Energy-based Predictive Representations for Partially Observed Reinforcement Learning (Supplementary Material)

Tianjun Zhang[1, 2, *] Tongzheng Ren[1, 3, *] Chenjun Xiao[4] Wenli Xiao[2]

Joseph E. Gonzalez[2] Dale Schuurmans[1, 4] Bo Dai[1, 5]

[1]Google Research, Brain Team
[2]Department of EECS, UC Berkeley
[3] Department of Computer Science, UT Austin
[4]Department of Computer Science, University of Alberta
[5]School of Computational Science and Engineering, Georgia Tech

A DERIVATION OF THE RANDOM FEATURE IN (14)

We have that

$$P(o_{t+1}|x_t, a_t) = p(o_{t+1}) \exp\left(f(x_t, a_t)^\top \left(g(o_{t+1}) + \lambda f(x_t, a_t)\right)\right) \tag{1}$$

$$= p(o_{t+1}) \exp\left(\left(\lambda - \frac{1}{2}\right) \|f(x_t, a_t)\|^2\right) \exp\left(-\frac{\|g(o_{t+1})\|^2}{2}\right) \exp\left(\frac{\|f(x_t, a_t) + g(o_{t+1})\|^2}{2}\right), \tag{2}$$

where we have that

$$\exp\left(\frac{\|f(x_t, a_t) + g(o_{t+1})\|^2}{2}\right) \tag{3}$$

$$= (2\pi)^{-d/2} \exp\left(\frac{\|f(x_t, a_t) + g(o_{t+1})\|^2}{2}\right) \int \exp\left(-\frac{\|\omega - (f(x_t, a_t) + g(o_{t+1}))\|^2}{2}\right) d\omega \tag{4}$$

$$= (2\pi)^{-d/2} \int \exp\left(-\frac{\|\omega\|^2}{2} + \omega^\top (f(x_t, a_t) + g(o_{t+1}))\right) d\omega \tag{5}$$

$$= \mathbb{E}_{\omega \sim \mathcal{N}(0, I_d)} \left[\exp\left(\omega^\top f(x_t, a_t)\right) \exp\left(\omega^\top g(o_{t+1})\right)\right], \tag{6}$$

which concludes the proof for (14).

B OBSERVABLE LQG AS EPR

Follow the standard notations, the dynamics of Linear-Quadratic Gaussian is defined as

$$s_t = As_{t-1} + Ba_t + w_t, \tag{7}$$

$$o_t = Cs_{t-1} + z_t, \tag{8}$$

where w_t and z_t are Gaussian noise. Define the matrix

$$G_L = [C^\top, CA^\top, \ldots, \left(CA^{L-1}\right)^\top]^\top,$$

and reduced observation

$$\tilde{o}_t = o_t - z_t - C\left[\sum_{k=0}^{t-2} A^k B a_{t-k-1} + \sum_{k=0}^{t-2} A^k w_{t-k-2}\right].$$

By the observability condition of LQG, G_L is full column rank, one can identify s_0 by

$$s_0 = \left(G_L^\top G_L\right)^{-1} \sum_{j=1}^{L} \left(A^\top\right)^{j-1} C^\top \tilde{o}_j.$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Therefore, we have

$$s_1 = As_0 + Ba_0 + w_0 = A\left(\left(G_L^\top G_L\right)^{-1}\sum_{j=1}^{L}\left(A^\top\right)^{j-1}C^\top\tilde{o}_j\right) + Ba_1 + w_0, \tag{9}$$

$$s_2 = As_1 + Ba_1 + w_1 = A^2\left(\left(G_L^\top G_L\right)^{-1}\sum_{j=1}^{L}\left(A^\top\right)^{j-1}C^\top\tilde{o}_j\right) + ABa_1 + Ba_2 + Aw_0 + w_1, \tag{10}$$

$$s_{L+1} = As_L + Ba_L + w_L = A^L\left(\left(G_L^\top G_L\right)^{-1}\sum_{j=1}^{L}\left(A^\top\right)^{j-1}C^\top\tilde{o}_j\right) + \sum_{j=0}^{L}A^{L-j}Ba_{j+1} + \sum_{j=0}^{L}A^{L-j}w_j, \tag{11}$$

$$o_{L+1} = Cs_{t+1} + z_t = CA^L\left(\left(G_L^\top G_L\right)^{-1}\sum_{j=1}^{L}\left(A^\top\right)^{j-1}C^\top\tilde{o}_j\right) + C\sum_{j=0}^{L}A^{L-j}Ba_{j+1} + C\sum_{j=0}^{L}A^{L-j}w_j + z_t, \tag{12}$$

which means o_{L+1} follows a Gaussian distribution with mean as a function of history $x_L = \left\{(o_{i-1}, a_i)_{i=1}^{L}\right\}$ and action a_{L+1}, and variance as a function of σ_w, σ_z, and (A, B, C). Therefore, we have some function $f_{A,B,C,\sigma_w,\sigma_z}$ and $g_{A,B,C,\sigma_w,\sigma_z}$, such that

$$g_{A,B,C,\sigma_w,\sigma_z}\left(o_{L+1}\right) = f_{A,B,C,\sigma_w,\sigma_z}\left(x_L, a_{L+1}\right) + \xi, \quad \xi \sim \mathcal{N}\left(0, \mathbf{I}\right).$$

On the other hand, we set $\lambda = -\frac{1}{2}$, and $p(o) = \mathcal{N}\left(0, \mathbf{I}\right)$ in (12), then, we obtain

$$p(o_{L+1}|x_L, a_L) \propto \exp\left(-\frac{\|g(o_{L+1}) - f(x_L, a_L)\|_2^2}{2}\right),$$

which reproduces the observable LQG with specific $f_{A,B,C,\sigma_w,\sigma_z}$ and $g_{A,B,C,\sigma_w,\sigma_z}$.

C EXPERIMENT DETAILS

C.1 ONLINE SETTING

In Table 5, we list all the hyperparameters and network architecture we use for our experiments. We see that we don't use the additional exploration bonus term in the mojuco tasks. But this is very helpful in DM control suite tasks, especially in those sparse-reward tasks.

For evaluation in Mujoco, in each evaluation (every 5K steps) we test our algorithm for 10 episodes. We average the results over the last 4 evaluations and 4 random seeds. For Dreamer and Proto-RL, we change their network from CNN to 3-layer MLP and disable the image data augmentation part (since we test on the state space). The architecture we used for the transformer is following the Trajectory Transformer [Janner et al., 2021]. The attention used is the causal attention.

C.2 LEARNING CURVES

We provide the performance curves for online DM Control Suite experiments in Figure 1. As we can see in the figures, the proposed EPR converges faster and achieve the state-of-the-art performances in most of the environments, demonstrating the sample efficiency and the ability to balance of exploration vs. exploitation of EPR. We also provide additional curves for POMDP setting in Figure 2.

C.3 IMAGE-BASED EXPERIMENTS

We provide the details of metaworld image-based experiments here. We first provide an illustration of the reach environment in Figure 3. We then provide some more experiment details in the following section.

Table 1: Hyperparameters used for EPR in all the environments in MuJoCo and DM Control Suite.

	Hyperparameter Value
Bonus Coefficient (MuJoCo)	0.0
Bonus Coefficient (DM Control)	5.0
Actor lr	0.0003
Model lr	0.0003
Actor Network Size (MuJoCo)	(256, 256)
Actor Network Size (DM Control)	(1024, 1024)
ERP Embedding Network Size (MuJoCo)	(1024, 1024, 1024)
ERP Embedding Network Size (DM Control)	(1024, 1024, 1024)
Critic Network Size (MuJoCo)	(1024, 1)
Critic Network Size (DM Control)	(1024, 1)
Discount	0.99
Target Update Tau	0.005
Model Update Tau	0.005
Batch Size	256

Table 2: Settings of adapted OpenAI Fetch-Reach Environment.

	Hyperparameter Value
Maximum Episode Steps	50
Reward Type	'sparse'
Observation Size	(3, 64, 64)
Fixed Goal Position	(1.27, 0.90, 0.66)

References

M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. *NeurIPS*, 2021.

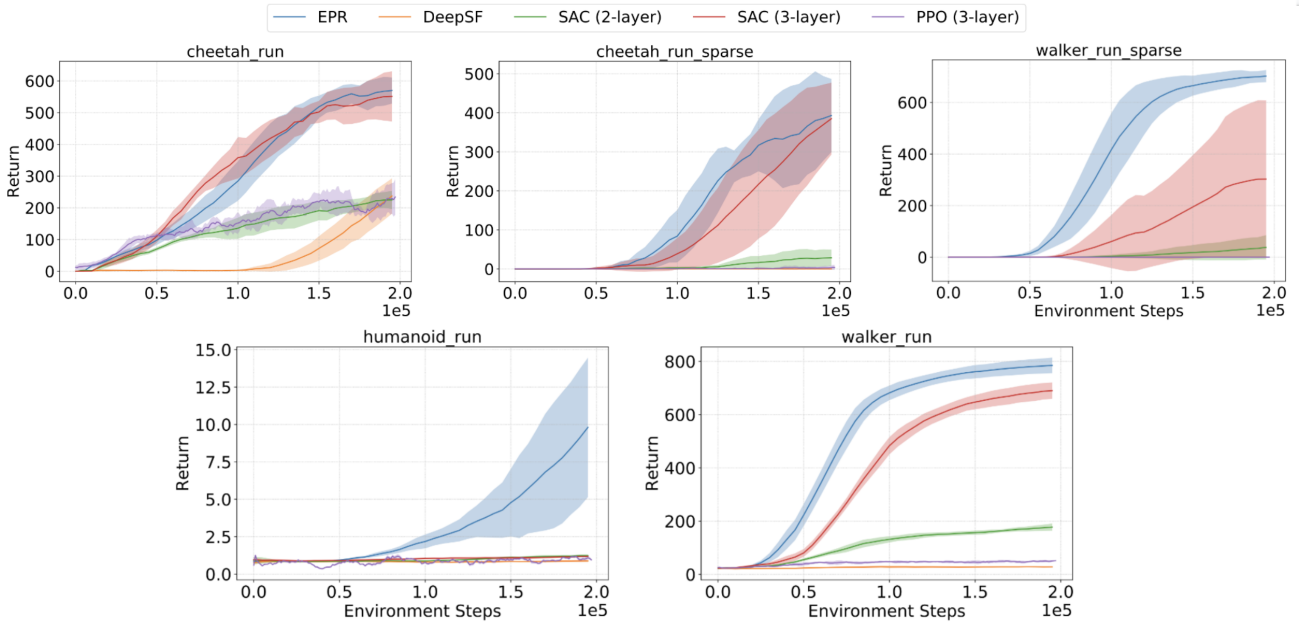

Figure 1: Performance Curves for online DM Control Suite.

Table 3: Hyperparameters used for EPR in FetchReachImage.

	Hyperparameter Value
Bonus Coefficient (MuJoCo)	0.0
Bonus Coefficient (DM Control)	5.0
Actor lr	0.0003
Model lr	0.0003
Actor Network Size (MuJoCo)	(256, 256)
Actor Network Size (DM Control)	(1024, 1024)
ERP Embedding Network Size (MuJoCo)	(1024, 1024, 1024)
ERP Embedding Network Size (DM Control)	(1024, 1024, 1024)
Critic Network Size (MuJoCo)	(1024, 1)
Critic Network Size (DM Control)	(1024, 1)
Discount	0.99
Target Update Tau	0.005
Model Update Tau	0.005
Batch Size	256

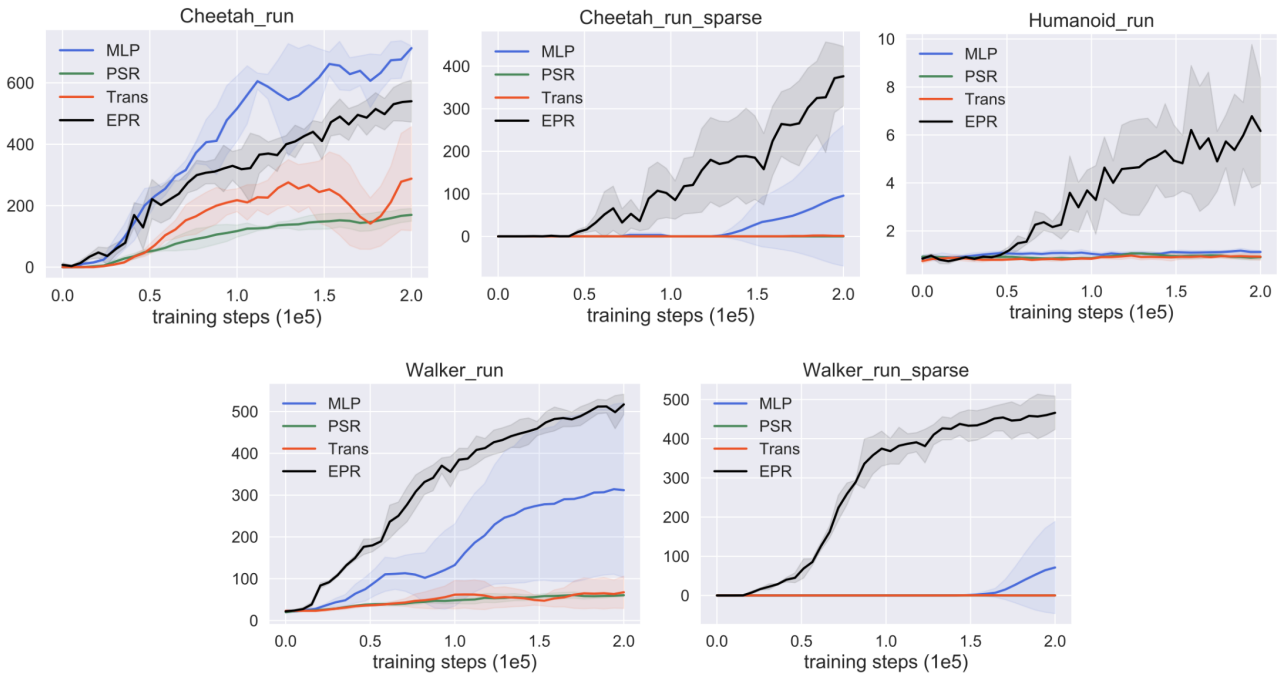

Figure 2: Performance Curves for online POMDP DM Control Suite.

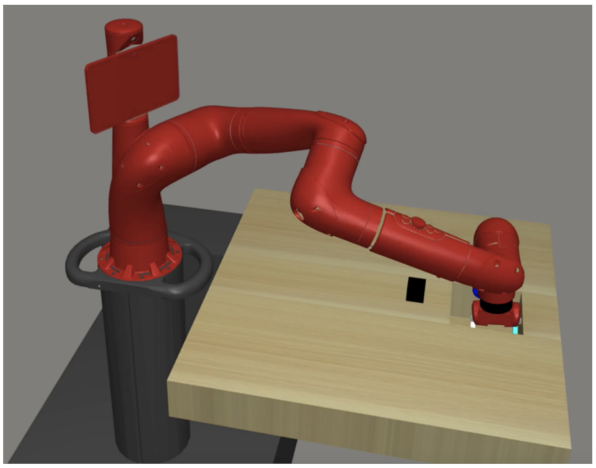

Figure 3: **Reach environment:** Using a robot arm to reach a specific position.

Table 4: Hyperparameters used for SPR in FetchReachImage.

	Hyperparameter Value
lr	0.0001
Dropout	0.5
Discount	0.99
Batch Size	32
Augmentation	off
Target Update Tau	0.005
Model Update Tau	0.005
Batch Size	256
Update	Distributional Q
Dueling	True
Optimizer	Adam
Optimizer: learning rate	0.0001
Max gradient norm	10
Priority exponent	0.5
Noisy nets parameter	0.5
Min replay size for sampling	2000
Replay period every	1 step
Updates per step	2
Multi-step return length	10
Q network: channels	32, 64, 64
Q network: filter size	8×8, 4×4, 3×3
Q network: stride	4, 2, 1
Q network: hidden units	256
Non-linearity	ReLU
Target network: update period	1
λ (SPR loss coefficient	2
K (Prediction Depth)	5

Table 5: Hyperparameters used for SAC-AE in FetchReachImage.

	Hyperparameter Value
Critic lr	0.001
Actor lr	0.001
Discount	0.99
Batch Size	128
Critic Q-function soft-update rate τ_Q	0.01
Critic encoder soft-update rate τ_{enc}	0.05
Critic target update frequency	2
Actor update frequency	2
Actor standard deviation bounds	$[-10, 2]$
Autoencoder learning rate	0.001
Temperature learning rate	0.0001
Temperature Adam's β_1	0.5
Init temperature	0.1