Predictive Coding for Locally-Linear Control

Rui Shu [*1] **Tung Nguyen** [*2 3] **Yinlam Chow** [4] **Tuan Pham** [2 3] **Khoat Than** [2 3] **Mohammad Ghavamzadeh** [5]
Stefano Ermon [1] **Hung Bui** [2]

Abstract

High-dimensional observations and unknown dynamics are major challenges when applying optimal control to many real-world decision making tasks. The Learning Controllable Embedding (LCE) framework addresses these challenges by embedding the observations into a lower dimensional latent space, estimating the latent dynamics, and then performing control directly in the latent space. To ensure the learned latent dynamics are predictive of next-observations, all existing LCE approaches decode back into the observation space and explicitly perform next-observation prediction—a challenging high-dimensional task that furthermore introduces a large number of nuisance parameters (i.e., the decoder) which are discarded during control. In this paper, we propose a novel *information-theoretic* LCE approach and show theoretically that explicit next-observation prediction can be replaced with predictive coding. We then use predictive coding to develop a *decoder-free* LCE model whose latent dynamics are amenable to locally-linear control. Extensive experiments on benchmark tasks show that our model reliably learns a controllable latent space that leads to superior performance when compared with state-of-the-art LCE baselines.

1. Introduction

With the rapid growth of systems equipped with powerful sensing devices, it is important to develop algorithms that are capable of controlling systems from high-dimensional raw sensory inputs (e.g., pixel input). However, scaling stochastic optimal control and reinforcement learning (RL) methods to high-dimensional unknown environments remains an open challenge. To tackle this problem, a common approach is to employ various heuristics to embed the high-dimensional observations into a lower-dimensional latent space (Finn et al., 2016; Kurutach et al., 2018; Kaiser et al., 2019). The class of Learning Controllable Embedding (LCE) algorithms (Watter et al., 2015; Banijamali et al., 2018; Hafner et al., 2018; Zhang et al., 2019; Levine et al., 2020) further supplies the latent space with a latent dynamics model to enable planning directly in latent space.

Our present work focuses on this class of LCE algorithms and takes a critical look at the prevailing heuristic used to learn the controllable latent space: next-observation prediction. To ensure that the learned embedding and latent dynamics are predictive of future observations, existing LCE algorithms introduce a decoder during training and explicitly perform next-observation prediction by decoding the predicted latent states back into the observation space. Despite its empirical success (Watter et al., 2015; Banijamali et al., 2018; Zhang et al., 2019; Levine et al., 2020), this approach suffers from two critical drawbacks that motivate the search for better alternatives: (i) it requires the model to handle the challenging task of high-dimensional prediction; (ii) it does so in a parameter-inefficient manner—requiring the use of a decoder that is discarded during control.

To address these concerns, we propose a novel information-theoretic LCE approach for learning a controllable latent space. Our contributions are as follows.

1. We characterize the quality of the learned embedding through the lens of *predictive suboptimality* and show that predictive coding (van den Oord et al., 2018) is sufficient for minimizing predictive suboptimality.

2. Based on predictive coding, we propose a simpler and parameter-efficient model that jointly learns a controllable latent space and latent dynamics specifically amenable to locally-linear controllers.

3. We conduct detailed analyses and empirically characterize how model ablation impacts the learned latent space and control performance.

4. Finally, we show that our method out-performs state-of-the-art LCE algorithms on several benchmark tasks,

[*]Equal contribution [1]Stanford University [2]VinAI [3]Hanoi University of Science and Technology [4]Google Research [5]Facebook AI Research. Correspondence to: Rui Shu <ruishu@stanford.edu>, Tung Nguyen <v.tungnd13@vinai.io>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

demonstrating predictive coding as a superior alternative to next-observation prediction when learning controllable embeddings.

2. Background

We are interested in controlling non-linear dynamical systems of the form $s_{t+1} = f_{\mathcal{S}}(s_t, u_t) + w$, over the horizon T. In this definition, $s_t \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$ and $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ are the state and action of the system at time step $t \in \{0, \dots, T-1\}$, w is the Gaussian system noise, and $f_{\mathcal{S}}$ is the smooth non-linear system dynamics. We are particularly interested in the scenario in which we only have access to the high-dimensional observation $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ of each state s_t ($n_x \gg n_s$). This scenario has application in many real-world problems, such as visual-servoing (Espiau et al., 1992), in which we only observe high-dimensional images of the environment and not its underlying state. We further assume that the high-dimensional observations x have been selected such that for any arbitrary control sequence $U = \{u_t\}_{t=0}^{T-1}$, the observation sequence $\{x_t\}_{t=0}^{T}$ is generated by a stationary Markov process, i.e., $x_{t+1} \sim p(\cdot|x_t, u_t),\ \forall t \in \{0, \dots, T-1\}$.[1]

A common approach to control the non-linear dynamical system described above is to solve the following stochastic optimal control (SOC) problem (Shapiro et al., 2009) that minimizes the expected cumulative cost

$$\min_U \ L(U, p, c, x_0) := \mathbb{E}\Big[\sum_{t=0}^{T-1} c(x_t, u_t) \mid p, x_0\Big], \quad \text{(SOC1)}$$

where $c : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is the immediate cost function and x_0 is the observation at the initial state s_0. Throughout the paper, we assume that all immediate cost functions are bounded by $c_{\max} > 0$ and Lipschitz with constant $c_{\text{lip}} > 0$. One form of the immediate cost function that is particularly common in goal tracking problems is $c(x, u) = \|x - x_{\text{goal}}\|^2$, where x_{goal} is the observation at the goal state.

The application of SOC to high-dimensional environments, however, faces several challenges. Since the observations x are high-dimensional and the dynamics in the observation space $p(\cdot|x_t, u_t)$ is unknown, solving (SOC1) is often intractable as it requires solving two difficult problems: high-dimensional dynamics estimation and high-dimensional optimal control. To address these issues, the Learning Controllable Embedding (LCE) framework proposes to learn a low-dimensional latent (embedding) space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$ ($n_z \ll n_x$) and a latent state dynamics, and then perform optimal control in the latent space instead. This framework includes algorithms such as E2C (Watter et al., 2015), RCE (Banijamali et al., 2018), SOLAR (Zhang et al., 2019), and PCC (Levine et al., 2020). By learning a stochastic encoder $E : \mathcal{X} \to \mathbb{P}(\mathcal{Z})$ and latent dynamics $F : \mathcal{Z} \times \mathcal{U} \to \mathbb{P}(\mathcal{Z})$, LCE algorithms defines a new SOC in the latent space,

$$\min_U \ \mathbb{E}\Big[L(U, F, \bar{c}, z_0) \mid E(x_0)\Big], \quad \text{(SOC2)}$$

where z_0 is sampled from the distribution $E(x_0)$, i.e., $z_0 \sim E(z_0 \mid x_0)$, and $\bar{c} : \mathcal{Z} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is the latent cost function. By solving the much lower-dimensional (SOC2), the resulting optimal control U_2^* is then applied as a feasible solution to (SOC1) and incurs a suboptimality that depends on the choice of the encoder E and latent dynamics F.[2]

Although Levine et al. (2020) provided an initial theoretical characterization of this SOC suboptimality, the selection of E and F ultimately remains heuristically-driven in all previous works. These heuristics vary across different studies (Levine et al., 2020; Banijamali et al., 2018; Watter et al., 2015; Zhang et al., 2019; Hafner et al., 2018), but the primary approach employed by the existing LCE algorithms is explicit next-observation prediction. By introducing a decoder $D : \mathcal{Z} \to \mathbb{P}(\mathcal{X})$, the composition $D \circ F \circ E$ is cast as an action-conditional latent variable model; the advances in latent variable modeling (Kingma & Welling, 2013; Rezende et al., 2014; Burda et al., 2015; Johnson et al., 2016; Sohn et al., 2015) are then leveraged to train E, F, and D to perform explicit next-observation prediction by maximizing a lower bound on the log-likelihood $\ln \int D(x_{t+1} \mid z_{t+1}) F(z_{t+1} \mid z_t, u_t) E(z_t \mid x_t)\, \mathrm{d}z_{t:t+1}$, over the dataset whose trajectories are drawn from $p(x_t, u_t, x_{t+1})$.

Next-observation prediction offers a natural way to learn a non-degenerate choice of E and F, and enjoys the merit of being empirically successful. However, it requires the introduction of a decoder D as nuisance parameter that only serves the auxiliary role of training the encoder E and latent dynamics F. The focus of our paper is whether E and F can be successfully selected via a decoder-free heuristic.

3. Information-Theoretic LCE

Existing methods instantiate (SOC2) by learning the encoder E and latent dynamics model F in conjunction with an auxiliary decoder D to explicitly perform next-observation prediction. The auxiliary decoder ensures that the learned representation *can* be used for next-observation prediction, and is discarded after the encoder and latent dynamics model are learned. Not only is this a parameter-inefficient procedure for learning (E, F), this approach also learns (E, F) by explicitly performing the challenging high-dimensional next-observation prediction. In this section, we

[1] One method to enable this Markovian assumption is by buffering observations (Mnih et al., 2013) for a number of time steps.

[2] This suboptimality also depends on $\bar{c}$, but we assume $\bar{c}$ to be simple, e.g., $\bar{c}(z, u) = \|z - z_{\text{goal}}\|^2$, where $z_{\text{goal}} = E(x_{\text{goal}})$. E thus subsumes the responsibility of defining a latent space that is compatible with $\bar{c}$. See Appendix A.1 for further justification.

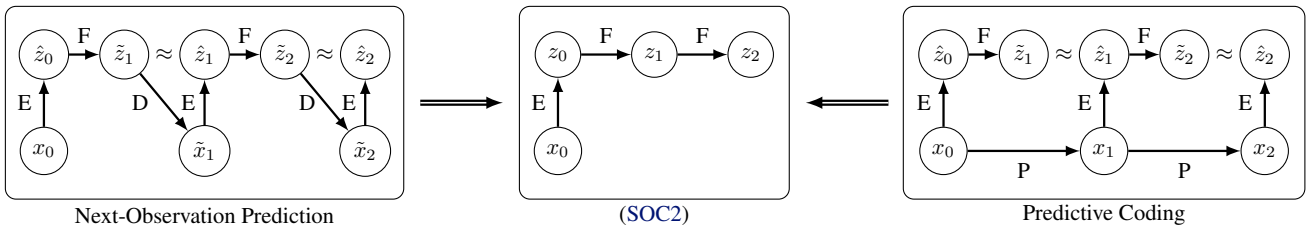

Figure 1. Two high-level approaches to learn an E and F to instantiate (SOC2). One way is to explicitly introduce a decoder D and do next-observation prediction (left), whereas our method uses F as a variational device to train E via predictive coding (right).

propose an information-theoretic approach that can learn (E, F) without decoding and next-observation prediction.

3.1. Predictive Suboptimality of a Representation

Our approach exploits the observation that the sole purpose of the decoder is to ensure that the learned representation is good for next-observation prediction. In other words, the decoder is used to characterize the suboptimality of next-observation prediction when the prediction model is forced to rely on the learned representation. We refer to this concept as *predictive suboptimality* of the learned representation and formally define it as follows.

Definition 1. *Let $p(x_{t+1}, x_t, u_t)$ denote the data distribution. Given an encoder $E : \mathcal{X} \to \mathcal{Z}$,* [3] *let $q(x_{t+1} \mid x_t, u_t)$ denote the prediction model*

$$q(x_{t+1} \mid x_t, u_t) \propto \psi_1(x_{t+1})\psi_2(E(x_{t+1}), E(x_t), u_t),$$

where ψ_1 and ψ_2 are expressive non-negative functions. We define the predictive suboptimality $\ell^*_{pred}(E)$ of a representation induced by E as the best-case prediction loss

$$\min_q \mathbb{E}_{p(x_{t+1}, x_t, u_t)} D_{KL}\left[p(x_{t+1} \mid x_t, u_t) || q(x_{t+1} \mid x_t, u_t)\right].$$

Importantly, the function ψ_2 should measure the compatibility of the triplet (x_{t+1}, x_t, u_t)—but is only allowed to do so via the representations $E(x_{t+1})$ and $E(x_t)$. Thus, the behavior of the representation bottleneck plays a critical role in modulating the expressivity of the model q. If E is invertible, then q is a powerful prediction model; if E is a constant, then q can do no better than marginal density estimation of $p(x_{t+1})$.

While it is possible to minimize the predictive suboptimality of E by introducing the latent dynamics model F and decoder D, and then performing next-observation prediction via $D \circ F \circ E$, our key insight is that predictive suboptimality can be bounded by the following mutual information gap (see Appendix A.2 for proof).

[3] For simplicity, we assume that the encoder E considered here is deterministic.

Lemma 1. *Let X_{t+1}, X_t, and U_t be the random variables associated with the data distribution $p(x_{t+1}, x_t, u_t)$. The predictive suboptimality $\ell^*_{pred}(E)$ is upper bounded by the mutual information gap*

$$I(X_{t+1}\,;\, X_t, U_t) - I(E(X_{t+1})\,;\, E(X_t), U_t).$$

Since $I(X_{t+1}\,;\, X_t, U_t)$ is a constant and upper bounds $I(E(X_{t+1})\,;\, E(X_t), U_t)$ by the data processing inequality, this means we can minimize the predictive suboptimality of E simply by maximizing the mutual information between the future latent state $E(X_{t+1})$ and the current latent state and action pair $(E(X_t), U_t)$—a form of predictive coding. We denote this mutual information $\ell_{\text{MI}}(E)$ as a function of E. To maximize this quantity, we can then leverage the recent advances in variational mutual information approximation (van den Oord et al., 2018; Poole et al., 2019; Belghazi et al., 2018; Nguyen et al., 2010; Hjelm et al., 2018) to train the encoder in a decoder-free fashion.

3.2. Consistency in Prediction of the Next Latent State

A notable consequence of introducing the encoder E is that it can be paired with a latent cost function $\bar{c}$ to define an alternative cost function in the observation space,

$$c_E(x, u) := \mathbb{E}\Big[\bar{c}(z, u) \mid E(x)\Big],$$

where z is sampled from $E(x)$.[4] This is particularly useful for high-dimensional SOC problems, where it is difficult to prescribe a meaningful cost function *a priori* in the observation space. For example, for goal tracking problems using visuosensory inputs, prescribing the cost function to be $c(x, u) = \|x - x_{\text{goal}}\|^2$ suffers from the uninformative nature of the 2-norm in high-dimensional pixel space (Beyer et al., 1999). In the absence of a prescribed c, a natural proxy for the unknown cost function is to replace it with c_E and consider the new SOC problem,

$$\min_U \; L(U, p, c_E, x_0). \qquad \text{(SOC1-E)}$$

[4] In Sections 3.2 and 3.3 , we consider the general case of the stochastic encoder in order to extend the analysis in (Levine et al., 2020). This analysis readily carries over to the limiting case when E becomes deterministic.

Assuming (SOC1-E) to be the de facto SOC problem of interest, we wish to learn an F such that the optimal control U_2^* in (SOC2) approximately solves (SOC1-E). One such consideration for the latent dynamics model would be to set F as the true latent dynamics induced by (p, E), and we refer to such F as the one that is *consistent* with (p, E).

Our main contribution in this section is to justify—from a control perspective—why selecting a consistent F with respect to (p, E) minimizes the suboptimality incurred from using (SOC2) as an approximation to (SOC1-E). The following lemma (see Appendix A.3 for proof) provides the suboptimality performance gap between the solutions of (SOC2) and (SOC1-E).

Lemma 2. *For any given encoder E and latent dynamics F, let $U_{I\text{-}E}^*$ be the solution to (SOC1-E) and U_2^* be a solution to (SOC2). Then, we have the following performance bound between the costs of the control signals $U_{I\text{-}E}^*$ and U_2^*:*

$$L(U_{I\text{-}E}^*, p, c_E, x_0) \geq L(U_2^*, p, c_E, x_0) - 2\lambda_C \cdot \sqrt{2R_C(E, F)}, \tag{1}$$

where $R_C(E, F) = \mathbb{E}_{p(x_{t+1}, x_t, u_t)}[D_{KL}(E(z_{t+1}|x_{t+1}) || (F \circ E)(z_{t+1}|x_t, u_t))]$ and $\lambda_C = T^2 c_{max} \overline{U}$.

In Eq. 1, the expectation is over the state-action stationary distribution of the policy used to generate the training samples (uniformly random policy in this work), and $\overline{U}$ is the *Lebesgue measure* of $\mathcal{U}$.[5] Moreover, $E(z_{t+1}|x_{t+1})$ and $\big(F \circ E\big)(z_{t+1}|x_t, u_t) = \int F(z_{t+1}|z_t, u_t) E(z_t|x_t) \, \mathrm{d}z_t$ are the probability over the next latent state z_{t+1}. Based on Figure 1, we therefore interpret $R_C(E, F)$ as the measure of discrepancy between the dynamics $x_t \to x_{t+1} \to z_{t+1}$ induced by (p, E) versus the latent dynamics model $x_t \to z_t \to z_{t+1}$ induced by (E, F). which we term the *consistency* regularizer. We note that while our resulting bound is similar to Lemma 2 in Levine et al. (2020), there are two key differences. First, our analysis makes explicit the assumption that the cost function c is not prescribed and thus replaced in practice with the proxy cost function c_E based on the heuristically-learned encoder. Second, by making this assumption explicit, our bound is based on samples from the environment dynamics p instead of the next-observation prediction model dynamics $\hat{p}$ as required in Levine et al. (2020).

By restricting the stochastic encoder E to be a distribution with fixed entropy (e.g., by fixing the variance if E is conditional Gaussian), the minimization of the consistency regularizer corresponds to maximizing the log-likelihood of F for predicting z_{t+1}, given (z_t, u_t), under the dynamics induced by (p, E). This correspondence holds even in the limiting case of E being deterministic (e.g., fixing the variance to an arbitrarily small value). In other words, for (SOC2) to approximate (SOC1-E) well, we select F to be a good predictor of the true latent dynamics.

[5] In the case when sampling policy is non-uniform and has no measure-zero set, $1/\overline{U}$ is its minimum measure.

3.3. Suboptimality in Locally-Linear Control

In Section 3.2, we derived the suboptimality of using (SOC2) as a surrogate control objective for (SOC1-E), and showed that the suboptimality depends on the consistency of latent dynamics model F with respect to the true latent dynamics induced by (p, E).

We now shift our attention to the optimization of (SOC2) itself. Similar to previous works (Watter et al., 2015; Banijamali et al., 2018; Zhang et al., 2019; Levine et al., 2020), we shall specifically consider the class of locally-linear control (LLC) algorithms, e.g., iLQR (Li & Todorov, 2004), for solving (SOC2). The main idea in LLC algorithms is to compute an optimal action sequence by linearizing the dynamics around some nominal trajectory. This procedure implicitly assumes that the latent dynamics F has low curvature, so that local linearization via first-order Taylor expansion yields to a good linear approximation over a sufficiently large radius. As a result, the *curvature* of F will play an important role in the optimizability of (SOC2) via LLC algorithms.

Levine et al. (2020) analyzed the suboptimality incurred from applying LLC algorithms to (SOC2) as a function of the curvature of F. For self-containedness, we summarize their analysis as follows. We shall assume F to be a conditional Gaussian model with a mean prediction function $f_{\mathcal{Z}}(z, u)$. The curvature of $f_{\mathcal{Z}}$ can then be measured via

$$R_{\mathrm{LLC}}(F) = \mathbb{E}_{x,u,\eta}\big[\|f_{\mathcal{Z}}(z + \eta_z, u + \eta_u) - f_{\mathcal{Z}}(z, u) - (\nabla_z f_{\mathcal{Z}}(z, u) \cdot \eta_z + \nabla_u f_{\mathcal{Z}}(z, u) \cdot \eta_u)\|_2^2 \mid E\big].$$

where $\eta = (\eta_z, \eta_u)^\top \sim \mathcal{N}(0, \delta^2 I)$, $\delta > 0$ is a tunable parameter that characterizes the radius of latent state-action space in which the latent dynamics model should have low curvature. Let U_{LLC}^* be a LLC solution to (SOC2). Suppose the nominal latent state-action trajectory $\{(z_t, u_t)\}_{t=0}^{T-1}$ satisfies the condition: $(z_t, u_t) \sim \mathcal{N}((z_{2,t}^*, u_{2,t}^*), \delta^2 I)$, where $\{(z_{2,t}^*, u_{2,t}^*)\}_{t=0}^{T-1}$ is the optimal trajectory of (SOC2). Using Eq. 29 of Levine et al. (2020), one can show that with probability $1 - \eta$, the LLC solution of (SOC2) has the following suboptimality performance gap when compared with the optimal cost of this problem using the solution U_2^*,

$$L(U_2^*, F, \bar{c}, z_0) \geq L(U_{\mathrm{LLC}}^*, F, \bar{c}, z_0) - 2\lambda_{\mathrm{LLC}} \cdot \sqrt{R_{\mathrm{LLC}}(F)},$$

where

$$\lambda_{\mathrm{LLC}} = T^2 c_{\max} c_{\mathrm{lip}} (1 + \sqrt{2\log(2T/\eta)}) \sqrt{\overline{UX}}/2,$$

and $\overline{X}$ is the Lebesgue measure with respect to $\mathcal{X}$. We therefore additionally constrain F to have low curvature so that it is amenable to the application of LLC algorithms.

4. Predictive Coding, Consistency, Curvature

Based on the analysis in Section 3, we identify three desiderata for guiding the selection of the encoder E and latent dynamics model F. We summarize them as follows: (i) *predictive coding* minimizes the predictive suboptimality of the encoder E; (ii) *consistency* of the latent dynamics model F with respective to (p, E) enables planning directly in the latent space; and (iii) *low-curvature* enables planning in latent space specifically using locally-linear controllers. We refer to these heuristics collectively as Predictive Coding-Consistency-Curvature (PC3). PC3 can be thought of as an information-theoretic extension of the Prediction-Consistency-Curvature (PCC) framework described by Levine et al. (2020)—differing primarily in the replacement of explicit next-observation prediction with predictive coding in the latent space.

In this section, we highlight some of the key design choices involved when instantiating PC3 in practice. In particular, we shall show how to leverage the CPC variational mutual information bound in a parameter-efficient manner and how to enforce the consistency of F with respect to (p, E) without destabilizing training.

4.1. Enforcing Predictive Codes

To estimate the mutual information $\ell_{\text{MI}}(E)$, we employ contrastive predictive coding (CPC) proposed by van den Oord et al. (2018). We perform CPC by introducing a critic $f : \mathcal{Z} \times \mathcal{Z} \times \mathcal{U} \to \mathbb{R}$ to construct the lower bound

$$I(E(X_{t+1}) ; E(X_t), U_t) \tag{2}$$
$$\geq \mathbb{E} \frac{1}{K} \sum_i \ln \frac{\exp f(E(x_{t+1}^{(i)}), E(x_t^{(i)}), u_t^{(i)})}{\frac{1}{K} \sum_j \exp f(E(x_{t+1}^{(i)}), E(x_t^{(j)}), u_t^{(j)})},$$

where the expectation is over K i.i.d. samples of (x_{t+1}, x_t, u_t). Notice that the current latent state-action pair $(E(x_t), u_t)$ is specifically designated as the source of negative samples and used for the contrastive prediction of the next latent state $E(x_{t+1})$. We then tie the critic f to our latent dynamics model F,

$$\exp f(z_{t+1}, z_t, u_t) := F(z_{t+1} \mid z_t, u_t).$$

This particular design of the critic has two desirable properties. First, it exploits parameter-sharing to circumvent the instantiation of an auxiliary critic f. Second, it takes advantage of the property that an optimal critic for the lower bound in Eq. (2) is the true latent dynamics (Poole et al., 2019; Ma & Collins, 2018)—which we wish F to approximate. The resulting CPC objective is thus

$$\mathbb{E} \frac{1}{K} \sum_i \ln \frac{F(E(x_{t+1}^{(i)}) \mid E(x_t^{(i)}), u_t^{(i)})}{\frac{1}{K} \sum_j F(E(x_{t+1}^{(i)}) \mid E(x_t^{(j)}), u_t^{(j)})},$$

which we denote as $\ell_{\text{cpc}}(E, F)$.

4.2. Enforcing Consistency

Since the true latent dynamics is an optimal critic for the CPC bound, it is tempting to believe that optimizing (E, F) to maximize $\ell_{\text{cpc}}(E, F)$ should be sufficient to encourage the learning of a latent dynamics model F that is consistent with the true latent dynamics induced by (p, E).

In this section, we show that it is easy to construct a simple counterexample illustrating the non-uniqueness of the true latent dynamics as an optimal critic—and that F may learn to be arbitrarily *inconsistent* with (p, E) while still maximizing $\ell_{\text{cpc}}(E, F)$ under a fixed choice of E. Our simple counterexample proceeds as follows: let E be the identity function, let $\mathcal{X} = \mathcal{U} = \mathbb{R}$, and let $p(x_{t+1}, x_t, u_t)$ be a uniform distribution over the tuples $(1, 1, 1)$ and $(-1, -1, -1)$. Let $F(z_{t+1} \mid z_t, u_t)$ be a conditional Gaussian distribution with learnable variance $\sigma^2 > 0$ and mean function

$$\mu(z_t, u_t) = \text{sign}(z_t) \cdot \eta,$$

where $\eta > 0$ is a learnable parameter. By symmetry, the bound $\ell_{\text{cpc}}(E, F)$ where $K = 2$ becomes

$$\ln \frac{\exp(-(\eta - 1)^2/\sigma^2)}{\exp(-(\eta - 1)^2/\sigma^2) + \exp(-(\eta + 1)^2/\sigma^2)} + \ln 2.$$

In the denominator, the first term arises from the positive sample (e.g., $(1, 1, 1)$) whereas the second term arises from the negative sample (e.g., $(1, -1, -1)$). One way to maximize this bound would be to set $\eta = 1$ and let $\sigma \to 0$. Correspondingly, F would approach the true latent dynamics and precisely predict how (z_t, u_t) transitions to z_{t+1}. However, an alternative procedure for maximizing this bound is to fix σ to any positive constant and let $\eta \to \infty$. In this scenario, F becomes an arbitrarily poor predictor of the underlying latent dynamics.

This counterexample highlights a simple but important characteristic of the CPC bound. In contrast to direct maximum likelihood training of $F(z_{t+1} \mid z_t, u_t)$ using samples of (z_{t+1}, z_t, u_t) from the true latent dynamics, the contrastive predictive training of the latent dynamics model simply ensures that $F(z_{t+1} \mid z_t, u_t)$ assigns a *relatively* much higher value to the positive samples than to the negative samples. The fact that the CPC bound may be maximized without learning a consistent dynamics model F may be why previous work by Nachum et al. (2018) using CPC for representation learning in model-free RL chose not to perform model-based latent space control despite also learning an F as a variational artifact from their CPC bound.

Since our goal is to use F in (SOC2) for optimal control, it is critical that we ensure the latent dynamics model F indeed approximates the true latent dynamics. We therefore additionally train F via the maximum likelihood objective

$$\ell_{\text{cons}}(E, F) = \mathbb{E}_{p(x_{t+1}, x_t, u_t)} \ln F(E(x_{t+1}) \mid E(x_t), u_t).$$

However, naively optimizing (E, F) to maximize both ℓ_{cpc} and ℓ_{cons} is unstable; whereas ℓ_{cpc} is geometry-invariant, ℓ_{cons} is sensitive to non-volume preserving transformations of the latent space (Rezende & Mohamed, 2015; Dinh et al., 2016) and can increase arbitrarily simply by collapsing the latent space. To resolve this issue, we add Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ with fixed variance to the next-state encoding $E(x_{t+1})$. Doing so yields the noise-perturbed objectives $\ell_{\text{cpc}+\epsilon}$ and $\ell_{\text{cons}+\epsilon}$. The introduction of noise has two notable effects. First, it imposes an upper bound on the achievable log-likelihood

$$\ell_{\text{cons}+\epsilon}(E, F) \leq -\frac{n_z}{2} \ln 2\pi e\sigma^2$$

based on the entropy of the Gaussian noise. Second, $\ell_{\text{cpc}+\epsilon}$ is now a lower bound to the mutual information between $(E(X_t), U_t)$ and the noise-perturbed $E(X_{t+1}) + \mathcal{E}$,

$$I(E(X_{t+1}) + \mathcal{E}\,;\, E(X_t), U_t)$$
$$\geq \mathbb{E}\frac{1}{K}\sum_i \ln \frac{F(E(x_{t+1}^{(i)}) + \epsilon^{(i)} \mid E(x_t^{(i)}), u_t^{(i)})}{\frac{1}{K}\sum_j F(E(x_{t+1}^{(i)}) + \epsilon^{(i)} \mid E(x_t^{(j)}), u_t^{(j)})}.$$

Since the noise variance σ^2 is fixed, $\ell_{\text{cpc}+\epsilon}$ can only be maximized by expanding the latent space. By tuning the noise variance σ^2 as a hyperparameter, we can balance the latent space retraction encouraged by $\ell_{\text{cons}+\epsilon}$ with the latent space expansion encouraged by $\ell_{\text{cpc}+\epsilon}$ and thus stabilize the learning of the latent space. For notational simplicity, we shall treat all subsequent mentions of ℓ_{cpc} and ℓ_{cons} to mean their respective noise-perturbed variants, except in the specific ablation conditions where noise is explicitly removed (e.g., the "w/o ϵ" condition in our experiments). At this point, we wish to note that, given the introduction of ℓ_{cons}, a natural question is whether we should in fact parameter-tie the critic in ℓ_{cpc} to the latent dynamics model in ℓ_{cons}. We demonstrate the value of doing so in Appendix B.5.

4.3. Enforcing Low Curvature

We measure the curvature of F by computing the first-order Taylor expansion error incurred when evaluating at $\bar{z} = z + \eta_z$ and $\bar{u} = u + \eta_u$,

$$\ell_{\text{curv}}(F) = \mathbb{E}_{\eta\sim\mathcal{N}(0,\delta I)}[\|f_{\mathcal{Z}}(\bar{z}, \bar{u}) - (\nabla_z f_{\mathcal{Z}}(\bar{z}, \bar{u})\eta_z + \nabla_u f_{\mathcal{Z}}(\bar{z}, \bar{u})\eta_u) - f_{\mathcal{Z}}(z, u)\|_2^2].$$

Levine et al. (2020) further proposes an amortized version of this objective to accelerate training when the latent dimensionality n_z is large. However, since n_z is relatively small in our benchmark tasks, our initial experimentation suggests amortization to have little wall-clock time impact on these tasks. Our overall objective is thus

$$\max_{E,F} \lambda_1\ell_{\text{cpc}}(E, F) + \lambda_2\ell_{\text{cons}}(E, F) - \lambda_3\ell_{\text{curv}}(F),$$

which maximizes the CPC bound and consistency, while minimizing curvature.

5. Experiments

In this section, we report a thorough ablation study on various components of PC3, as well as compare the performance of our proposed model[6] with two state-of-the-art LCE baselines: PCC (Levine et al., 2020) and SOLAR (Zhang et al., 2019).[7] The experiments are based on four image-based control benchmark domains: Planar System, Inverted Pendulum,[8] Cartpole, and 3-Link Manipulator.

Data generation procedure: In PCC and PC3, each sample is a triplet (x_t, u_t, x_{t+1}) where we (1) sample an underlying state s_t and generate its observation x_t, (2) sample an action u_t, and (3) obtain the next state s_{t+1} from the true dynamics and generate its observation x_{t+1}. In SOLAR, each training sample is an episode $\{x_1, u_1, x_2, \ldots, x_T, u_T, x_{T+1}\}$, where T is the control horizon. For fair comparison, we allow SOLAR to collect data uniformly in the state space. See Appendix B.4 for SOLAR performance under its original data generation procedure.

Evaluation metric: We evaluate PC3 and the baselines in terms of control performance. For PC3 and PCC, we apply iLQR algorithm in the latent space with a quadratic cost, $c(z_t, u_t) = (z_t - z_{\text{goal}})^\top Q(z_t - z_{\text{goal}}) + u_t^\top R u_t$, where z_t and z_{goal} are the encoded vectors of the current and goal observation, and $Q = \alpha \cdot I_{n_z}$, $R = \beta \cdot I_{n_u}$. For SOLAR, we use their original local-inference-and-control algorithm.[9] We report the percentage of time spent in the goal region in the underlying system (Levine et al., 2020).

5.1. Ablation Study

We characterize PC3 by ablating ℓ_{cons}, ℓ_{curv}, and the noise ϵ added to z_{t+1}. For each setting, we report the latent map size,[10] ℓ_{cpc}, ℓ_{cons}, ℓ_{curv}, and the control performance. These statistics are averaged over 10 different models. All settings are run on Pendulum (Balance and Swing Up).

Consistency: In Table 1, we can see that when ℓ_{cons} is omitted, the control performance drops. As discussed in Section 3.2, the latent dynamics model F performs poorly when not explicitly optimized for consistency. This is further demonstrated in Table 2, where we take a pretrained PC3 model, freeze the encoder, and retrain F to maximize either ℓ_{cpc} or ℓ_{cons}. Despite both retrained models achieving similar ℓ_{cpc} scores, it is easy to see that training via ℓ_{cpc} results in much worse latent dynamics in terms of prediction.

[6] https://github.com/VinAIResearch/PC3-pytorch

[7] E2C and RCE, two closely related baselines, are not included, since they are often inferior to PCC (Levine et al., 2020).

[8] Pendulum has two separate tasks: Balance and Swing Up

[9] https://github.com/sharadmv/parasol

[10] We add the loss $||\frac{1}{N}\sum_{i=1}^N z_i||_2^2$ to center the latent map at the origin, then report $\frac{1}{N}\sum_{i=1}^N ||z_i||_2^2$ as the latent map size.

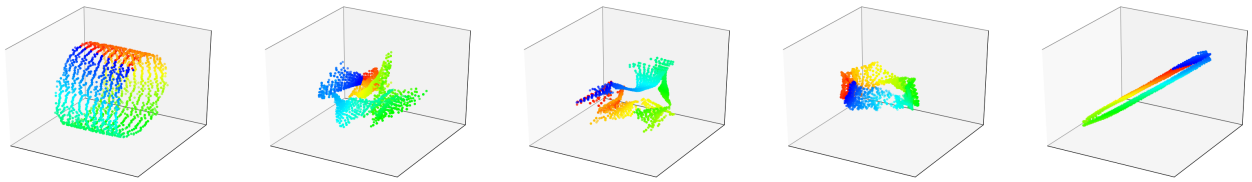

Figure 2. Inverted pendulum representations. From left to right: PC3, w/o $(\ell_{\text{cons}}, \epsilon)$, w/o (ℓ_{cons}), w/o ϵ, w/o ℓ_{curv}

Table 1. Ablation analysis. Percentage of steps spent in goal state. From top to bottom: full-model PC3, excluding consistency and latent noise, excluding consistency, excluding latent noise, excluding curvature. For each setting we report the latent map scale, CPC, consistency and curvature loss, and control performance on balance and swing.

Setting	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Balance	Swing Up
PC3	16.2	4.58	2.13	0.03	$\mathbf{99.12 \pm 0.66}$	$\mathbf{58.4 \pm 3.53}$
w/o $(\ell_{\text{cons}}, \epsilon)$	10.47	5.07	-4.13	0.001	34.55 ± 3.69	17.83 ± 2.9
w/o ℓ_{cons}	101.52	5.03	-4.87	0.0025	31.08 ± 3.57	7.46 ± 1.32
w/o ϵ	0.04	3.27	20.83	0.0009	65.2 ± 1.11	0 ± 0
w/o ℓ_{curv}	66.1	4.8	2.34	0.56	96.89 ± 0.97	21.69 ± 2.73

Table 2. We took a pretrained PC3 model, froze the encoder E, and then retrained only the latent dynamics model F either without ℓ_{cons} (first row) or without ℓ_{cpc} (second row). Note that we continue to use ℓ_{curv} and add ϵ noise in both settings.

Setting	Latent map size	ℓ_{cpc}	ℓ_{cons}	ℓ_{curv}	Balance	Swing Up
Retrain F w/o ℓ_{cons}	16.2	4.57	-21.93	0.02	46.77 ± 3.66	18.06 ± 1.87
Retrain F w/o ℓ_{cpc}	16.2	4.6	2.17	0.03	90.85 ± 2.33	50.11 ± 3.74

Noise: The control performance also decreases when we do not add noise to z_{t+1}. This is because the model will collapse the latent space to inflate ℓ_{cons} as shown in Table 1, leading to a degenerate solution. Adding noise to z_{t+1} prevents the map from collapsing; since the noise variance is fixed, ℓ_{cpc} is only maximized by pushing points apart. Indeed, Table 1 shows that when noise is added but ℓ_{cons} is removed, the latent map expands aggressively.

Curvature: Finally, as previously observed in (Levine et al., 2020), imposing low curvature is an important component if we want to use locally-linear control algorithms such as iLQR. Without the curvature loss, the Taylor approximation when running iLQR might not be accurate, leading to poor control performance. The right-most map in Figure 2 shows a depiction of this setting, in which the map has very sharp regions, requiring the transition function to have high-curvature to move in these regions.

5.2. Control Performance Comparison

Experimental pipeline: For each control domain, we run 10 different subtasks (different initial and/or goal states), and report the average performance among these subtasks. For PCC and PC3, we train 10 different models, and each of them will perform all 10 subtasks (which means a total of $10 \times 10 = 100$ subtasks), and we additionally report the performance of the best model. SOLAR training procedure depends on the specific subtask (i.e., initial and goal state), and since we cannot train 100 different models due to huge computation cost, we train only 1 model for each subtask. All subtasks are shared for three methods.

Result: Table 3 shows that our proposed PC3 model significantly outperforms the baselines by comparing the means and standard error of means on the different control tasks.[11] PCC and SOLAR often fail at difficult tasks such as Swing Up and 3-Link. Moreover, SOLAR training procedure depends on the specific task, which makes them unsuitable to be reused for different tasks in the same environment. Figure 3 demonstrates some (randomly selected) latent maps of Planar and Inverted Pendulum domains learned by PCC and PC3. In general, PC3 produces more interpretable latent representation for Pendulum, due to the fact that next observation prediction is too conservative and may force a model to care about things that do not matter to downstream tasks. Finally, in terms of computation, PC3 enjoys huge improvements over the baselines, with $1.85\times$ faster than PCC and $52.8\times$ faster than SOLAR.

[11] Due to huge computation cost of SOLAR, we lower control horizon for Balance, Swing Up, Cartpole and 3-Link, compared to what was used in the PCC paper.

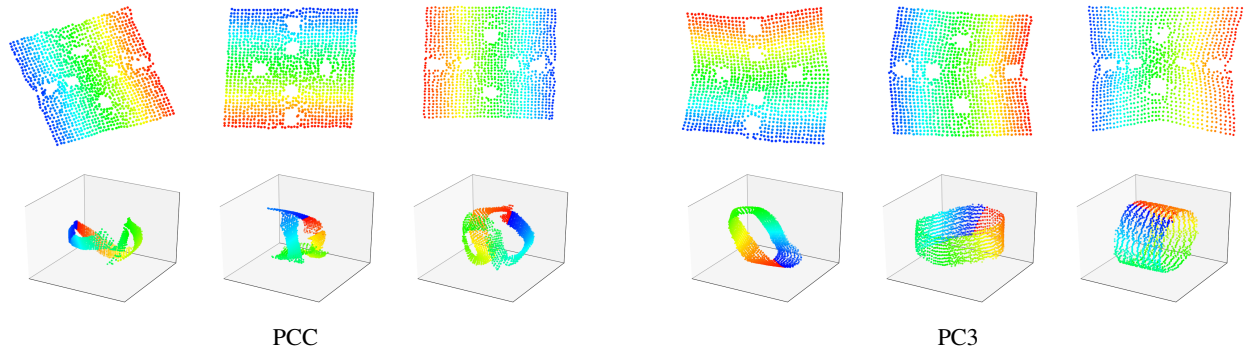

Figure 3. Top: Planar latent representations; Bottom: Inverted Pendulum latent representations. Left three: PCC, right three: PC3.

Table 3. 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	71.25 ± 1.51	$\mathbf{75.5 \pm 0.32}$	$\mathbf{75.5 \pm 0.32}$
Balance	$\mathbf{99.12 \pm 0.66}$	91.9 ± 1.72	55.2 ± 1.1	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$
Swing Up	$\mathbf{58.4 \pm 3.53}$	26.41 ± 2.64	37.78 ± 1.44	$\mathbf{84 \pm 0}$	66.9 ± 3.8
Cartpole	$\mathbf{96.26 \pm 0.95}$	94.44 ± 1.34	74.8 ± 14.95	$\mathbf{97.8 \pm 1.4}$	$\mathbf{97.8 \pm 1.4}$
3-link	$\mathbf{42.4 \pm 3.23}$	14.17 ± 2.2	6 ± 3.79	$\mathbf{78 \pm 1.04}$	45.8 ± 6.4

6. Related Work

LCE Approaches. In contrast to existing LCE methods (Watter et al., 2015; Banijamali et al., 2018; Levine et al., 2020; Zhang et al., 2019; Hafner et al., 2018), our main contribution in PC3 is the development of an information-theoretic approach for minimizing the predictive suboptimality of the encoder and circumventing the need to perform explicit next-observation prediction. In particular, PC3 can be seen as a natural information-theoretic extension of PCC (Levine et al., 2020), which itself extended and improved upon E2C (Watter et al., 2015) and RCE (Banijamali et al., 2018). Compared to SOLAR (Zhang et al., 2019), PC3 (as well as PCC, RCE, and E2C) decouples the representation learning and latent dynamics estimation from control—once the encoder and latent dynamics have been learned for a particular environment, it can be used to solve many SOC problems within the same environment. In contrast, SOLAR is an online algorithm that interleaves model learning with policy optimization. Furthermore, the latent model in SOLAR is restricted to be globally linear, which can potentially impact the control performance.

Information-Theoretic Approaches. Several works have previously explored information-theoretic approaches for representation learning in the reinforcement learning context (Nachum et al., 2018; Anand et al., 2019; Lu et al., 2019). However, these works do not test the quality of their learned representations for the purposes of model-based planning in the latent space, opting instead to leverage the representations for model-free RL. This is particularly notable in the case of Nachum et al. (2018), who explicitly learned both an encoder E and latent dynamics model F. As we showed in Section 4.2, maximizing the CPC bound alone may not be sufficient for ensuring that F is a good predictor of the latent dynamics induced by (p, E). Thus, the resulting (E, F) from predictive coding alone may be unsuitable for multi-step latent planning, as we demonstrate in our ablation analysis in Section 5.1.

7. Conclusion

In this work, we propose a novel information-theoretic Learning Controllable Embedding approach for handling high-dimensional stochastic optimal control. Our approach challenges the necessity of the next-observation prediction in existing LCE algorithms. We show theoretically that predictive coding is a valid alternative to next-observation prediction for learning a representation that minimizes predictive suboptimality. To instantiate information-theoretic LCE, we develop the Predictive Coding-Consistency-Curvature (PC3) model and show that PC3 is a simpler, yet more effective method than existing next-observation prediction-based LCE approaches. We also provide a thorough study on various components of the PC3 objective via ablation analysis to assist the adoption of predictive coding in future LCE research. A natural follow-up would be to study the efficacy of predictive coding when used in conjunction with other techniques in the LCE literature (e.g. latent overshooting) as well as with other controllers beyond the class of locally-linear controllers considered in our present work.

References

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R. D. Unsupervised state representation learning in atari. In *Advances in Neural Information Processing Systems*, pp. 8766–8779, 2019.

Banijamali, E., Shu, R., Ghavamzadeh, M., Bui, H., and Ghodsi, A. Robust locally-linear controllable embedding. In *Proceedings of the Twenty First International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759, 2018.

Belghazi, M., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540, 2018.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is "nearest neighbor" meaningful? In *International conference on database theory*, pp. 217–235. Springer, 1999.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *preprint arXiv:1605.08803*, 2016.

Espiau, B., Chaumette, F., and Rives, P. A new approach to visual servoing in robotics. *ieee Transactions on Robotics and Automation*, 8(3):313–326, 1992.

Finn, C., Tan, X., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519. IEEE, 2016.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Hjelm, R., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *preprint arXiv:1808.06670*, 2018.

Johnson, M., Duvenaud, D., Wiltschko, A., Adams, R., and Datta, S. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pp. 2946–2954, 2016.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. *preprint arXiv:1903.00374*, 2019.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. and Welling, M. Auto-encoding variational bayes. *preprint arXiv:1312.6114*, 2013.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Kurutach, T., Tamar, A., Yang, G., Russell, S. J., and Abbeel, P. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pp. 8733–8744, 2018.

Levine, N., Chow, Y., Shu, R., Li, A., Ghavamzadeh, M., and Bui, H. Prediction, consistency, curvature: Representation learning for locally-linear control. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxG_0EtDS.

Li, W. and Todorov, E. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pp. 222–229, 2004.

Lu, X., Tiomkin, S., and Abbeel, P. Predictive coding for boosting deep reinforcement learning with sparse rewards. *arXiv preprint arXiv:1912.13414*, 2019.

Ma, Z. and Collins, M. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *preprint arXiv:1809.01812*, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with deep reinforcement learning. *Preprint arXiv:1312.5602*, 2013.

Nachum, O., Gu, S., Lee, H., and Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. *preprint arXiv:1810.01257*, 2018.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Petrik, M., Ghavamzadeh, M., and Chow, Y. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, pp. 2298–2306, 2016.

Poole, B., Ozair, S., van den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. *preprint arXiv:1905.06922*, 2019.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. *preprint arXiv:1505.05770*, 2015.

Rezende, D., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *preprint arXiv:1401.4082*, 2014.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *preprint arXiv:1807.03748*, 2018.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured latent representations for model-based reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

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

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

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), $U_{\text{I-E}}^*$:

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

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

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

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 $\ell^* \leq \ell$, 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

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

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