NEURAL INFORMATION PROCESSING SYSTEMS

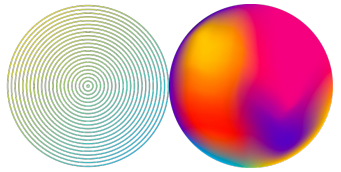
LECO: Learnable Episodic Count for Task-Specific Intrinsic Reward

Daejin Jo* Sungwoong Kim* Daniel Wontae Nam*

Taehwan Kwon Seungeun Rho Jongmin Kim Donghoon Lee

kakaobrain

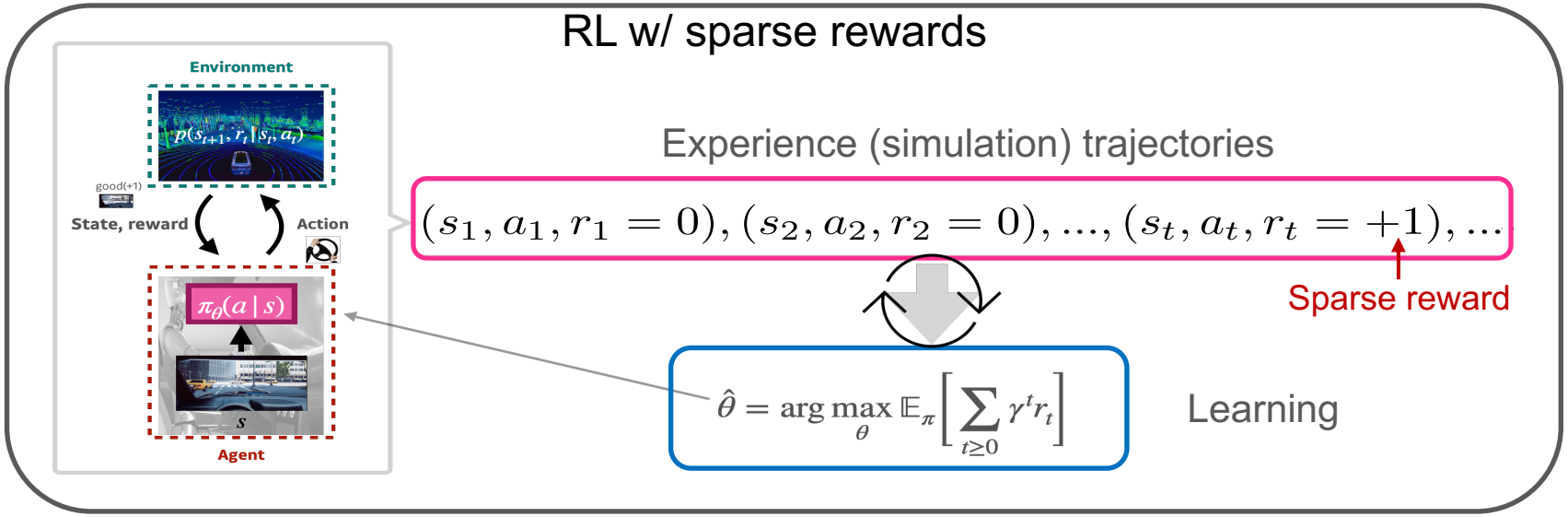
Neural Information Processing Systems, 2022



NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

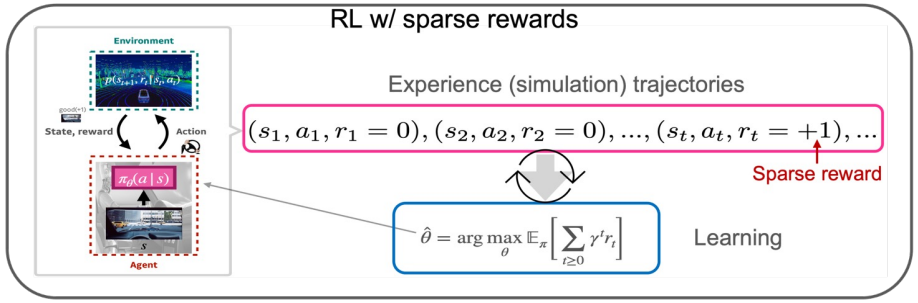
- Reinforcement learning (RL) with sparse rewards
  - Many works have proposed dense intrinsic rewards for sufficient exploration.
  - State count has been widely used as a simple yet effective intrinsic motivation to visit novel states.



NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

- Hard exploration problems
  - High-dimensional state space and a long episode time
  - Procedurally generated and partially observable environments
  - Task-irrelevant observations



Introduction

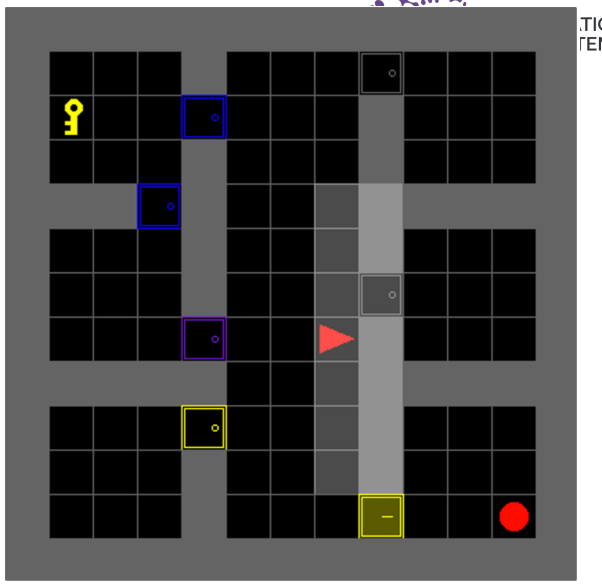
- Minigrid environment
  - 2D partially-observable, procedurally generated
  - A reward of 1 is given for success, and zero for failure
  - Actions

    turn left & right, move forward, pickup & drop

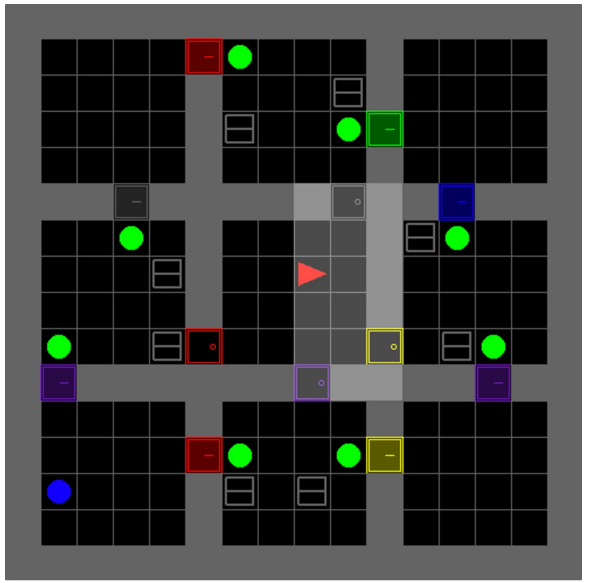
    toggle (open doors, interact with objects, …)

```
def _reward(self):
"""
    Compute the reward to be given upon success
    """

return 1 - 0.9 * (self.step_count / self.max_steps)
```



<KeyCorridor>



<ObstructedMaze>

Introduction

- Reinforcement learning (RL) with sparse rewards
  - Many works have proposed dense intrinsic rewards for sufficient exploration.
    - ex) finding key in Minigrid
  - State count has been widely used as a simple yet effective intrinsic motivation to visit novel states.

Introduction

- Reward with Intrinsic reward

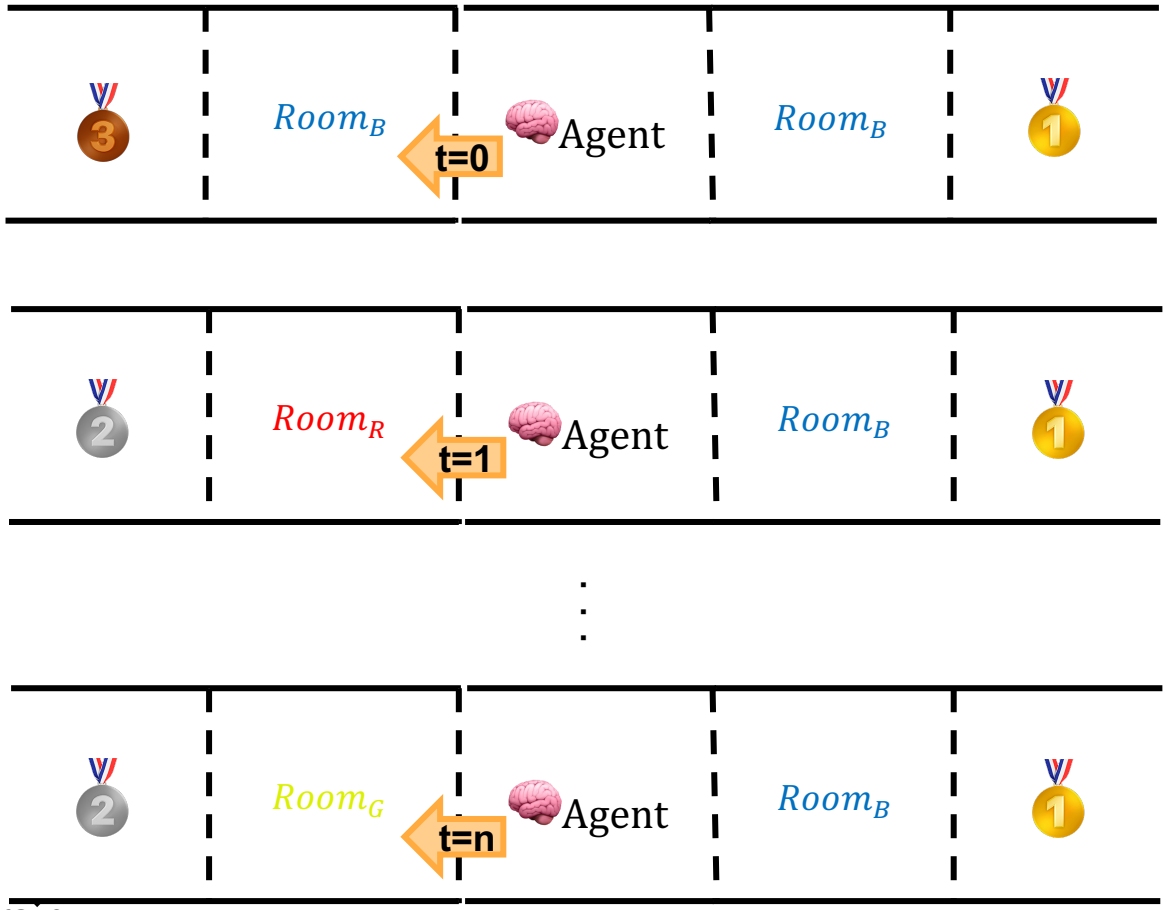
$$r_t = r_t^e + \alpha r_t^i$$

- Count-based exploration

$$r_t^i = \frac{1}{\sqrt{N_{ep}(s_t)}}$$

Introduction

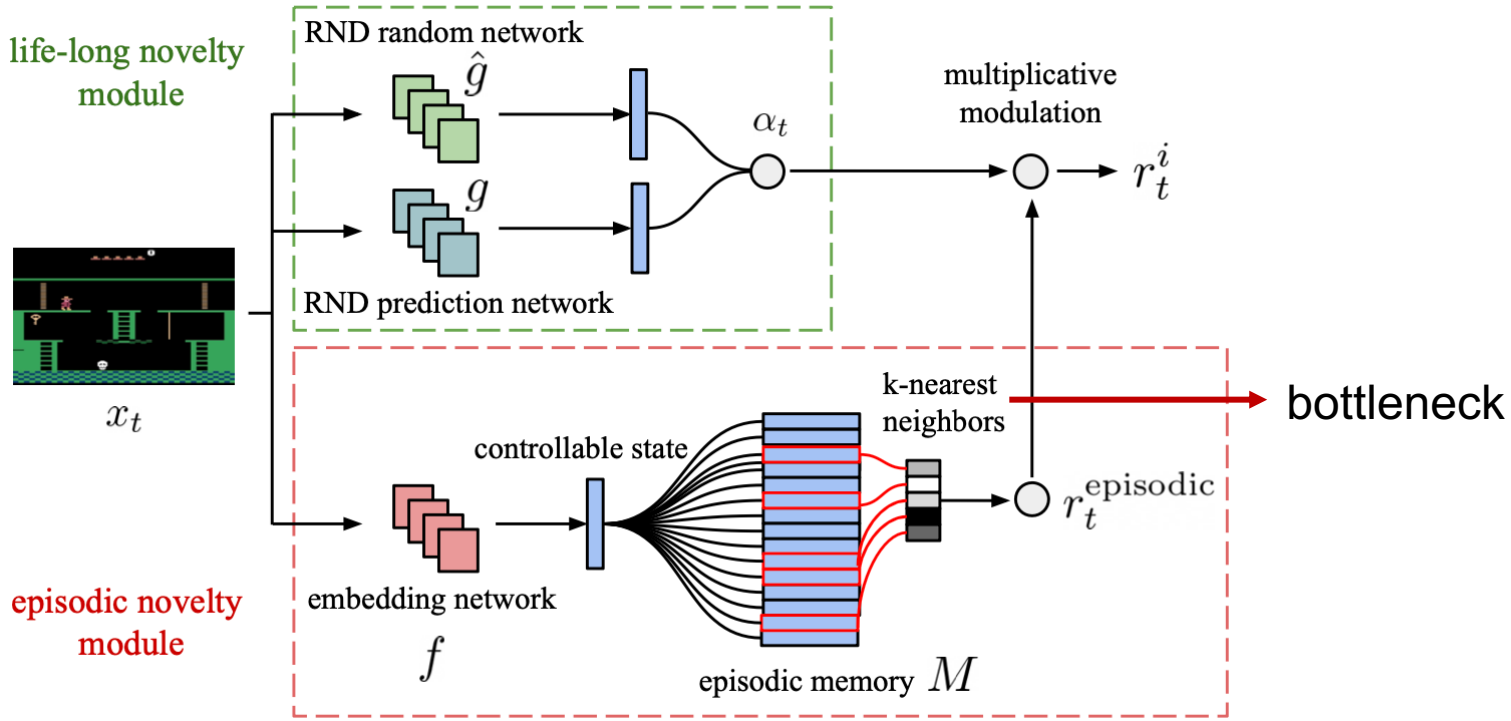
1. Inefficiency of existing count-based methods



- partial observation
- the agent is located in the middle every time you score
- the visited room color is changed every time you score

Introduction

2. Inefficiency of existing count-based methods



NGU (Badia et al., 2020) reward generator

NEURAL INFORMATION PROCESSING SYSTEMS

Learnable Episodic Count via Task-Specific Modulation

- We propose a *learnable hash-based episodic count* for a *task-specific intrinsic reward*.

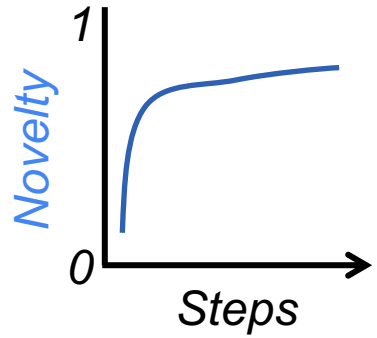
Extrinsic reward + **LECO** *intrinsic reward*

= *Episodic novelty* + *Task-specific modulation*

Learnable Episodic Count via Task-Specific Modulation

- *Episodic novelty* encourages the agent to explore novel state, it can be helpful especially in early RL phase, but would discourage the agent to visit even the task-important states after a certain period of time.

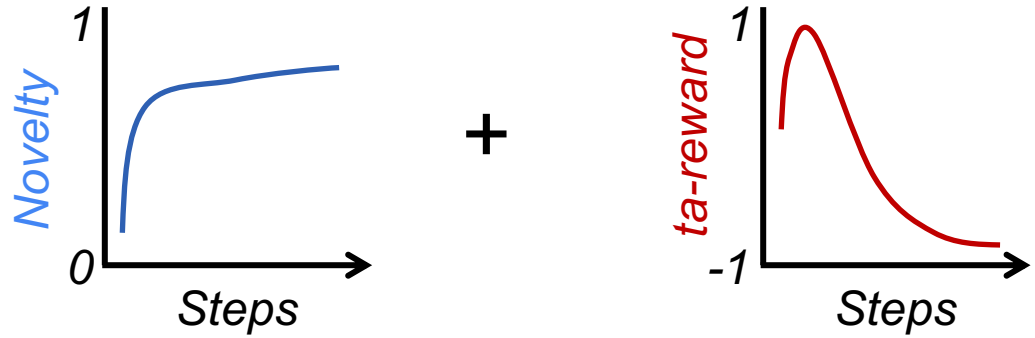
Episodic novelty + *Task-specific modulation* = **LECO** *intrinsic reward*



Learnable Episodic Count via Task-Specific Modulation

- *Task-specific modulation* learns to modulate the novelty according to whether the state is beneficial or harmful in achieving the task-specific objective.

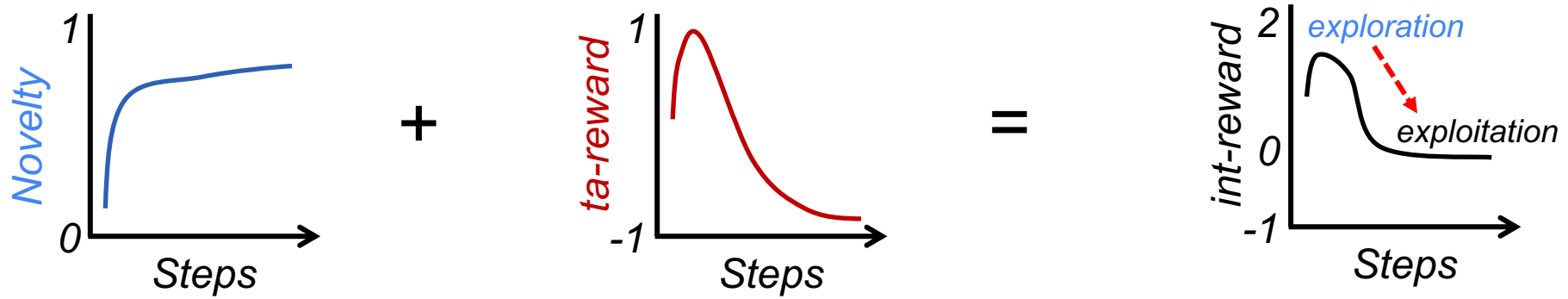
Episodic novelty + *Task-specific modulation* = **LECO** *intrinsic reward*



Learnable Episodic Count via Task-Specific Modulation

- Proposed LECO intrinsic reward specifically enables the automatic transition from exploration to exploitation during RL.

Episodic novelty + *Task-specific modulation* = **LECO** *intrinsic reward*

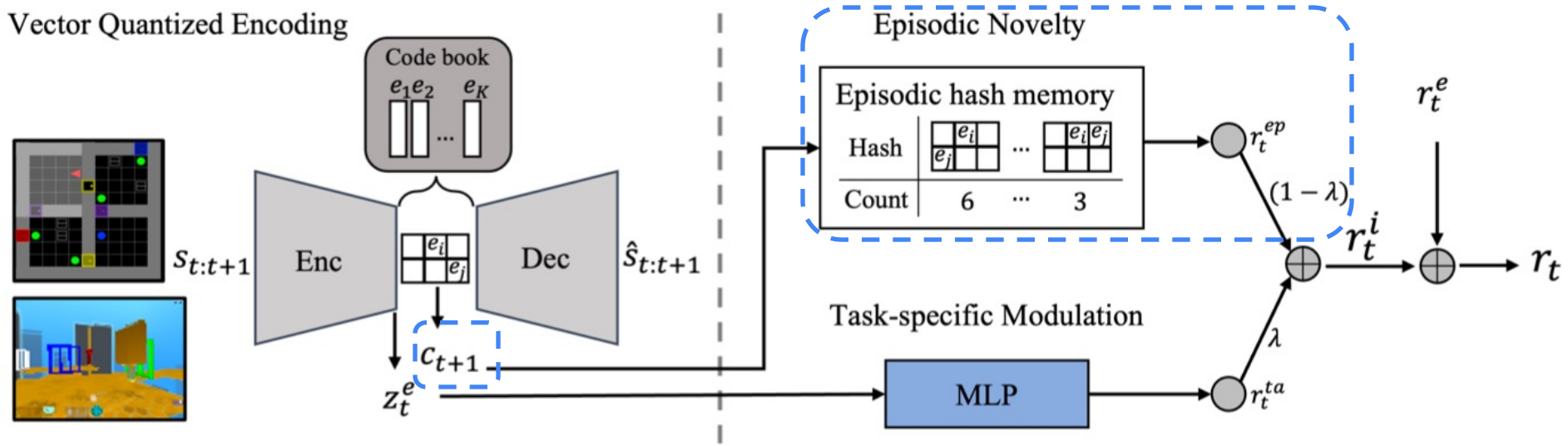


NEURAL INFORMATION PROCESSING SYSTEMS

Learnable Episodic Count via Task-Specific Modulation

- *Episodic novelty* is defined as inverse of episodic count by *vector-quantized hashing* which enables counting the occurrences in a constant time.

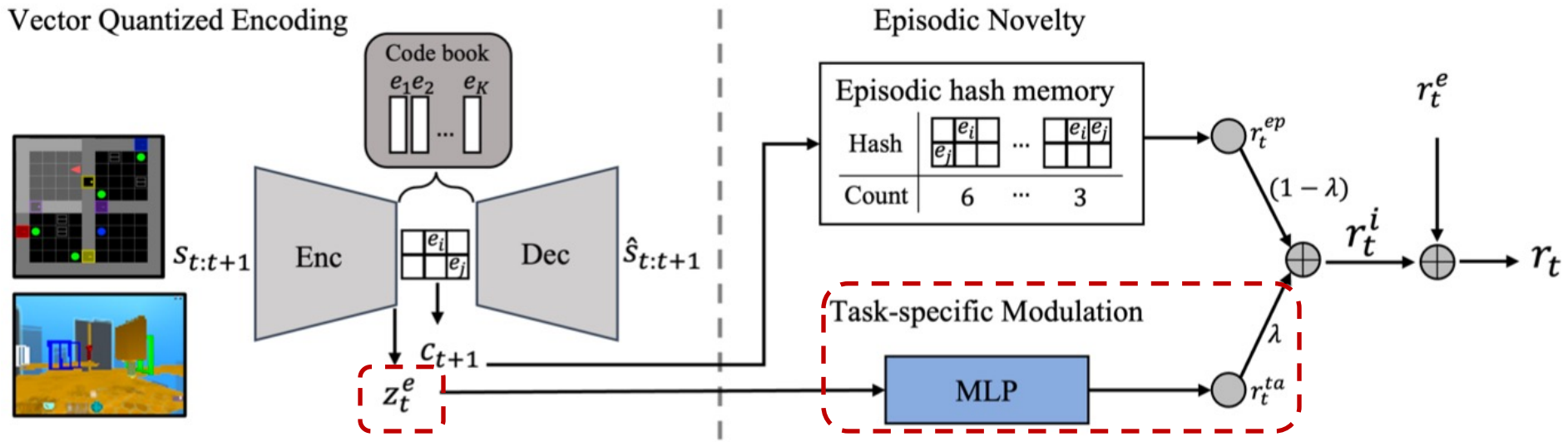
$$r_t^i = (1-\lambda)\frac{1}{\sqrt{N_{\text{ep}}(s_{t+1})}} + \lambda r_t^{\text{ta}}(a_{t-1}, s_t, a_t)$$



Learnable Episodic Count via Task-Specific Modulation

- *Task-specific modulation* is a dynamic factor for controlling exploration, which is trained to maximize only the extrinsic rewards by the *bi-level optimization*.

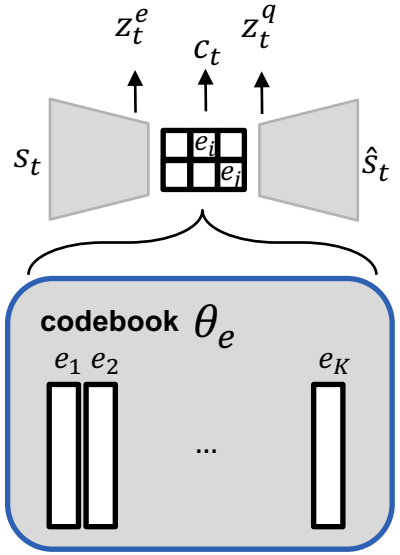
$$r_t^i = (1-\lambda)\frac{1}{\sqrt{N_{\text{ep}}(s_{t+1})}} + \lambda r_t^{\text{ta}}(a_{t-1}, s_t, a_t)$$



$$r_t^{\text{ta}}(a_{t-1}, s_t, a_t; \theta_{\text{ta}}) = \tanh\left(\mathbb{1}_{a_{t-1}}{}^{\top} f_{\text{ta}}(z_t^e, a_t; \theta_{\text{ta}})\right)$$

Learnable Episodic Count via Task-Specific Modulation

- *Vector-Quantized Hashing* [1]



$$c_t = [k_i]_{i=1}^{w \times h}, k_i = argmin_k \|z_t^e - e_k\|_2$$

$$z_t^q(i) = argmin_{e_k \in \theta_e} \|z_t^e(i) - e_k\|_2$$

$$\nabla_{\theta_{\text{vq}}} \mathcal{J}^{\text{vq}} = -\sum_{t=0}^{T} \nabla \left(\log p(s_t | z_t^q) + \|\mathbf{sg}[z_t^e] - z_t^q\|_2^2 + \|z_t^e - \mathbf{sg}[z_t^q]\|_2^2 \right)$$

* sg: stop gradient

[1] Neural Discrete Representation Learning, Oord et al., 2018

Learnable Episodic Count via Task-Specific Modulation

- *Bi-level Optimization (LIRPG)* [2]

$$\theta' = \theta + \eta \nabla_\theta \mathcal{J}^{e+i} = \theta + \eta \nabla_\theta \left(\mathbb{E}_{\pi_\theta} [\sum_{t=0}^{\infty} \gamma^t (r_t^e + \alpha r_t^i)] \right)$$

→ update policy

$$\theta'_{\text{ta}} = \theta_{\text{ta}} + \eta_{\text{ta}} \nabla_{\theta_{\text{ta}}} \mathcal{J}^e = \theta_{\text{ta}} + \eta_{\text{ta}} \nabla_{\theta_{\text{ta}}} \left(\mathbb{E}_{\pi_{\theta'}} [\sum_{t=0}^{\infty} \gamma^t r_t^e] \right)$$

→ update task-specific modulator

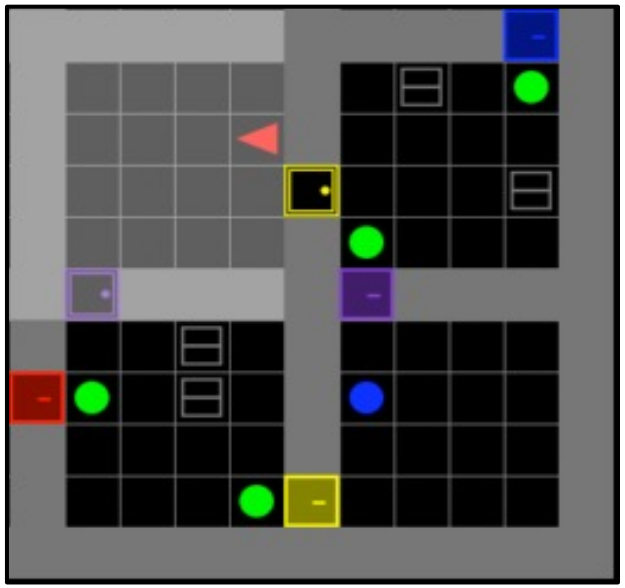
* η: learning rate

[2] On learning intrinsic rewards for policy gradient methods, Zheng et al., 2018

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments: Benchmarks – heavy exploration tasks

MiniGrid Environment



- MultiRoom-N6
- KeyCorridor-S4R3
- KeyCorridor-S6R3
- ObstructedMaze-1Q
- ObstructedMaze-2Q
- ObstructedMaze-Full

Dmlab Environment

- lasertag_three_opponents_small
- lasertag_three_opponents_large

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments

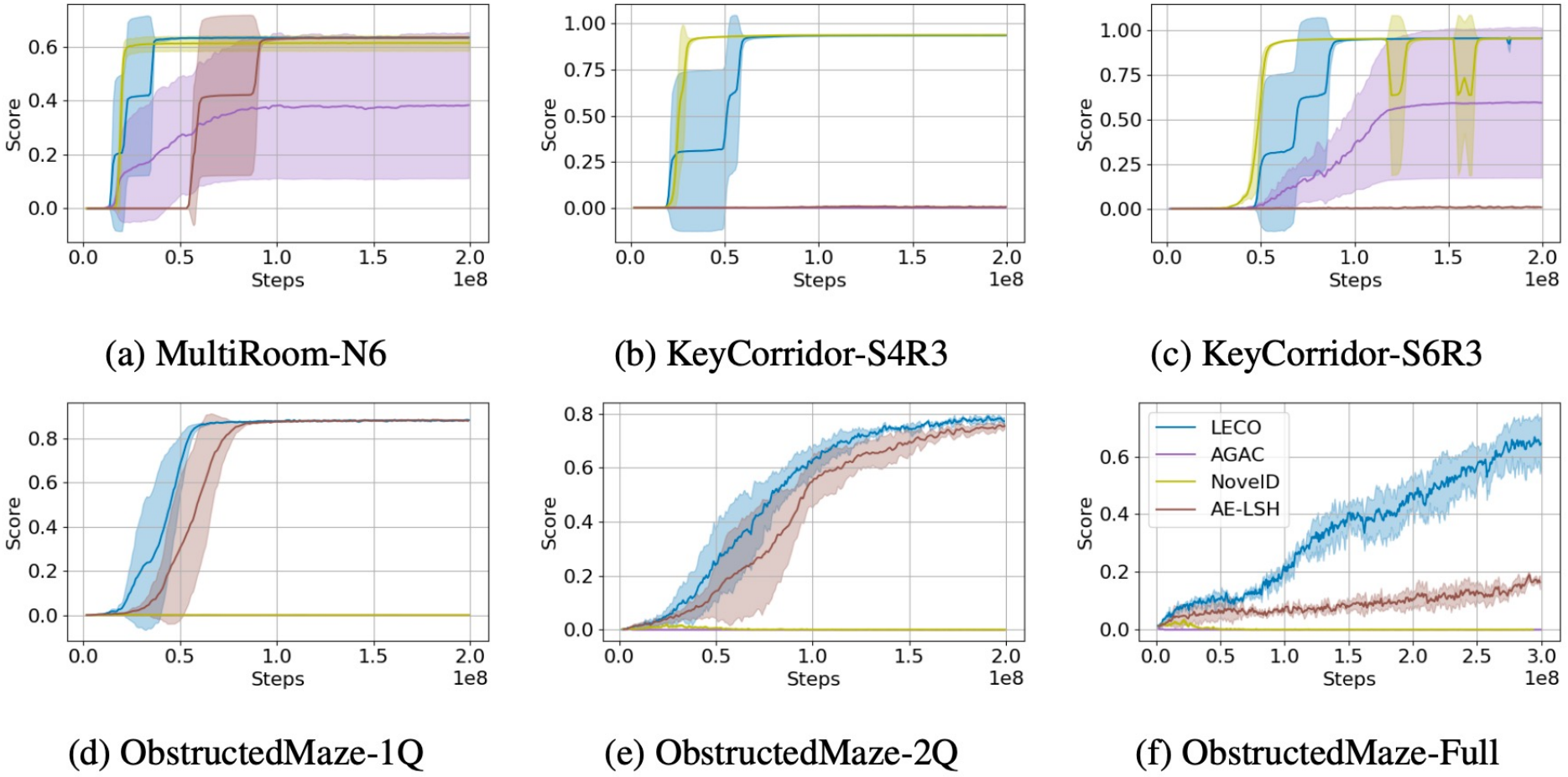
- Baseline
  - IMPALA without intrinsic reward: *no-int*
  - episodic count based models: *DSC, AE-LSH, VQ-only*
  - SOTA models: *NovelD, AGAC, RIDE*
  - LECO *variations: LECO, LECO(AE-LSH)*

- The same episodic count module (VQ) is employed in VQ-only, *NovelD, AGAC, RIDE, and LECO.*

- Minigrid setup
  - We emphasize that for all tasks the baselines and LECO only use the original partial input observations, in which a 7x7x3 tensor given as the partially-observable grid cell is the input observation for both the policy network and intrinsic module as in [3].

[3] LEARNING WITH AMIGO: ADVERSARIALLY MOTIVATED INTRINSIC GOALS, Campero et al., 2021

Experiments: Performance comparison - Minigrid



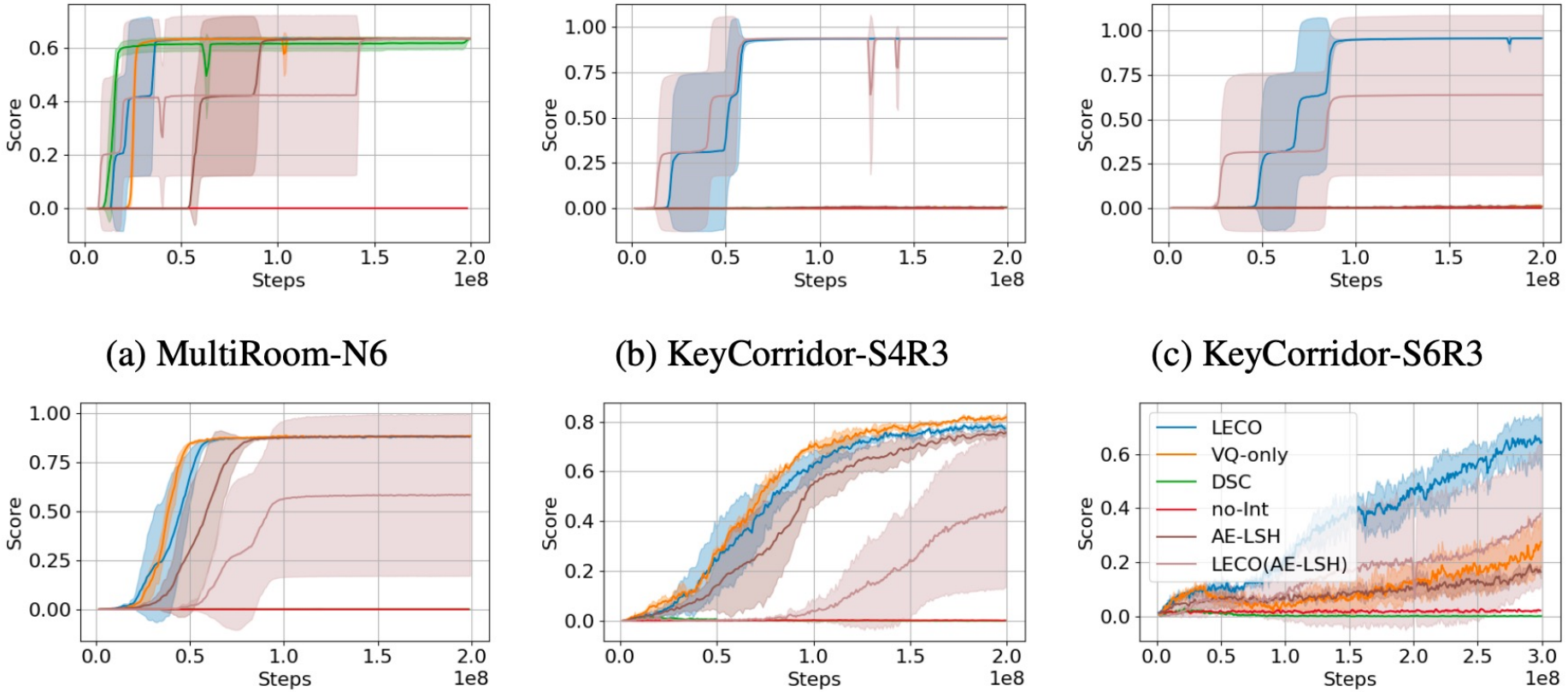
(a) MultiRoom-N6 (b) KeyCorridor-S4R3 (c) KeyCorridor-S6R3

(d) ObstructedMaze-1Q (e) ObstructedMaze-2Q (f) ObstructedMaze-Full

Figure 4: Comparison of LECO to baselines on six selected MiniGrid tasks. LECO shows the most promising performance compared to related existing methods in the field. The shaded area represents a range of standard deviation over 3 runs with different random seeds.

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments: Performance comparison - Minigrid

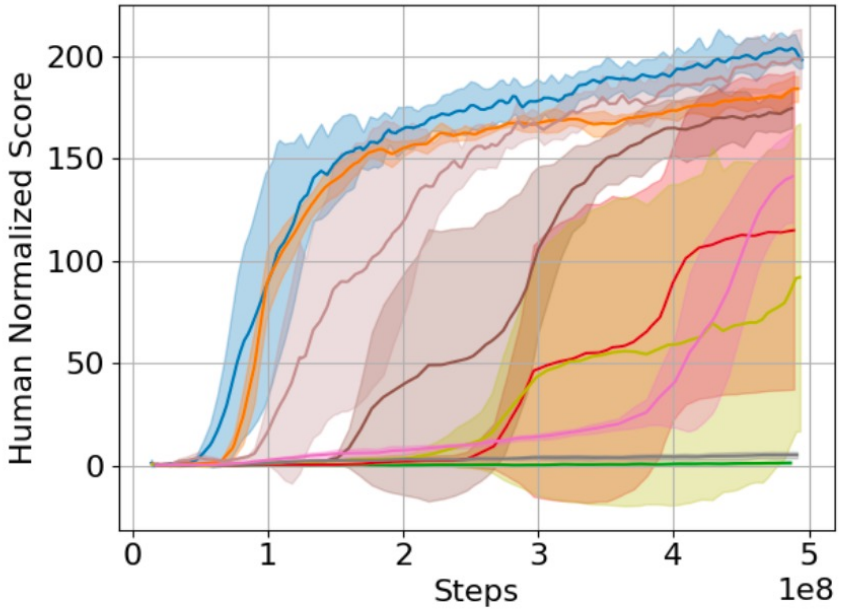


(a) MultiRoom-N6 (b) KeyCorridor-S4R3 (c) KeyCorridor-S6R3

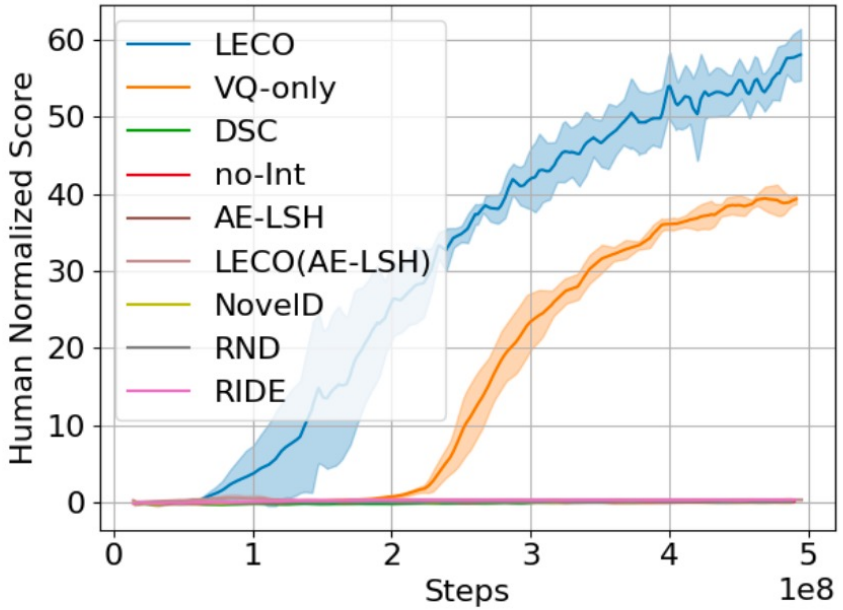
(d) ObstructedMaze-1Q (e) ObstructedMaze-2Q (f) ObstructedMaze-Full

Figure 3: Comparison of LECO and its variations on six selected MiniGrid tasks. The final choice of design (LECO) shows superior performance compared to its variants. The shaded area represents a range of standard deviation over 3 runs with different random seeds.

Experiments: Performance comparison - DMLab



(a) lasertag_three_opponents_small



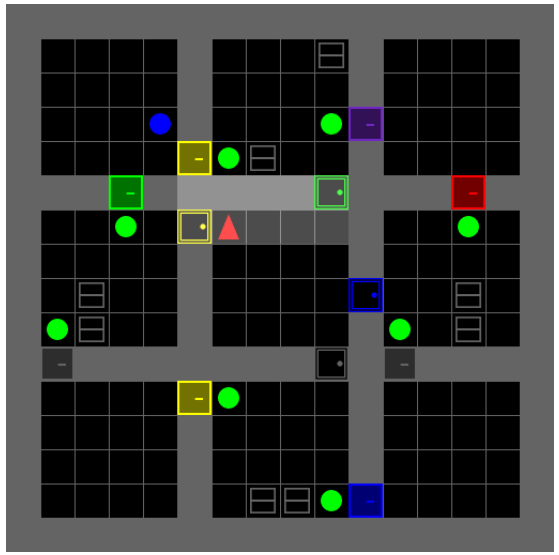
(b) lasertag_three_opponents_large

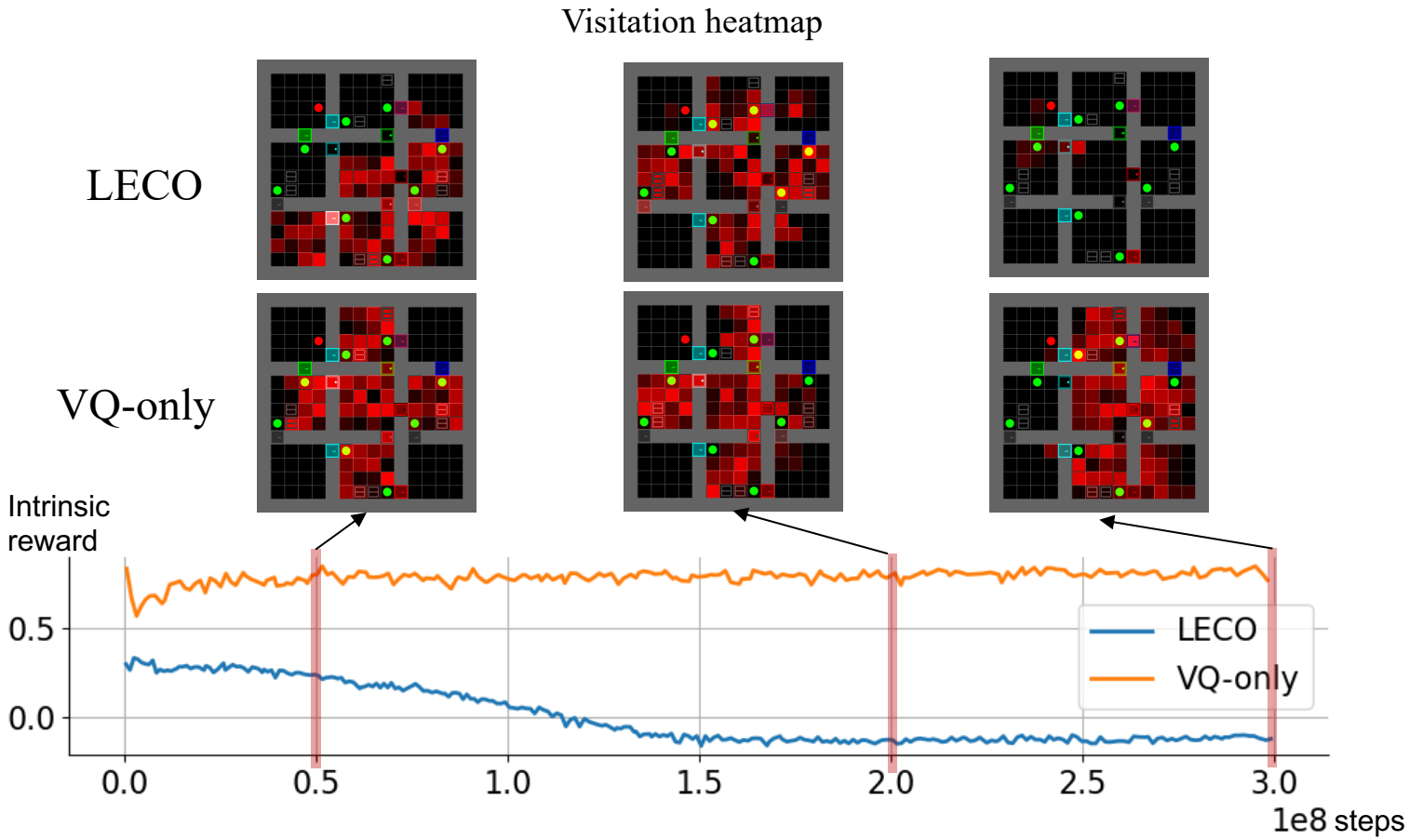
Figure 5: Learning curves on two selected DMLab tasks. Compared to all of its variants and baselines LECO shows the highest performance. Especially in `lasertag_three_opponents_large`, only LECO and VQ-only is able to solve the task. The y-axis represents the human normalized score. The shaded area represents the range of a standard deviation over 3 runs of random seeds.

NEURAL INFORMATION PROCESSING SYSTEMS

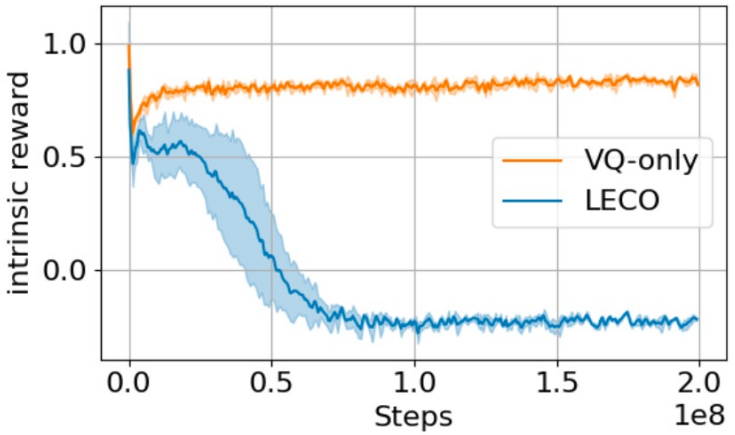
Experiments: Intrinsic modulation via extrinsic motivation

ObstructedMaze-Full

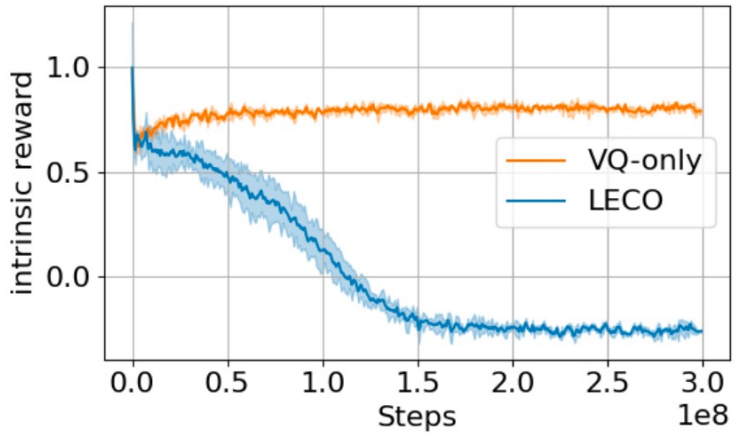




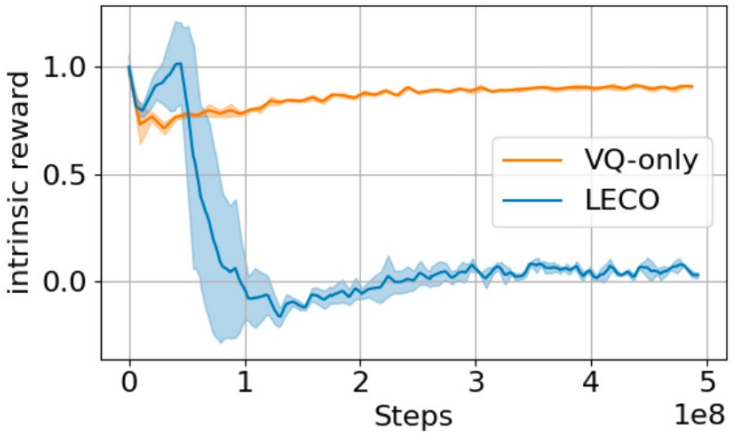
Experiments: Intrinsic modulation via extrinsic motivation



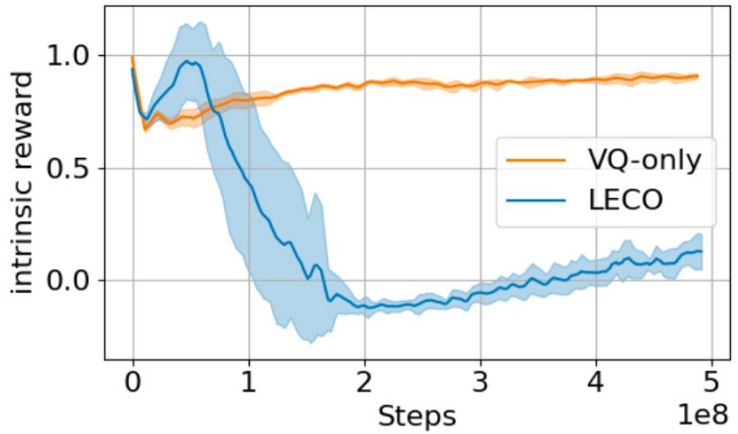
(e) MiniGrid-ObstructedMaze-2Q



(f) MiniGrid-ObstructedMaze-Full



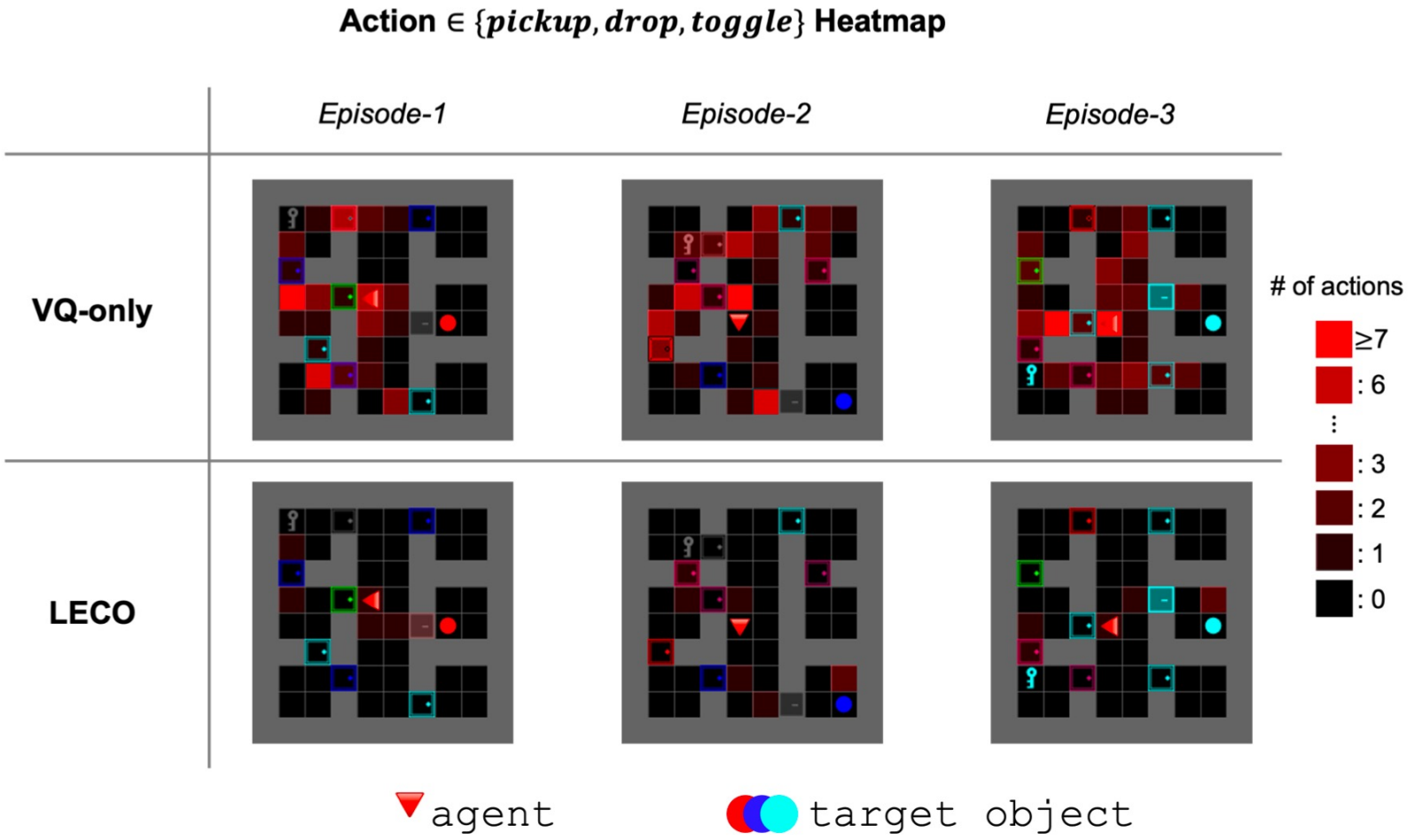
(g) DMLab-lasertag_three_opponents_small



(h) DMLab-lasertag_three_opponents_large

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments: Intrinsic modulation via extrinsic motivation



Experiments: Hashing Results

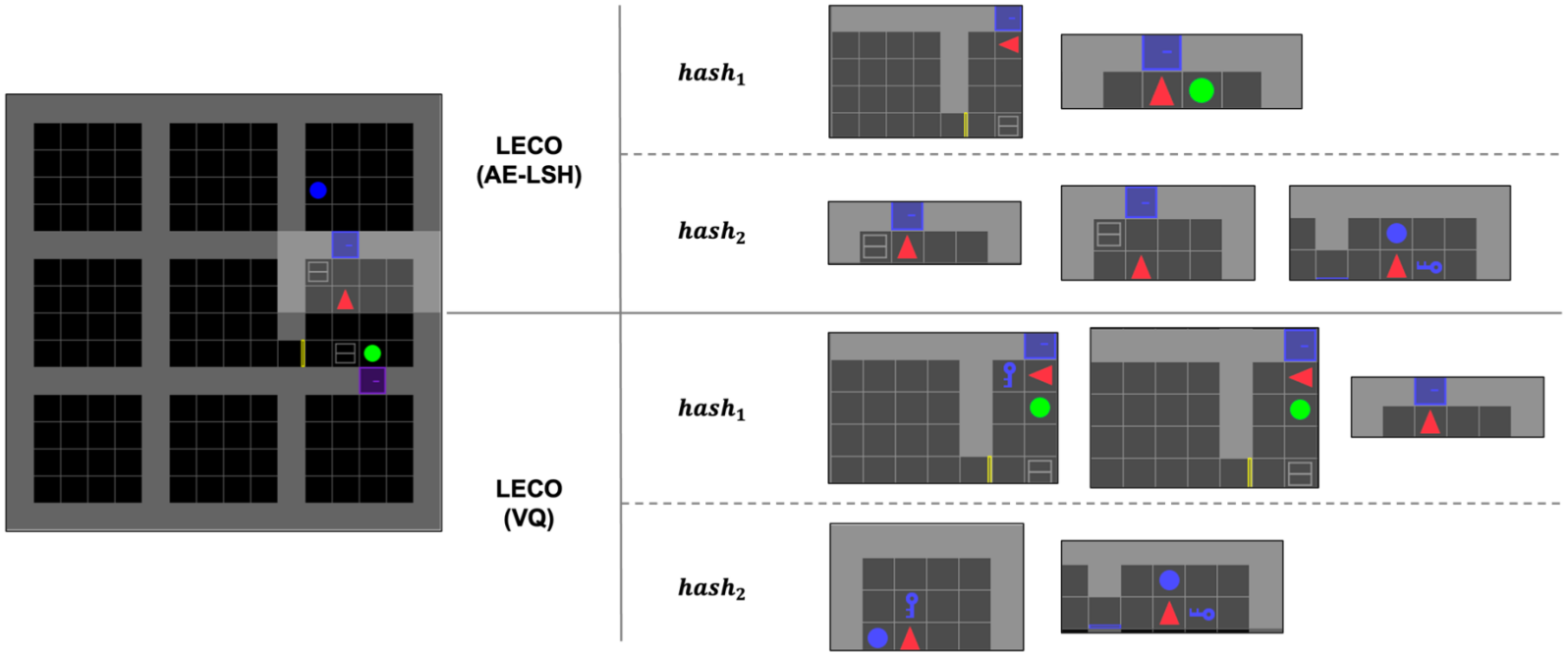
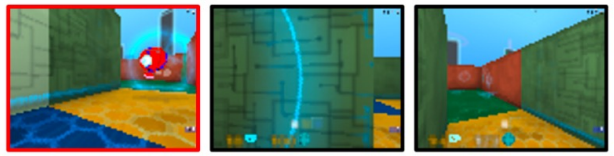


Figure 14: (**Left:**) An episode of `ObstructedMaze-1Q` Environment. The bright area is the partial observation. In this episode, the agent must find a blue key, open the locked blue door, and obtain the blue ball. (**Right:**) Hash samples from the episodic hash memory of LECOs. Each row represents a hash index and the partially observed states that are mapped into this hash.

NEURAL INFORMATION PROCESSING SYSTEMS

Experiments: Hashing Results

LECO (AE-LSH)	$hash_1$	
	$hash_2$	
	$hash_3$	
LECO (VQ)	$hash_1$	
	$hash_2$	
	$hash_3$	

NEURAL INFORMATION PROCESSING SYSTEMS

Conclusion

- We proposed a learnable episodic count to produce task-specific intrinsic rewards for hard exploration problems including high-dimensional state spaces and procedurally generated environments.

- Experimental results demonstrate that the proposed task-specific exploration based on the episodic count significantly outperforms the previous SOTA exploration methods.

- For future works, we will extend our work to combine the inter-episode novelty and devise a more efficient training of task-specific intrinsic reward to further improve the scalability.

Kakaobrain