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

$O_t, r_t$

Manager

Worker(s)

Agent

$\alpha_t$

$r_I, g_I$
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)

DEEP LEARNING
Detour: Dilated RNN

- Able to preserve memories over longer periods

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)
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

\[ r_{t+c}^I = \frac{1}{c} \sum_{i=t}^{t+c} d_{cos}(s_{t+c} - s_i, g_i) \]
FeUdal Networks (FUN)

Manager

Worker

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

\[ f_{Mrnn} \]

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

\[ f_{Mspace} \]

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

\[ \phi \]

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

\[ \times \]

\[ a_{t+1} \]

\[ f_{Wrrn} \]

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

\[ k = 16 \ll d = 256 \]
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 Sampling**
  \[
  \pi_{t+1} = \text{softmax}(U_t w_t) \\
  a_{t+1} = \text{argmax}_a \pi_{t+1}
  \]

**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

Example frame  LSTM  Full FuN

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

Ablative Analysis

![Graphs showing score vs. training epochs for ms_pacman, amidar, and gravitar.]
FeUdal Networks (FUN)

Ablative Analysis

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

Scores vs. Training epochs for different models:
- **FuN, 0.95**
- **FuN, 0.99**
- **LSTM, 0.95**
- **LSTM, 0.99**
- **LSTM, 0.99, BPTT=100**
FeUdal Networks (FUN)

Comparison

- asterix
  - FuN
  - Option-Critic

- zaxxon
  - FuN
  - Option-Critic
FeUdal Networks (FUN)

Action Repeat Transfer

---

**frostbite**

**ms_pacman**

**amidar**

---

**Score**

**Training epochs**

---

**FuN**  **FuN transfer**  **LSTM**  **LSTM transfer**
FeUdal Networks (FUN)

On-Policy Learning 😞
FeUdal Networks (FUN)

On-Policy Learning 😞

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

Learning

Wastage!
Can we do better?
Can we do better?

Off-Policy Learning 😊

Experiences

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

Replay Buffer

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

\[ s = (q, \dot{q}, z) \]

Goal

\[ g = (\Delta q, \Delta \dot{q}, \Delta z) \]

Action

\[ \alpha = \tau_{act} \]

Raw Observation Space
Data-Efficient HRL (HIRO)

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

Environment

$O_t, r_t$

Manager

Worker(s)

Agent

$\alpha_t$

$\theta^I, g_t$

Replay Buffer

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

- Environment
  - \( O_t, r_t \)
  - \( a_t \)

- Agent
  - Manager
    - \( r_t^I, g_t \)
  - Worker(s)

- Replay Buffer
  - \( (s_t, g_t, \sum_{i=t}^{t+c-1} r_i, s_{t+c}) \)
  - \( (s_t, g_t, a_t, r_t^I, s_{t+1}, g_{t+1}) \)
Data-Efficient HRL (HIRO)

Environment

Agent

Manager

Worker(s)

Replay Buffer

\( O_t, r_t \)

\( a_t \)

\( r^I_t, g_t \)

\( s_t, g_t, a_t, r^I_t, s_{t+1}, g_{t+1} \)

\( \sigma_{t+c} \)
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

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

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

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

Off-Policy Correction for Manager

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

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

Environment

Manager

Worker(s)

Agent

Replay Buffer

\[
\begin{align*}
(s_t, \tilde{g}_t, \sum_{i=t}^{t+c-1} r_i, s_{t+c})
\end{align*}
\]

Replay Buffer

\[
\begin{align*}
(s_t, g_t, a_t, r^I_t, s_{t+1}, g_{t+1})
\end{align*}
\]
Data-Efficient HRL (HIRO)

Ant Push
Data-Efficient HRL (HIRO)

Qualitative Analysis
Data-Efficient HRL (HIRO)

Ablative Analysis

Ant Gather

Ant Maze

Ant Push

Experience Samples (in millions)

Performance

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

![Ant Gather Graph]

- HIRO
- VIME
- SNN4HRL

Experience Samples (in millions)

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

How to generalize this to behavior learning?

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

Environment A

Environment B

...
Meta-Learning Shared Hierarchies (MLSH)

Environment A

Environment B

Meta-RL Algorithm

“Fast” RL Agent

Environment H

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 $\theta$
Meta-Learning Shared Hierarchies (MLSH)

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

$$\text{maximize}_{\phi} \ E_{M \sim P_M, t=0 \ldots T-1}[R]$$
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 $\pi_{\phi_u, \theta_u}$
    Update $\theta$ to maximize expected return from $1/N$ timescale viewpoint
    Update $\phi$ to maximize master reward from $1/N$ timescale viewpoint
  end for
until convergence
Meta-Learning Shared Hierarchies (MLSH)

Initialize $\phi$
repeat
    Initialize $\theta$
    Sample task $M \sim \mathcal{D}$
    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
end for
until convergence
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
Meta-Learning Shared Hierarchies (MLSH)

Comparison

MLSH

Naive PPO
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis

![Graph showing performance over gradient updates for different methods.](image-url)
Meta-Learning Shared Hierarchies (MLSH)

Ablative Analysis

Movement Bandits Hyperparameter Comparison

- Reward
- MLSH Iterations (Warmup + Joint Update periods)
- 1 Subpolicy
- 2 Subpolicies
- 4 Subpolicies
- Warmup Duration: 1
- Warmup Duration: 20
Meta-Learning Shared Hierarchies (MLSH)

Four Rooms

- 4 rooms
- 4 hallways
- 4 unreliable primitive actions

Goal states are given a terminal value of 1

8 multi-step options
(to each room's 2 hallways)

Given goal location, quickly plan shortest route

All rewards zero
\( \gamma = .9 \)
Comparison

Meta-Learning Shared Hierarchies (MLSH)
**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?


Appendix
Hierarchical RL

Environment

\[ O_t, r_t \]

Manager

Worker(s)

Agent

\[ r_I, g_I \]

\[ a_t \]
Hierarchical RL

**Detour: A2C**

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

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

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

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

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

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

**Advantage Function:**

\[ A^W_t = r_t + \alpha r^I_t - \hat{V}^W_t (o_t; \theta) \]

**Update Rule:**

\[ \nabla \pi_t = A^W_t \nabla_\theta \log \pi (a_t | o_t; \theta) \]

**Policy Gradient**
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} [(R_t - V(s_t)) \nabla_\theta \log(\pi_{t,\theta}^{TP}(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

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

Learnt sub-goals by Manager
FeUdal Networks (FUN)

Memory Task: Non-Match
FeUdal Networks (FUN)

Memory Task: 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

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

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

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

<table>
<thead>
<tr>
<th>Number of sub-policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension of Action Space</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manager</th>
<th>Each sub-policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-layer MLP with 64 hidden units</td>
<td>2-layer MLP with 64 hidden units</td>
</tr>
</tbody>
</table>
Meta-Learning Shared Hierarchies (MLSH)

Training
Meta-Learning Shared Hierarchies (MLSH)

Comparison

![Graph showing comparison between single policy, MLSH, and shared policy over gradient updates. The graph plots reward against gradient updates, with MLSH consistently outperforming the other two strategies.]
Meta-Learning Shared Hierarchies (MLSH)

Comparison

![Graph showing comparison between MLSH and Shared Policy]

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

### Comparison

<table>
<thead>
<tr>
<th>Reward on Ant Obstacle task</th>
<th>MLSSH Transfer Single Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>193</td>
<td>0</td>
</tr>
</tbody>
</table>