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

Environment

$O_t, r_t$

Agent

Worker(s)

$\alpha_t$

Manager

$r_I, g_I$
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 Networks (FUN)
Detour: Dilated RNN

- Able to preserve memories over longer periods

For more details: Chang, S. et al. (2017). Dilated Recurrent Neural Networks, *NIPS.*
FeUdal Networks (FUN)
FeUdal Networks (FUN)
FeUdal Networks (FUN)
FeUdal Networks (FUN)
FeUdal Networks (FUN)
FeUdal Networks (FUN)

Absolute Goal
\[ \hat{g} = \Delta s = (6, 8) \]

\[ S_t \]
\(-3, 1\)

\[ S_{t+c} \]
\((3, 9)\)

c : Manager’s Horizon
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)

- 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)

\[ \sum_{i=t-c}^{t} g_i \]

\[ w_t = \phi \left( \sum_{i=t-c}^{t} g_i \right) \]
FeUdal Networks (FUN)

- **Action**
  
  \[
  \pi_{t+1} = \text{softmax}(U_t \omega_t) \\
  a_{t+1} = \arg\max_a \pi_{t+1}
  \]
FeUdal Networks (FUN)

Why not do end-to-end learning?
FeUdal Networks (FUN)

Manager & Worker: Separate Actor-Critic

Manager

Worker

Agent
FeUdal Networks (FUN)

Qualitative Analysis

Example frame    LSTM    Full FuN

sub-policy 1    sub-policy 2    sub-policy 3    sub-policy 4
FeUdal Networks (FUN)

Ablative Analysis
FeUdal Networks (FUN)

Ablative Analysis

- ms_pacman
- amidar
- gravitar
- space_invaders
- hero
- seaquest

Score vs. Training epochs graphs for different games and models.
FeUdal Networks (FUN)

Comparison

- **Asterix**
  - FU N
  - Option-Critic

- **Zaxxon**
  - FU N
  - Option-Critic
FeUdal Networks (FUN)

Action Repeat Transfer

![Graphs showing score progression for different environments and models.](image-url)
FeUdal Networks (FUN)

On-Policy Learning 😞

Experiences $(o_t, a_t, o_{t+1}, r_t)$

Learning

Wastage!
Can we do better?

Off-Policy Learning 😊

Experiences $(o_t, a_t, o_{t+1}, r_t)$

Replay Buffer

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)

Input

Goal

Action

\[ s = (q, \dot{q}, z) \quad g = (\Delta q, \Delta \dot{q}, \Delta z) \quad a = \tau_{act} \]

Raw Observation Space
Data-Efficient HRL (HIRO)
Data-Efficient HRL (HIRO)
Data-Efficient HRL (HIRO)
Data-Efficient HRL (HIRO)

\[ s_{t+c} \approx s_t + g_t \]

\( c \): Manager's Horizon
Data-Efficient HRL (HIRO)

\[ s_{t+c} \approx s_t + g_t \]

\[ g_{t+1} = h(s_t, g_t, s_{t+1}) = s_t + g_t - s_{t+1} \]
\[ s_{t+c} \approx s_t + g_t \]

- 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)

Environment

$O_t, r_t$

Manager

$\pi^I_t, g_t$

Worker(s)

Agent

$a_t$

Replay Buffer

$(s_t, g_t, a_t, r^I_t, s_{t+1}, g_{t+1})$
Data-Efficient HRL (HIRO)

Environment

Manager

Worker(s)

Agent

Replay Buffer

\[
(s_t, g_t, \sum_{i=t}^{t+c-1} r_i, s_{t+c})
\]

Replay Buffer

\[
(s_t, g_t, a_t, r_t^I, s_{t+1}, g_{t+1})
\]
Data-Efficient HRL (HIRO)

Environment

$O_t, r_t$

Manager

Worker(s)

Agent

$a_t$

$\text{Replay Buffer} (s_t, g_t, a_t, r_t^I, s_{t+1}, g_{t+1})$

Replay Buffer

$\text{PROBLEM} (s_{t+c}, o_{t+c})$
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 😞

Goal: “wear a shirt”

$t = 12$ yrs
Can we do better?

Off-Policy Learning 😞

Goal: “wear a shirt”

Same goal induces different behavior

t = 22 yrs
Can we do better?

Off-Policy Learning 😞

Goal: “wear a shirt”
Goal: “wear a dress”

Goal relabelling required!

$t = 22$ yrs
Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

\[
\begin{align*}
(s_{t'}, g_t, 
\sum_{i=t'}^{t'+c-1} r_i, s_{t'+c})
\end{align*}
\]

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

where \[
\tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1})
\]
Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

\[ \tilde{g}_{t'} = \arg\max_{\tilde{g}_{t'}} \mu_{lo}^{lo}(a_{t':t'+c-1}|s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}) \]

where \[ \tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1}) \]
Data-Efficient HRL (HIRO)
Data-Efficient HRL (HIRO)

Ant Push
Data-Efficient HRL (HIRO)

Qualitative Analysis
Data-Efficient HRL (HIRO)

Ablative Analysis

- **Ant Gather**
- **Ant Maze**
- **Ant Push**

Performance vs. Experience Samples (in millions)

- HIRO
- With lower-level re-labelling
- With pre-training
- No off-policy correction
- No HRL
## Data-Efficient HRL (HIRO)

### Comparison

<table>
<thead>
<tr>
<th></th>
<th>Ant Gather</th>
<th>Ant Maze</th>
<th>Ant Push</th>
<th>Ant Fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIRO</td>
<td>3.02±1.49</td>
<td>0.99±0.01</td>
<td>0.92±0.04</td>
<td>0.66±0.07</td>
</tr>
<tr>
<td>FuN representation</td>
<td>0.03 ± 0.01</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>FuN transition PG</td>
<td>0.41 ± 0.06</td>
<td>0.0 ± 0.0</td>
<td>0.56 ± 0.39</td>
<td>0.01 ± 0.02</td>
</tr>
<tr>
<td>FuN cos similarity</td>
<td>0.85 ± 1.17</td>
<td>0.16 ± 0.33</td>
<td>0.06 ± 0.17</td>
<td>0.07 ± 0.22</td>
</tr>
<tr>
<td>FuN</td>
<td>0.01 ± 0.01</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>SNN4HRL</td>
<td>1.92 ± 0.52</td>
<td>0.0 ± 0.0</td>
<td>0.02 ± 0.01</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>VIME</td>
<td>1.42 ± 0.90</td>
<td>0.0 ± 0.0</td>
<td>0.02 ± 0.02</td>
<td>0.0 ± 0.0</td>
</tr>
</tbody>
</table>
Data-Efficient HRL (HIRO)

Comparison

![Graph showing performance of HIRO, VIME, and SNN4HRL](image-url)

- **HIRO**
- **VIME**
- **SNN4HRL**
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
Computer Vision practice:
- Train on ImageNet
- Fine tune on actual task
Meta-Learning Shared Hierarchies (MLSH)

Computer Vision practice:
- Train on ImageNet
- Fine tune on actual task

How to generalize this to behavior learning?

Slide Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)
Meta-Learning Shared Hierarchies (MLSH)

Environment A

Environment B

Meta-RL Algorithm

“Fast” RL Agent

Image Credits: Pieter Abbeel, Metal-Learning Symposium (NIPS 2017)
Meta-Learning Shared Hierarchies (MLSH)

Environment A

Environment B

...
Meta-Learning Shared Hierarchies (MLSH)

**GOAL:** Find sub-policies that enable fast learning of master policy $\theta$
Meta-Learning Shared Hierarchies (MLSH)

**GOAL:** Find sub-policies that enable fast learning of master policy $\theta$

$$\max_\phi E_{M \sim P_M, t=0 \ldots T-1}[R]$$
Initialize $\phi$
repeat
    Initialize $\theta$
    Sample task $M \sim P_M$
    for $w = 0, 1, \ldots, W$ (warmup period) do
        Collect $D$ timesteps of experience using $\pi_{\phi, \theta}$
        Update $\theta$ to maximize expected return from $1/N$ timescale viewpoint
    end for
    for $u = 0, 1, \ldots, U$ do
        Collect $D$ timesteps of experience using $\pi_{\phi, \theta}$
        Update $\theta$ to maximize expected return from $1/N$ timescale viewpoint
        Update $\phi$ to maximize expected return from $1/N$ timescale viewpoint
    end for
until convergence
Meta-Learning Shared Hierarchies (MLSH)

Initialize $\phi$
repeat
  Initialize $\theta$
  Sample task $M \sim P$
  for $w = 0, 1, \ldots, W$
    Collect $D$ timesteps of experience using $\pi_{\phi, \theta}$
    Update $\theta$ to maximize expected return from $1/N$ timescale viewpoint
    Update $\phi$ to maximize expected return from full timescale viewpoint
  end for
for $u = 0, 1, \ldots, U$ (joint update period) do
  Collect $D$ timesteps of experience using $\pi_{\phi, \theta}$
  Update $\theta$ to maximize expected return from $1/N$ timescale viewpoint
  Update $\phi$ to maximize expected return from full timescale viewpoint
end for
until convergence
Meta-Learning Shared Hierarchies (MLSH)

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

\[ \phi_1 \quad \phi_2 \]
Meta-Learning Shared Hierarchies (MLSH)

Comparison

MLSH

Naive PPO
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis
Meta-Learning Shared Hierarchies (MLSH)

Four Rooms

Goal states are given a terminal value of 1

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

Four Rooms on Sampled Task

- Reward
- Timesteps

- MLSH
- PPO baseline
- Option Critic (4 options)
- Actor Critic baseline
Summary

**FUN**
- Directional goals
- Dilated RNN
- Transition Policy Gradient

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

**MLSH**
- Generalization in RL algorithm
- Inspired from “Options” framework
Discussion

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

▪ Do we need discrete # of sub-policies?

▪ Future prospects of HRL? More hierarchies?
Thank you for your attention!
Any Questions?
Let’s go and have lunch!
References


Appendix
Hierarchical RL

Environment

Manager

Worker(s)

Agent

$O_t, r_t$

$\alpha_t$

$r_I, g_I$
Hierarchical RL

**Detour: A2C**

- **generate samples (i.e. run the policy)**
- **fit a model to estimate return**
- **fit $\hat{V}^\pi$**
- **improve the policy**

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

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

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

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

*Image Credits: Sergey Levine (2018), CS 294-112 (Lecture 6)*
FeUdal Networks (FUN)

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

Update Rule: \[
\nabla \pi_t = A_t^W \nabla_\theta \log \pi(a_t|o_t; \theta)
\]

Policy Gradient

Worker

\[ z_t \in \mathbb{R}^d \]

\[ f_{\text{RNN}} \]

\[ w_t \in \mathbb{R}^{k \times 1} \]

\[ \phi \]

\[ U_t \in \mathbb{R}^{a \times k} \]

Agent

\[ a_{t+1} \]
FeUdal Networks (FUN)

Manager

Advantage Function:

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

Update Rule:

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

Transition Policy Gradient
FeUdal Networks (FUN)

Transition Policy Gradient

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

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

Assumption:

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

\[ p(s_{t+c} | s_t, \theta) \propto \exp(d_{cos}(s_{t+c} - s_t, g_t(\theta))) \]
FeUdal Networks (FUN)

Learnt sub-goals by Manager
FeUdal Networks (FUN)

Memory Task: Non-Match

non-match
FeUdal Networks (FUN)

Memory Task: T-Maze
FeUdal Networks (FUN)

Memory Task: Water-Maze
FeUdal Networks (FUN)

Comparison

![Graphs comparing non-match, T-maze, and Water maze performance between FuN and LSTM models over training epochs.](image)
FeUdal Networks (FUN)

Comparison

Graphs showing performance comparison of FuN, LSTM, and dLSTM across different games:
- ms_pacman
- amidar
- gravitar
- space_invaders
- hero
- seaquest

Each graph plots score against training epochs, illustrating how different models perform over time.
Data-Efficient HRL (HIRO)

Network Structure: TD3

Manager
Actor-Critic with 2-layer MLP each

Worker
Actor-Critic with 2-layer MLP each

Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

\[ \tilde{g}_{t'} = \arg\max_{\tilde{g}_{t'}} \mu^{lo}(a_{t':t'+c-1}|s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}) \]

where \[ \tilde{g}_{t'+1} = h(s_{t'}, \tilde{g}_{t'}, s_{t'+1}) \]
Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

\[ \tilde{g}_{t'} = \arg\max_{\tilde{g}_{t'}} \mu^{lo}(a_{t':t'+c-1}|s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}) \]

\[ = \arg\max_{\tilde{g}_{t'}} \log(\mu^{lo}(a_{t':t'+c-1}|s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1})) \]

\[ \alpha - \frac{1}{2} \sum_{i=t'}^{t'+c-1} ||a_i - \mu^{lo}(s_i, \tilde{g}_i)||^2_2 + \text{constant} \]

Approximately solved by generating candidate goals \( \tilde{g}_{t'} \)
Data-Efficient HRL (HIRO)

Off-Policy Correction for Manager

\[ \tilde{g}_{t'} = \underset{\tilde{g}_{t'}}{\text{argmax}} \mu^l_0(a_{t':t'+c-1}|s_{t':t'+c-1}, \tilde{g}_{t':t'+c-1}) \]

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 $\mu^{lo}$ 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 $\mu^{hi}$ 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

Each sub-policy
2-layer MLP with 64 hidden units

Meta-Learning Shared Hierarchies (MLSH)

Training
Meta-Learning Shared Hierarchies (MLSH)

Comparison

![Graph showing comparison between different policies in terms of reward and gradient updates. The y-axis represents reward, ranging from 0 to 25. The x-axis represents gradient updates, ranging from 0 to 8. The graph compares single policy, MLSH, and shared policy.](image)
Meta-Learning Shared Hierarchies (MLSH)

Comparison

<table>
<thead>
<tr>
<th>Reward on Walk/Crawl combination task</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MLSH Transfer</td>
<td>14333</td>
</tr>
<tr>
<td>Shared Policy Transfer</td>
<td>6055</td>
</tr>
<tr>
<td>Single Policy</td>
<td>-643</td>
</tr>
</tbody>
</table>
## Meta-Learning Shared Hierarchies (MLSH)

### Comparison

<table>
<thead>
<tr>
<th>Reward on Ant Obstacle task</th>
<th>MLSH Transfer Single Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Recurrent Neural Network

- Useful when input data is sequential (such as in speech recognition, language modelling)

For more details: [http://colah.github.io/posts/2015-08-Understanding-LSTMs/](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
Stochastic NN for HRL (SNN4HRL)

Aims to learn useful skills during pre-training and then leverage them for learning faster in future tasks

Variational Information Maximizing Exploration (VIME)

Exploration based on maximizing information gain about agent’s belief of the environment