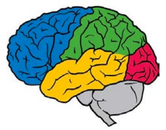
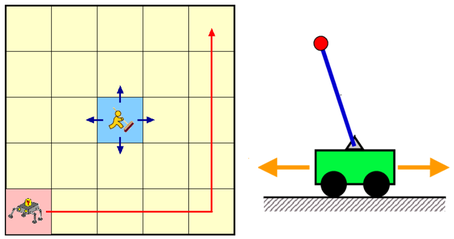
Deep Reinforcement Learning toward Robot Learning in the Wild

Shixiang (Shane) Gu (顾世翔)

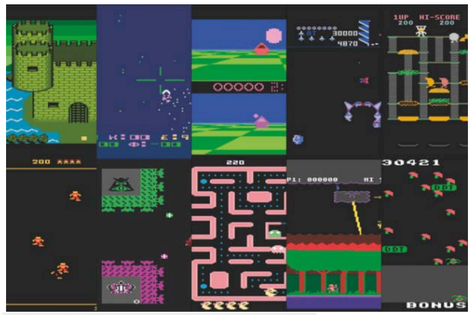


Deep RL: successes and limitations

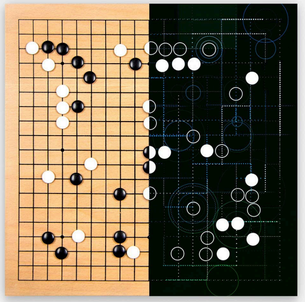


Classic RL
[1990s-2014]

Fast simulators = success is guaranteed



Atari games
[Mnih et. al., 2015]

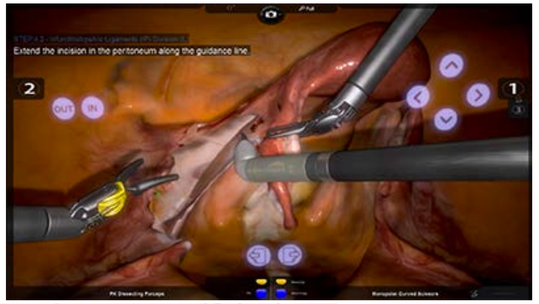


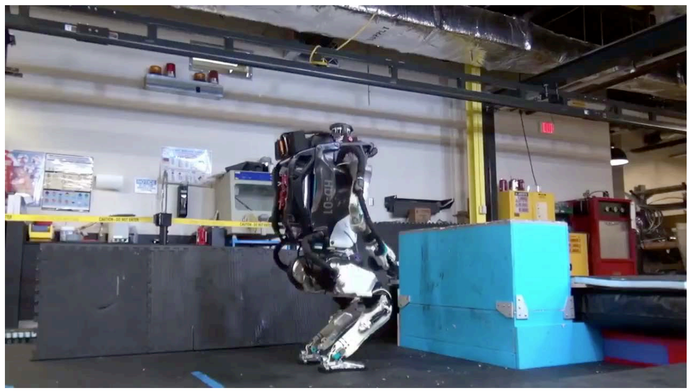
AlphaGo/AlphaZero
[Silver et. al., 2016; 2017]

Parkour
[Heess et. al., 2017]

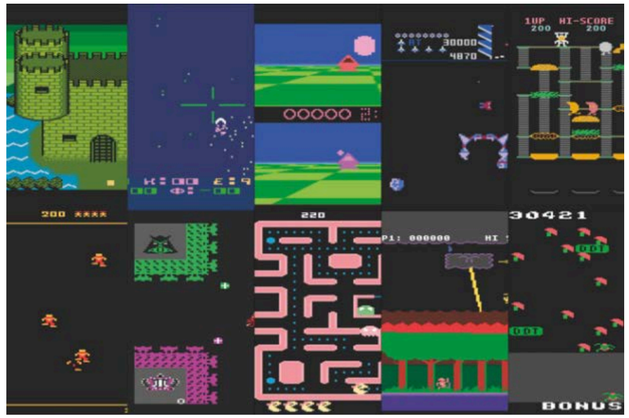
?

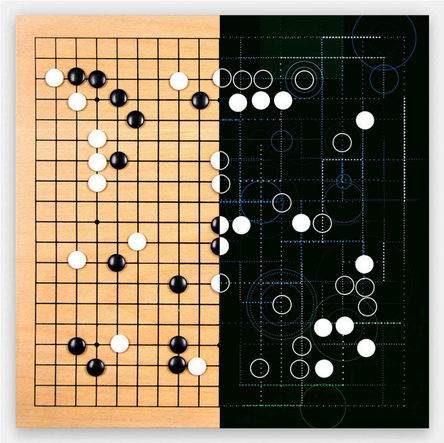
Slow simulators/Real-world = require smart (sample-efficient) algorithms





Why Robotics?

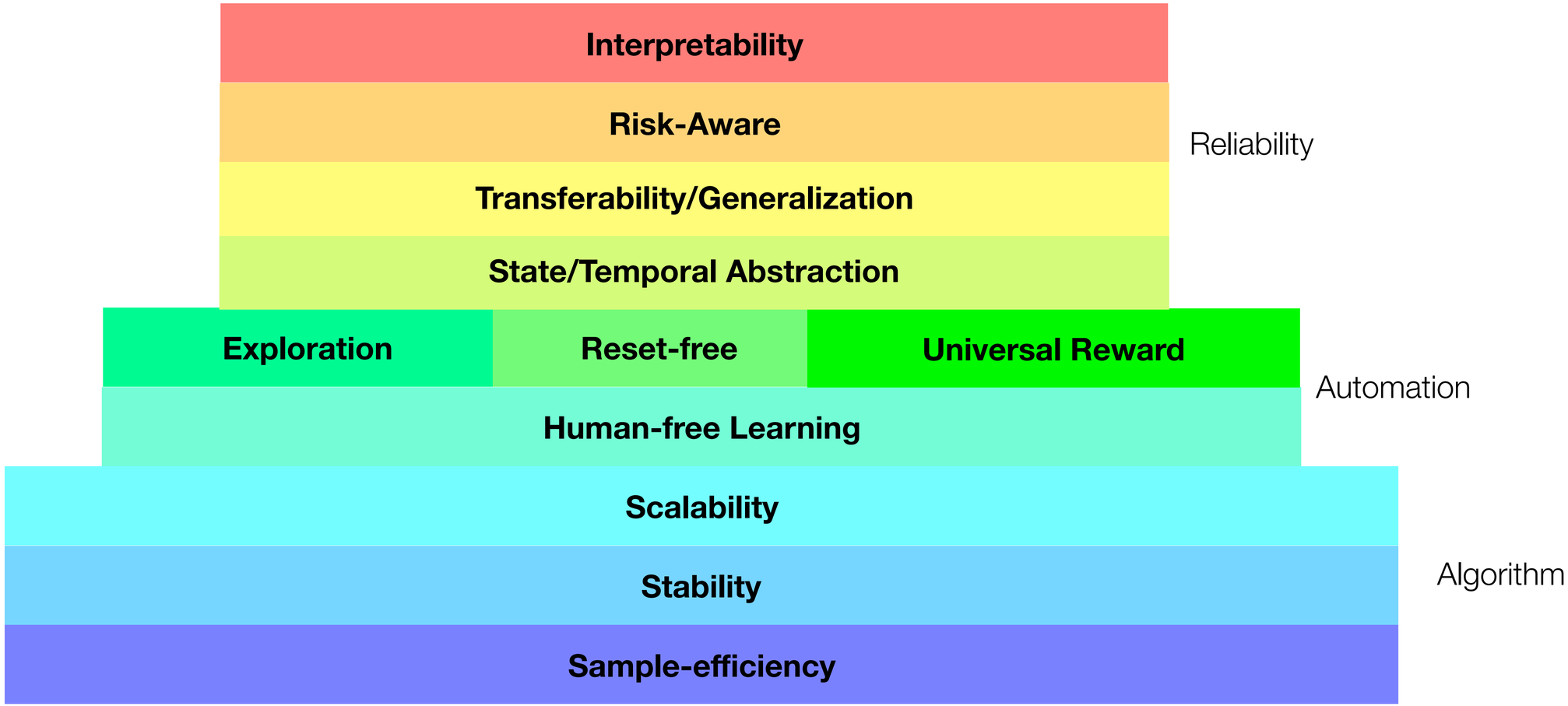




vs

Generally continuous control is a much harder exploration problem than any game.

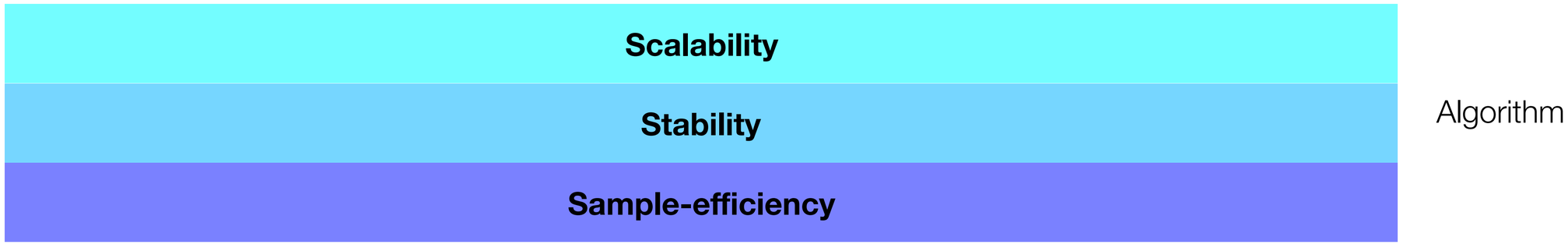
Recipe for a Smart Deep RL Algorithm



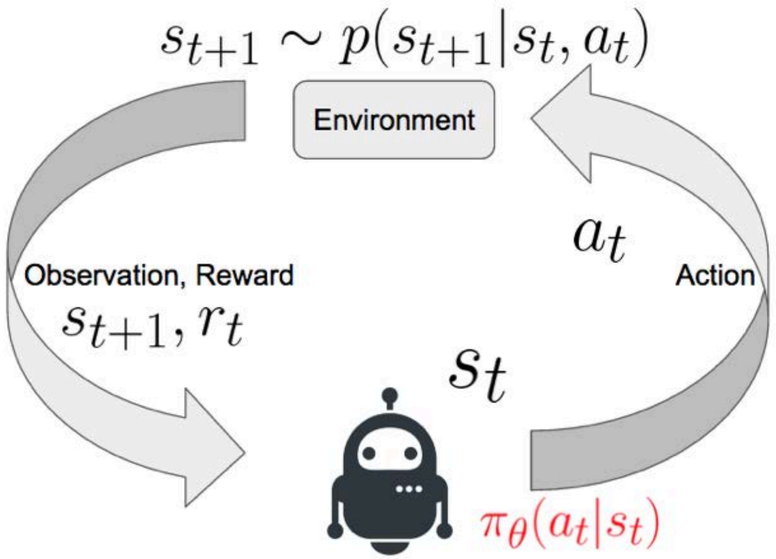
Outline of the talk

- **Sample-efficiency**
  - Good Off-policy Algorithm： **Normalized Advantage Functions** [Gu et al, 2016], **Q-Prop/Interpolated Policy Gradient** [Gu et al, 2017/2017]
  - Good Model-based Algorithm： **Temporal Differene Models** [Pong*, Gu* et al, 2018]
- **Human-free Learning**
  - Safe & reset-free RL： **Leave No Trace** [Eysenbach, Gu et al, 2018]
  - “Universal” reward function： **Temporal Difference Models** [Pong*, Gu* et al, 2018]
- **Hierarchical RL**
  - Data-efficient hierarchical RL： **HIRO** [Nachum, Gu et al, 2018, 2018]

Toward Efficient and Stable RL Algorithms

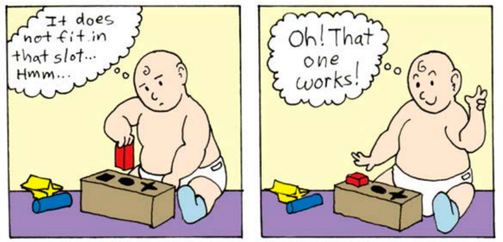


RL: Notations & Definitions



on-policy model-free: e.g. policy search ~ trial and error

$$\pi^* = \arg\max_\pi \mathbb{E}_\pi \left[\sum_t \gamma^t r_t \right]$$



off-policy model-free: e.g. Q-learning ~ introspection

$$Q^* = \arg\min_Q \mathbb{E}_\beta \left[\left(Q(s_t, a_t) - r_t - \gamma \max_a Q(s_{t+1}, a) \right)^2 \right]$$

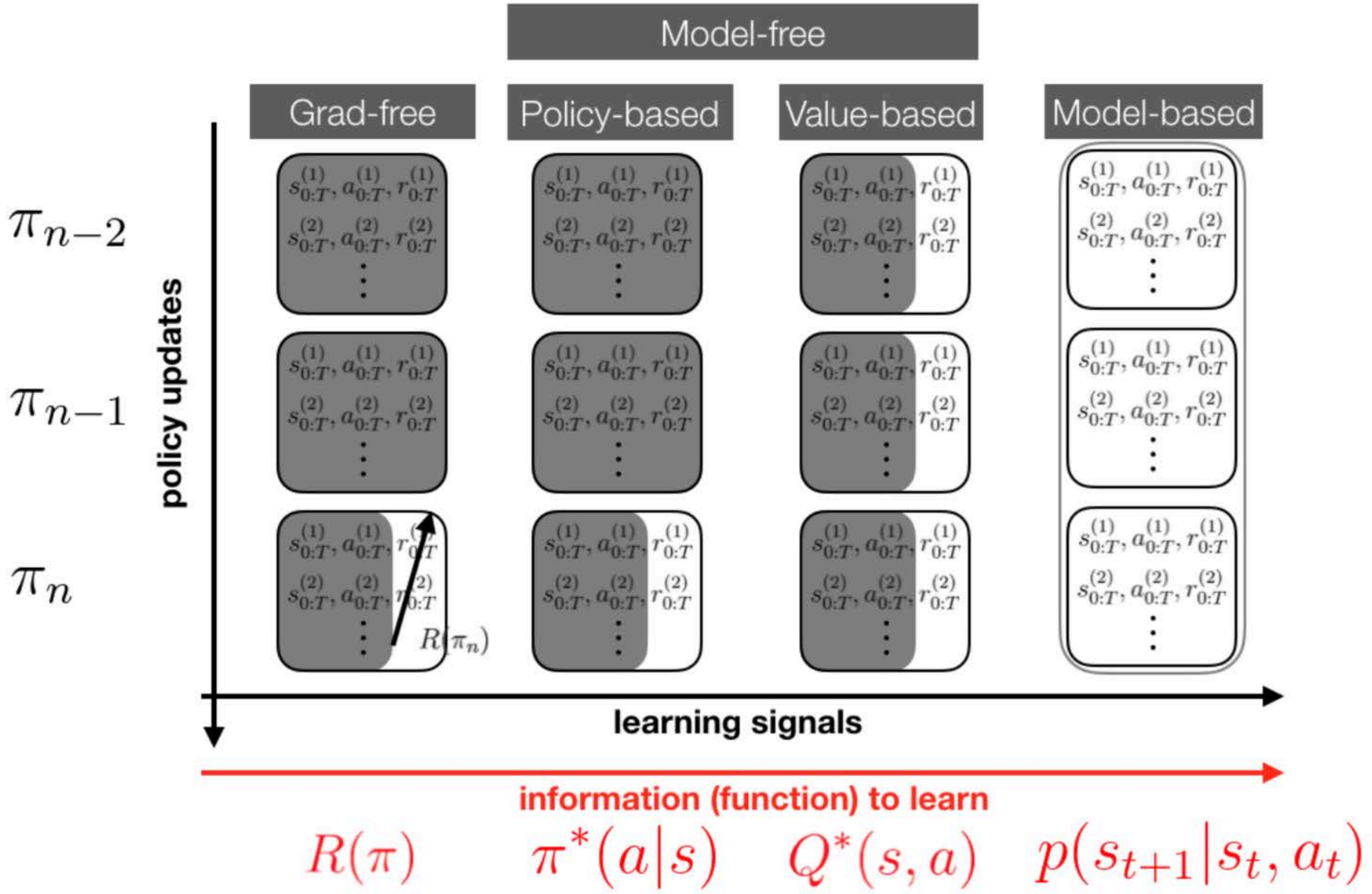
$$\pi^*(a_t|s_t) = \delta \left(a_t = \arg\max_a Q^*(s_t, a) \right)$$

model-based: e.g. MPC ~ imagination

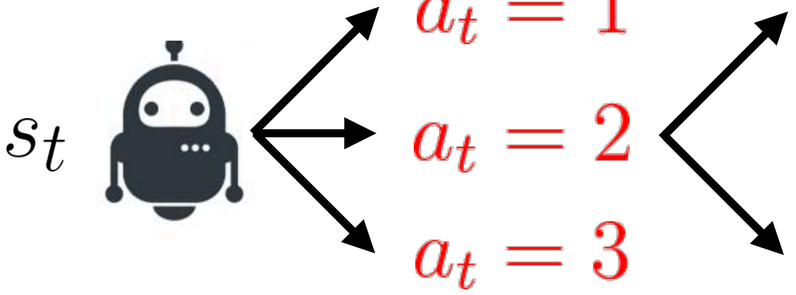
$$f^* = \arg\min_f \mathbb{E}_\beta \left[||f(s_t, a_t) - s_{t+1}||_2 \right]$$

$$\pi^* : a_t^* = \arg\max_{a_{t:t+T}} \sum_{i=0}^{T} \gamma^i r_{t+i}, \quad where \quad s_{t+i+1} = f(s_{t+i}, a_{t+i})$$

Sample-efficiency: trade-off between learning signal & how much to learn



Toward Good Off-policy Deep RL Algorithm



On-policy Monte Carlo policy gradient, e.g. TRPO [Schulman et al, 2015]

- **Many new samples needed per update.**
- Stable but very sample-intensive

$$\hat{Q}(s_t, a_t = 2) = \sum_{t' \geq t} r(s_{t'}, a_{t'})$$

$$\mathbb{E}\left[\hat{Q}(s_t, a_t = 2)\right] = Q^\pi(s_t, a_t = 2)$$

$$Q_w(s_t, a_t = 2) \approx Q^\pi(s_t, a_t = 2)$$

Off-policy actor-critic, e.g. DDPG [Lillicrap et al, 2016]

- **No new samples needed per update!**
- Quite sensitive to hyper-parameters

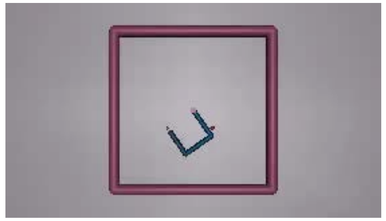
"Better" DDPG

- NAF [Gu et al 2016], Double DQN [Hasselt et al 2016], Dueling DQN [Wang et al 2016], Q-Prop/IPG [Gu et al 2017/2017], ICNN [Amos et al 2017], SQL/SAC [Haarnoja et al 2017/2017], GAC [Tangkaratt et al 2018], MPO [Abdolmaleki et al 2018], TD3 [Fujimoto et al 2018], …

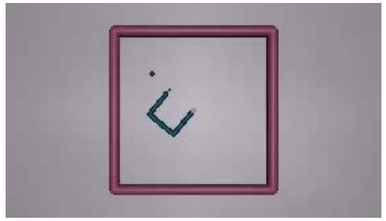
Normalized Advantage Functions (NAF)

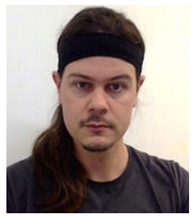
- Benefit: 2 objectives (actor-critic) to 1 objective (Q-learning)
  - Halve #hyperparameters
- Limitation:
  - Doesn't work well on locomotion
  - Works well on manipulation

DDPG



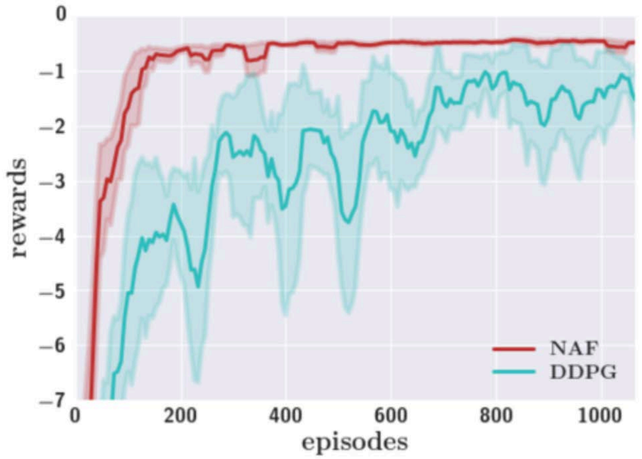
NAF





Continuous Deep Q-Learning with Model-based Acceleration [Gu, Lillicrap, Sutskever, Levine, ICML 2016]

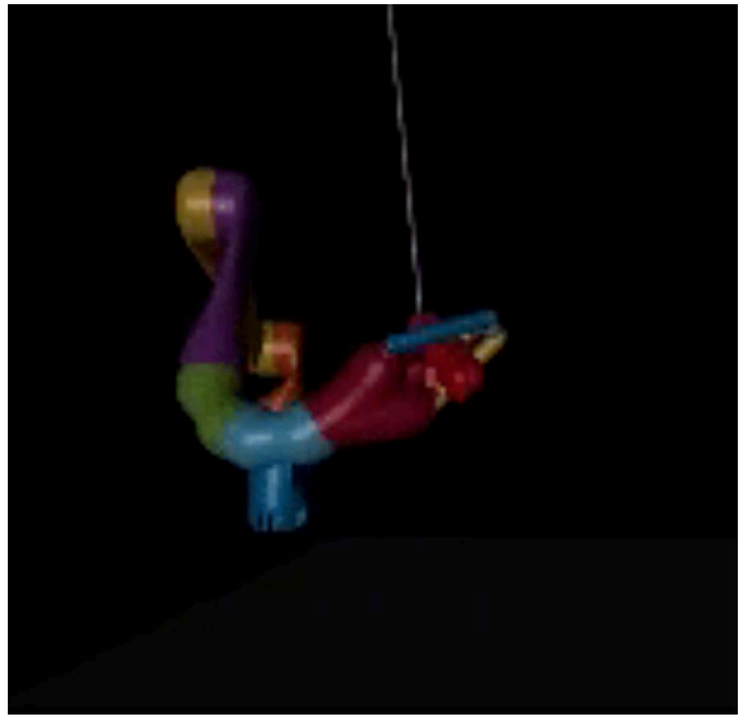
3-joint peg insertion



Related/later work:

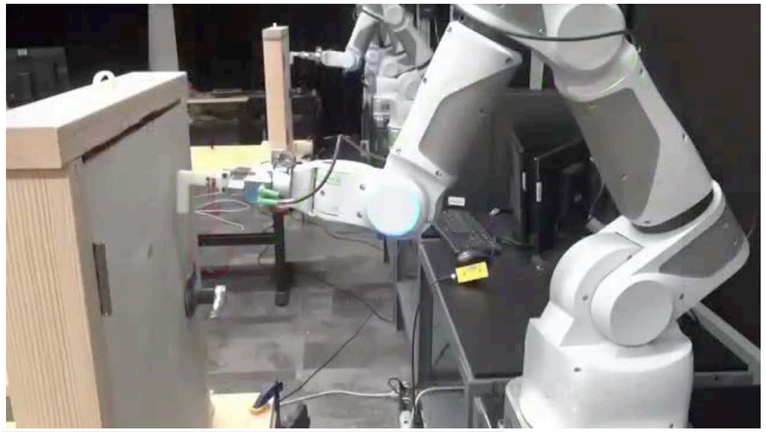
- Dueling Network [Wang et al 2016]
- ICNN [Amos et al 2017]
- SQL [Hajaorna et al 2017]

NAF on JACO arm grasp & reach (100hz)

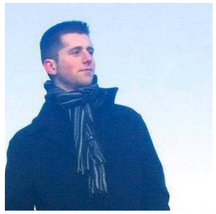


Asynchronous NAF for Simple Manipulation

Train time/Exploration

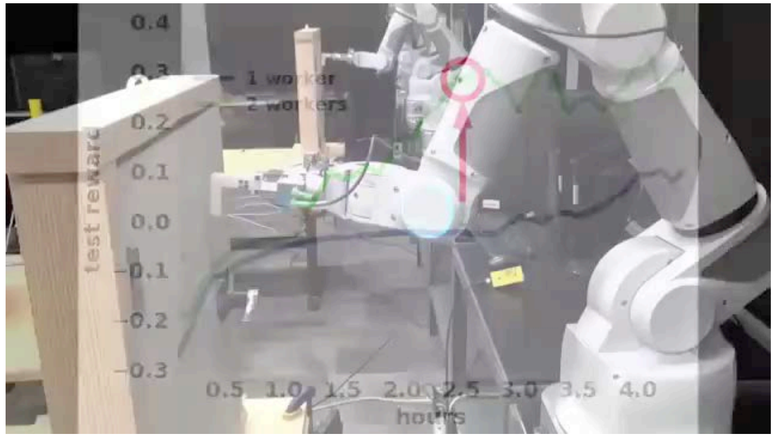


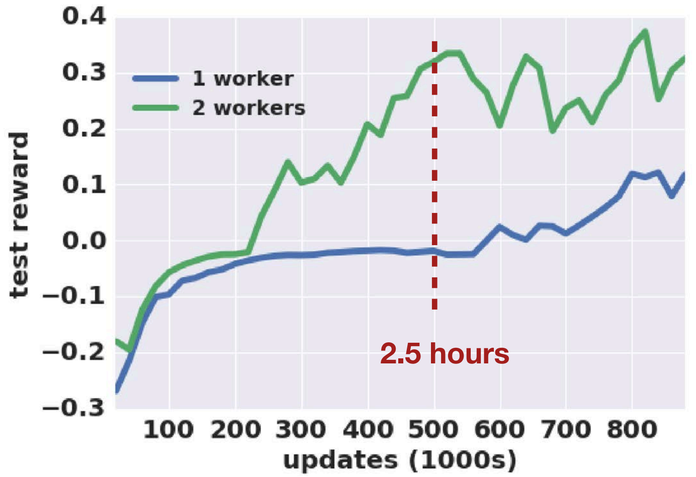
Google



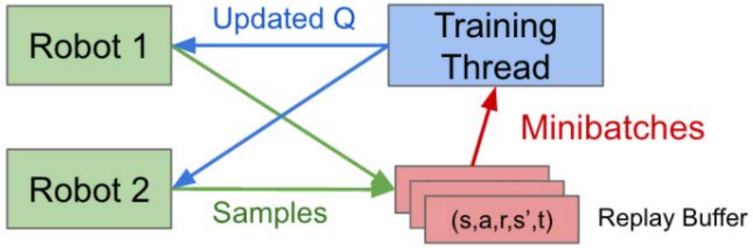
Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates [Gu*, Holly*, Lillicrap, Levine, ICRA 2017]

Test time



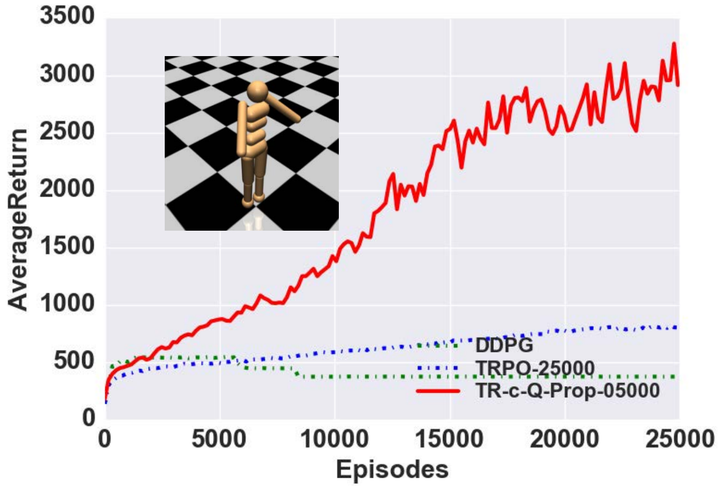


Disturbance test



Q-Prop & Interpolated Policy Gradient (IPG)

- On-policy algorithms are stable. How to make off-policy more on-policy?
  - Mixing Monte Carlo returns
  - Trust-region policy update
  - On-policy exploration
  - Bias trade-offs (theoretically bounded)



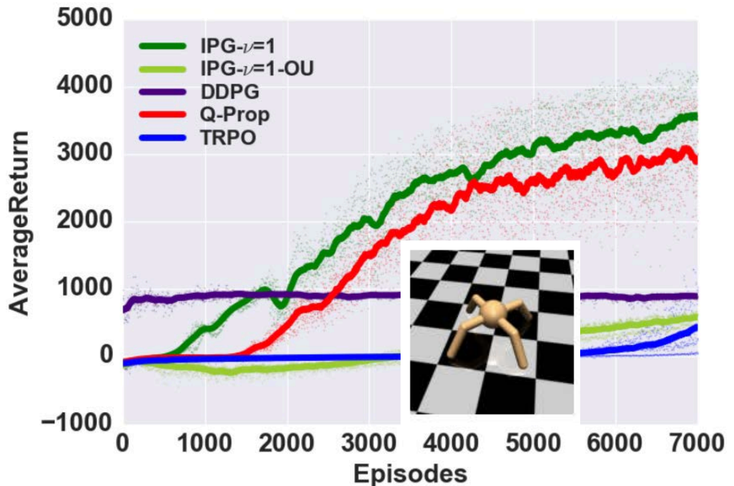
$$(1-\nu)\mathbb{E}_{\rho^\pi,\pi}[\nabla_\theta \log \pi_\theta(a_t|s_t)(\hat{A}(s_t,a_t) - A^\pi_w(s_t,a_t))] + \mathbb{E}_{\rho^\beta}[\nabla_\theta \bar{Q}^\pi_w(s_t)]$$

On-policy Monte Carlo

On-policy deterministic

[Gu, Lillicrap, Ghahramani, Turner, Levine, ICLR 2017]
[Gu, Lillicrap, Ghahramani, Turner, Schoelkopf, Levine, NIPS 2017]

Related concurrent work:
- PGQ [O'Donoghue et al 2017]
- ACER [Wang et al 2017]

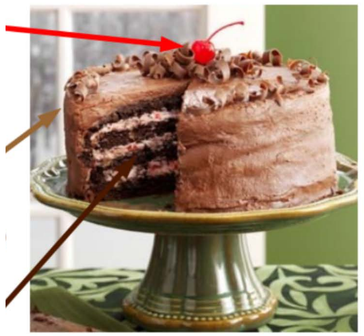


Toward Good Model-based Deep RL Algorithm

- Often, off-policy model-free has higher asymptotic performance than model-based
  - Off-policy model-free is sort of like model-based (model = experience replay). What's the connection/gap?

- Q-learning vs Goal-Conditioned Q-learning

$$Q(s,a) : (s_t, a_t, s_{t+1}, r_t) \sim \beta$$



$$Q(s,a,g) : (s_t, a_t, s_{t+1}) \sim \beta, r_t = r(s_t, a_t, s_{t+1}, g)$$

Off-policy + **Relabeling trick** from HER [Andrychowicz et al, 2017]

Examples:
- UVF [Schaul et al, 2015]
- TDM [Pong*, Gu* et al 2017]

- Off-policy learning can use same memory to learn wrt any goal

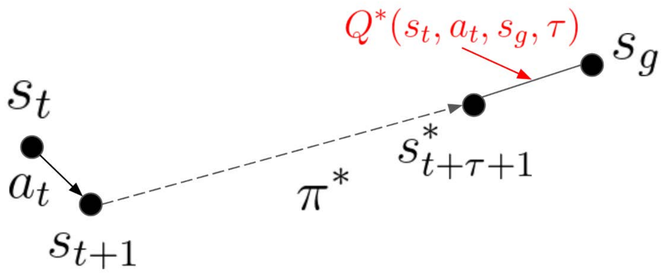
$$Q^*(s_t, a_t, g) = r(s_t, a_t, s_{t+1}, g) + \gamma \max_a Q^*(s_{t+1}, a, g)$$

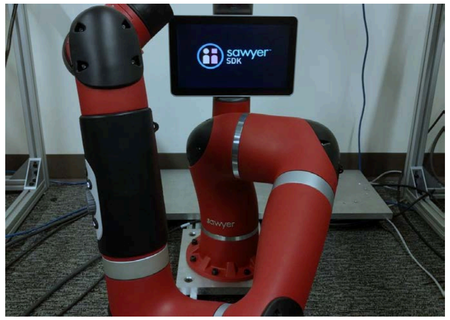
Temporal Difference Models (TDM)

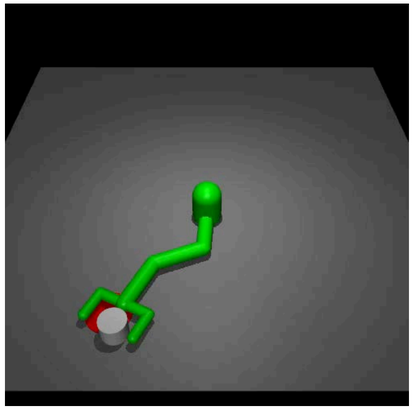
- A certain parameterized Q-function is **a generalization of dynamics model**

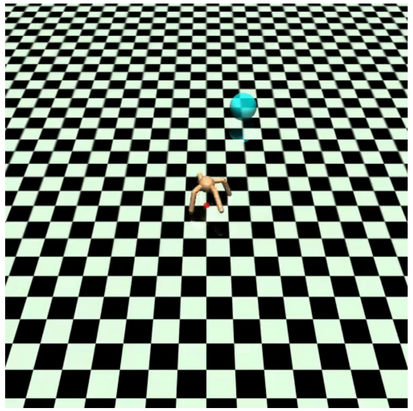
$$r_d(s_t, a_t, s_{t+1}, s_g, \tau) = -D(s_{t+1}, s_g)1[\tau = 0]$$

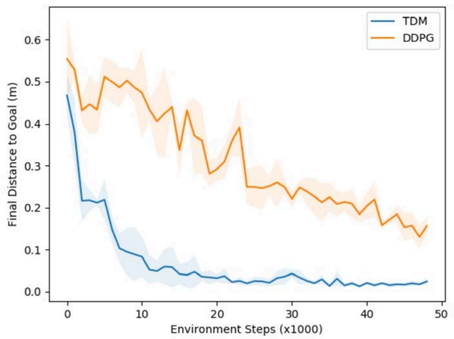
$$a_t = \underset{a_t, a_{t+T}, s_{t+T}}{\text{argmax}} \; r_c(s_{t+T}, a_{t+T}) \text{ such that } Q(s_t, a_t, s_{t+T}, T-1) = 0$$

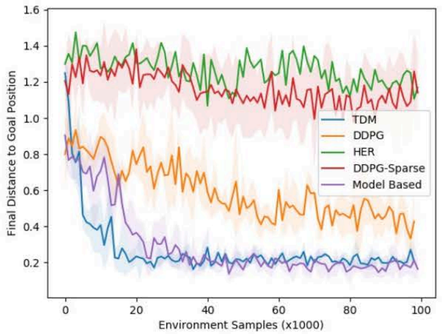


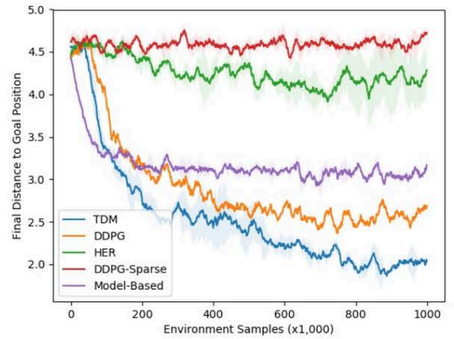






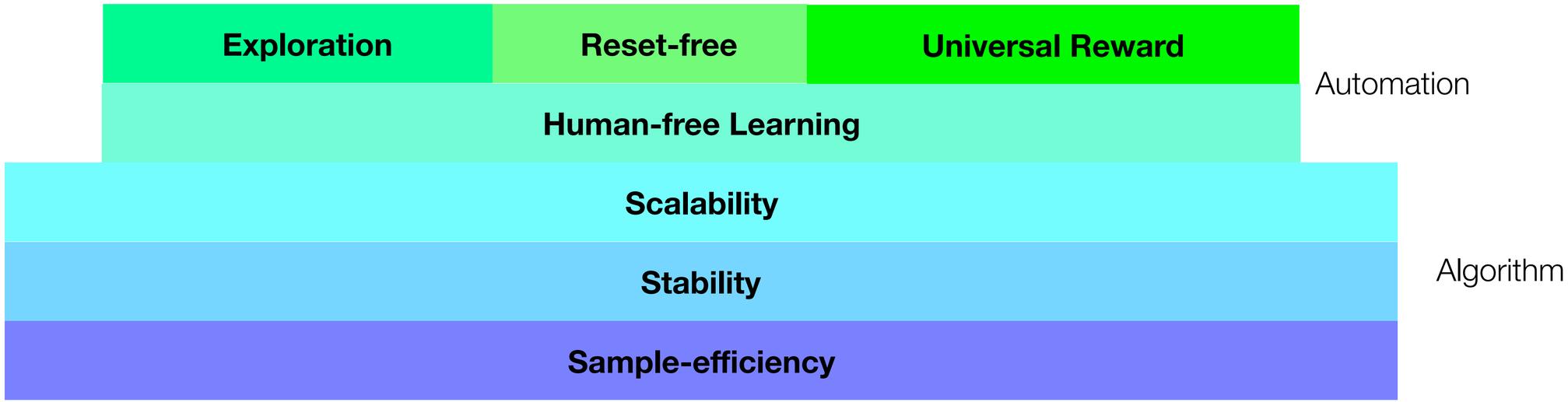




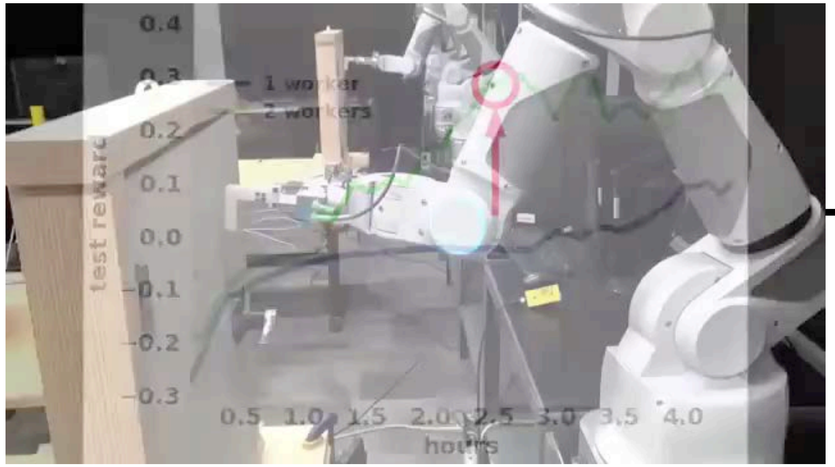


Temporal Difference Models: Model-Free Deep RL for Model-Based Control [Pong*, Gu*, Dalal, Levine, ICLR 2018]

Toward Unsupervised Reinforcement Learning



Toward Human-free Learning



?

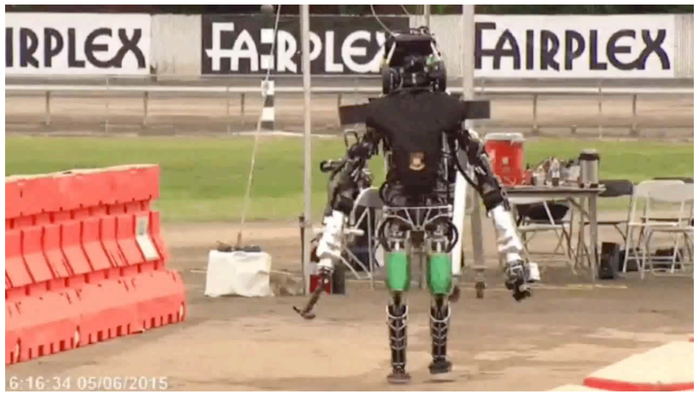
Manual resetting,
Reward engineering,
Human-administered,

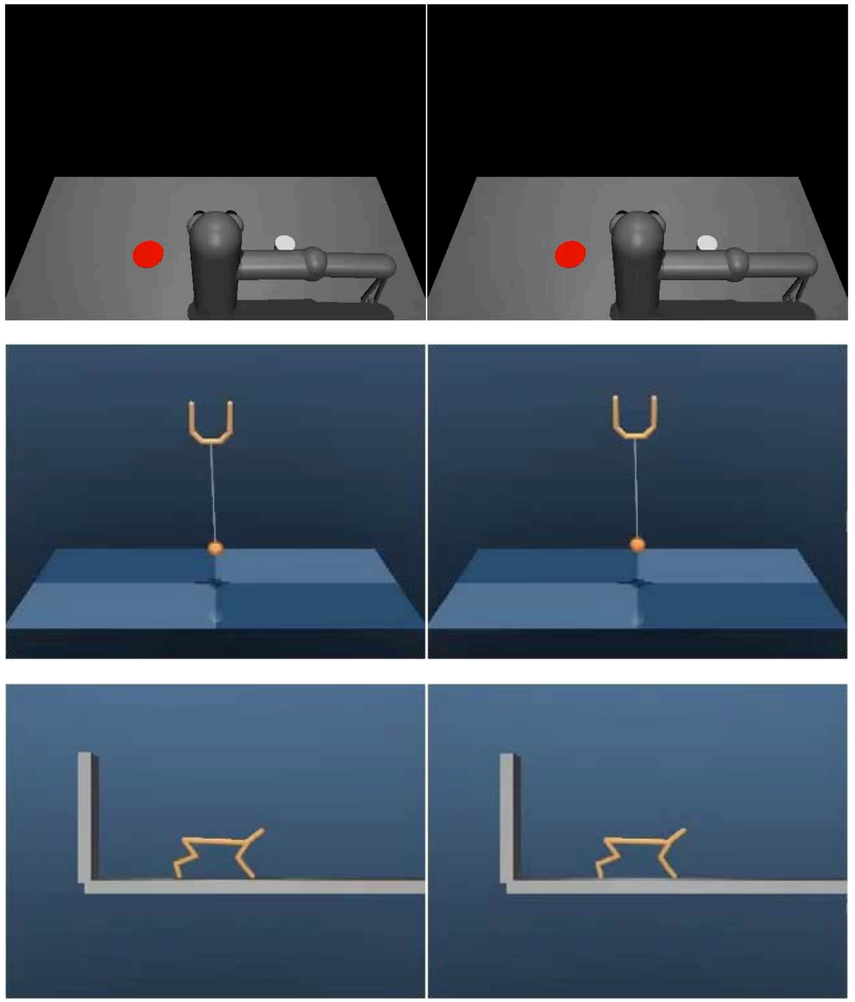
Autonomous, Continual,
Safe, Human-free

Leave No Trace (LNT)

Who resets the robot?

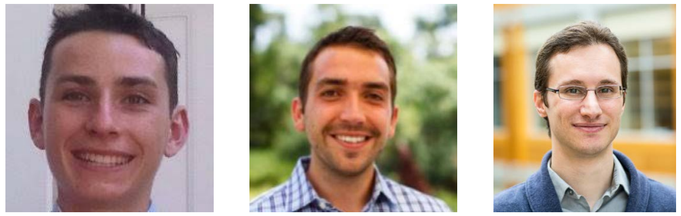
- PhD students





- Learn to reset

$$\mathcal{E}^* \triangleq \{(s,a) \in \mathcal{E} \mid Q_{reset}(s,a) > Q_{min}\}$$



Leave no Trace: Learning to Reset for Safe and Autonomous Reinforcement Learning [Eysenbach, Gu, Ibarz, Levine, ICLR 2018]

Related work:

- Asymmetric self-play [Sukhbaatar et al 2017]
- Automatic goal generation [Held et al 2017]
- Reverse curriculum [Florensa et al 2017]

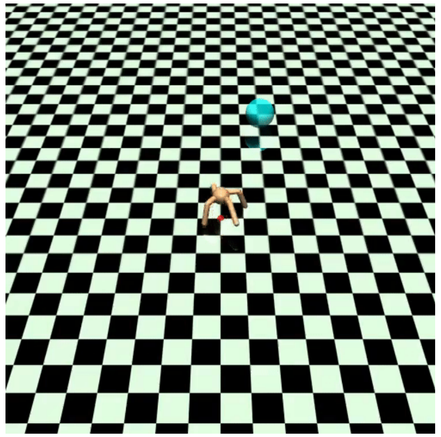
A “Universal” Reward Function + Off-Policy Learning

One reward, One skill → One reward, Many skills

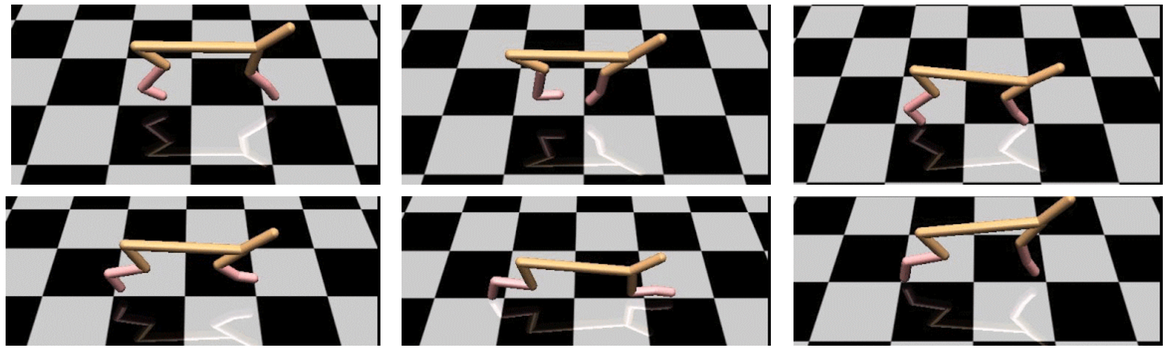
$r_1(s,a)$ $r_2(s,a)$ $r_3(s,a)$ $r(s,a)$

- Goal: learn as **many useful skills** as possible **sample-efficiently** with **minimal reward engineering**
- Examples:

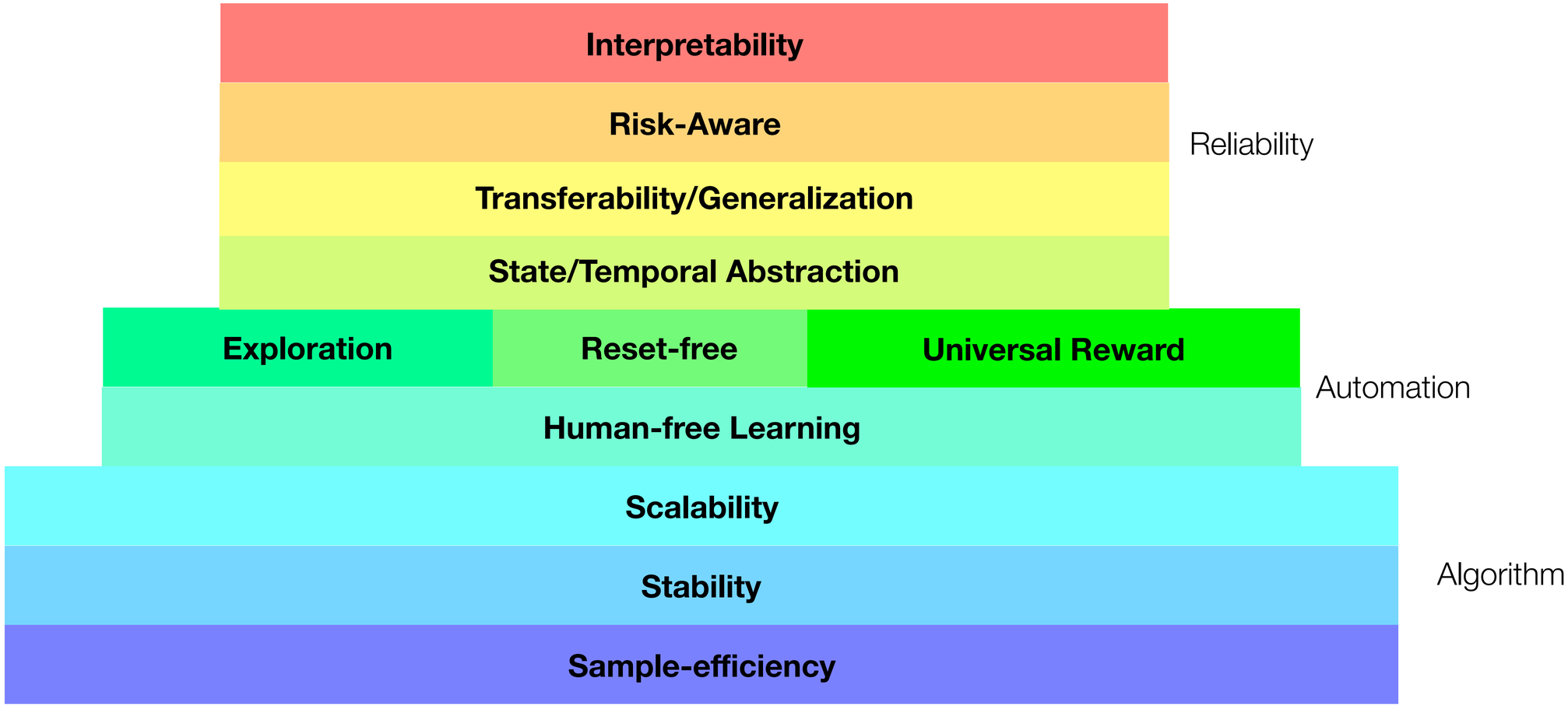
Goal-reaching reward, e.g.
UVF [Schaul et al 2015]/HER[Andrychowicz], **TDM** [Pong*, Gu* et al 2018]



Diversity reward, e.g.
SNN4HRL [Florensa et al 2017], **DIAYN** [Eysenbach et al 2018]

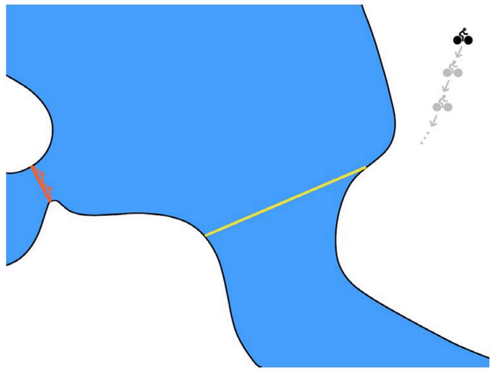


Toward Reliable RL



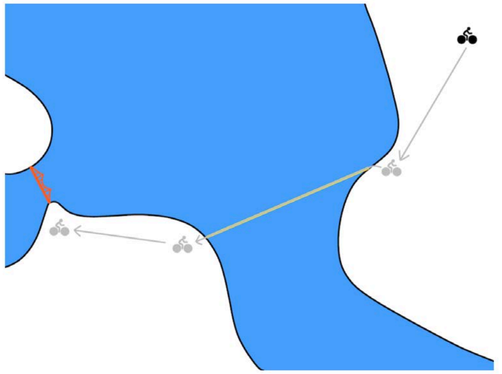
Toward Hierarchical Reinforcement Learning (HRL)

When you don't know how to ride bike…

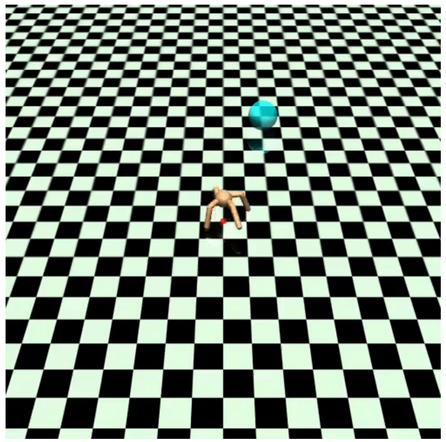


?

When you know how to ride bike…



TDM learns many skills very quickly…



?

How to efficiently solve other problems?

HRL:

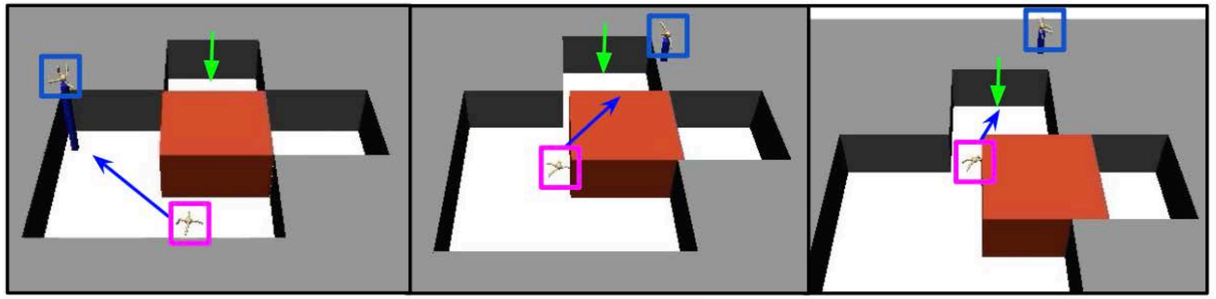
- Longer horizon problems
- More interpretable
- More transferrable policies

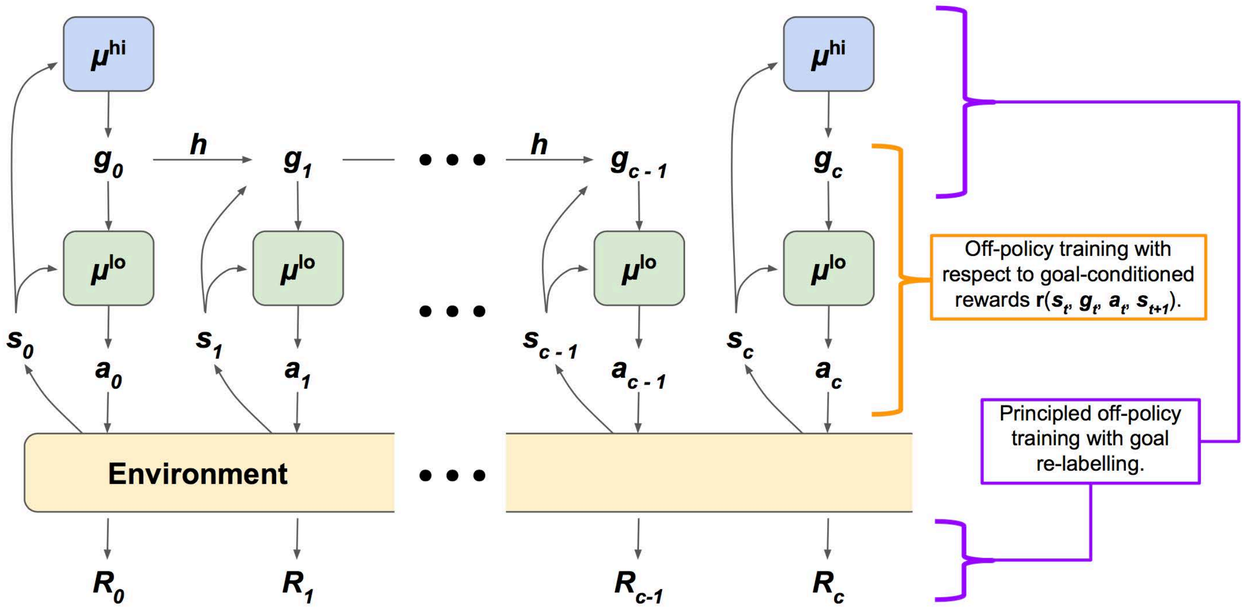
HIerarchical Reinforcement learning with Off-policy correction (HIRO)

- Most recent HRL work is on-policy
  - e.g. option-critic [Bacon et al 2015], FuN [Vezhnevets et al 2017], SNN4HRL [Florensa et al 2017], MLSH [Frans et al 2018]
  - **VERY data-intensive**
- How to correct for off-policy? Relabel the action.

$$(s_t, g_t, s_{t+c}) \longrightarrow (s_t, \tilde{g}_t, s_{t+c})$$

$$\tilde{g}_t = \arg\max_g \log \mu^{lo,new}(a_{t:t+c-1}|s_t, g)$$



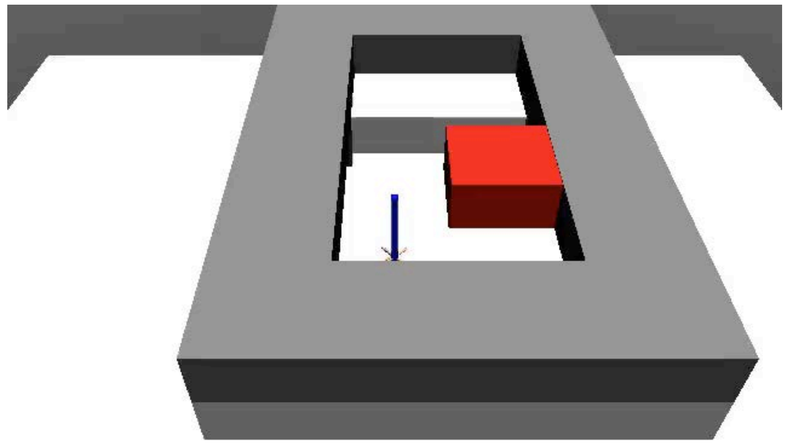




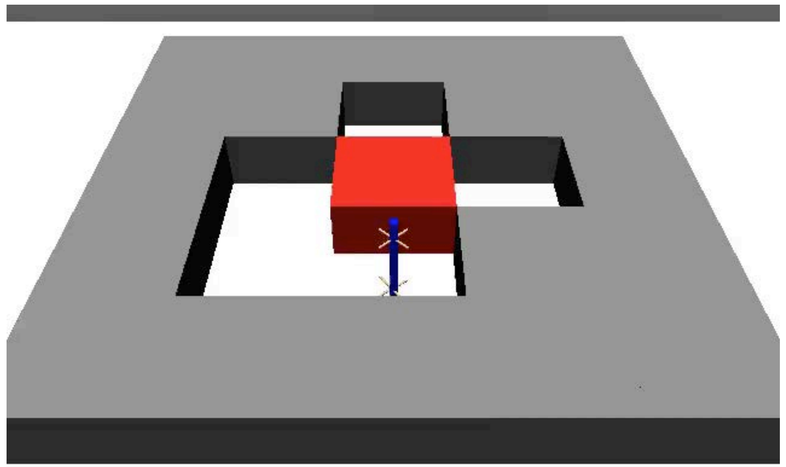
Data-efficient Hierarchical Reinforcement Learning
[Nachum, Gu, Lee, Levine, NIPS 2018]

HIRO (cont.)

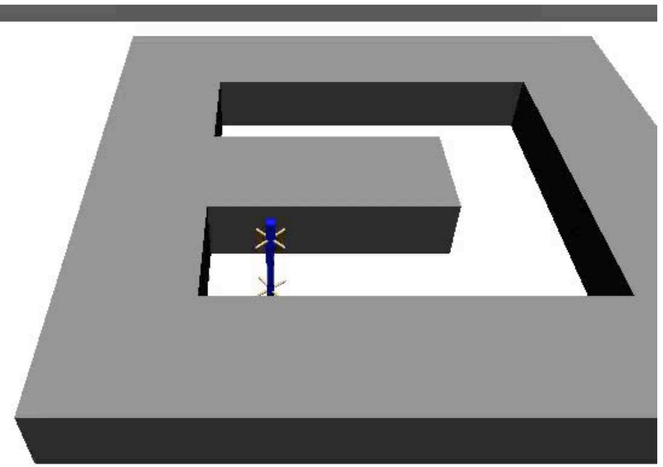
Ant Maze



Ant Push



Ant Fall



Data-efficient Hierarchical Reinforcement Learning
[Nachum, Gu, Lee, Levine, NIPS 2018]

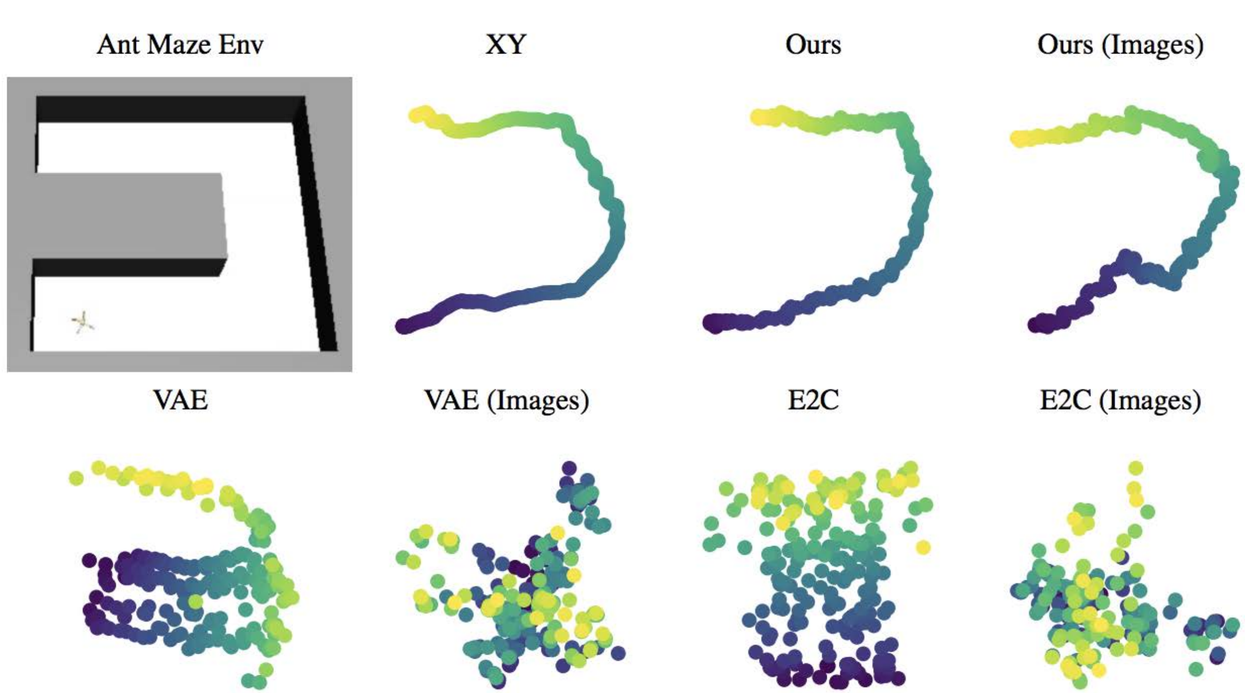
	Ant Gather	Ant Maze	Ant Push	Ant Fall
HIRO	**3.02±1.49**	**0.99±0.01**	**0.92±0.04**	**0.66±0.07**
FuN representation	0.03 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
FuN transition PG	0.41 ± 0.06	0.0 ± 0.0	0.56 ± 0.39	0.01 ± 0.02
FuN cos similarity	0.85 ± 1.17	0.16 ± 0.33	0.06 ± 0.17	0.07 ± 0.22
FuN	0.01 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
SNN4HRL	1.92 ± 0.52	0.0 ± 0.0	0.02 ± 0.01	0.0 ± 0.0
VIME	1.42 ± 0.90	0.0 ± 0.0	0.02 ± 0.02	0.0 ± 0.0

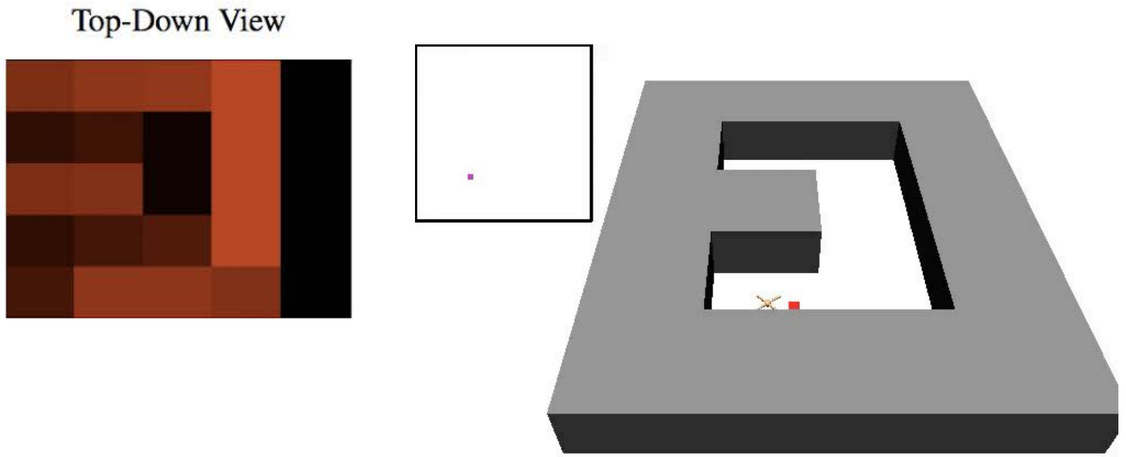
Test rewards at 20000 episodes

[Vezhnevets et al, 2017]

[Florensa et al, 2017]
[Houthooft et al, 2016]

HIRO + Principled Representation Learning





$$\text{SubOpt}(\Psi) = \sup_{s \in S} V^{\pi^*}(s) - V^{\pi^*_{\text{hier}}}(s)$$

Near-Optimal Representation Learning for Hierarchical Reinforcement Learning [Nachum, Gu, Lee, Levine, preprint 2018]

Theorem 1. *If there exists* $\varphi : S \times A \to G$ *such that,*

$$\sup_{s \in S, a \in A} D_{\text{TV}}(P(s'|s,a)||P(s'|s,\Psi(s,\varphi(s,a)))) \leq \epsilon,$$

then $\text{SubOpt}(\Psi) \leq C\epsilon$*, where* $C = \frac{2\gamma}{(1-\gamma)^2} R_{max}$.

Discussion

Natural language, Causality

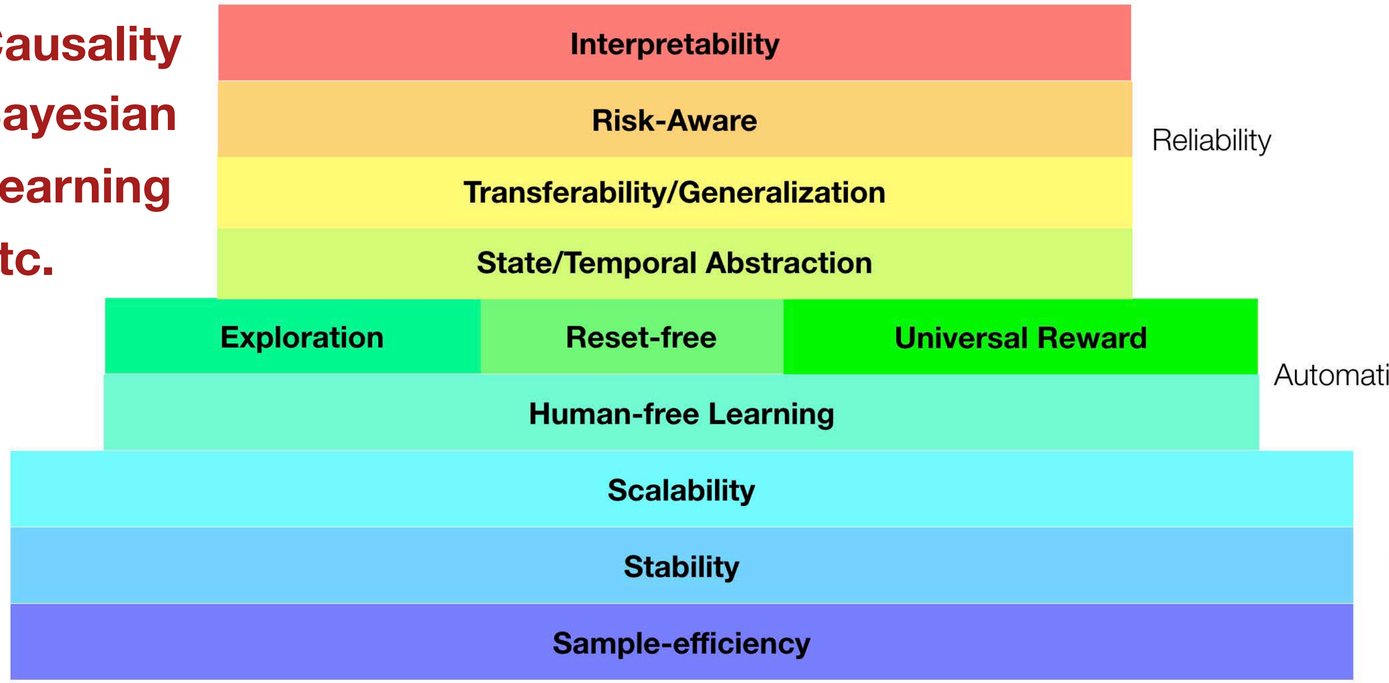
Distributional, Bayesian

Sim2Real, Meta learning

HIRO + etc.

LNT/TDM + etc.

NAF/Q-Prop/IPG/TDM + etc.



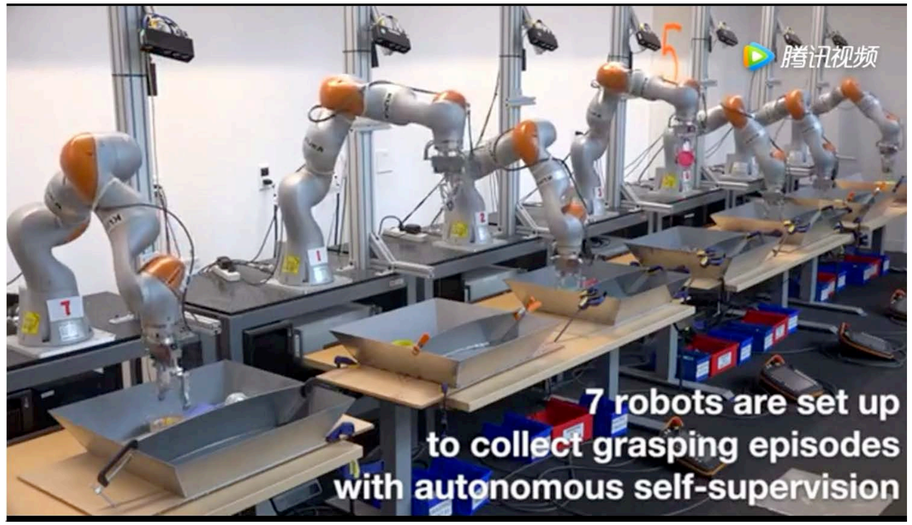
+ Simulation

+ Multi-task

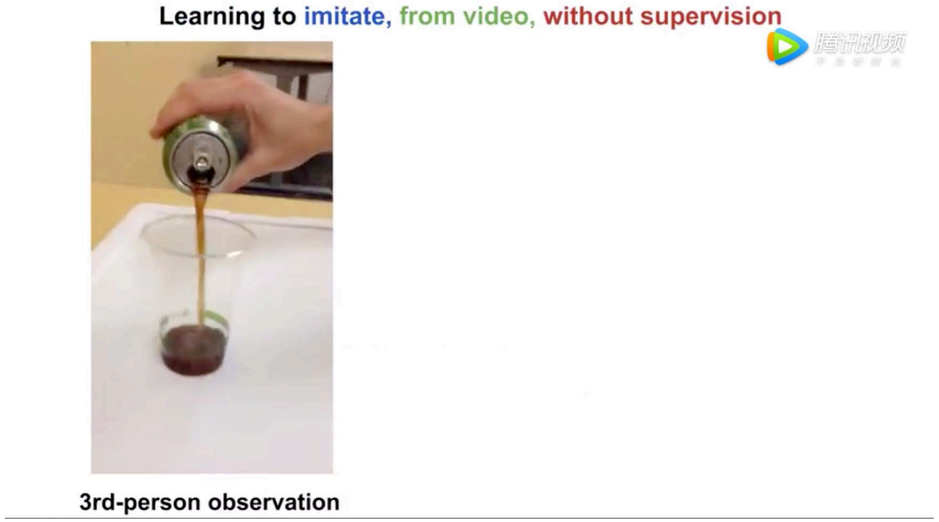
+ Imitation

+ Human-feedback

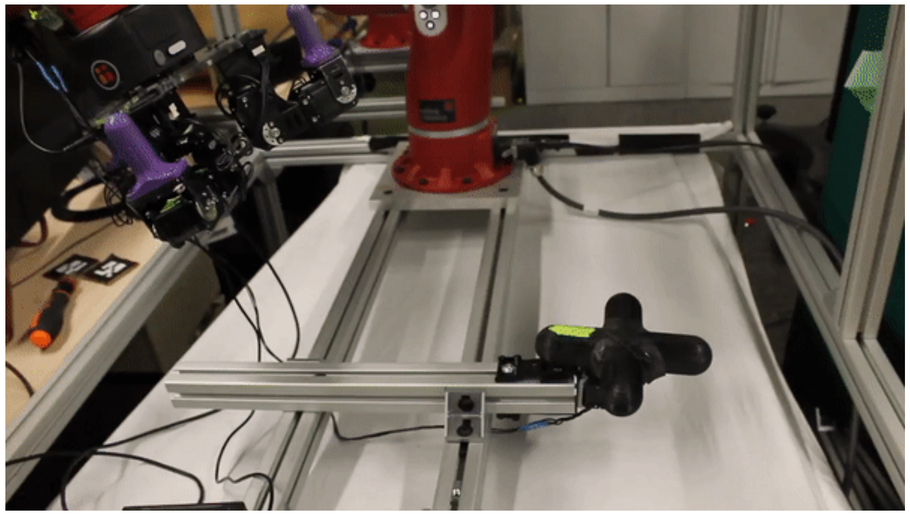
Other work from Google Brain Robotics Team



QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation [2018]



Time-Contrastive Networks: Self-Supervised Learning from Video [2017]



Dexterous Manipulation with Reinforcement Learning: Efficient, General, and Low-Cost [2018]

Check out more at: https://ai.google/research/teams/brain/robotics

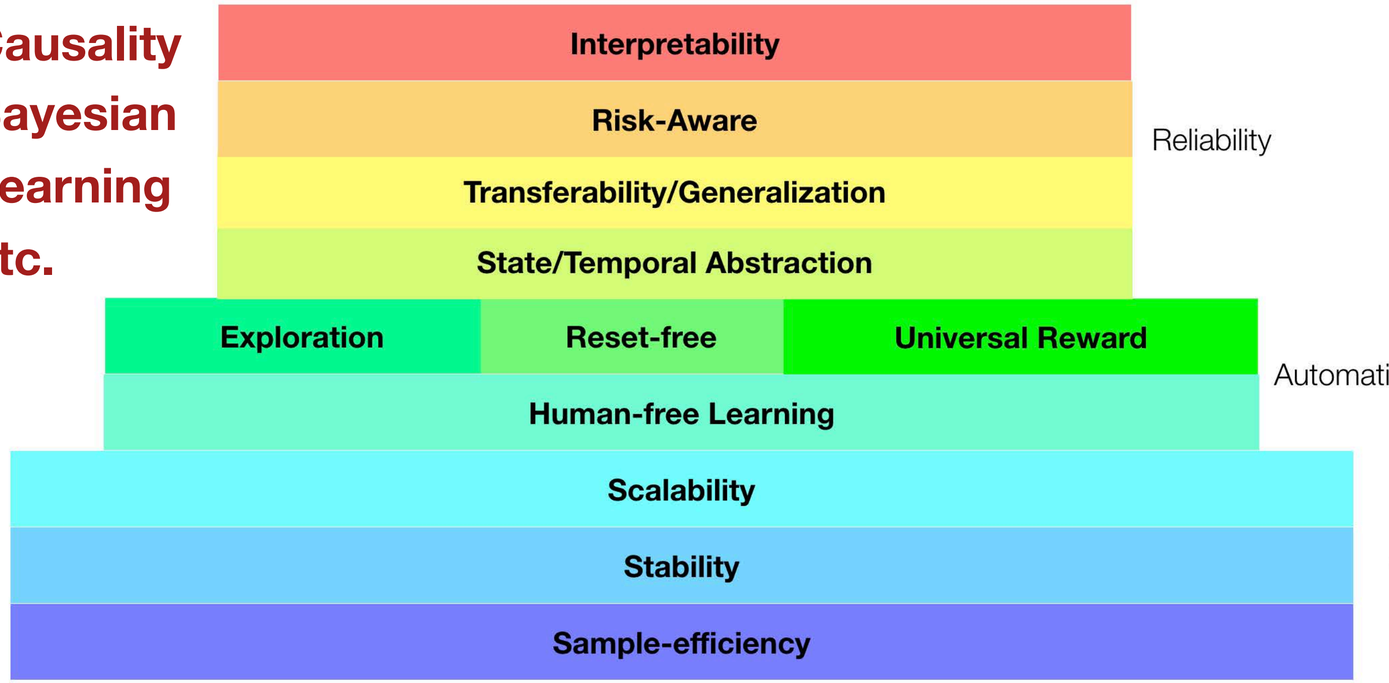
- Legged locomotion
- Navigation
- etc…

Discussion

Natural language, Causality
Distributional, Bayesian
Sim2Real, Meta learning
HIRO + etc.

LNT/TDM + etc.

NAF/Q-Prop/IPG/TDM + etc.



+ Simulation
+ Multi-task
+ Imitation
+ Human-feedback

Unleash them into the wild!

Thank you!

Contact: shanegu@google.com


Richard E. Turner, Zoubin Ghahramani


Sergey Levine, Vitchyr Pong

Timothy Lillicrap

Bernhard Schoelkopf





Ilya Sutskever (now at OpenAI), Ethan Holly, Ben Eysenbach, Ofir Nachum, Honglak Lee

…and other amazing colleagues from: Cambridge, MPI Tuebingen, Google Brain, and DeepMind