上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Imitation Learning

Weinan Zhang

Shanghai Jiao Tong University

http://wnzhang.net

Apr. 2024

2024年上海交通大学ACM班强化学习课程大纲

强化学习基础部分

（中文课件）

1. 强化学习、探索与利用
2. MDP和动态规划
3. 值函数估计
4. 无模型控制方法
5. 规划与学习
6. 参数化的值函数和策略
7. 深度强化学习价值方法
8. 深度强化学习策略方法

强化学习前沿部分

（英文课件）

9. 基于模型的深度强化学习
10. 模仿学习
11. 离线强化学习
12. 多智能体强化学习基础
13. 多智能体强化学习前沿
14. 基于扩散模型的强化学习
15. AI Agent与决策大模型
16. 技术交流与回顾

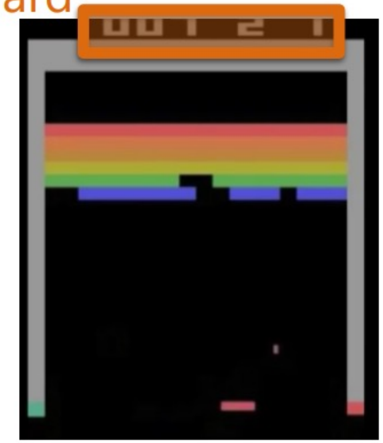
Overview

- Introduction to imitation learning
- Core methods of imitation learning
- Advanced methods of imitation learning
- Connection between imitation learning and GANs
- Recent applications of IL for robotics

Where does the Reward Function Come from?

Computer Games

reward



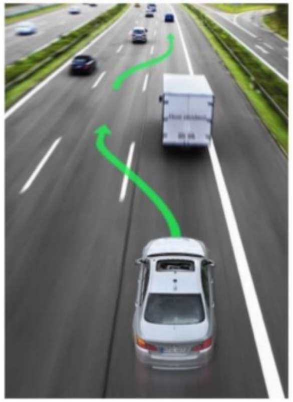
Mnih et al. '15

Real World Scenarios

robotics

dialog

autonomous driving

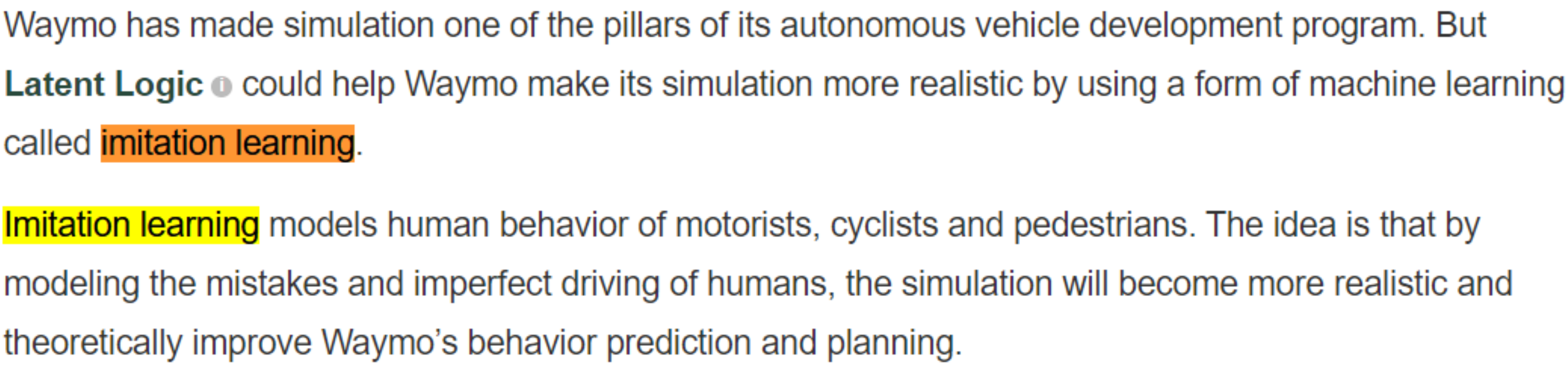


what is the reward?
often use a proxy

- Frequently easier to provide expert data than reasonable reward function
- Inverse reinforcement learning: infer reward function from demonstrations (rollouts) of expert policy

Slide credit Sergey Levine

Imitation Learning for Auto-driving



Waymo has made simulation one of the pillars of its autonomous vehicle development program. But **Latent Logic** could help Waymo make its simulation more realistic by using a form of machine learning called imitation learning.

Imitation learning models human behavior of motorists, cyclists and pedestrians. The idea is that by modeling the mistakes and imperfect driving of humans, the simulation will become more realistic and theoretically improve Waymo's behavior prediction and planning.

https://techcrunch.com/2019/12/12/waymo-buys-latent-logic-drives-deeper-into-simulation-and-europe/

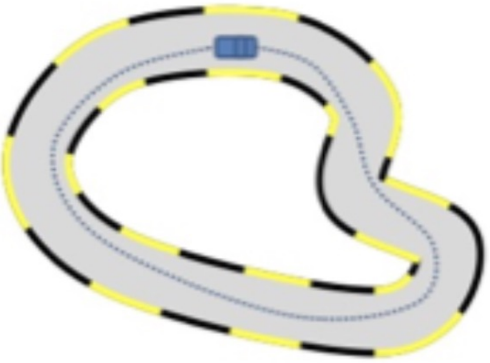
Imitation Learning in a Nutshell

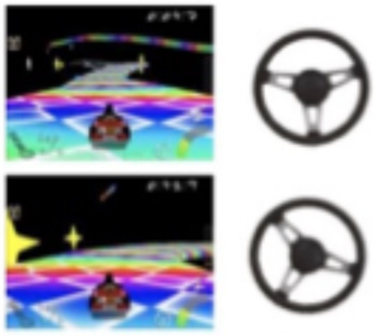
- Given: demonstrations or demonstrator
  - Normally without any reward signals
- Goal: train a policy to mimic demonstrations
  - And achieve good policy performance

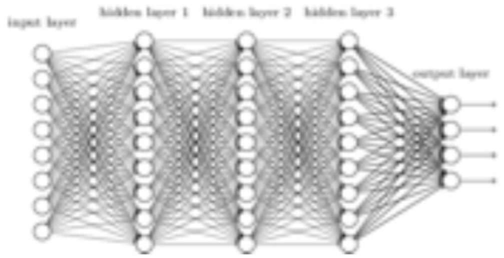
Expert Demonstrations

(*s*, *a*) pairs

Imitation Learning







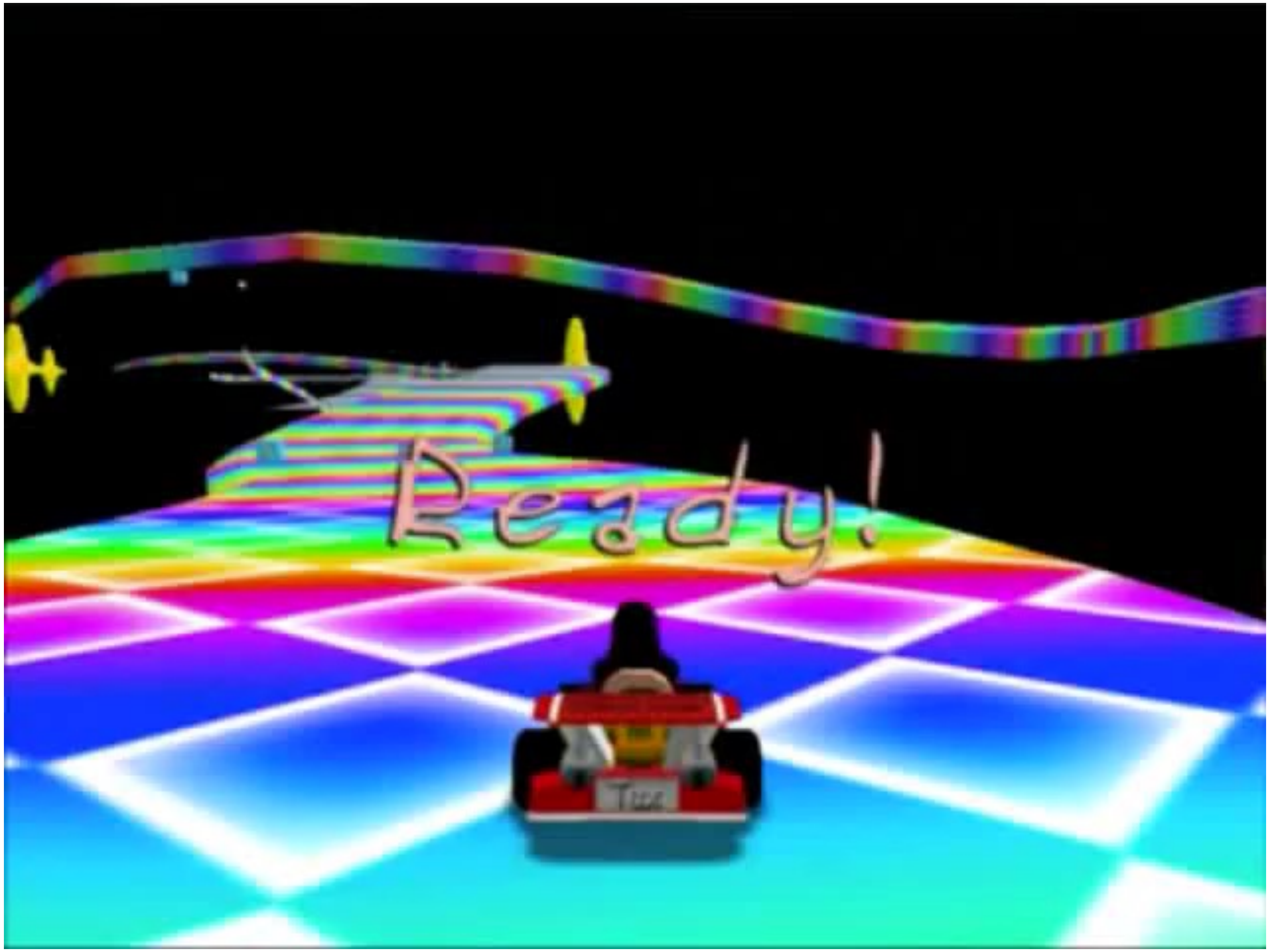
Agent Experience

(*s*, *a*, *r*, *s'*, *a'*) tuples

Reinforcement Learning

Images from Stephane Ross

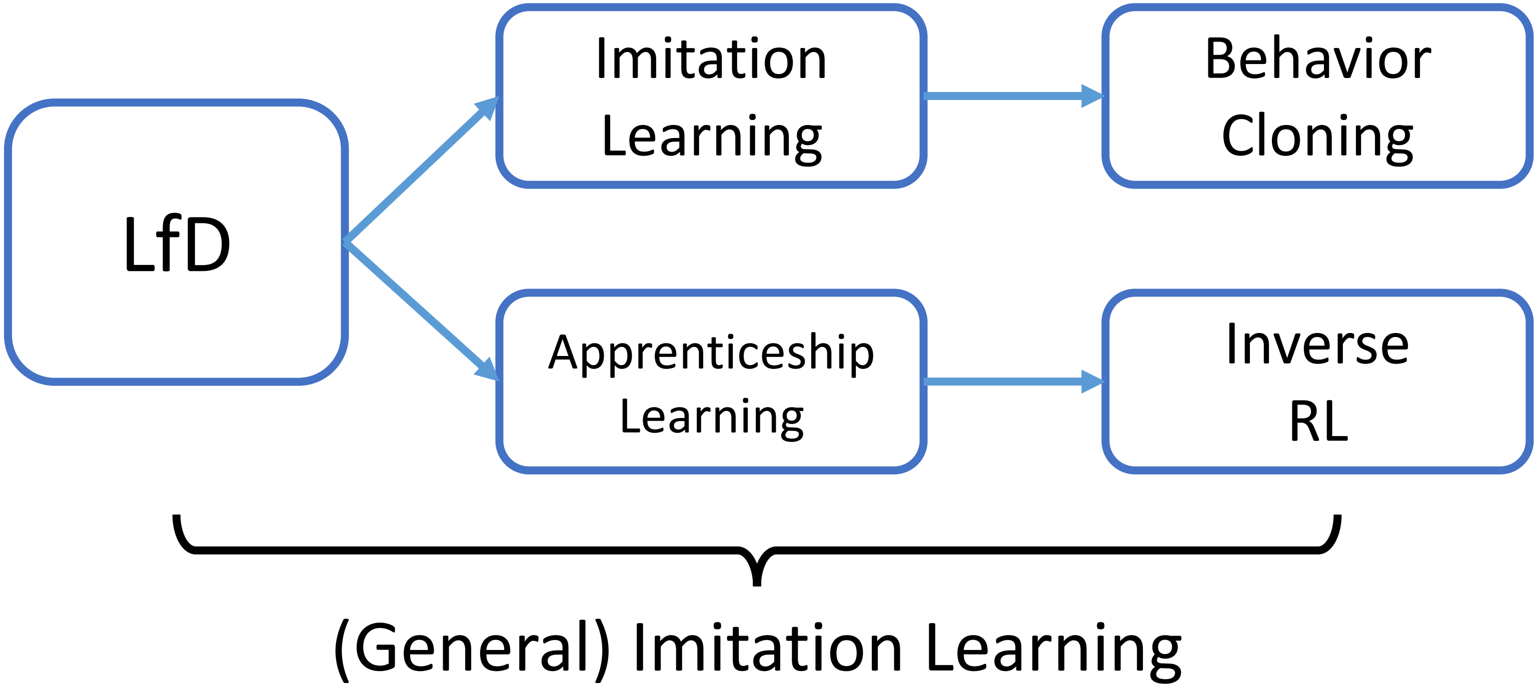
Imitation Learning Approaches on Super Tux Kart



https://www.youtube.com/watch?v=V00npNnWzSU

What is Imitation Learning

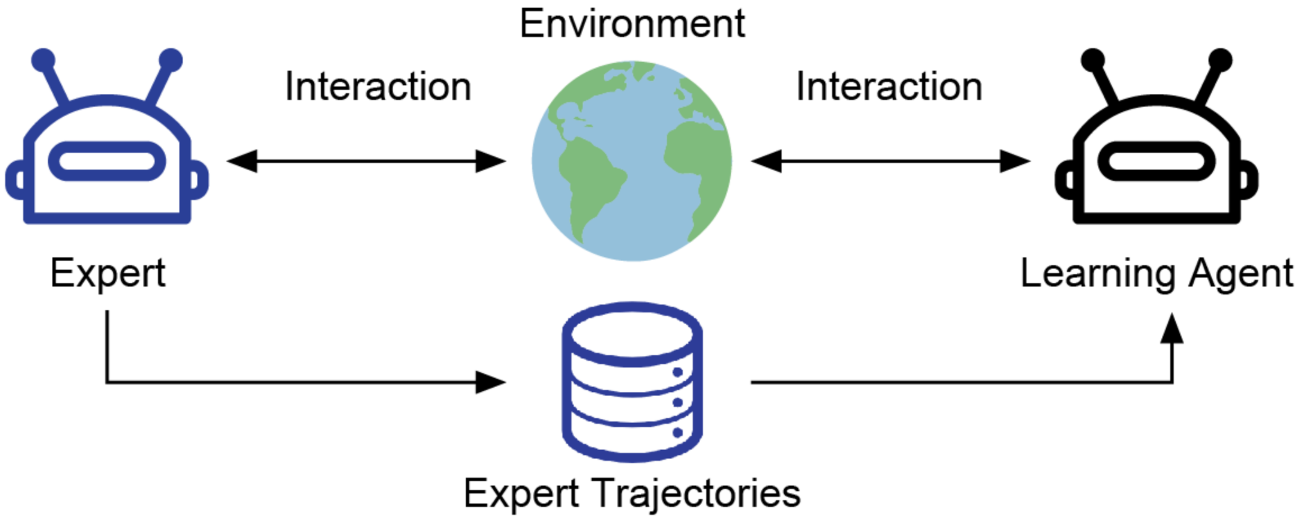
- Where from?
  - Learning from expert demonstration (LfD)
  - Try to imitate from the expert demonstrations.



What is Imitation Learning

General setting (in this talk):

- The learning agent
  1. can obtain pre-collected trajectories ((s,a) pairs) from uninteractive expert
  2. can interact with the environments (with simulators)
  3. cannot access reward signals.



What is Imitation Learning

Other optional settings

- No actions and only state / observations -> Imitation Learning From Observations (ILFO)
- With reward signals -> Imitation Learning with Rewards
- Interactive expert for correctness and data aggregation -> On-policy Imitation Learning (begin as Dagger, Dataset Aggregation)
- Cannot interact with Environments -> A special case of Batch RL (data in Batch RL can contain more than expert demos)

What is Imitation Learning

More Considerations

- Imitation loss
- Suboptimal demonstrations
- Partial demonstrations (e.g., weak feedback)
- Domain transfer (e.g., few-shot learning)
- Structured domains (e.g., multi-agent systems, structured prediction)

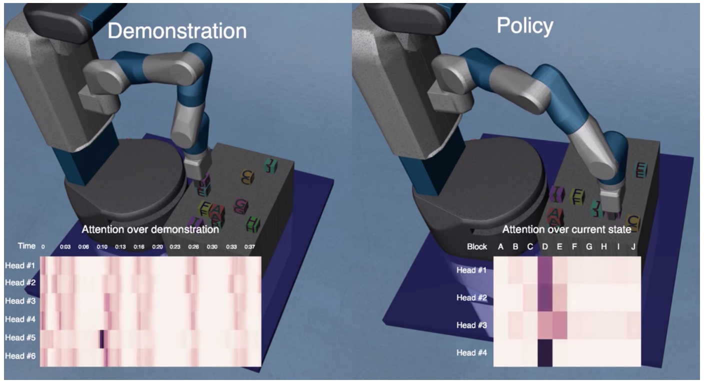
What is Imitation Learning

- When we necessarily want IL?
  - Hard to define the reward in some tasks
  - Hand-crafted rewards can lead to unwanted behavior

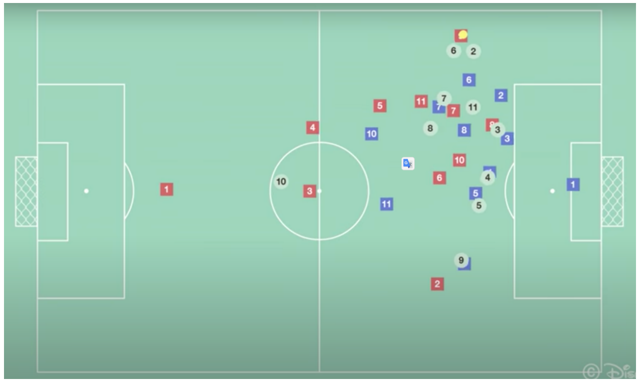
- What we want from IL?
  - Less interact with the **real-world** environments with expert demonstrations to improve sample efficiency and learn good policies
  - A fast and not bad policy initialization
  - A good solution that is robust to environment slight changes (compared to RL normally overfitting the env.)

Applications

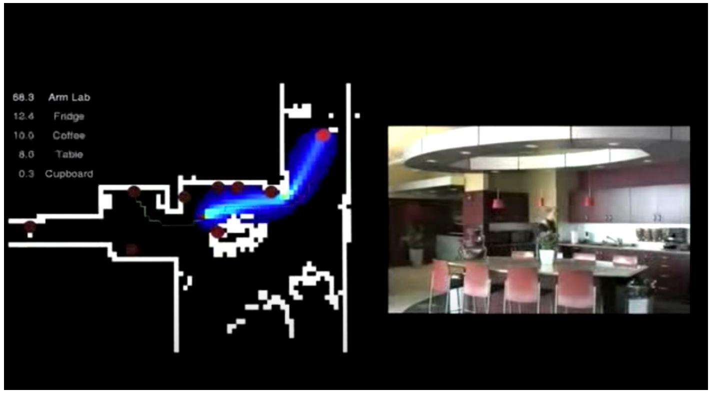
Helicopter Acrobatics [1,2]


Demonstration
Policy
Attention over demonstration
Time 0 0:03 0:06 0:10 0:13 0:16 0:20 0:23 0:26 0:30 0:33 0:37
Head #1
Head #2
Head #3
Head #4
Head #5
Head #6
Attention over current state
Block A B C D E F G H I J
Head #1
Head #2
Head #3
Head #4

Robotics Arm [3]



Sports Analysis [4]



Robotics Movement [5]

Formulation

- Notation & setup 1
  - State: s
    - State may only be partially observed, i.e., *o*
  - Action: a
  - Policy: π
    - Policy maps states to actions: $\pi(s) \rightarrow a$
    - or distributions over actions: $\pi(s) \rightarrow P(a)$
  - State transition dynamics: $P(s'|s, a)$
    - Typically not known to policy.
    - Essentially the simulator/environment

Formulation

- Notation & setup 2
    - Rollout: sequentially execute $\pi(s_0)$ on an initial state
        - Produce trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$
    - $P(\tau|\pi)$: distribution of trajectories induced by a policy
        - 1. Sample s_0 from ρ_0 (distribution over initial states), initialize $t = 0$.
        - 2. Sample action a_t from $\pi(s_t)$
        - 3. Sample next state s_{t+1} from applying a_t to s_t (requires access to environment)
        - 4. Repeat from Step 2 with $t = t + 1$

Formulation

- Notation & setup 3
  - $P(s|\pi) \rightarrow \rho_\pi(s)$: distribution of states induced by a policy

$$\rho_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t\, P(s_t = s|\pi)$$

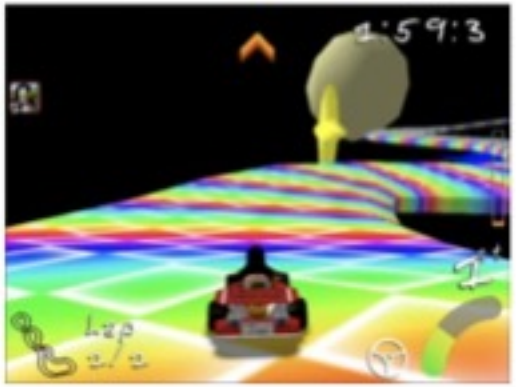
  - $P(s,a|\pi) \rightarrow \rho_\pi(s,a)$: distribution of state-action pairs induced by a policy (known as occupancy measure)

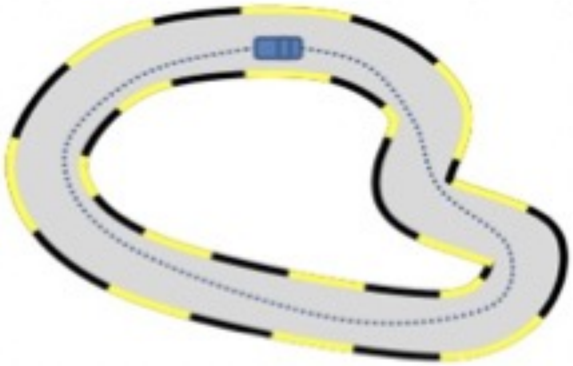
$$\rho_\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t\, P(s_t = s, a_t = a|\pi)$$

$$\rho_\pi(s,a) = \pi(a|s)\rho_\pi(s)$$

Formulation & Example 1

- Observation s = game screen
- Action a = turning angle
- Training set $D = \{\tau = [(s, a)]\}$ from an expert policy π*
- **Goal:** learn a good policy $\pi(s) \rightarrow a$ that achieves high value



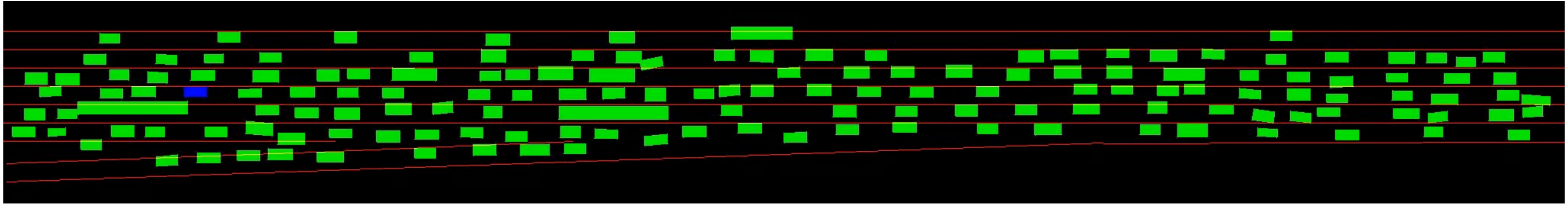


Formulation & Example 1

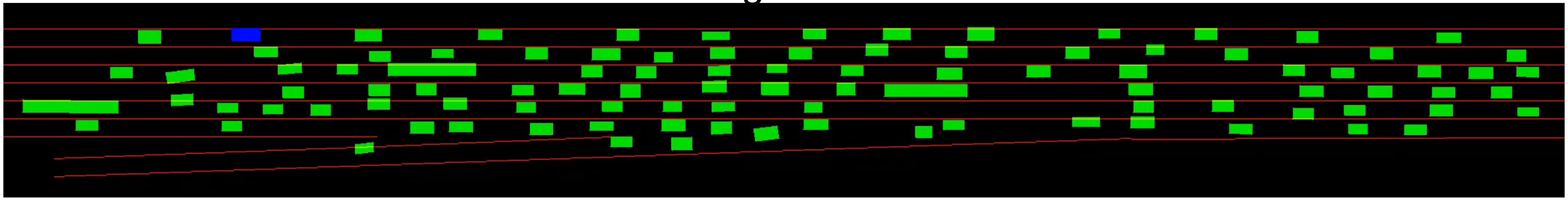
- NGSim dataset

🟥 : History Data Replay 🟦 : Policy Controlled Agent

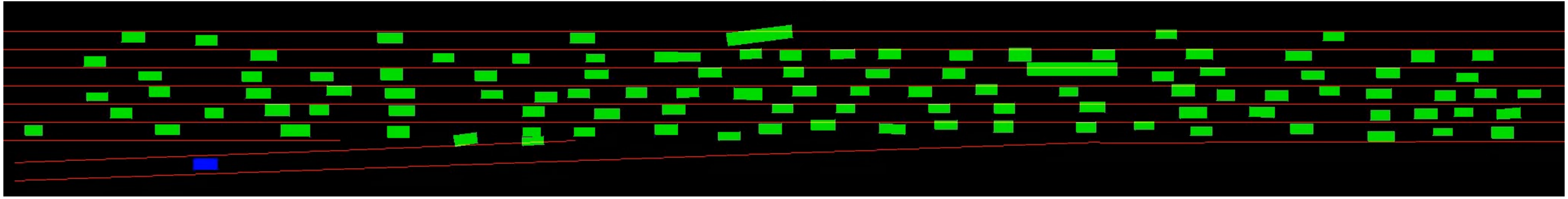
Heavy Traffic



Light Traffic

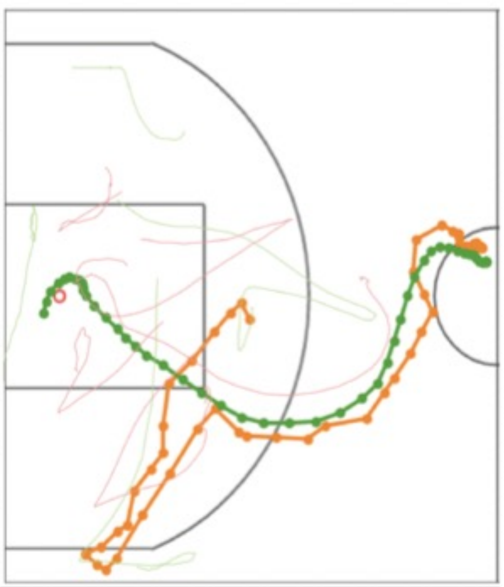


Ramp In



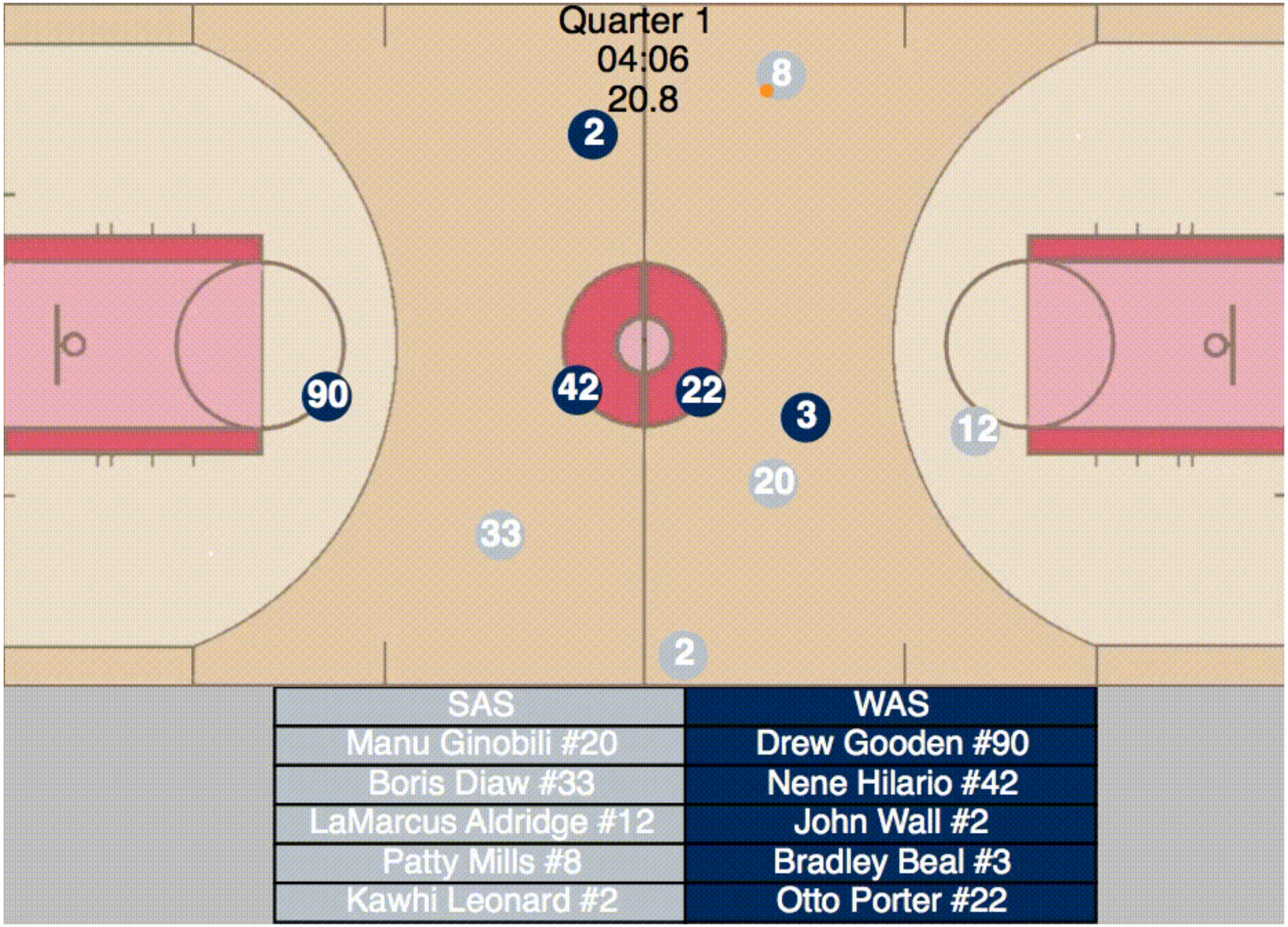
Formulation & Example 2

- Observation s = location of players & ball
- Action a = next location of player
- Training set $D = \{\tau = [(s, a)]\}$ from an expert policy π^*
- **Goal:** learn a good policy $\pi(s) \rightarrow a$ that achieves high value



Right Image from Yisong Yue

Formulation & Example 2

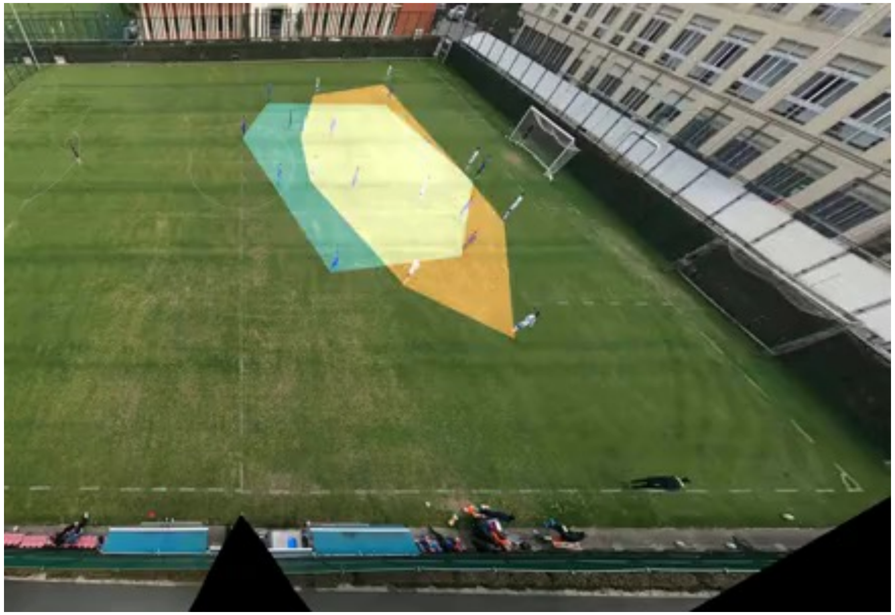


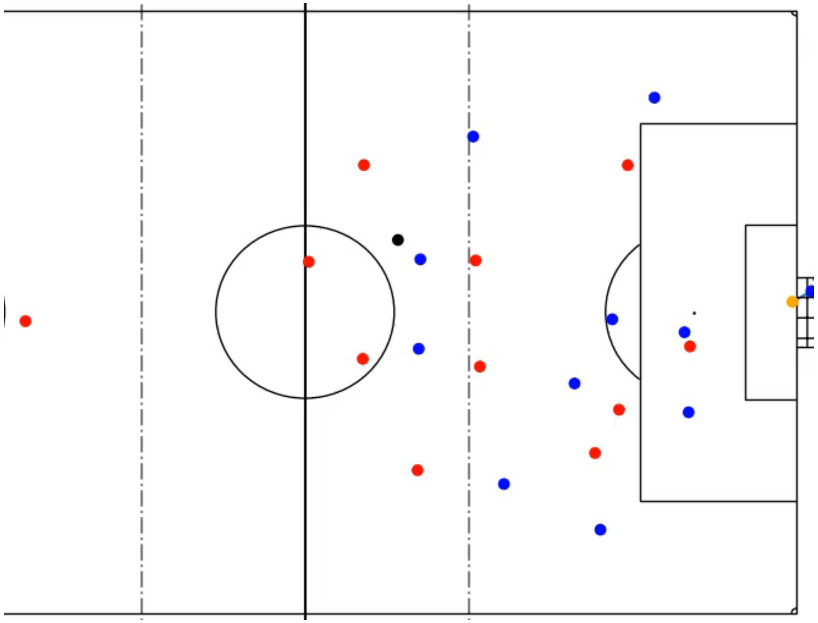
SAS	WAS
Manu Ginobili #20	Drew Gooden #90
Boris Diaw #33	Nene Hilario #42
LaMarcus Aldridge #12	John Wall #2
Patty Mills #8	Bradley Beal #3
Kawhi Leonard #2	Otto Porter #22

https://github.com/linouk23/NBA-Player-Movements

Formulation & Example 2

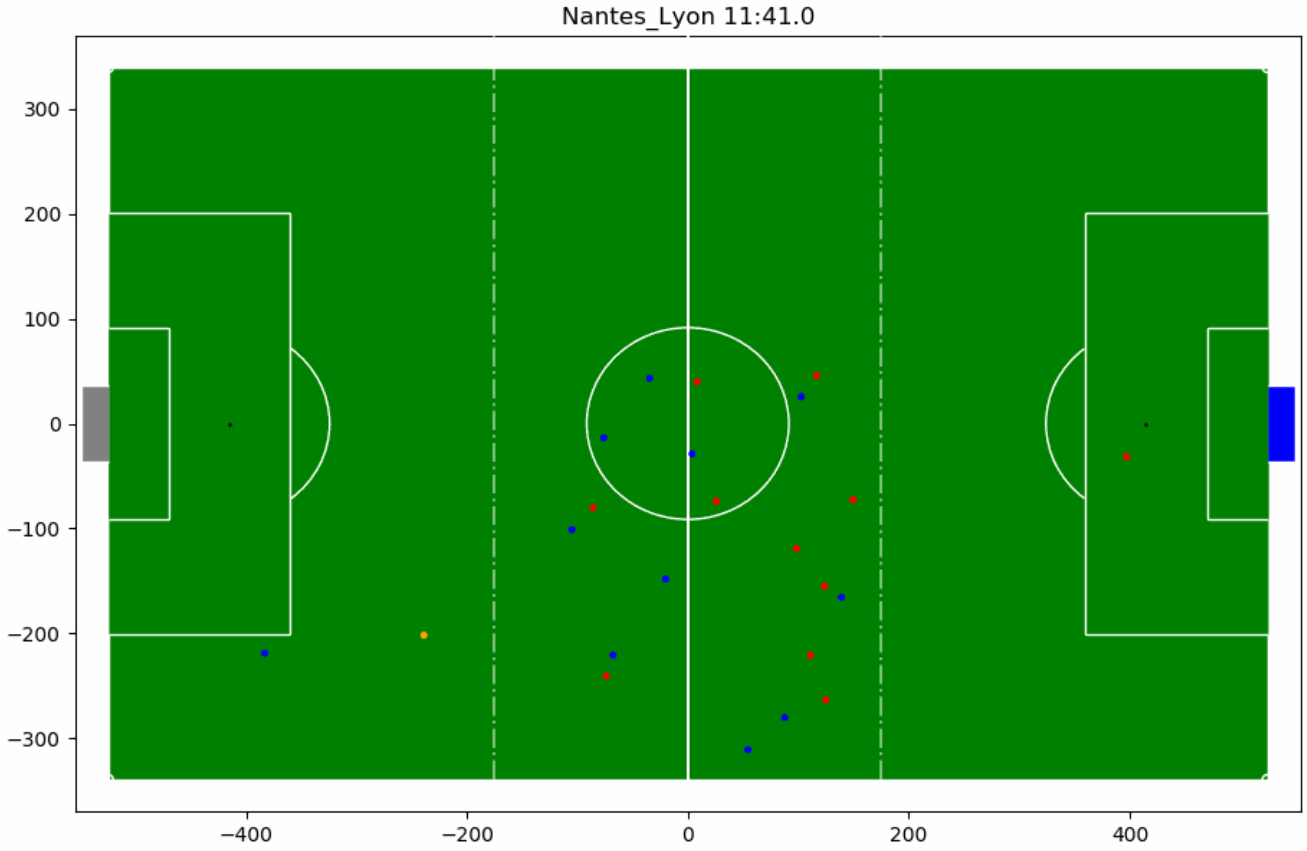
- Observation s = location of players & ball
- Action a = next location of player
- Training set $D = \{\tau = [(s, a)]\}$ from an expert policy π^*
- **Goal:** learn a good policy $\pi(s) \rightarrow a$ that achieves high value





Formulation & Example 2

- Nantes vs. Lyon. Red/Blue are real trajectory. Yellow is the generated



Overview

- Introduction to imitation learning
- Core methods of imitation learning
- Advanced methods of imitation learning
- Connection between imitation learning and GANs
- Recent applications of IL for robotics

Core Methods

- Behavior Cloning (BC)
  - Learn a direct mapping from states/contexts to trajectories/actions without recovering the reward function
- Inverse Reinforcement Learning (IRL)
  - Finds a reward function which makes expert trajectories better than others.
- Generative Adversarial Imitation Learning (GAIL)
  - Apply GAN under the structure of IRL to make close the two occupancy measures between the expert and the agent

General Imitation Learning

- Objective

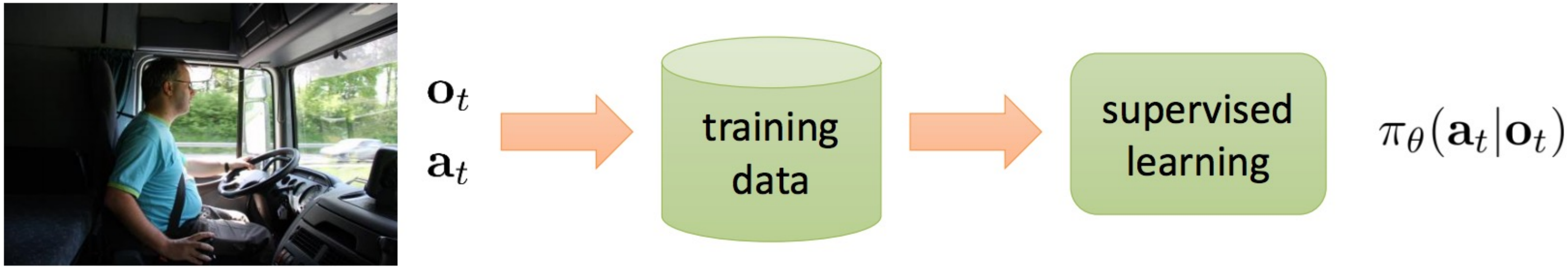
$$\pi^* = \arg\min_{\pi} \mathbb{E}_{s\sim\rho_\pi^s}\left[\ell\left(\pi(\cdot|s), \pi_E(\cdot|s)\right)\right]$$

  - l denotes some loss function or some distance metric.
  - Distribution of s depends on rollout from π.
    - $P(s|\pi) \rightarrow \rho_\pi(s)$: distribution of states sampled by a policy

$$\rho_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\, P(s_t = s|\pi)$$

- Problem
  - Cannot get access to the expert during sampling!

Behavioral Cloning



- Learning objective of BC

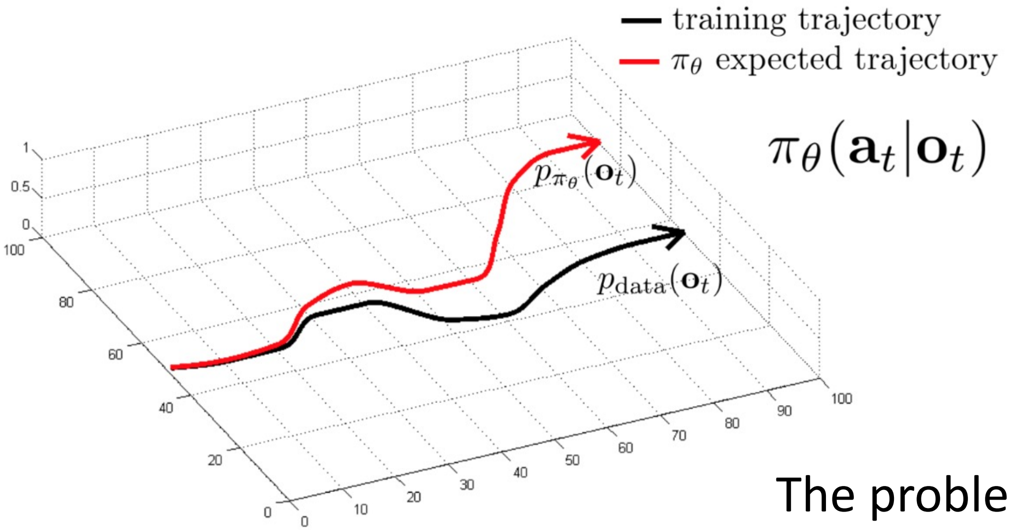
$$\hat{\pi}^* = \arg\min_{\pi} \mathbb{E}_{s\sim\rho^s_{\pi_E}} \left[\ell\left(\pi_E(\cdot|s), \pi(\cdot|s)\right)\right]$$

- Compared with the original objective

$$\pi^* = \arg\min_{\pi} \mathbb{E}_{s\sim\rho^s_{\pi}} \left[\ell\left(\pi(\cdot|s), \pi_E(\cdot|s)\right)\right]$$

- Distribution provided exogenously
- Essentially a Maximum Likelihood Estimation (MLE) on single step

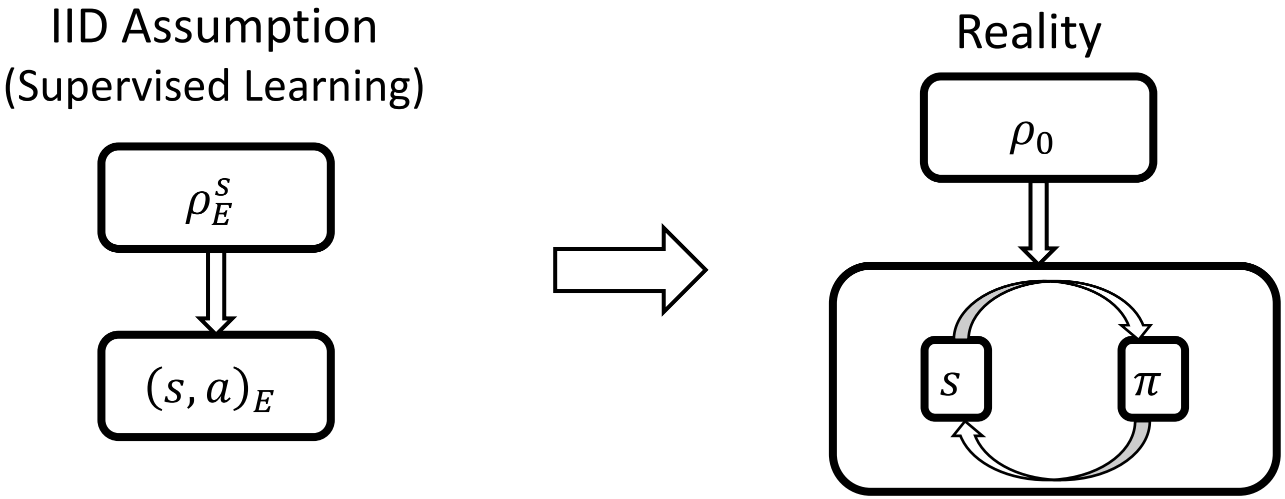
Limitations of Behavioral Cloning



$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$$

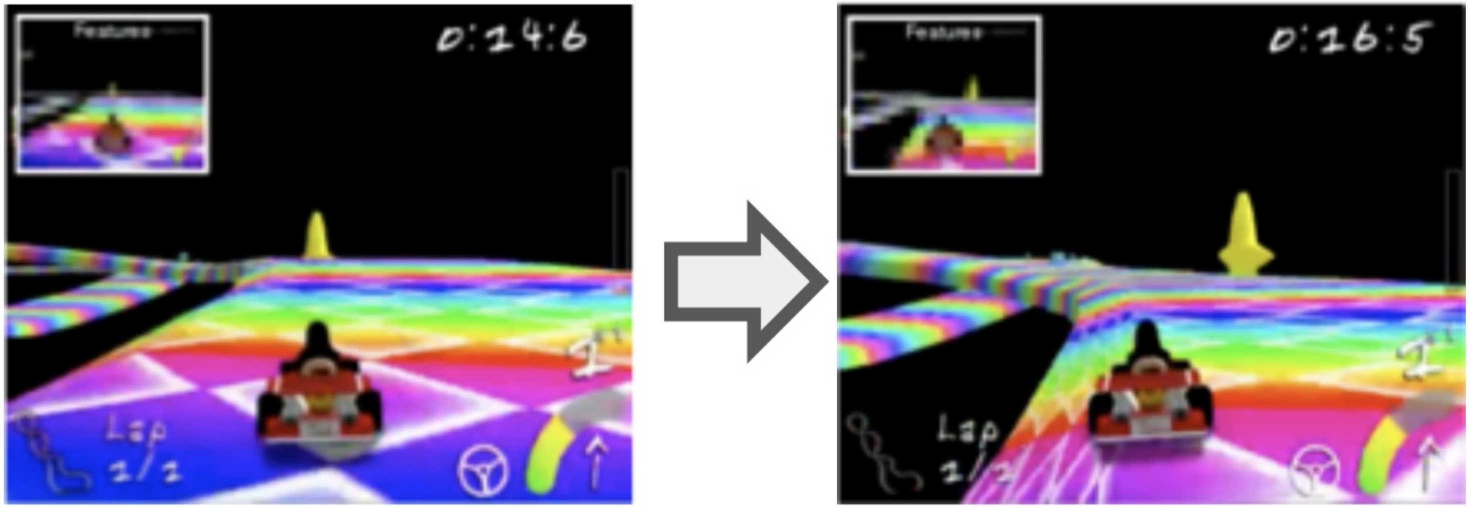
Distributional Shift

The problem is like a common problem in supervised learning, but more serious


IID Assumption
(Supervised Learning)

Reality

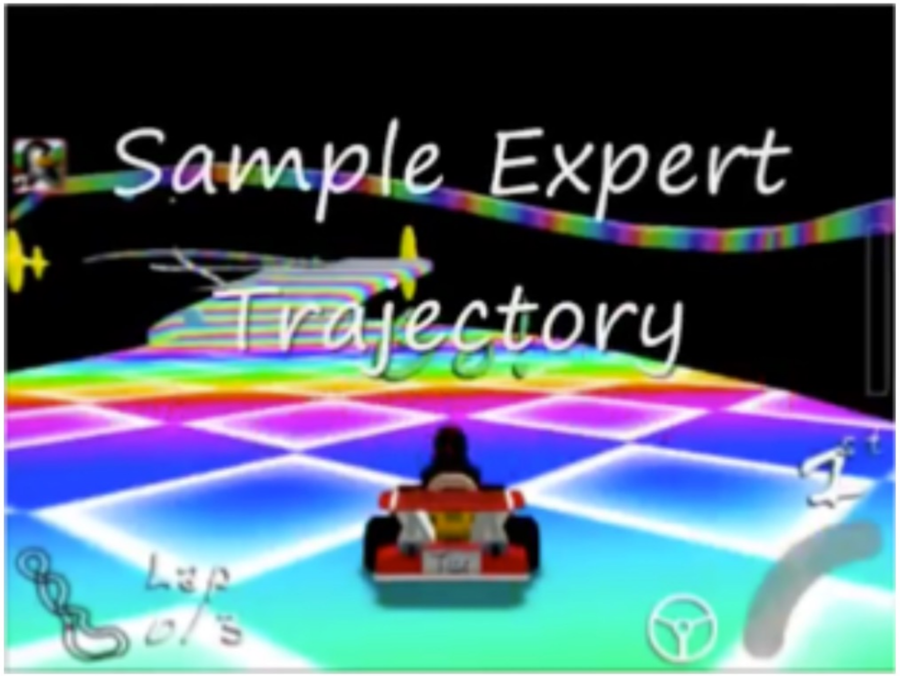
Limitations of Behavioral Cloning



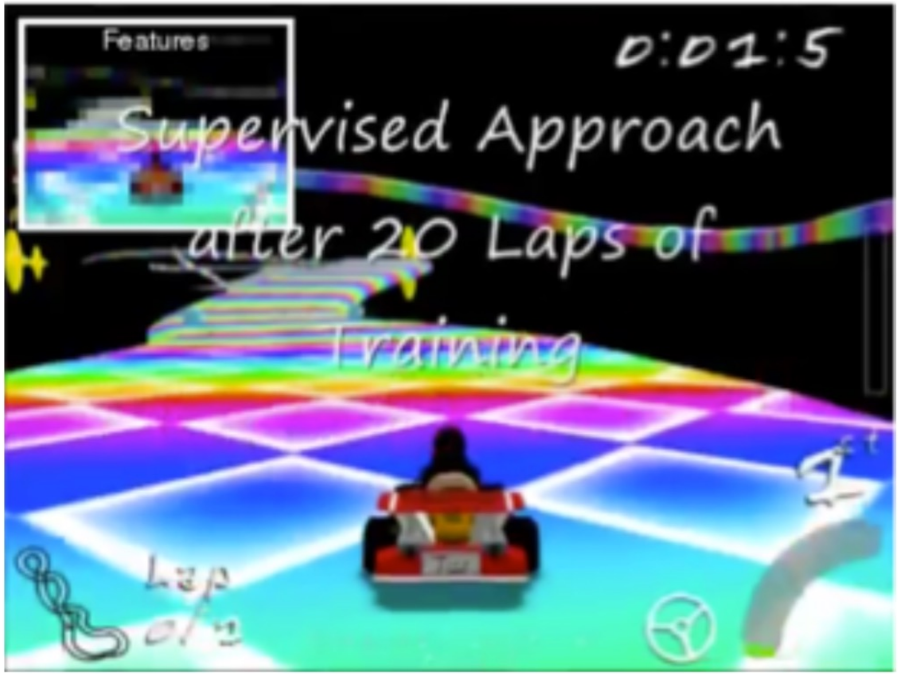
When π_θ makes a mistake, i.e., starts to diverge from the expert

- New state sampled not from ρ_E^S!
- Worst case is catastrophic!

Limitations of Behavioral Cloning



Expert Trajectories
(Training Distribution)



Behavioral Cloning

Makes mistakes, enters new states

Cannot recover from new states

Images from Stephane Ross

Imitation Learning vs. Supervised Learning

- The solution may have important structural properties including constraints (for example, robot joint limits), dynamic smoothness and stability, or leading to a coherent, multi-step plan
- The interaction between the learner's decisions and its own input distribution (an on-policy versus off-policy distinction)
- Along side of the policy similarity, IL further cares about the policy performance

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

When to use BC?

- Advantages
  - Simple
  - Efficient

- When to use
  - 1-step deviations not too bad
  - Learning reactive behaviors
  - Expert trajectories “cover” state space

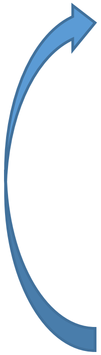
- Disadvantages
  - State distribution mismatch between training and test
  - No long-term planning

- When not to use
  - 1-step deviations can lead to catastrophic error
  - Optimizing long-term objective

Slide credit to Yisong Yue

BC with Dataset Aggregation

- Samples from a stable trajectory distribution
  - Learning from a stabilizing controller (with noise)
- Add more on-policy data
  - e.g. DAgger
- DAgger: Dataset Aggregation

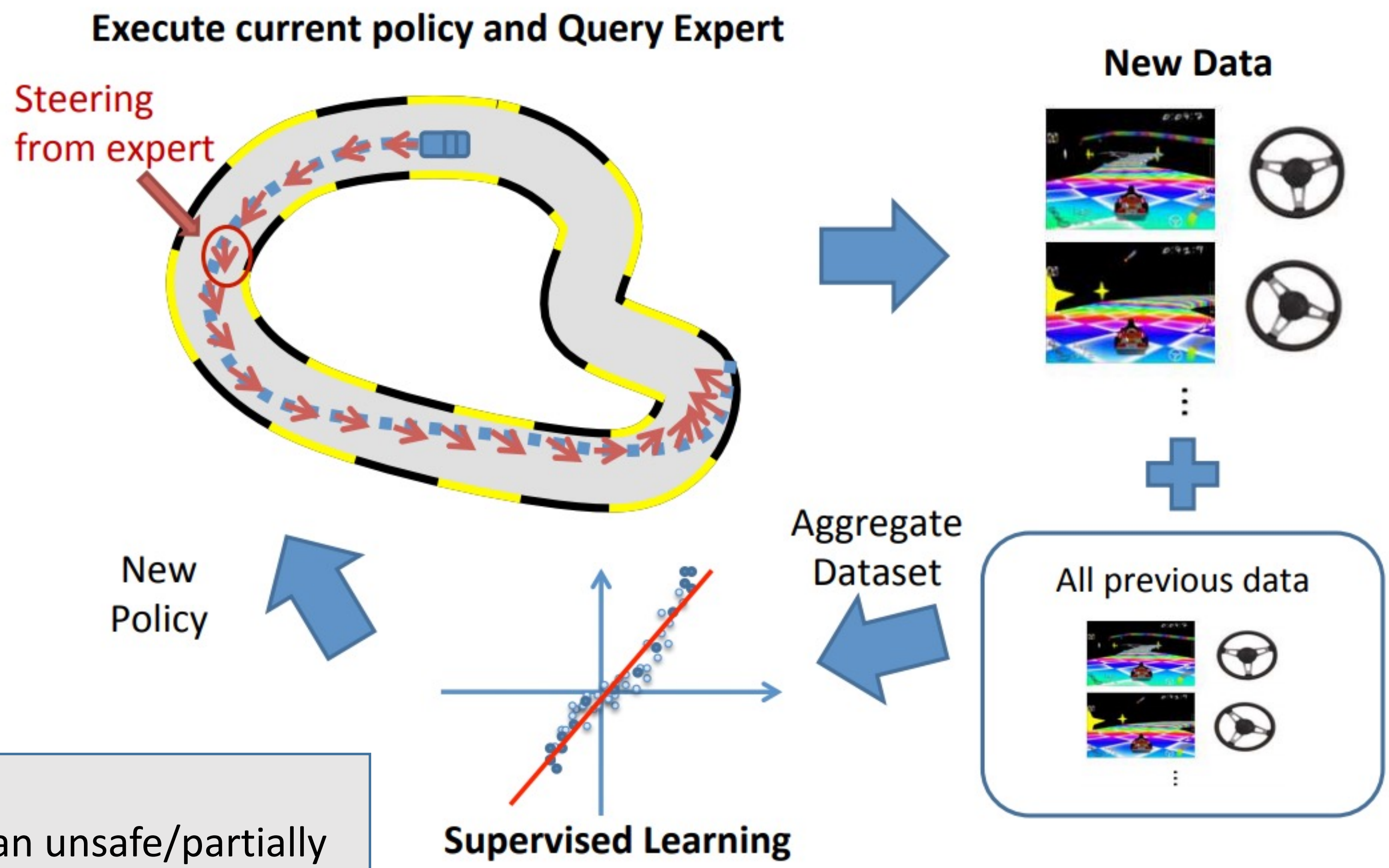


  - train $\pi_\theta(a_t, o_t)$ from a human data $\mathcal{D} = \{o_1, a_1, \dots, o_N, a_N\}$.
  - run $\pi_\theta(a_t|o_t)$ to get dataset $\mathcal{D}_\pi = \{o_1, a_1, \dots, o_N, a_N\}$.
  - Ask human to label states in $\mathcal{D}_\pi$ with action a_t.
  - Aggregate $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Human-in-the-loop IL

Stéphane Ross et al. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. AISTATS 2011.

Illustration of DAgger



Problems:

- execute an unsafe/partially trained policy
- repeatedly query the expert

Inverse Reinforcement Learning

"Forward" RL

- Given:
  - State $s \in \mathcal{S}, a \in \mathcal{A}$
  - (sometimes) transitions $p(s'|s,a)$
  - Reward function $r(s,a)$
- Learn $\pi^*(a|s)$

Inverse RL

- Given:
  - State $s \in \mathcal{S}, a \in \mathcal{A}$
  - (sometimes) transitions $p(s'|s,a)$
  - Samples $\{\tau_i\}$ sampled from $\pi^*(\tau_i)$
- Learn $r_\psi(s,a)$
- Then use it to learn $\pi^*(a|s)$

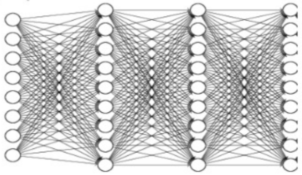
Linear reward function

$$r_\psi(s,a) = \sum_i \psi_i f_i = \psi^T f(s,a)$$

Neural net reward function

s

a



$r_\psi(s,a)$

Parameters ψ

Inverse Reinforcement Learning

- Objective

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s,a)\sim\rho_\pi}\left[r^*(s,a)\right]$$

  - looks for a reward function r^* under which the expert policy is the optimal solution.

- Need to recover r^*
  - Principle: expert is optimal, i.e., find *r** such that

$$r^* = \arg\max_{r} \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t r(s,a)|\pi^*\Big] - \mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t r(s,a)|\pi\Big]$$

- Usually with a bi-level optimization
  - Outer loop: find *r*
    - Inner loop: train policy π with *r*
    - Check whether $V(\pi^*) - V(\pi)$ is minimized

This is ambiguous and the solution of *r* may not be unique.

Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning. ICML 2004.

Inverse Reinforcement Learning

- A formulation of max-entropy IRL

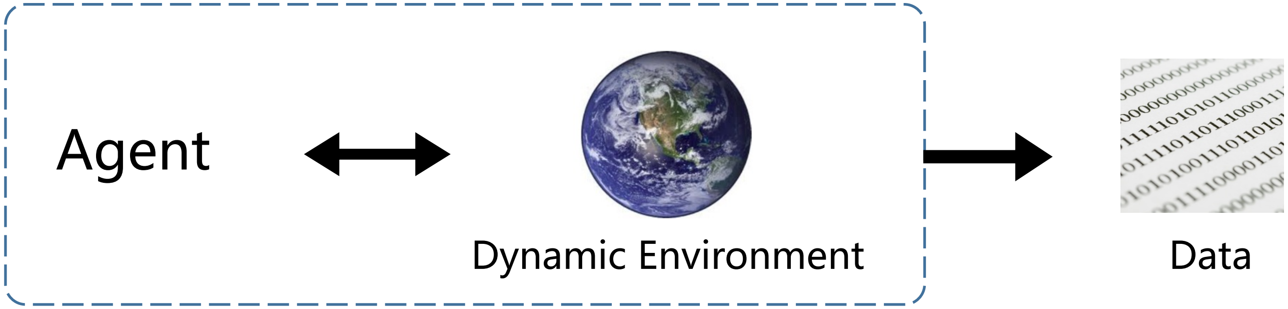
$$\tilde{c} = IRL(\pi_E) = \arg\max_{c} \left\{ (\min_{\pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)]) - \mathbb{E}_{\pi_E}[c(s,a)] \right\}$$

- Looks for a cost function $\tilde{c} \in C$ that assigns low cost to the expert policy and high cost to other policies while maximizing the entropy of the policy.
- Recover the expert policy by running RL under the learned cost function $\tilde{c}$:

$$\tilde{\pi} = RL(\tilde{c}) = \arg\min_{\pi \in \Pi} - H(\pi) + E_{\pi}[\tilde{c}(s,a)]$$

- Limitations: most of them do not gain the policy directly. Bi-level optimization is usually expensive to run, especially in high-dimensional and continuous space.

Generative Adversarial Imitation Learning



Agent

Dynamic Environment

Data

- Review the occupancy measure of each policy interacting with the environment

$$\rho^{\pi}(s,a) = (1-\gamma)\mathbb{E}_{a\sim\pi(s),\, s'\sim p(s,a)}\left[\sum_{t=0}^{T} \gamma^{t}\, p(s_t = s, a_t = a)\right]$$

- Theorem 1: for two policies π_1, π_2 and their occupancy measures $\rho^{\pi_1}, \rho^{\pi_2}$, it has $\rho^{\pi_1} = \rho^{\pi_2}$ iff $\pi_1 = \pi_2$
- Theorem 2: given an occupancy measure ρ, the only policy generating ρ is $\pi_\rho = \rho(s,a)/\sum_{a'}\rho(s,a')$

Ho J, Ermon S. Generative adversarial imitation learning. NIPS 2016.

Generative Adversarial Imitation Learning

- GAIL: match the occupancy measures with GAN

$$\min_{\pi}\max_{D}\mathbb{E}_{\pi_E}[\log D(s,a)]+\mathbb{E}_{\pi}[\log(1-D(s,a))]-\lambda H(\pi)$$

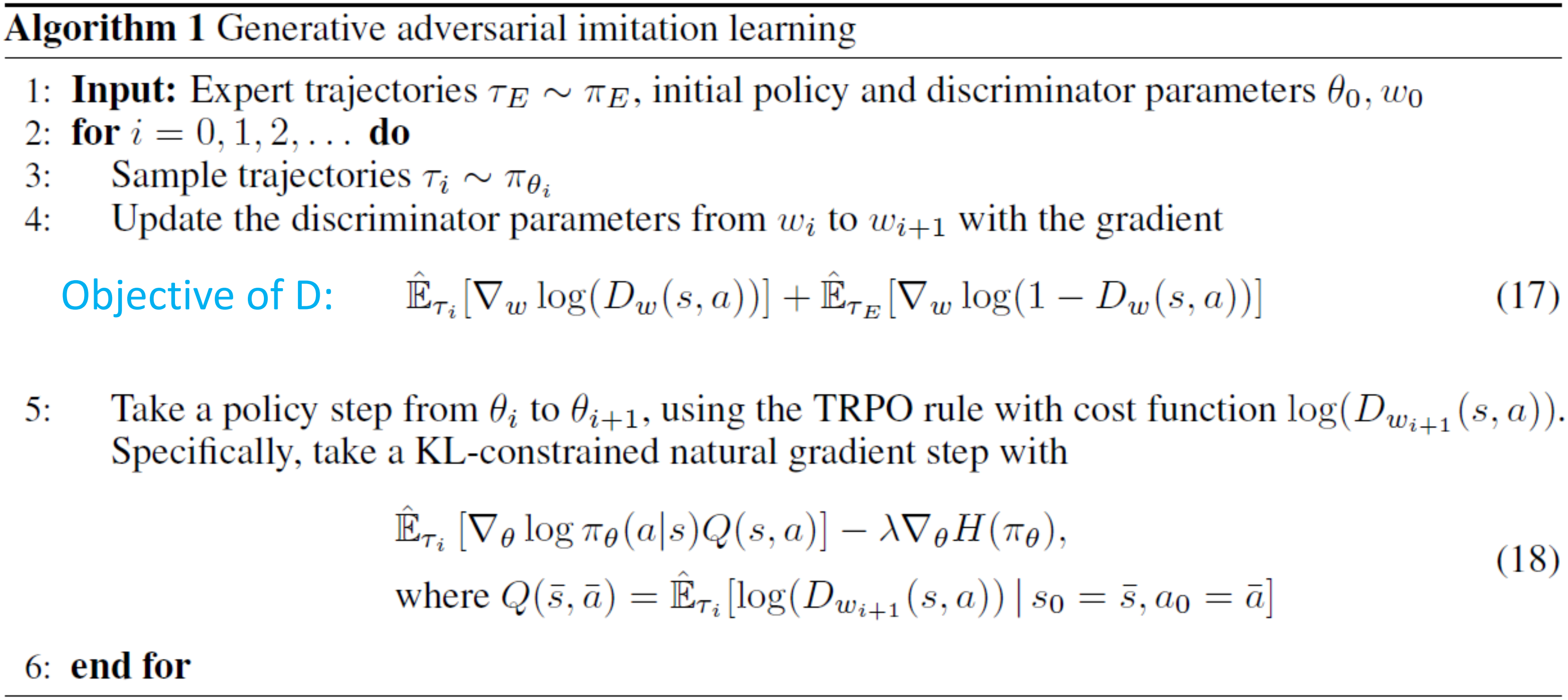
- GAN

$$\min_{G}\max_{D}\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)]+\mathbb{E}_{x\sim G}[\log(1-D(x))]$$

- Occupancy measure is analogous to the data distribution
- Discriminator *D* distinguishes between the distribution of data generated by *G* (π in GAIL) and the true data distribution (π_E in GAIL)

Ho J, Ermon S. Generative adversarial imitation learning. NIPS 2016.

GAIL Algorithm



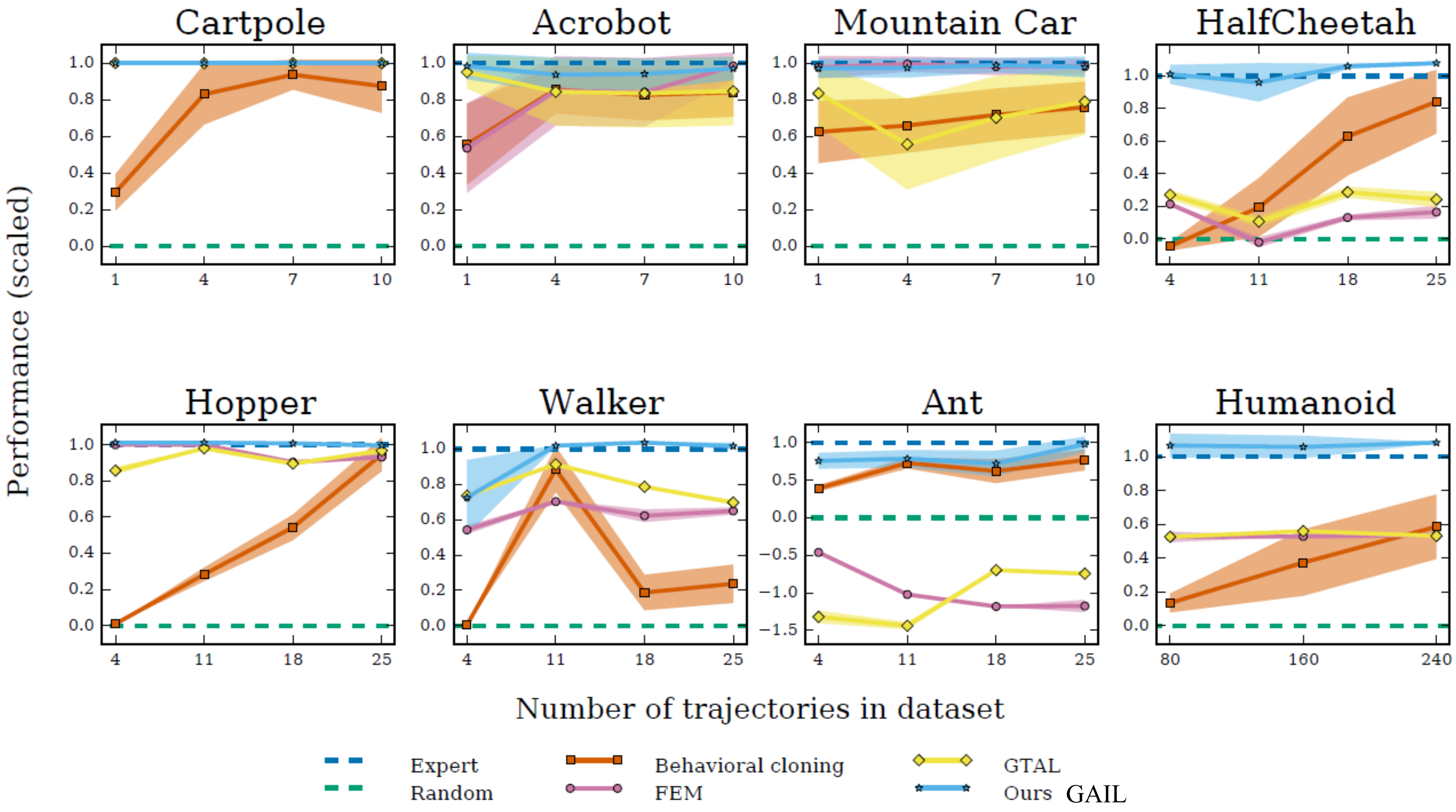
Algorithm 1 Generative adversarial imitation learning

```
Input: Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters $\theta_0, w_0$
for $i = 0, 1, 2, \dots$ do
    Sample trajectories $\tau_i \sim \pi_{\theta_i}$
    Update the discriminator parameters from $w_i$ to $w_{i+1}$ with the gradient
        Objective of D:  $\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))]$   (17)
    Take a policy step from $\theta_i$ to $\theta_{i+1}$, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$.
    Specifically, take a KL-constrained natural gradient step with
        $\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta),$
        where $Q(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}]$   (18)
end for
```

- GAIL alternates between
  - An Adam gradient step of w to increase the objective of D
  - A TRPO step of θ to decrease the objective of D

Ho J, Ermon S. Generative adversarial imitation learning. NIPS 2016.

GAIL Experiments



Ho J, Ermon S. Generative adversarial imitation learning. NIPS 2016.

Overview

- Introduction to imitation learning
- Core methods of imitation learning
- Advanced methods of imitation learning
- Connection between imitation learning and GANs
- Recent applications of IL for robotics

Recent Works

- One-pass IL methods with fixed reward function
    - First estimate the reward then apply a forward RL procedure
    - Set the reward with specific intention
- Soft Q imitation Learning (SQIL)
- Random Expert Distillation (RED)
- Disagreement-Regularized Imitation Learning (DRIL)
- Energy-Based Imitation Learning (EBIL)

Soft Q Imitation Learning (SQIL)

- Reward definition
  - Expert data $r(s^*, a^*) = 1$
  - New interaction data $r(s, a) = 0$

- Off-policy learning
  - A replay buffer initialized with expert data
  - Then add new interaction data (50% each)
  - RL algorithm: Soft Q-Learning (or Soft Actor-Critic)

$$\delta^2(\mathcal{D}, r) \triangleq \frac{1}{|\mathcal{D}|} \sum_{(s,a,s') \in \mathcal{D}} \left(Q_{\boldsymbol{\theta}}(s, a) - \left(r + \gamma \log \left(\sum_{a' \in \mathcal{A}} \exp\left(Q_{\boldsymbol{\theta}}(s', a')\right) \right) \right) \right)^2$$

Reddy S, Dragan A D, Levine S. SQIL: imitation learning via regularized behavioral cloning. ICLR 2020.

Random Expert Distillation (RED)

- Random Network Distillation (RND) for exploration
  - Fit a randomly initialized neural network $f_\theta(s, a)$
  - Use the MSE prediction error as the intrinsic reward

$$\|f_{\hat{\theta}}(s, a) - f_\theta(s, a)\|^2$$

- Similar idea for imitation learning

$$\hat{\theta} = \arg\min_{\theta'} \|f_{\theta'}(s, a) - f_\theta(s, a)\|^2 \mid \mathcal{D}$$

$$r(\cdot) = \exp(-\sigma \|f_{\hat{\theta}}(s, a) - f_\theta(s, a)\|^2)$$

Reward is high on *familiar* state-actions of expert

- Then run an RL algorithm with the reward function

Burda, Yuri, et al. Exploration by random network distillation. 2018.

Wang R, et al. Random expert distillation: Imitation learning via expert policy support estimation. ICML 2019.

Disagreement-Regularized IL (DRIL)

- Motivation
  - Policy should move towards the expert data distribution if it is away from it
- How to train the policy?
  - Variance (uncertainty) minimization $C_U(s,a) = \mathrm{Var}_{\pi \sim p(\pi|\mathcal{D})}(\pi(a|s))$
  - Minimizing variance encourages the policy to return to regions of dense coverage by the expert, where the variance is low
- How to estimate the variance?
  - Ensemble (bagging) of policies
  - The disagreement in imitation serves as the prediction variance
- Clipped reward definition $C_U^{\mathrm{clip}}(s,a) = \begin{cases} -1 & \text{if } C_U(s,a) \leq q \\ +1 & \text{otherwise} \end{cases}$

Brantley K et al. Disagreement-Regularized Imitation Learning. ICLR 2020.

Energy-Based imitation Learning (EBIL)

- Motivation
  - Estimate the energy (can be regarded as an unnormalized density) of expert's occupancy measure and use it as the surrogate reward to run an RL algorithm.

$$\rho_\pi(s,a) = \frac{1}{Z}\exp(-E(s,a))$$

$$\pi^* = \arg\max_\pi \mathbb{E}_\pi\left[-E_{\pi_E}(s,a)\right] + H(\pi)\ .$$

  - This can be regarded as minimizing the KL-divergence:

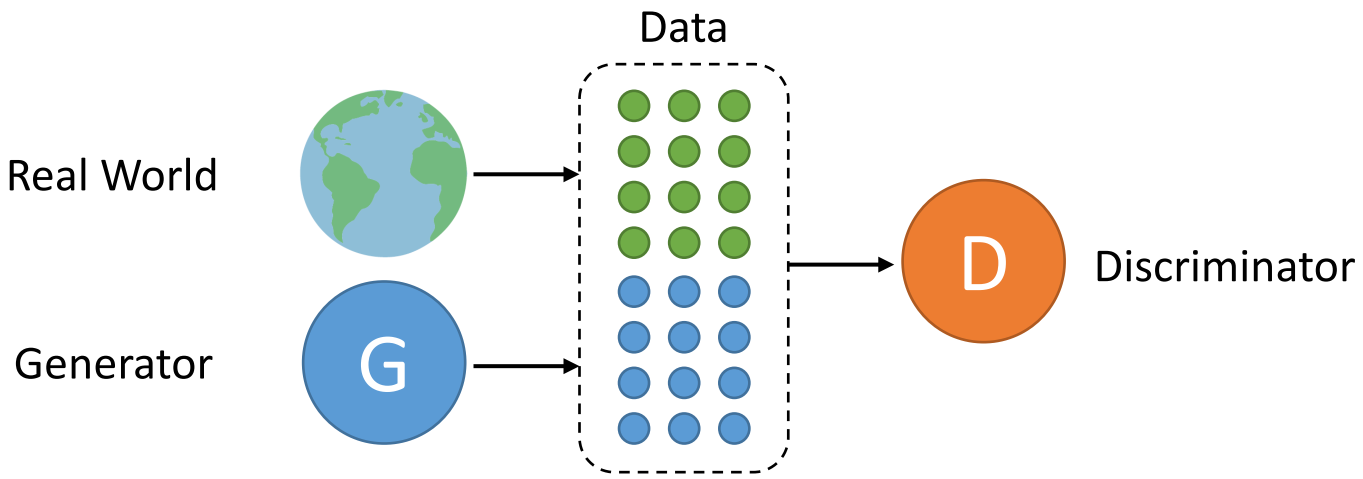
$$\pi^* = \arg\min_\pi \mathrm{D}_{\mathrm{KL}}(\rho_\pi \| \rho_{\pi_E})$$

Liu M et al. Energy-Based Imitation Learning. AAMAS 2021.

Overview

- Introduction to imitation learning
- Core methods of imitation learning
- Advanced methods of imitation learning
- Connection between imitation learning and GANs
- Recent applications of IL for robotics

GAN: A Minimax Game



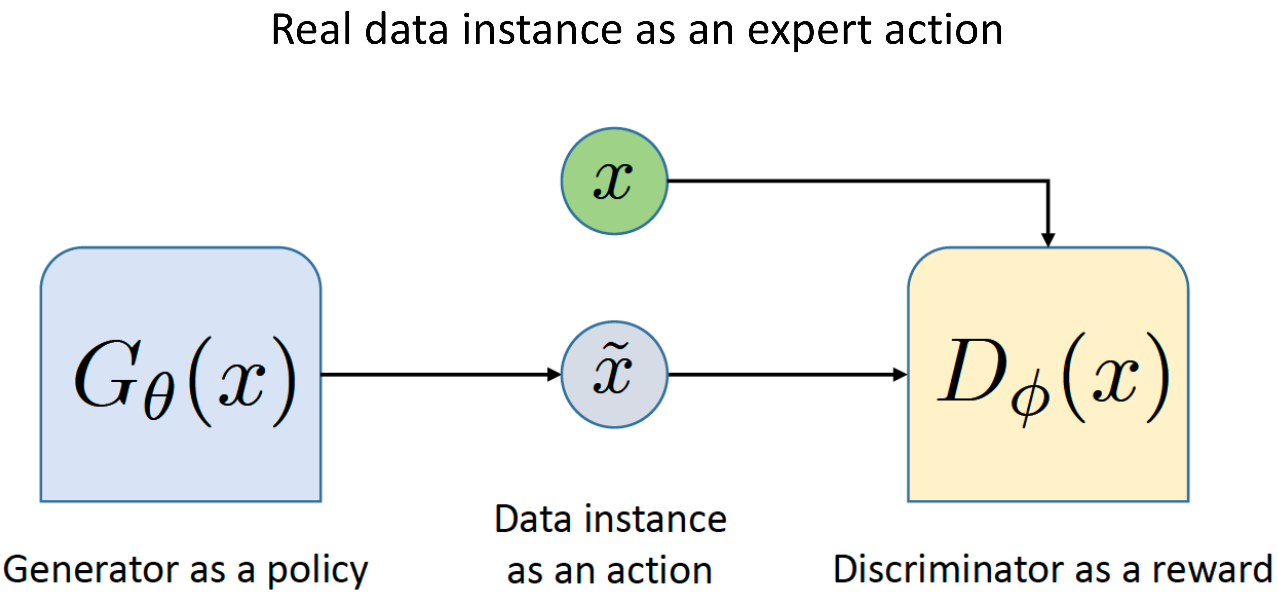
$$\min_{G} \max_{D} J(G; D) \qquad \max_{D} J(G; D)$$

The joint objective function

$$J(G, D) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

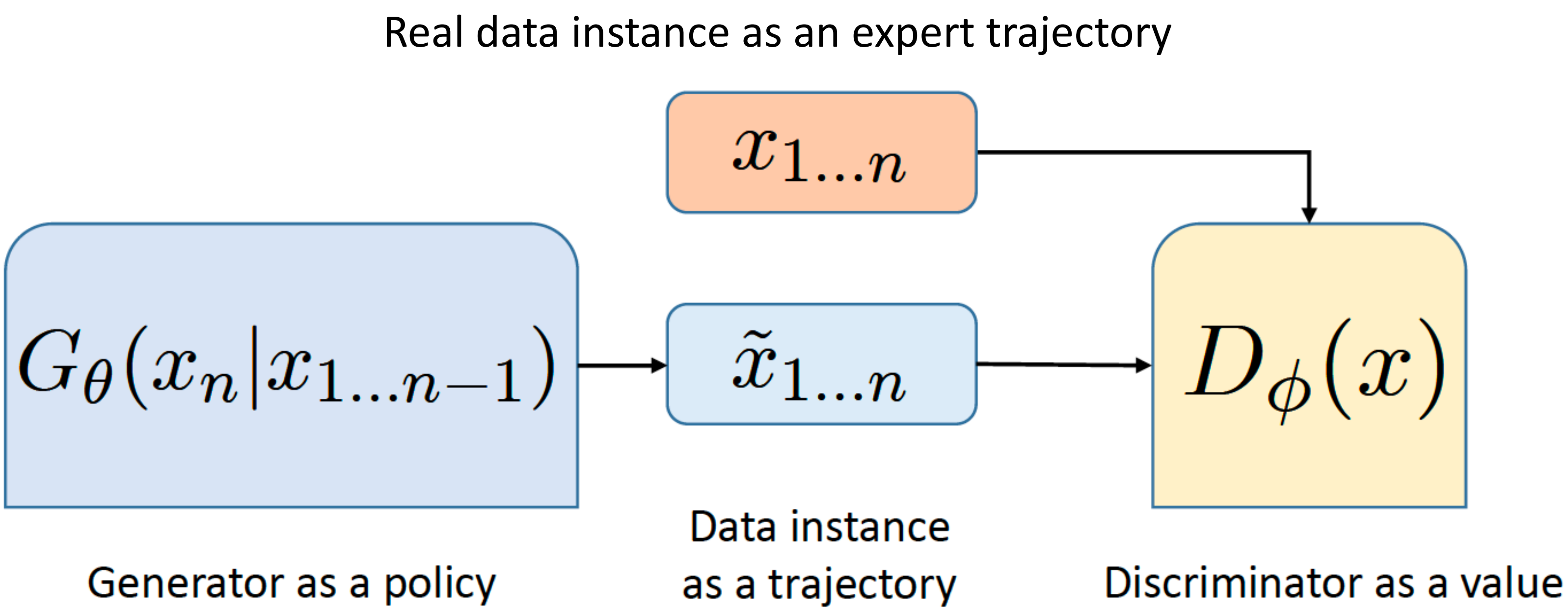
Connection between GAN and IL

- Analogy to Imitation learning
  - In imitation learning, a value function is learned from expert data to guide the policy optimization
  - In GAN, a discriminator is trained with real (positive) data and generated (negative) data to guide the generator optimization
- One step generation: stateless or one-step MDP


Real data instance as an expert action
x
$G_\theta(x)$
$\tilde{x}$
$D_\phi(x)$
Generator as a policy
Data instance as an action
Discriminator as a reward

Connection between GAN and IL

- Analogy to Imitation learning
  - In imitation learning, a value function is learned from expert data to guide the policy optimization
  - In GAN, a discriminator is trained with real (positive) data and generated (negative) data to guide the generator optimization
- Multi-step generation: MDP


Real data instance as an expert trajectory
$x_{1\dots n}$
$G_\theta(x_n|x_{1\dots n-1})$
$\tilde{x}_{1\dots n}$
$D_\phi(x)$
Generator as a policy
Data instance as a trajectory
Discriminator as a value

GAN and RL on Learning Rules

For continuous data/action

- Deterministic policy gradient (DPG)

$$\frac{\partial J(\pi_\theta)}{\partial \theta} = \mathbb{E}_{s\sim\rho^\pi}\left[\frac{\partial Q^\pi(s,a)}{\partial a}\frac{\partial \pi_\theta(s)}{\partial \theta}\Big|_{a=\pi_\theta(s)}\right]$$

- GAN for continuous data

$$\frac{\partial J(G_\theta, D)}{\partial \theta} = \mathbb{E}_{z\sim p(z)}\left[\frac{\partial J(G_\theta, D(x))}{\partial x}\frac{\partial G_\theta(z)}{\partial \theta}\Big|_{x=G_\theta(z)}\right]$$

For discrete data/action

- Stochastic policy gradient (PG)

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\pi_\theta}\left[\frac{\partial \log \pi_\theta(a|s)}{\partial \theta} Q^{\pi_\theta}(s,a)\right]$$

- GAN for discrete data

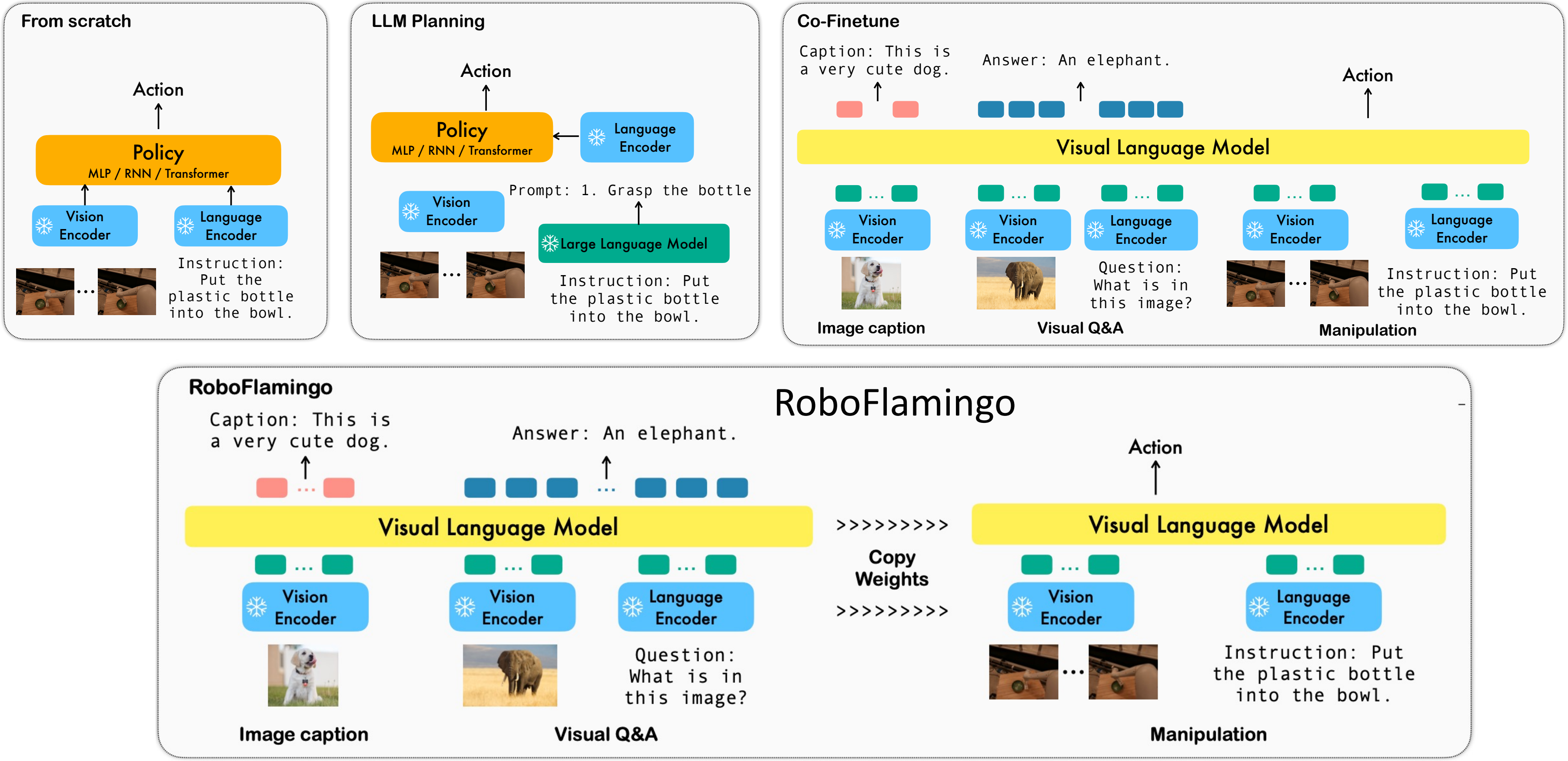
$$\frac{\partial J(G_\theta, D)}{\partial \theta} = \mathbb{E}_{x\sim G_\theta}\left[\frac{\partial \log G_\theta(x)}{\partial \theta} D(x)\right]$$

Overview

- Introduction to imitation learning
- Core methods of imitation learning
- Advanced methods of imitation learning
- Connection between imitation learning and GANs
- Recent applications of IL for robotics

RoboFlamingo: VLMs as Effective Robot Imitators

- Vision-Language Foundation Models for Robotics

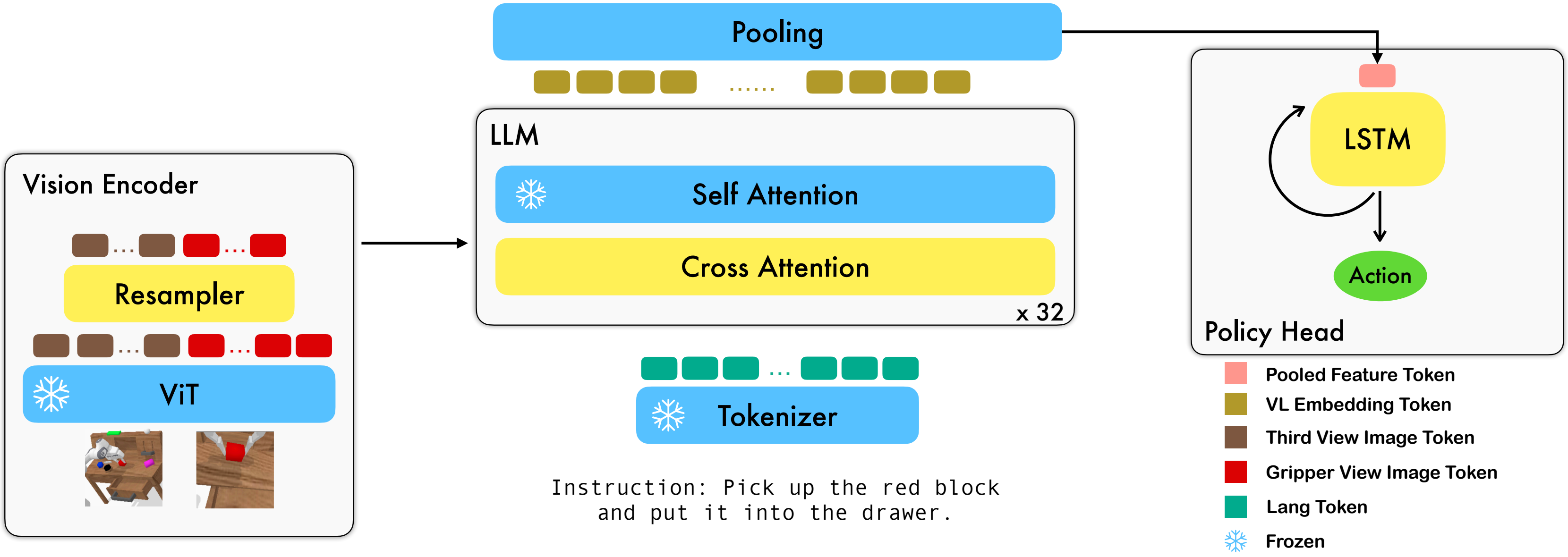


😃 Utilizing existing VLMs 😃 Decreasing the training and inference cost! 😃 Open-source!

Li, Xinghang, et al. "Vision-language foundation models as effective robot imitators." *arXiv preprint arXiv:2311.01378* (2023).

RoboFlamingo: VLMs as Effective Robot Imitators

- Vision-Language Foundation Models for Robotics



$$\ell = \sum_{t} \mathrm{MSE}(a_t^{pose}, \hat{a}_t^{pose}) + \lambda_{gripper}\mathrm{BCE}(a_t^{gripper}, \hat{a}_t^{gripper})$$

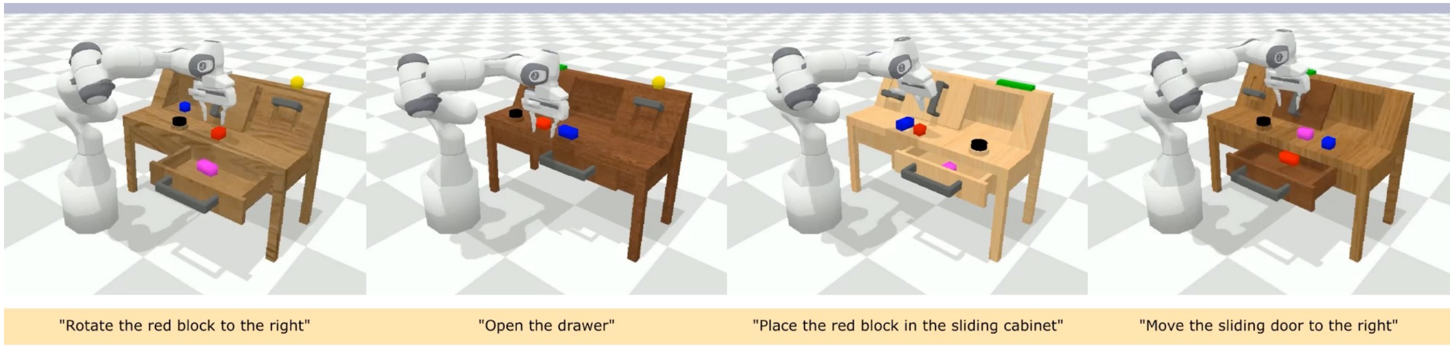
position&rotation

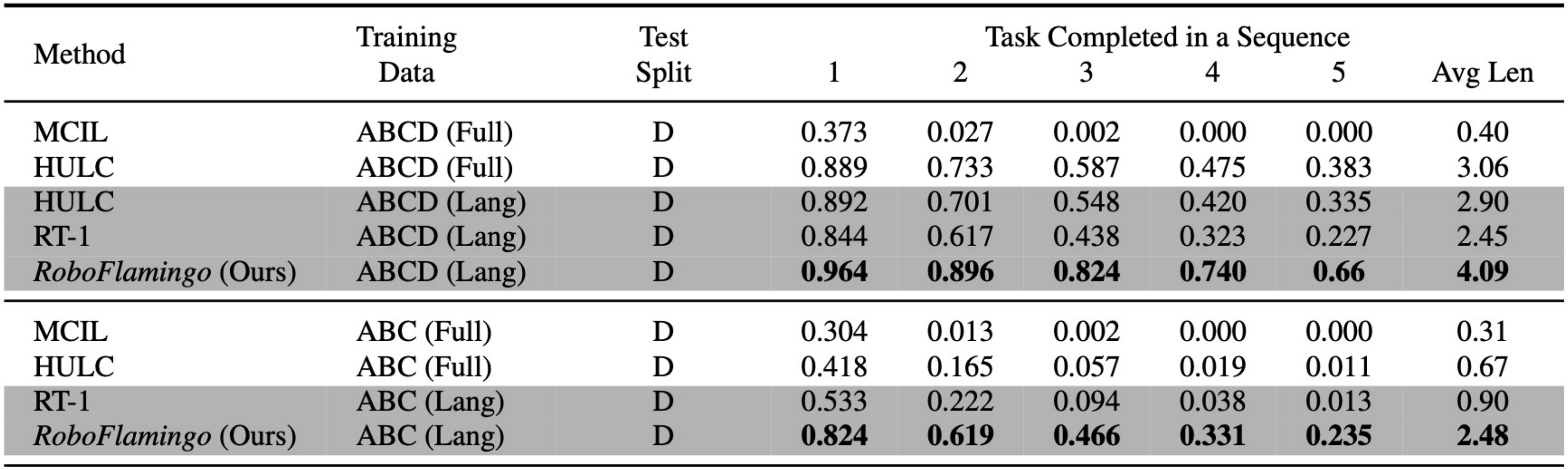
gripper

Li, Xinghang, et al. "Vision-language foundation models as effective robot imitators." *arXiv preprint arXiv:2311.01378* (2023).

RoboFlamingo: VLMs as Effective Robot Imitators

- Experiments on CALVIN dataset
  - A total of 34 distinct tasks and evaluates 1000 unique instruction chains for sequential tasks.
  - A total of 24,000 language-annotated demonstrations, which could be used for model training.


"Rotate the red block to the right"
"Open the drawer"
"Place the red block in the sliding cabinet"
"Move the sliding door to the right"

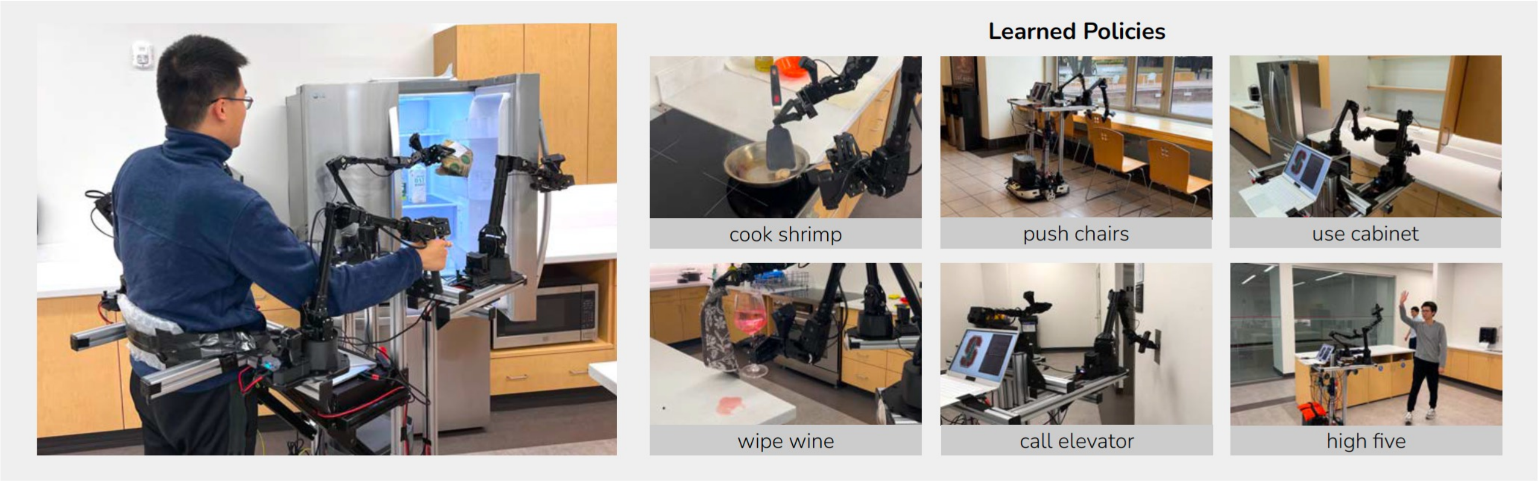


Method	Training Data	Test Split	Task Completed in a Sequence					
			1	2	3	4	5	Avg Len
MCIL	ABCD (Full)	D	0.373	0.027	0.002	0.000	0.000	0.40
HULC	ABCD (Full)	D	0.889	0.733	0.587	0.475	0.383	3.06
HULC	ABCD (Lang)	D	0.892	0.701	0.548	0.420	0.335	2.90
RT-1	ABCD (Lang)	D	0.844	0.617	0.438	0.323	0.227	2.45
RoboFlamingo (Ours)	ABCD (Lang)	D	**0.964**	**0.896**	**0.824**	**0.740**	**0.66**	**4.09**
MCIL	ABC (Full)	D	0.304	0.013	0.002	0.000	0.000	0.31
HULC	ABC (Full)	D	0.418	0.165	0.057	0.019	0.011	0.67
RT-1	ABC (Lang)	D	0.533	0.222	0.094	0.038	0.013	0.90
RoboFlamingo (Ours)	ABC (Lang)	D	**0.824**	**0.619**	**0.466**	**0.331**	**0.235**	**2.48**

Li, Xinghang, et al. "Vision-language foundation models as effective robot imitators." *arXiv preprint arXiv:2311.01378* (2023).

Mobile Aloha

- A $32k-cost mobile manipulation system that is bimanual and supports whole-body teleoperation.



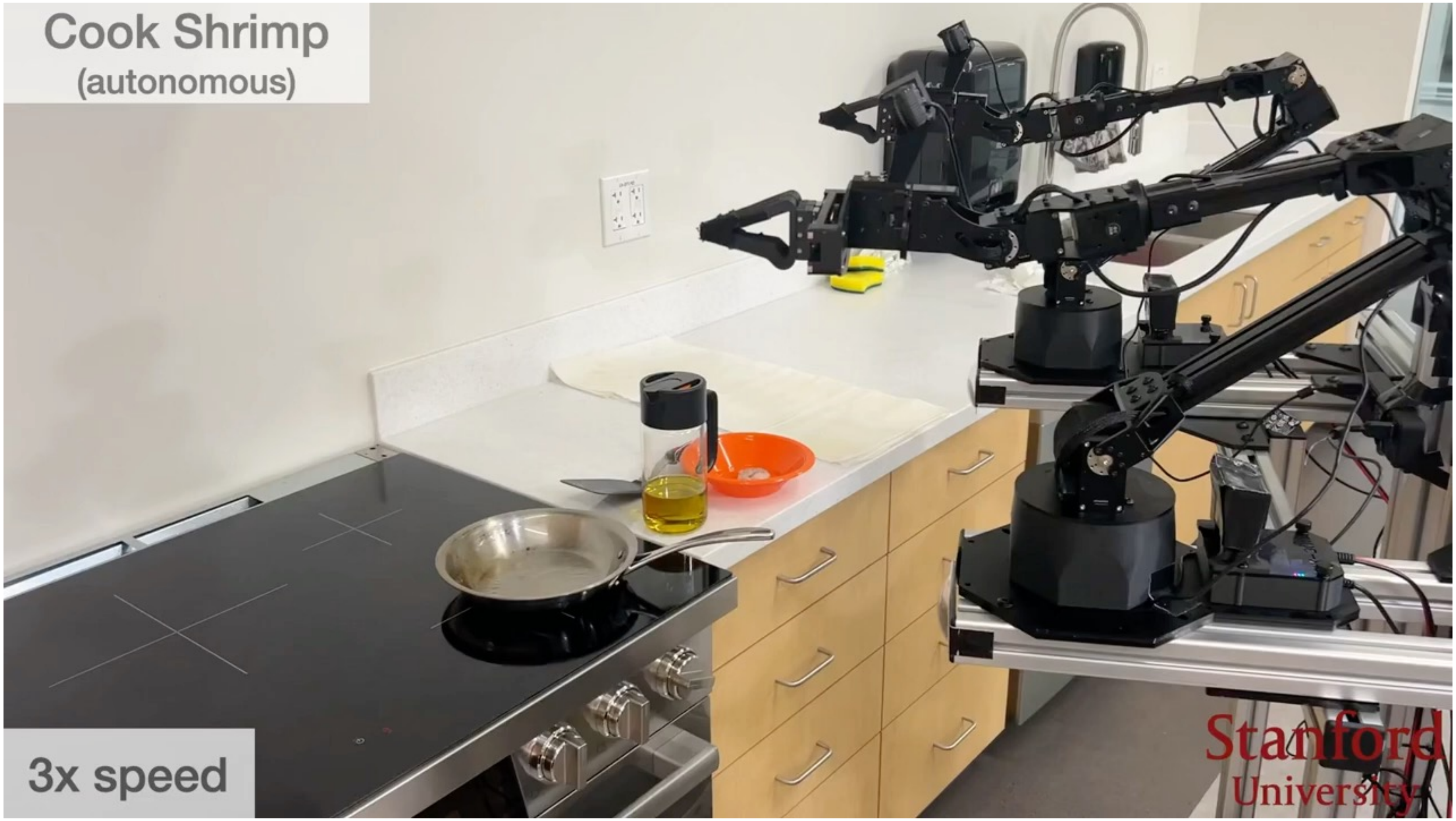
- Imitation learning from expert demonstrations from static ALOHA dataset and mobile ALOHA dataset

Loss

$$\mathbb{E}_{(o^i, a^i_{\text{arms}}, a^i_{\text{base}})\sim D^m_{\text{mobile}}}\left[L(a^i_{\text{arms}}, a^i_{\text{base}}, \pi^m(o^i))\right] + \mathbb{E}_{(o^i, a^i_{\text{arms}})\sim D_{\text{static}}}\left[L(a^i_{\text{arms}}, [0,0], \pi^m(o^i))\right]$$

Fu, Z., Zhao, T.Z. and Finn, C. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *2024*.

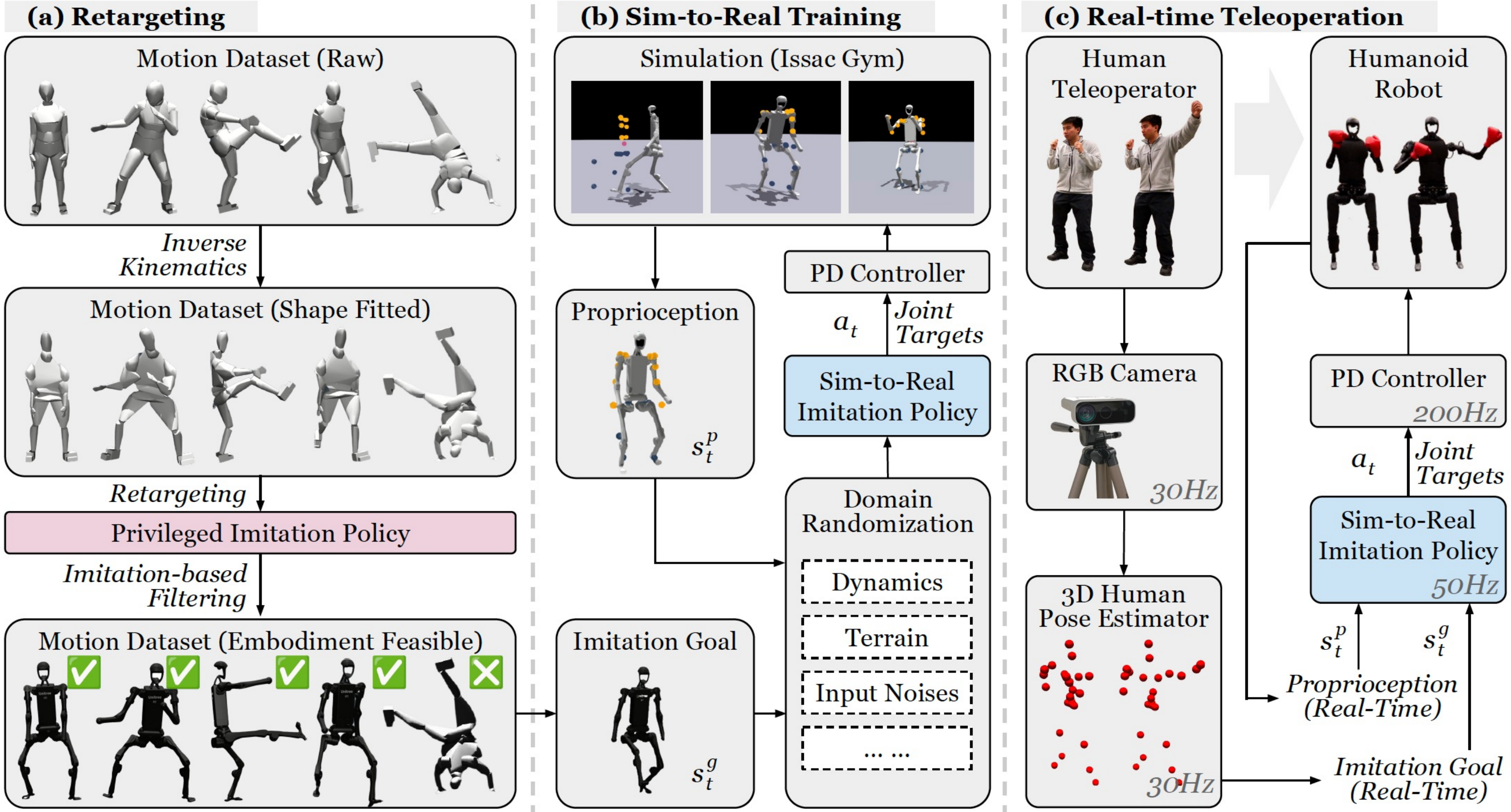
Mobile Aloha



Fu, Z., Zhao, T.Z. and Finn, C. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *2024*.

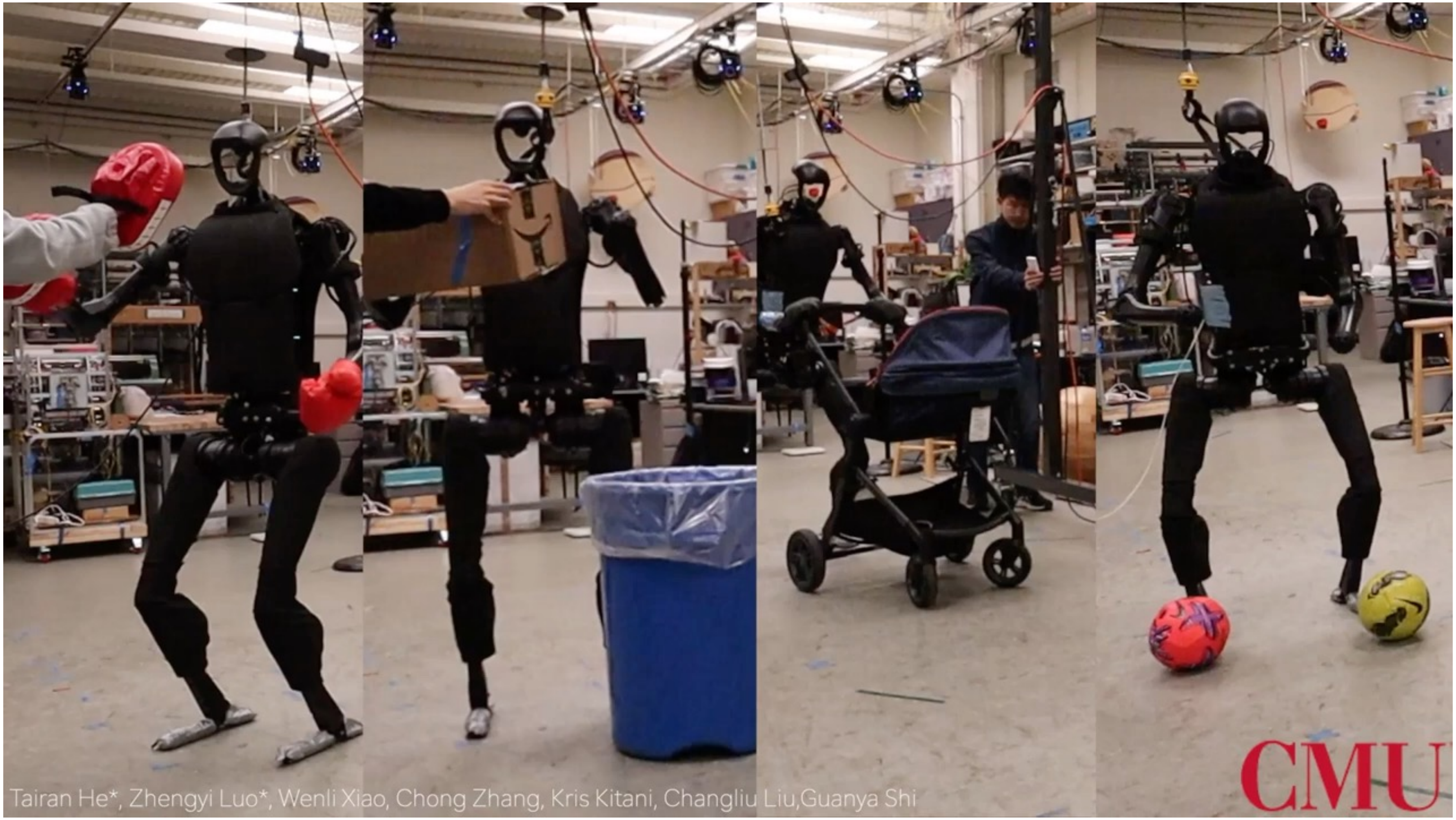
Human2Humanoid (H2O)

Task: control humanoid robots with human motion via teleoperation



He, Tairan, et al. "Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation." *arXiv:2403.04436* (2024).

Human2Humaniod (H2O)


Tairan He*, Zhengyi Luo*, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu,Guanya Shi
CMU

He, Tairan, et al. "Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation." *arXiv:2403.04436* (2024).

Summary of Imitation Learning

- Imitation learning is important when reward function is unavailable or hard to properly define
- Behavior cloning is straightforward and easy to implement, but suffer from distribution shift or exposure bias
- Inverse RL first recover the underlying reward of the expert from the trajectory and then perform RL to obtain the policy, but suffer from high-complexity of bi-level optimization
- GAIL aims to match the occupancy measures with GAN
- IL & GAN are highly related. You can say GAN is IL for data generation tasks
- IL has been recently leveraged to tuning large models for agents or robotics

Thank You!
Questions?

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Weinan Zhang

Associate Professor

APEX Data & Knowledge Management Lab

John Hopcroft Center for Computer Science

Shanghai Jiao Tong University

http://wnzhang.net