Supplementary Materials to Predictive Coding for Locally-Linear Control

A. Proofs in Section 3

A.1. Connecting (SOC1) and (SOC1-E) with Next-observation Prediction

Recall that for an arbitrarily given encoder E the proxy cost function in the observation space is given by $c_E(x,u) := \mathbb{E}\Big[\bar{c}(z,u) \mid E(x)\Big]$, where z is sampled from $E(x)$. Equipped with this cost the only difference between (SOC1-E), i.e., $\min_U \ L(U,p,c_E,x_0)$, and the original problem (SOC1), i.e., $\min_U \ L(U,p,c,x_0)$, is on the cost function used.

To motivate the heuristic method of learning an encoder E by maximizing the likelihood of the next-observation prediction model, we want to show there exists at least one latent cost function $\bar{c}$ such that the aforementioned approach makes sense. Followed from the equivalence of the energy-based graphical model (Markov random field) and Bayesian neural network (Koller & Friedman, 2009), for any arbitrary encoder E there exists a latent dynamics model $\tilde{F}$ and decoder $\tilde{D}$ such that any energy-based LCE model that has an encoder model E, namely $q_E(x'|x,u)$, can be written as $(\tilde{D}\circ\tilde{F}\circ E)(x'|x,u)$.

Now, suppose for simplicity the observation cost is only state-dependent, and the latent cost $\bar{c}$ is constructed as follows: $\bar{c}(z,u) := \int_{x'}\int_{z'} c(x')d\tilde{F}(z'|z,u)d\tilde{D}(x'|z')$. Then one can write $c_E(x,u) = \int_{x'} dq_E(x'|x,u)c(x')$, and this implies

$$\Big|\mathbb{E}_{x'\sim p(\cdot|x,u)}[c(x')] - c_E(x,u)\Big| \le c_{\max}\cdot D_{\text{TV}}(p(\cdot|x,u)||q_E(\cdot|x,u)),$$

where D_{TV} is the total variation distance of two distributions. Using analogous derivations of Lemma 11 in (Petrik et al., 2016), for the case of finite-horizon MDPs, one has the following chain of inequalities for any given control sequence $\{u_t\}_{t=0}^{T-1}$ and initial observation x_0:

$$\begin{aligned}
|L(U,p,c,x_0) - L(U,p,c_E,x_0)| =& \left|\mathbb{E}\left[\sum_{t=1}^{T} c_t(x_t)\,|\,P,x_0\right] - \mathbb{E}\left[\sum_{t=0}^{T-1} c_{E,t}(x_t,u_t)\,|\,P,x_0\right]\right| \\
\le& T^2\cdot c_{\max}\,\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} D_{\text{TV}}(p(\cdot|x_t,u_t)||q_E(\cdot|x_t,u_t))\mid P,x_0\right] \\
\le& \sqrt{2}T^2\cdot c_{\max}\,\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\sqrt{D_{\text{KL}}(p(\cdot|x_t,u_t)||\widehat{p}_E(\cdot|x_t,u_t))}\mid P,x_0\right] \\
\le& \sqrt{2}T^2\cdot c_{\max}\sqrt{\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} D_{\text{KL}}(p(\cdot|x_t,u_t)||\widehat{p}_E(\cdot|x_t,u_t))\mid P,x_0\right]},
\end{aligned}$$

The first inequality is based on the result of the above lemma, the second inequality is based on Pinsker's inequality, and the third inequality is based on Jensen's inequality of $\sqrt{(\cdot)}$ function.

Notice that for any arbitrary action sequence it can always be expressed in form of deterministic policy $u_t = \pi'(x_t,t)$ with some non-stationary state-action mapping π'. Therefore, the KL term can be written as:

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} D_{\text{KL}}(p(\cdot|x_t,u_t)||q_E(\cdot|x_t,u_t))\mid p,\pi,x_0\right] \\
=&\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\int D_{\text{KL}}(p(\cdot|x_t,u_t)||q_E(\cdot|x_t,u_t))d\pi'(u_t|x_t,t)\mid p,x_0\right] \\
=&\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\int D_{\text{KL}}(p(\cdot|x_t,u_t)||q_E(\cdot|x_t,u_t))\cdot\frac{d\pi'(u_t|x_t,t)}{dU(u_t)}\cdot dU(u_t)\mid p,x_0\right] \le \overline{U}\cdot\mathbb{E}_{x,u}\left[D_{\text{KL}}(p(\cdot|x,u)||q_E(\cdot|x,u))\right],
\end{aligned} \tag{3}$$

where the expectation is taken over the state-action stationary distribution of the finite-horizon problem that is induced by data-sampling policy U. The last inequality is due to change of measures in policy, and the last inequality is due to the facts that (i) π is a deterministic policy, (ii) $dU(u_t)$ is a sampling policy with lebesgue measure $1/\overline{U}$ over all control actions, (iii) the following bounds for importance sampling factor holds: $\left|\frac{d\pi'(u_t|x_t,t)}{dU(u_t)}\right| \leq \overline{U}$.

Combining the above arguments we have the following inequality for any given encoder model E and any control sequence U:

$$|L(U, p, c, x_0) - L(U, p, c_E, x_0)| \leq \sqrt{2}T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{\mathbb{E}_{x,u}\left[D_{\text{KL}}(p(\cdot|x,u)||q_E(\cdot|x,u))\right]}. \tag{4}$$

Using the above results we now have the following sub-optimality performance bound between the optimizer of (SOC1), U_1^*, and the optimizer of (SOC1-E), $U_{\text{1-E}}^*$:

$$\begin{aligned} L(U_1^*, p, c, x_0) \geq & L(U_1^*, p, c_E, x_0) - \sqrt{2}T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{\mathbb{E}_{x,u}\left[D_{\text{KL}}(p(\cdot|x,u)||q_E(\cdot|x,u))\right]} \\ \geq & L(U_{\text{1-E}}^*, p, c_E, x_0) - \sqrt{2}T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{\mathbb{E}_{x,u}\left[D_{\text{KL}}(p(\cdot|x,u)||q_E(\cdot|x,u))\right]}. \end{aligned} \tag{5}$$

This shows that the performance gap between (SOC1) and (SOC1-E) is bounded by the prediction loss $\sqrt{2}T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{\mathbb{E}_{x,u}\left[D_{\text{KL}}(p(\cdot|x,u)||q_E(\cdot|x,u))\right]}$. Thus this result motivates the approach of learning the encoder model E of proxy cost by maximizing the likelihood of the next-observation prediction LCE model.

A.2. Proof of Lemma 1

We first provide the proof in a more general setting. Consider the data distribution $p(x, y)$. Given any two representation functions $e : \mathcal{X} \to \mathcal{A}$ and $f : \mathcal{Y} \to \mathcal{B}$, we wish to inquire how good these two functions are for constructing a predictor of y given x. To do so, we introduce a restricted class of prediction models of the form

$$q_\psi(y \mid x) \propto \psi_1(y)\psi_2(e(x), f(y)), \tag{6}$$

Let $q^*(y \mid x)$ denote the model that minimizes

$$\ell^* = \min_q \mathbb{E}_{p(x)} D_{KL}(p(y \mid x) || q_\psi(y \mid x)). \tag{7}$$

Our goal is to upper bound the best possible loss ℓ^* based on the mutual information gap $I(X\,;Y) - I(e(X)\,;f(Y))$. In particular, we find that

$$\mathbb{E}_{p(x)} D_{KL}(p(y \mid x) || q^*(y \mid x)) \le I(X\,;Y) - I(e(X)\,;f(Y)). \tag{8}$$

We prove via explicit construction of a model $q(y \mid x)$ whose corresponding loss ℓ is exactly the mutual information gap. Let (X, Y) be joint random variables associated with $p(x, y)$. Let $r(a \mid b)$ be the conditional distribution of $a = e(x)$ given $b = f(y)$ associated with the joint random variables $(A, B) = (e(X), f(Y))$. Simply choose

$$q(y \mid x) \propto p(y)r(e(x) \mid f(y)) \implies q(y \mid x) = \frac{p(y)r(e(x) \mid f(y))}{\mathbb{E}_{p(y')} r(e(x) \mid f(y'))}. \tag{9}$$

Then, by law of the unconscious statistician, we see that

$$\begin{aligned}
\mathbb{E}_{p(x,y)} \ln q(y \mid x) &= -H(Y) + \mathbb{E}_{p(x,y)} \ln \frac{r(e(x) \mid f(y))}{\mathbb{E}_{p(y')} r(e(x) \mid f(y'))} && (10)\\
&= -H(Y) + \mathbb{E}_{r(a,b)} \ln \frac{r(a \mid b)}{\mathbb{E}_{r(b')} r(a \mid b')} && (11)\\
&= -H(Y) + I_r(A\,;B) && (12)\\
&= -H(Y) + I(e(X)\,;f(Y)). && (13)
\end{aligned}$$

Finally, we see that

$$\begin{aligned}
\ell = \mathbb{E}_{p(x)} D_{KL}(p(y \mid x) || q(y \mid x)) &= -H(Y \mid X) - \mathbb{E}_{p(x,y)} \ln q(y \mid x) && (14)\\
&= H(Y) - H(Y \mid X) - I(e(X)\,;f(Y)) && (15)\\
&= I(X\,;Y) - I(e(X)\,;f(Y)). && (16)
\end{aligned}$$

Since $\ell^* \le \ell$, the mutual information gap thus upper bounds the loss associated with the best restricted predictor q^*.

To complete the proof for Lemma 1, simply let

$$\begin{aligned}
X &:= (X_t, U_t) && (17)\\
Y &:= X_{t+1} && (18)\\
e(X) &:= (E(X_t), U_t) && (19)\\
f(Y) &:= E(X_{t+1}). && (20)
\end{aligned}$$

A.3. Proof of Lemma 2

For the first part of the proof, at any time-step $t \geq 1$, for any arbitrary control action sequence $\{u_t\}_{t=0}^{T-1}$, and any arbitrary latent dynamics model F, with a given encoder E consider the following decomposition of the expected cost: $\mathbb{E}[c(x_t, u_t) \mid P, x_0] = \mathbb{E}[\bar{c}(z_t, u_t) \mid E, P, x_0] = \int_{x_{0:t}} \prod_{k=1}^{t} P(x_k|x_{k-1}, u_{k-1}) \cdot \int_{z_t} E(z_t|x_t)\bar{c}(z_t, u_t)$. Now consider the two-stage cost function: $\mathbb{E}[c(x_{t-1}, u_{t-1}) + c(x_t, u_t) \mid P, x_0]$. One can express this cost function as

$$
\begin{aligned}
&\mathbb{E}[\bar{c}(z_{t-1}, u_{t-1}) + \bar{c}(z_t, u_t) \mid E, P, x_0] \\
=&\int_{x_{0:t-1}} \prod_{k=1}^{t-1} P(x_k|x_{k-1}, u_{k-1}) \cdot \left(\int_{z_{t-1}} E(z_{t-1}|x_{t-1})\bar{c}(z_{t-1}, u_{t-1}) + \int_{x_t} P(x_t|x_{t-1}, u_{t-1}) \int_{z_t} E(z_t|x_t)\bar{c}(z_t, u_t) \right) \\
\leq&\int_{x_{0:t-2}} \prod_{k=1}^{t-2} P(x_k|x_{k-1}, u_{k-1}) \cdot \Bigg(\int_{z_{t-2}} E(z_{t-2}|x_{t-2}) \int_{z_{t-1}} F(z_{t-1}|z_{t-2}, u_{t-2})\bar{c}(z_{t-1}, u_{t-1}) \\
&\quad + \int_{x_{t-1}} P(x_{t-1}|x_{t-2}, u_{t-2}) \int_{z_{t-1}} E(z_{t-1}|x_{t-1}) \int_{z_t} F(z_t|z_{t-1}, u_{t-1})\bar{c}(z_t, u_t) \Bigg) \\
&\quad + c_{\max} \cdot \int_{x_{0:t-2}} \prod_{k=1}^{t-2} P(x_k|x_{k-1}, u_{k-1}) \cdot \Big(D_{\text{TV}} \left(E \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ E(\cdot|x_{t-2}, u_{t-2}) \right) \\
&\qquad\qquad + \mathbb{E}_{x_{t-1} \sim P(\cdot|x_{t-2}, u_{t-2})} \left[D_{\text{TV}} \left(E \circ P(\cdot|x_{t-1}, u_{t-1}) || F \circ E(\cdot|x_{t-1}, u_{t-1}) \right) \right] \Big) \\
\leq&\int_{x_{0:t-2}} \prod_{k=1}^{t-2} P(x_k|x_{k-1}, u_{k-1}) \int_{z_{t-2}} E(z_{t-2}|x_{t-2}) \int_{z_{t-1}} F(z_{t-1}|z_{t-2}, u_{t-2}) \cdot \\
&\qquad\qquad \left(\bar{c}(z_{t-1}, u_{t-1}) + \int_{z_t} F(z_t|z_{t-1}, u_{t-1})\bar{c}(z_t, u_t) \right) \\
&\quad + c_{\max} \cdot \int_{x_{0:t-2}} \prod_{k=1}^{t-2} P(x_k|x_{k-1}, u_{k-1}) \cdot (2 \cdot D_{\text{TV}} \left(E \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ E(\cdot|x_{t-2}, u_{t-2}) \right) \\
&\qquad\qquad + \mathbb{E}_{x_{t-1} \sim P(\cdot|x_{t-2}, u_{t-2})} \left[D_{\text{TV}} \left(E \circ P(\cdot|x_{t-1}, u_{t-1}) || F \circ E(\cdot|x_{t-1}, u_{t-1}) \right) \right] \Big)
\end{aligned}
$$

The last inequality is based on the chain of inequalities at any $(x_{t-2}, u_{t-2}) \in \mathcal{X} \times \mathcal{U}$:

$$
\begin{aligned}
&D_{\text{TV}} \left(E \circ P \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ F \circ E(\cdot|x_{t-2}, u_{t-2}) \right) \\
\leq &D_{\text{TV}} \left(F \circ E \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ F \circ E(\cdot|x_{t-2}, u_{t-2}) \right) \\
&\qquad + D_{\text{TV}} \left(E \circ P \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ E \circ P(\cdot|x_{t-2}, u_{t-2}) \right) \\
\leq &D_{\text{TV}} \left(E \circ P(\cdot|x_{t-2}, u_{t-2}) || F \circ E(\cdot|x_{t-2}, u_{t-2}) \right) \\
&\qquad + \mathbb{E}_{x_{t-1} \sim P(\cdot|x_{t-2}, u_{t-2})} \left[D_{\text{TV}} \left(E \circ P(\cdot|x_{t-1}, u_{t-1}) || F \circ E(\cdot|x_{t-1}, u_{t-1}) \right) \right],
\end{aligned}
$$

in which the first one is based on triangle inequality and the second one is based on the non-expansive property of D_{TV}. By continuing the above expansion, one can show that

$$
\begin{aligned}
&\left| \mathbb{E} \left[L(U, F, \bar{c}, z_0) \mid E, x_0 \right] - L(U, P, c, x_0) \right| \\
= &\left| \mathbb{E} \left[L(U, F, \bar{c}, z_0) \mid E, x_0 \right] - L(U, P, \bar{c} \circ E, x_0) \right| \\
\leq &T^2 \cdot c_{\max} \, \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} D_{\text{TV}}((E \circ P)(\cdot|x_t, u_t) || (F \circ E)(\cdot|x_t, u_t)) \mid P, x_0 \right] \\
\leq &T^2 \cdot c_{\max} \, \mathbb{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{x_{t+1} \sim P(\cdot|x_t, u_t)} \left[D_{\text{TV}}(E(\cdot|x_{t+1}) || (F \circ E)(\cdot|x_t, u_t)) \right] \mid P, x_0 \right] \\
\leq &\sqrt{2} \cdot \mathbb{E}_{x, u, x' \sim P(\cdot|x, u)} \left[\sqrt{D_{\text{KL}} \Big(E(\cdot|x') || \big(F \circ E \big)(\cdot|x, u) \Big)} \right] \\
\leq &\sqrt{2 \cdot \mathbb{E}_{x, u, x' \sim P(\cdot|x, u)} \left[D_{\text{KL}} \Big(E(\cdot|x') || \big(F \circ E \big)(\cdot|x, u) \Big) \right]},
\end{aligned} \tag{21}
$$

where the second inequality is based on convexity of D_{TV}, the third inequality is based on Pinsker's inequality and the last inequality is based on Jensen's inequality of $\sqrt{(\cdot)}$ function.

For the second part of the proof, one can show the following chain of inequalities for solution of (SOC1-E) and (SOC2):

$$\begin{aligned}
&L(U^*_{\text{1-E}}, P, \overline{c} \circ E, x_0)\\
\geq&\mathbb{E}\left[L(U^*_{\text{1-E}}, F, \overline{c}, z_0) \mid E, x_0\right] - T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
=&\mathbb{E}\left[L(U^*_{\text{1-E}}, F, \overline{c}, z_0) \mid E, x_0\right] + T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
&- 2T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
\geq&\mathbb{E}\left[L(U^*_{\text{2-EF}}, F, \overline{c}, z_0) \mid E, x_0\right] + T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
&- 2T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
\geq&L(U^*_{\text{2-EF}}, P, \overline{c}\circ E, x_0) - 2T^2 \cdot c_{\max}\overline{U} \cdot \sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}\\
\geq&L(U^*_{\text{2-EF}}, P, c, x_0) - 2\underbrace{T^2 \cdot c_{\max}\overline{U}}_{\lambda_{\text{CON}}} \cdot \underbrace{\sqrt{2 \cdot \mathbb{E}_{x,u,x'\sim P(\cdot|x,u)}\left[D_{\text{KL}}(E(\cdot|x_{t+1})||(F\circ E)(\cdot|x_t,u_t))\right]}}_{R_{\text{CON}}(E,F)},
\end{aligned}$$

where the first and third inequalities are based on the first part of this lemma, and the second inequality is based on the optimality condition of problem (SOC2). This completes the proof.

B. Experiment Details

In the following sections we will provide the description 4 control domains and implementation details used in the experiments.

B.1. Description of the domains

All control environments are the same as reported in (Levine et al., 2020), except that we report both balance and swing up tasks for pendulum, where the author only reported swing up.

B.2. Implementation details

B.2.1. Hyperparameters

SOLAR training specifics: We use their default setting:

- Batch size of 2.
- ADAM (Kingma & Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Learning rate $\alpha_{\text{model}} = 2 \cdot 10^{-5} \times$ horizon for learning $\mathcal{MNIW}$ prior and $\alpha = 10^{-3}$ for other parameters.
- $(\beta_{\text{start}}, \beta_{\text{end}}, \beta_{\text{rate}}) = (10^{-4}, 10.0, 5 \cdot 10^{-5})$
- Local inference and control:
  - Data strength: 50
  - KL step: 2.0
  - Number of rollouts per iteration: 100
  - Number of iterations: 10

PCC training specifics: We use their reported setting:

- Batch size of 128[12].
- ADAM with $\alpha = 5 \cdot 10^{-4}$, $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$.
- L2 regularization with a coefficient of 10^{-3}.
- $(\lambda_p, \lambda_c, \lambda_{\text{cur}}) = (1, 8, 8)$, and $\delta = 0.01$ for the curvature loss. This setting is shared across all domains.
- Additional VAE (Kingma & Welling, 2013) loss term $\ell_{\text{VAE}} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] + D_{\text{KL}}(q(z|x)||p(z))$ with a very small coefficient of 0.01, where $p(z) = \mathcal{N}(0, 1)$.
- Additional deterministic reconstruction loss with coefficient 0.3: given the current observation x, we take the means of the encoder output and the dynamics model output, and decode to get the reconstruction of the next observation.

PC3 training specifics:

- Batch size of 256.
- ADAM with $\alpha = 5 \cdot 10^{-4}$, $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$.
- L2 regularization with a coefficient of 10^{-3}.
- Latent noise $\epsilon = 0.1$ and $\lambda_1 = 1$ across all domains without any tuning.
- λ_2 was set to be 1 across all domains, after it was tuned using grid search in range $\{0.5, 0.75, 1\}$ on Planar system.

[12] Training with batch size of 256 gives worse results.

- λ_3 was set to be 7 across all domains, after it was tuned using grid search in range $\{1, 3, 7\}$ on Planar system.
- $\delta = 0.01$ for the curvature loss.
- Additional loss $\ell_{\text{add}} = ||\frac{1}{N}\sum_{i=1}^{N} z_i||_2^2$ with a very small coefficient of 0.01, which is used to center the latent space around the origin. We found this term to be important to stabilize the training process.

B.2.2. NETWORK ARCHITECTURES

We next present the specific architecture choices for each domain. For fair comparison, the architectures were shared across all algorithms when possible, ReLU non-linearities were used between each two layers.

Encoder: composed of a backbone (either a MLP or a CNN, depending on the domain) and an additional fully-connected (FLC) layer that outputs either a vector (for PC3) or a Gaussian distribution (for PCC and SOLAR).

Latent dynamics (PCC and PC3): the path that leads from $\{z, u\}$ to z', composed of a MLP backbone and an additional FLC layer that outputs either a vector (for PC3) or a Gaussian distribution (for PCC and SOLAR).

Decoder (PCC and SOLAR): composed of a backbone (either a MLP or a CNN, depending on the domain) and an additional FLC layer that outputs a Bernoulli distribution.

Backward dynamics: the path that leads from $\{z', u, x\}$ to z. Each of the inputs goes through a FLC network $\{N_z, N_u, N_x\}$, respectively. The outputs are concatenated and passed through another FLC network N_{joint}, and finally an additional FLC network which outputs a Gaussian distribution.

Planar system

- Input: 40×40 images. 5000 training samples of the form (x, u, x') for PCC and PC3, and 125 rollouts for SOLAR.
- Actions space: 2-dimensional
- Latent space: 2-dimensional
- Encoder: 3 Layers: 300 units - 300 units - 4 units for PCC and SOLAR (2 for mean and 2 for variance) or 2 units for PC3
- Dynamics: 3 Layers: 20 units - 20 units - 4 units for PCC and SOLAR or 2 units for PC3
- Decoder: 3 Layers: 300 units - 300 units - 1600 units (logits)
- Backward dynamics: $N_z = 5, N_u = 5, N_x = 100 - N_{\text{joint}} = 100 - 4$ units
- Planning horizon: $T = 40$
- iLQR horizon: 10 for PCC and PC3[13]
- Initial standard deviation for collecting data (SOLAR): 1.5 for both global and local traning.

Inverted Pendulum $-$ Swing up and Balance

- Input: Two 48×48 images. 20000 training samples of the form (x, u, x') for PCC and PC3, and 200 rollouts for SOLAR.
- Actions space: 1-dimensional
- Latent space: 3-dimensional
- Encoder: 3 Layers: 500 units - 500 units - 6 units for PCC and SOLAR or 3 units for PC3

[13] In PCC and PC3, we utilize the concept of model predictive control (MPC) and follow the iLQR-MPC procedure, as similarly done in PCC(Levine et al., 2020)

- Dynamics: 3 Layers: 30 units - 30 units - 4 units for PCC and SOLAR or 2 units for PC3
- Decoder: 3 Layers: 500 units - 500 units - 4608 units (logits)
- Backward dynamics: $N_z = 10, N_u = 10, N_x = 200 - N_{\text{joint}} = 200 - 6$ units
- Planning horizon: $T = 100$
- iLQR horizon: 10 for PCC and PC3
- Initial standard deviation for collecting data (SOLAR): 0.5 for both global and local training.

Cartpole

- Input: Two 80×80 images. 15000 training samples of the form (x, u, x') for PCC and PC3, and 300 rollouts for SOLAR.
- Actions space: 1-dimensional
- Latent space: 8-dimensional
- Encoder: 6 Layers: Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Convolutional layer: $32 \times 5 \times 5$; stride (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (2, 2) - Convolutional layer: $10 \times 5 \times 5$; stride (2, 2) - 200 units - 16 units for PCC and SOLAR or 8 units for PC3
- Dynamics: 3 Layers: 40 units - 40 units - 16 units for PCC and SOLAR or 8 units for PC3
- Decoder: 6 Layers: 200 units - 1000 units - 100 units - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $2 \times 5 \times 5$; stride (1, 1)
- Backward dynamics: $N_z = 10, N_u = 10, N_x = 300 - N_{\text{joint}} = 300 - 16$ units
- Planning horizon: $T = 50$
- iLQR horizon: 5 for PCC and PC3
- Initial standard deviation for collecting data (SOLAR): 10 for global and 5 for local training.

3-link Manipulator $-$ Swing up

- Input: Two 80×80 images. 30000 training samples of the form (x, u, x') for PCC and PC3, and 150 rollouts for SOLAR.
- Actions space: 3-dimensional
- Latent space: 8-dimensional
- Encoder: 6 Layers: Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Convolutional layer: $32 \times 5 \times 5$; stride (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (2, 2) - Convolutional layer: $10 \times 5 \times 5$; stride (2, 2) - 200 units - 16 units for PCC and SOLAR or 8 units for PC3
- Dynamics: 3 Layers: 40 units - 40 units - 16 units for PCC and SOLAR or 8 units for PC3
- Decoder: 6 Layers: 200 units - 1000 units - 100 units - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $32 \times 5 \times 5$; stride (1, 1) - Upsampling (2, 2) - Convolutional layer: $2 \times 5 \times 5$; stride (1, 1)
- Backward dynamics: $N_z = 10, N_u = 10, N_x = 300 - N_{\text{joint}} = 300 - 16$ units
- Planning horizon: $T = 200$
- iLQR horizon: 20 for PCC and PC3
- Initial standard deviation for collecting data (SOLAR): 1 for global and 0.5 for local training.

B.3. PC3 hyperparameters tuning

In this section, we present how we select the hyperparameters for PC3. There are 4 hyperparameters that we need to decide, which are λ_1, λ_2, λ_3 and the noise added to future encoded vector σ. We fix $\sigma = 0.1$, $\lambda_1 = 1$ and perform grid search to choose $\lambda_2 \in \{0.5, 0.75, 1\}$ and $\lambda_3 \in \{1, 3, 7\}$. We perform tuning on Planar system, and the best set of hyperparameters is then used in all other domains.

Table 4. Grid search results on Planar system

σ^2	λ_1	λ_2	λ_3	Control result
0.1	1	0.5	1	69.9
0.1	1	0.5	3	70.85
0.1	1	0.5	7	73.28
0.1	1	0.75	1	68.8
0.1	1	0.75	3	72.45
0.1	1	0.75	7	70.7
0.1	1	1	1	72.68
0.1	1	1	3	74.15
0.1	1	1	7	$\mathbf{74.35}$

B.4. SOLAR with SOLAR Data Sampling Scheme

For fair comparison with PC3 and PCC, we allowed SOLAR to collect data uniformly in the state space (specifically, in line 2 in Algorithm 1 in SOLAR paper, for each episode we sample uniformly the initial state, and the rest of the algorithm is kept the same).

In contrast, the original SOLAR scheme samples T actions from the action space and applies the dynamics T times from a same initial state for all episodes. For completeness, Table 5 shows a modified version of Table 3 where the SOLAR results are acquired using SOLAR's original sampling scheme.

Table 5. Percentage steps in goal state for the average model (all) and top 1 model. Since SOLAR is task-specific, it does not have top 1.

Task	PC3 (all)	PCC (all)	SOLAR (all)	PC3 (top 1)	PCC (top 1)
Planar	$\mathbf{74.35 \pm 0.76}$	56.6 ± 3.15	68 ± 3.8	$\mathbf{75.5 \pm 0.32}$	$\mathbf{75.5 \pm 0.32}$
Balance	$\mathbf{99.12 \pm 0.66}$	91.9 ± 1.72	67 ± 2.6	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$
Swing Up	$\mathbf{58.4 \pm 3.53}$	26.41 ± 2.64	35.4 ± 1.9	$\mathbf{84 \pm 0}$	66.9 ± 3.8
Cartpole	$\mathbf{96.26 \pm 0.95}$	94.44 ± 1.34	91.2 ± 5.4	$\mathbf{97.8 \pm 1.4}$	$\mathbf{97.8 \pm 1.4}$
3-link	$\mathbf{42.4 \pm 3.23}$	14.17 ± 2.2	0 ± 0	$\mathbf{78 \pm 1.04}$	45.8 ± 6.4

B.5. PC3 with a dedicated critic

In the main experiments, we use the latent dynamics F as the critic for the CPC loss. There are two reasons to do this, which are mentioned in the main text. However, we also tried to train CPC using a dedicated critic. There are three types of critics that we consider: a separate dynamics, a bilinear critic and a concatenate critic. For each type, we perform hyperparameters tuning carefully and for each setting, we report the latent map size, ℓ_{cpc}, ℓ_{cons}, ℓ_{curv} and the control results. All experiments are run on Planar and Pendulum - Swing up.

B.5.1. Critic as a separate dynamics

We use $F_1(z_{t+1}|z_t, u_t)$ as the critic to optimize the CPC loss, and use $F_2(z_{t+1}|z_t, u_t)$ to optimize the consistency loss. After training, we use F_2 to perform optimal control.

Table 6. Results when using a separate dynamics as the critic for Planar system.

σ^2	λ_1	λ_2	λ_3	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	3.34	3.02	0.85	0.0009	41.33
0.1	1	0.75	1	2.53	3.0	1.08	0.001	29
0.1	1	1	1	2.24	3.1	1.2	0.0012	34.18
0.1	1	0.5	3	3.54	3.35	0.83	0.001	49.63
0.1	1	0.75	3	2.43	2.81	1.07	0.0008	35.9
0.1	1	1	3	2.09	2.71	1.24	0.0009	32.93
0.1	1	0.5	7	3.7	3.07	0.85	0.0007	40.67
0.1	1	0.75	7	2.73	2.92	1.11	0.0006	35.95
0.1	1	1	7	2.37	2.77	1.24	0.0004	29.35

Table 7. Results when using a separate dynamics as the critic for Pendulum

σ^2	λ_1	λ_2	λ_3	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	11.84	3.95	1.7	0.025	46.01
0.1	1	0.75	1	9.19	3.85	1.95	0.024	29.79
0.1	1	1	1	8.58	3.86	2.12	0.03	26.77
0.1	1	0.5	3	10.11	3.92	1.69	0.016	25.34
0.1	1	0.75	3	6.71	3.7	1.97	0.017	33.52
0.1	1	1	3	6.69	3.79	2.1	0.02	35.7
0.1	1	0.5	7	8.53	3.91	1.67	0.01	34.56
0.1	1	0.75	7	5.45	3.59	1.95	0.01	37.1
0.1	1	1	7	4.9	3.63	2.08	0.01	39.45

B.5.2. BILINEAR CRITIC

We use a bilinear function $z_{t+1}^T W(z_t, u_t)$ as the critic in CPC loss. This is implemented as follows: first we feed the concatenation of z_t and u_t through a linear function parameterized by W, then take the dot product of that output with z_{t+1} to finally output the score.

Table 8. Results when using a dedicated bilinear critic for Planar system.

σ^2	λ_1	λ_2	λ_3	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	1.6	0.9	1.4	0.0015	0.25
0.1	1	0.75	1	0.8	0.5	1.62	0.0006	0.75
0.1	1	1	1	0.21	0.16	1.73	0.0002	3.2
0.1	1	0.5	3	1.67	0.98	1.38	0.0008	0
0.1	1	0.75	3	0.94	0.58	1.6	0.0005	0
0.1	1	1	3	0.22	0.17	1.73	0.0001	1.95
0.1	1	0.5	7	1.67	0.96	1.39	0.0007	0
0.1	1	0.75	7	0.83	0.51	1.62	0.0003	0
0.1	1	1	7	0.22	0.17	1.72	0.0001	3.9

Table 9. Results when using a dedicated bilinear critic for Pendulum

σ^2	λ_1	λ_2	λ_3	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	3.17	1.63	2.12	0.03	26.38
0.1	1	0.75	1	2.2	1.44	2.39	0.04	25.76
0.1	1	1	1	1.93	1.35	2.46	0.045	26.76
0.1	1	0.5	3	1.84	1.27	2.19	0.007	26.38
0.1	1	0.75	3	1.4	1.32	2.3	0.01	25.76
0.1	1	1	3	1.19	1.27	2.42	0.02	26.76
0.1	1	0.5	7	2.3	1.4	2.13	0.004	34.8
0.1	1	0.75	7	1.54	1.3	2.27	0.006	19.15
0.1	1	1	7	1.65	1.43	2.33	0.01	37.18

B.5.3. CONCATENATE CRITIC

The critic is a neural network which receives the concatenate of (z_t, u_t, z_{t+1}) as the input and outputs the score.

Table 10. Results when using a concatenate critic for Planar system.

σ^2	λ_1	λ_2	λ_3	Map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	3.88	3.02	0.97	0.0007	39.18
0.1	1	0.75	1	2.72	2.46	1.27	0.0007	20.18
0.1	1	1	1	1.58	1.47	1.53	0.0003	20.13
0.1	1	0.5	3	4.07	2.55	1.11	0.0004	20.08
0.1	1	0.75	3	3.38	2.95	1.19	0.0002	31.95
0.1	1	1	3	0.76	0.77	1.64	0.0001	6.88
0.1	1	0.5	7	3.42	3.08	1.79	0.006	31.05
0.1	1	0.75	7	2.59	1.1	2.4	0.002	29
0.1	1	1	7	2.5	1.43	2.4	0.003	22.48

Table 11. Results when using a concatenate critic for Pendulum

σ^2	λ_1	λ_2	λ_3	Map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Control result
0.1	1	0.5	1	4.61	1.94	2.14	0.007	22.8
0.1	1	0.75	1	2.04	1.08	2.4	0.004	3.62
0.1	1	1	1	1.47	1.08	2.47	0.005	2.61
0.1	1	0.5	3	3.17	1.53	2.22	0.003	15.44
0.1	1	0.75	3	1.04	0.89	2.45	0.0015	6.13
0.1	1	1	3	1.26	1.05	2.47	0.003	4.99
0.1	1	0.5	7	5.17	3.1	1.8	0.006	41.89
0.1	1	0.75	7	1.42	1.1	2.4	0.002	9.29
0.1	1	1	7	1.62	1.43	2.4	0.003	12.92