UNIVERSITY OF MICHIGAN

DeepMind

What Can Learned Intrinsic Rewards Capture?

Zeyu Zheng*, Junhyuk Oh*, Matteo Hessel, Zhongwen Xu,
Manuel Kroiss, Hado van Hasselt, David Silver, Satinder Singh

zeyu@umich.edu
junhyuk@google.com

Motivation: Loci of Knowledge in RL

- Common structures to store knowledge in RL
    - Policies, value functions, models, state representations, ...

Motivation: Loci of Knowledge in RL

- Common structures to store knowledge in RL
  - Policies, value functions, models, state representations, …
- Uncommon structure: reward function
  - Typically from environment & immutable

Motivation: Loci of Knowledge in RL

- Common structures to store knowledge in RL
  - Policies, value functions, models, state representations, …
- Uncommon structure: reward function
  - Typically from environment & immutable
- Existing methods to store knowledge in rewards are hand-designed (e.g., reward shaping, novelty-based reward).

Motivation: Loci of Knowledge in RL

- Common structures to store knowledge in RL
  - Policies, value functions, models, state representations, ...
- Uncommon structure: reward function
  - Typically from environment & immutable
- Existing methods to store knowledge in rewards are hand-designed (e.g., reward shaping, novelty-based reward).
- Research questions
  - Can we “learn” a useful intrinsic reward function in a data-driven way?
  - What kind of knowledge can be captured by a learned reward function?

Overview

- A scalable meta-gradient framework for learning useful intrinsic reward functions across multiple lifetimes

Overview

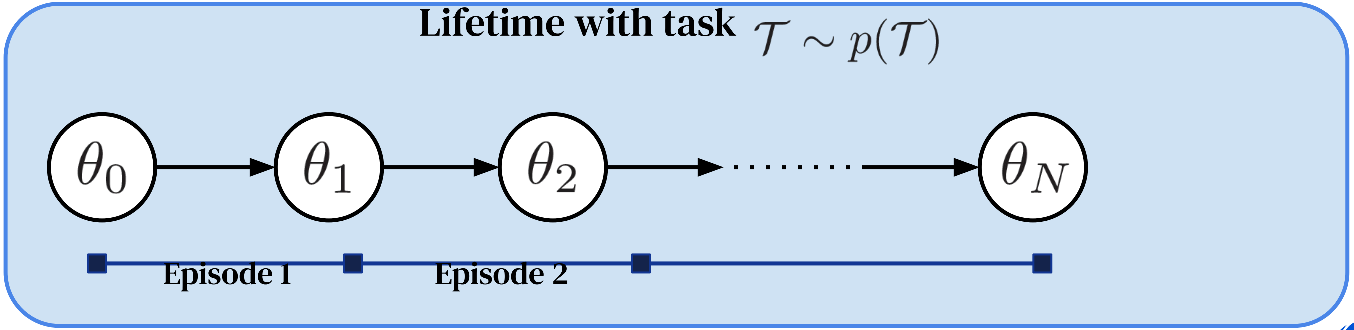
- A scalable meta-gradient framework for learning useful intrinsic reward functions across multiple lifetimes
- Learned intrinsic rewards can capture
  - interesting regularities that are useful for exploration/exploitation

Overview

- A scalable meta-gradient framework for learning useful intrinsic reward functions across multiple lifetimes
- Learned intrinsic rewards can capture
  - interesting regularities that are useful for exploration/exploitation
  - knowledge that generalises to different learning agents and different environment dynamics
  - “what to do” instead of “how to do”

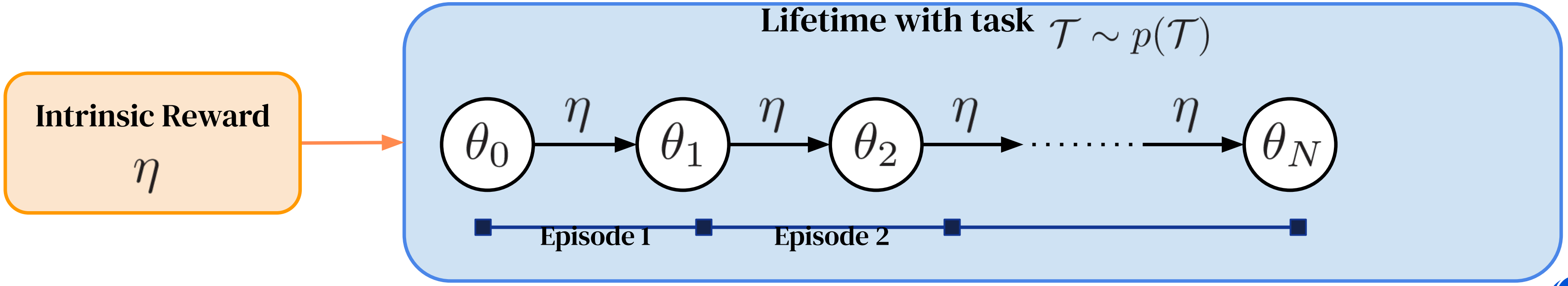
Problem Formulation: Optimal Reward Framework[Singh et al. 2010]

- **Lifetime**: an agent's entire training time which consists of many episodes and parameter updates (say *N*) given a task drawn from some distribution.



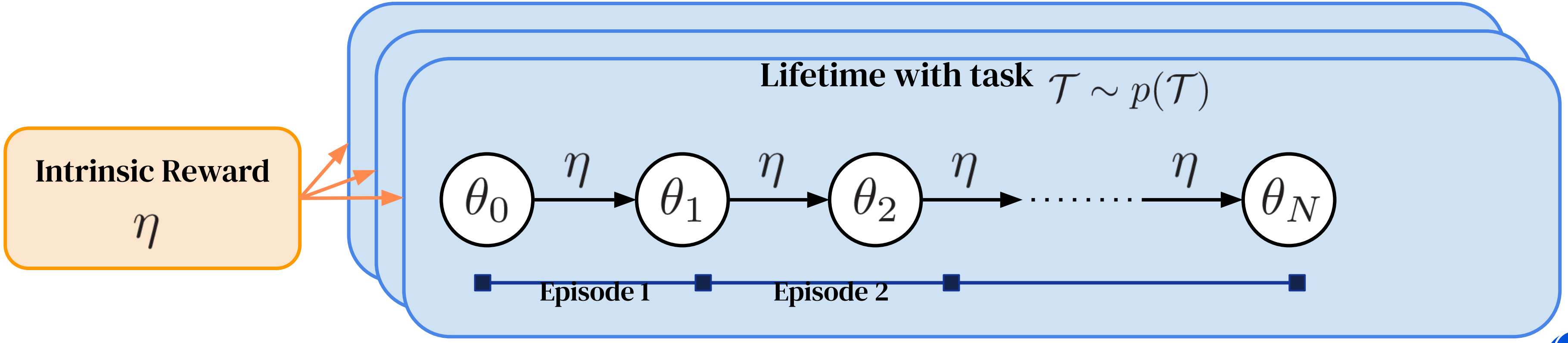
Problem Formulation: Optimal Reward Framework[Singh et al. 2010]

- **Lifetime**: an agent's entire training time which consists of many episodes and parameter updates (say *N*) given a task drawn from some distribution.
- **Intrinsic reward**: mapping from a history to a scalar.
    - Acts as a reward function when updating an agent's parameters.

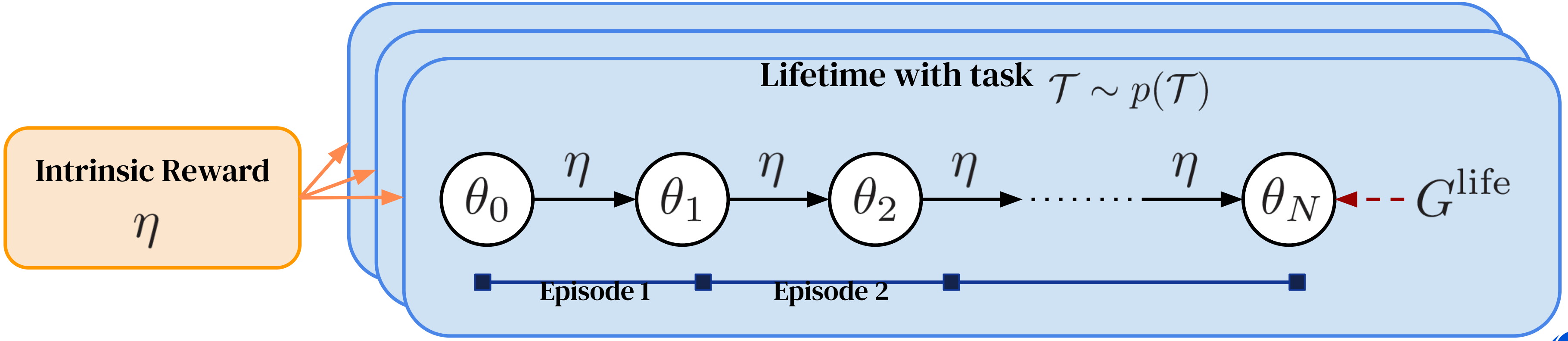


Problem Formulation: Optimal Reward Framework [Singh et al. 2010]

- **Optimal Reward Problem**: learn a single intrinsic reward function across multiple lifetimes that is optimal to train any randomly initialised policies to maximise their extrinsic rewards.

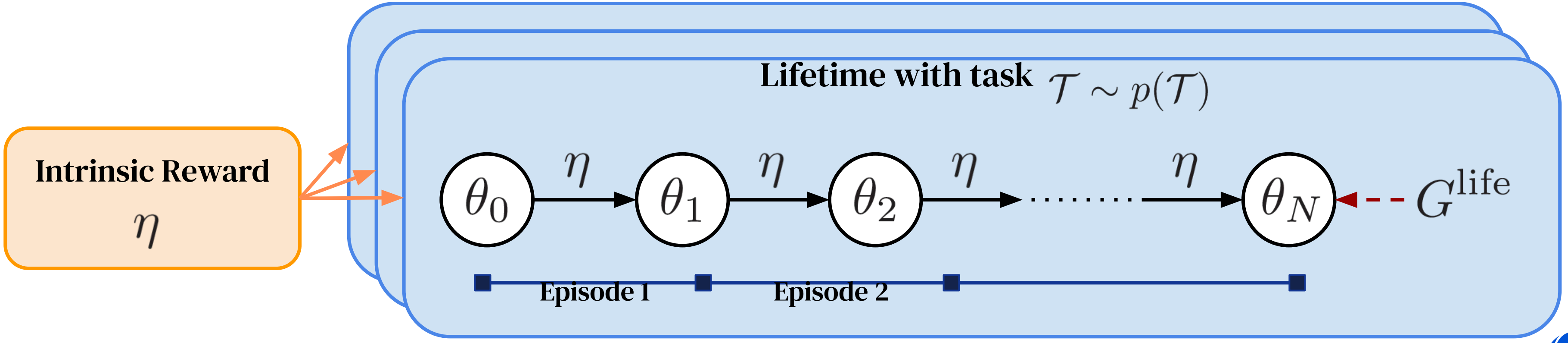


Under-explored Aspects of Good Intrinsic Rewards



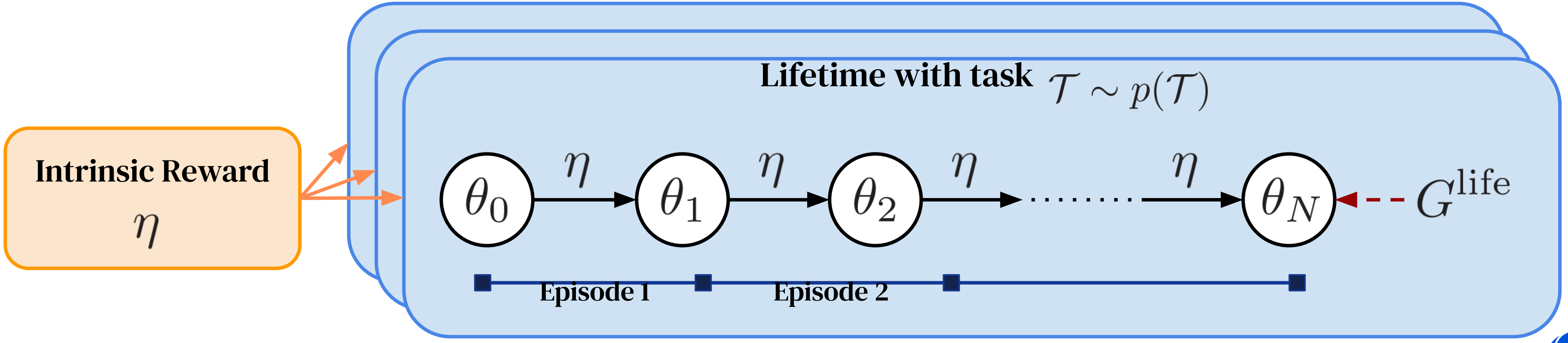
Under-explored Aspects of Good Intrinsic Rewards

- Should take into account the entire **lifetime history** for exploration



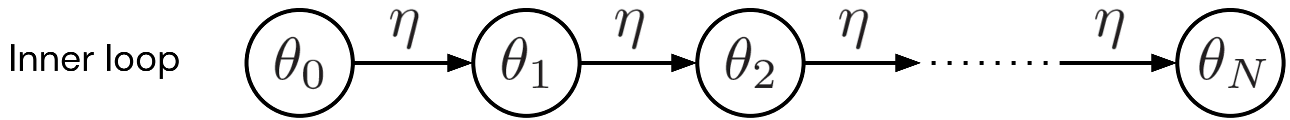
Under-explored Aspects of Good Intrinsic Rewards

- Should take into account the entire **lifetime history** for exploration
- Should maximise long-term **lifetime return** rather than episodic return to give more room for balancing exploration and exploitation across multiple episodes



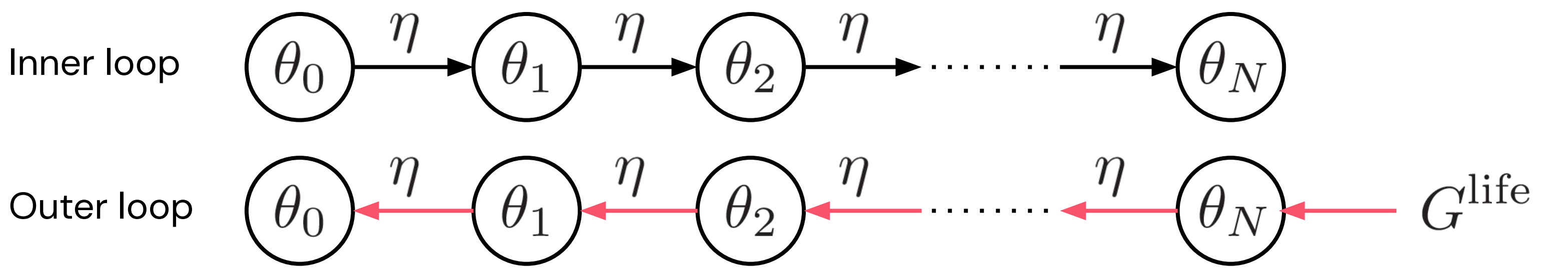
Method: Truncated Meta-Gradients with Bootstrapping

- **Inner loop**: unroll the computation graph until the end of the lifetime.



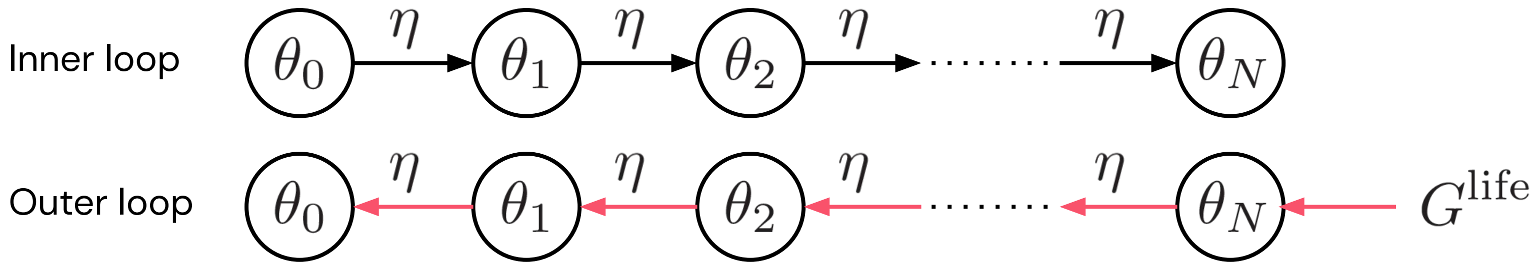
Method: Truncated Meta-Gradients with Bootstrapping

- **Inner loop**: unroll the computation graph until the end of the lifetime.
- **Outer loop**: compute the meta-gradient w.r.t. the intrinsic rewards by back-propagating through the entire lifetime.



Method: Truncated Meta-Gradients with Bootstrapping

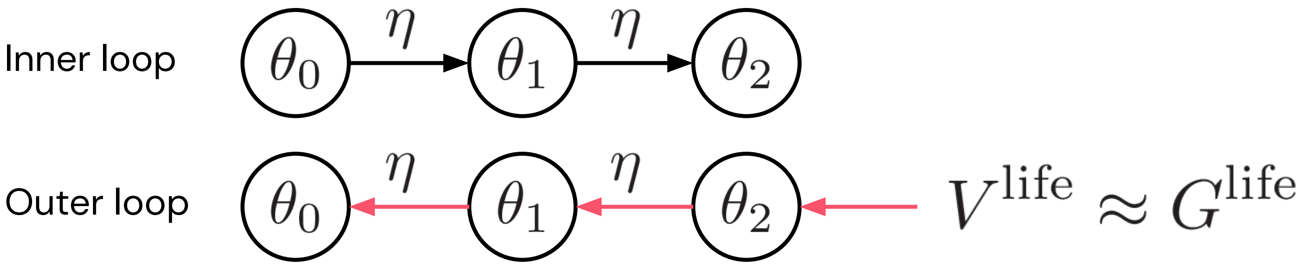
- **Inner loop**: unroll the computation graph until the end of the lifetime.
- **Outer loop**: compute the meta-gradient w.r.t. the intrinsic rewards by back-propagating through the entire lifetime.



Challenge: cannot unroll the full graph due to the memory constraint.

Method: Truncated Meta-Gradients with Bootstrapping

- Truncate the computation graph up to a few parameter updates.
- Use a **lifetime value function** to approximate the remaining rewards.
  - Assign credits to actions that lead to a larger lifetime return.



Experiments: Methodology

Experiments: Methodology

- Design a domain and a set of tasks with specific regularities

Experiments: Methodology

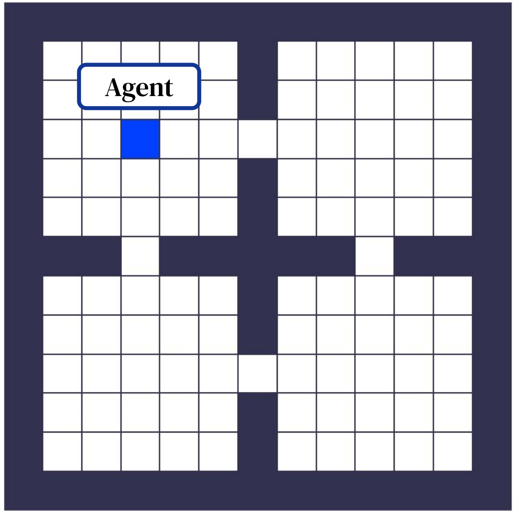
- Design a domain and a set of tasks with specific regularities
- Train an intrinsic reward function across multiple lifetimes

Experiments: Methodology

- Design a domain and a set of tasks with specific regularities
- Train an intrinsic reward function across multiple lifetimes
- Fix the intrinsic reward function and evaluate and analyse it on a new lifetime

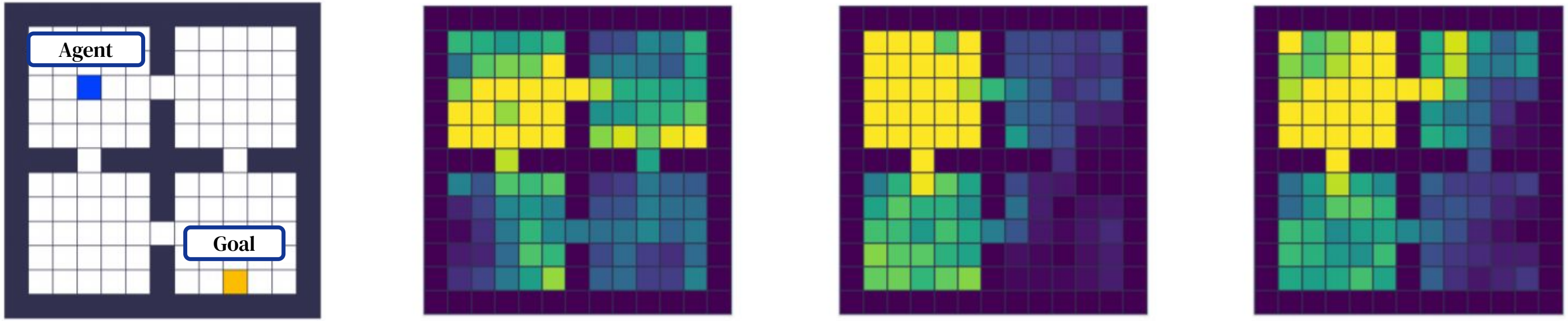
Experiment: Exploring uncertain states

- Task: find and reach the goal location (**invisible**).
    - Randomly sampled for each lifetime but fixed within a lifetime.
- An episode terminates if the agent reaches the goal.



Experiment: Exploring uncertain states

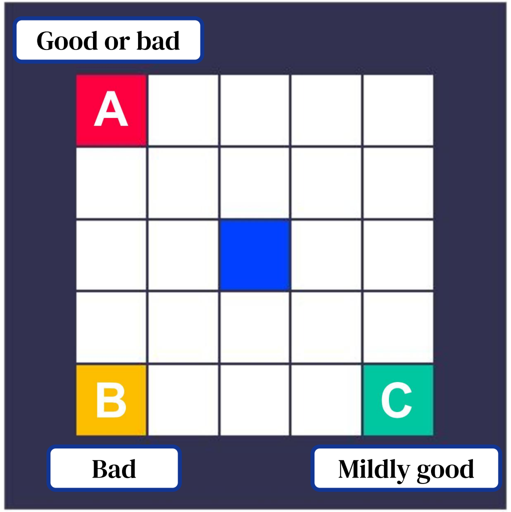
- The learned intrinsic reward encourages the agent to explore uncertain states (more efficient than count-based exploration).



(a) Room instance (b) Intrinsic (ours) (c) Extrinsic (d) Count-based

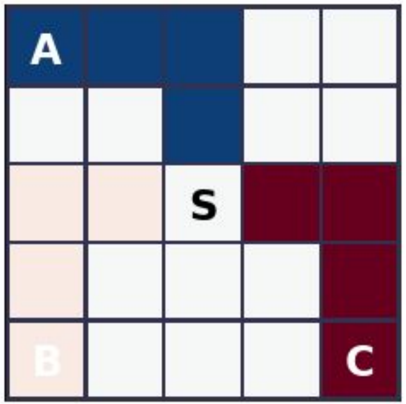
Experiment: Exploring uncertain objects

- Task: find and collect the largest rewarding object.
  - Reward for each object is randomly sampled for each lifetime.
- Requires multi-episode exploration.



Experiment: Exploring uncertain objects

- The intrinsic reward has learned to encourage exploring uncertain objects (A and C) while avoiding harmful object (B).

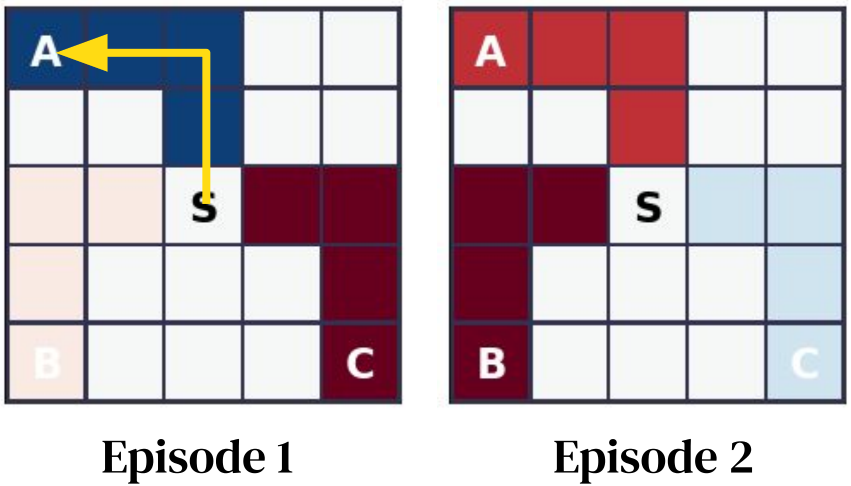


Episode 1

Visualisation of learned intrinsic rewards for each trajectory

Experiment: Exploring uncertain objects

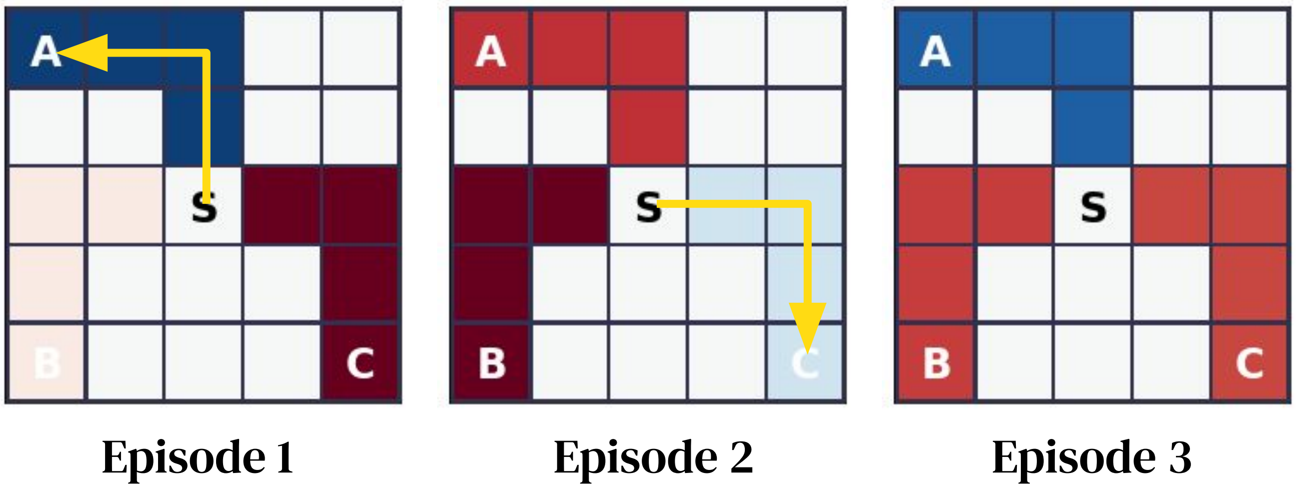
- The intrinsic reward has learned to encourage exploring uncertain objects (A and C) while avoiding harmful object (B).



Visualisation of learned intrinsic rewards for each trajectory

Experiment: Exploring uncertain objects

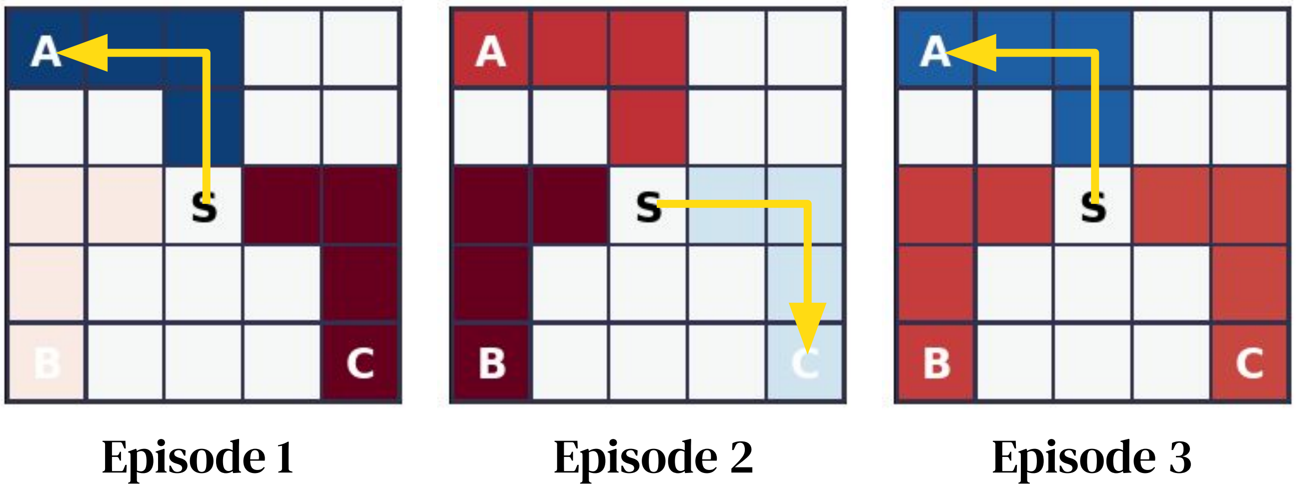
- The intrinsic reward has learned to encourage exploring uncertain objects (A and C) while avoiding harmful object (B).



Visualisation of learned intrinsic rewards for each trajectory

Experiment: Exploring uncertain objects

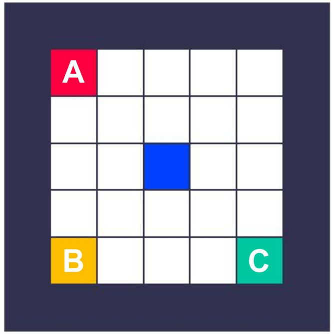
- The intrinsic reward has learned to encourage exploring uncertain objects (A and C) while avoiding harmful object (B).



Visualisation of learned intrinsic rewards for each trajectory

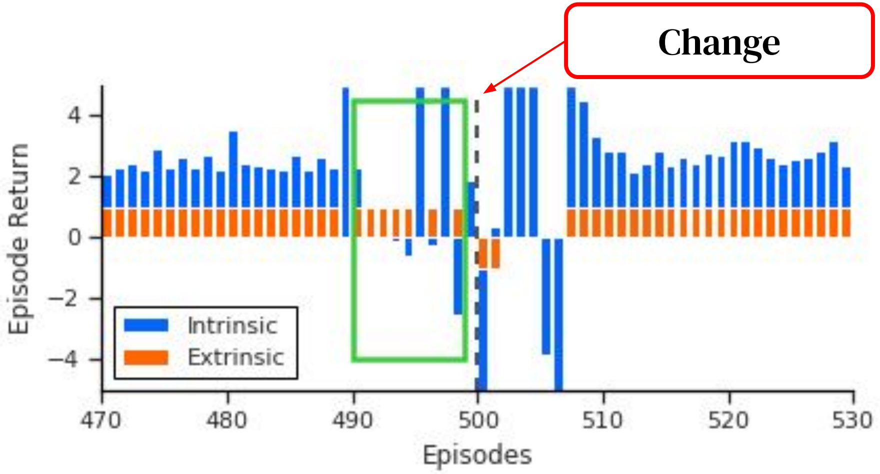
Experiment: Dealing with non-stationary tasks

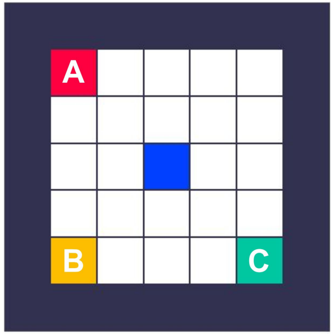
- The rewards for A and C exchange periodically within a lifetime



Experiment: Dealing with non-stationary tasks

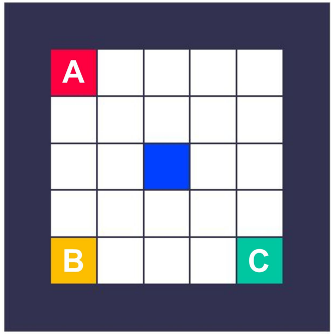
- The rewards for A and C exchange periodically within a lifetime
- The intrinsic reward starts to give negative rewards to increase entropy in **anticipation** of the change (green box).

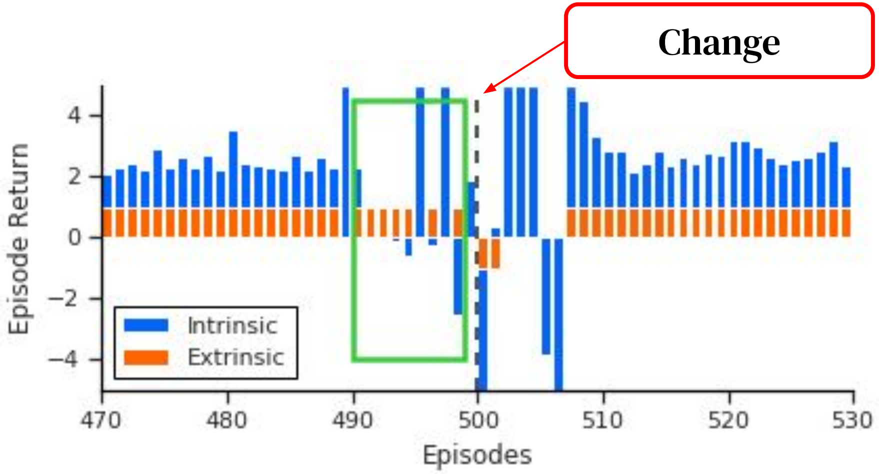


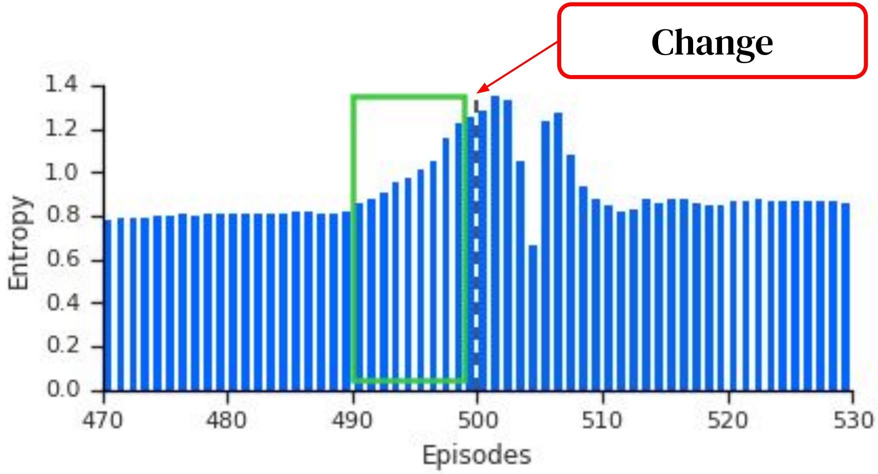


Experiment: Dealing with non-stationary tasks

- The rewards for A and C exchange periodically within a lifetime
- The intrinsic reward starts to give negative rewards to increase entropy in **anticipation** of the change (green box).
- The intrinsic reward has learned not to fully commit to the optimal behaviour in anticipation of environment changes.

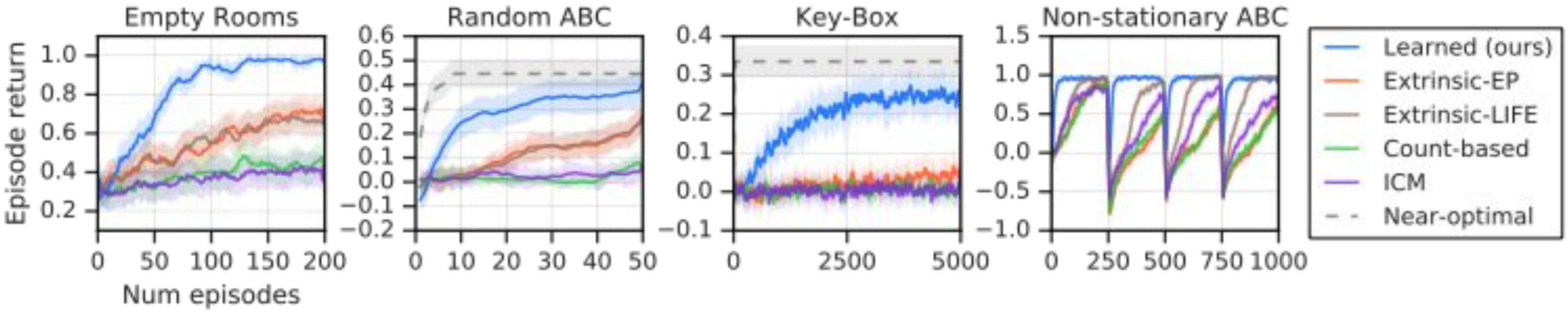






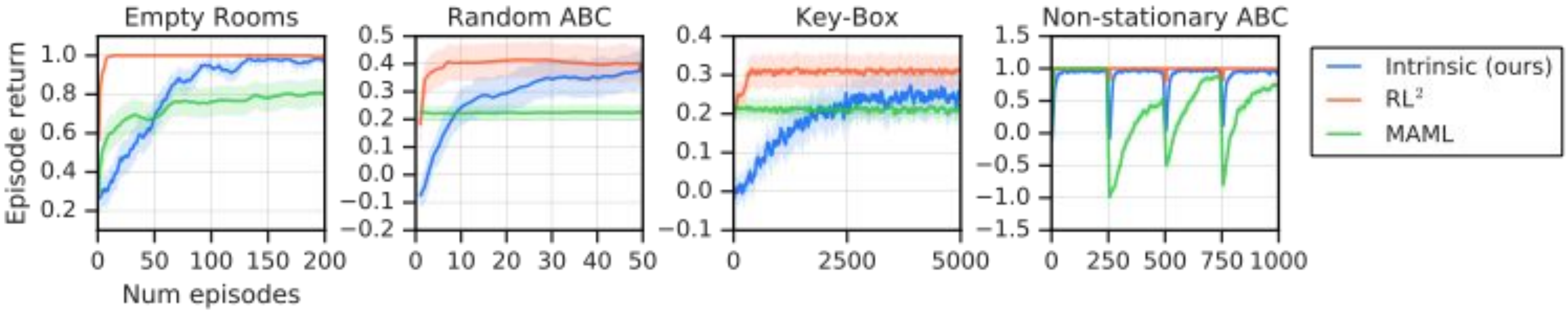
Performance (v.s. Handcrafted Intrinsic Rewards)

- Learned rewards > hand-designed rewards



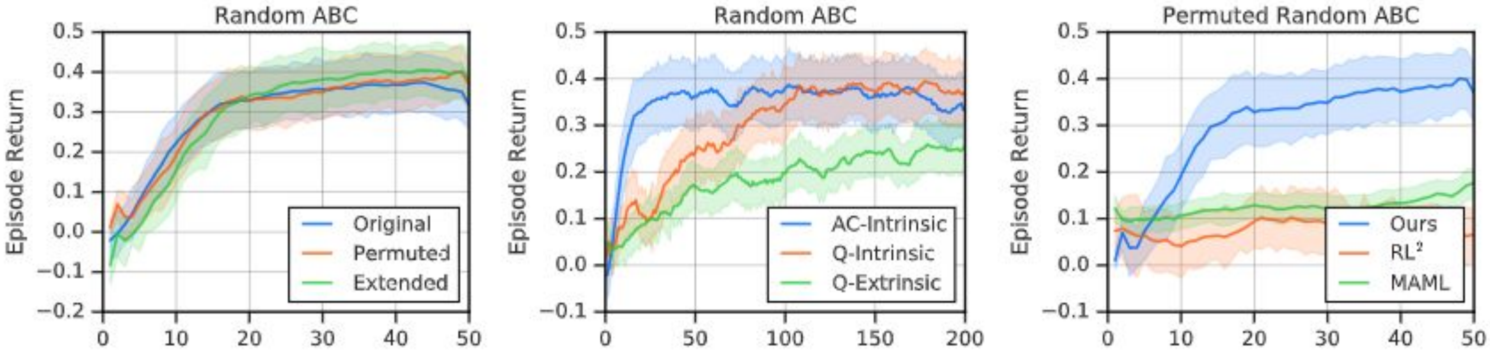
Performance (v.s. Policy Transfer Methods)

- Our method outperformed MAML and matched the final performance of RL^2
    - Our method needed to train a random policy from scratch while RL^2 started with a good initial policy



Generalisation to unseen agent-environment interfaces

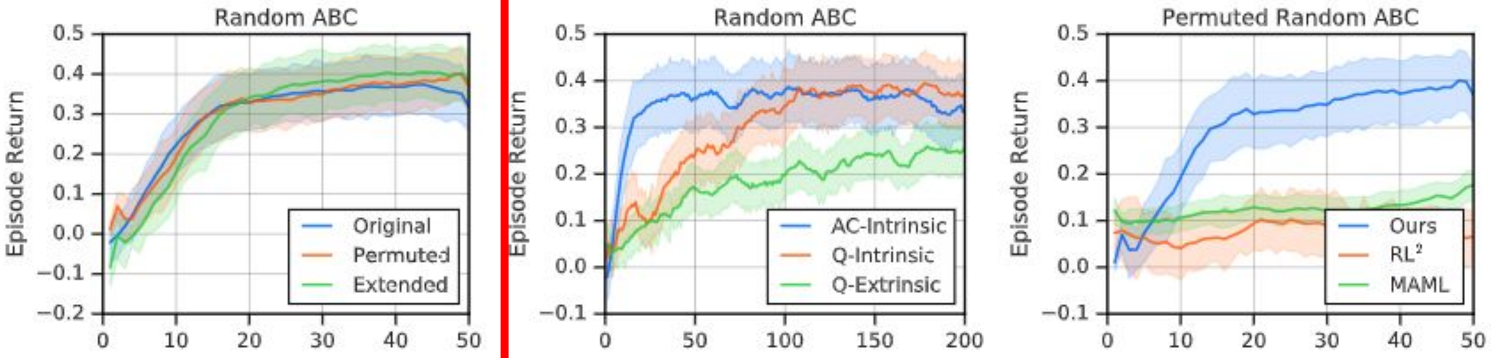
- The learned intrinsic reward could generalise to



(a) Action space (b) Algorithm (c) Comparison to baselines

Generalisation to unseen agent-environment interfaces

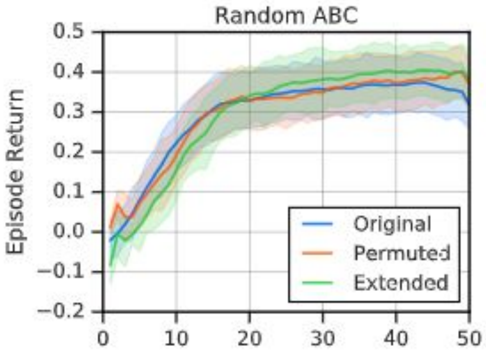
- The learned intrinsic reward could generalise to
  - Different action spaces



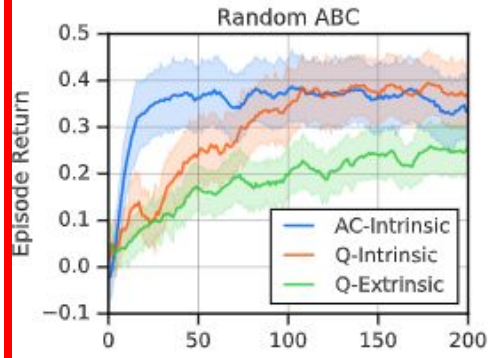
(a) Action space (b) Algorithm (c) Comparison to baselines

Generalisation to unseen agent-environment interfaces

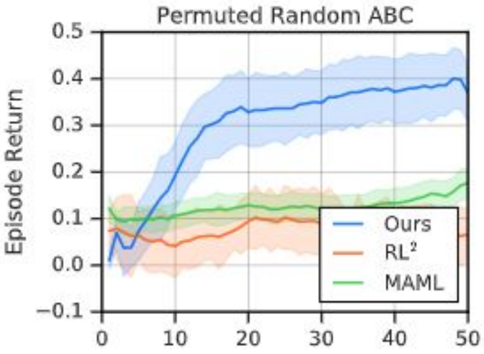
- The learned intrinsic reward could generalise to
  - Different action spaces



(a) Action space



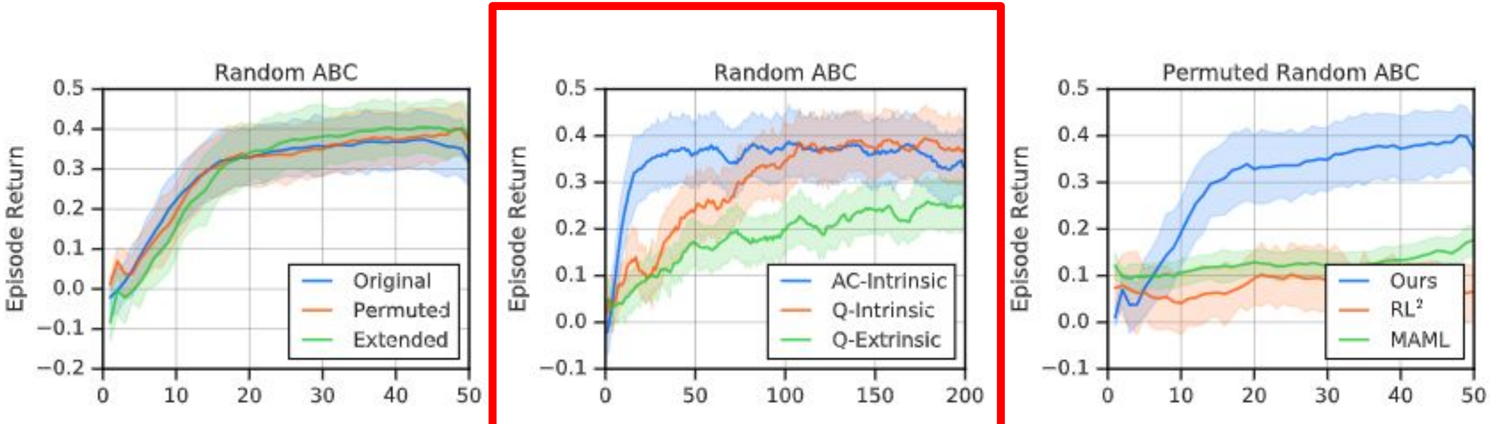
(b) Algorithm



(c) Comparison to baselines

Generalisation to unseen agent-environment interfaces

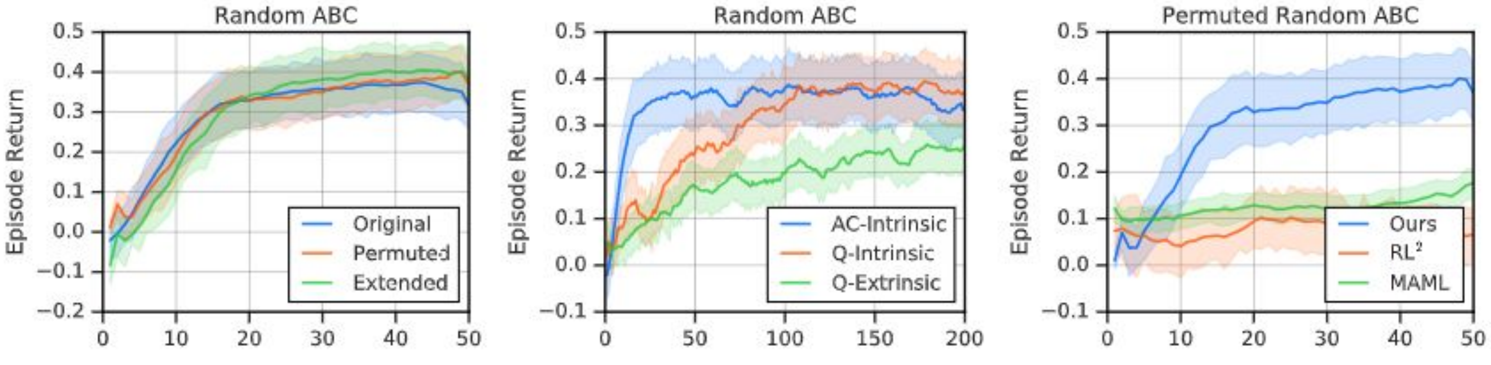
- The learned intrinsic reward could generalise to
  - Different action spaces
  - Different inner-loop RL algorithms (Q-learning)



(a) Action space (b) Algorithm (c) Comparison to baselines

Generalisation to unseen agent-environment interfaces

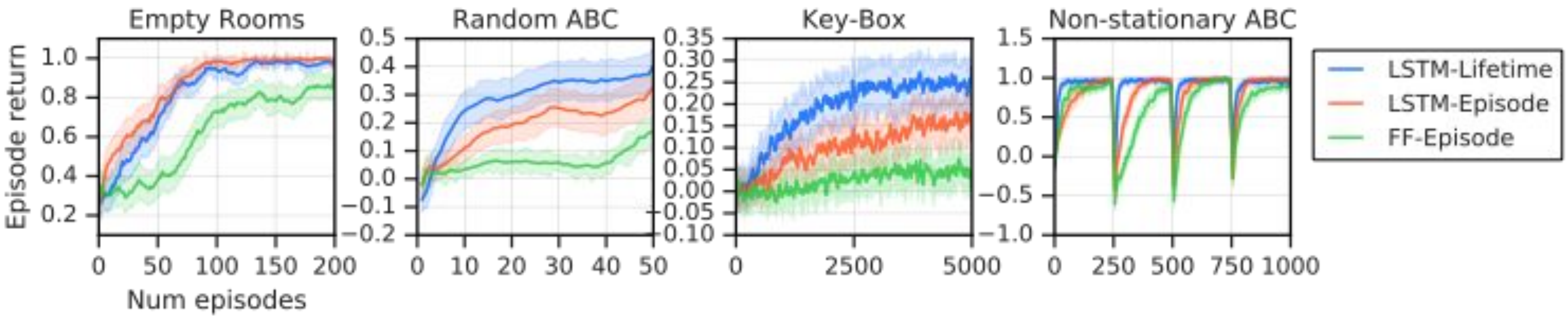
- The learned intrinsic reward could generalise to
  - Different action spaces
  - Different inner-loop RL algorithms (Q-learning)
- The intrinsic reward captures “**what to do**” instead of “**how to do**”



(a) Action space (b) Algorithm (c) Comparison to baselines

Ablation Study

- Lifetime history is crucial for exploration
- Lifetime return allows cross-episode exploration & exploitation



Takeaways / Limitations / Next steps

Takeaways

- Learned intrinsic rewards can capture
  - interesting regularities that are useful for exploration/exploitation

Takeaways / Limitations / Next steps

Takeaways

- Learned intrinsic rewards can capture
  - interesting regularities that are useful for exploration/exploitation
  - knowledge that generalises to different learning agents
  - “what to do” instead of “how to do”

Takeaways / Limitations / Next steps

Takeaways

- Learned intrinsic rewards can capture
  - interesting regularities that are useful for exploration/exploitation
  - knowledge that generalises to different learning agents
  - “what to do” instead of “how to do”

Limitations

- Empirical studies are conducted on toy domains.

Next steps

- Learning intrinsic rewards in much richer environments

Thank you!

Contact us

- Zeyu Zheng: zeyu@umich.edu
- Junhyuk Oh: junhyuk@google.com