Hierarchical Reinforcement Learning (Part II)

Mayank Mittal

What are humans good at?

Let’s go and have lunch!

Let’s go and have lunch!

1. Exit ETZ building

2. Cross the street

3. Eat at mensa

Let’s go and have lunch!

1. Exit ETZ building

- ➔ Open door
- ➔ Walk to the lift
- ➔ Press button
- ➔ Wait for lift
- ➔ …..

2. Cross the street

- ➔ Find shortest route
- ➔ Walk safely
- ➔ Follow traffic rules
- ➔ …..

3. Eat at mensa

- ➔ Open door
- ➔ Wait in a queue
- ➔ Take food
- ➔ …..

What are humans good at?

Temporal abstraction

Let's go and have lunch!

1. Exit ETZ building

- ➔ Open door
- ➔ Walk to the lift
- ➔ Press button
- ➔ Wait for lift
- ➔ …..

2. Cross the street

- ➔ Find shortest route
- ➔ Walk safely
- ➔ Follow traffic rules
- ➔ …..

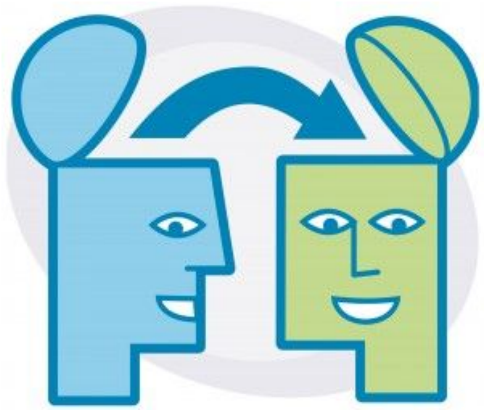
3. Eat at mensa

- ➔ Open door
- ➔ Wait in a queue
- ➔ Take food
- ➔ …..

What are humans good at?

Temporal abstraction

Transfer/Reusability of Skills



Let's go and have lunch!

1. Exit ETZ building

→ Open door
→ Walk to the lift
→ Press button
→ Wait for lift
→ …..

2. Cross the street

→ Find shortest route
→ Walk safely
→ Follow traffic rules
→ …..

3. Eat at mensa

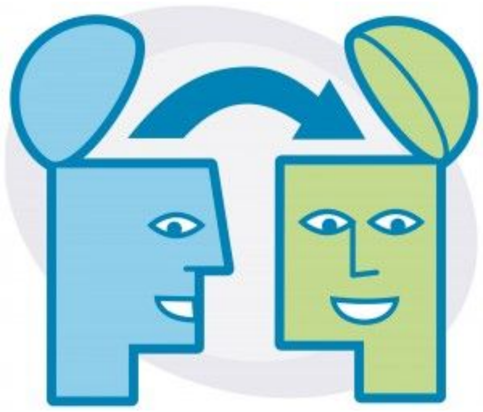
→ Open door
→ Wait in a queue
→ Take food
→ …..

How to represent these different goals?

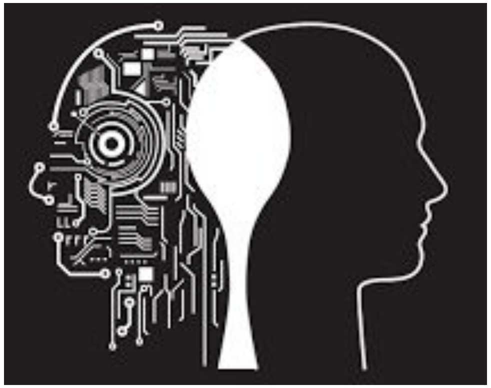
What are humans good at?

Temporal abstraction

Transfer/Reusability of Skills



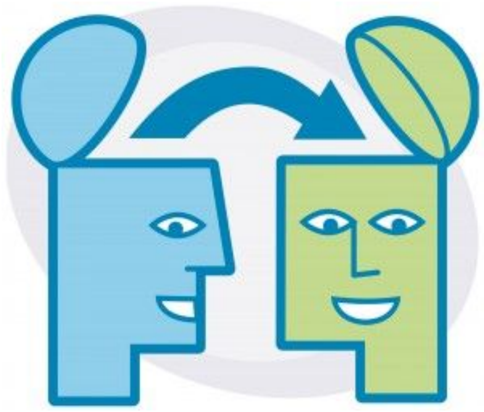
Powerful/meaningful state abstraction



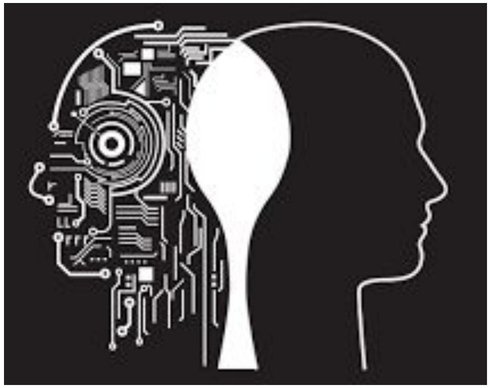
What are humans good at?

Temporal abstraction

Transfer/Reusability of Skills



Powerful/meaningful state abstraction



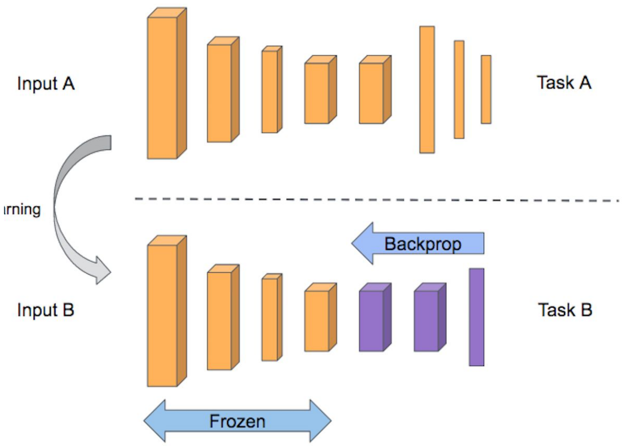
Can a learning-based agent do the same?

Promise of Hierarchical RL

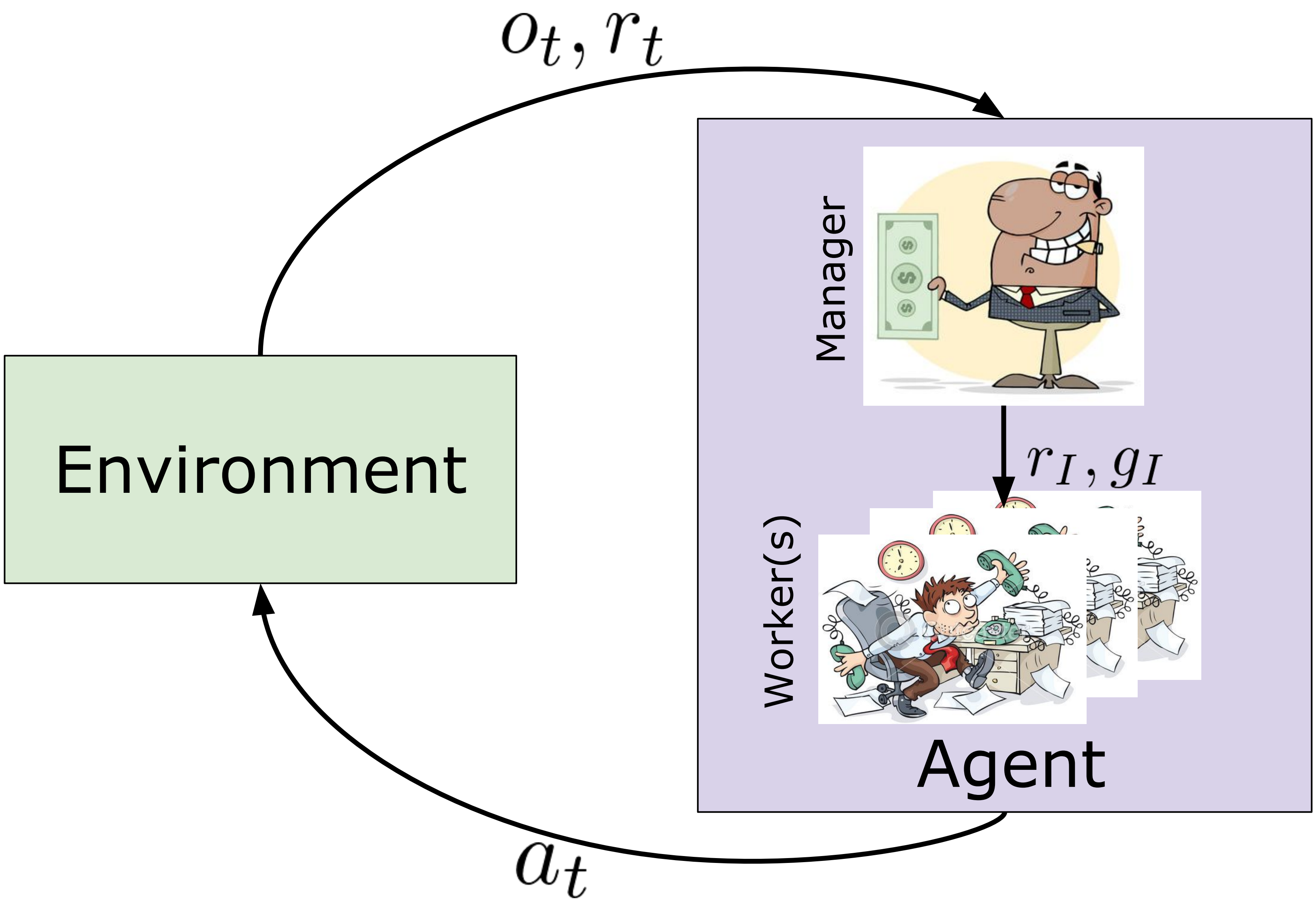
Structured exploration

Long-term credit assignment (and memory)

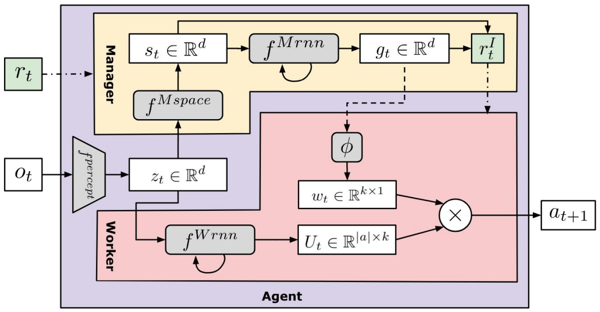
Transfer learning



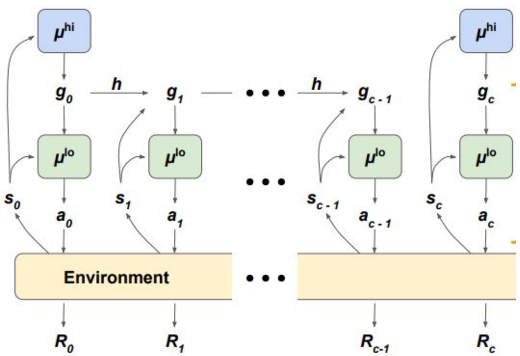
Hierarchical RL



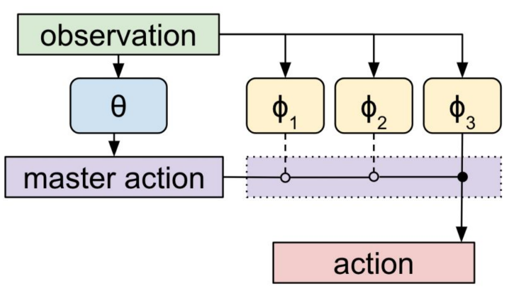
Hierarchical RL



FeUdal Networks for Hierarchical Reinforcement Learning (ICML 2017)

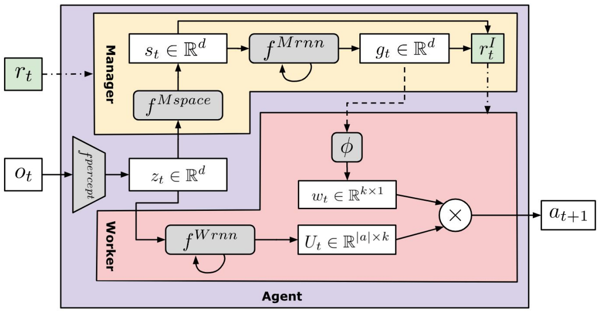


Data-Efficient Hierarchical Reinforcement Learning (NeurIPS 2018)



Meta-Learning Shared Hierarchies (ICLR 2018)

Hierarchical RL



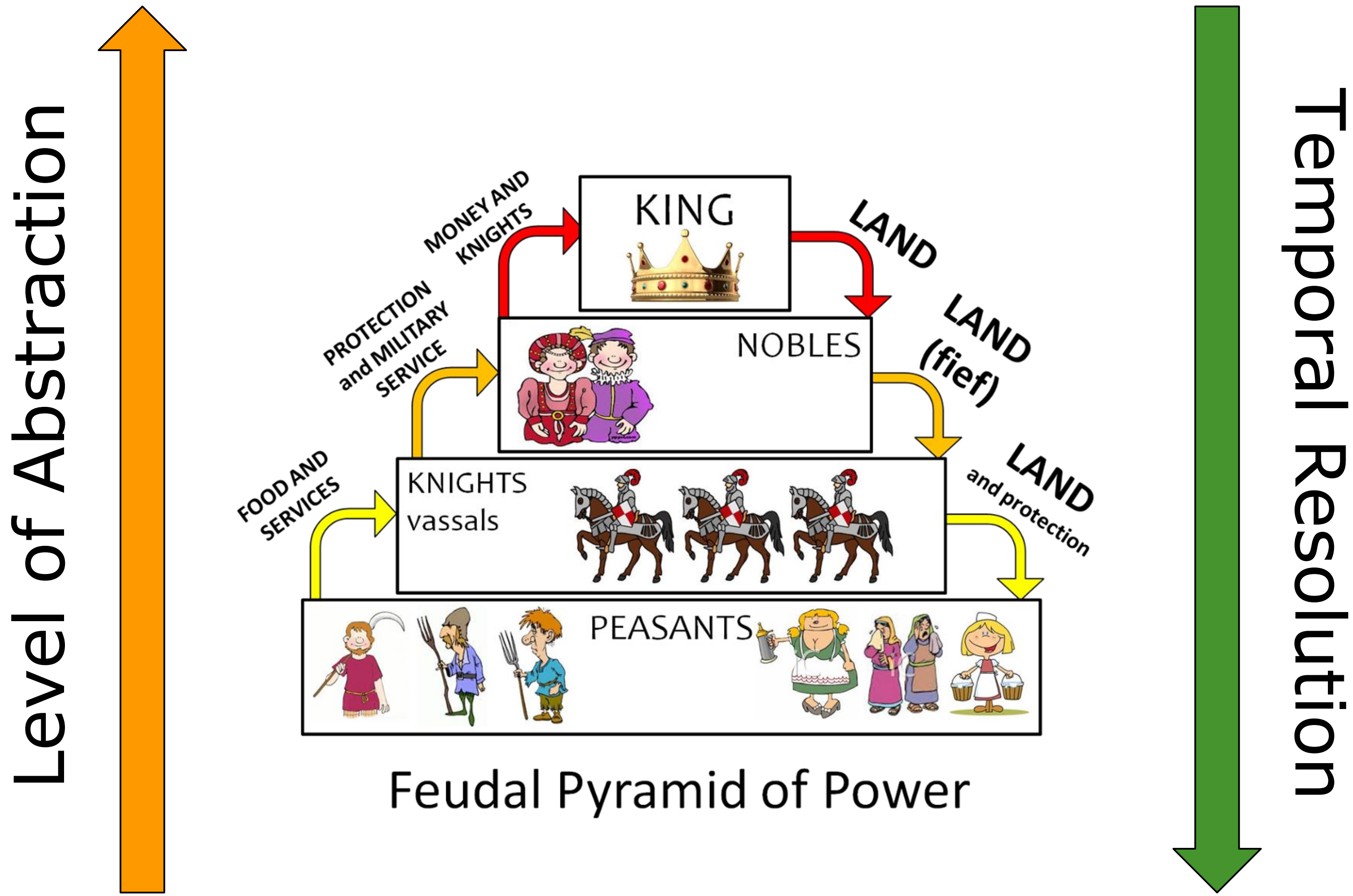
FeUdal Networks for Hierarchical Reinforcement Learning (ICML 2017)

Data-Efficient Hierarchical Reinforcement Learning (NeurIPS 2018)

Meta-Learning Shared Hierarchies (ICLR 2018)

FeUdal Networks (FUN)

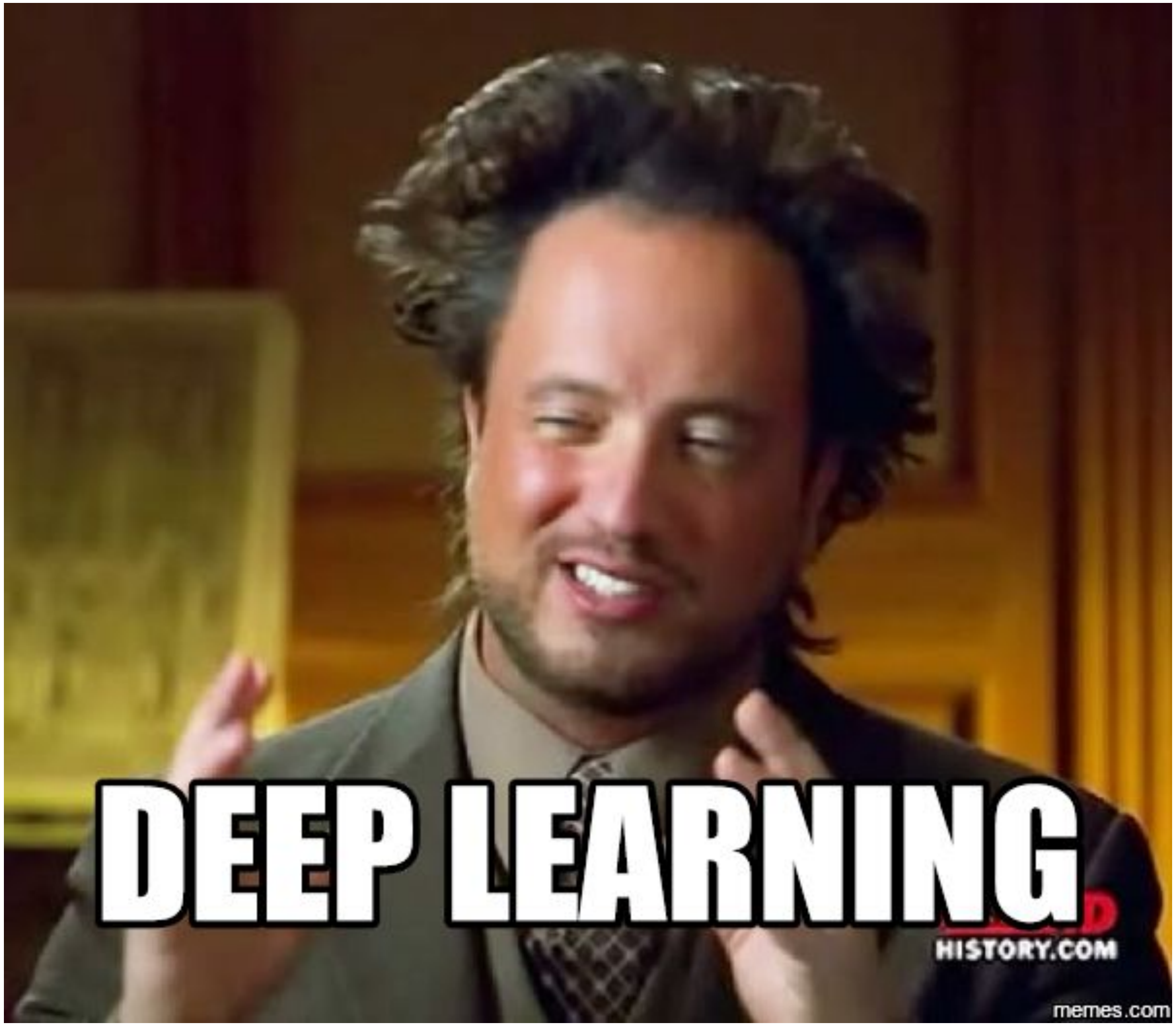
FeUdal Networks (FUN)



Feudal Pyramid of Power

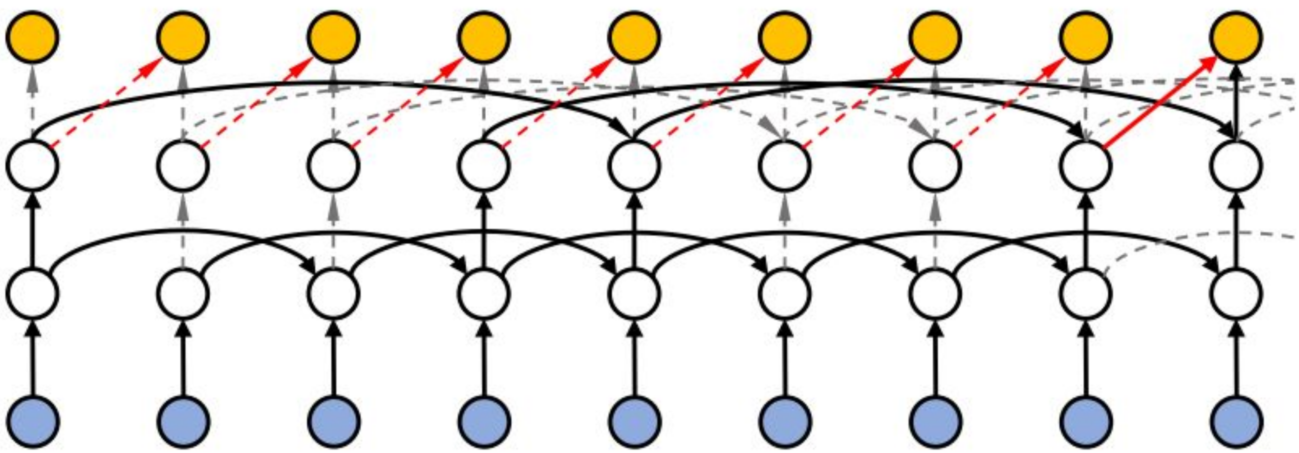
Dayan, Peter and Geoffrey E. Hinton. "Feudal Reinforcement Learning." NIPS (1992).

FeUdal Networks (FUN)



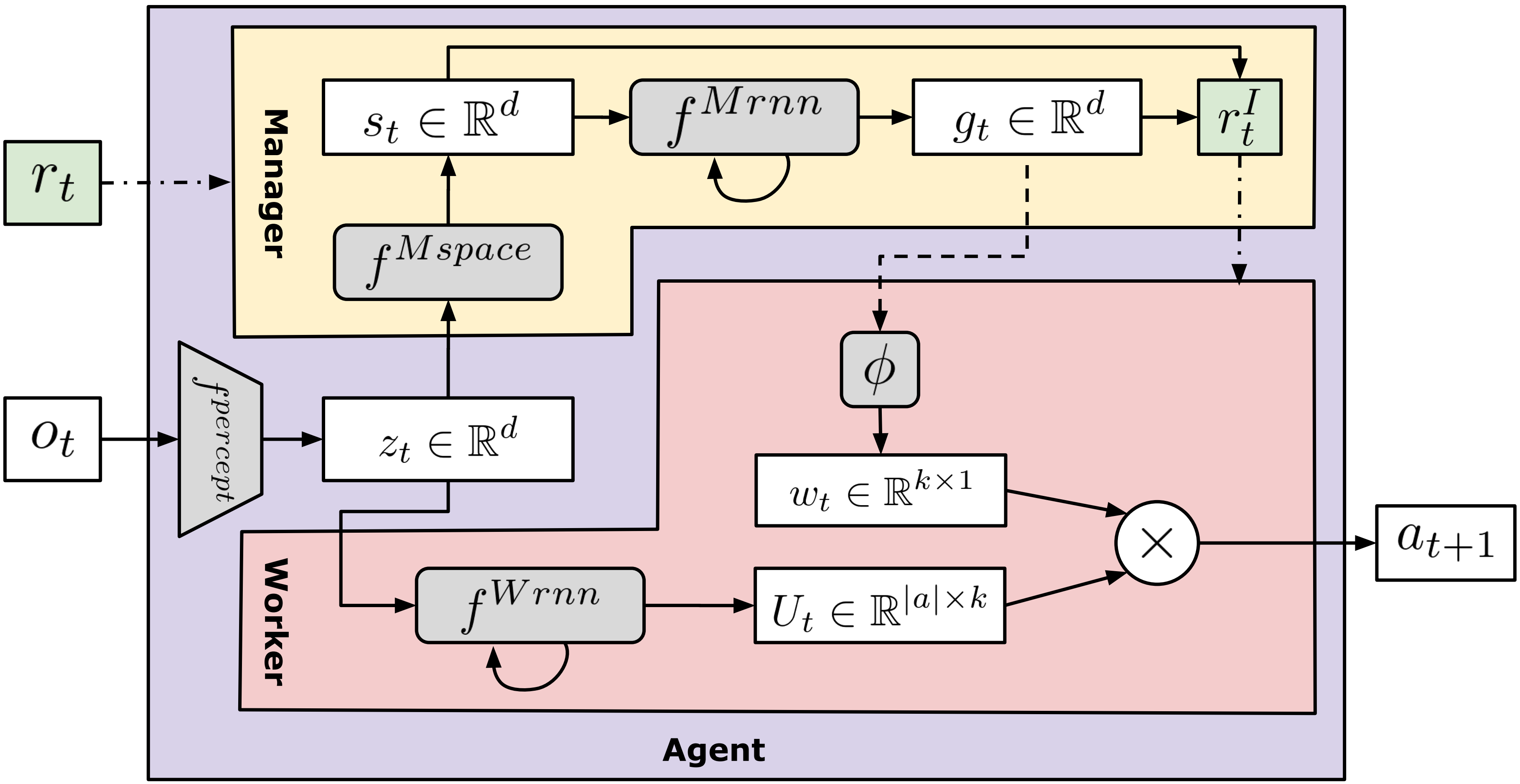
Detour: Dilated RNN

- Able to preserve memories over longer periods

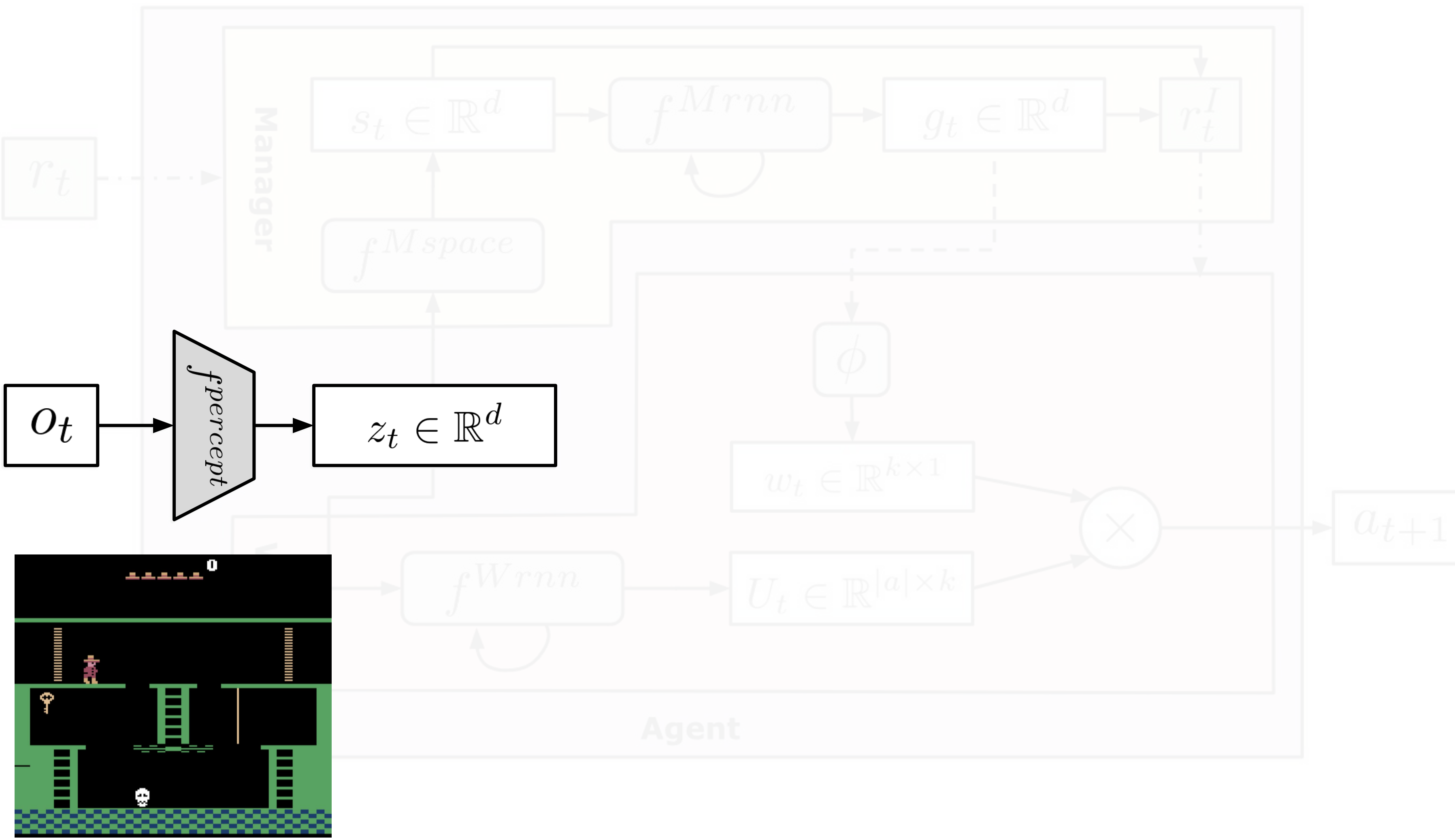


Chang, Shiyu et al. "Dilated Recurrent Neural Networks." *NIPS* (2017).

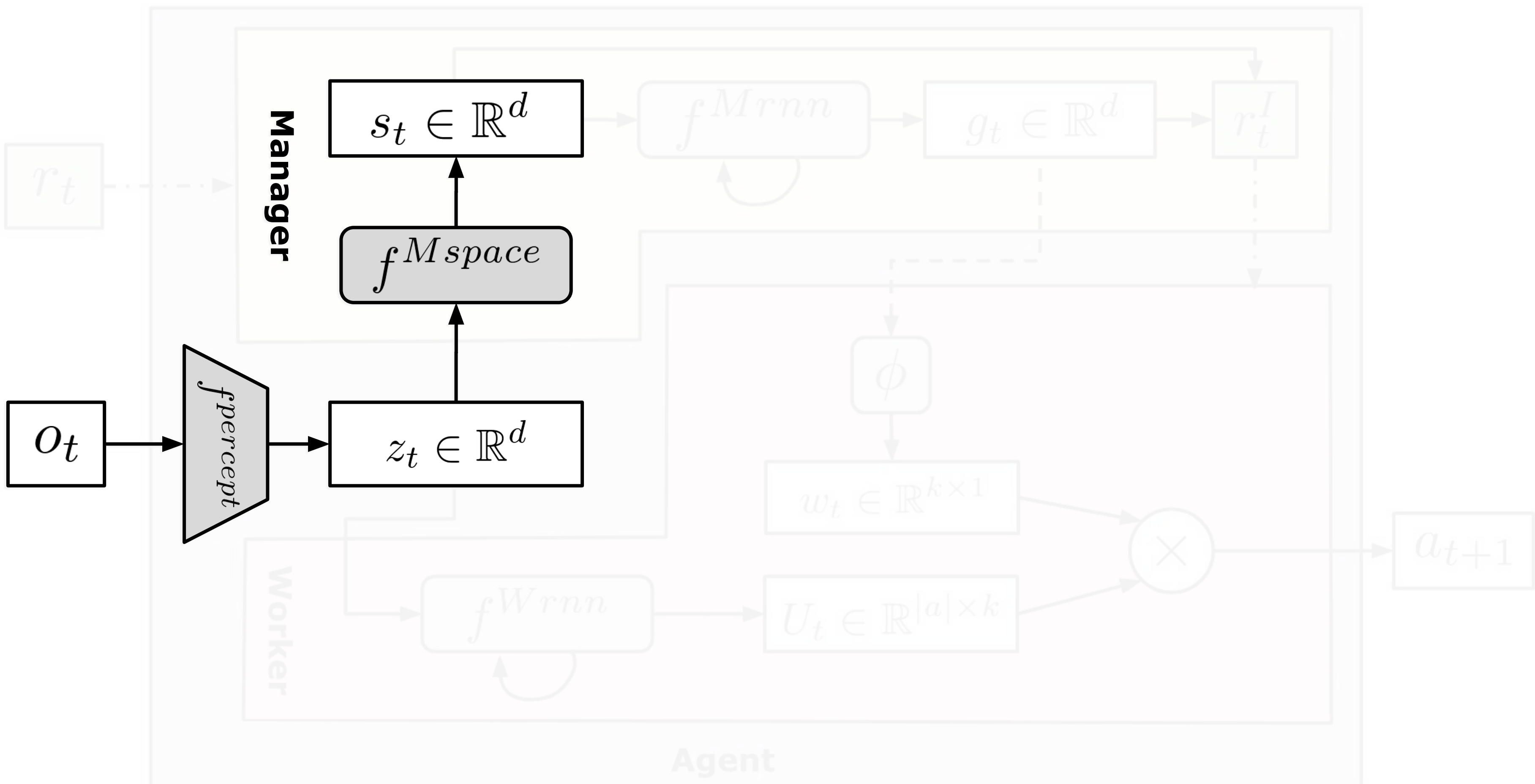
FeUdal Networks (FUN)



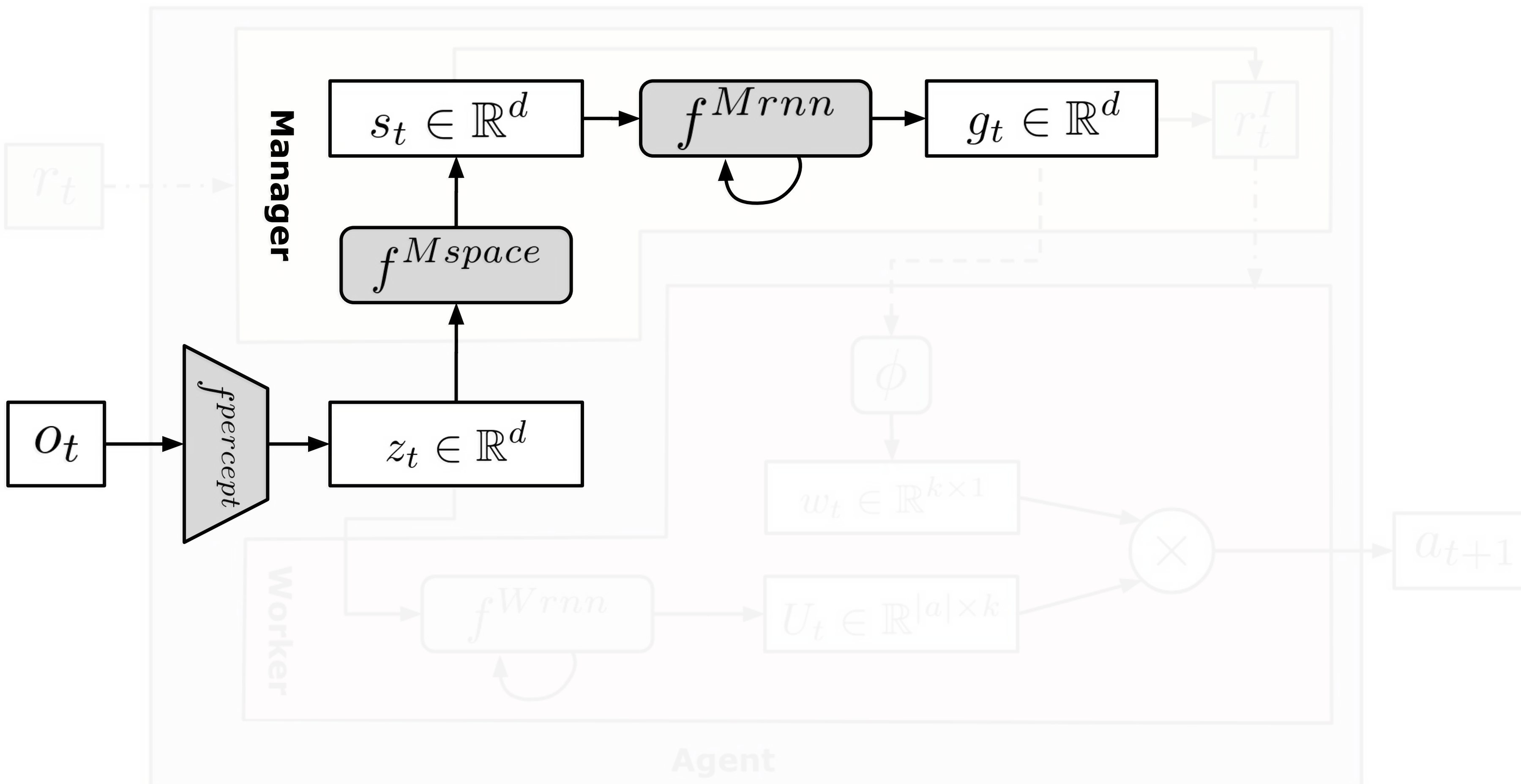
FeUdal Networks (FUN)



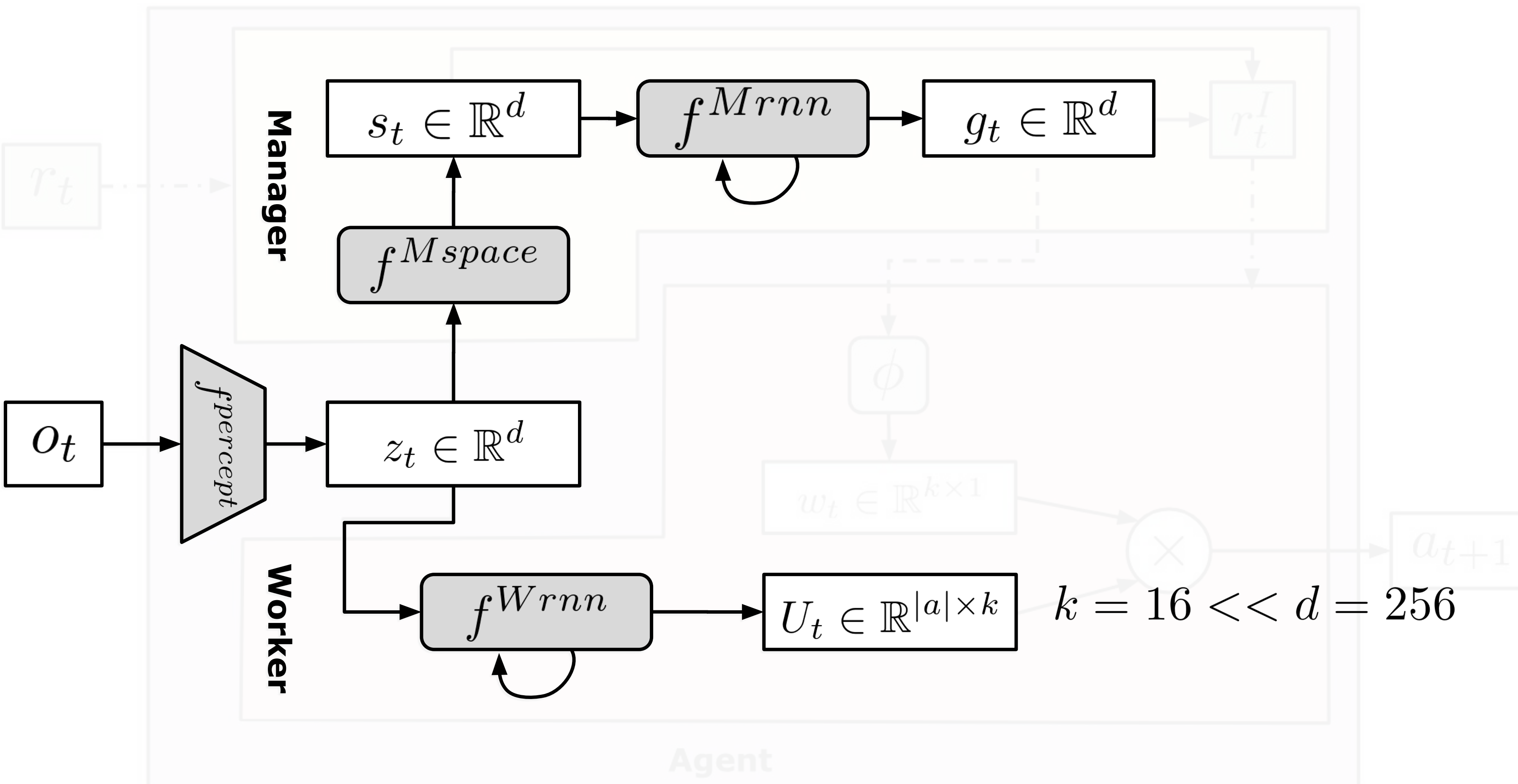
FeUdal Networks (FUN)



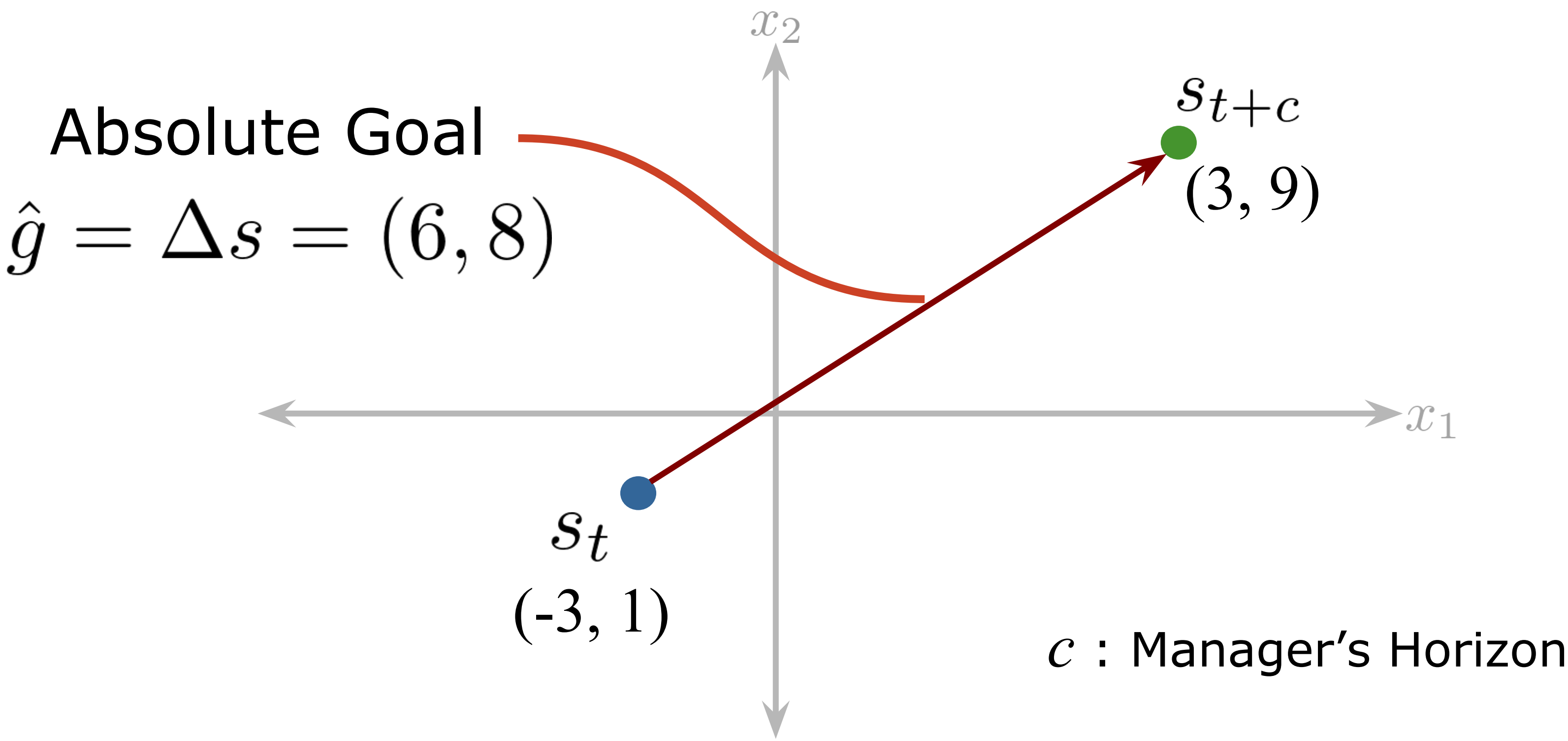
FeUdal Networks (FUN)



FeUdal Networks (FUN)



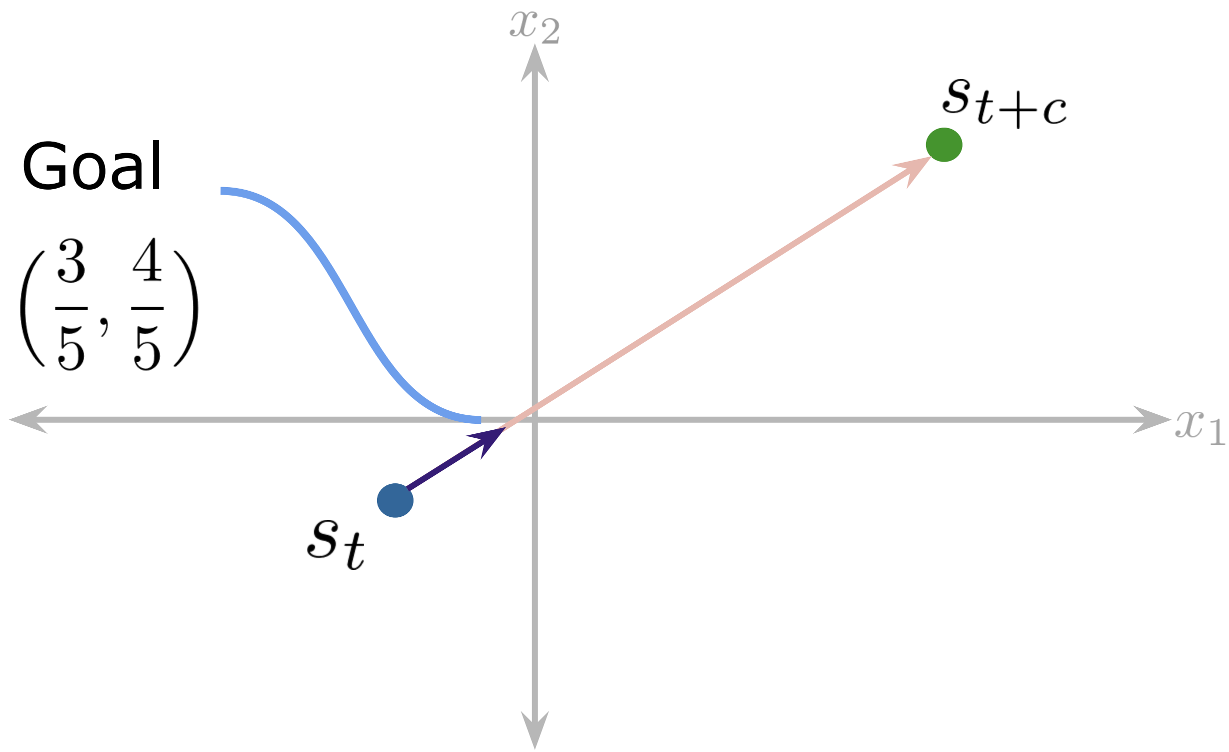
FeUdal Networks (FUN)



FeUdal Networks (FUN)

Directional Goal

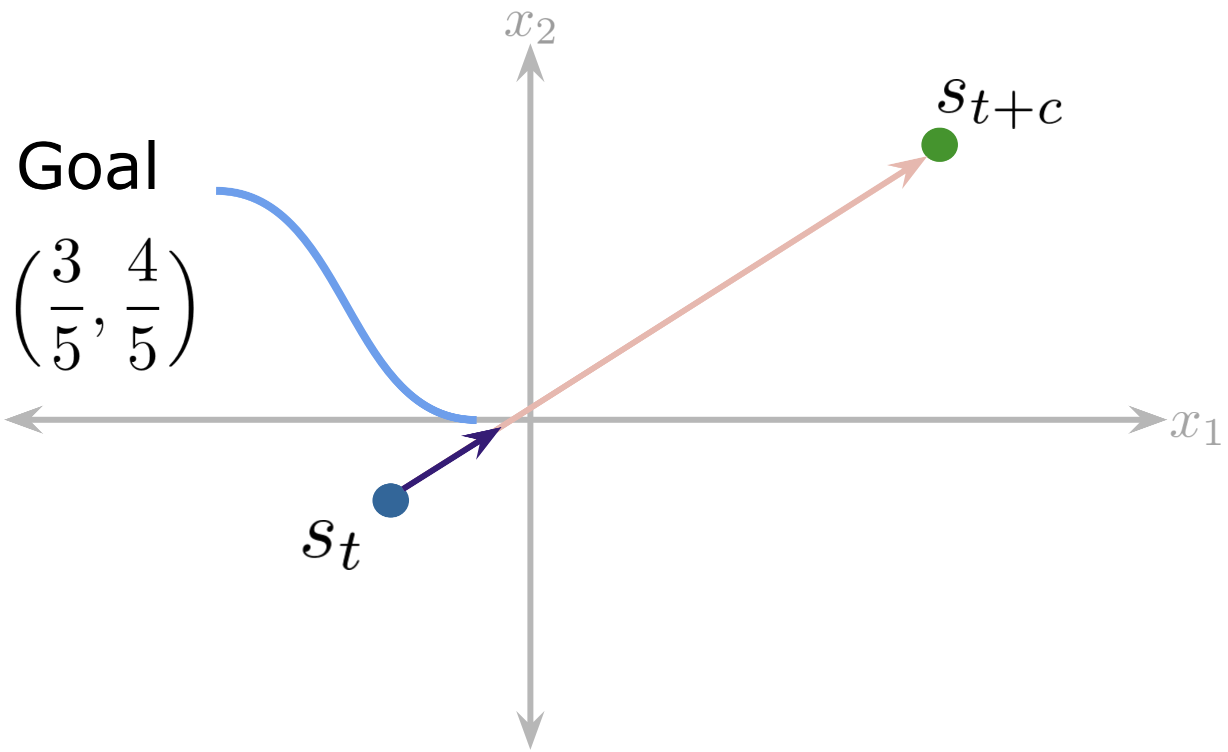
$$g = \frac{\hat{g}}{||\hat{g}||} = \left(\frac{3}{5}, \frac{4}{5}\right)$$



FeUdal Networks (FUN)

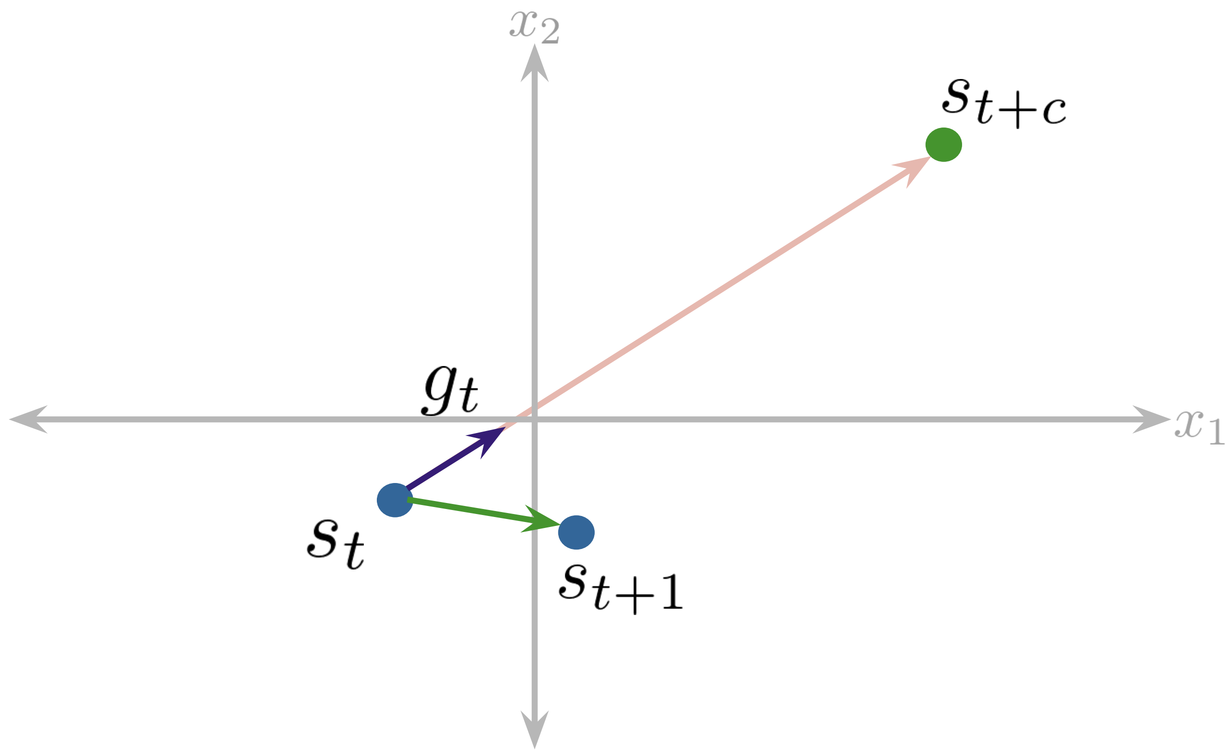
Directional Goal

$$g = \frac{\hat{g}}{||\hat{g}||} = \left(\frac{3}{5}, \frac{4}{5}\right)$$

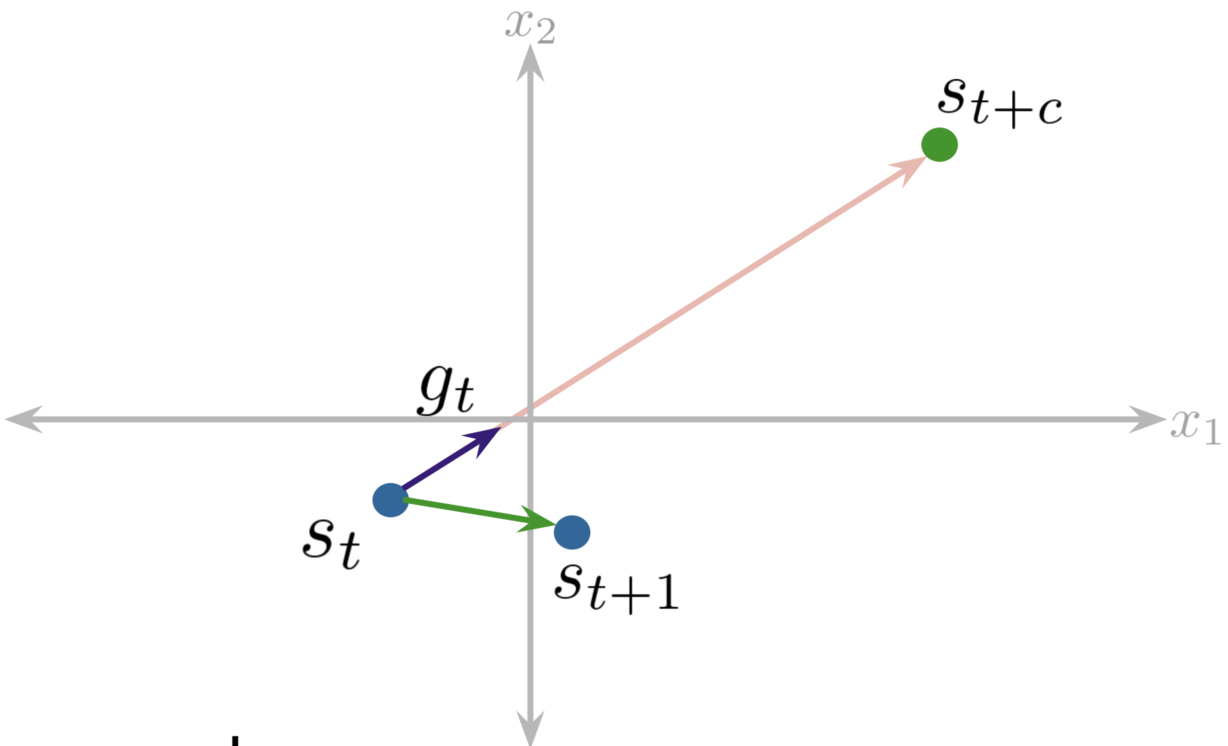


Idea: A single sub-goal (direction) can be reused in many different locations in state space

FeUdal Networks (FUN)



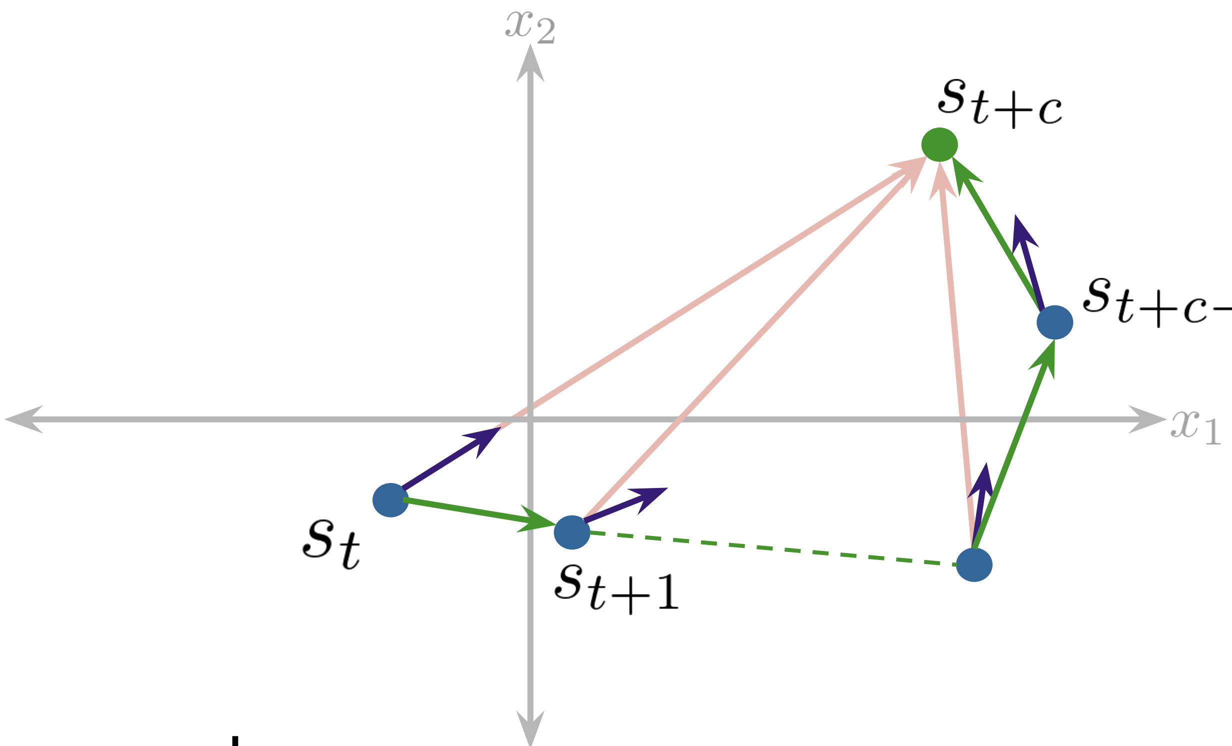
FeUdal Networks (FUN)



- Intrinsic reward

$$d_{cos}(s_{t+1} - s_t, g_t) = \frac{(s_{t+1} - s_t)^T g_t}{|s_{t+1} - s_t||g_t|}$$

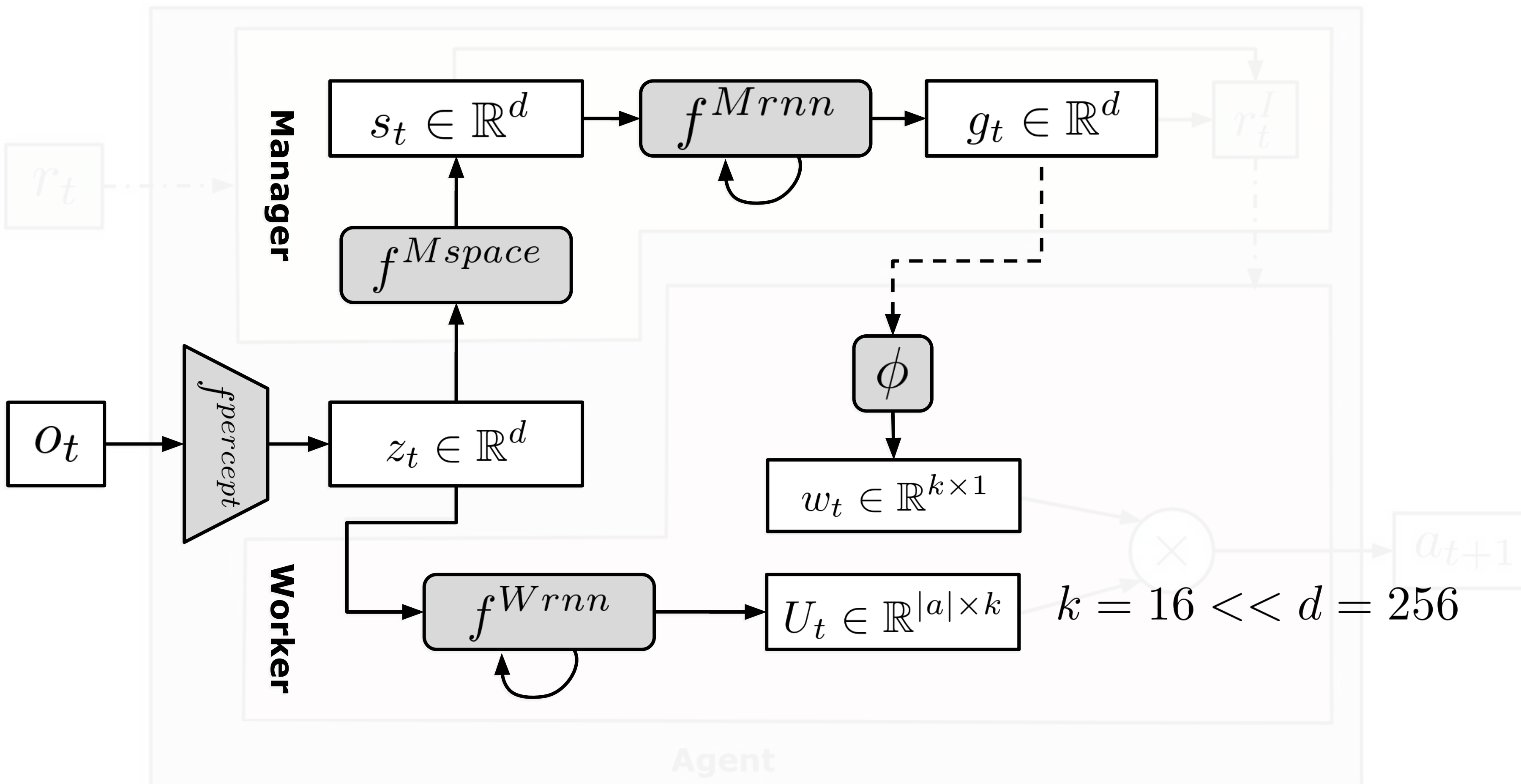
FeUdal Networks (FUN)



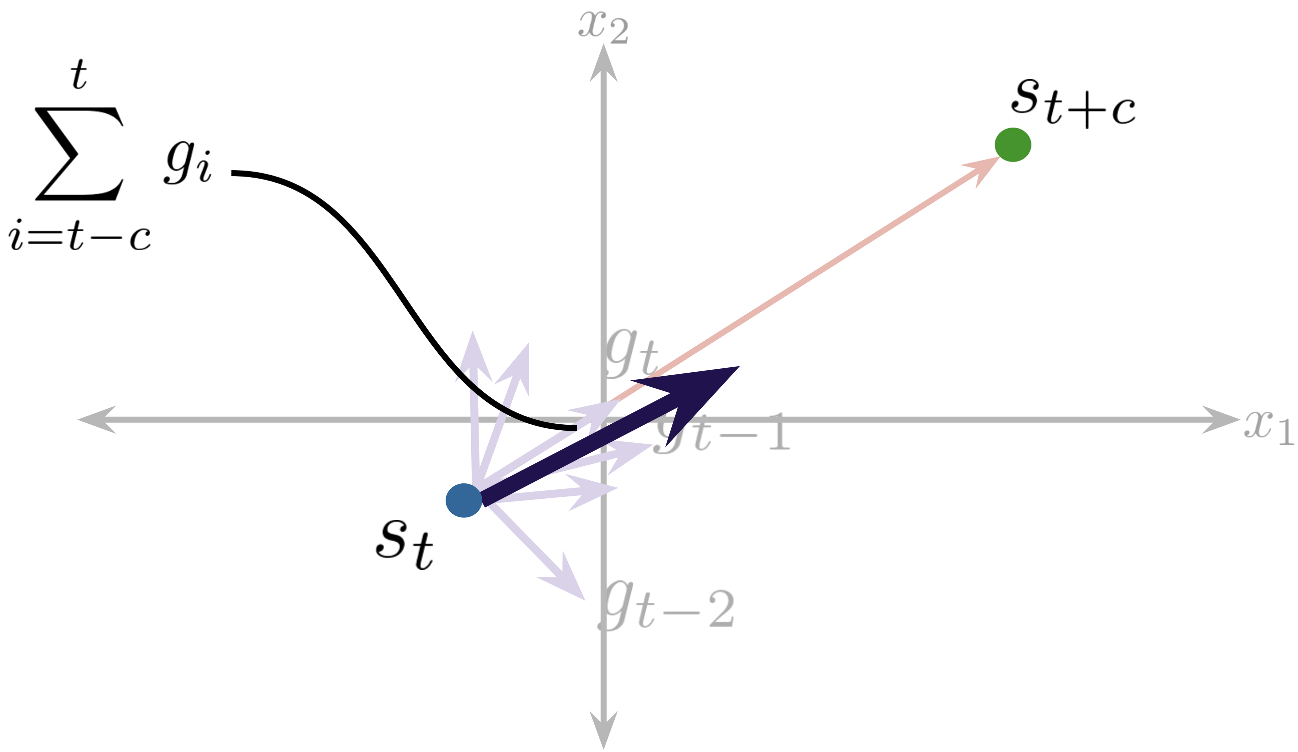
- Intrinsic reward

$$r^I_{t+c} = \frac{1}{c}\sum_{i=t}^{t+c} d_{cos}(s_{t+c} - s_i, g_i)$$

FeUdal Networks (FUN)



FeUdal Networks (FUN)



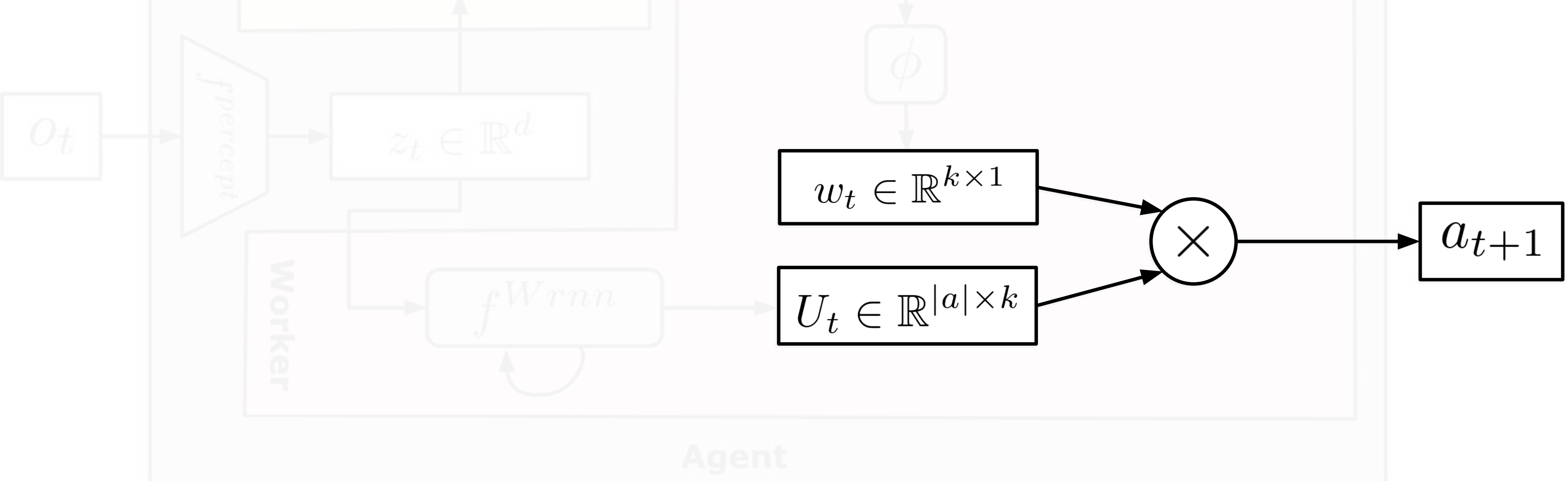
$$w_t = \phi\left(\sum_{i=t-c}^{t} g_i\right)$$

FeUdal Networks (FUN)

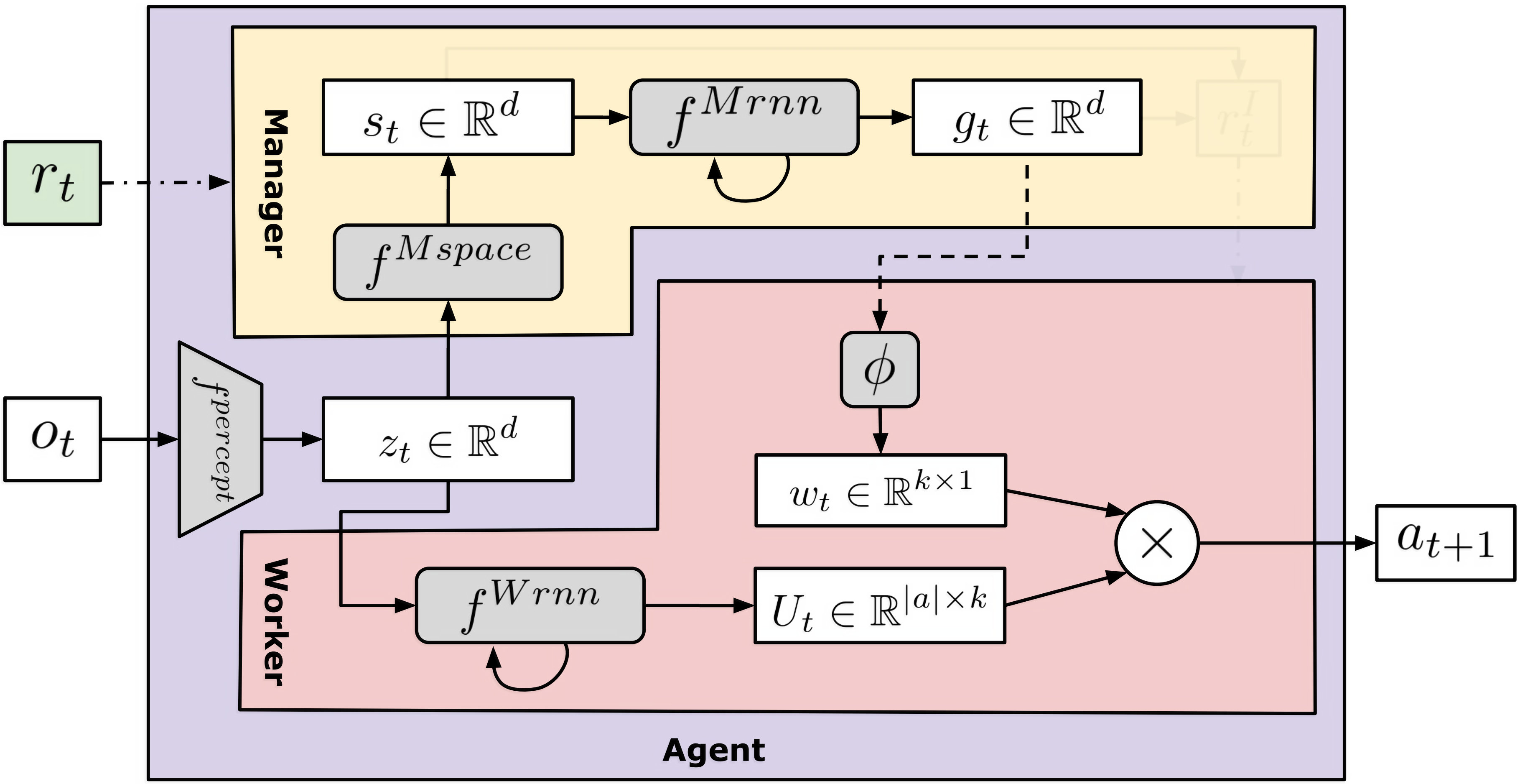
- Action Sampling

$$\pi_{t+1} = \mathrm{softmax}(U_t w_t)$$
$$a_{t+1} = \mathrm{argmax}_a \pi_{t+1}$$

Stochastic Policy!

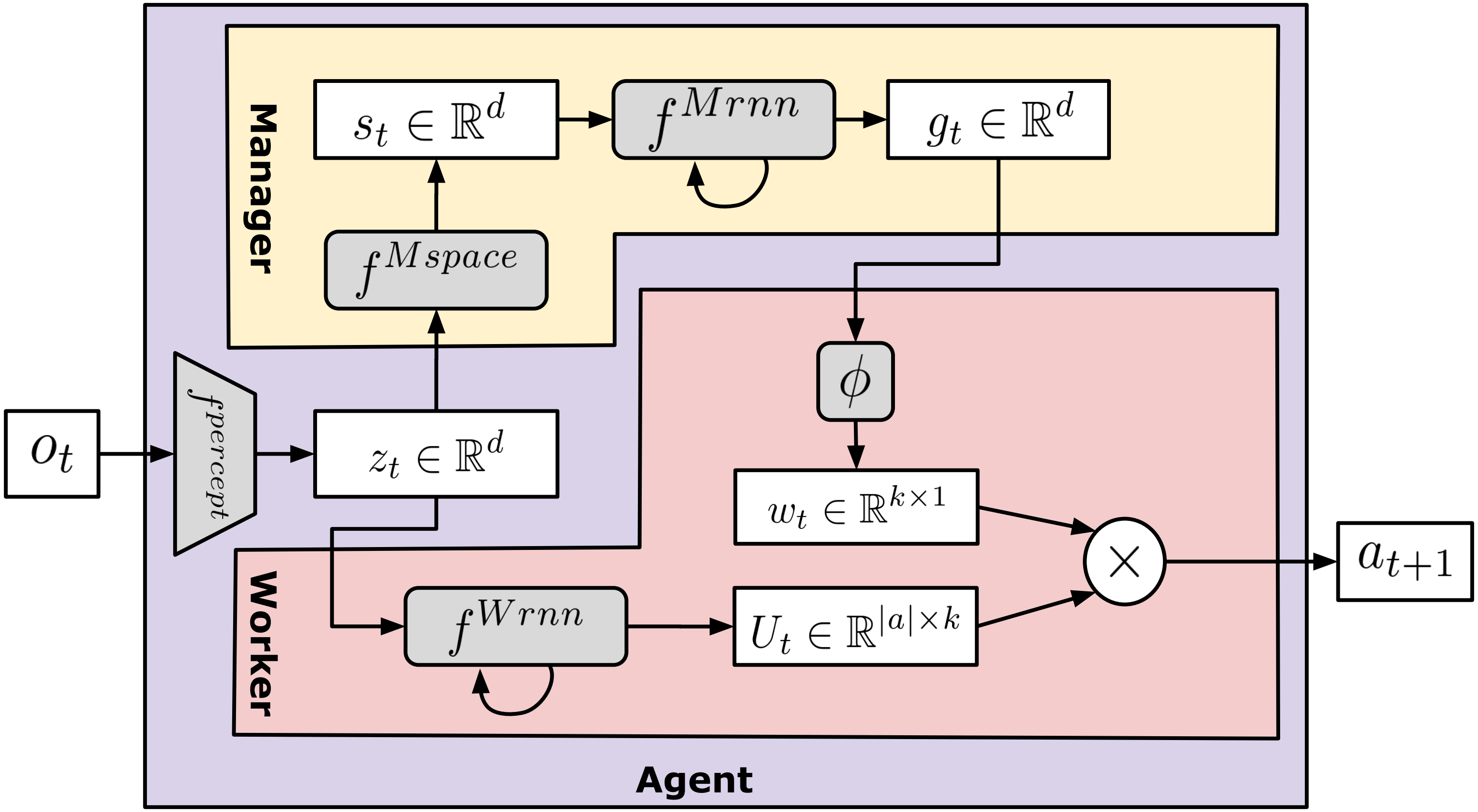


FeUdal Networks (FUN)



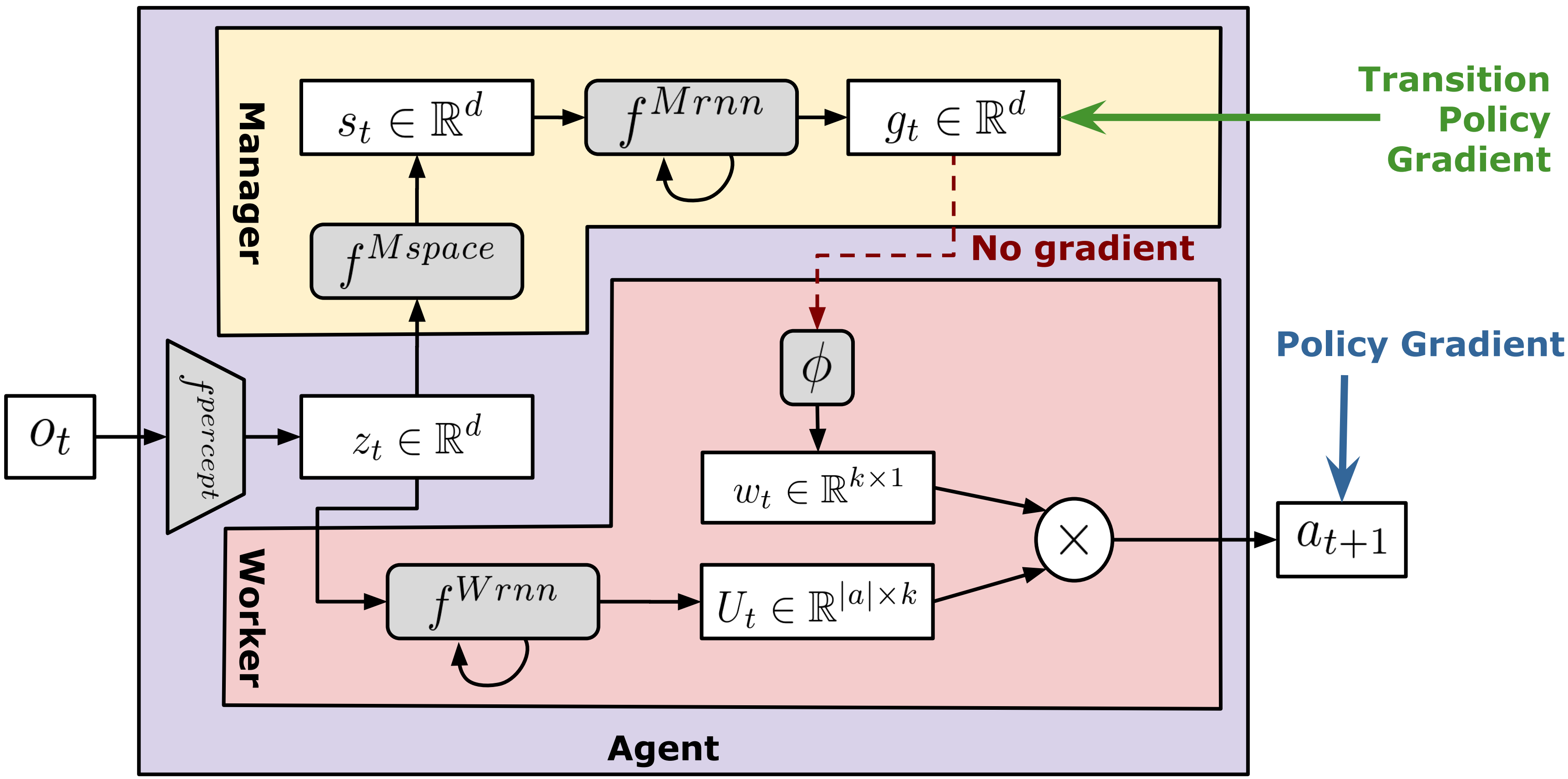
FeUdal Networks (FUN)

Why not do end-to-end learning?



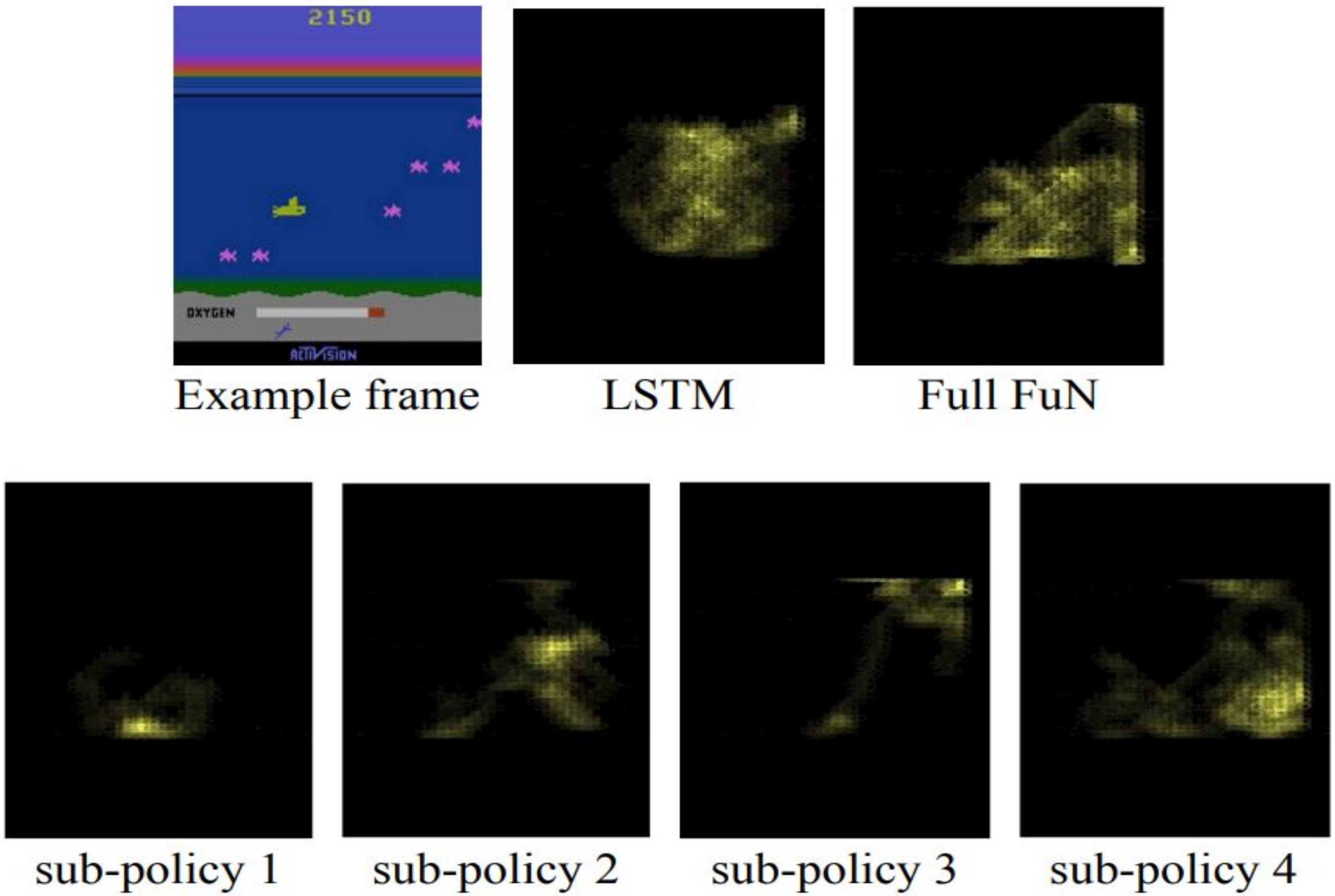
FeUdal Networks (FUN)

Manager & Worker: Separate Actor-Critic



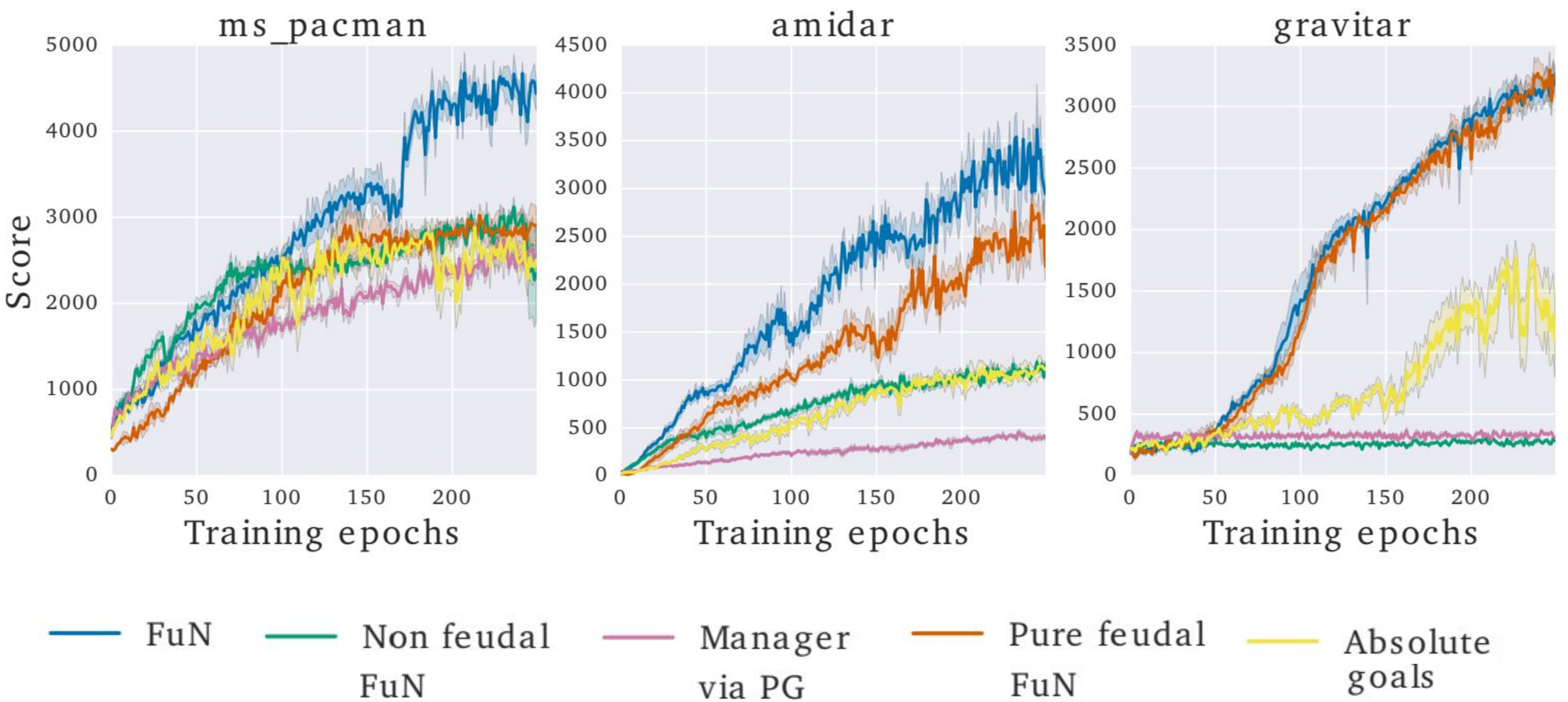
FeUdal Networks (FUN)

Qualitative Analysis



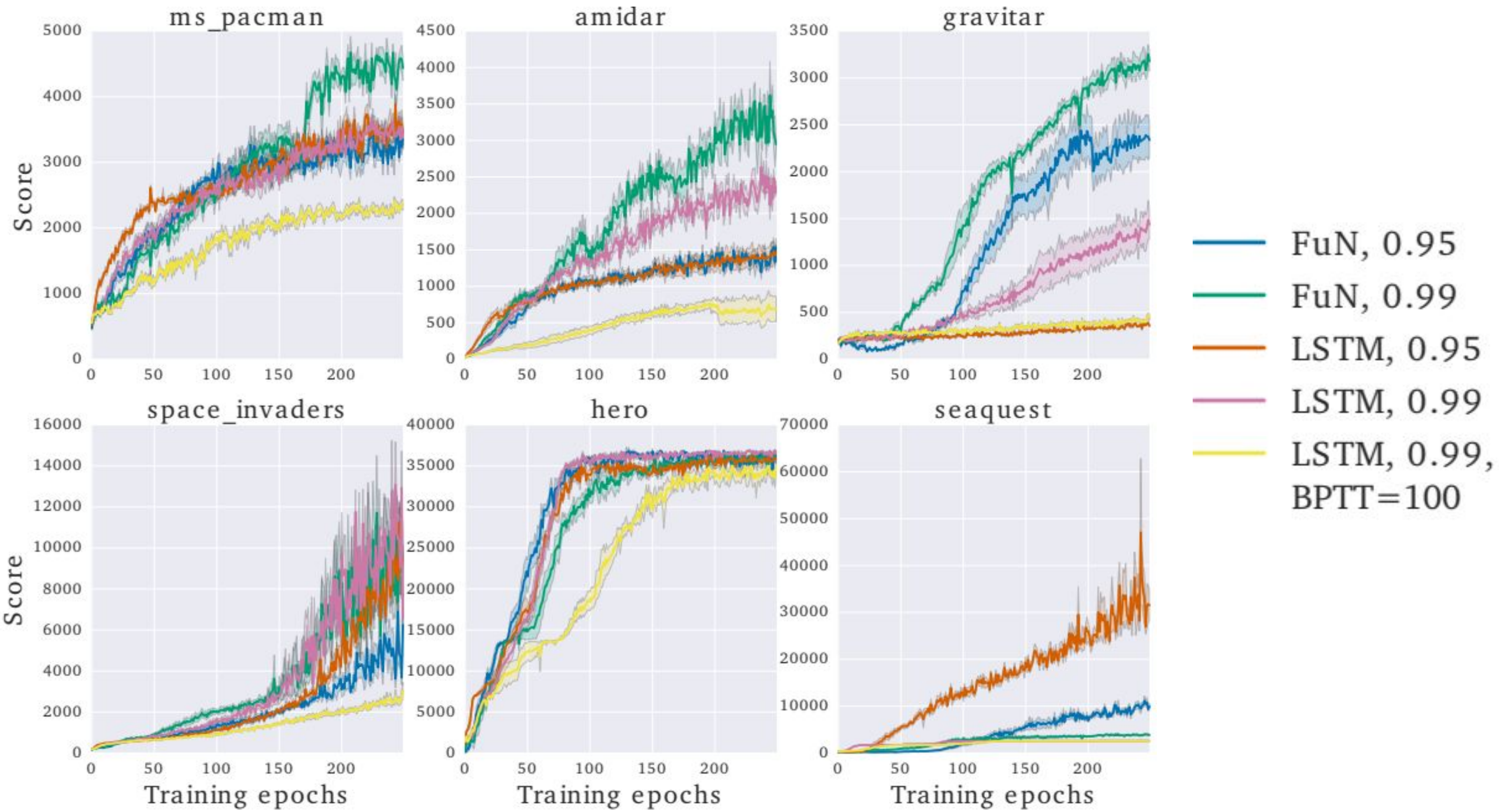
FeUdal Networks (FUN)

Ablative Analysis



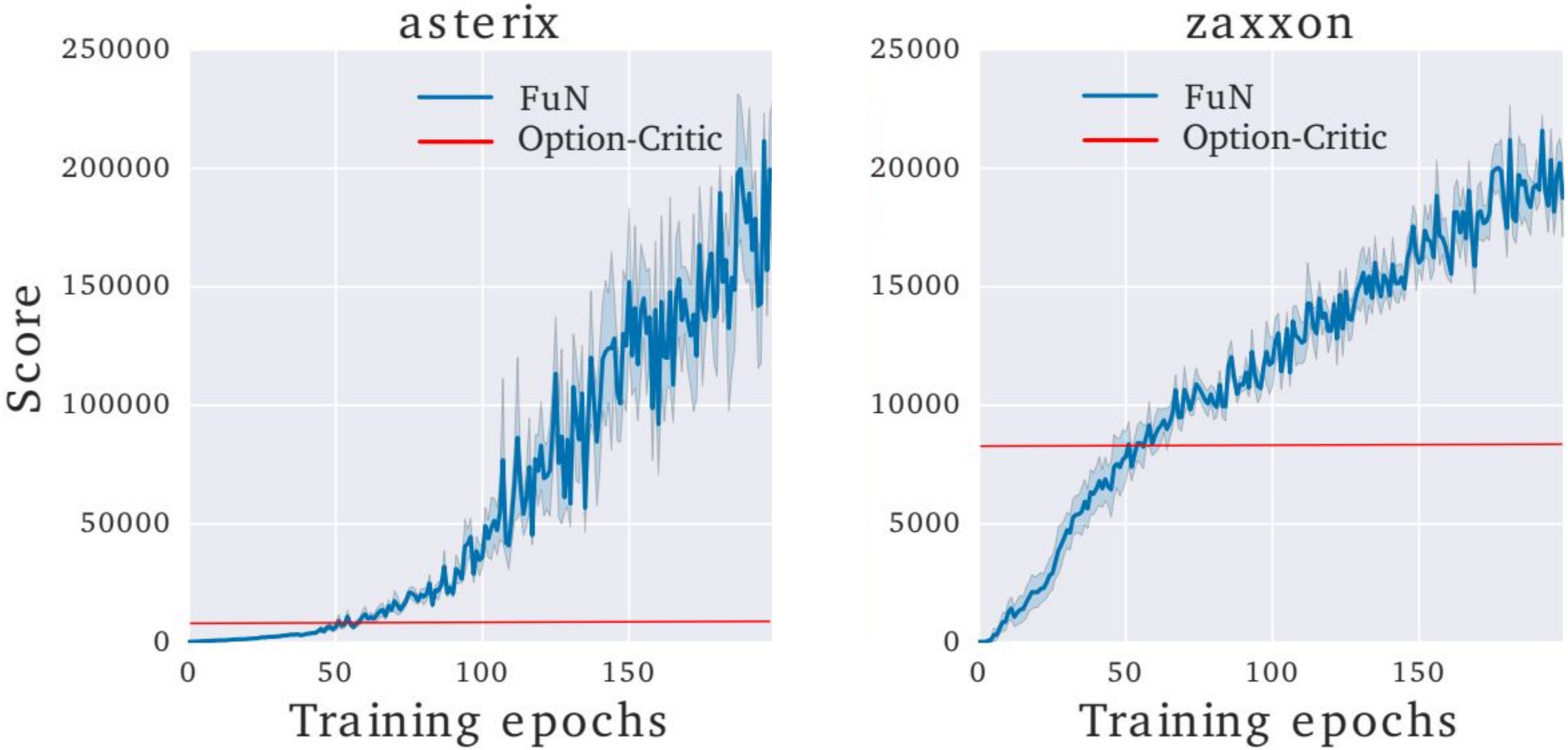
FeUdal Networks (FUN)

Ablative Analysis



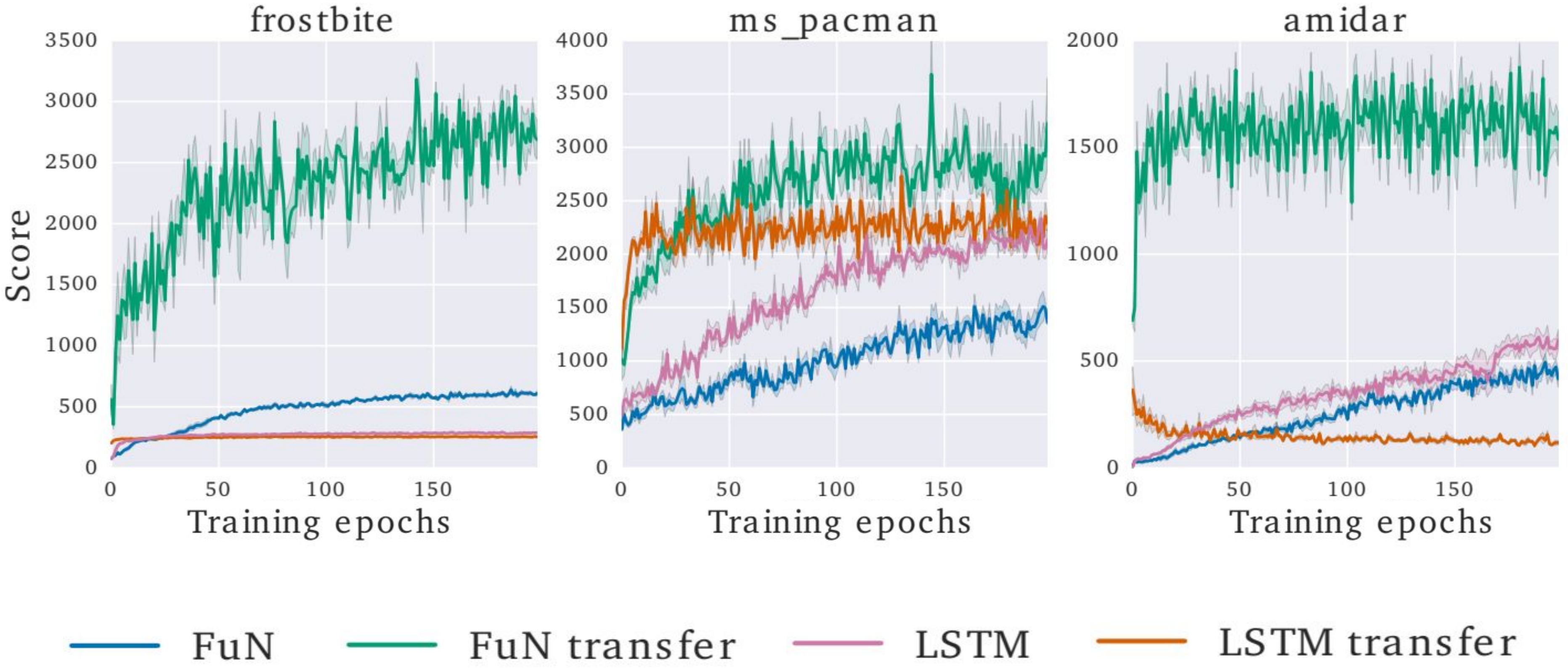
FeUdal Networks (FUN)

Comparison



FeUdal Networks (FUN)

Action Repeat Transfer

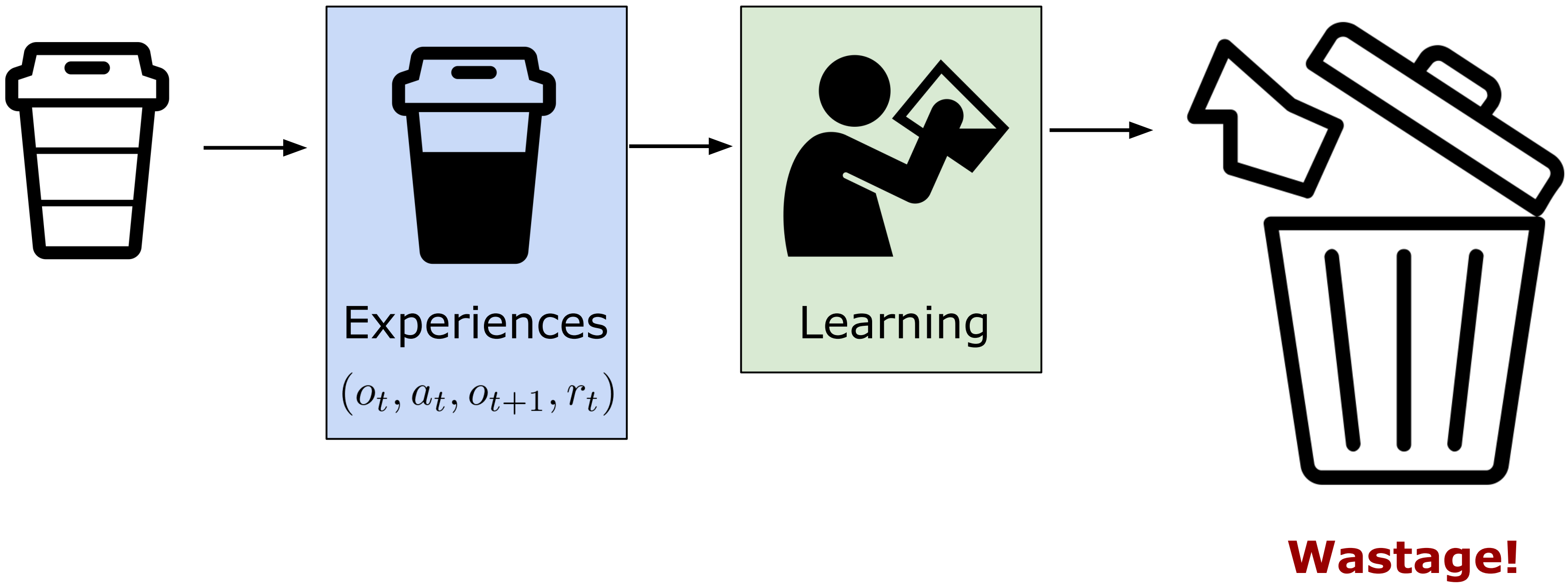


FeUdal Networks (FUN)

On-Policy Learning ☹️

FeUdal Networks (FUN)

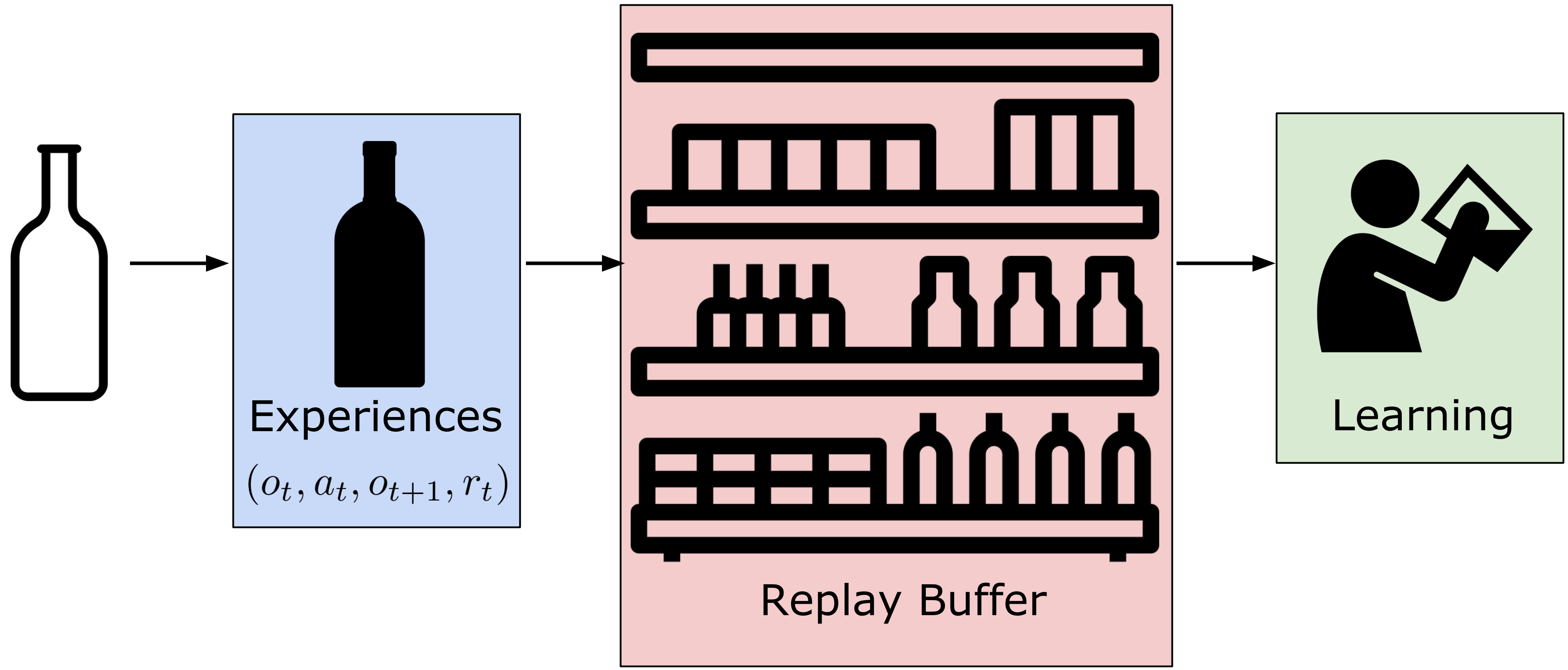
On-Policy Learning ☹️



Can we do better?

Can we do better?

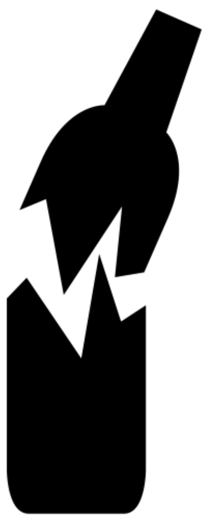
Off-Policy Learning 🙂



Reusage!

Can we do better?

Off-Policy Learning 🙃

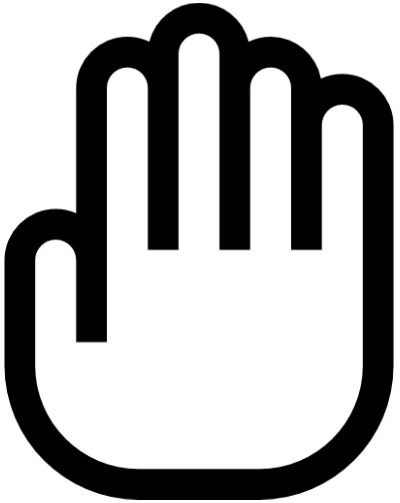


Unstable Learning

Can we do better?

Off-Policy Learning 🙃

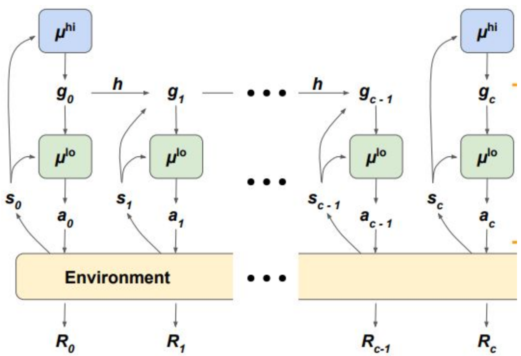
Unstable Learning



To-Be-Disclosed

Hierarchical RL

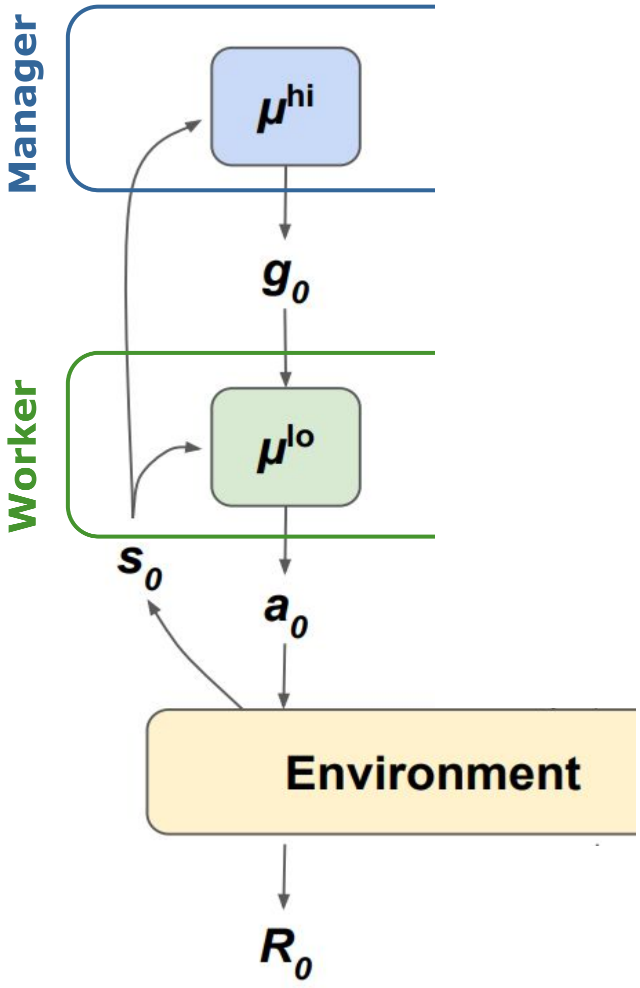
FeUdal Networks for Hierarchical Reinforcement Learning (ICML 2017)



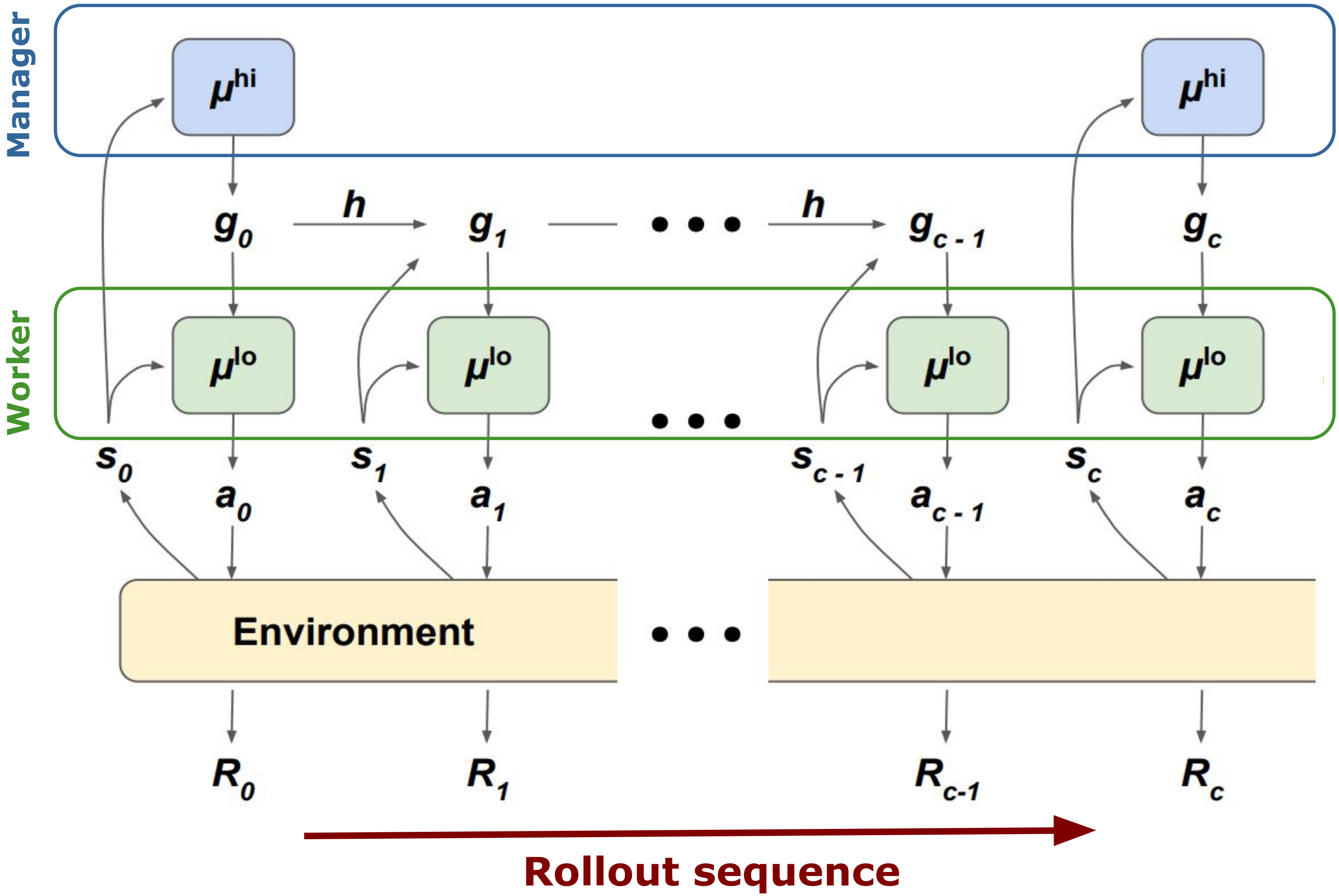
Data-Efficient Hierarchical Reinforcement Learning
(NeurIPS 2018)

Meta-Learning Shared Hierarchies (ICLR 2018)

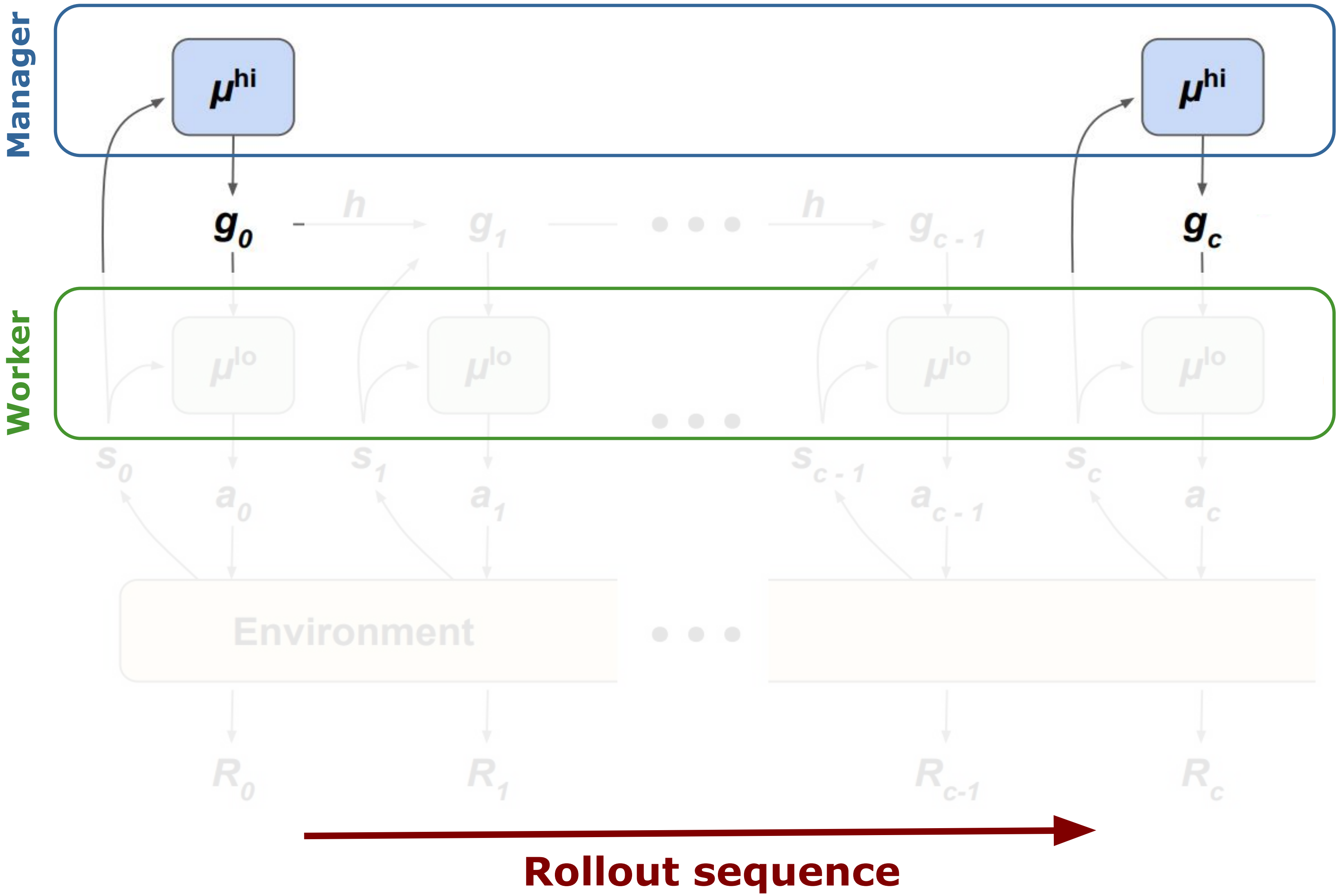
Data-Efficient HRL (HIRO)



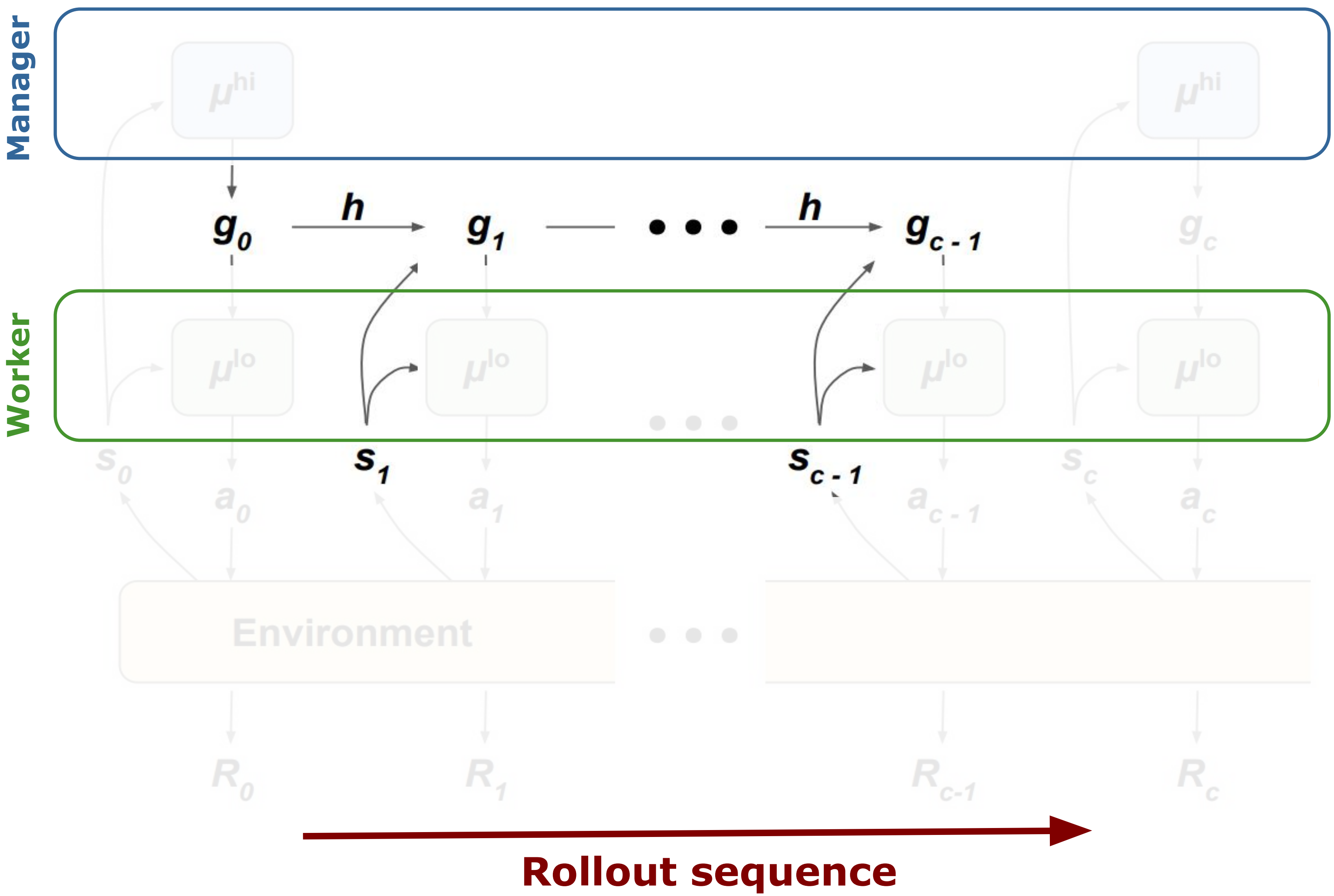
Data-Efficient HRL (HIRO)



Data-Efficient HRL (HIRO)

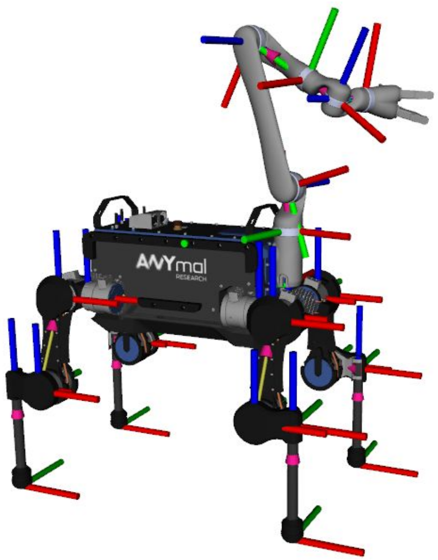


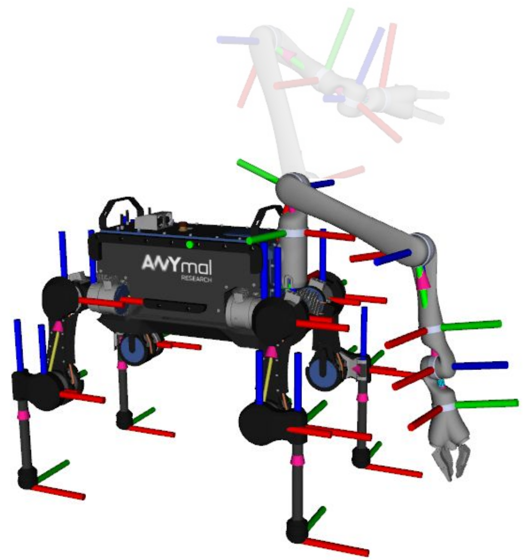
Data-Efficient HRL (HIRO)

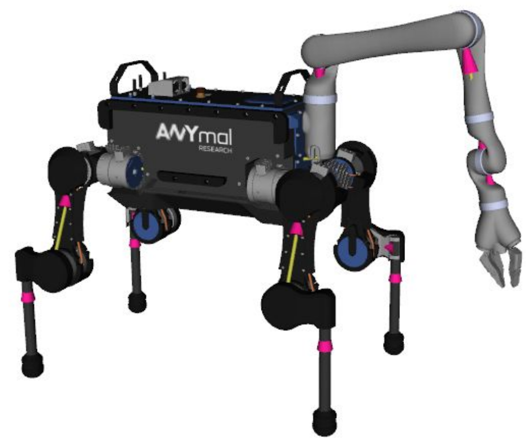


Data-Efficient HRL (HIRO)

Input | **Goal** | **Action**





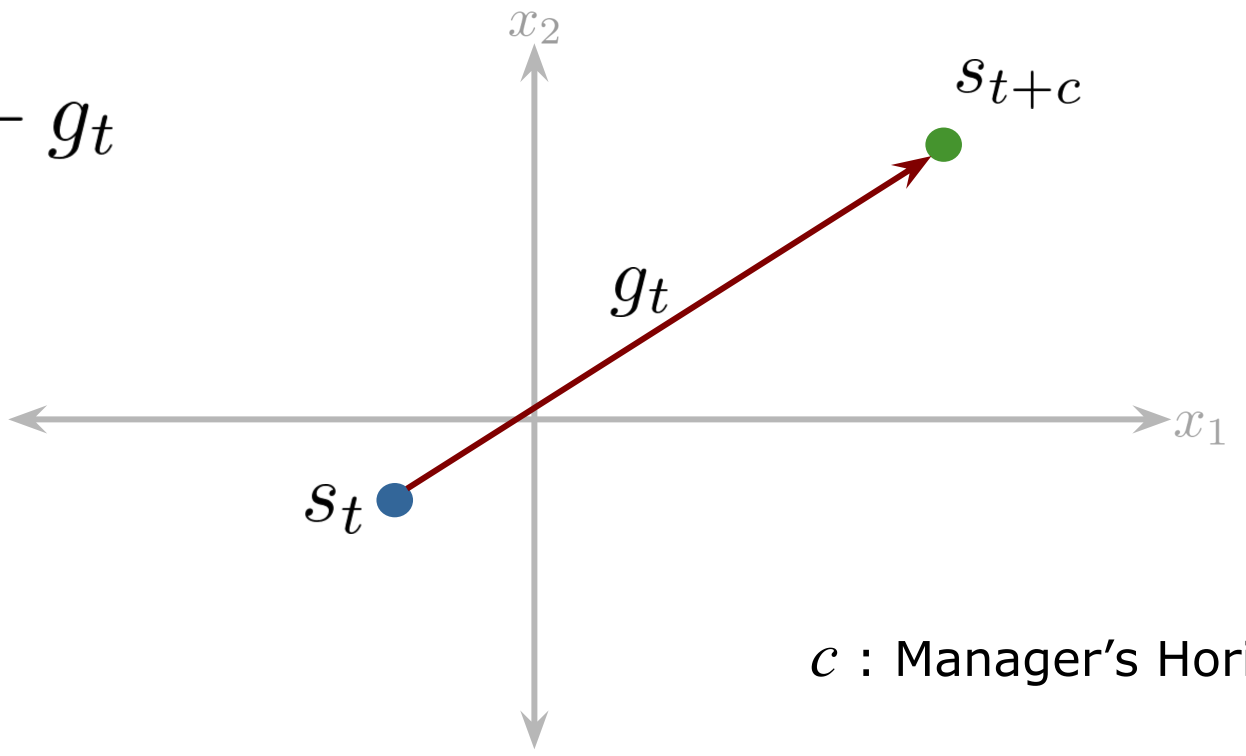


$$s = (q, \dot{q}, z) \qquad g = (\Delta q, \Delta \dot{q}, \Delta z) \qquad a = \tau_{act}$$

Raw Observation Space

Data-Efficient HRL (HIRO)

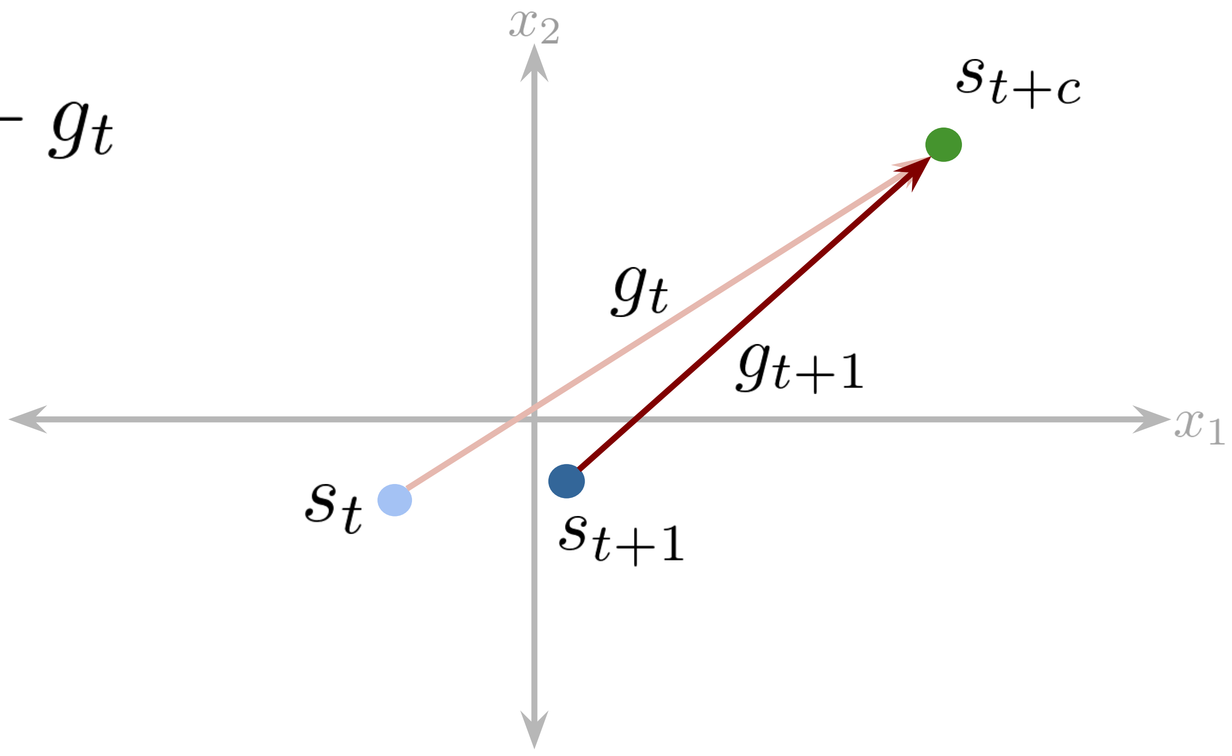
$$s_{t+c} \approx s_t + g_t$$



c : Manager's Horizon

Data-Efficient HRL (HIRO)

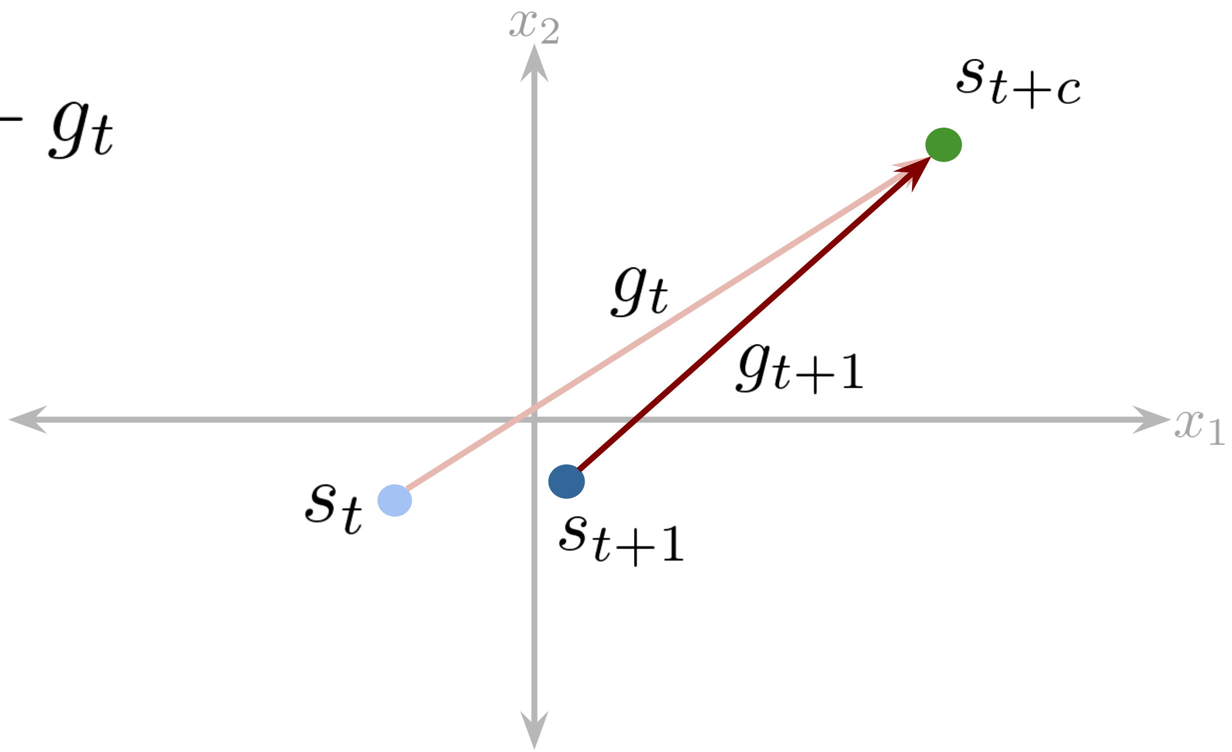
$$s_{t+c} \approx s_t + g_t$$



$$g_{t+1} = h(s_t, g_t, s_{t+1}) = s_t + g_t - s_{t+1}$$

Data-Efficient HRL (HIRO)

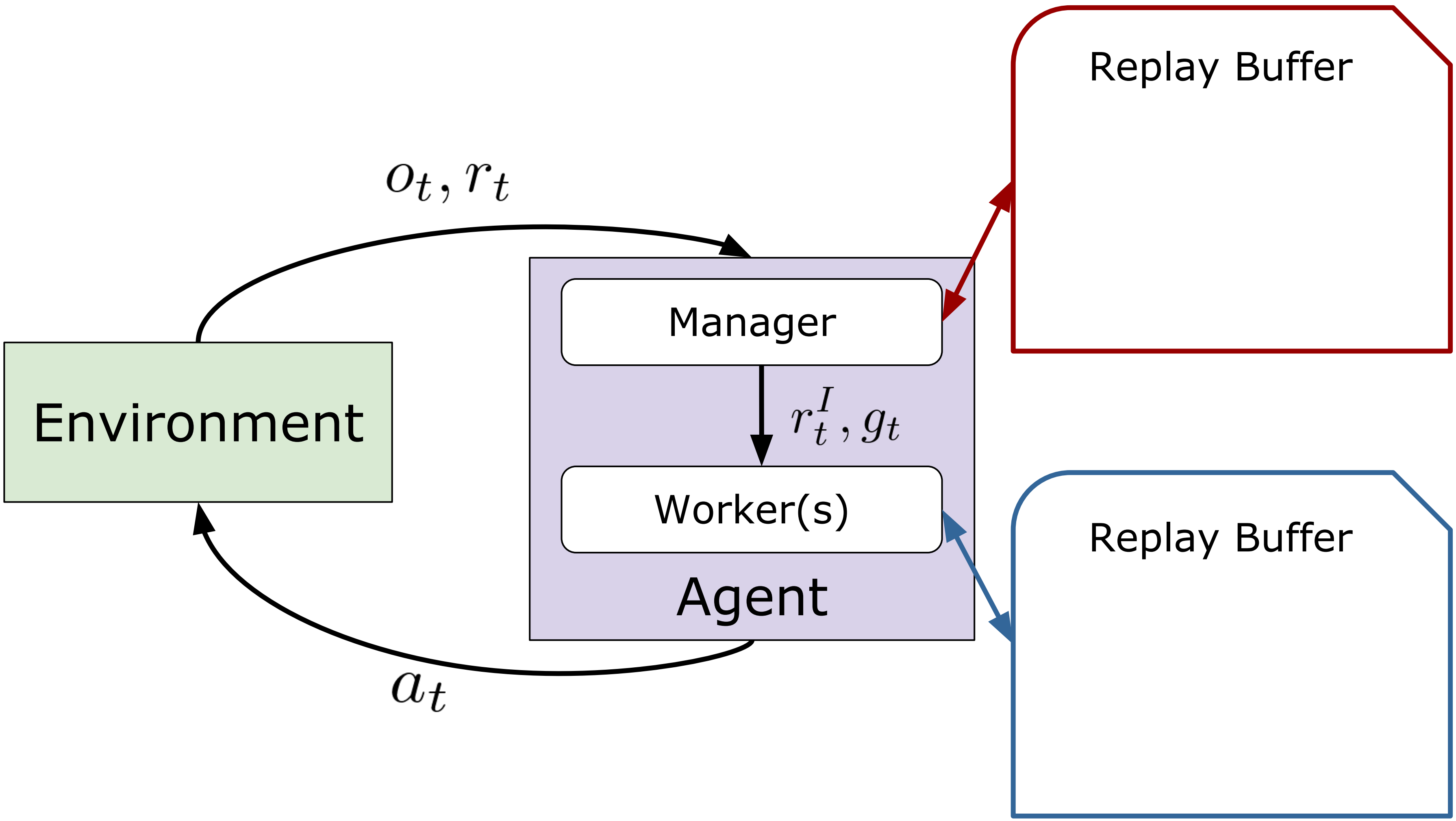
$$s_{t+c} \approx s_t + g_t$$



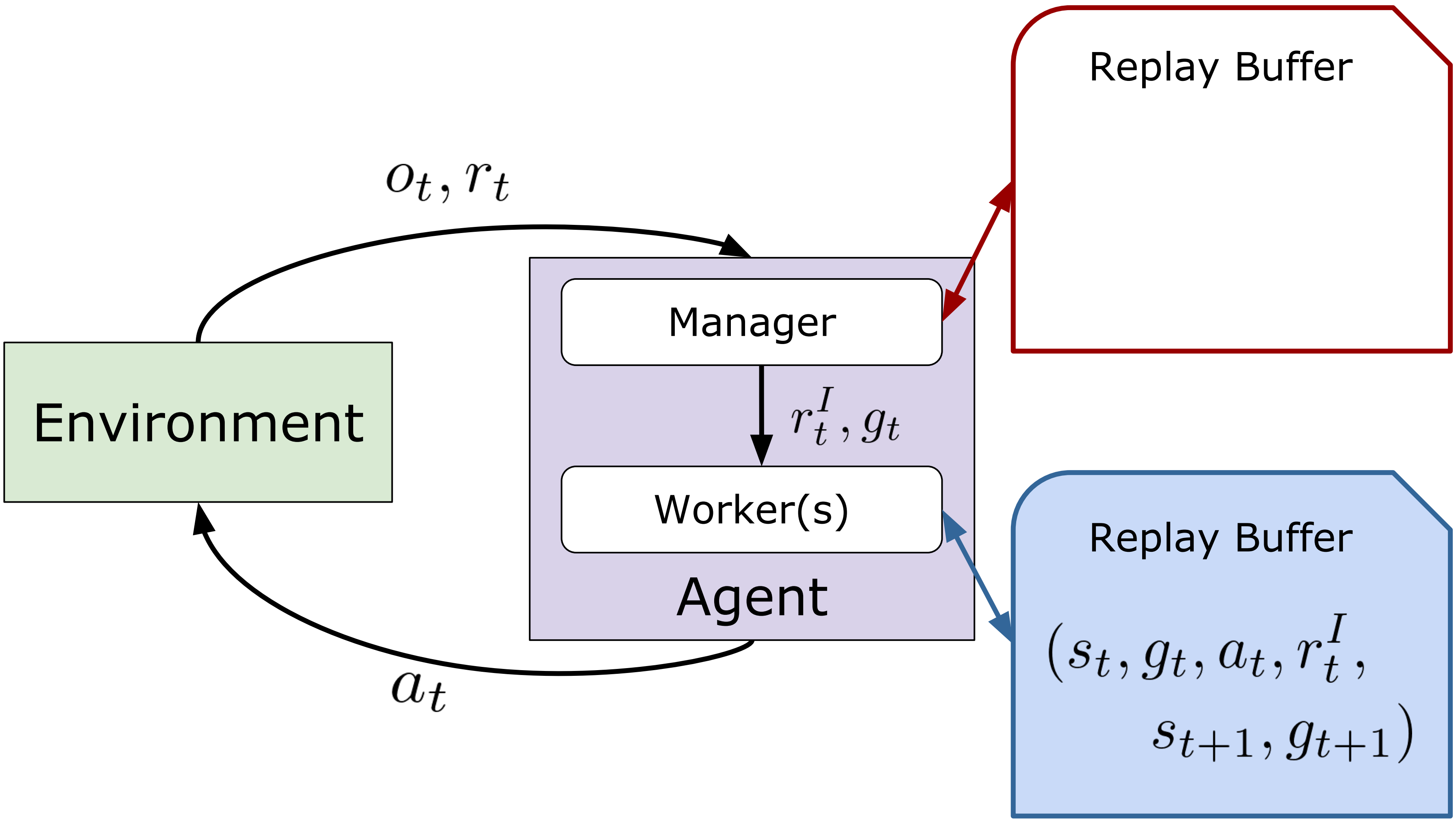
- Intrinsic reward

$$r_I(s_t, g_t, a_t, s_{t+1}) = -||s_t + g_t - s_{t+1}||_2$$

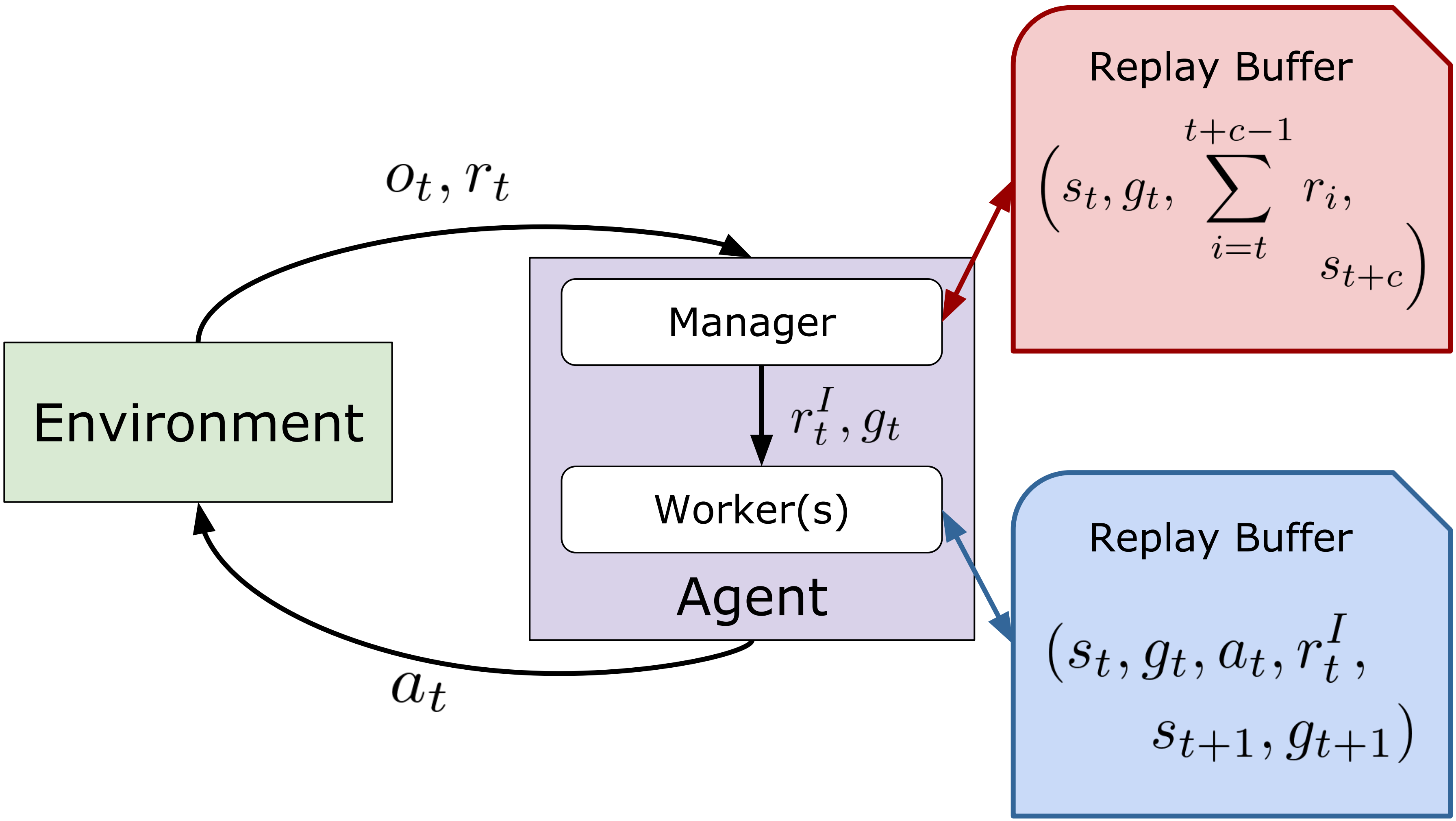
Data-Efficient HRL (HIRO)



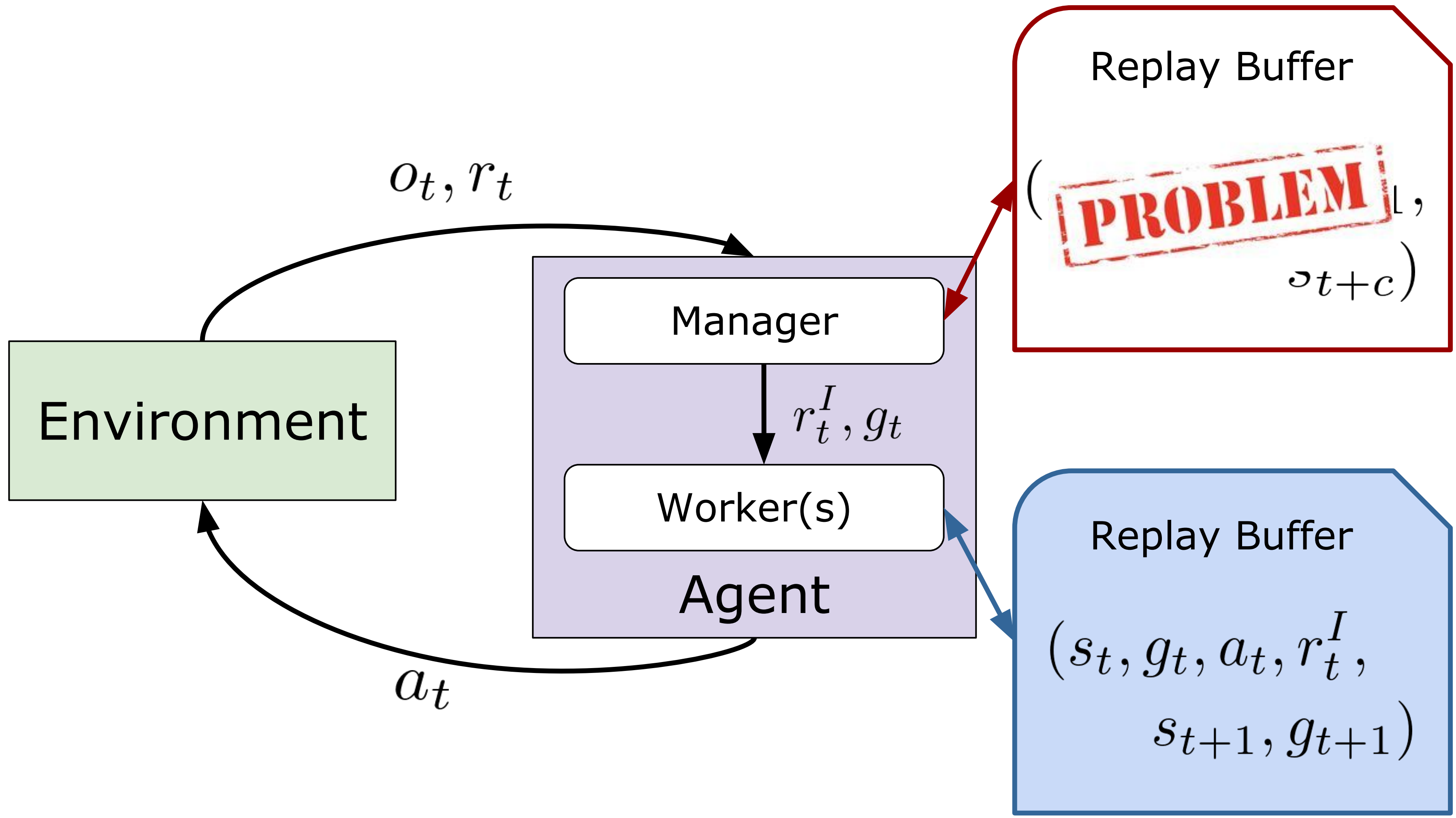
Data-Efficient HRL (HIRO)



Data-Efficient HRL (HIRO)



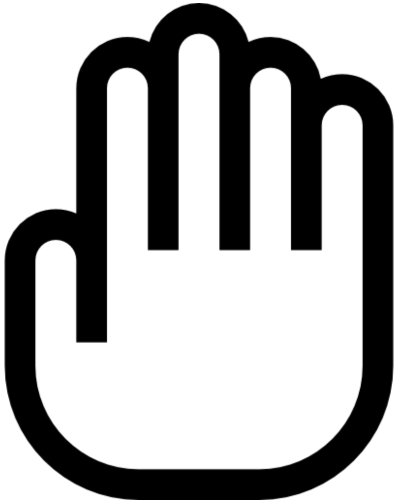
Data-Efficient HRL (HIRO)



Can we do better?

Off-Policy Learning 

Unstable Learning

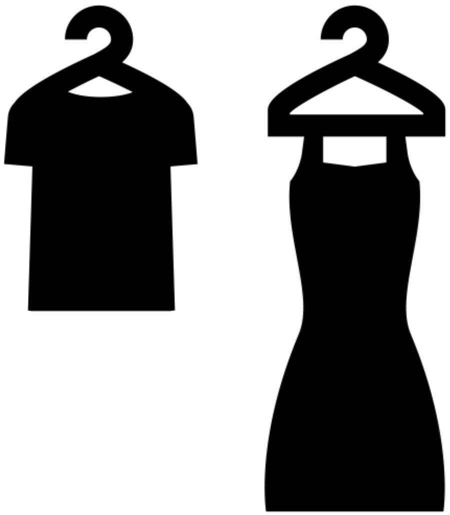


To-Be-Disclosed

Can we do better?

Off-Policy Learning 🙃

Unstable Learning



Manager's past experience might become useless

Can we do better?

Off-Policy Learning 🙃

t = 12 yrs

Can we do better?

Off-Policy Learning

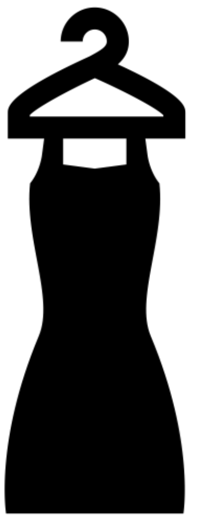
t = 22 yrs

Same goal induces
different behavior

Can we do better?

Off-Policy Learning

t = 22 yrs



Goal relabelling required!

Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

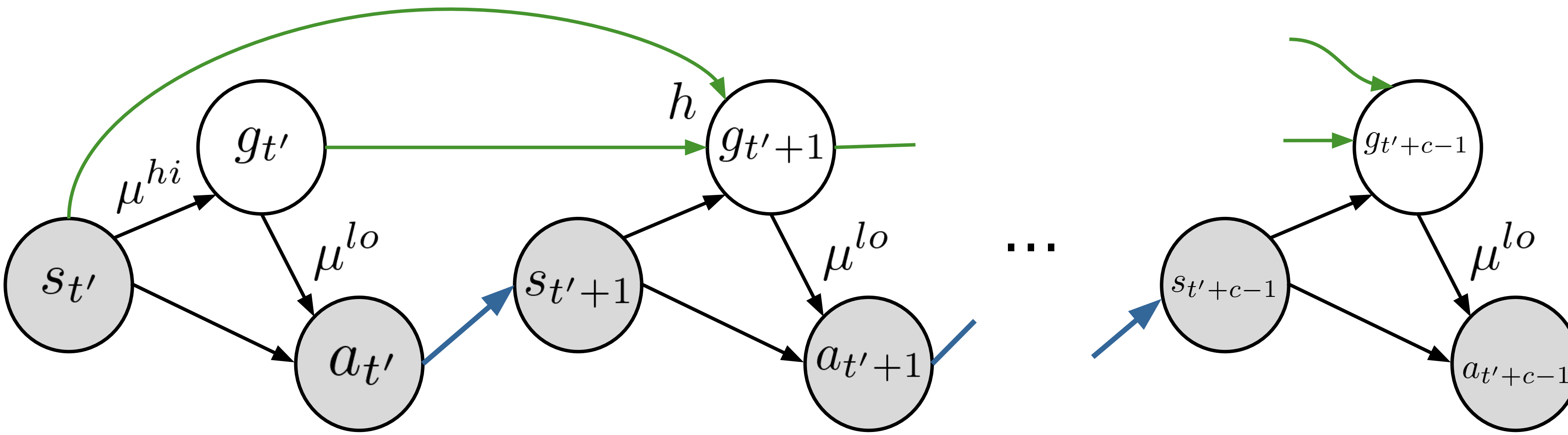
$$\left(s_{t'}, g_t, \sum_{i=t'}^{t'+c-1} r_i, s_{t'+c}\right)$$

$$\tilde{g}_{t'} = \operatorname{argmax} \mu^{lo}(a_{t':t'+c-1} | s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1})$$

$$\text{where} \quad \tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1})$$

Data-Efficient HRL (HIRO)

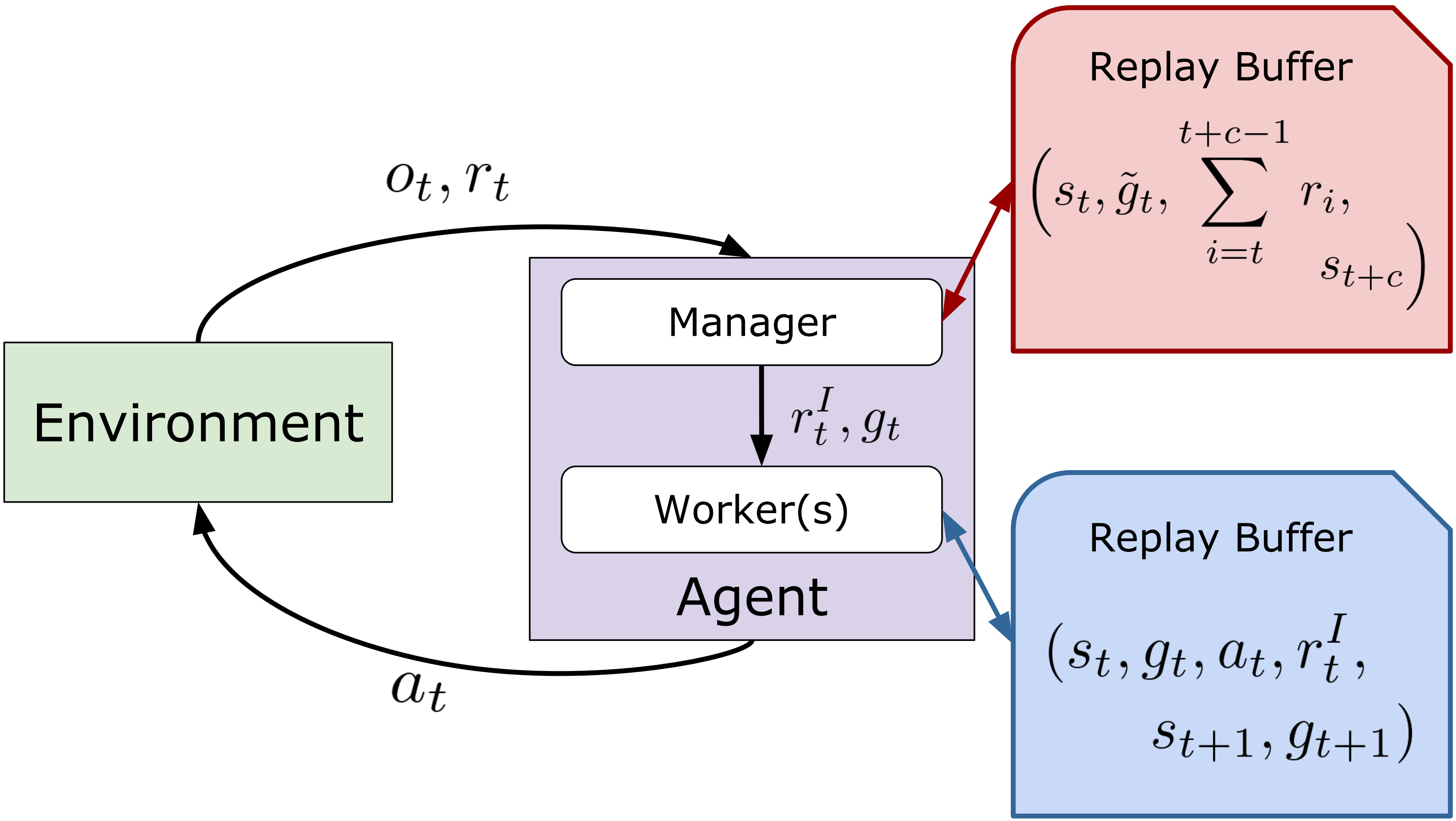
Off-Policy Correction for Manager



$$\tilde{g}_{t'} = \underset{\tilde{g}_{t'}}{\operatorname{argmax}}\ \mu^{lo}(a_{t':t'+c-1} | s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1})$$

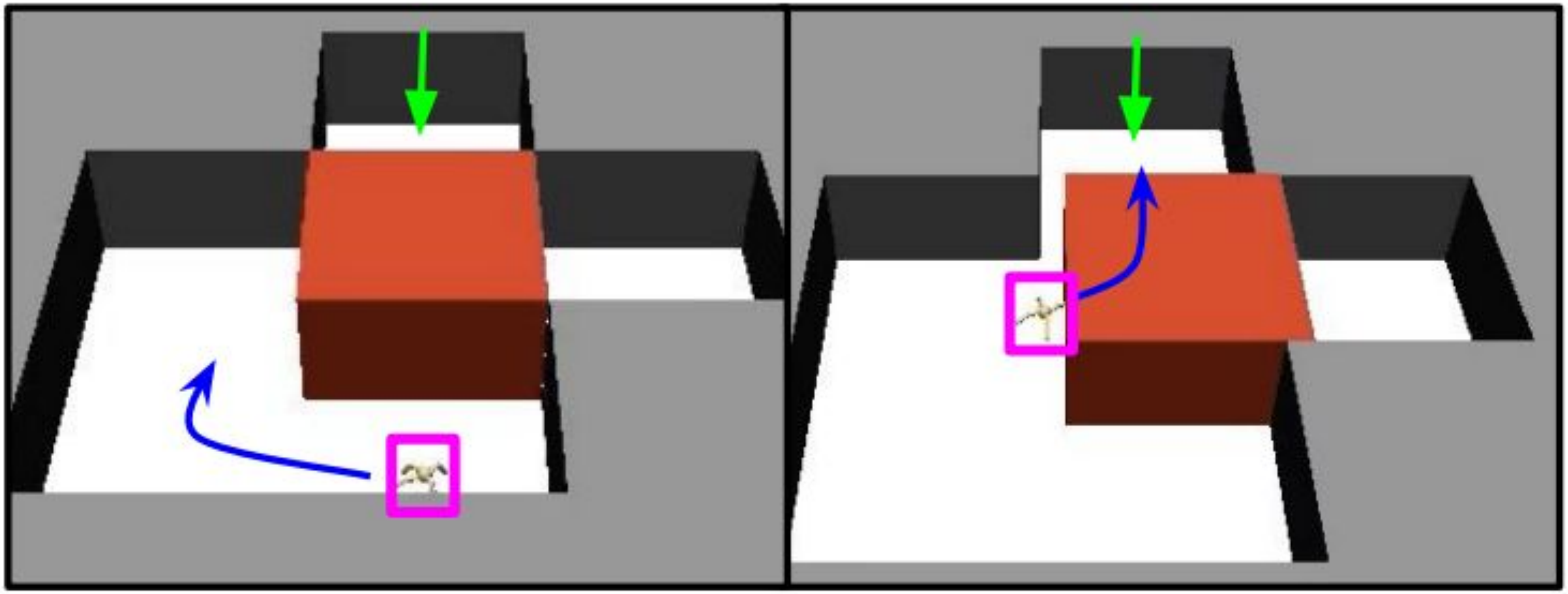
$$\text{where} \quad \tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1})$$

Data-Efficient HRL (HIRO)



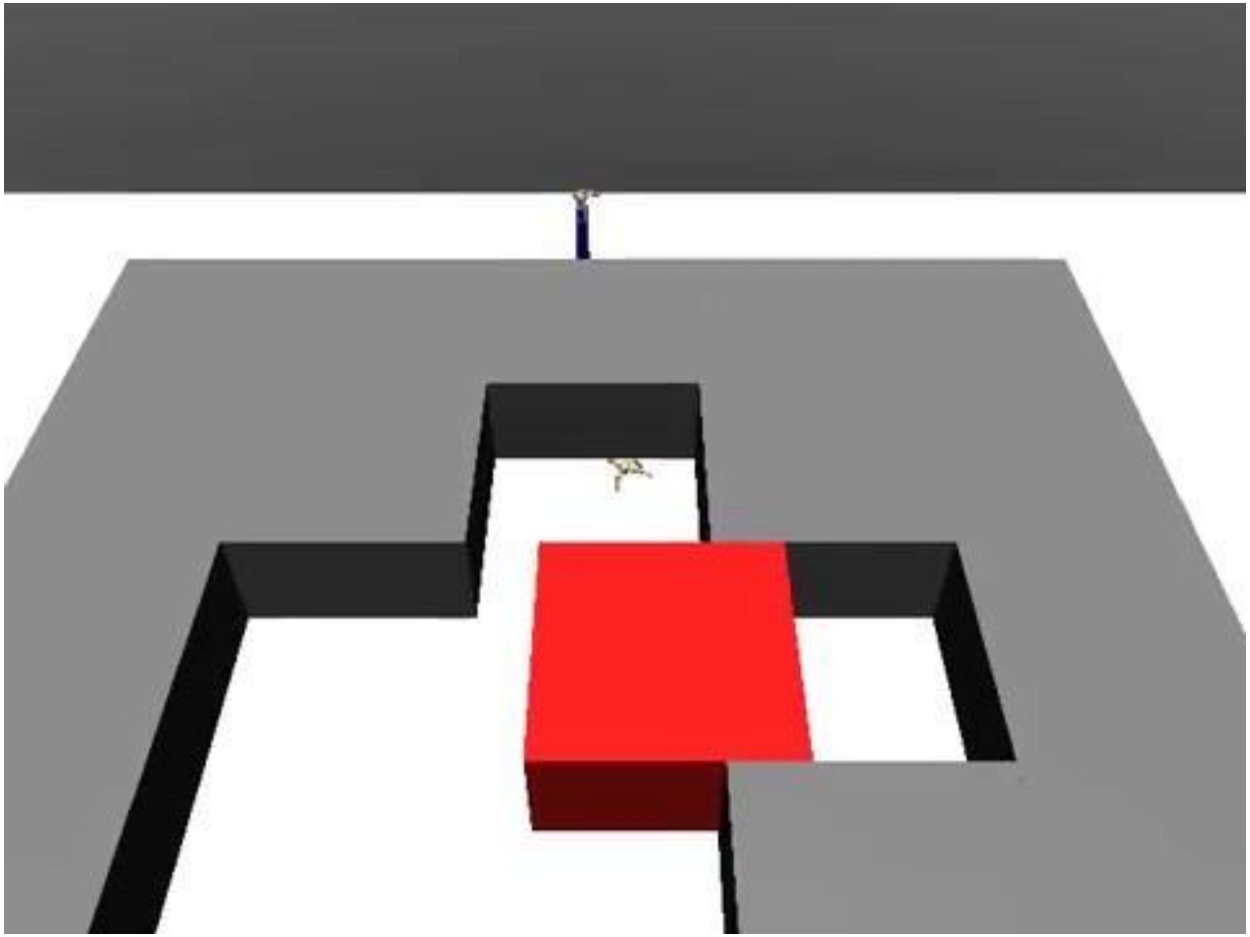
Data-Efficient HRL (HIRO)

Ant Push



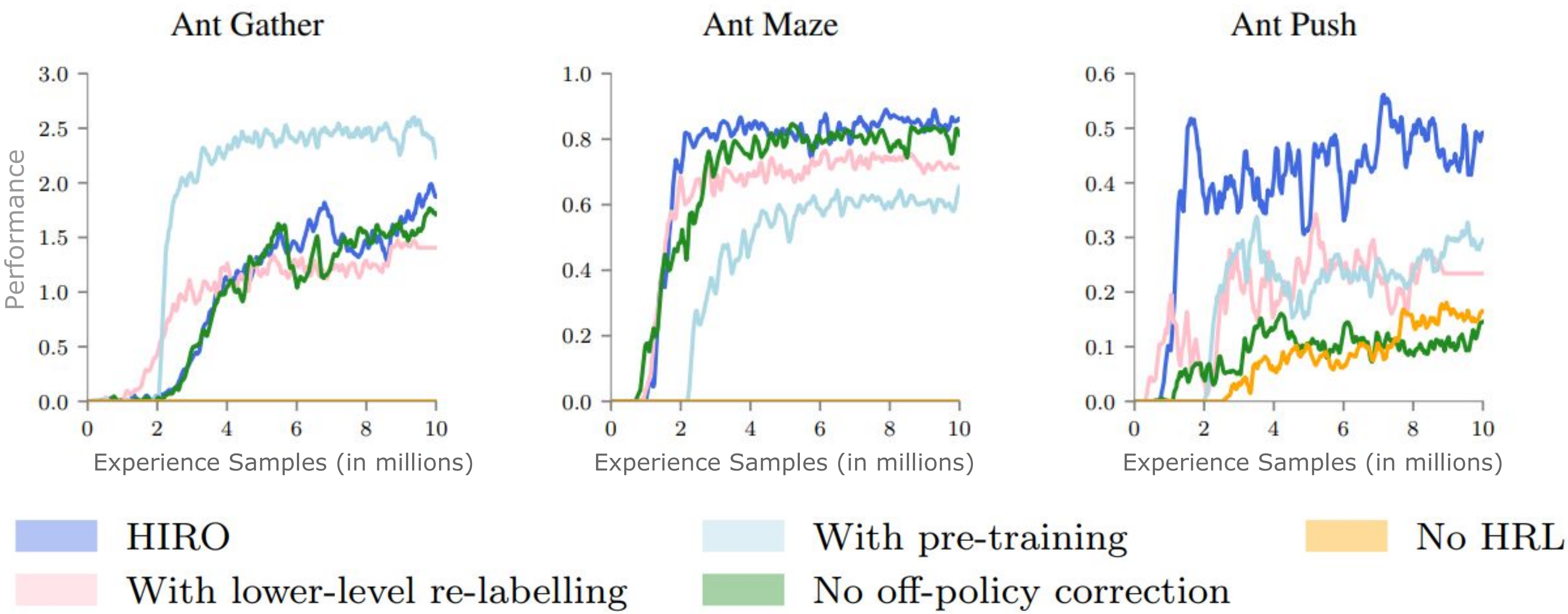
Data-Efficient HRL (HIRO)

Qualitative Analysis



Data-Efficient HRL (HIRO)

Ablative Analysis



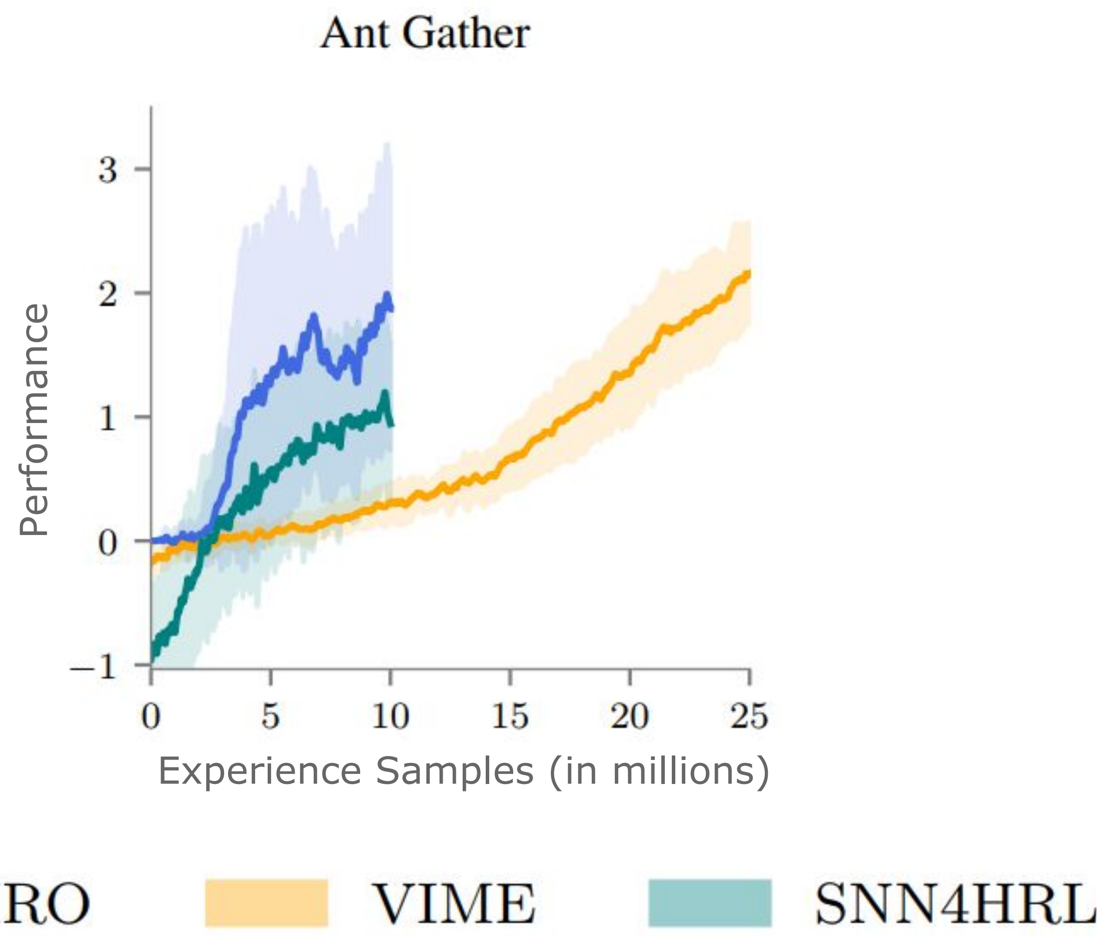
Data-Efficient HRL (HIRO)

Comparison

	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

Data-Efficient HRL (HIRO)

Comparison



Can we do better?

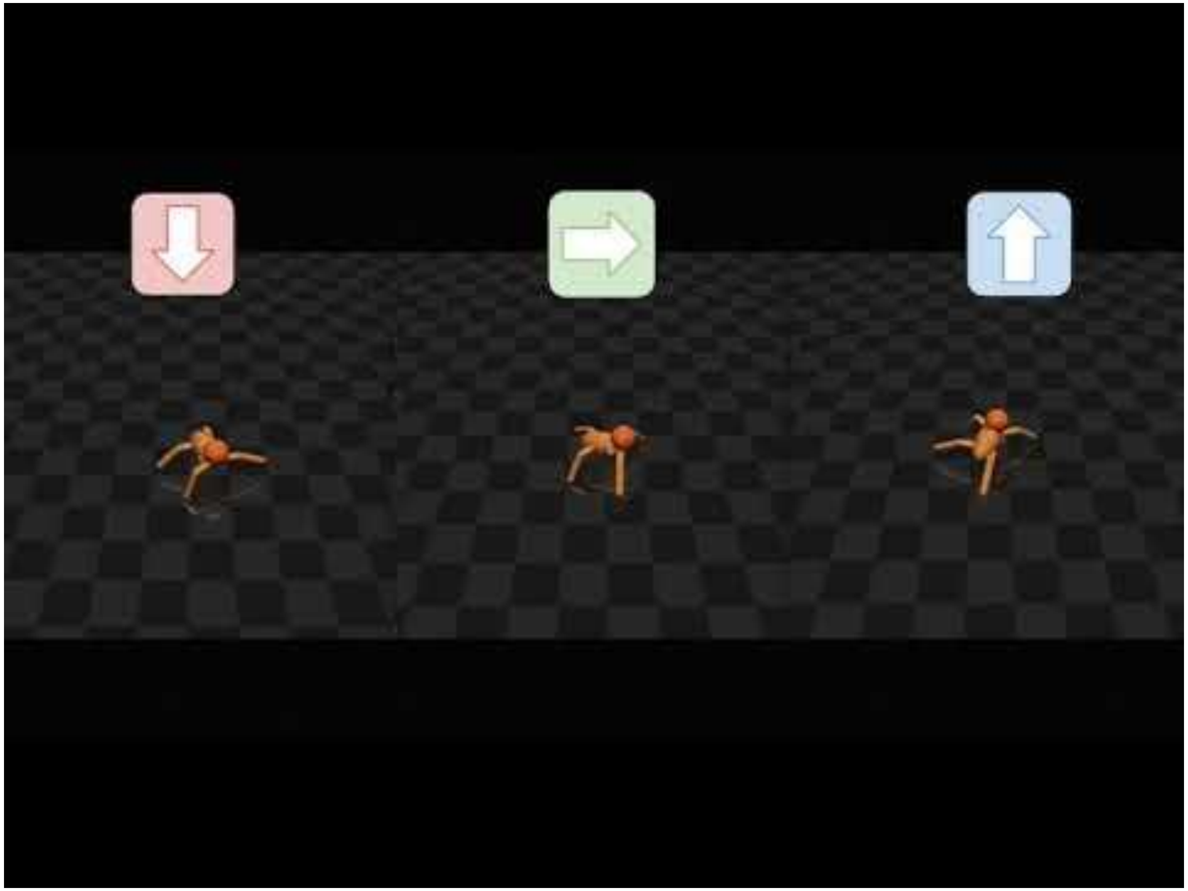
Can we do better?

What is missing?

Can we do better?

What is missing?

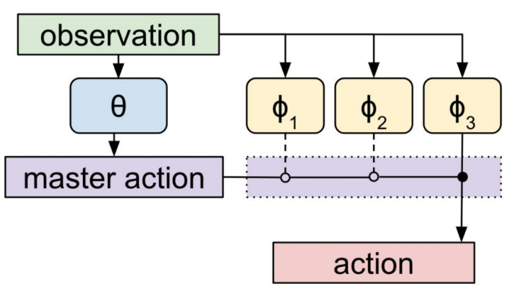
Structured exploration



Hierarchical RL

FeUdal Networks for Hierarchical Reinforcement Learning (ICML 2017)

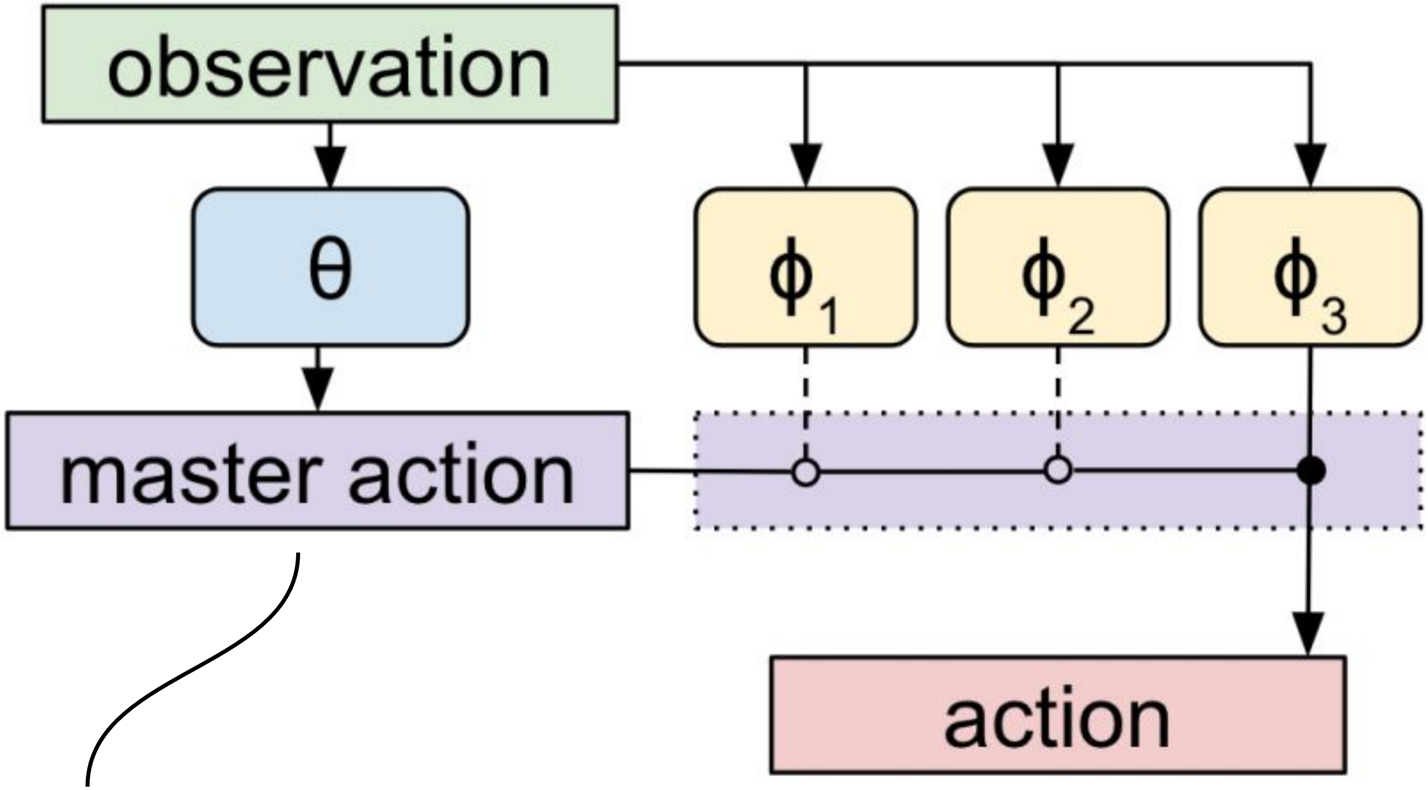
Data-Efficient Hierarchical Reinforcement Learning (NeurIPS 2018)



Meta-Learning Shared Hierarchies (ICLR 2018)

Meta-Learning Shared Hierarchies (MLSH)

Meta-Learning Shared Hierarchies (MLSH)



Taken after every
N steps

Meta-Learning **Shared Hierarchies (MLSH)**

Computer Vision practice:

- Train on ImageNet
- Fine tune on actual task

Slide Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)

Meta-Learning Shared Hierarchies (MLSH)

Computer Vision practice:

- Train on ImageNet
- Fine tune on actual task

How to generalize this to behavior learning?

Meta-Learning Shared Hierarchies (MLSH)

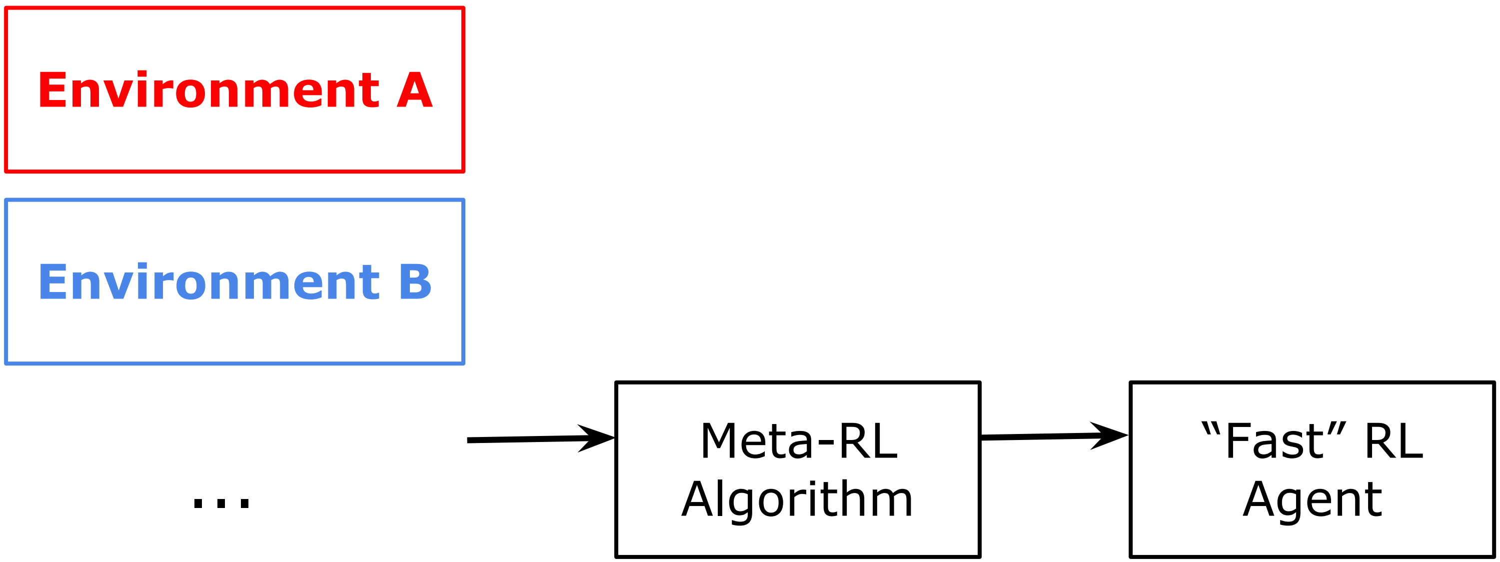


Image Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)

Meta-Learning Shared Hierarchies (MLSH)

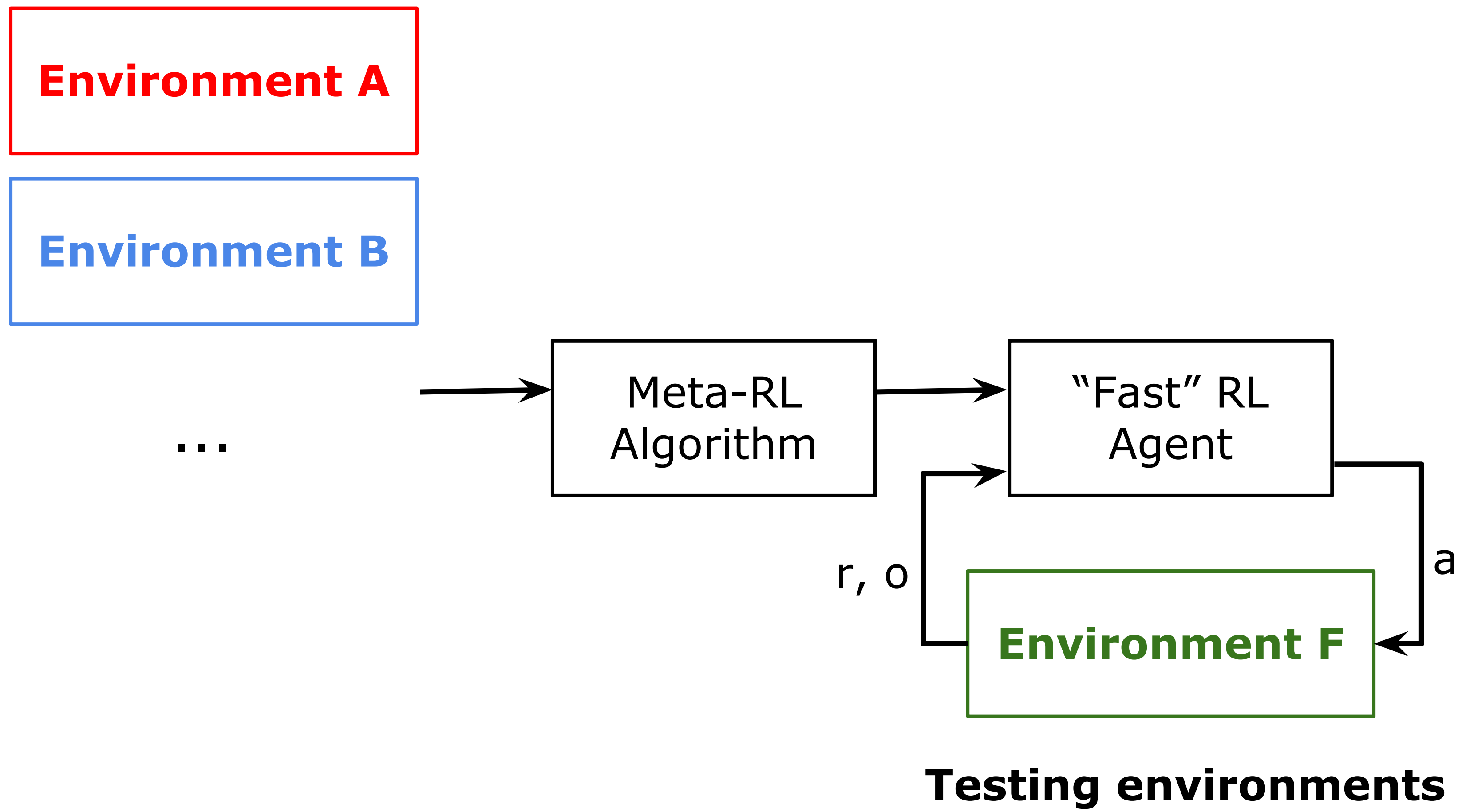


Image Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)

Meta-Learning Shared Hierarchies (MLSH)

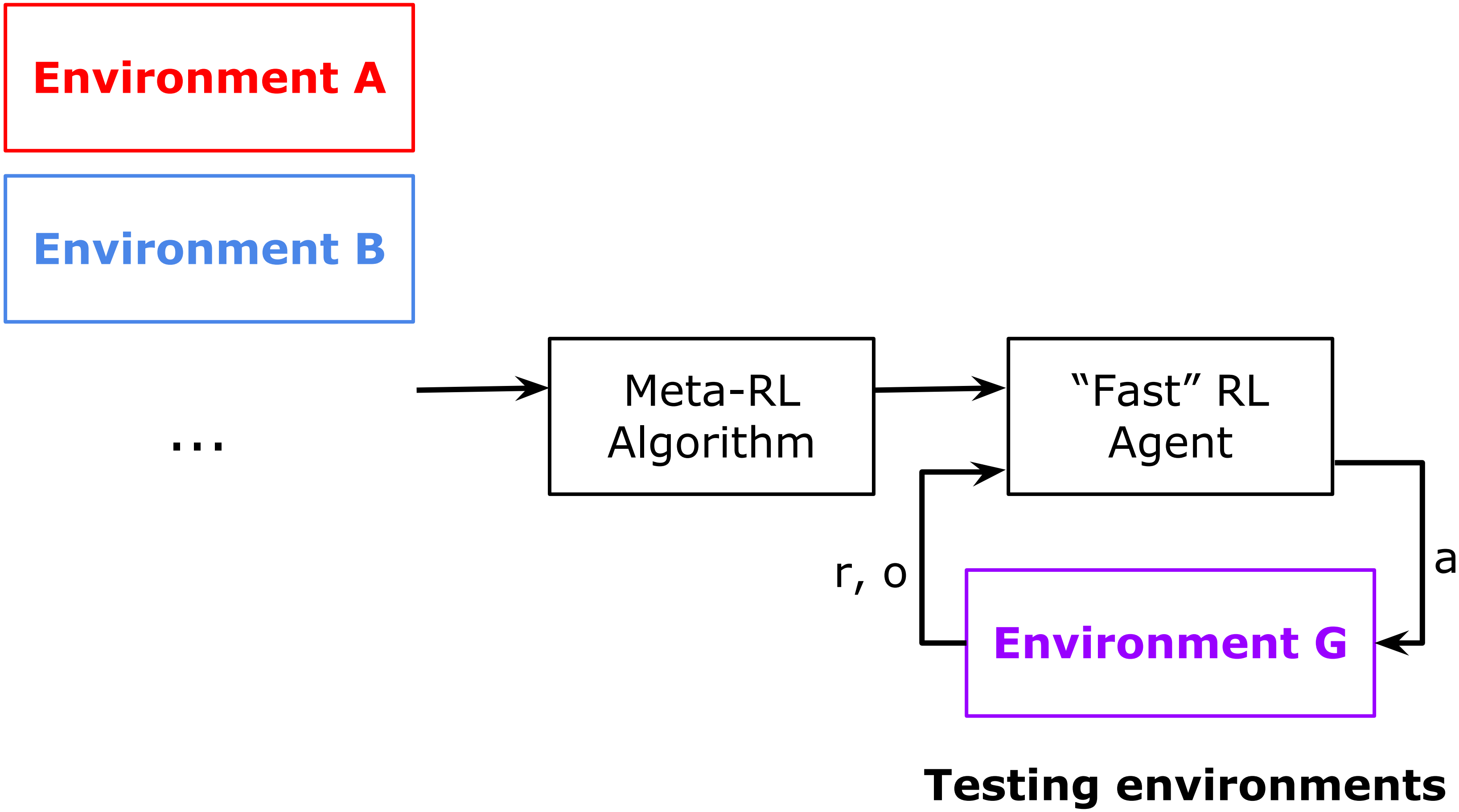
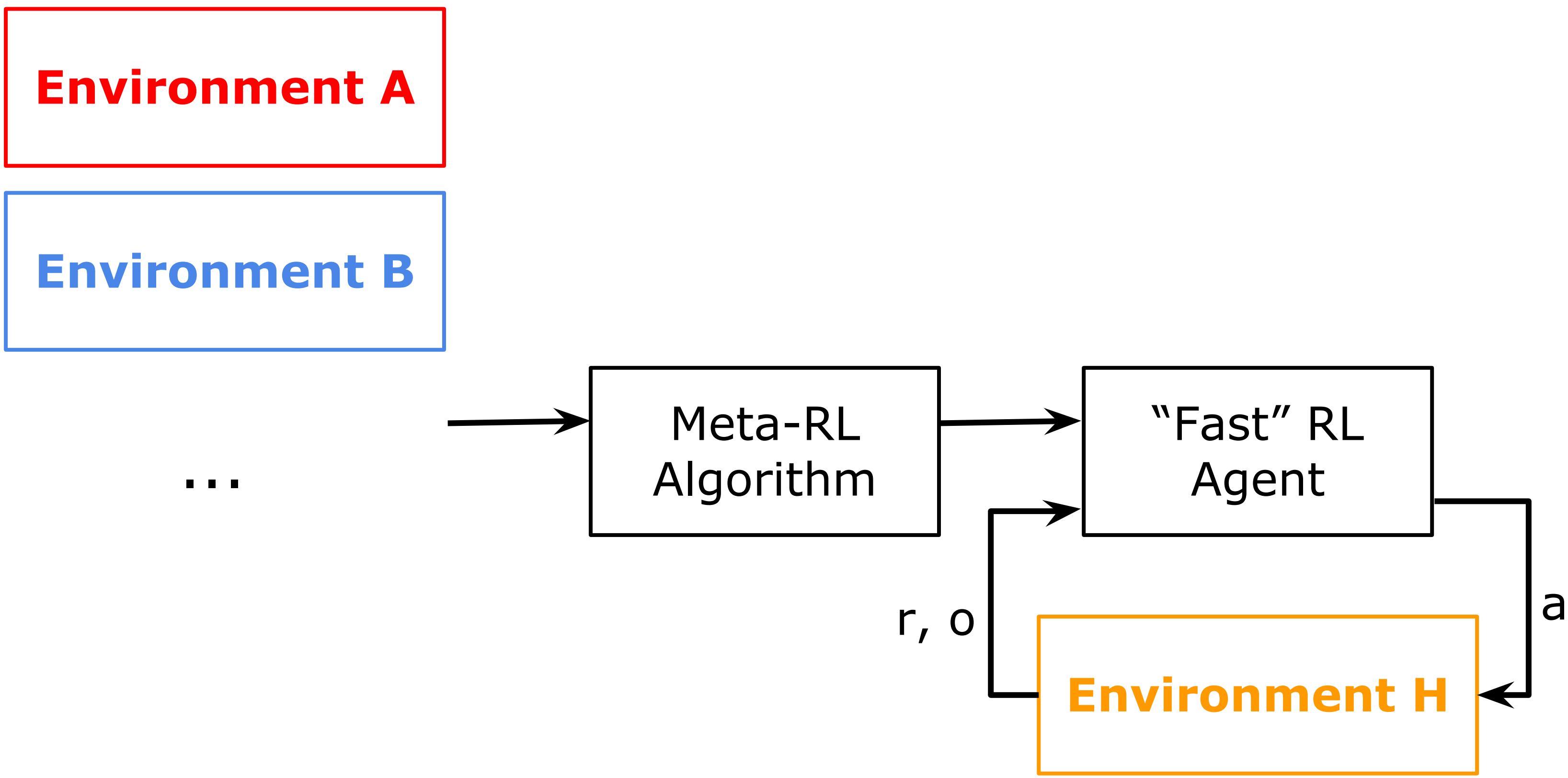


Image Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)

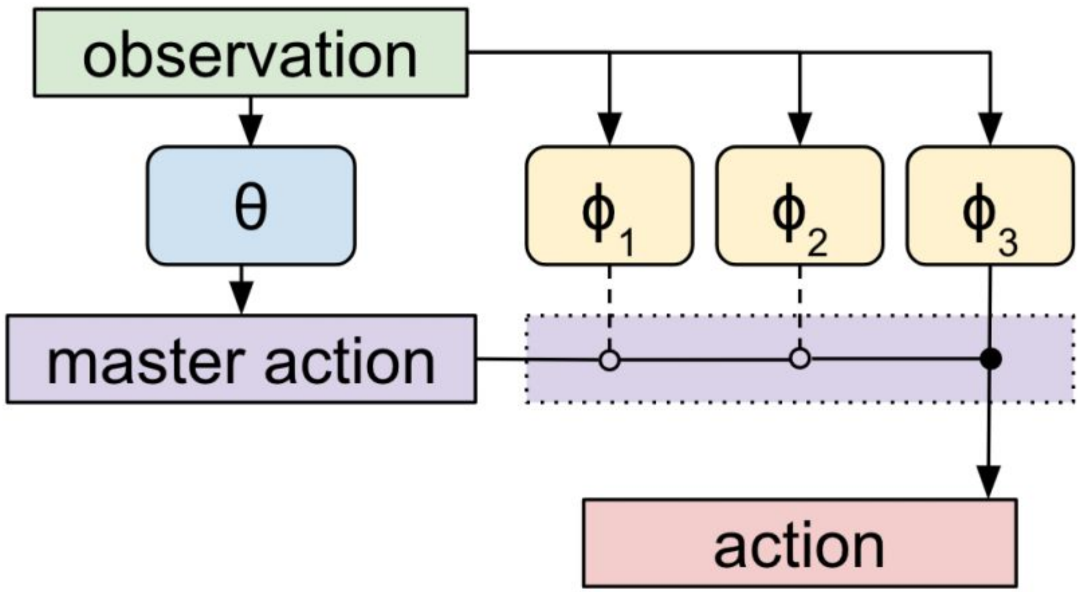
Meta-Learning Shared Hierarchies (MLSH)



Testing environments

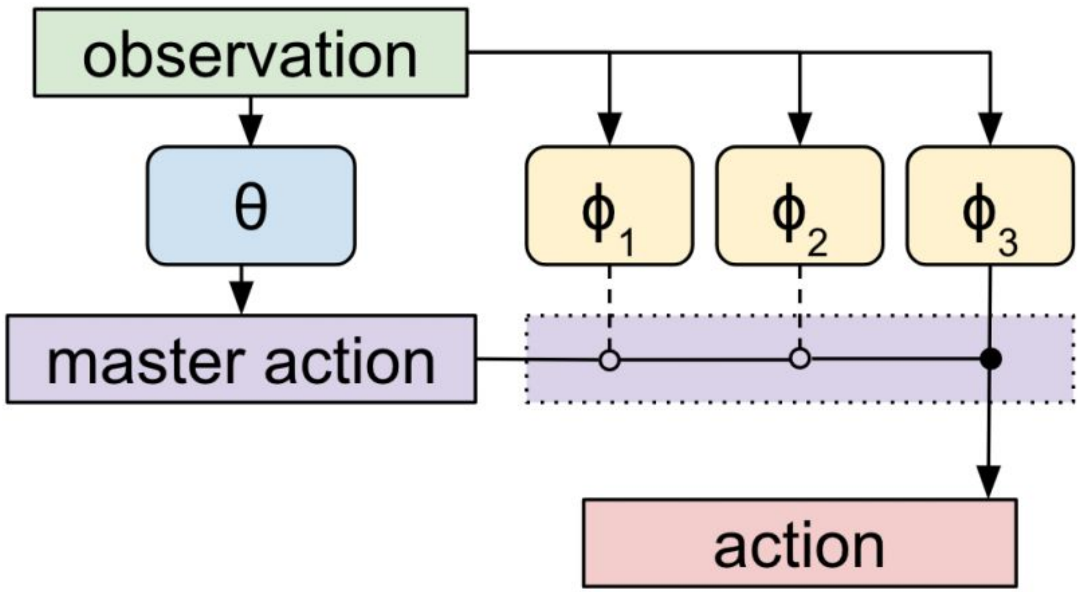
Image Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)

Meta-Learning Shared Hierarchies (MLSH)



GOAL: Find sub-policies that enable fast learning of master policy θ

Meta-Learning Shared Hierarchies (MLSH)

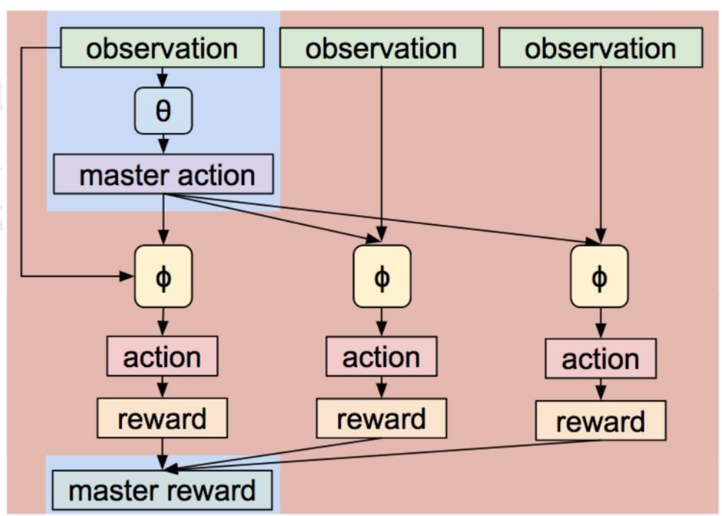


GOAL: Find sub-policies that enable fast learning of master policy θ

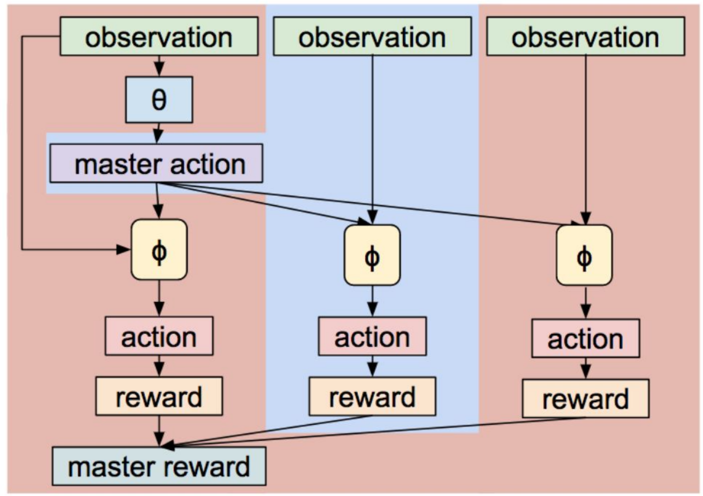
$$\text{maximize}_\phi \, E_{M \sim P_M, t=0 \ldots T-1}[R]$$

Meta-Learning Shared Hierarchies (MLSH)

Initialize ϕ
repeat
  Initialize θ
  Sample task $M \sim P_M$
  for $w = 0, 1, ...W$ (warmup period) **do**
    Collect D timesteps of experience using $\pi_{\phi,\theta}$
    Update θ to maximize expected return from $1/N$ timescale viewpoint
  end for
  for $u = 0, 1, ...U$
    Collect D timest
    Update θ to max timescale viewpoint
    Update ϕ to max imescale viewpoint
  end for
until convergence



Meta-Learning Shared Hierarchies (MLSH)



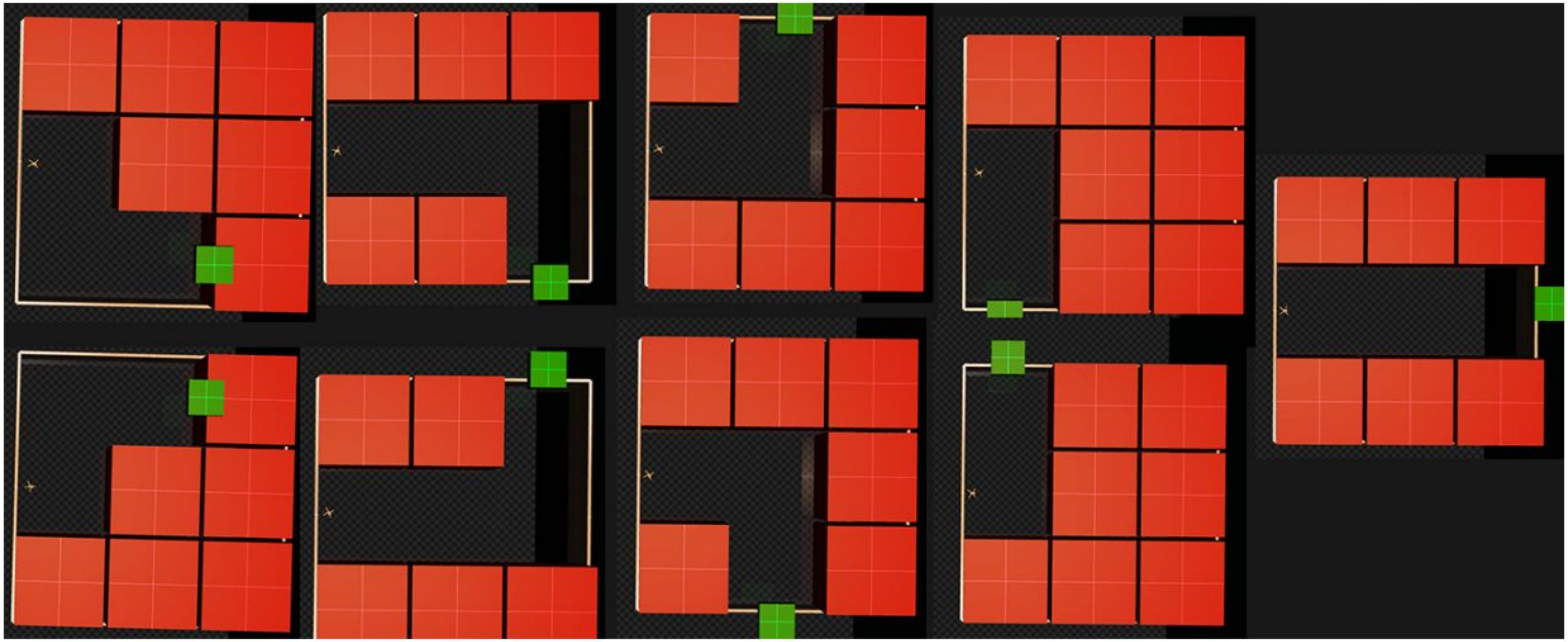
Initialize ϕ
repeat
Initialize θ
Sample task $M \sim$
for $w = 0, 1, ...W$
Collect D timest
Update θ to maxi timescale viewpoint
end for
for $u = 0, 1, ...U$ (joint update period) **do**
Collect D timesteps of experience using $\pi_{\phi,\theta}$
Update θ to maximize expected return from $1/N$ timescale viewpoint
Update ϕ to maximize expected return from full timescale viewpoint
end for
until convergence

Meta-Learning Shared Hierarchies (MLSH)

Initialize ϕ
repeat
    Initialize θ
    Sample task $M \sim P_M$
    for $w = 0, 1, ...W$ (warmup period) **do**
        Collect D timesteps of experience using $\pi_{\phi,\theta}$
        Update θ to maximize expected return from $1/N$ timescale viewpoint
    end for
    for $u = 0, 1, ...U$ (joint update period) **do**
        Collect D timesteps of experience using $\pi_{\phi,\theta}$
        Update θ to maximize expected return from $1/N$ timescale viewpoint
        Update ϕ to maximize expected return from full timescale viewpoint
    end for
until convergence

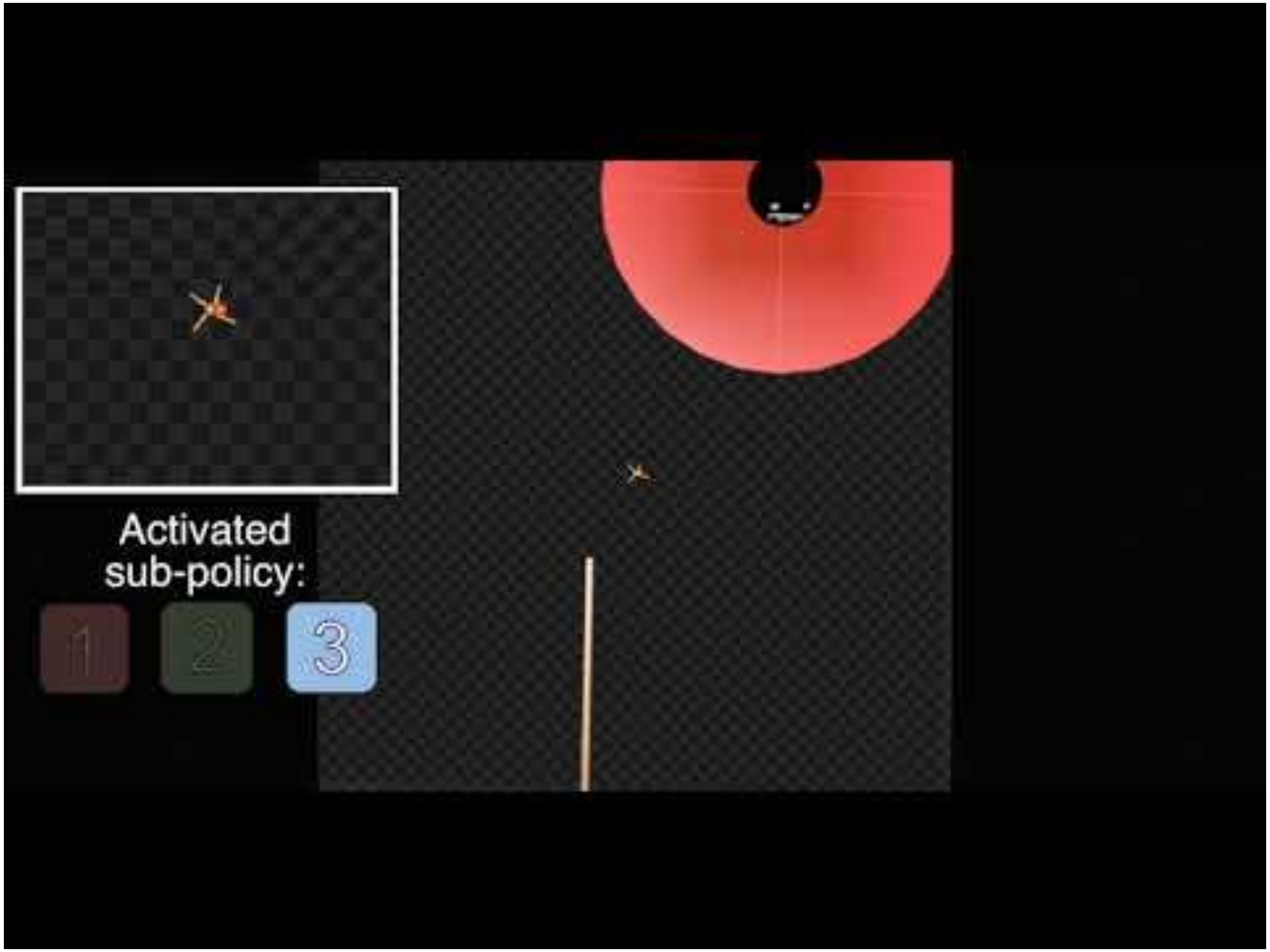
Meta-Learning Shared Hierarchies (MLSH)

Ant Two-walks



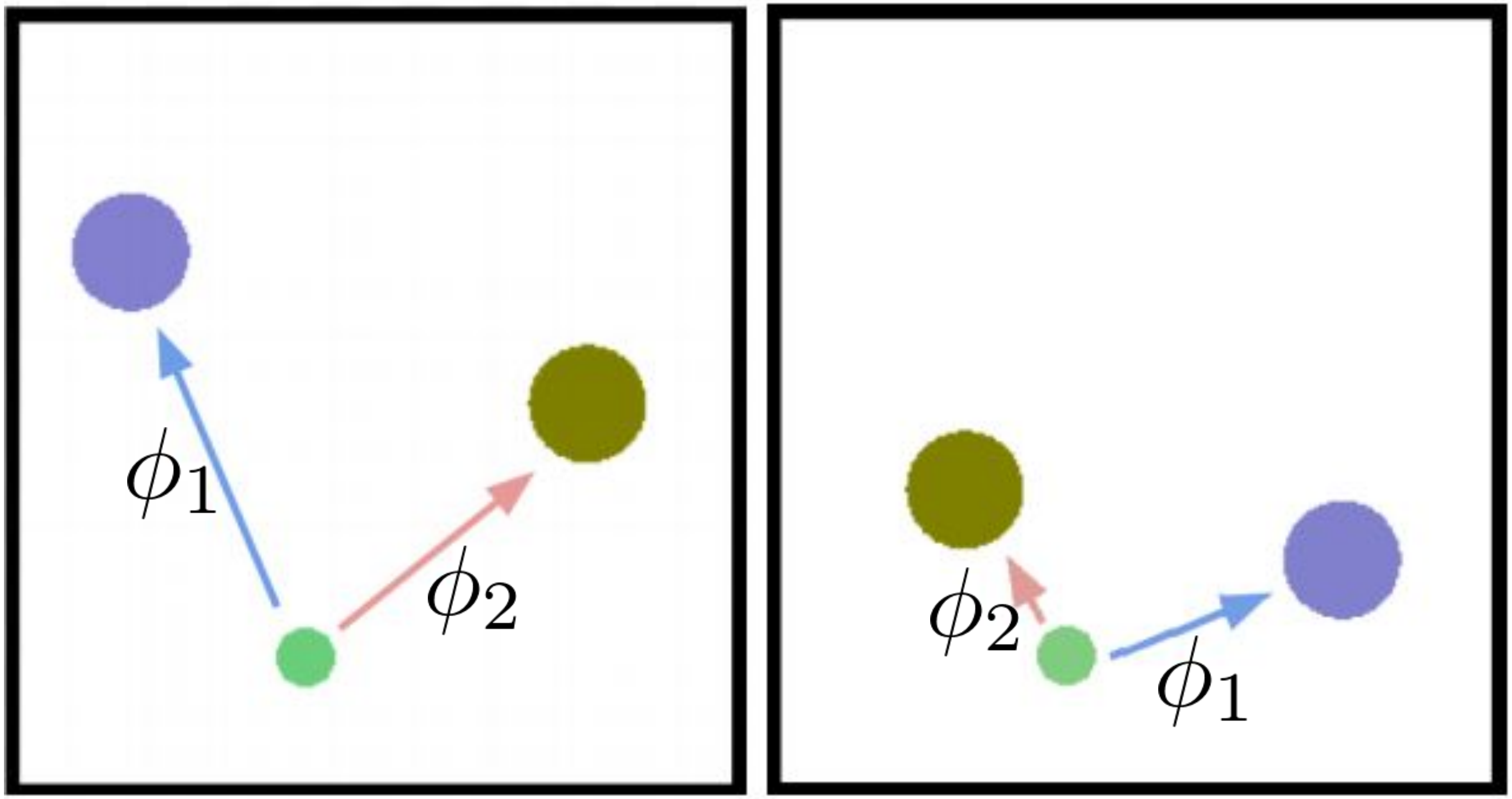
Meta-Learning Shared Hierarchies (MLSH)

Ant Obstacle Course



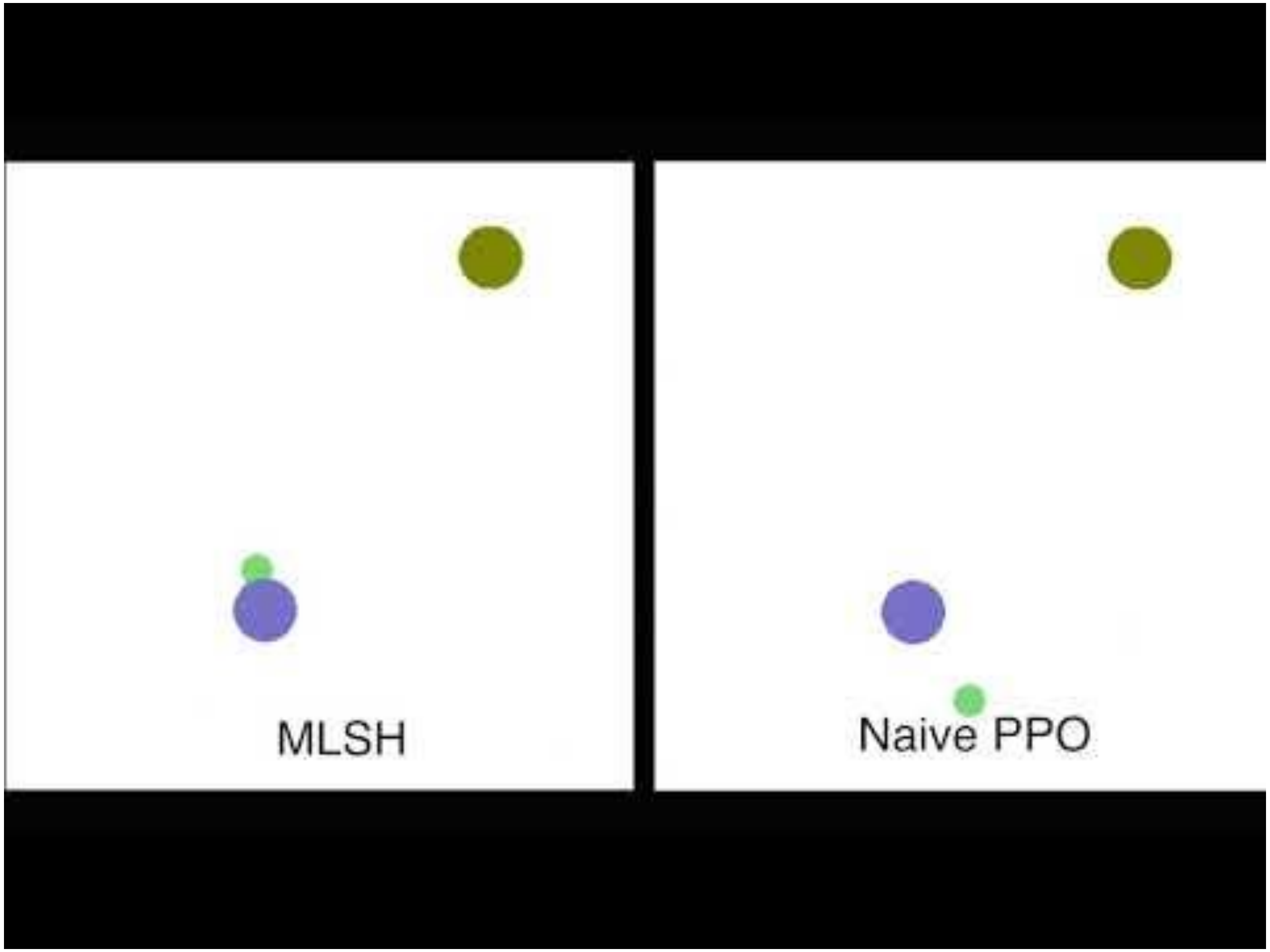
Meta-Learning Shared Hierarchies (MLSH)

Movement Bandits



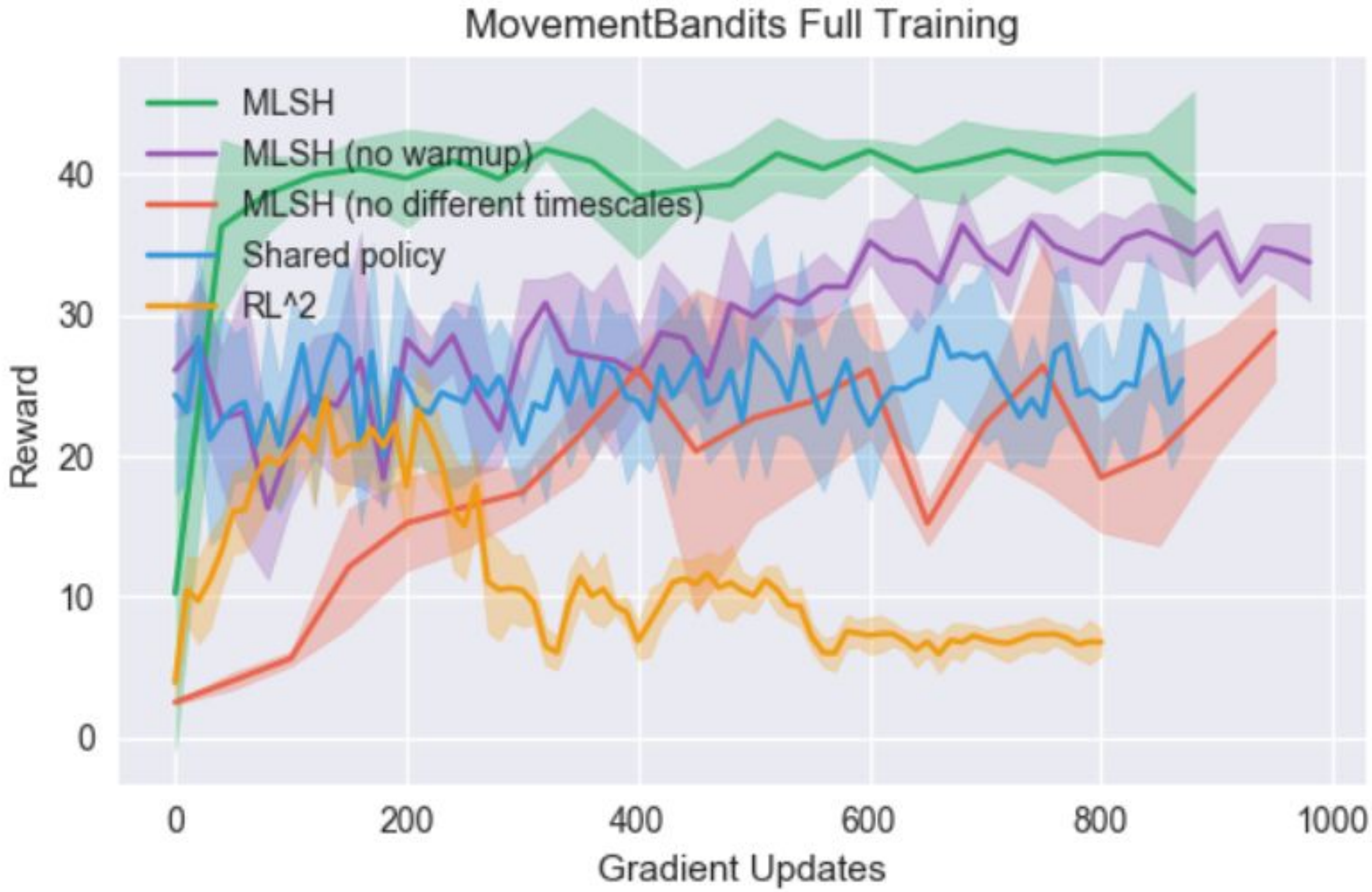
Meta-Learning Shared Hierarchies (MLSH)

Comparison



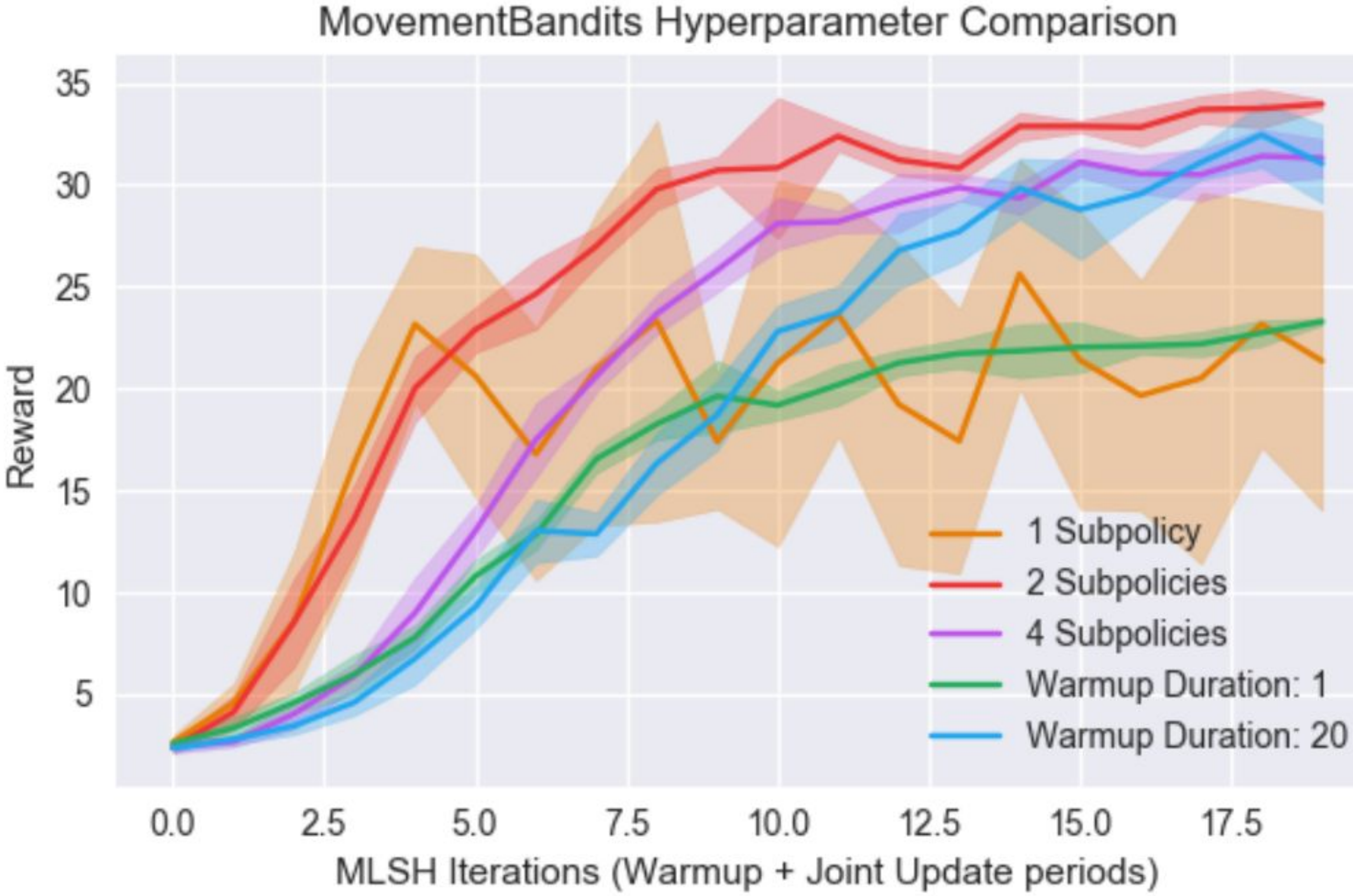
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis



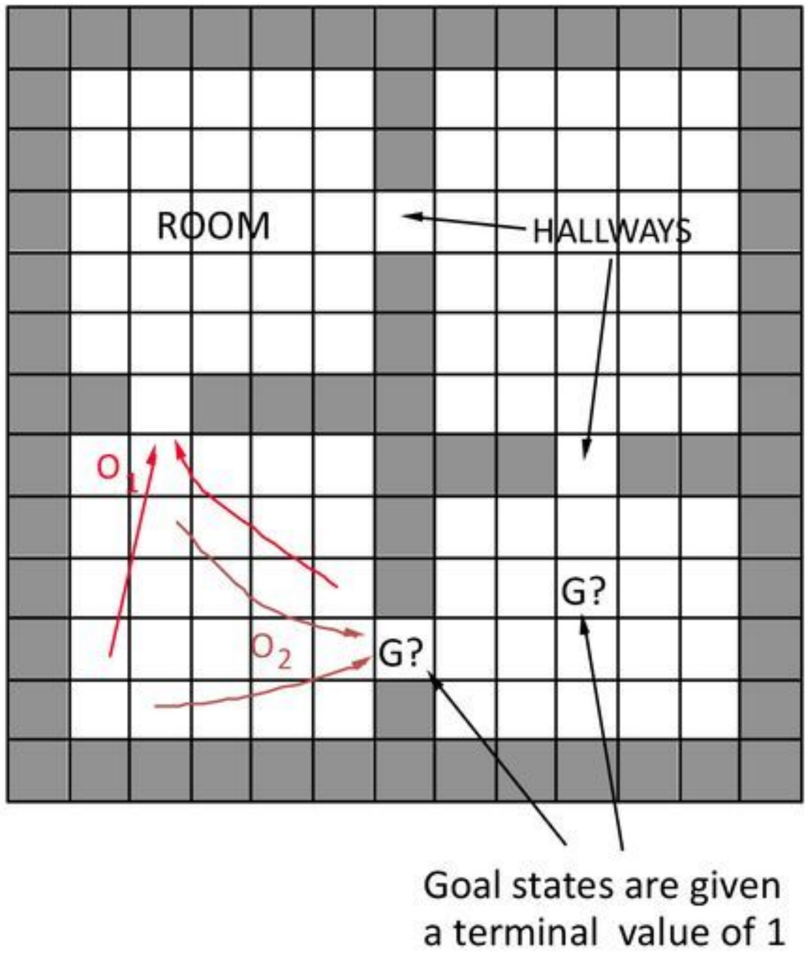
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis



Meta-Learning Shared Hierarchies (MLSH)

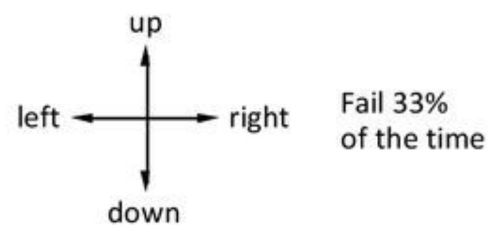
Four Rooms



4 rooms

4 hallways

4 unreliable
primitive actions



8 multi-step options

(to each room's 2 hallways)

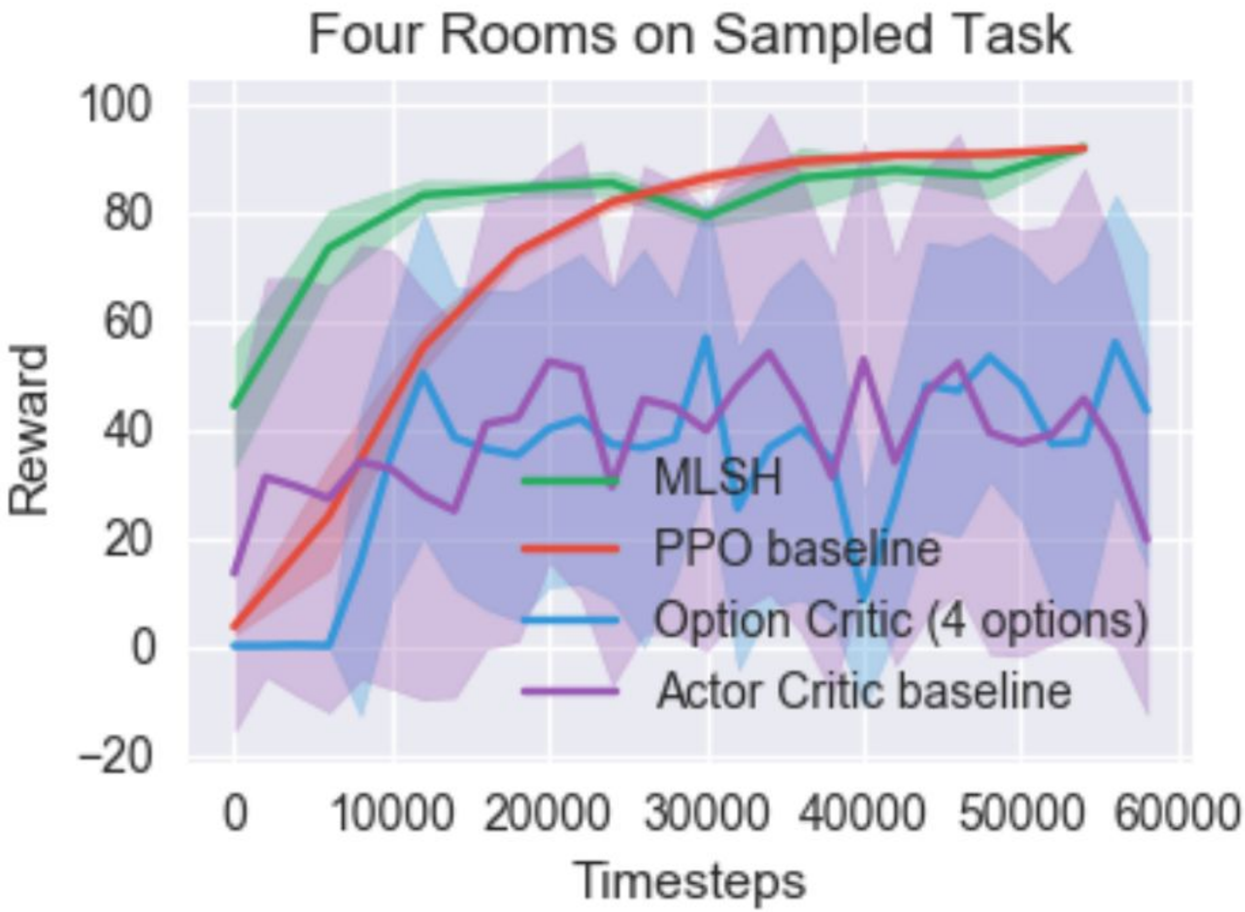
Given goal location,
quickly plan shortest route

All rewards zero

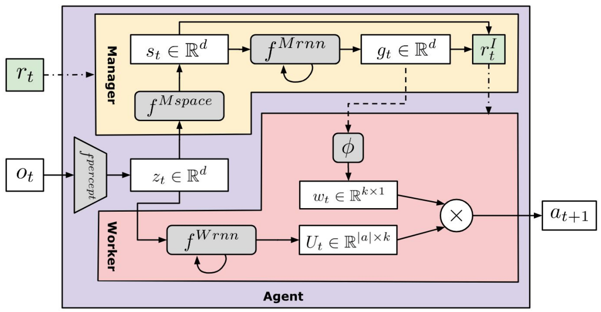
$\gamma = .9$

Meta-Learning Shared Hierarchies (MLSH)

Comparison

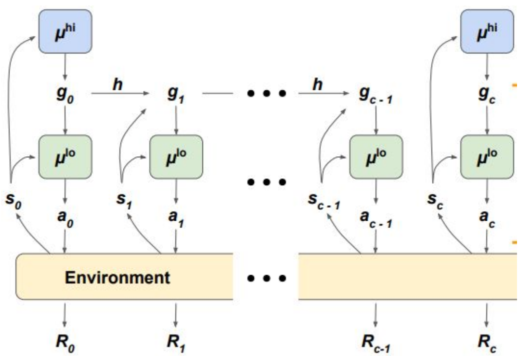


Summary



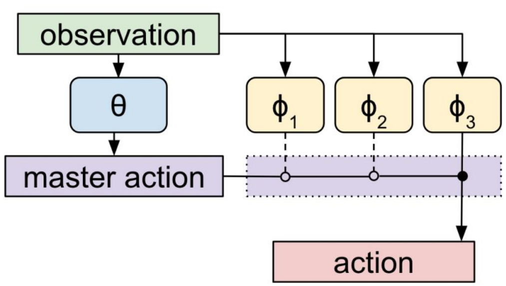
FUN

- Directional goals
- Dilated RNN
- Transition Policy Gradient



HIRO

- Absolute goals in observation space
- Data-efficient
- Off-policy label correction



MLSH

- Generalized RL algorithm
- Inspired from “Options” framework

Future Work

- How to decide temporal resolution (i.e. c, N)?

- Do we need discrete sub-policies?

- Future prospects of HRL? More hierarchies?

Thank you for your attention!

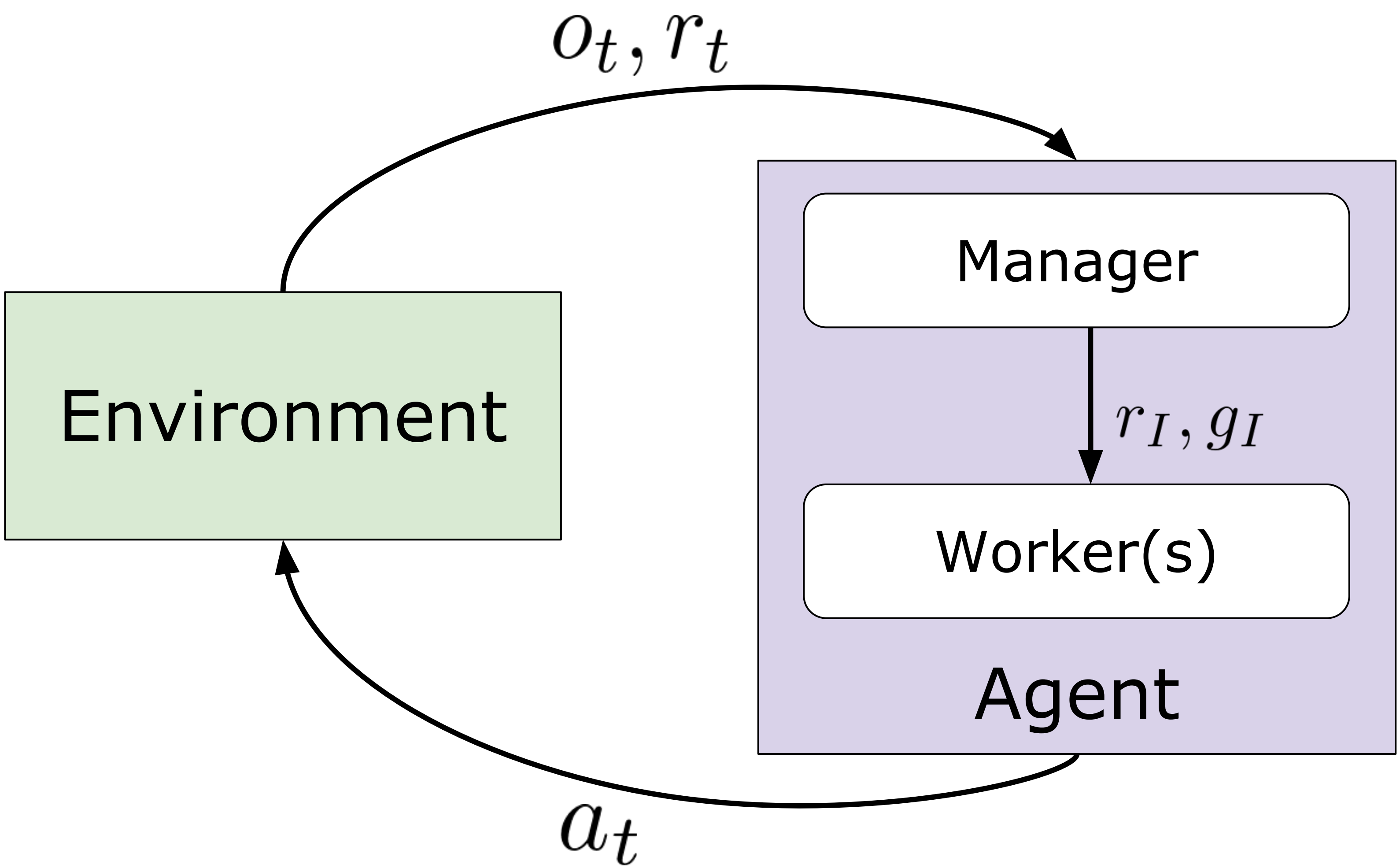
Any Questions?

References

- Vezhnevets, A.S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., & Kavukcuoglu, K. (2017). **FeUdal Networks for Hierarchical Reinforcement Learning**. *ICML*.
- Nachum, O., Gu, S., Lee, H., & Levine, S. (2018). **Data-Efficient Hierarchical Reinforcement Learning**. *NeurIPS*.
- Frans, K., Ho, J., Chen, X., Abbeel, P., & Schulman, J. (2018). **Meta Learning Shared Hierarchies**. *CoRR, abs/1710.09767*.

Appendix

Hierarchical RL



Hierarchical RL

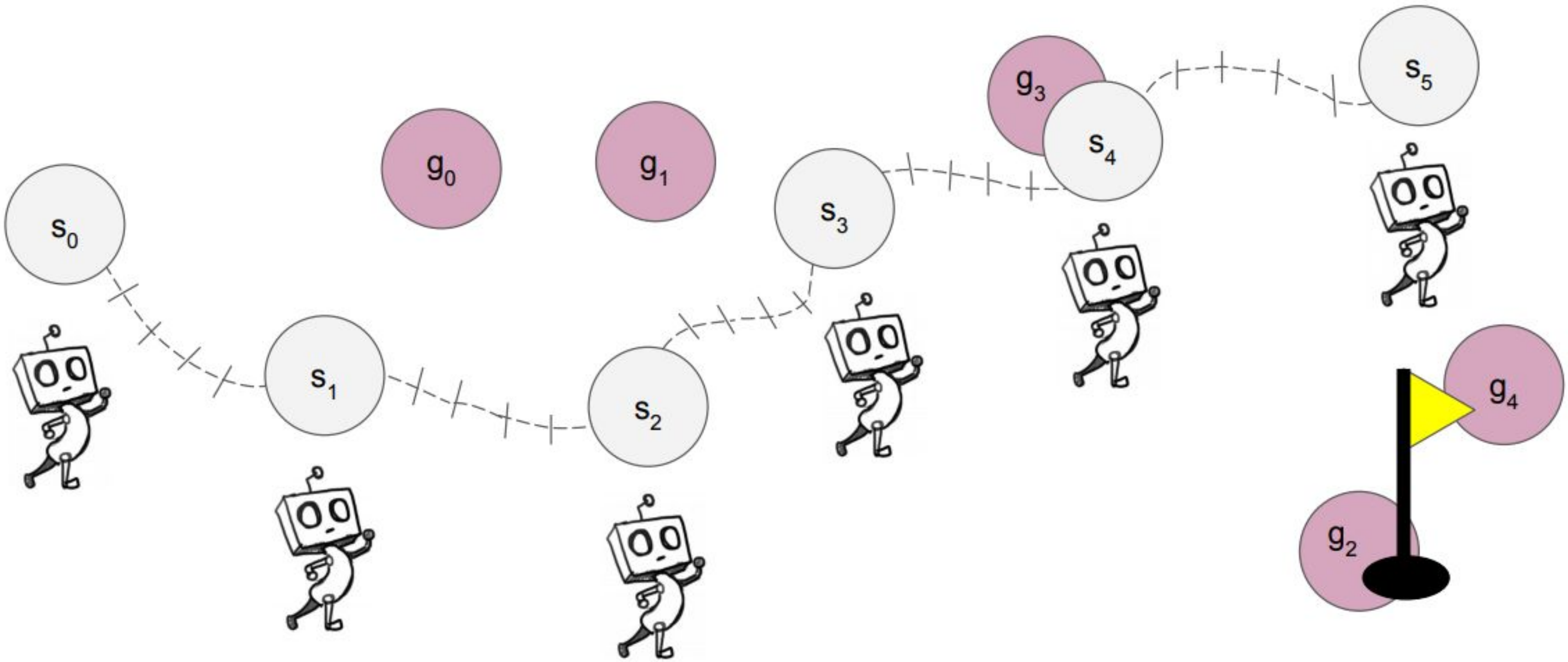
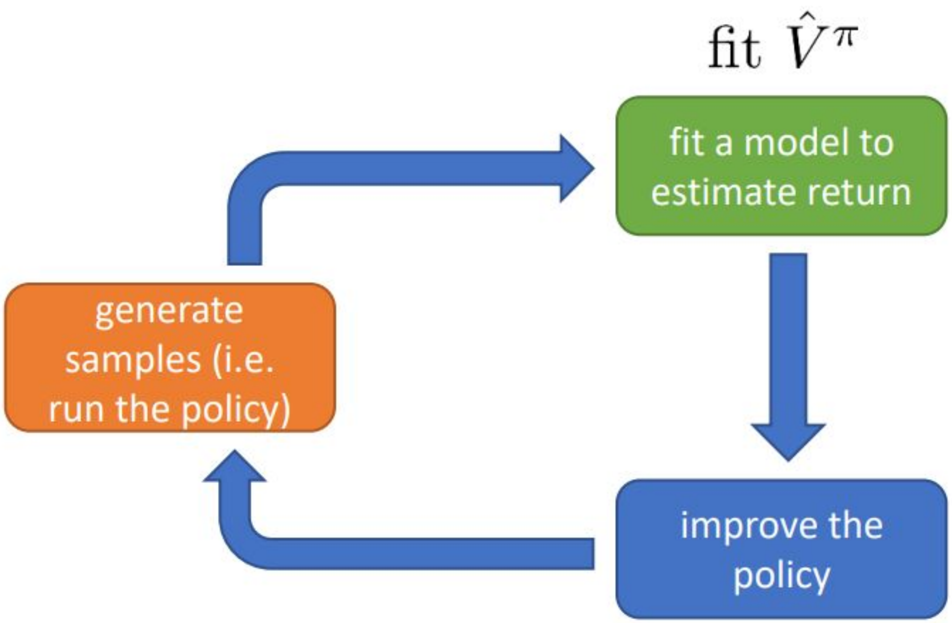
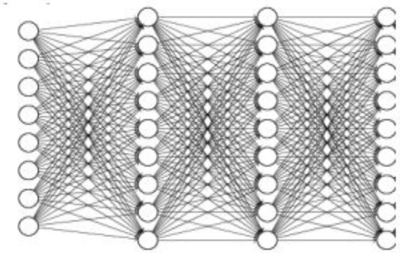


Image Credits: Levy A. et. al (2019) Learning Multi-Level Hierarchies With Hindsight, *ICLR*

Detour: A2C



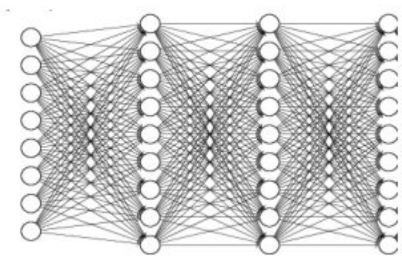
$\mathbf{s}$



$\hat{V}^{\pi}_{\phi}(\mathbf{s})$

update $\hat{V}^{\pi}_{\phi}$ using target $r + \gamma \hat{V}^{\pi}_{\phi}(\mathbf{s}')$

$\mathbf{s}$



$\pi_{\theta}(\mathbf{a}|\mathbf{s})$

evaluate $\hat{A}^{\pi}(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \hat{V}^{\pi}_{\phi}(\mathbf{s}') - \hat{V}^{\pi}_{\phi}(\mathbf{s})$

$\nabla_{\theta} J(\theta) \approx \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}|\mathbf{s}) \hat{A}^{\pi}(\mathbf{s}, \mathbf{a})$

$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta)$

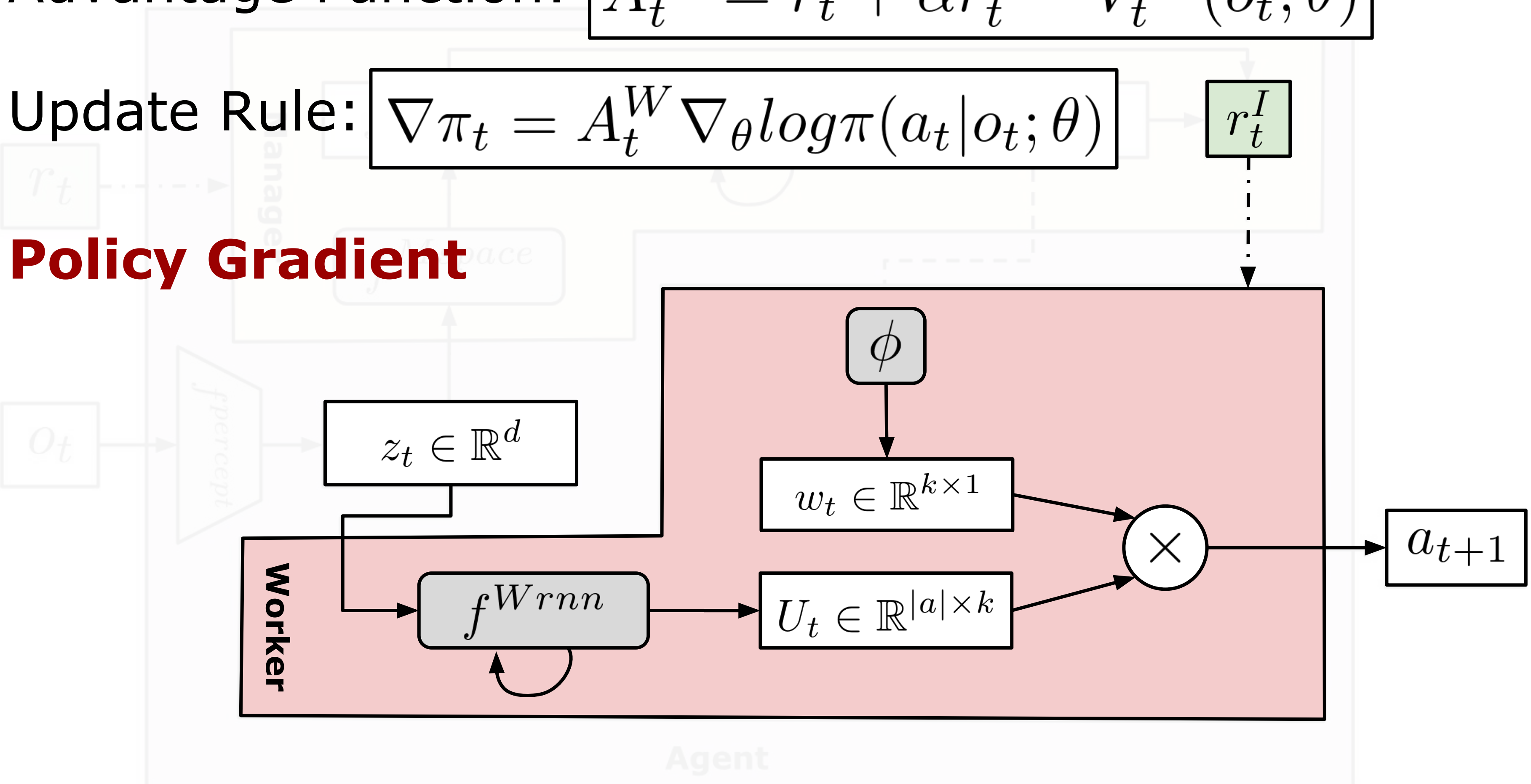
Image Credits: Sergey Levine (2018), CS 294-112 (Lecture 6)

FeUdal Networks (FUN)

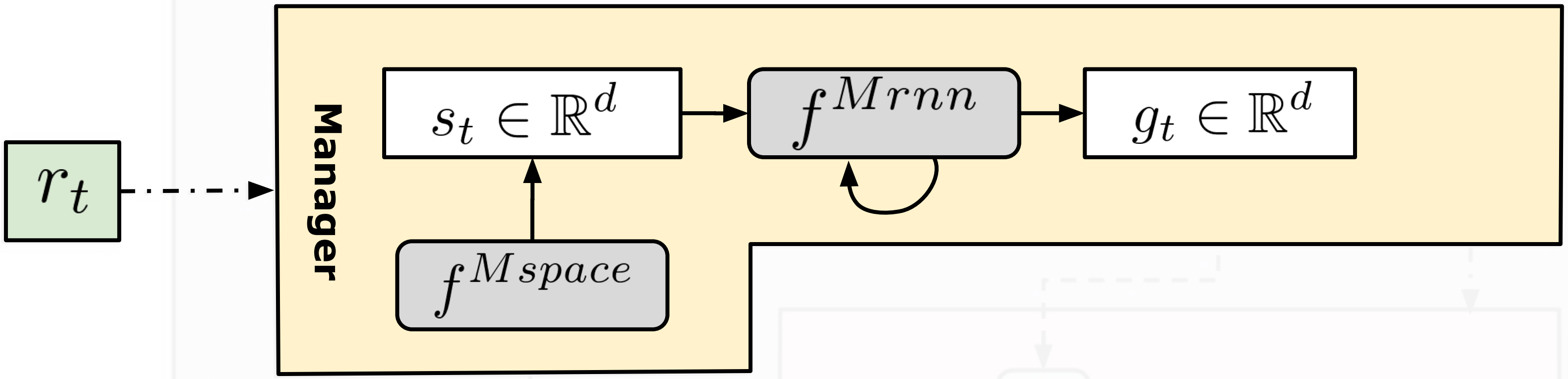
Advantage Function: $$\boxed{A_t^W = r_t + \alpha r_t^I - \hat{V}_t^W(o_t;\theta)}$$

Update Rule: $$\boxed{\nabla \pi_t = A_t^W \nabla_\theta log\pi(a_t|o_t;\theta)}$$

Policy Gradient



FeUdal Networks (FUN)



Advantage Function: $\boxed{A_t^M = r_t - \hat{V}_t^M(o_t;\theta)}$

Update Rule: $\boxed{\nabla g_t = A_t^M \nabla_\theta d_{cos}(s_{t+c} - s_t, g_t(\theta))}$

Transition Policy Gradient

FeUdal Networks (FUN)

Transition Policy Gradient

$$\nabla_\theta g_t = \mathbb{E}_{\pi_{t,\theta}}[(R_t - V(s_t))\nabla_\theta log(\pi^{TP}_{t,\theta}(s_{t+c}|s_t))]$$

$$= \mathbb{E}[(R_t - V(s_t))\nabla_\theta log(p(s_{t+c}|s_t, \theta))]$$

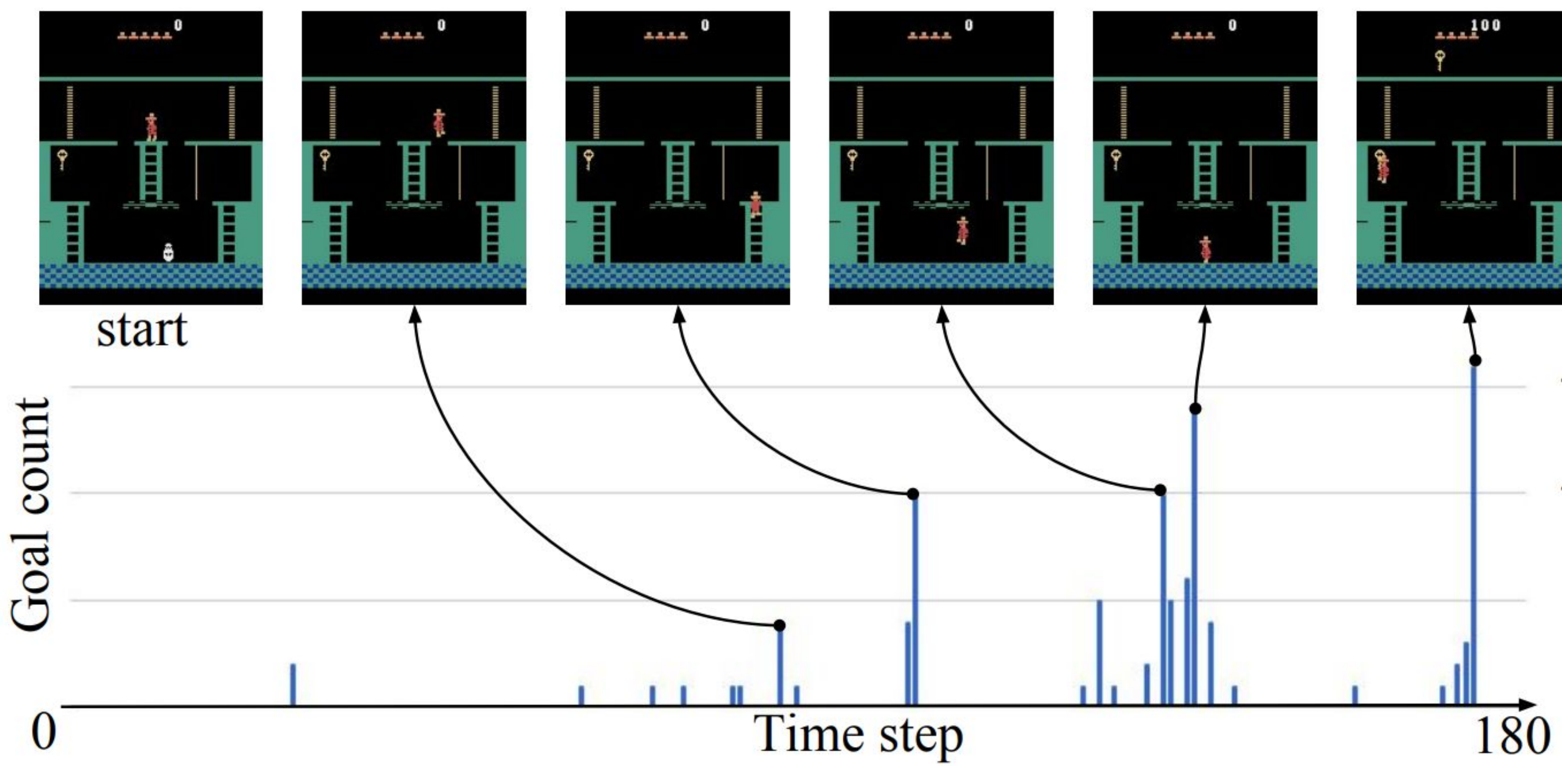
Assumption:

- Worker will eventually learn to follow the goal directions
- Direction in state-space follows von Mises-Fisher distribution

$$p(s_{t+c}|s_t, \theta) \; \alpha \; \exp(d_{cos}(s_{t+c} - s_t, g_t(\theta)))$$

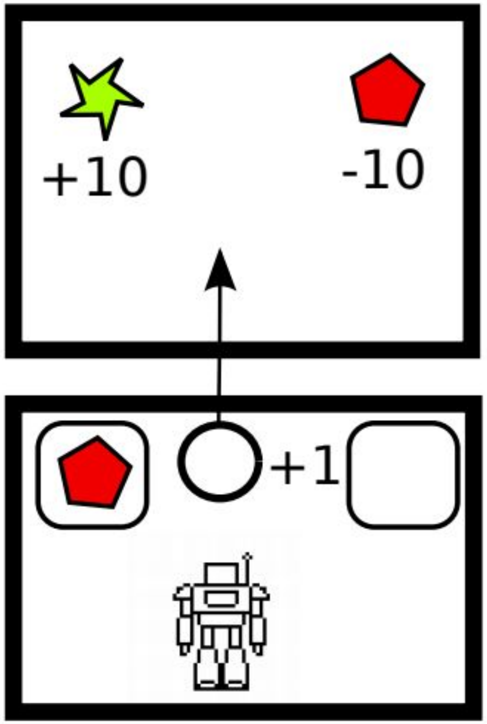
FeUdal Networks (FUN)

Learnt sub-goals by Manager



FeUdal Networks (FUN)

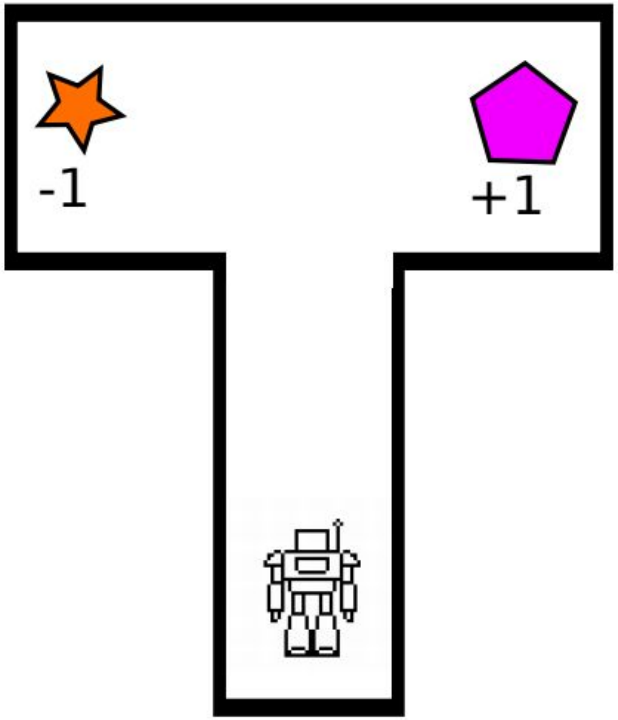
Memory Task: Non-Match



non-match

FeUdal Networks (FUN)

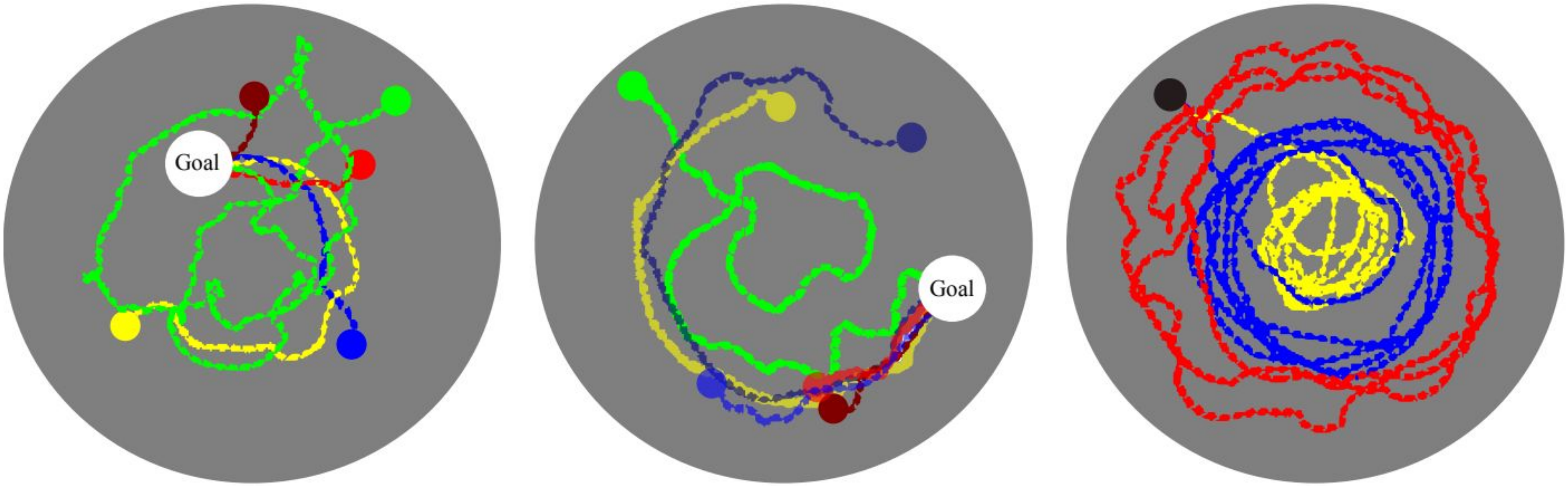
Memory Task: T-Maze



T-maze

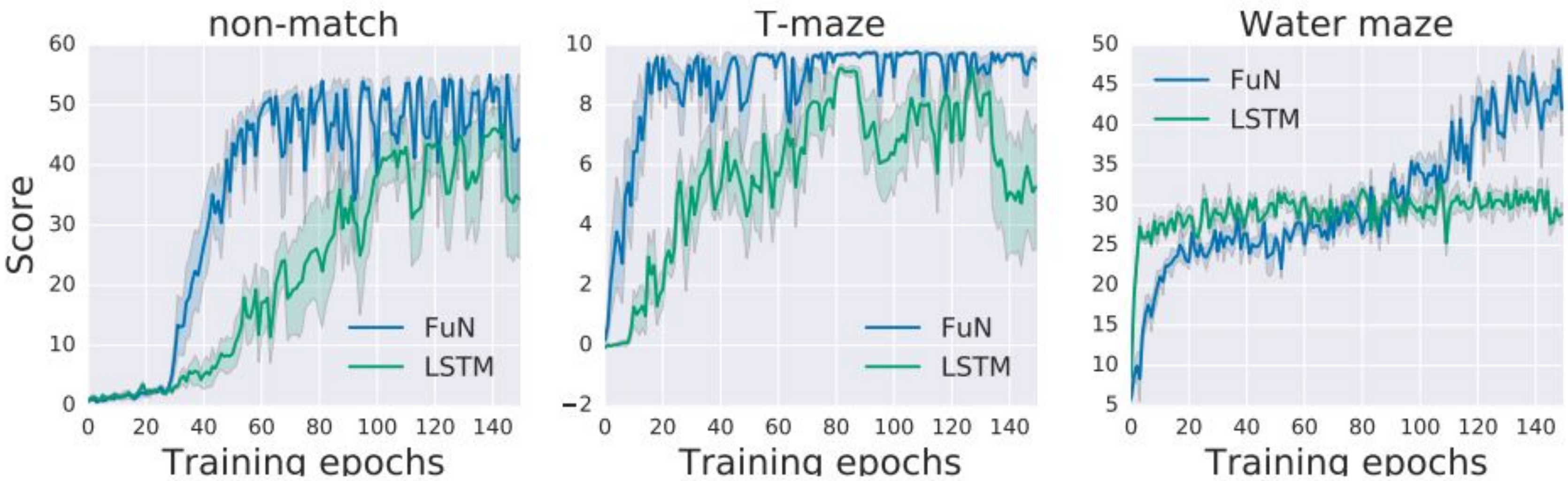
FeUdal Networks (FUN)

Memory Task: Water-Maze



FeUdal Networks (FUN)

Comparison

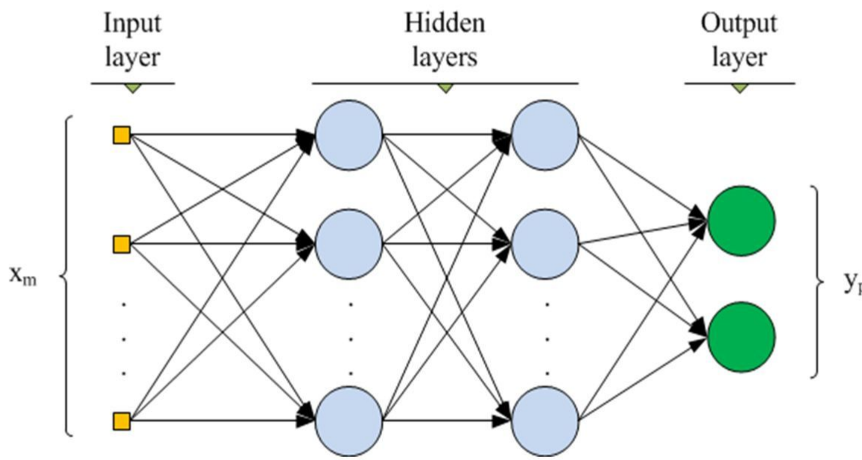


Data-Efficient HRL (HIRO)

Network Structure: TD3

Manager

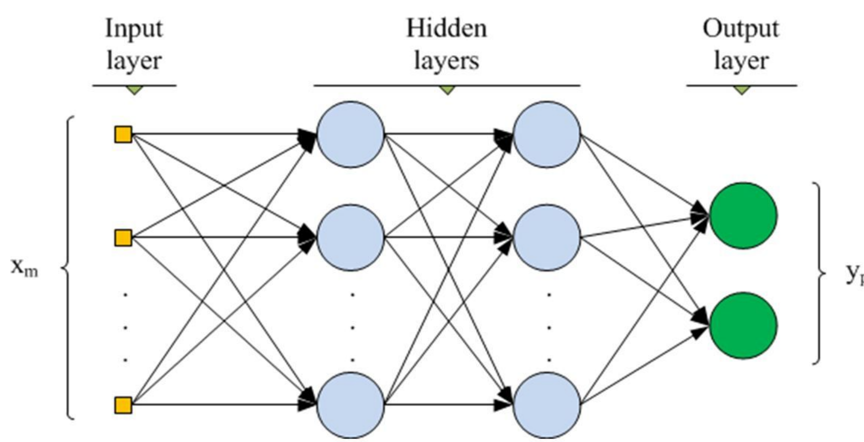
Actor-Critic with
2-layer MLP each



Dimension
of raw
observation
space

Worker

Actor-Critic with
2-layer MLP each

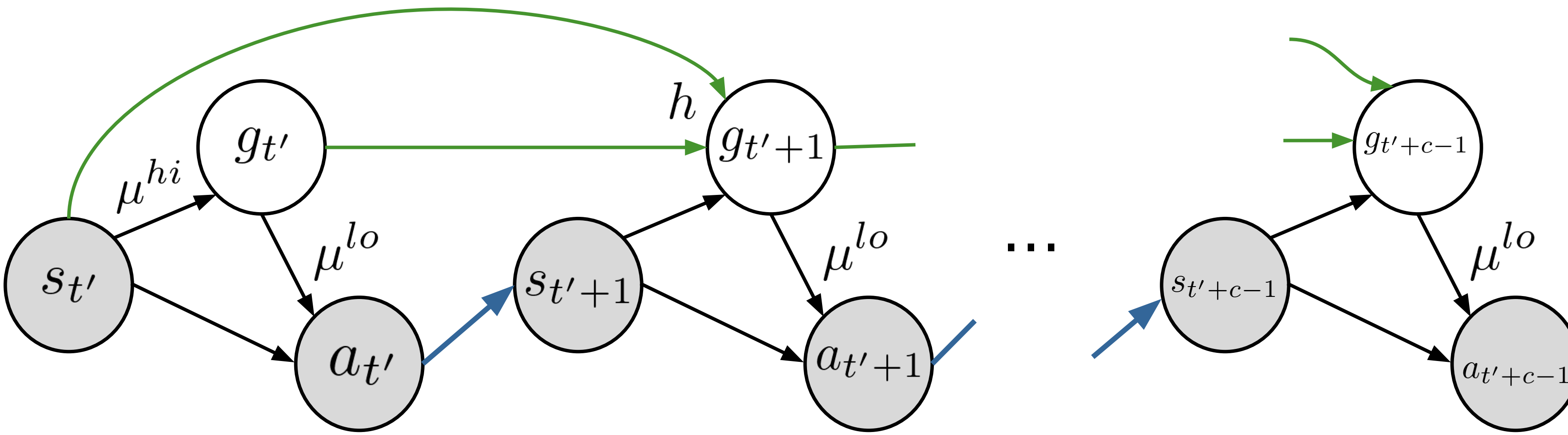


Dimension
of Action
Space

For more details: Fujimoto, S., et. al (2018). Addressing Function Approximation Error in Actor-Critic Methods. *ICML*.

Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager



$$\tilde{g}_{t'} = \underset{\tilde{g}_{t'}}{\operatorname{argmax}} \ \mu^{lo}(a_{t':t'+c-1} | s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1})$$

$$\text{where} \quad \tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1})$$

Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

$$\begin{aligned}\tilde{g}_{t'} &= \underset{\tilde{g}_{t'}}{\operatorname{argmax}}\ \mu^{lo}(a_{t':t'+c-1} \mid s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}) \\ &= \underset{\tilde{g}_{t'}}{\operatorname{argmax}}\ \underbrace{\log(\mu^{lo}(a_{t':t'+c-1} \mid s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}))}_{\alpha - \frac{1}{2}\sum_{i=t'}^{t'+c-1} ||a_i - \mu^{lo}(s_i, \tilde{g}_i)||_2^2 + \text{constant}}\end{aligned}$$

Approximately solved by generating candidate goals $\tilde{g}_{t'}$

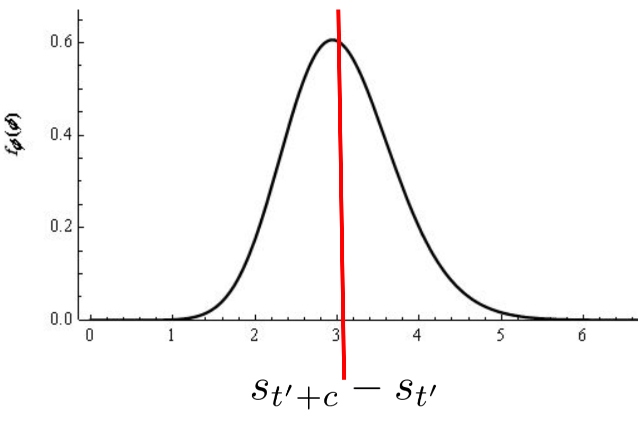
Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

$$\tilde{g}_{t'} = \underset{\tilde{g}_{t'}}{\operatorname{argmax}} \, \mu^{lo}(a_{t':t'+c-1} | s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1})$$

Approximately solved by generating candidate goals $\tilde{g}_{t'}$:

- Original goal given: $g_{t'}$
- Absolute goal based on transition observed: $s_{t'+c} - s_{t'}$
- Randomly sampled candidates:



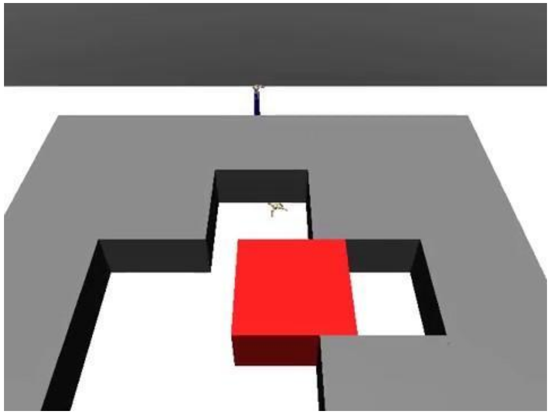
Data-Efficient HRL (HIRO)

Training

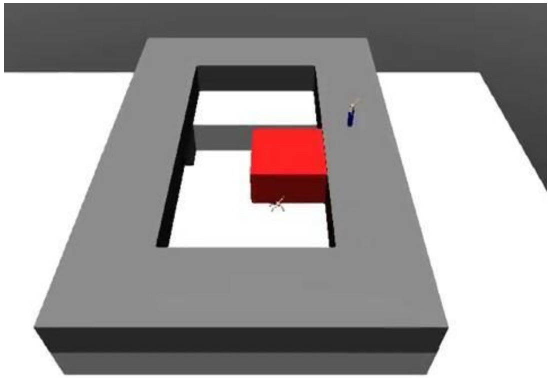
1. Collect experience $s_t, g_t, a_t, R_t, \ldots$.
2. Train μ^{lo} with experience transitions $(s_t, g_t, a_t, r_t, s_{t+1}, g_{t+1})$ using g_t as additional state observation and reward given by goal-conditioned function $r_t = r(s_t, g_t, a_t, s_{t+1}) = -||s_t + g_t - s_{t+1}||_2$.
3. Train μ^{hi} on temporally-extended experience $(s_t, \tilde{g}_t, \sum R_{t:t+c-1}, s_{t+c})$, where $\tilde{g}_t$ is re-labelled high-level action to maximize probability of past low-level actions $a_{t:t+c-1}$.
4. Repeat.

Data-Efficient HRL (HIRO)

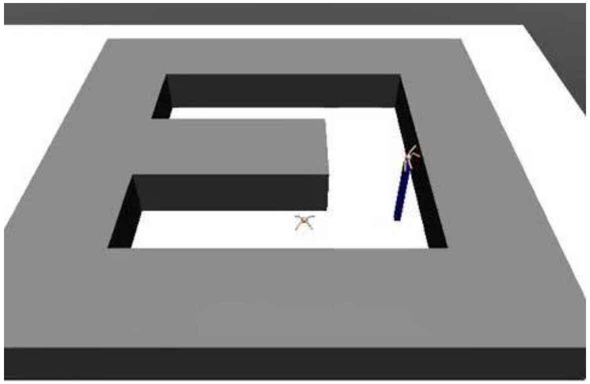
Environments



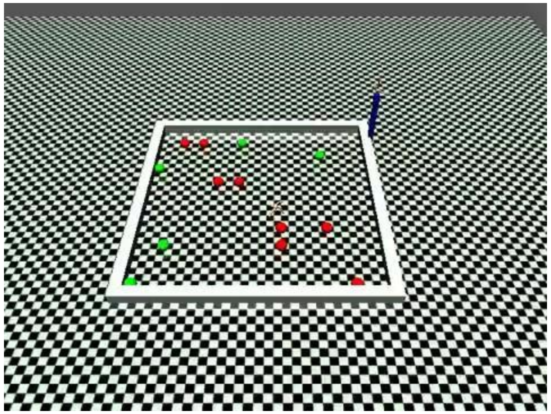
Ant Push



Ant Fall



Ant Maze



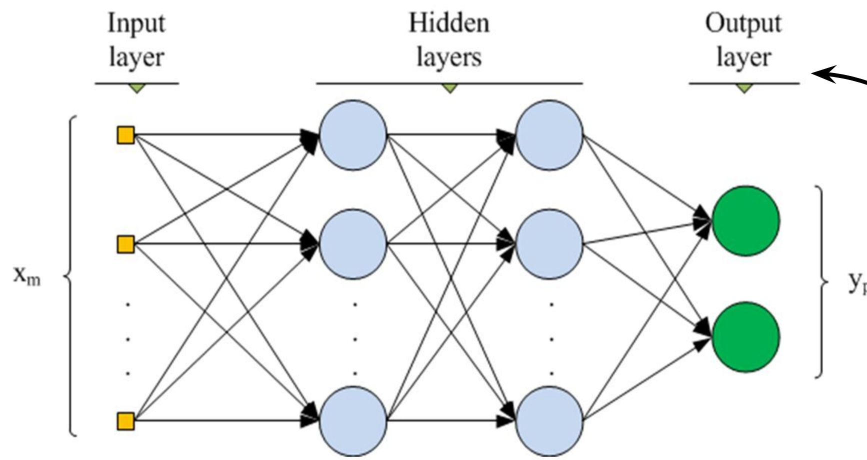
Ant Gather

Meta-Learning Shared Hierarchies (MLSH)

Network Structure: PPO

Manager

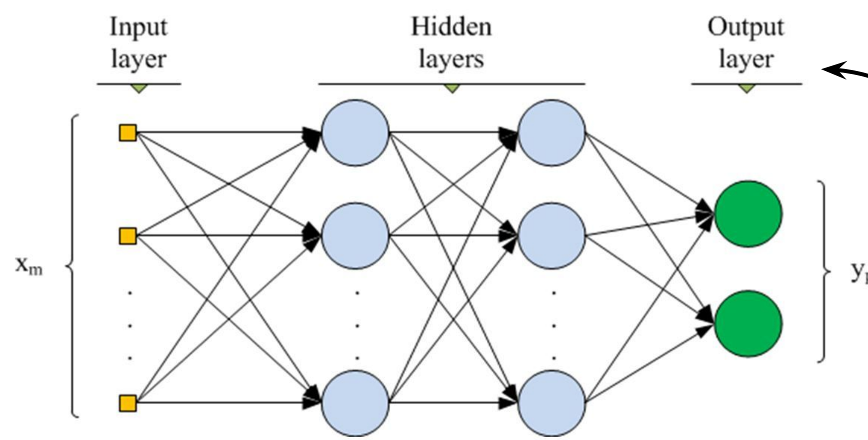
2-layer MLP with
64 hidden units



Number of
sub-policies

Each sub-policy

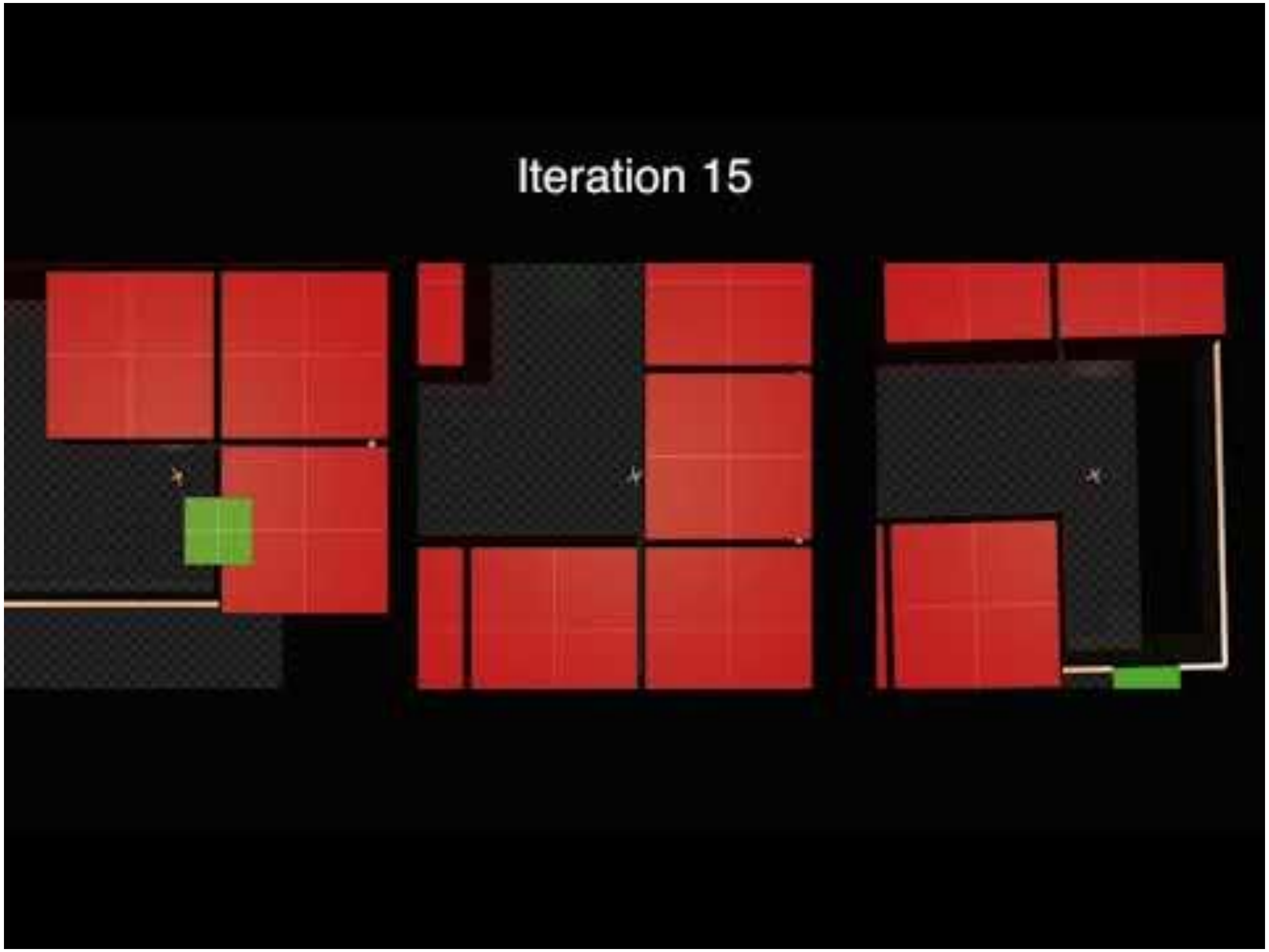
2-layer MLP with
64 hidden units



Dimension
of Action
Space

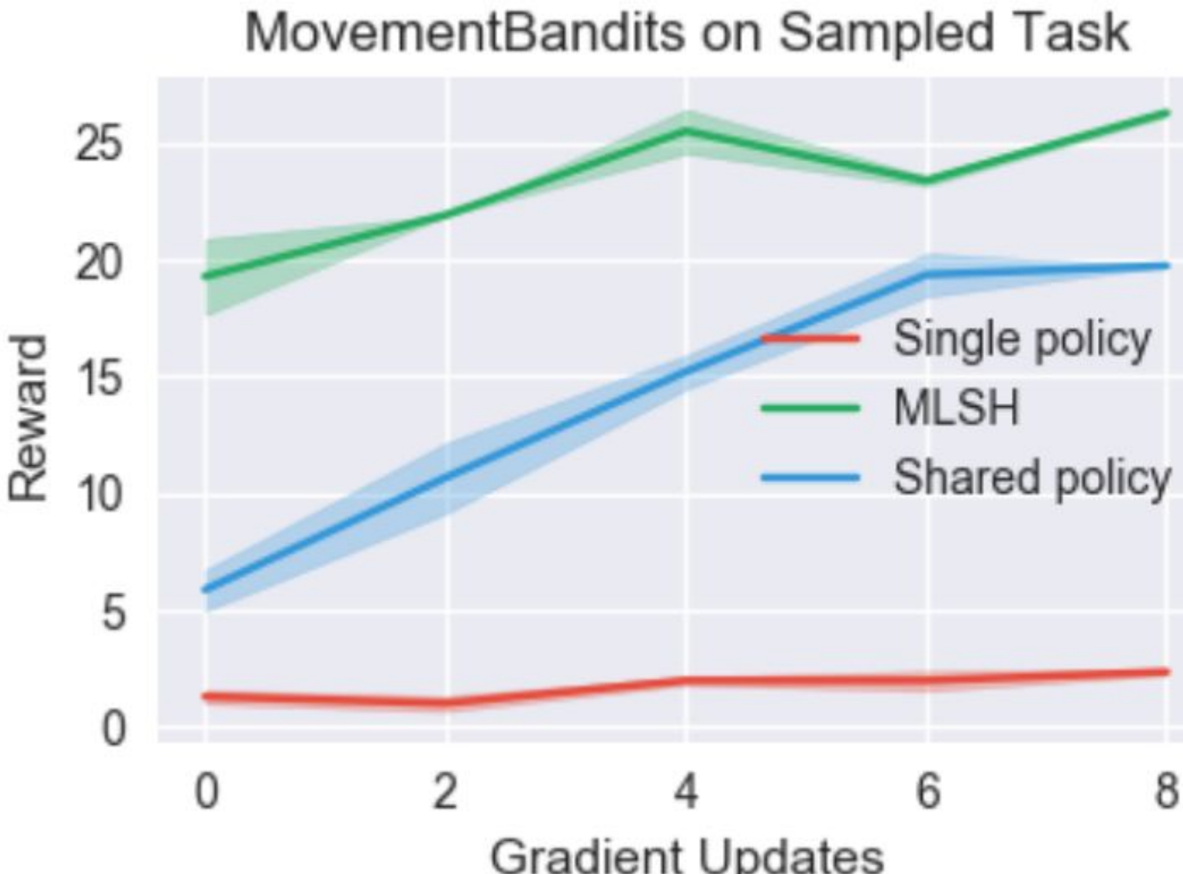
Meta-Learning Shared Hierarchies (MLSH)

Training



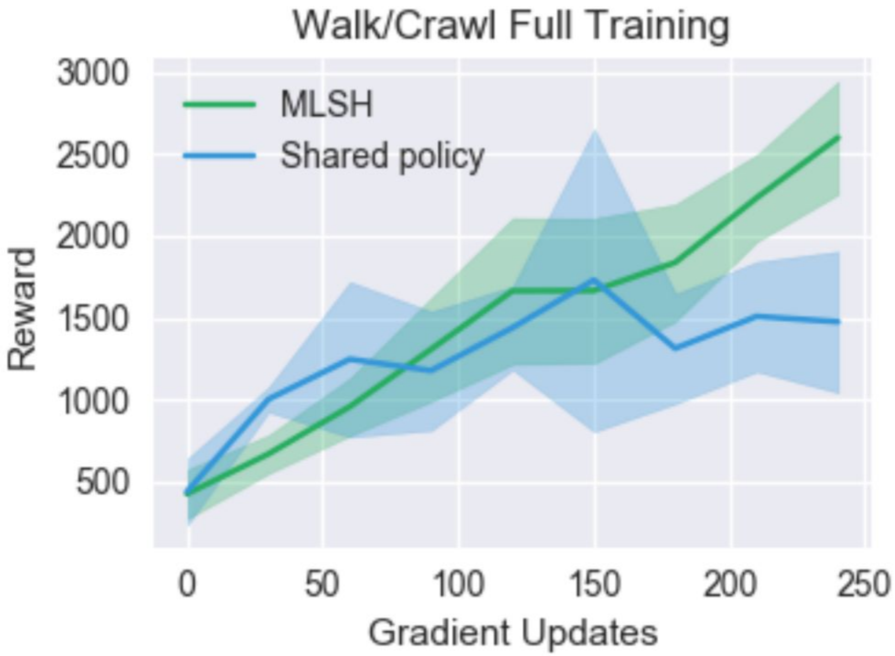
Meta-Learning Shared Hierarchies (MLSH)

Comparison



Meta-Learning Shared Hierarchies (MLSH)

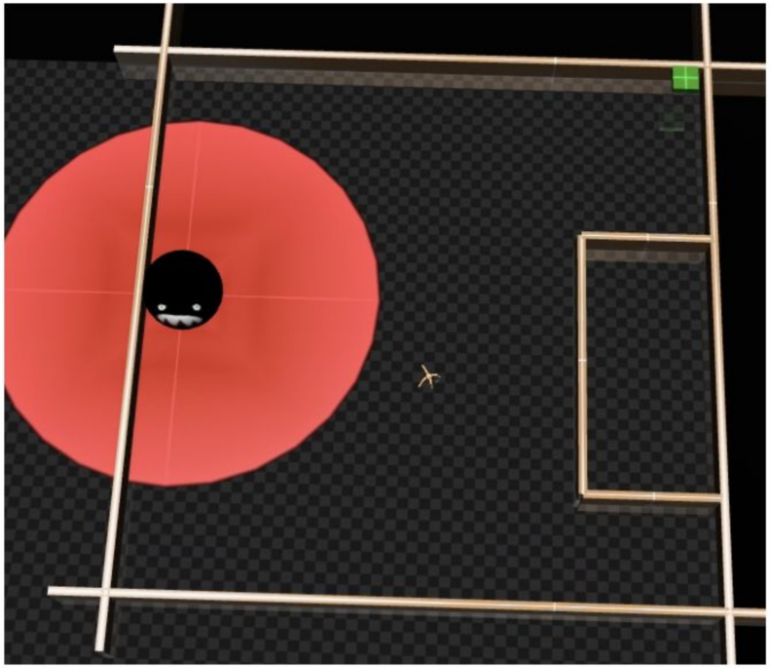
Comparison



Reward on Walk/Crawl combination task	
MLSH Transfer	14333
Shared Policy Transfer	6055
Single Policy	-643

Meta-Learning Shared Hierarchies (MLSH)

Comparison



Reward on Ant Obstacle task	
MLSH Transfer	193
Single Policy	0