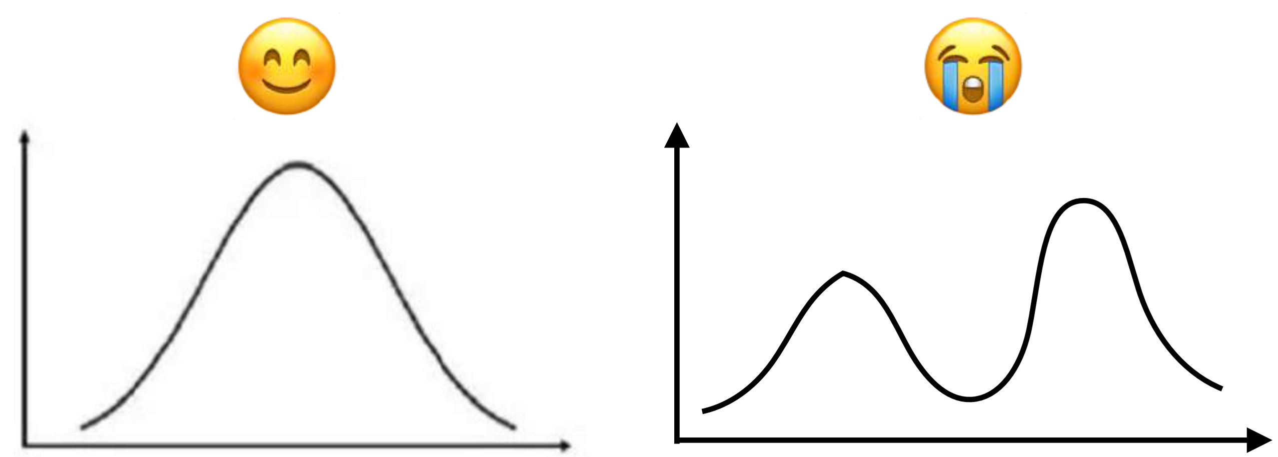


Find the model parameters:

$$\theta^* = \arg\min_\theta \boxed{\sum_i} \|f(x_i|\theta) - y_i\| + \|\theta\|$$



training data distribution



test data distribution

Unsupervised learning

a data set

without feedback information (label)

General task:

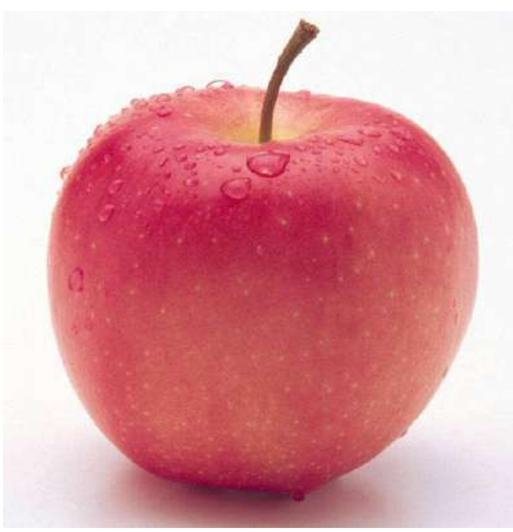
discover the structure information in the data

clustering, density discovery, feature representation …

Unsupervised learning

self-supervised learning

instance



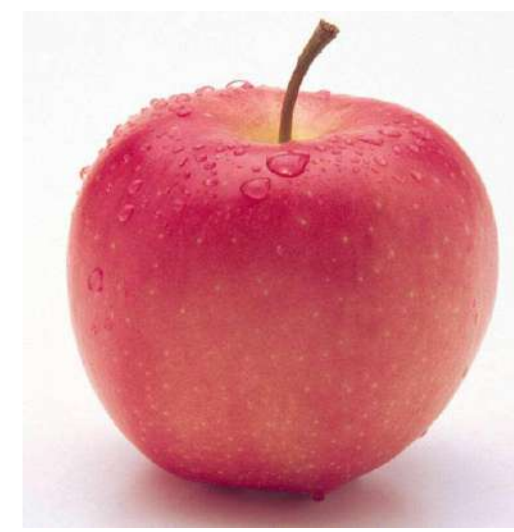
x

z

$f(x) = z$

encoder

instance

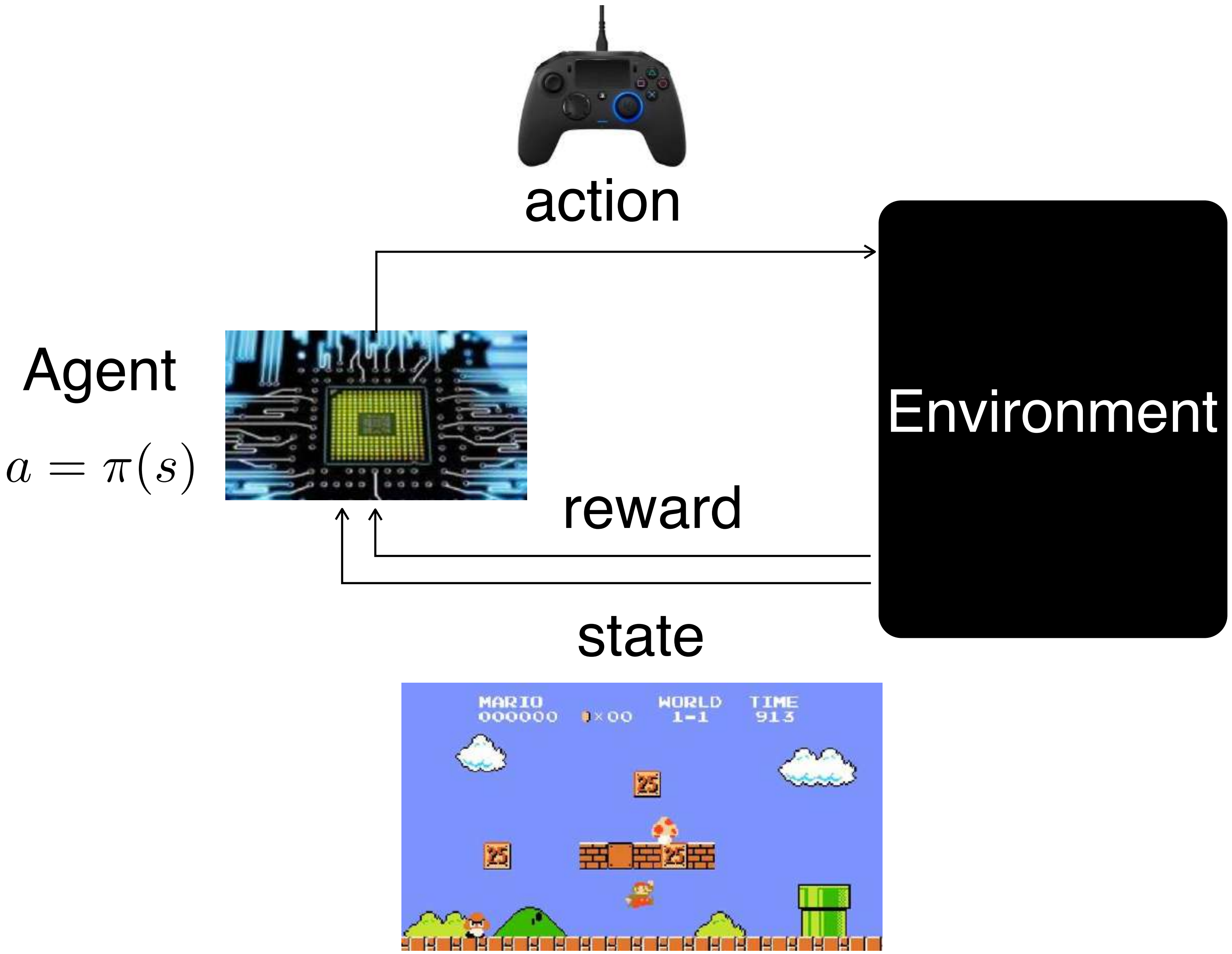


x

$g(z) = x$

decoder
generator

Reinforcement learning



Target: $\pi^* = \arg\max(r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots)$

RL from scratch is slow

- huge search space
- sampling-based exploration

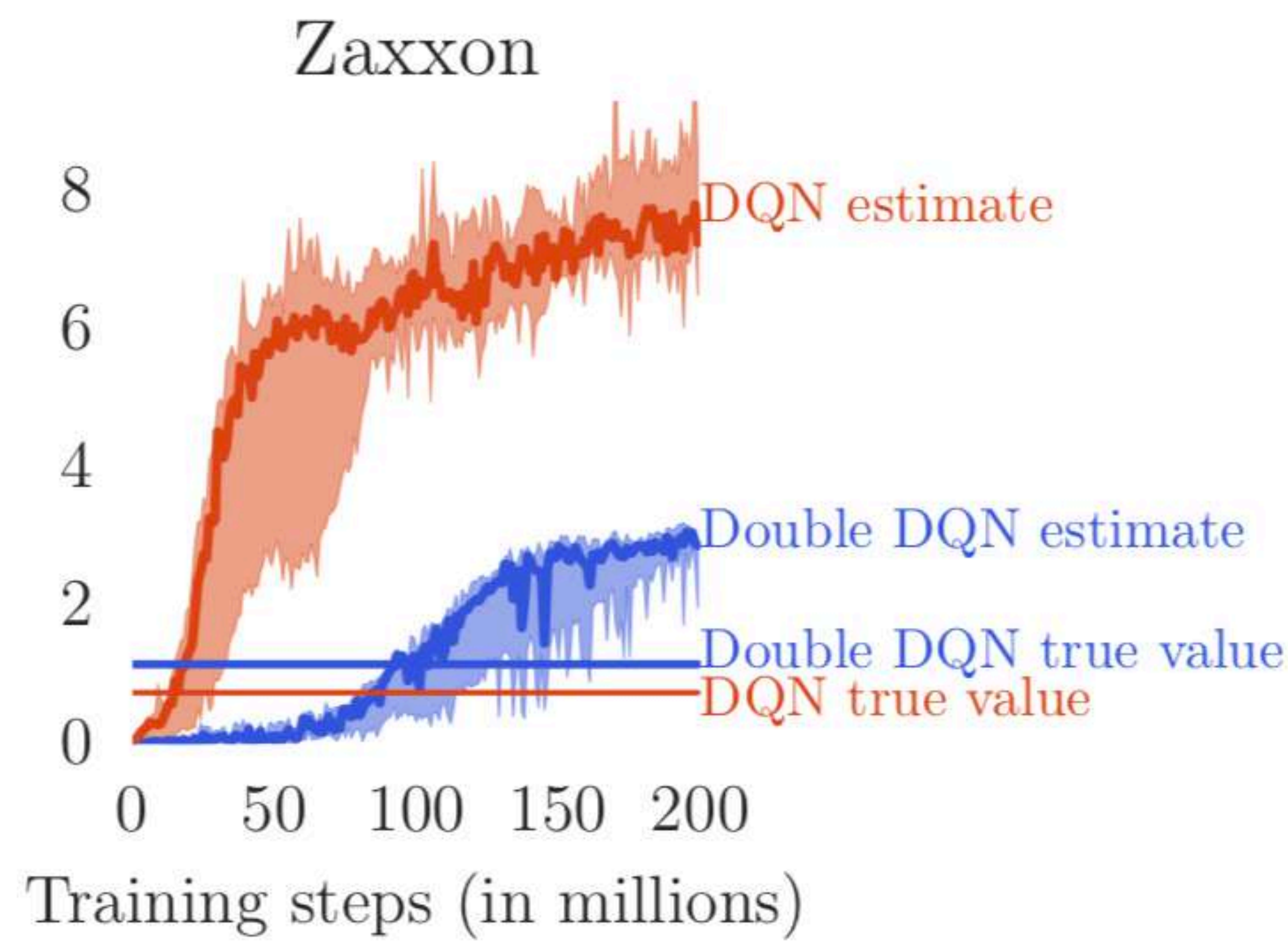


figure from [Hasselt et al., AAAI’16]

learning with experts/teachers

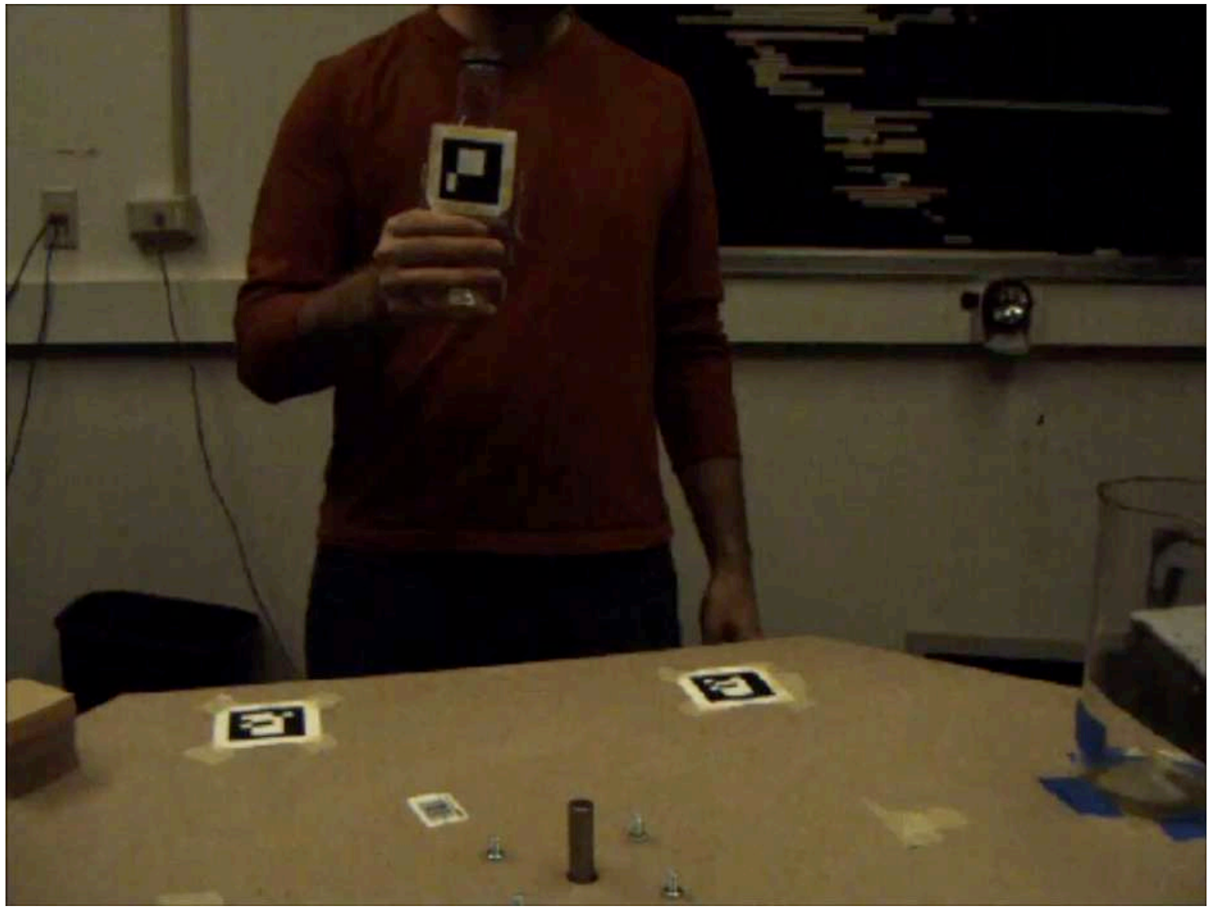

嬰兒的成長
"Imitation"
Andrew Meltzoff's Lab
Center for Mind Brain & Learning
University of Washington, USA
我要給你看這個
準備好了嗎?注意看


幼崽们之前从不明白它这样做的原因

Imitation learning

Expert/teacher provide demonstrations

$$s_0 \to a_1 \to s_1 \to a_2 \to s_2 \to \cdots \to a_m \to s_m$$



https://www.youtube.com/watch?v=ydnjS__8Ooc

agent learns from demonstrations to imitate the expert

Types of IL methods

- Copy actions : behavior cloning
- Copy intention: apprentice learning
- Copy distribution: generative adversarial IL

Copy actions: behavior cloning

demonstration data

$$s_0 \rightarrow a_1 \rightarrow s_1 \rightarrow a_2 \rightarrow s_2 \rightarrow \cdots \rightarrow a_m \rightarrow s_m$$

split into labeled data

$$D = \boxed{\begin{array}{r} s_0 \rightarrow a_1 \\ s_1 \rightarrow a_2 \\ \dots \\ s_{m-1} \rightarrow a_m \end{array}}$$

learning objective

$$\theta^* = \arg\min_{\theta} E_{s,a\sim D}\, \mathbf{loss}(\pi(s|\theta), a)$$

Behavior cloning examples

used human player data to initialize the policy in AlphaGo and AlphaStar

improvement in prepare the labeled data: remove highly correlated data

$$D = \begin{Bmatrix} s_0 \rightarrow a_1 \\ \cancel{s_1 \rightarrow a_2} \\ \cdots \\ s_{m-1} \rightarrow a_m \end{Bmatrix}$$

quickly learn a rough policy, no trial-and-error cost
but with limited power

Behavior cloning limitation

Supervised learning objective

$$\arg\min_\theta E_{x\sim\mathcal{D}} \operatorname{loss}(f_\theta(x), y(x))$$

Reinforcement learning objective

$$\arg\min_\theta E_{s\sim\mathcal{D}^{\pi_\theta}} \operatorname{cost}(s, \pi_\theta(s))$$

e.g. cost = -reward

Behavior cloning limitation

Supervised learning objective

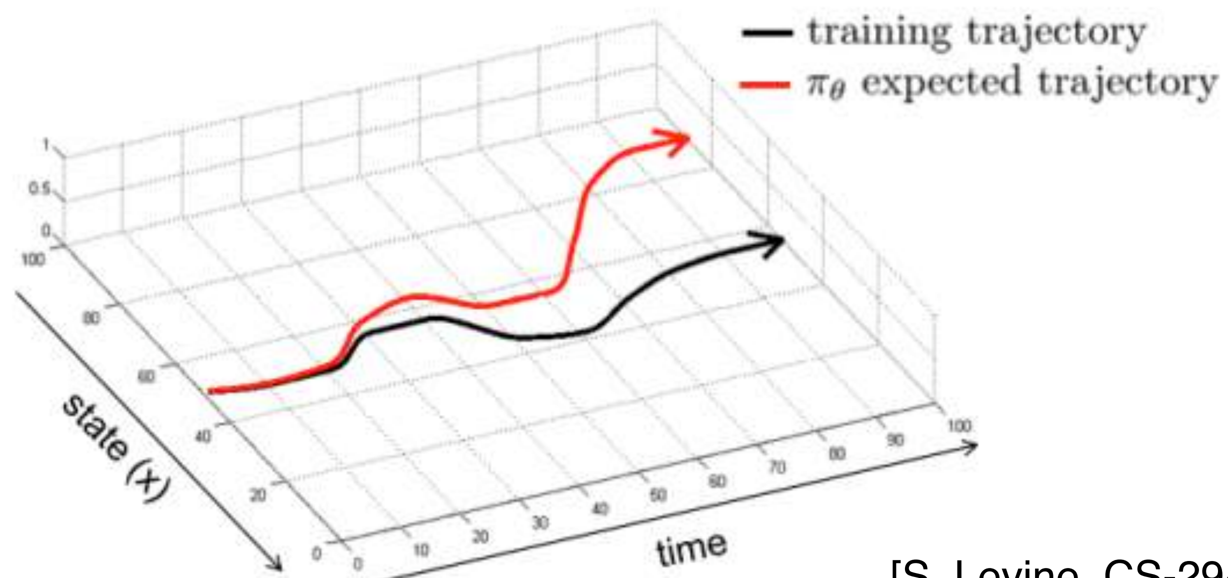
$$\arg\min_{\theta} E_{x\sim\mathcal{D}} \operatorname{loss}(f_\theta(x), y(x))$$

Reinforcement learning objective

$$\arg\min_{\theta} E_{s\sim\mathcal{D}^{\pi_\theta}} \textbf{cost}(s, \pi_\theta(s))$$

e.g. cost = -reward

Compounding error:



[S. Levine, CS-294-112-2]

Behavior cloning limitation - formally

Consider T-step reinforcement learning with bounded reward [0,1]

$$J(\theta) = E_{s,a,r\sim\pi_\theta}[\frac{1}{T}\sum_{t=1}^{T} r_t]$$

We have data from the optimal policy

$$s_0 \to a_1^* \to s_1 \to a_2^* \to s_2 \to \cdots \to a_T^* \to s_T$$

We apply BC(SL) to imitate the policy with a small classification error

$$E_{s,a^*}[\pi(s) \neq a^*] \leq \epsilon$$

Then the BC policy has a return as

$$J(\pi_{BC}) \geq J(\pi^*) - \frac{T+1}{2}\epsilon$$

[John Langford and Bianca Zadrozny Reducing T-step Reinforcement Learning to Classification.]

Proof idea

$$\begin{tikzcd}
 & 0 & & 0 & & & & J(\pi_{BC}) \\
s_0 \arrow[r] \arrow[ur, bend left, "\epsilon"] & a_1^* \arrow[r] & s_1 \arrow[r] \arrow[ur, bend left, "\epsilon"] & a_2^* \arrow[r] & s_2 \arrow[r] & \cdots \arrow[r] & a_T^* \arrow[r] & s_T
\end{tikzcd}$$

reward loss at step 1 $< \dfrac{T}{T}\epsilon$

reward loss at step 2 $< \dfrac{T-1}{T}\epsilon$

reward loss at step *t* $< \dfrac{T-t}{T}\epsilon$

total loss $< \dfrac{T+1}{2}\epsilon$

[John Langford and Bianca Zadrozny Reducing T-step Reinforcement Learning to Classification.]

More advanced theory

Consider continuing reinforcement learning with discount reward

$$J(\theta) = E_{s,a,r\sim\pi_\theta}[\sum_{t=1}^{\infty}\gamma^{t-1}r_t]$$

$\frac{1}{1-\gamma}$ is the total weights or *effective horizon*

Theorem 5.1. *Let π_E and π_{bc} denote the expert policy and BC imitator's policy. Assume that reward function is bounded in absolute value $R_{\max}$. Then the BC imitator has policy value error*

$$\left|V^{\pi_{bc}} - V^{\pi_E}\right| \le \frac{2R_{\max}}{(1-\gamma)^2}\mathbb{E}_{s\sim d_{\pi_E}}[D_{\mathrm{TV}}(\pi_{bc}(\cdot|s), \pi_E(\cdot|s))] \tag{16}$$

[Tian Xu, Ziniu Li, Yang Yu. On Value Discrepancy of Imitation Learning. https://arxiv.org/abs/1911.07027]

Copy actions: with super-expert

A sleepless expert is available to provide actions at any time

DAgger: Dataset Aggregation [S. Ross et al., AISTATS'2011]

1. start from random policy
2. run the policy to collect states
3. ask the expert to provide optimal actions
4. aggregate labeled dataset
5. learn policy by BC
6. repeat from step 2

The policy is closer to the expert

$$\left|V^{\pi} - V^{\pi_E}\right| \leq \frac{1}{1-\gamma}\left(\epsilon_T + \frac{1}{1-\gamma} + \sqrt{\frac{\log(1/\delta)(1-\gamma)}{mT}}\right)$$

Copy intention: apprentice learning

We view expert as a learning agent

environment: MDP (S,A,T)
reward function: R_E
learned optimal policy: π_E that achieves the optimal return

$$s_0 \rightarrow a_1^* \rightarrow s_1 \rightarrow a_2^* \rightarrow s_2 \rightarrow \cdots \rightarrow a_T^* \rightarrow s_T$$

Imitator:

environment: MDP (S,A,T)   reward function: ?

Inverse RL

Learn the reward function from expert data

Key assumption:
expert data is from the expert policy that maximizes the return

Recall the Bellman equation:

$$V^\pi = R + \gamma P^\pi V^\pi$$
$$V^\pi = (I - \gamma P^\pi)^{-1} R$$

Optimality condition: for any other policy

$$P^{\pi^*} V^{\pi^*} \geq P^\pi V^\pi$$

So the reward function needs to satisfy

$$(P^{\pi^*} - P^\pi)(I - \gamma P^{\pi^*})R \geq 0$$

Inverse RL algorithm

We cannot enumerate all policies, but we can find some

1. start from a random policy
2. find a reward function such that $\sum_{s,a \in D^*} R(s,a) \geq \sum_{s,a \sim \pi} R(s,a)$

   or more conveniently $R' = \arg\max_R \sum_{s,a \in D^*} R(s,a) - \sum_{s,a \sim \pi} R(s,a)$

3. lean a new policy according to *R'*
4. repeat from step 2

   and consider all generated policies $\sum_{s,a \in D^*} R(s,a) \geq \sum_{\pi} \sum_{s,a \sim \pi} R(s,a)$

Inverse RL algorithm

In linear reward function representation $R = w^\top \phi(s)$

the return of a trajectory is

$$w^\top \phi(s_0) + w^\top \phi(s_1) + \ldots + w^\top \phi(s_T) = w^\top \mu$$

1. start from a random policy
2. find a reward function by $w = \arg\max_w w^\top (\mu_E - \mu^i)$
3. lean a new policy according to the new reward function *w*
4. repeat from step 2

and consider all generated policies

$$w = \arg\max_w \min_i w^\top (\mu_E - \mu^i)$$

[Pieter Abbeel, Andrew Y. Ng: Apprenticeship learning via inverse reinforcement learning. ICML 2004]

Copy distribution

We want our policy generate state distribution close to the expert data

Distribution similarity measures

KL divergence: $$D_{KL}(P_r \| P_g) = \sum_{x \in X} P_r(x) \log \frac{P_r(x)}{P_g(x)}$$

Total Variation: $$D_{TV} = \sup_{x \in X} |P_r(x) - P_g(x)|$$

JS divergence: $$D_{JS} = D_{KL}(P_r \| P_g) + D_{KL}(P_g \| P_r)$$

Earth-Mover distance: $$D_W = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma} \|x - y\|$$

...

Match distributions

We want the imitator data distribution P_g = expert data distribution P_r

the goal is:

$$\frac{P_r(x)}{P_r(x)+P_g(x)} = \frac{P_g(x)}{P_r(x)+P_g(x)}$$

Use JS divergence:

$$D_{JS}(P_r, P_g) = \int \log\left(\frac{P_r(x)}{P_r(x)+P_g(x)}\right) P_r(x)\ \mathrm{d}\mu(x) + \int \log\left(\frac{P_g(x)}{P_r(x)+P_g(x)}\right) P_g(x)\ \mathrm{d}\mu(x)$$

But we only have data sets, but not distributions

Approximately match distributions

But we only have data sets, but not distributions

learn a distribution from data

Employ a classifier *D* to discriminate the two data sets

$$\begin{matrix} P_g \\ P_r \end{matrix} \rightarrow D \rightarrow \begin{matrix} \text{class 0} \\ \text{class 1} \end{matrix}$$

then we have an approximated distribution $D(x) = \dfrac{P_r(x)}{P_r(x) + P_g(x)}$

Approximately match distributions

But we only have data sets, but not distributions

learn a distribution from data

Employ a classifier *D* to discriminate the two data sets

P_g , P_r → D → class 0 , class 1

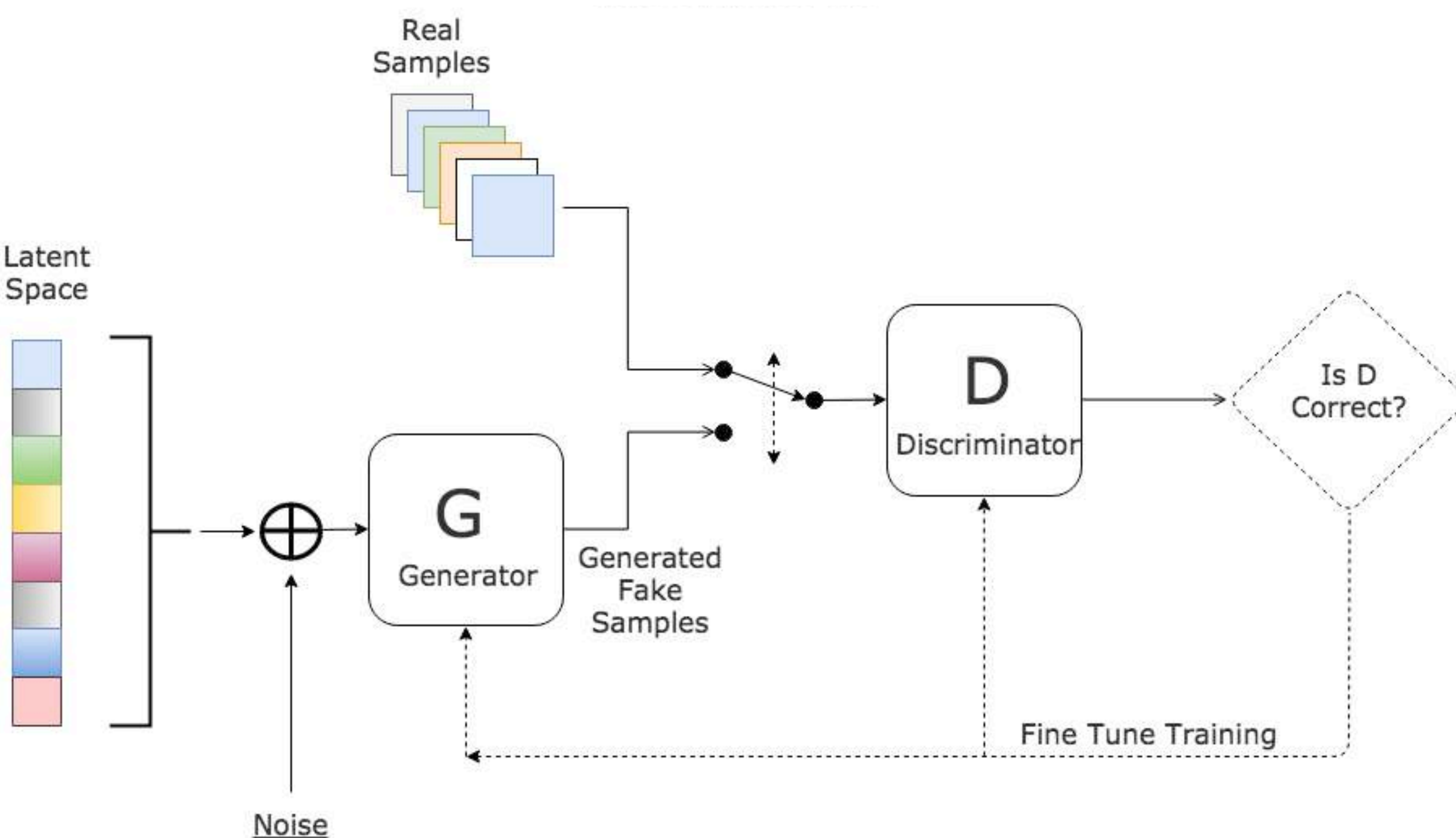
then we have an approximated distribution $D(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$

objective for minimize the JS divergence:

$$E_{x \sim P_r}[\log D(x)] + E_{x \sim P_g}[\log(1 - D(x))]$$

which is the objective of GAN [Goodfellow et al., NIPS'2014]

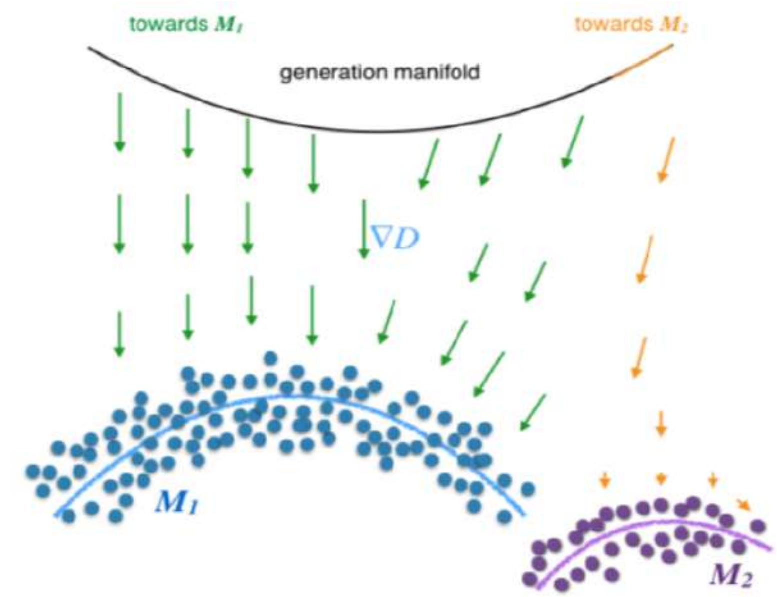
Generative Adversarial Networks



[Goodfellow et al., NIPS'2014]

Recent advances

Mode collapse problem



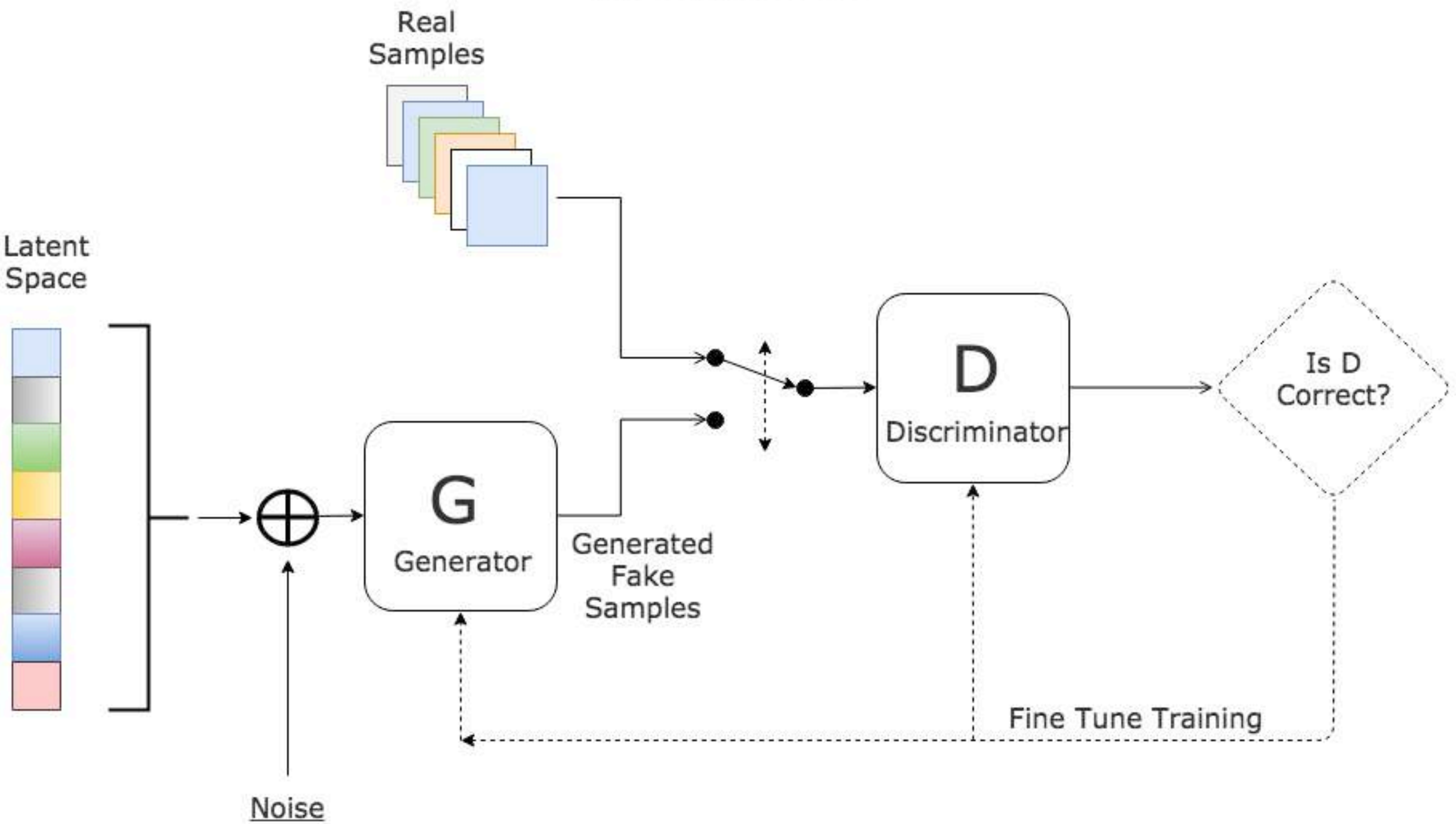
Are GANs Created Equal? A Large-Scale Study

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet

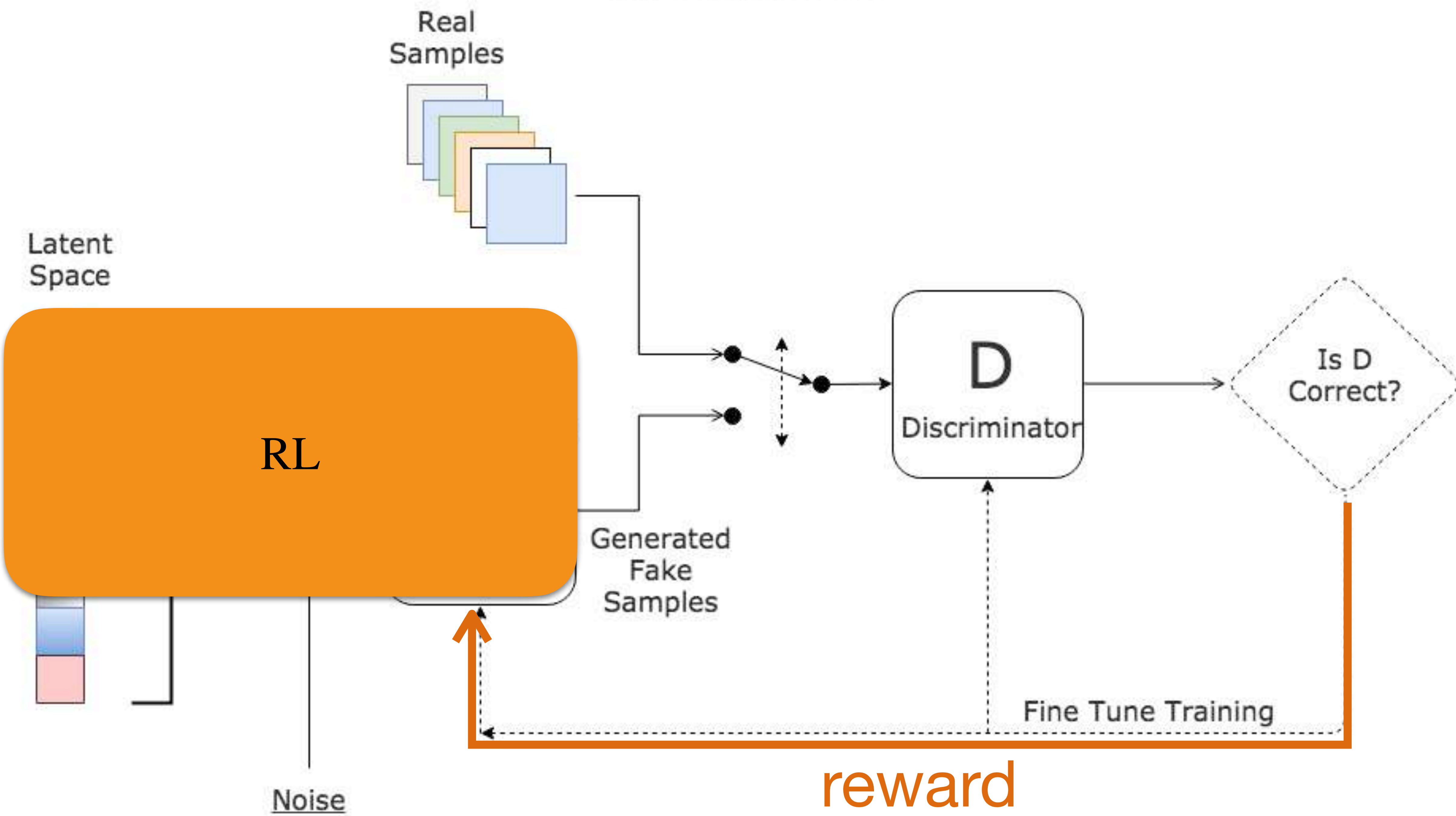
NIPS 2018
https://arxiv.org/abs/1711.10337

Many variants https://github.com/hindupuravinash/the-gan-zoo

GAN with RL generator



GAN with RL generator



Generative Adversarial Imitation Learning

(GAIL)

Start from a random policy

Loop:

1. learn the discriminator

$$\max_D E_{(s,a)\sim\pi_E}[\log(D(s,a))] + E_{(s,a)\sim\pi}[\log(1 - D(s,a))]$$

2. learn the policy using reward function

$$r(s,a) = \log(D(s,a))$$

Until convergence

[Jonathan Ho, Stefano Ermon: Generative Adversarial Imitation Learning. NIPS'2016, 4565-4573]

Connections between IRL & GAIL

IRL: iterates between reward function and policy

$$\max_{c \in C}(\min_{\pi \in \Pi} - H(\pi) + \mathrm{E}_{\pi}[c(s,a)]) - \mathrm{E}_{\pi_E}[c(s,a)]$$

$$H(\pi) = \mathrm{E}_{\pi}[-\log(\pi(a \mid s))]$$

GAIL: iterates between discriminator and policy

$$\min_{G}\max_{D} \mathrm{E}_{x \sim p_r}[\log(D(x))] + \mathrm{E}_{z \sim p_g}[\log(1 - D(G(z)))]$$

the discriminator (on trajectories) as the reward function for inverse reinforcement learning.

[Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. abs/1611.03852]

and better disentangled reward

[Justin Fu, Katie Luo and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. abs/1611.03852]

Various settings of imitation learning

RL problem: $<S,A,R,P>$

Observation: $<S',A',R',P'>$

Simplest: $S=S',A=A',P=P'$ internal data

Hardest: $S \neq S',A \neq A',P \neq P'$ observational data

Can practice: P ?

No — only from demonstration data
Yes — try in the environment

Environment reward accessible: R ?

No — simulate the expert
Yes — maximize the reward

Applications

- Initialize policy
- Learn without expressing reward function
- Simulate environments

[Finn et al., ICML'2016]

[Sliver et al., RSS'2008]

[Abbeel et al., IJRR'2010]

An example in real application

recommendation in E-commerce

→ prediction:
this sells good → sold more

→ prediction:
this looks bad → sold less

An example in real application

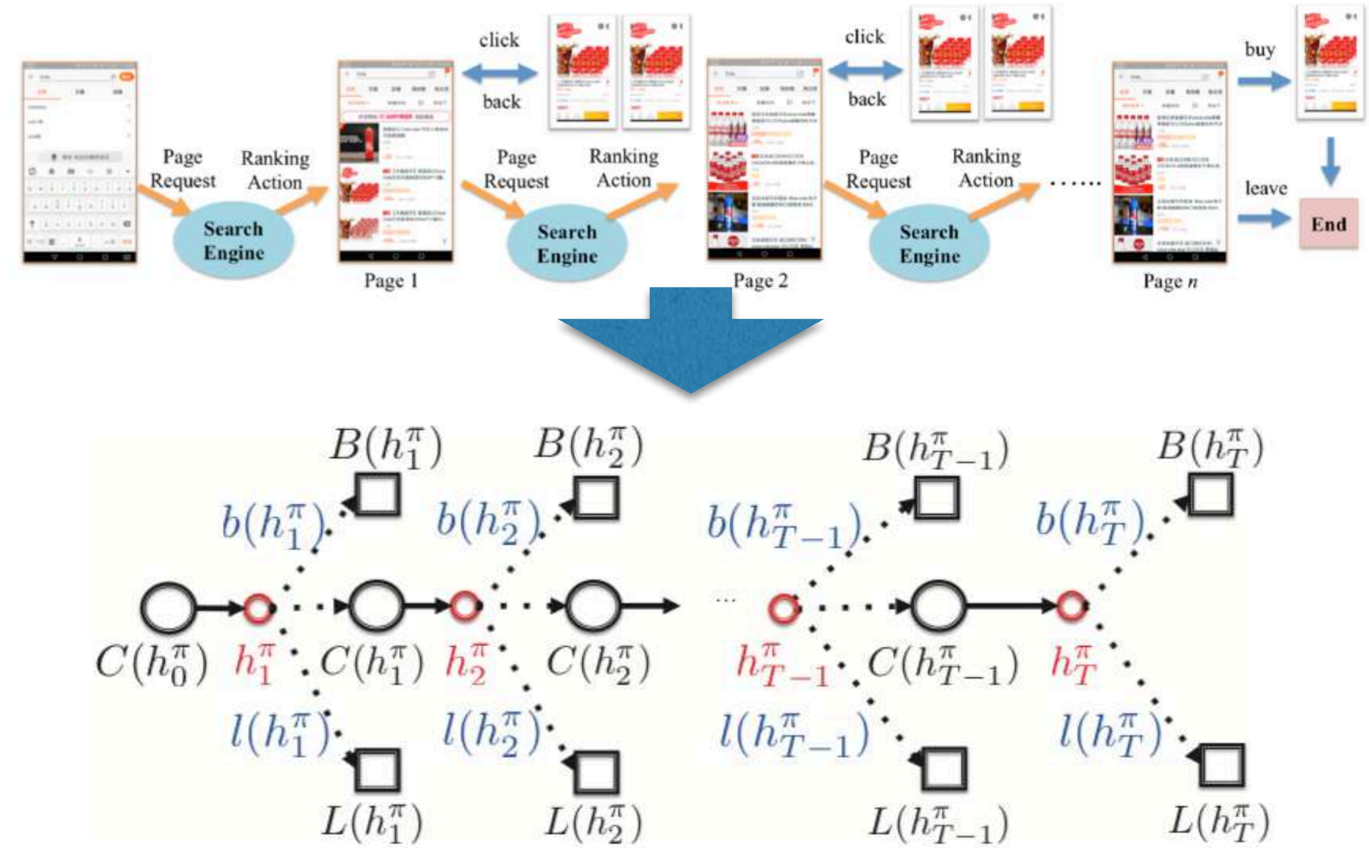
recommendation in E-commerce

→ prediction: this sells good → decision: recommend more → sold more

→ prediction: this looks bad → decision: recommend less → sold less

Experimental study

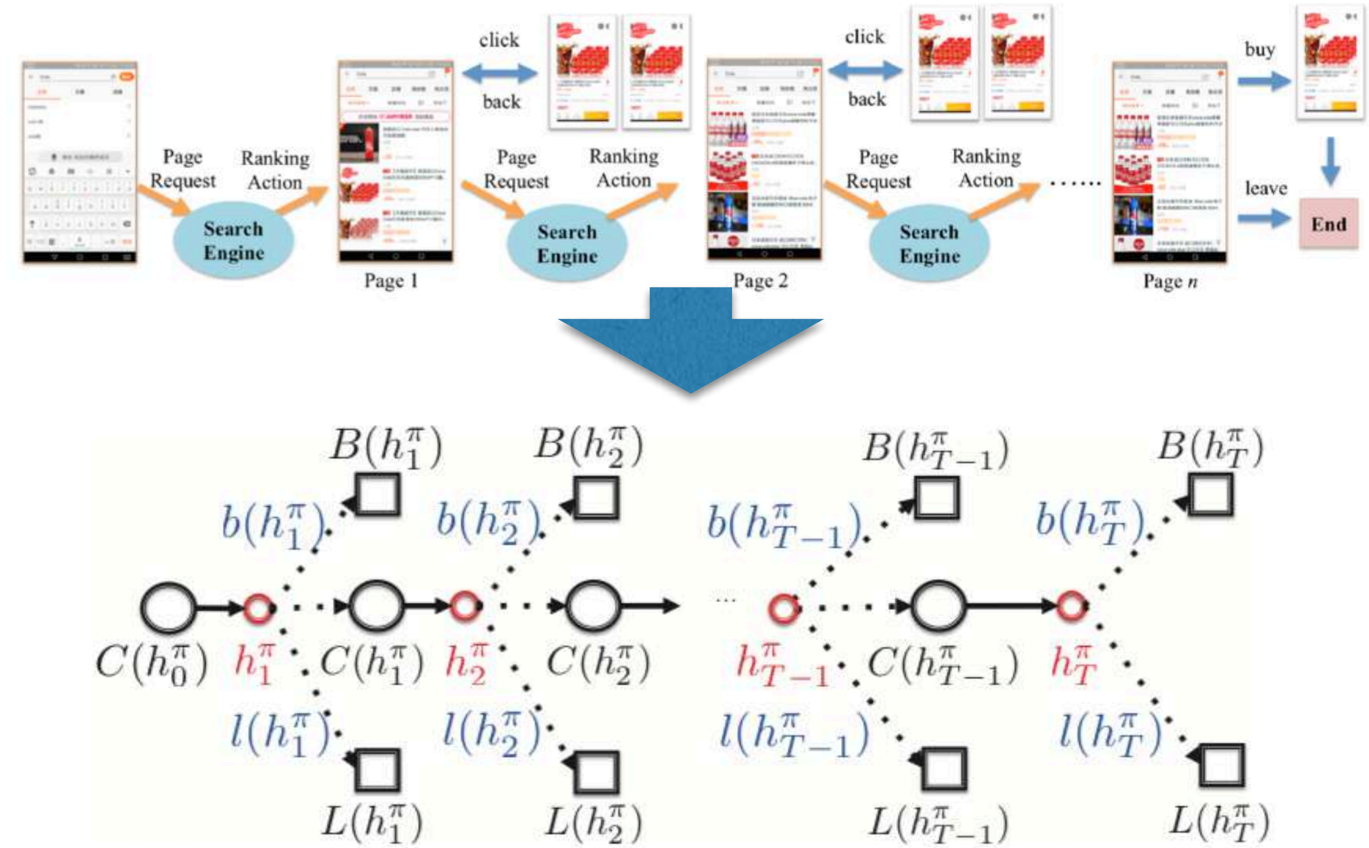
Initially:
simplified simulator



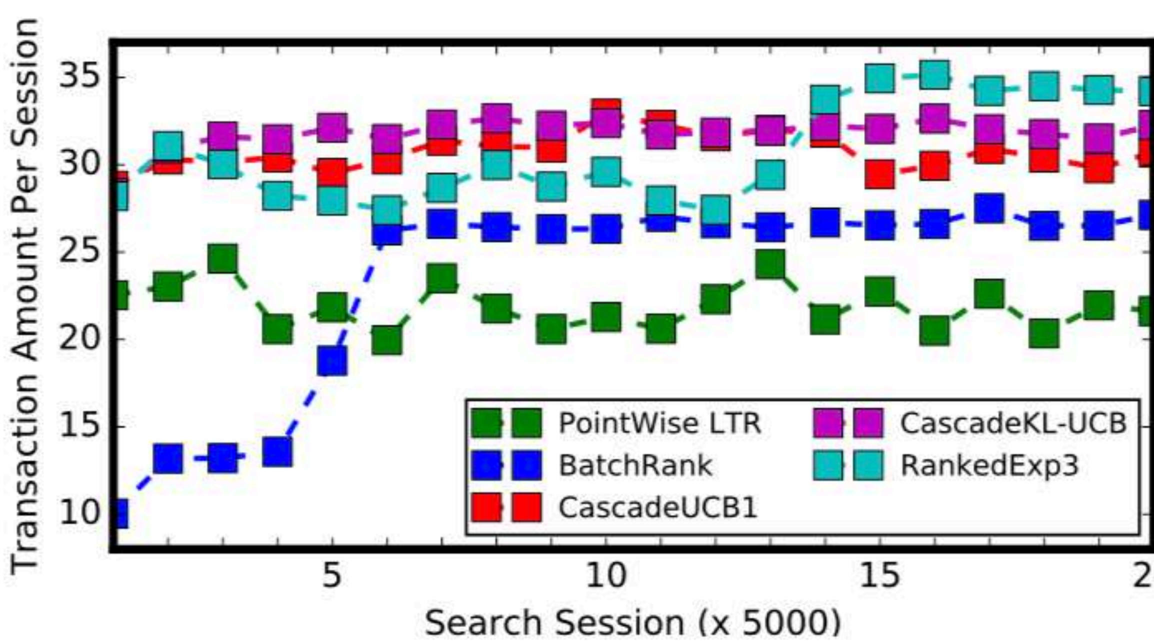
[Y. Hu, Q. Da, A. Zeng, Y. Yu and Y. Xu. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. KDD 2018.]

Experimental study

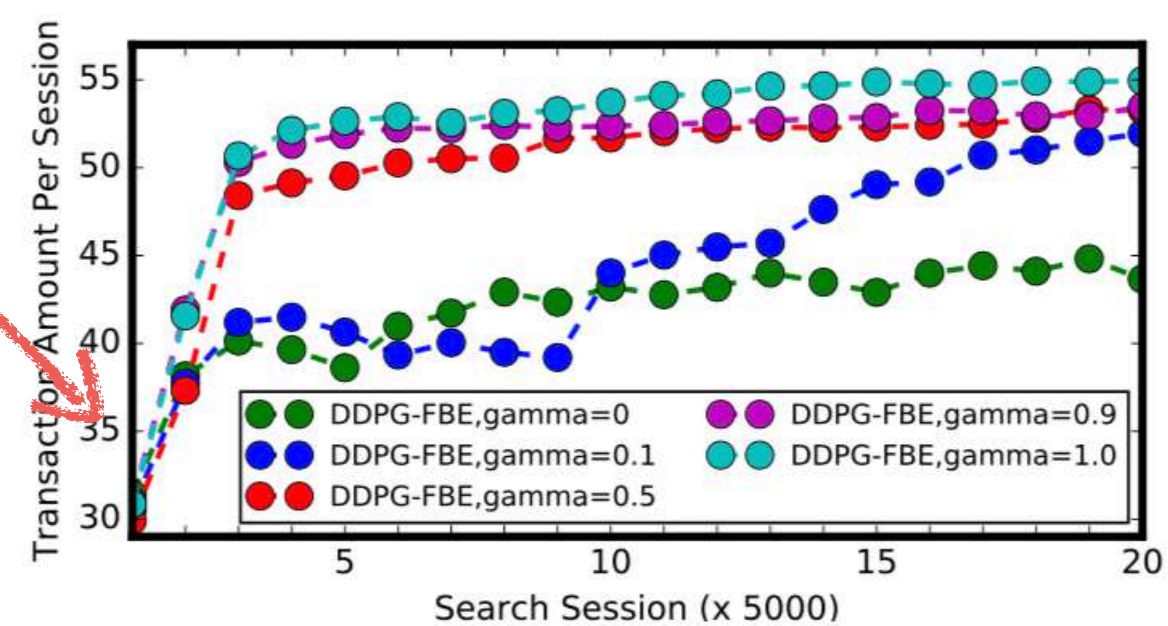
Initially:
simplified simulator



Experiment results



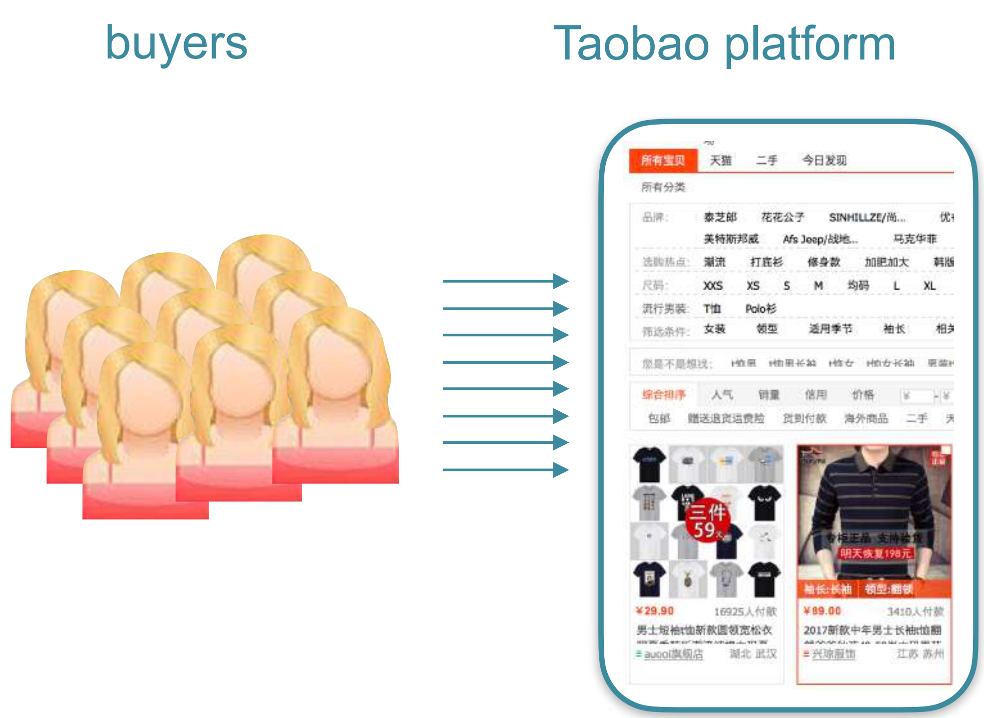
five online learning-to-rank algorithms



DDPG with full backup estimation of Q

[Y. Hu, Q. Da, A. Zeng, Y. Yu and Y. Xu. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. KDD 2018.]

Simulate Taobao !



real-world Taobao

How to simulate humans?

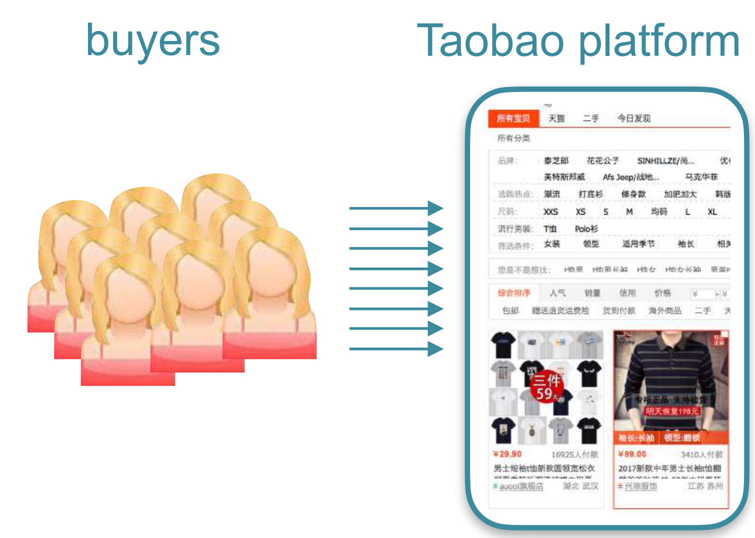
Learning buyers' actions ?

buyers data: observations -> actions

features labels

supervised learning ?

How to simulate humans?



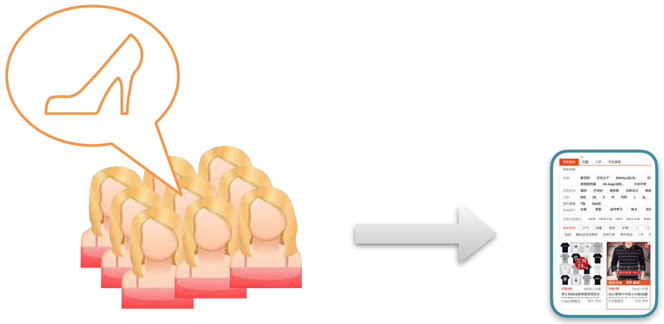
Learning buyers' actions ?

buyers data: observations -> actions

features labels

supervised learning ?

Idea: Motivated as buyers



1. buyers' motivation may keep unchanged in different platforms

learning reward function from observations via inverse reinforcement learning

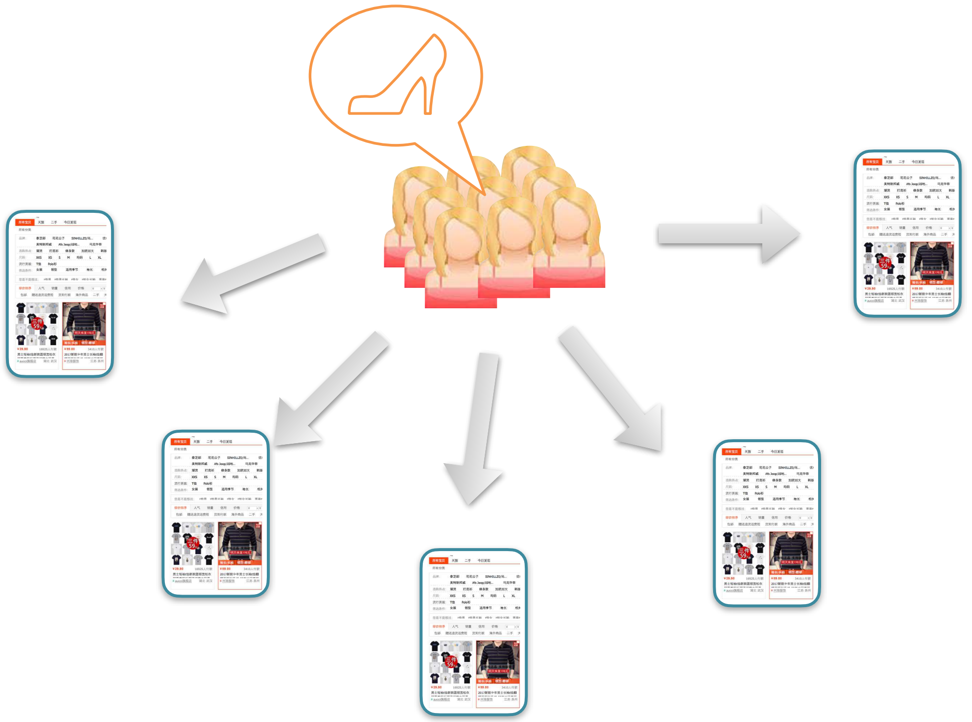
[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

Idea: Motivated as buyers

1. buyers' motivation may keep unchanged in different platforms

learning reward function from observations via inverse reinforcement learning

2. practice-to-learn by the agent in various situations



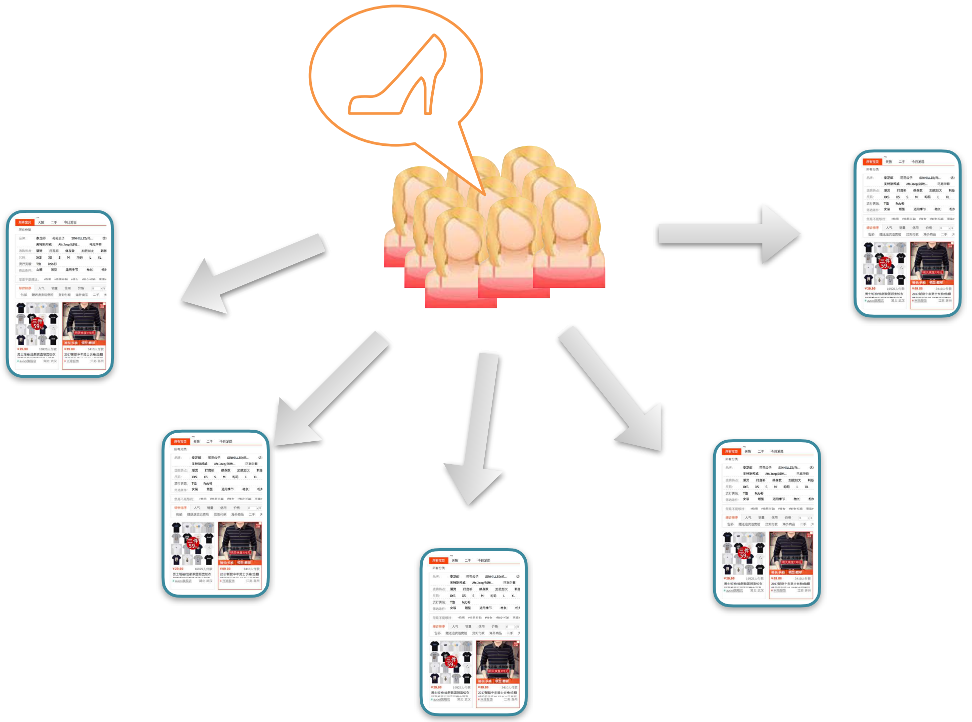
[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

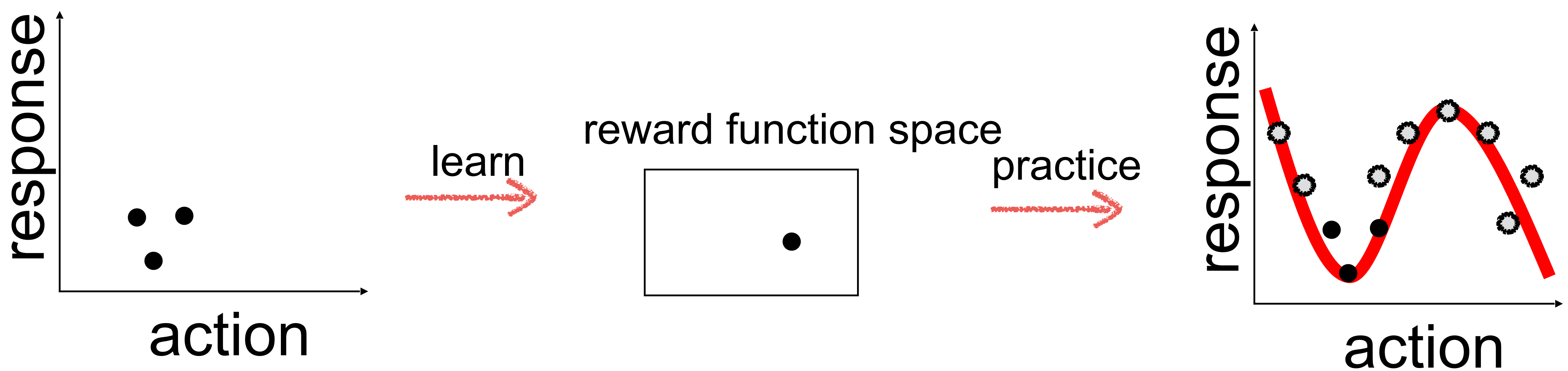
Idea: Motivated as buyers

1. buyers' motivation may keep unchanged in different platforms

learning reward function from observations via inverse reinforcement learning

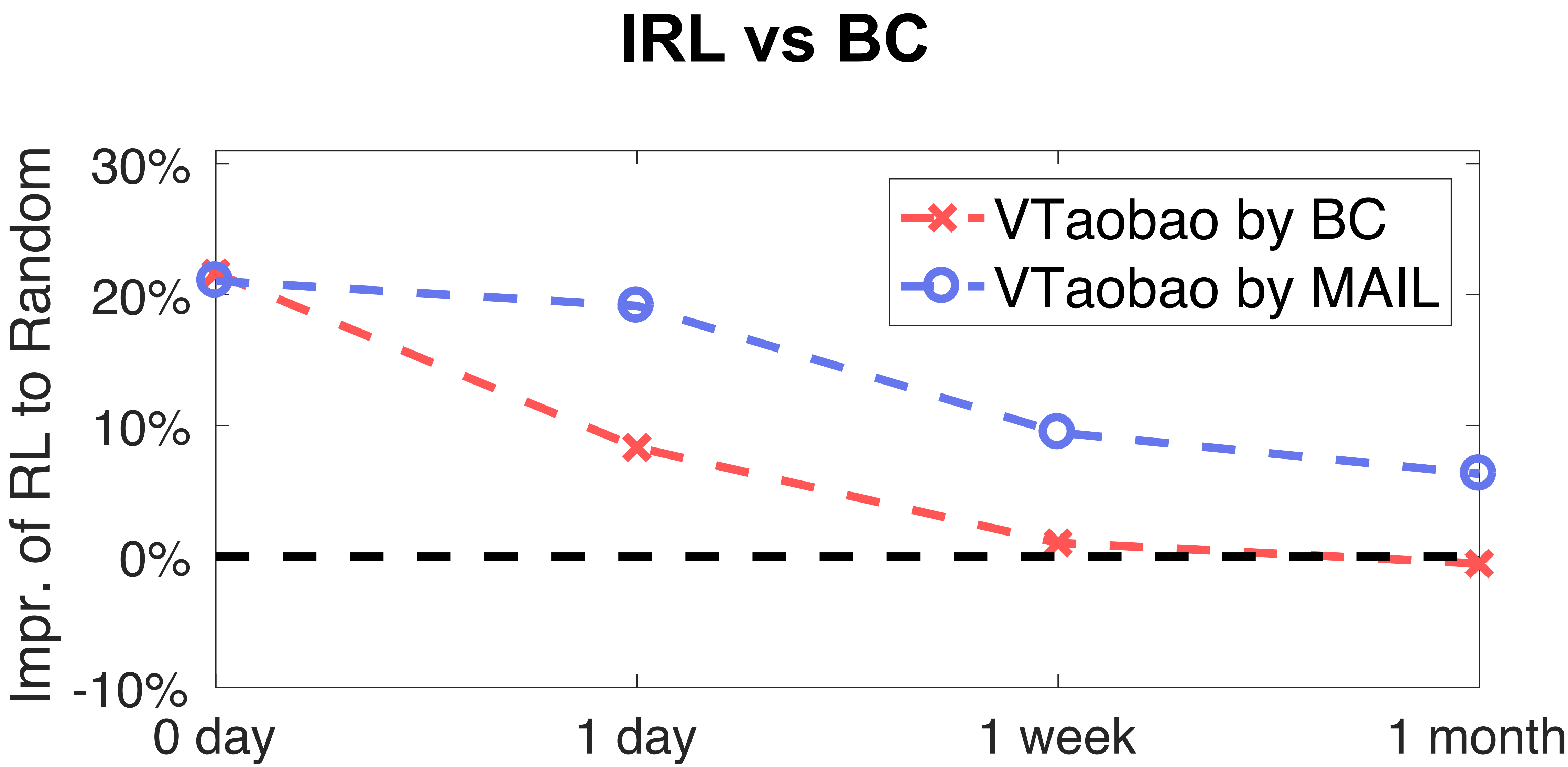
2. practice-to-learn by the agent in various situations





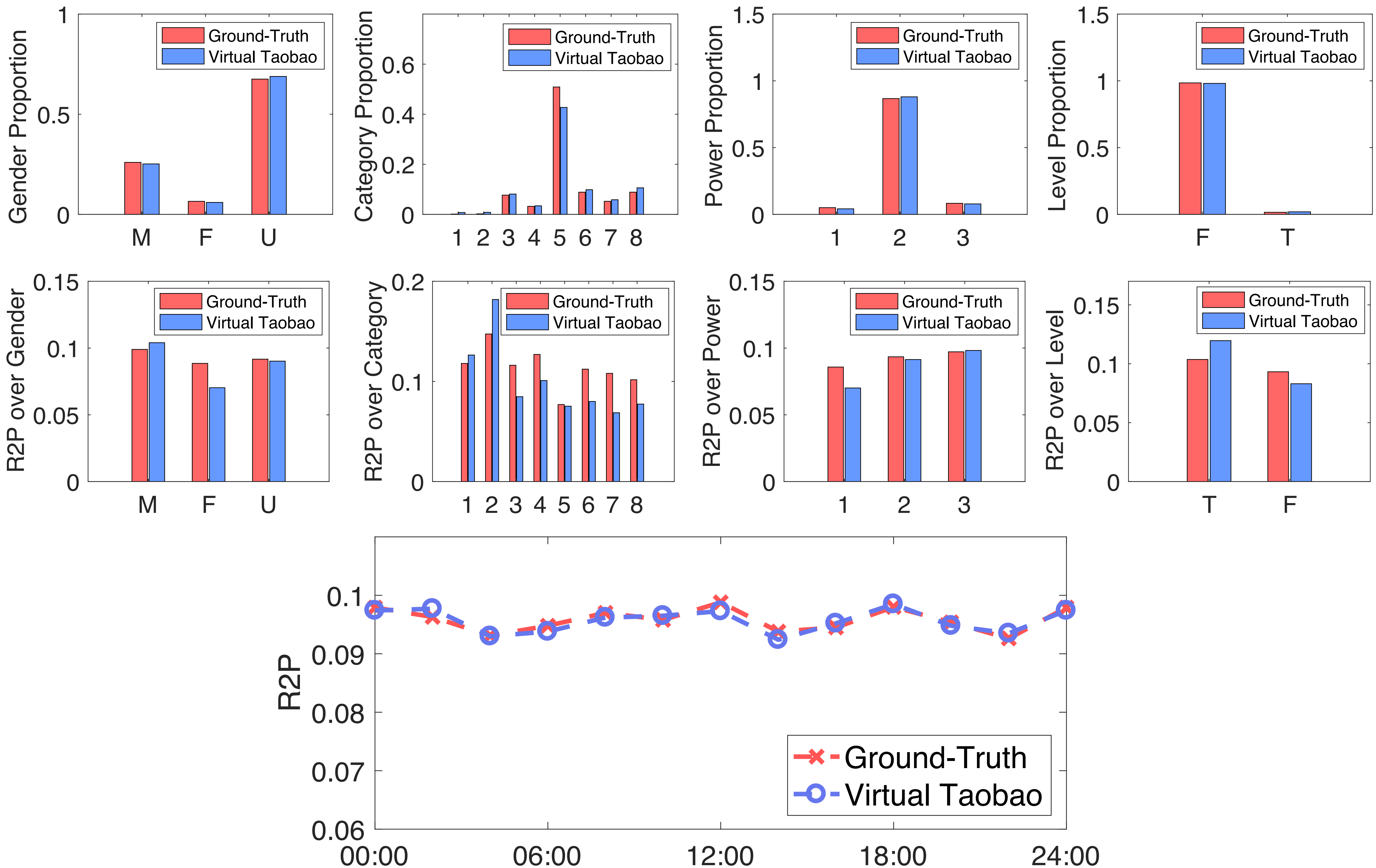
[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

Generalization



[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

Virtual vs real



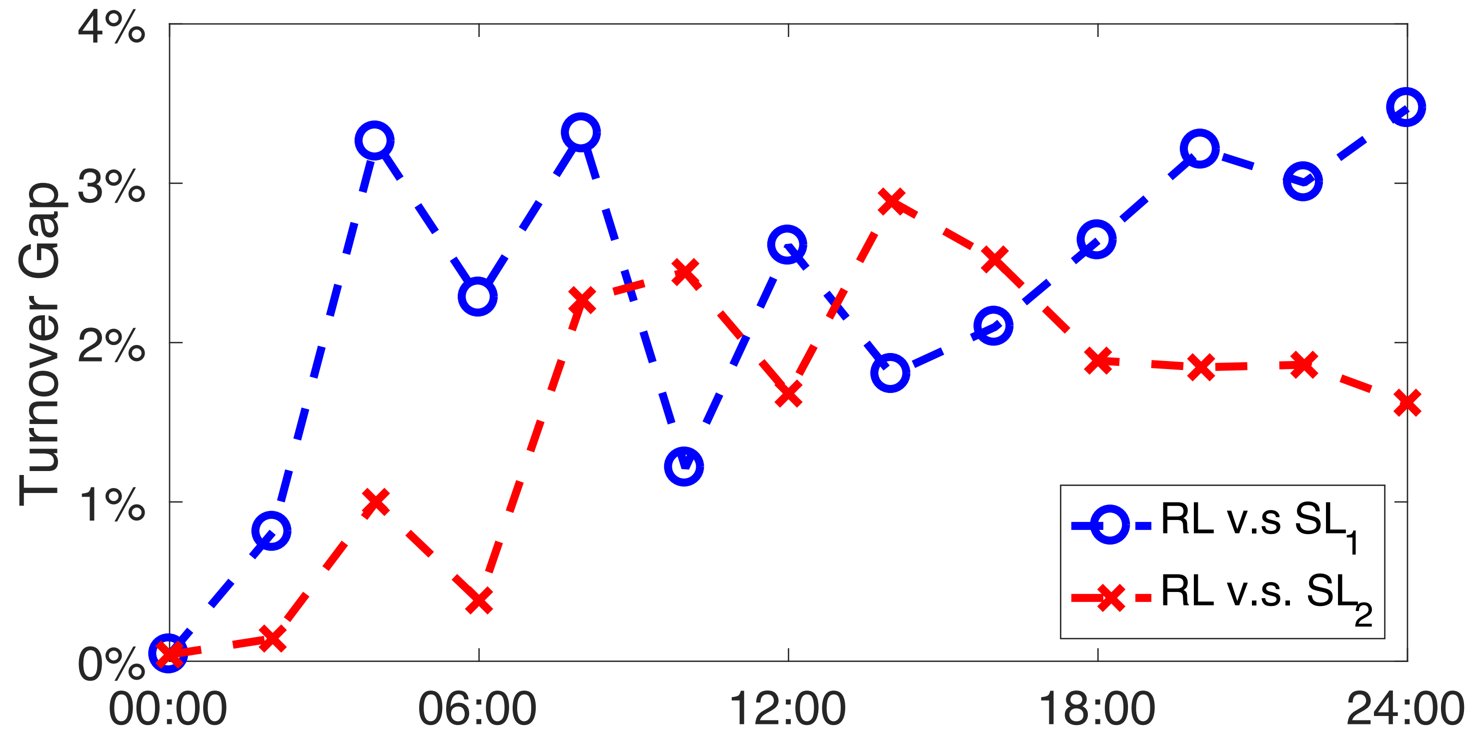
[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

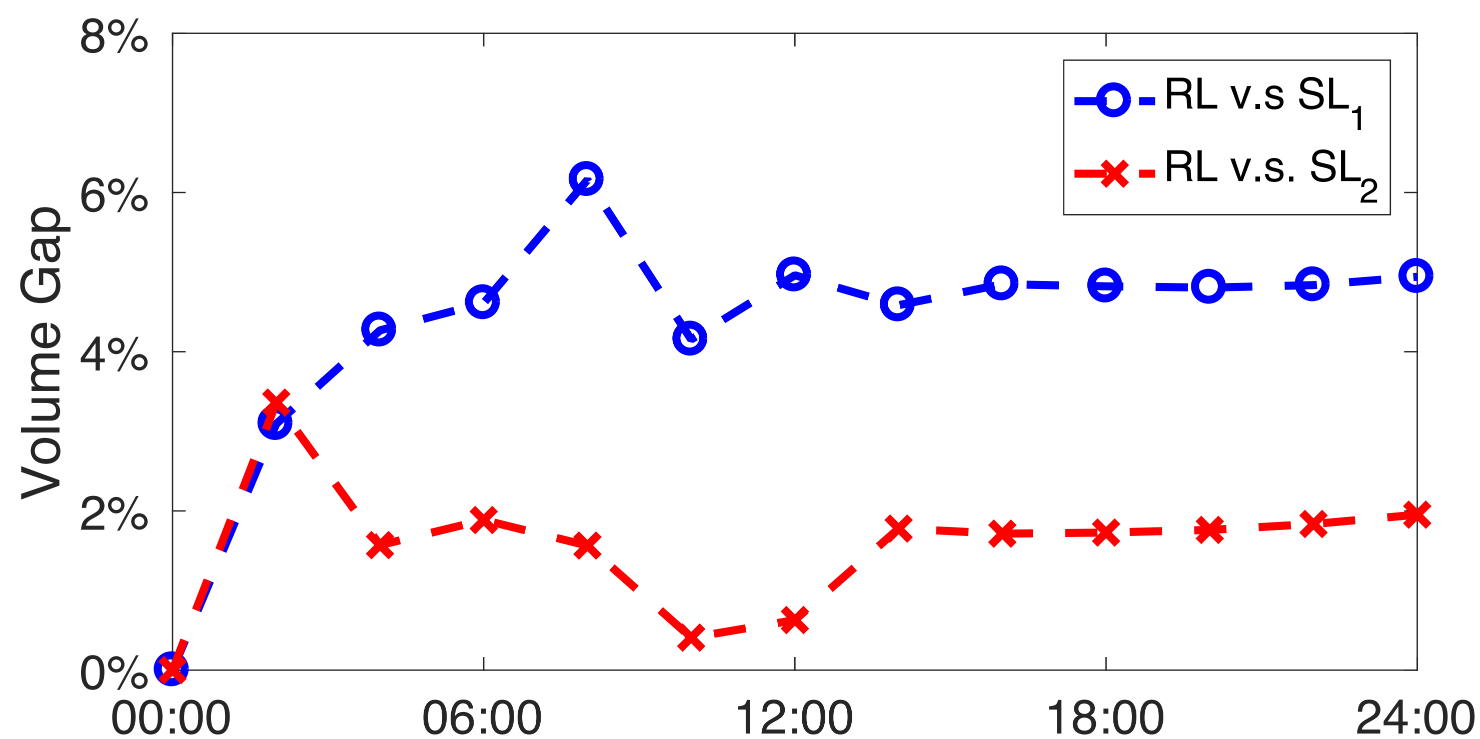
RL from Virtual Taobao

Train RL in Virtual Taobao

with conservative constraint

- 30% increase of GMV initially
- do not go far away from SL model



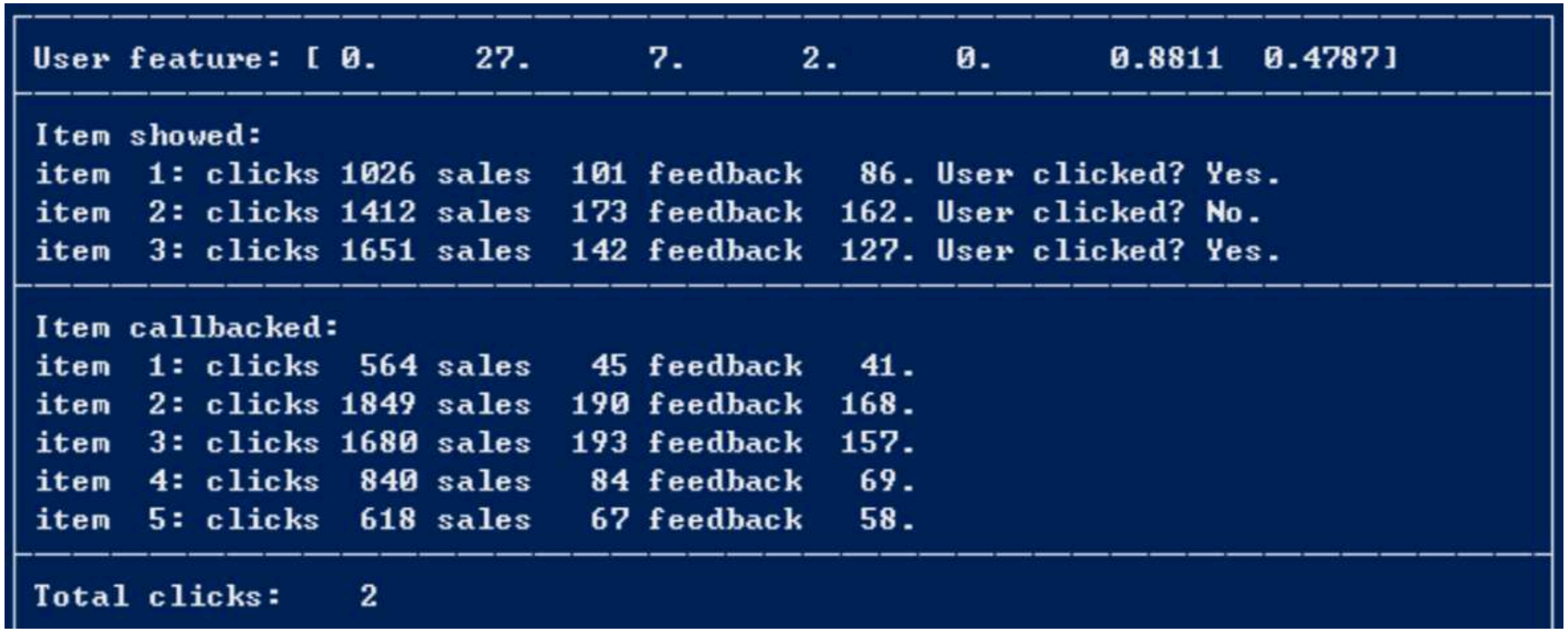


Improve online performance with **no cost**

[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

For academic purpose

VirtualTaobao simulator: https://agit.ai/Polixir/VirtualTaobao



- VirtualTaobao simulator provides a "live" environment just like the real Taobao
- anyone can test new recommendation algorithms interactively in their own laptops
- much more realistic than static data sets

A new simulator is on the way !

[J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. AAAI'19]

Thank you !

yuy@nju.edu.cn
http://lamda.nju.edu.cn/yuy