Reinforcement Learning with Robot Foundation Models: Data-Driven RL

Sung-eui Yoon

Scalable Graphics, Vision and Robotics

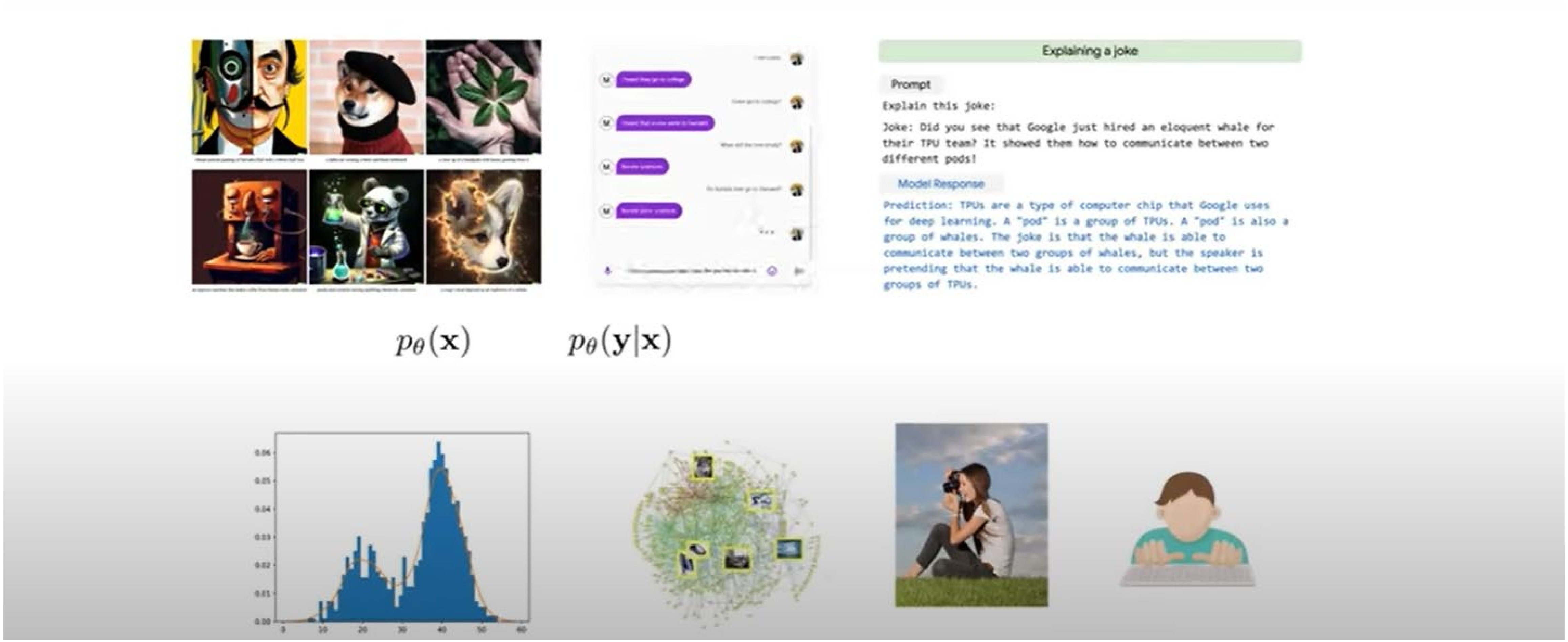
SGVR Lab
KAIST

Class Objectives

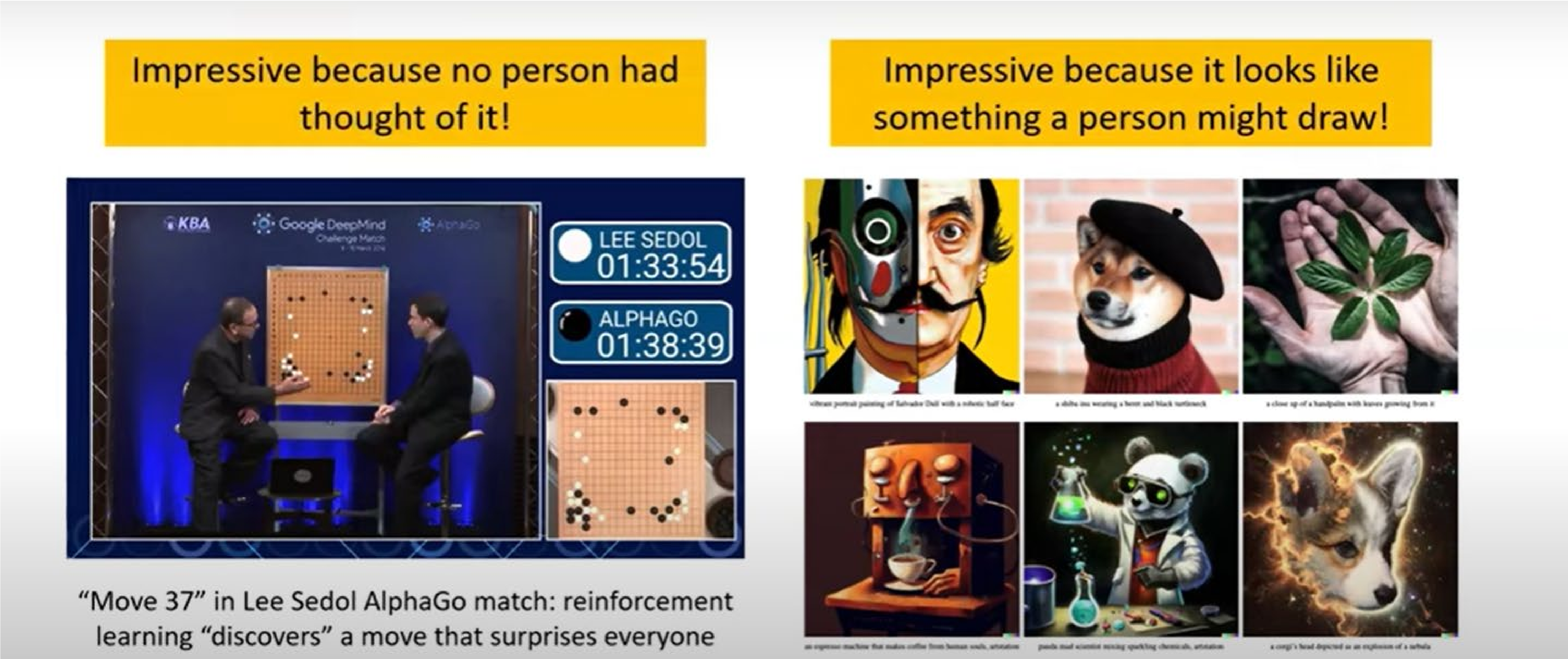
- **Data-driven RL**
  - Offline RL algorithms
  - Online finetuning from offline initializations
  - Making all this work with big scalable models (unclear yet)
  - Covered in draft of my book

Ack.: Sergey Levine's talk slides

From Modern data-driven AI (Estimation) to Goals: Rethinking Foundation Models



Emergent Behavior in RL vs. Human-Like Imitation in Data-Driven Models



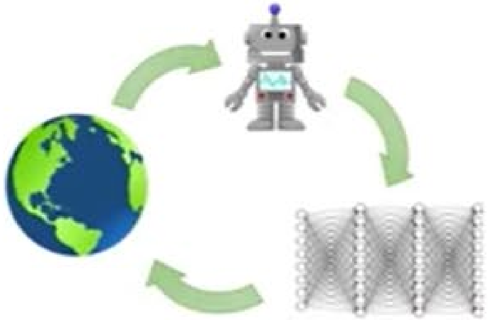
So, where are we now?

Data-Driven AI

+ learns about the real world from data

- doesn't try to do **better** than the data

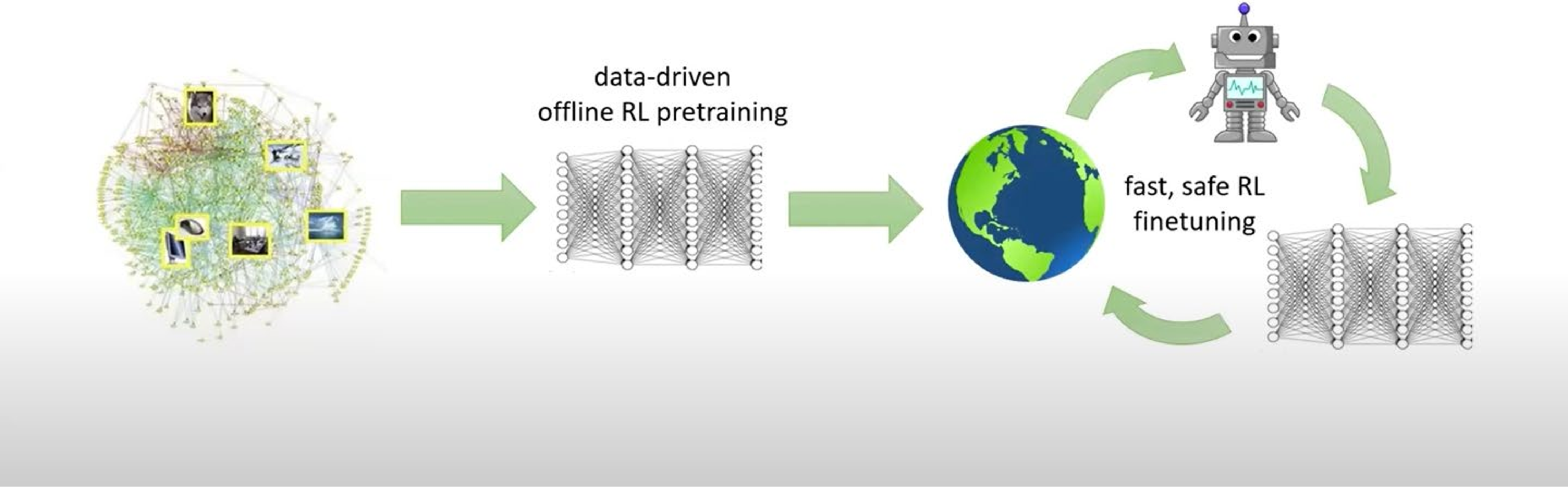
Reinforcement Learning



+ optimizes a goal with emergent behavior

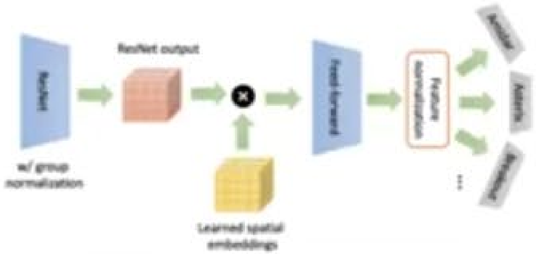
- doesn't make use of real-world data

The Recipe?



SGVR Lab KAIST

What can we accomplish when combining data and optimization?



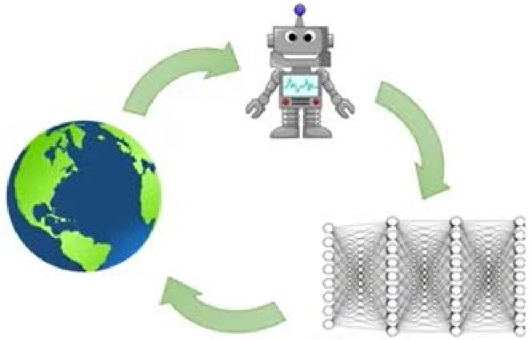
Data-driven RL algorithms

Robotic foundation models and RL

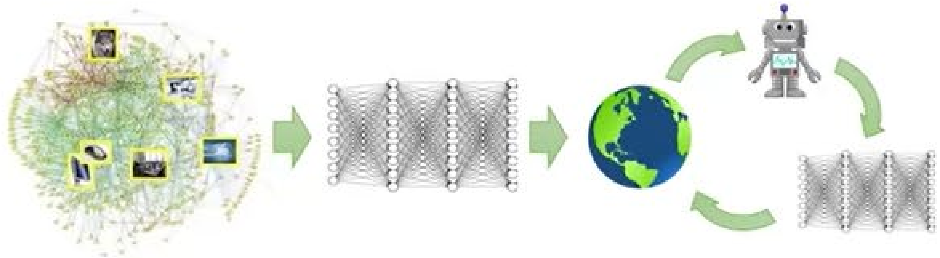
RL with generative models

What do we need to figure out?

Classic RL



Data-driven RL



Classic RL

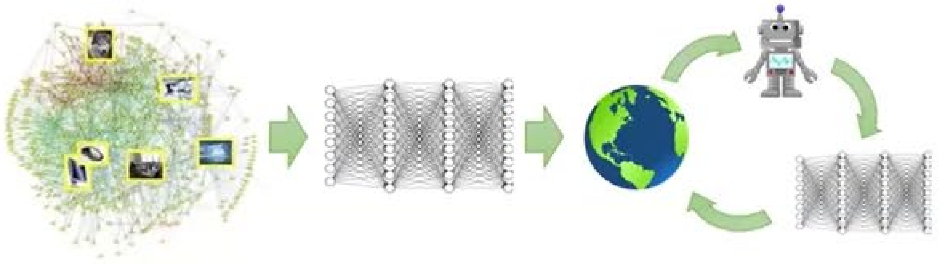
- Online, lifelong learning process
- Starts from scratch
- Largely **trial and error** driven
- **Central problems:**
  - Exploration
  - Sample efficiency
  - Optimization performance

Data-driven RL

- Offline pretraining + online finetuning
- Always start from data
- Largely **representation learning** driven
- **Central problems:**
  - Distributional shift
  - Scalability and stability
  - Representation learning with big models

To break this down..

Data-driven RL

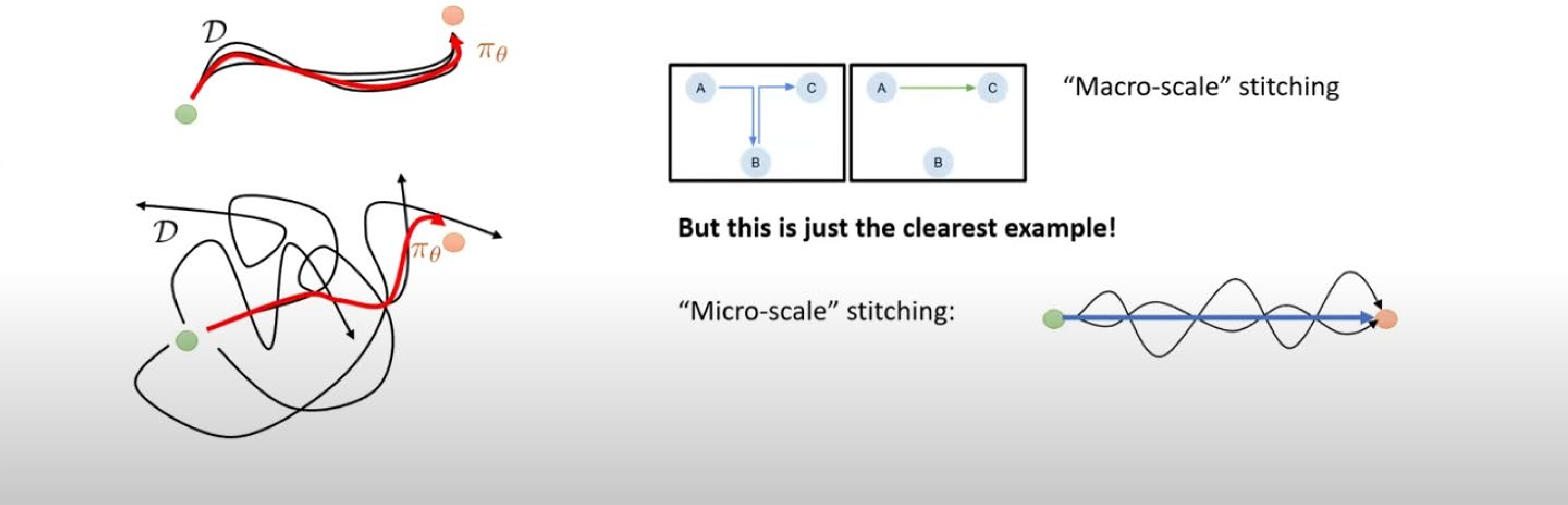


- Offline pretraining + online finetuning
- Always start from data
- Largely **representation learning** driven
- **Central problems:**
  - Distributional shift
  - Scalability and stability
  - Representation learning with big models

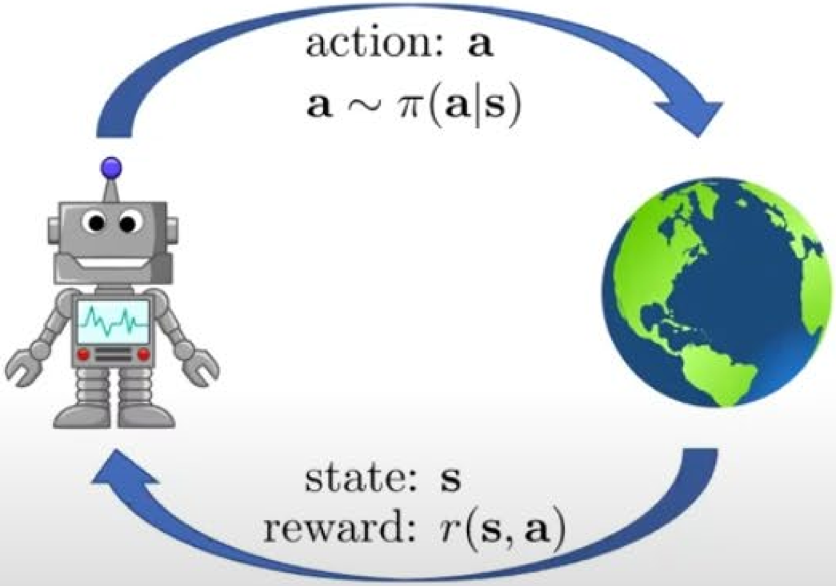
1. Offline RL algorithms — We understand this pretty well
2. Online finetuning from offline initializations — We understand this a little
3. Making all this work with big, scalable models — We hardly understand this at all

SGVR Lab KAIST

What do we expect offline RL to do?



Off-policy RL: a quick primer



RL objective: $\max_{\pi} \sum_{t=1}^{T} E_{\mathbf{s}_t, \mathbf{a}_t \sim \pi}[r(\mathbf{s}_t, \mathbf{a}_t)]$

Q-function: $Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^{T} E_{\mathbf{s}_{t'}, \mathbf{a}_{t'} \sim \pi}[r(\mathbf{s}_{t'}, \mathbf{a}_{t'})|\mathbf{s}_t, \mathbf{a}_t]$

$$\pi(\mathbf{a}|\mathbf{s}) = 1 \text{ if } \mathbf{a} = \arg\max_{\mathbf{a}} Q^{\pi}(\mathbf{s}, \mathbf{a})$$

$$Q^{\star}(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \max_{\mathbf{a}'} Q^{\star}(\mathbf{s}', \mathbf{a}')$$

enforce this equation at all states!

minimize $\sum_i (Q(\mathbf{s}_i, \mathbf{a}_i) - [r(\mathbf{s}_i, \mathbf{a}_i) + \max_{\mathbf{a}'_i} Q(\mathbf{s}'_i, \mathbf{a}'_i)])^2$

Some principles for offline RL

$$\cancel{Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')}$$

$$Q(\mathbf{s},\mathbf{a}) \leftarrow \underbrace{r(\mathbf{s},\mathbf{a}) + E_{\mathbf{a}'\sim\pi_{\text{new}}}[Q(\mathbf{s}',\mathbf{a}')]}_{y(\mathbf{s},\mathbf{a})}$$

what is the objective?

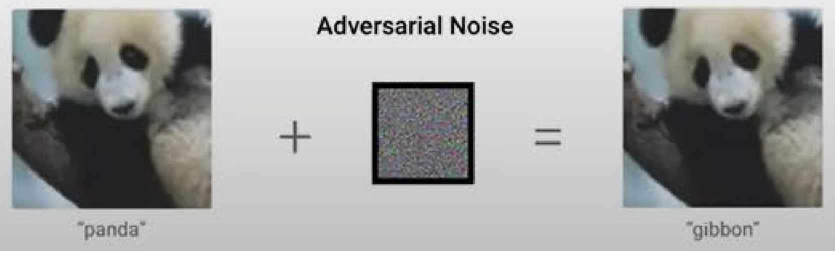
$$\min_Q E_{(\mathbf{s},\mathbf{a})\sim\pi_\beta(\mathbf{s},\mathbf{a})}\left[(Q(\mathbf{s},\mathbf{a}) - y(\mathbf{s},\mathbf{a}))^2\right]$$

behavior policy

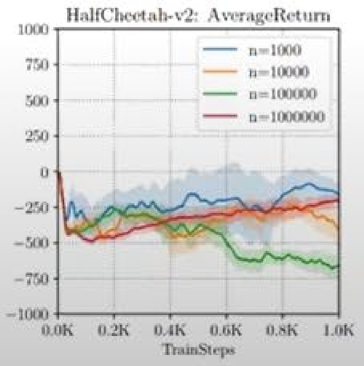
target value

expect good accuracy when $\pi_\beta(\mathbf{a}|\mathbf{s}) = \pi_{\text{new}}(\mathbf{a}|\mathbf{s})$

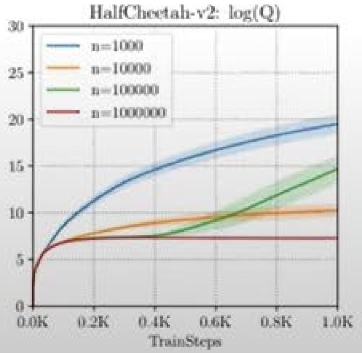
even *worse*: $\pi_{\text{new}} = \arg\max_\pi E_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$



how often does *that* happen?



how well it does



how well it *thinks* it does (Q-values)

"Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction" NeurIPS 2019

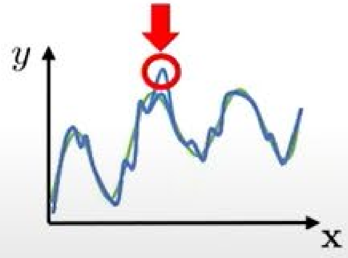
SGVR Lab KAIST

Some principles for offline RL

- Many different methods, similar principles seem to be effective:

use value-based methods (i.e., Q-learning or Q-function actor-critic)

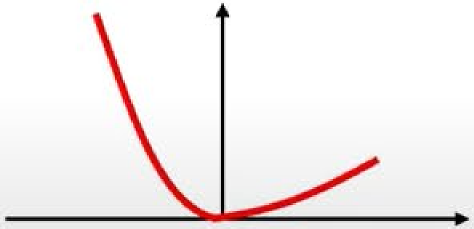
somehow fix the distributional shift problem



pessimism (e.g., CQL)

$$D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s}) \| \pi_\beta(\mathbf{a}|\mathbf{s})) \le \epsilon$$

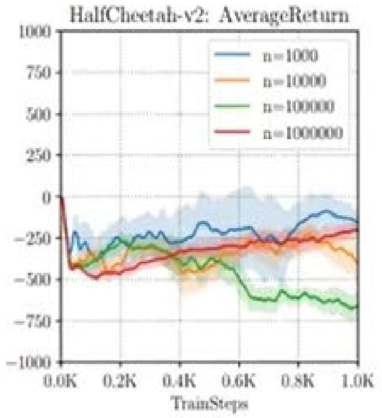
policy constraints (e.g., BRAC)



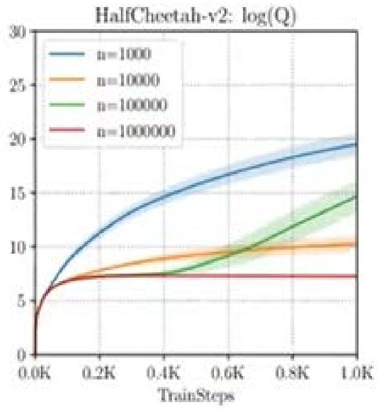
avoid OOD actions in updates (e.g., AWAC, IQL)

SGVR Lab KAIST

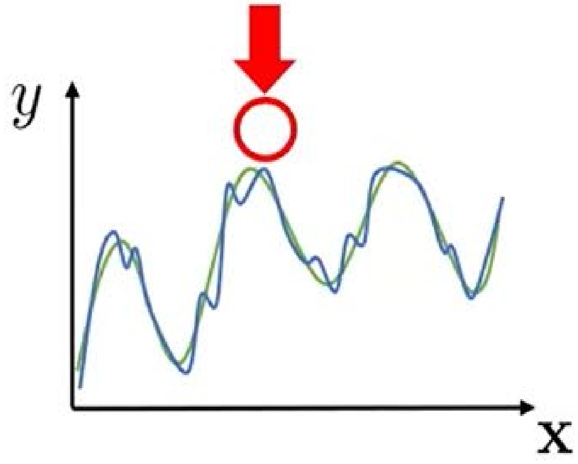
Conservative Q-learning



how well it does



how well it *thinks* it does (Q-values)



$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\pi} \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] - \alpha E_{\mathbf{s},\mathbf{a}\sim D}[Q(\mathbf{s},\mathbf{a})]$$ term to push down big Q-values

regular objective $$+E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + E_{\pi}[Q(\mathbf{s}',\mathbf{a}')])\right)^2\right]$$

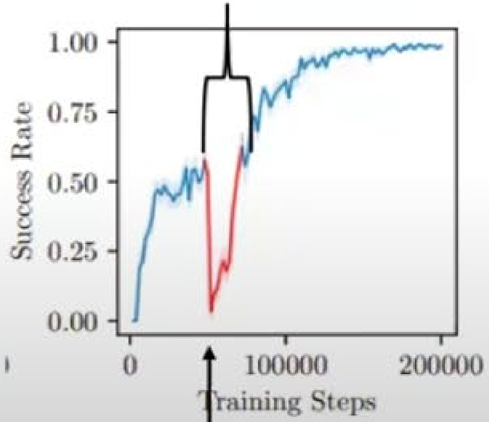
"Conservative Q-Learning for Offline Reinforcement Learning." NeurIPS 2020

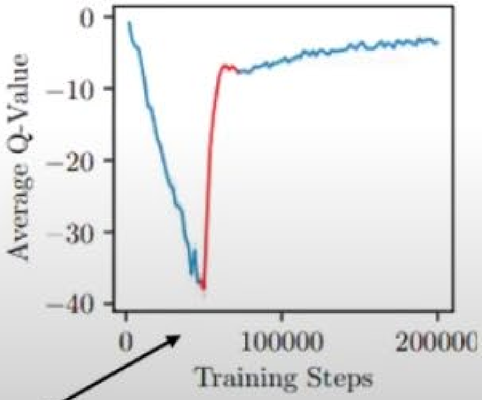
SGVR Lab KAIST

What about online finetuning?

$$\hat{Q}^{\pi} = \arg\min_{Q}\max_{\pi} \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] - \alpha E_{\mathbf{s},\mathbf{a}\sim D}[Q(\mathbf{s},\mathbf{a})]$$
$$+ E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + E_{\pi}[Q(\mathbf{s}',\mathbf{a}')])\right)^2\right]$$

this period wasted recovering
offline performance

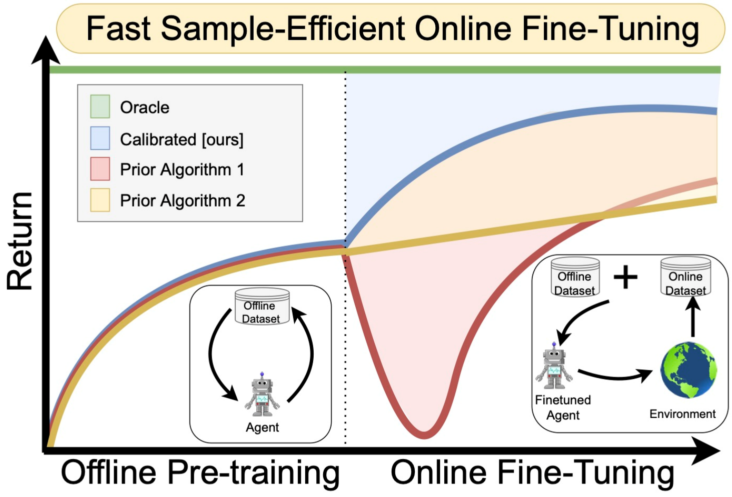


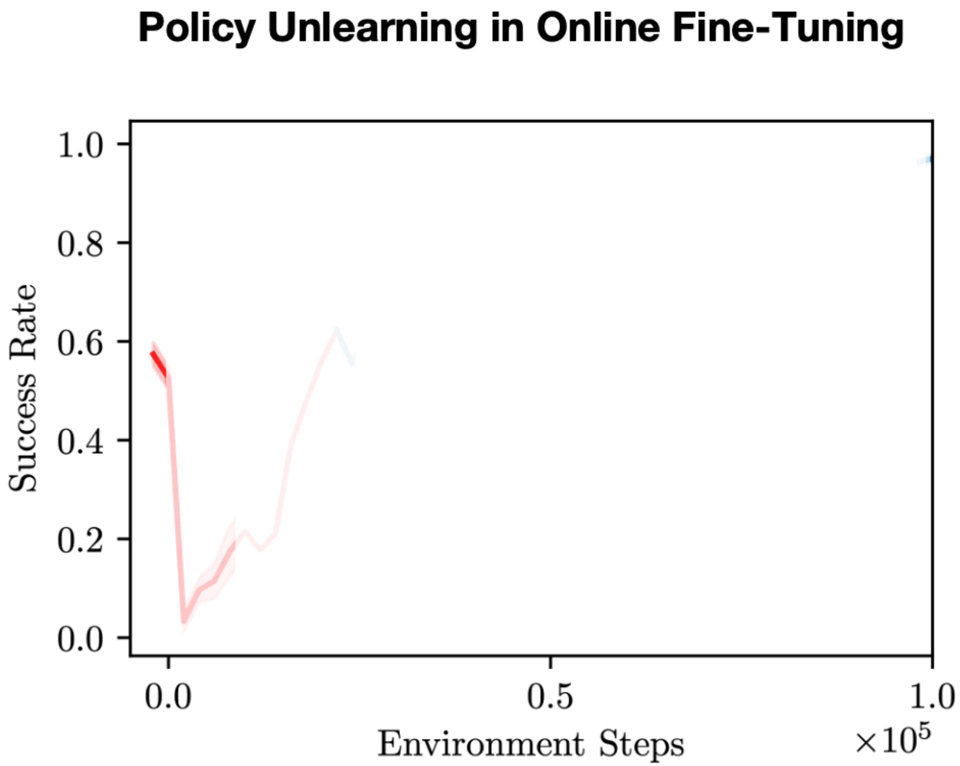


online training starts here (at 50k steps)

SGVR Lab KAIST

Example





SGVR Lab KAIST

Simple solution: Calibration

Key idea: need to put the offline-trained value function on the **right scale**

(we refer to this as *calibration*)

Calibration: learned values *upper bound* the values of *some* real policy

Conservatism: learned values *lower bound* the values of the *learned* policy

How do we ensure that our Q-function is calibrated?

before:

$$\min_Q \max_\pi \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$$

now:

$$\min_Q \max_\mu \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[\max\{Q(\mathbf{s},\mathbf{a}), V^\mu(\mathbf{s})\}]$$

how to get this?

$$V^{\pi_\beta}(\mathbf{s}_t) \approx \sum_{t'=t}^{\infty} \gamma^{t'-t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'})$$

SGVR Lab KAIST

The method: Cal-QL

Calibration: learned values *upper bound* the values of *some* real policy

Conservatism: learned values *lower bound* the values of the *learned* policy

ensures "conservatism"

ensures "calibration"

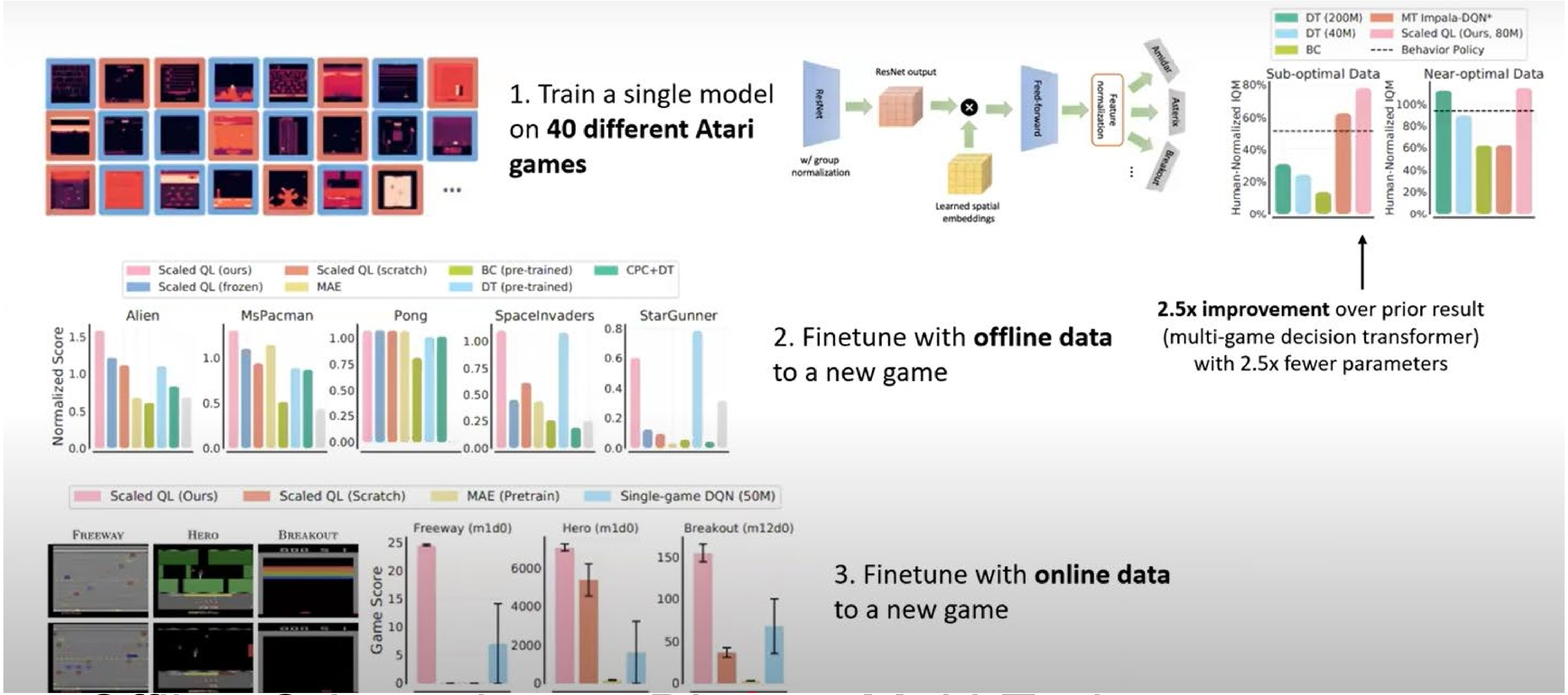
$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\pi} \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[\max\{Q(\mathbf{s},\mathbf{a}), V^{\pi_\beta}(\mathbf{s})\}] - \alpha E_{\mathbf{s},\mathbf{a}\sim D}[Q(\mathbf{s},\mathbf{a})]$$

$$+E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[\left(Q(\mathbf{s},\mathbf{a}) - r(\mathbf{s},\mathbf{a}) + E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]\right)^2\right]$$

Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning, NeurIPS 23

SGVR Lab KAIST

How about working with big models?

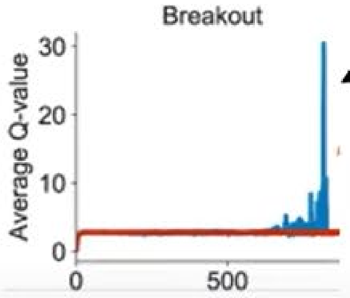


Offline Q-Learning on Diverse Multi-Task Data Both Scales and Generalizes, ICLR 23

The representation learning mystery

- Seems to be annoyingly hard to make this work with large transformer models
- Seems to require larger models with more capacity than we might expect (from, e.g., imitation learning)

Something about RL (i.e., TD learning) seems "harder" than supervised learning



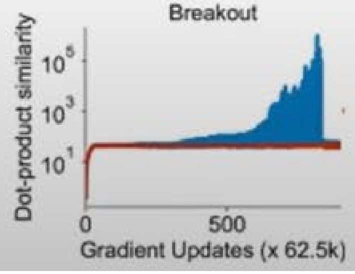
instability

action **resampled** from data distribution

$$Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}' \sim \pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s}',\mathbf{a}')]$$

$$Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \gamma Q(\mathbf{s}',\mathbf{a}')$$

action is always **exactly** the action in the dataset



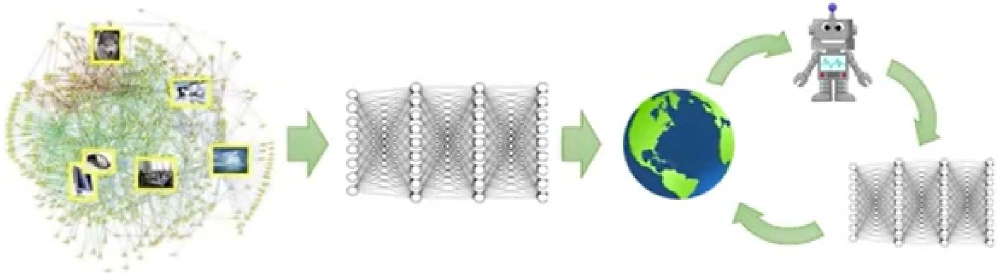
$$\phi(\mathbf{s},\mathbf{a})^T\phi(\mathbf{s}',\mathbf{a}')$$

TD learning updates value functions based on the difference between the predicted value and a new, partially observed value—hence, "temporal difference."

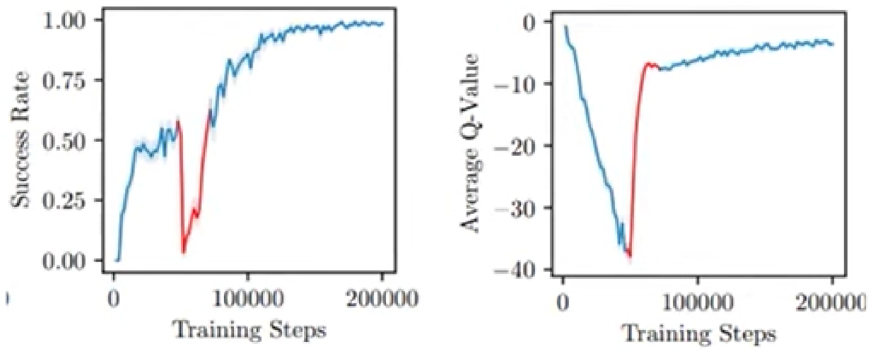
"DR3: Value-Based Deep Reinforcement Learning Requires Explicit Regularization" NeurIPS 2021

Summary and takeaways

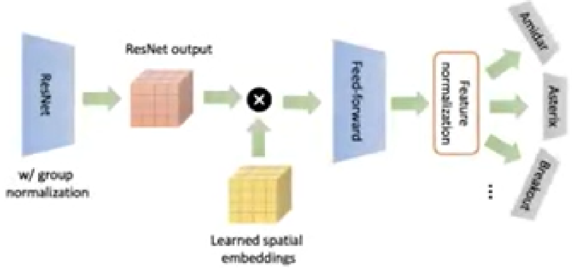
- Offline RL is an essential component of data-driven RL
  - We must handle the distributional shift between the offline data distribution and the new policy



- Online RL finetuning from offline initializations presents new challenges
  - We must be able to finetune via online RL without losing the benefits of the offline initialization



- Doing this with large models presents yet more challenges
  - Harder to make RL algorithms as scalable as supervised learning algorithms



SGVR Lab KAIST

Next Time

- **Robotics foundation models**

Homework

- **Come up with one question on what we have discussed today**
  - Write a question two times before the mid-term exam

- **Browse two papers**
  - Submit their summaries online before the Mon. Class