CIS centre for intelligent sensing

Queen Mary University of London

Learning Action Representations for Self-supervised Visual Exploration

Changjae Oh and Andrea Cavallaro

{c.oh,a.cavallaro}@qmul.ac.uk

1. Introduction

Objective: To exploit deep reinforcement learning (RL) for navigation

- To maximise the expected sum of rewards r_t during T steps by optimising θ_P in a parameterised policy, $\pi(s_t;\theta_P)$, generating the action a_t from the state s_t:

$$\max_{\theta_P} E_{\pi(s_t;\theta_P)}\left[\sum_{t=0}^{T} r_t\right]$$

Challenge

- The goal is far from the initial state: sparse extrinsic rewards
- Hard to train the policy $\pi(s_t;\theta_P)$ which determines action a_t

2. Related Work

Asynchronous Actor-Critic Agents (A3C) [1]

- RL approach handling multiple agents in training
- Hard to train the model with sparse extrinsic rewards

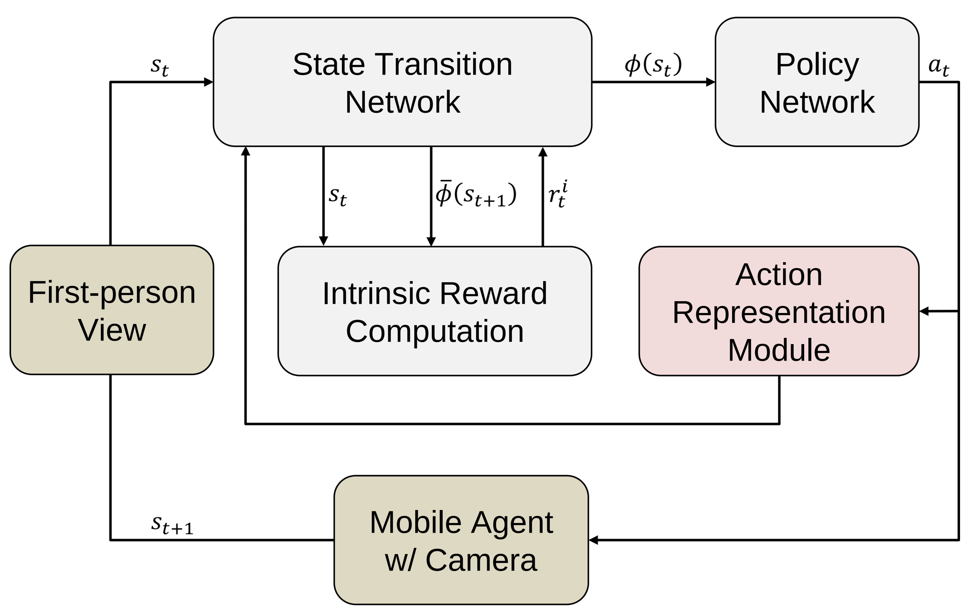
Curiosity-driven Exploration (ICM) [2,3]

- Intrinsic rewards to encourage an agent to explore unseen regions
- Using the prediction error-based loss function as intrinsic reward
- Handling various actions by one-hot encoding scheme
- Less capability to discriminate the predictions from different input actions

3. Proposed Approach: Action Representation for Exploration (AR4E)

Overview

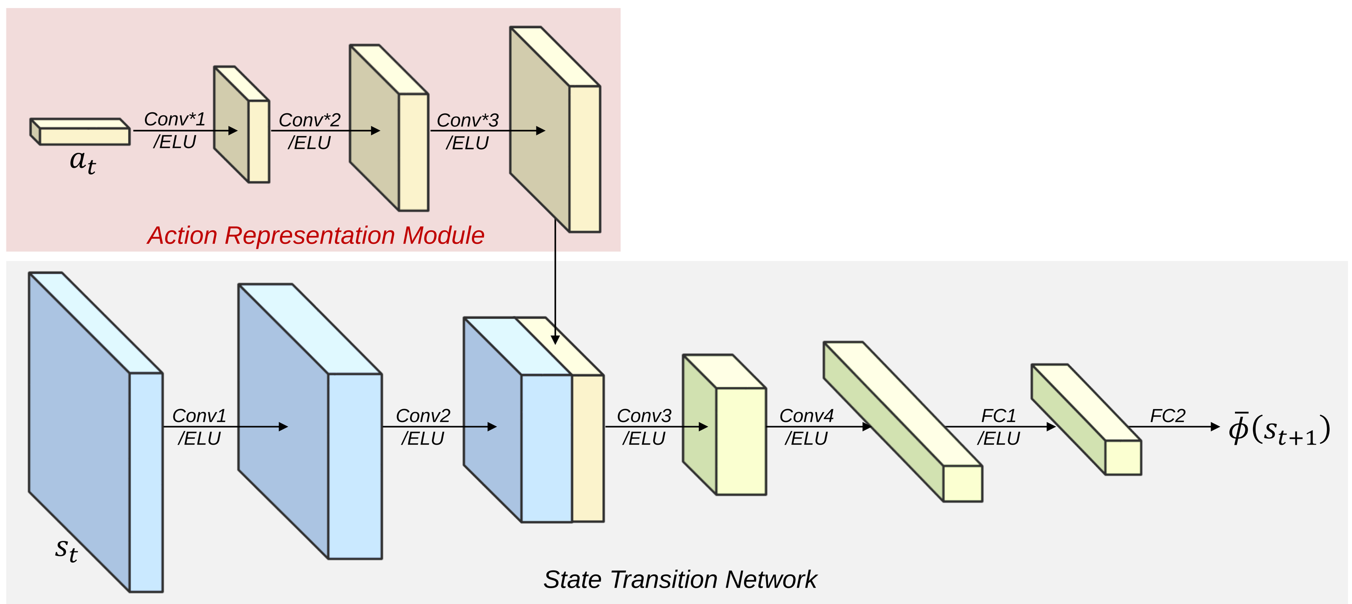
- A self-supervised prediction network that predicts the (future) state from a state-action pair
- An action representation module that boosts the representation power
- A joint regression and triplet ranking loss for learning features effectively



a_t	Action in step t
s_t	State in step t
$\phi(s_t)$	Representation of s_t
$\bar{\phi}(s_{t+1})$	Prediction from (s_t, a_t)
r_t^i	Intrinsic reward in step t

Explicit Modelling of the Action Representation

- Decoding one-hot codes of input actions to high-dimensional representations
- Generating more expressive features during training



Forward Model

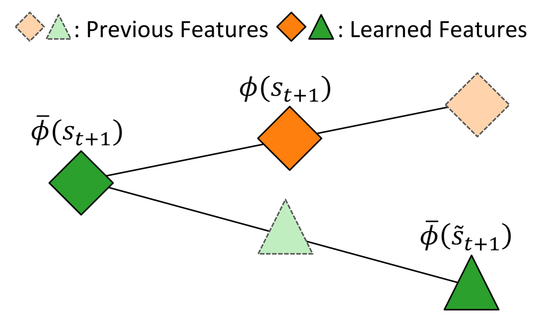
Forward Model: Learning to encode information relates to the performing task

1) Regression loss: learning to predict a future state from the current state and action

$$L_{F_1} = \left\|\bar{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$$

2) Triplet ranking loss [4]: discriminating $\bar{\phi}(s_{t+1})$ from a prediction with a different action $\tilde{a}_t$, $\bar{\phi}(\tilde{s}_{t+1})$:

$$L_{F_2} = \max(0, m + \left\|\bar{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2 - \left\|\bar{\phi}(s_{t+1}) - \bar{\phi}(\tilde{s}_{t+1})\right\|_2^2)$$



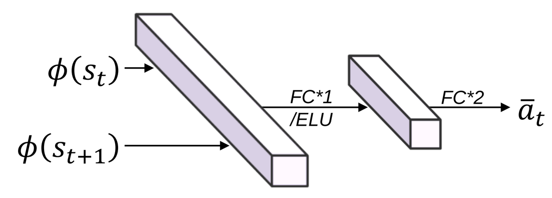
$\phi(s_{t+1})$	Representation of s_{t+1}
$\bar{\phi}(s_{t+1})$	Prediction from (s_t, a_t)
$\bar{\phi}(\tilde{s}_{t+1})$	Prediction from $(s_t, \tilde{a}_t)$
m	Margin
γ	Parameter for L_{F_2}

3) Joint Regression and Triplet loss functions:

$$L_F = L_{F_1} + \gamma L_{F_2}$$

Inverse Model: Learning to recognise an actual action a_t from states s_t and s_{t+1}

$L_I(a_t, \bar{a}_t)$: Softmax classification between a_t and predicted $\bar{a}_t$



Inverse Model

Intrinsic Rewards: Prediction error-based rewards with a scaling factor η:

$$r_t^i = \eta\left\|\bar{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$$

Policy Network: Generating the action a_t from the state s_t

$\pi(s_t;\theta_P)$: LSTM network to encode temporal information

Final Loss: Learning to maximise the extrinsic and intrinsic rewards, r_t^e and r_t^i, and minimise the losses for the forward and inverse models:

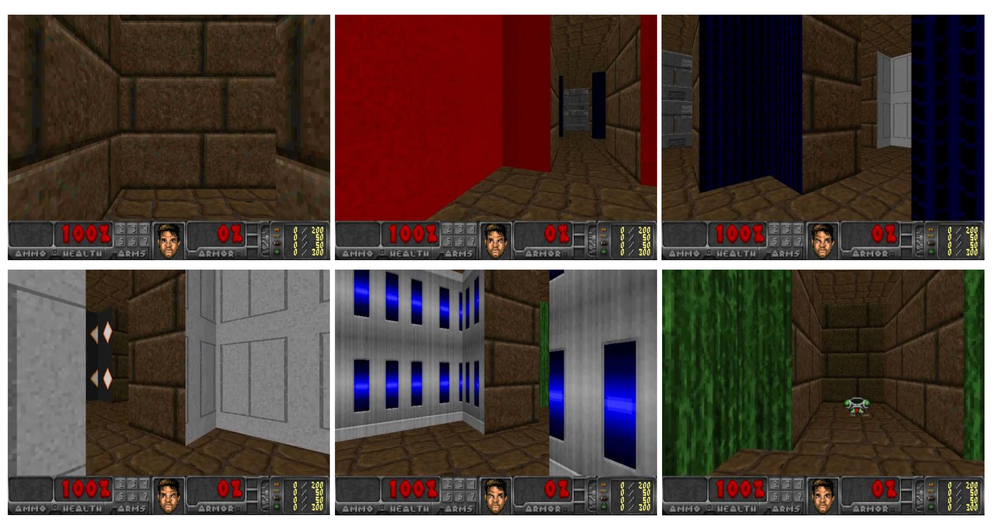
$$\min_{\theta_P,\theta_F,\theta_I} -\lambda E_{\pi(s_t;\theta_P)}\left[\sum_t r_t^e + r_t^i\right] + L_F + L_I$$

$\theta_P, \theta_F, \theta_I$	Network parameters for the policy, forward, and inverse model

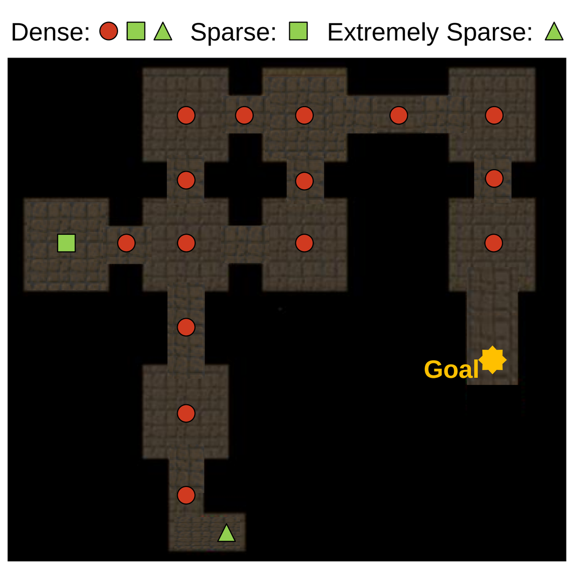
4. Experiments

Setup

- Approaches: A3C [1], ICM [2], and AR4E (Proposed)
- Action space: move forward, turn left, turn right, no action
- Environment: *VizDoom MyWayHome* [5]
- Reward Setting: Dense (random spawning in different locations), Sparse (270 steps), Extremely Sparse (350 steps)
- Total training steps: 20M steps
- RL method: A3C [1] with 16 agents



VizDoom Navigation



Spawning locations of the agent

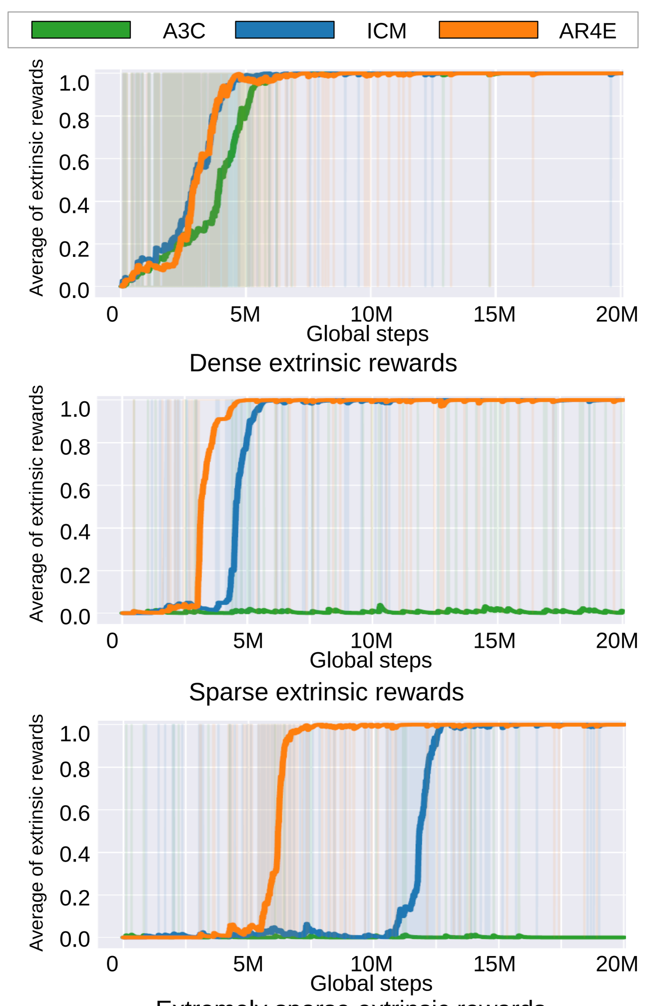
Results

Fine-tuning pre-trained models with sparse setting

Pre-training total steps	Success rate (%)
0 (from scratch)	94.45 ± 22.87
0.5M	91.89 ± 27.28
2M	92.01 ± 26.08
10M	**96.32 ± 18.89**

Success rates with different loss functions

Model		Success rate (%)		
Forward	Inverse	Dense	Sparse	Extremely Sparse
L_F	-	7.87 ± 26.93	0.01 ± 1.12	0.06 ± 2.50
-	L_I	9.12 ± 28.79	0.44 ± 6.60	0.09 ± 2.96
L_{F_1}	L_I	**96.78 ± 17.63**	91.33 ± 28.13	87.10 ± 33.51
L_{F_2}	L_I	8.13 ± 27.33	0.006 ± 0.00	11.25 ± 3.35
L_F	L_I	96.06 ± 19.43	**94.45 ± 22.87**	**87.13 ± 33.48**



Dense extrinsic rewards

Sparse extrinsic rewards

Extremely sparse extrinsic rewards

5. Conclusion

- Learning features by explicit modelling of action representations and with the joint regression and triplet ranking loss functions for efficient exploration
- Faster RL training convergence than A3C [1] or ICM [2] (with +0.5% parameters) as the sparsity of the extrinsic rewards increases
- Handling the repetitive movement during navigation as a future work

Acknowledgment

This work was supported by the EPSRC Project NCNR (EP/R02572X/1).

References

[1] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of Int. Conf. Mach. Learn. (ICML)*, 2016.

[2] D. Pathak, A. E. P. Agrawal, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. of Int. Conf. Mach. Learn. (ICML)*, 2017.

[3] O. Zhelo, J. Zhang, L. Tai, M. Liu, and W. Burgard, "Curiosity-driven exploration for mapless navigation with deep reinforcement learning," in *Proc. of ICRA Workshop on Machine Learning in Planning and Control of Robot Motion*, 2018.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. of IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[5] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski,"Vizdoom: A doom-based AI research platform for visual reinforcement learning," in *Proc. of IEEE Conf. Comput. Intell. Games (CIG)*, 2016.