EMI: Exploration with Mutual Information
Supplementary Material

Hyoungseok Kim [*1 2] **Jaekyeom Kim** [*1 2] **Yeonwoo Jeong** [1 2] **Sergey Levine** [3] **Hyun Oh Song** [1 2]

1. Experiment Hyperparameters

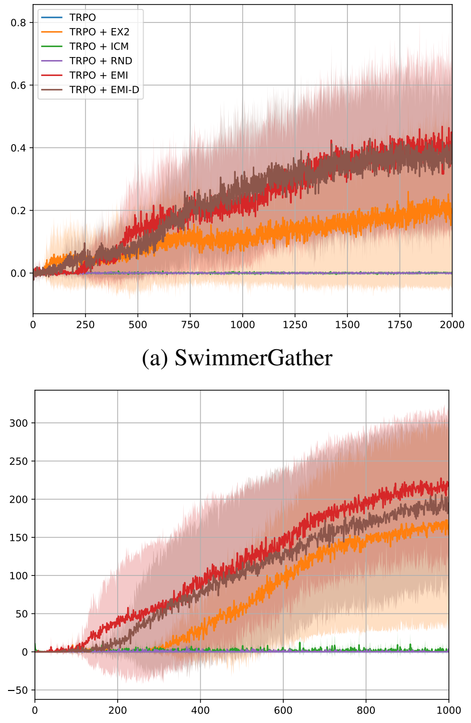
In all of our experiments, we use Adam optimizer with the learning rate of 0.001 and a minibatch size of 512 for 3 epochs to optimize embedding networks. In each iteration, we train the embedding networks. The embedding dimensionality is set to $d = 2$ and the intrinsic reward coefficient is set as 0.001 in all environments. Table 1 and Table 2 give the detailed information of the remaining hyperparameters.

2. Different intrinsic reward formulation

We evaluate the performance under another intrinsic reward function. Apart from prediction error formulation in our main paper, we also consider the relative difference in the novelty of state representations, based on the distance in the embedding space similar to (Oh et al., 2015) as shown in Equation (1).

$$r_d(s_t, a_t, s_t') = g(s_t) - g(s_t'), \tag{1}$$
$$\text{where} \quad g(s) = \frac{1}{n}\sum_{i=1}^{n} \exp\left\{\left(-\frac{\|\phi(s) - \phi(s_i)\|^2}{2\sigma^2}\right)\right\}$$

The relative difference makes sure the intrinsic reward diminishes to zero (Ng et al., 1999) once the agent has sufficiently explored the state space. We label EMI using this diversity based intrinsic reward Equation (1) as EMI-D. Figure 1 and Figure 3 show performance of EMI-D compared to EMI and the baseline exploration methods on MuJoCo and Atari domains respectively. The results show comparable performance in most environments with respect to EMI. In EMI-D, we set $\lambda_{\text{info}} = 0.05$, $\lambda_{\text{error}} = 10000$ and apply action embedding regularization for MuJoCo



(a) SwimmerGather

(b) SparseHalfCheetah

Figure 1: Performance of EMI and EMI-D on locomotion tasks with sparse rewards compared to the baseline methods. The solid lines show the mean reward (y-axis) of 5 different seeds at each iteration (x-axis).

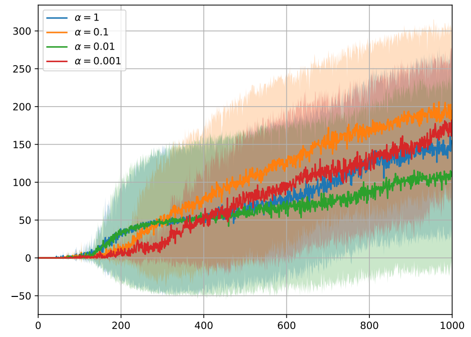


Figure 2: Study of intrinsic reward coefficient α in EMI-D on SparseHalfCheetah environment.

[*] Equal contribution [1] Seoul National University, Department of Computer Science and Engineering [2] Neural Processing Research Center [3] UC Berkeley, Department of Electrical Engineering and Computer Sciences. Correspondence to: Hyun Oh Song <hyunoh@snu.ac.kr>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

Environments	SwimmerGather	SparseHalfCheetah
TRPO method	Single Path	
TRPO step size	0.01	
TRPO batch size	50k	5k
Policy network	A 2-layer FC with (64, 32) hidden units (tanh)	
Baseline network	A 32 hidden units FC (ReLU)	Linear baseline
λ_{error}	0.001	5
λ_{info}	1	
ϕ network	Same structure as policy network	
ψ network	A 64 hidden units FC (ReLU)	
Information network	A 2-layer FC with (64, 64) hidden units (ReLU)	
Error network	State input passes the same network structure as policy network. Concat layer concatenates state output and action. A 256 units FC (ReLU)	
Max path length	500	
Discount factor	0.995	

Table 1: Hyperparameters for MuJoCo experiments.

Environments	Freeway, Frostbite, Venture, Montezuma's Revenge, Gravitar, Solaris
TRPO method	Single Path
TRPO step size	0.01
TRPO batch size	100k
Policy network	2 convolutional layers (16 8x8 filters of stride 4, 32 4x4 filters of stride 2), followed by a 256 hidden units FC (ReLU)
Baseline network	Same structure as policy network
ϕ network	Same structure as policy network
ψ network	A 64 hidden units FC (ReLU)
λ_{error}	100
λ_{info}	0.1
Information network	A 2-layer FC with (64, 64) hidden units (ReLU)
Error network	State input passes the same network structure as policy network. Concat layer concatenates state output and action. A 256 units FC (ReLU)
Max path length	4500
Discount factor	0.995

Table 2: Hyperparameters for Atari experiments.

experiments. For Atari experiments, we use the same hyperparameters as in EMI.

For reward augmentation, EMI-D uses intrinsic reward r_d and then learns from $r = r_{env} + \alpha r_d$. Figure 2 shows the impact of α in EMI-D. Although $\alpha = 0.1$ gives the best performance, other choices also give comparable performance.

3. Computation of the mutual information term

Given a minibatch $\{(s_{t_l}, a_{t_l}, s'_{t_l})\}_{l=1}^{m}$, we can construct the following inputs.

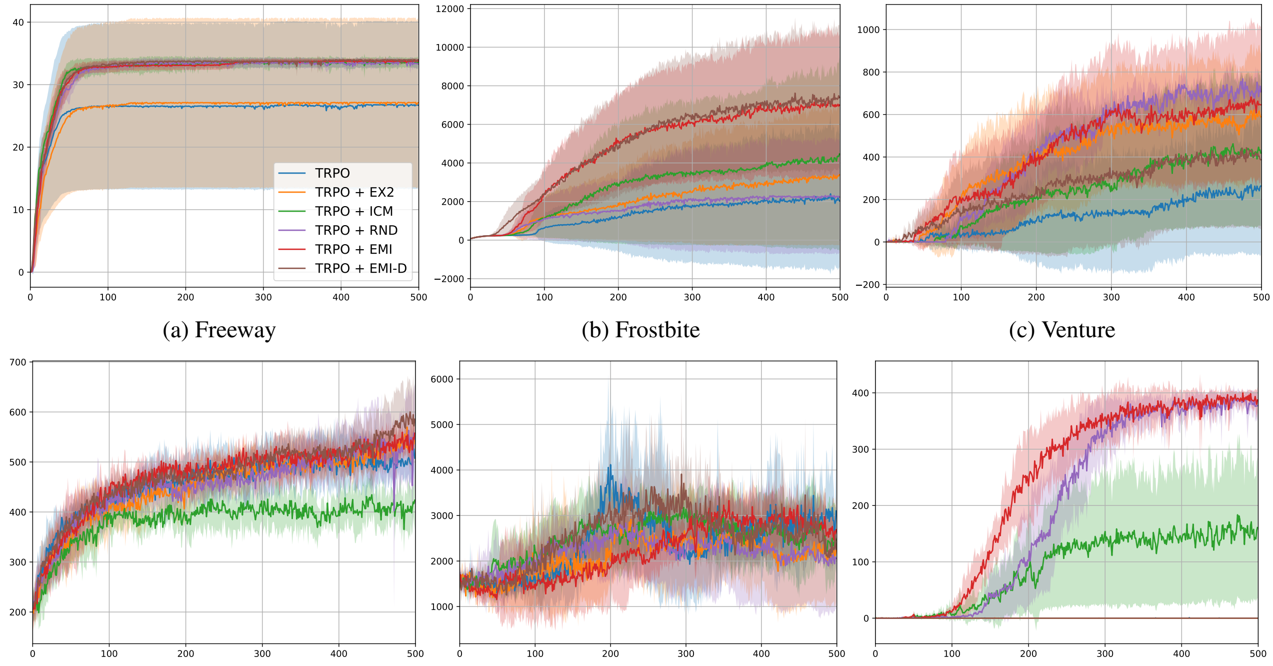
$$D = \left\{ \left(\phi(s_{t_l}), \psi(a_{t_l}), \phi(s'_{t_l}) \right) \right\}_{l=1}^{\lfloor \frac{m}{2} \rfloor}$$

$$D_s = \left\{ \left(\phi(s_{t_l}), \psi(a_{t_l}), \phi\left(s'_{t_{l+\lfloor \frac{m}{2} \rfloor}} \right) \right) \right\}_{l=1}^{\lfloor \frac{m}{2} \rfloor}$$

$$D_a = \left\{ \left(\phi(s_{t_l}), \psi\left(a_{t_{l+\lfloor \frac{m}{2} \rfloor}} \right), \phi(s'_{t_l}) \right) \right\}_{l=1}^{\lfloor \frac{m}{2} \rfloor}$$

Then the mutual information term $\mathcal{L}_{\text{info}}$ in Equation (7) from the main text, is computed as follows.

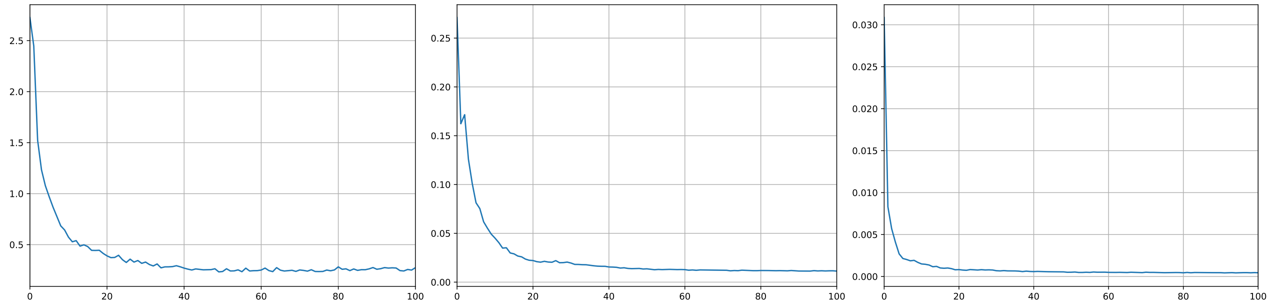
$$\mathcal{L}_{\text{info}} = \inf_{\omega_S \in \Omega_S} \left[\mathbb{E}_{d \in D} \operatorname{sp}\left(-T_{\omega_S}(d)\right) + \mathbb{E}_{d_s \in D_s} \operatorname{sp}\left(T_{\omega_S}(d_s)\right) \right] + \inf_{\omega_A \in \Omega_A} \left[\mathbb{E}_{d \in D} \operatorname{sp}\left(-T_{\omega_A}(d)\right) + \mathbb{E}_{d_a \in D_a} \operatorname{sp}\left(T_{\omega_A}(d_a)\right) \right]$$



(a) Freeway (b) Frostbite (c) Venture

(d) Gravitar (e) Solaris (f) Montezuma's Revenge

Figure 3: Performance of EMI and EMI-D on sparse reward Atari environments compared to the baseline methods. The solid lines show the mean reward (y-axis) of 5 different seeds at each iteration (x-axis).



(a) Mutual information loss (b) Linear dynamics loss (c) Model error loss

Figure 4: Convergence of loss term values (y-axis) across the iterations (x-axis) in SparseHalfCheetah.

4. Experimental evaluation of the error model

To get an understanding of the empirical behavior of the error model, we visualize the evolution of the error model norm $\|S(s_t, a_t)\|_2$ throughout a full episode from one of our experiments on Montezuma's Revenge, in Figure 5. We picked five representative transitions with high values of the error model norm from the episode. The upper and lower images of each transition in the figure represent s_t and s'_t, respectively.

In the case of transitions b, c, and e, due to the discrepancy between the two distinct background images, $\|\phi(s_t) - \phi(s'_t)\|_2$ easily becomes large which makes the residual error as well as the error term larger, too. Transitions a and d belong to the case where the action chosen by the policy has no or almost no effect on s'_t i.e. $P(s'_t|s_t, a_t) \approx P(s'_t|s_t)$. Linear models without any error terms can fail in such events easily. Thus, the error term in our model gets bigger to mitigate the modeling error.

In conclusion, we observed the error terms generally had much larger norms in the cases such as the representative transitions, in order to alleviate the occasional irreducible large residual errors under the linear dynamics model.

5. Convergence of loss terms

Figure 4 shows the convergence of loss terms in Equation (7) from the main text. All loss terms reach convergence within the first 50 iterations, which verifies that EMI successfully

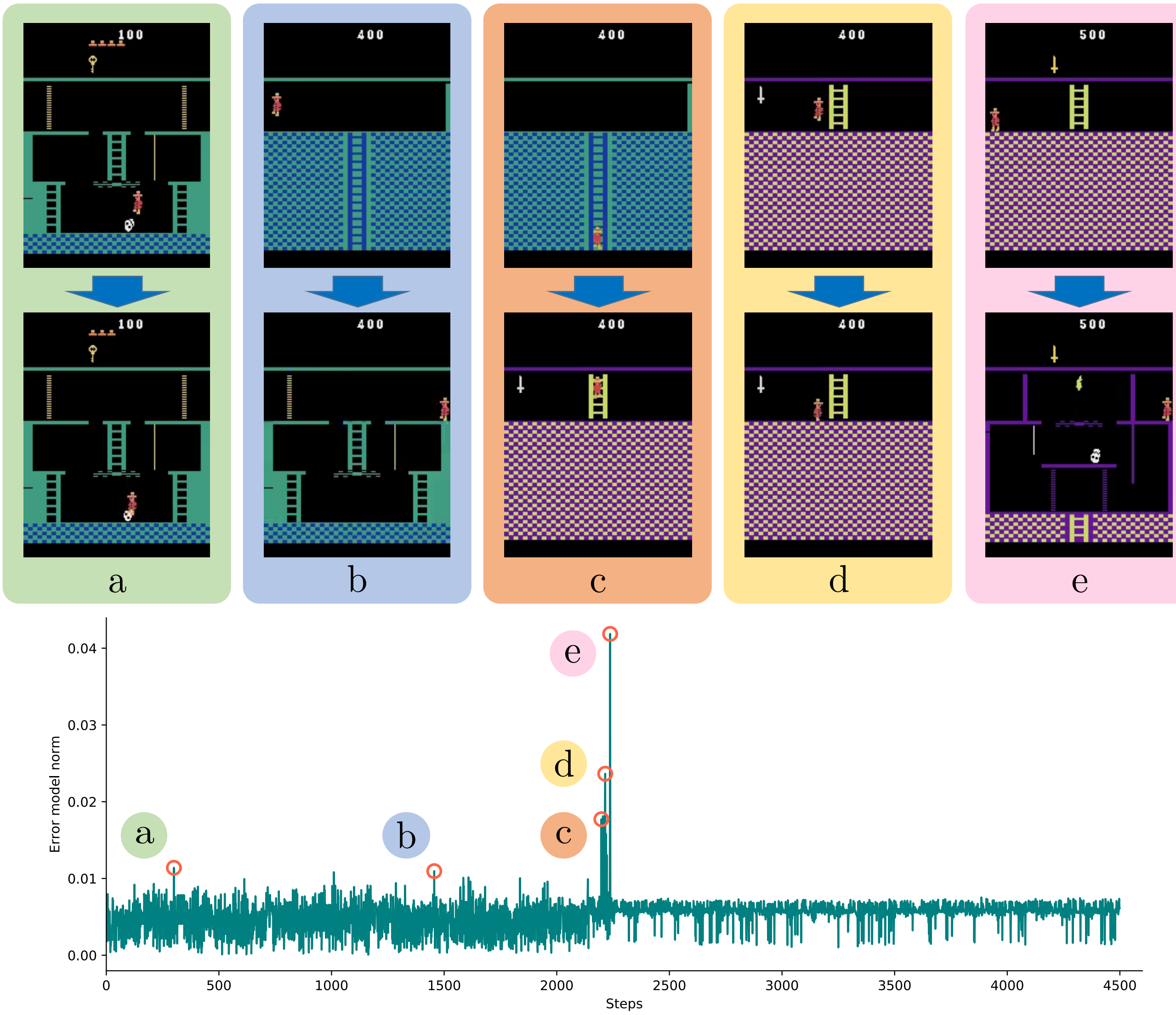


Figure 5: Evolution of the error model norm, and the five representative transitions with the high norm values, in a sample episode of the EMI agent on Montezuma's Revenge. The y-axis and the x-axis mean the error model norm and the step number in the episode, respectively. Each colored pair of two images represent s_t (upper) and s_t' (lower) of its corresponding transition. In transitions b, c, and e, s_t and s_t' are from different rooms with distant background images. In transitions a and d, the agent is off the platform and thus has no control over itself in s_t.

learns desired embedding representations.

6. Statistical tests

As TRPO exhibits high-variance results, we ran more seeds to verify the statistical significance of EMI. We ran 15 random seeds on the SparseHalfCheetah environment which we claim EMI outperforms other baselines, the difference in the mean returns is relatively small, and the variance is high. We then performed the t-test to confirm the statistical significance following the practice from Colas et al. (2018). For each baseline methods, we report t-values with p-values in parentheses. (Results are significant when $p < 0.05$)

- EMI vs ICM: 8.58 (2.99e-7)
- EMI vs RND: 8.57 (2.96e-7)
- EMI vs EX2: 1.81 (0.0410)

The results show that in SparseHalfCheetah environment, EMI outperforms the baseline methods within the 95% confidence level.

7. The BoxImage experiment

The intrinsic position of the agent, x, is constrained within $x \in [0, 100]^2$. Observations the agent receives are $52 \times 52 \times 1$ images, each of which has a white circle that corresponds to the intrinsic position of the agent on a black background. The agent can move itself by performing an action $a \in [-1, 1]^2$. Concretely, if the agent performs a at x, its next intrinsic position will be $\min(\max(x + a, (0, 0)), (100, 100))$. The initial intrinsic position of the agent, x_i, is randomly chosen satisfying $\|x_i\| \geq 75$.

We collected 30,000 samples with a randomly initialized TRPO policy in BoxImage. Using the above samples, we trained two set of embedding functions each with the same hyper-parameters ($\lambda_{\text{error}} = 100, \lambda_{\text{info}} = 0.01, d = 2$) but with an exception of whether to regularize the *state* or the *action* embeddings.

References

Colas, C., Sigaud, O., and Oudeyer, P.-Y. Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*, 2018.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pp. 2863–2871, 2015.