

Controllability-Constrained Deep Network Models for Enhanced Control of Dynamical Systems

Suruchi Sharma¹, Volodymyr Makarenko¹, Gautam Kumar², Stas Tiomkin^{1,*}

Charles W. Davidson College Of Engineering, San Jose State University, CA, USA

¹Computer Engineering Dept., ²Chemical and Materials Engineering Dept.

{suruchi.sharma, volodymyr.makarenko, gautam.kumar, stas.tiomkin}@sjsu.edu

Abstract— Control of a dynamical system without the knowledge of dynamics is an important and challenging task. Modern machine learning approaches, such as deep neural networks (DNNs), allow for the estimation of a dynamics model from control inputs and corresponding state observation outputs. Such data-driven models are often utilized for the derivation of model-based controllers. However, in general, there are no guarantees that a model represented by DNNs will be controllable according to the formal control-theoretical meaning of controllability, which is crucial for the design of effective controllers. This often precludes the use of DNN-estimated models in applications, where formal controllability guarantees are required. In this proof-of-the-concept work, we propose a control-theoretical method that explicitly enhances models estimated from data with controllability. That is achieved by augmenting the model estimation objective with a controllability constraint, which penalizes models with a low degree of controllability. As a result, the models estimated with the proposed controllability constraint allow for the derivation of more efficient controllers, they are interpretable by the control-theoretical quantities and have a lower long-term prediction error. The proposed method provides new insights on the connection between the DNN-based estimation of unknown dynamics and the control-theoretical guarantees of the solution properties. We demonstrate the superiority of the proposed method in two standard classical control systems with state observation given by low resolution high-dimensional images.

I. INTRODUCTION

In recent years, the development of dynamical control models from high-dimensional data, such as images, using deep neural networks (DNNs) has garnered substantial interest within various fields, ranging from robotics to neuroscience [1], [2]. Such data-driven models allow for the design of optimal control strategies for intricate dynamical control systems. However, these data-driven DNN-based models solely rely on the quality of input-output data acquired from a system. So, these data-driven models often lack the desired control-theoretic properties, such as controllability, necessary for effective system control. Consequently, designing optimal controllers using these models can lead to sub-optimal performance. In this paper, we address this issue by explicitly incorporating a controllability metric into the development of latent models derived from high-dimensional imaging data. By using these models within a model predictive control

This work was funded by Charles W. Davidson College Of Engineering (CoE), San Jose State University (SJSU).

* Corresponding Author.

(MPC) framework, we demonstrate significant improvement in the MPC performance on two classical control systems: the inverted pendulum and a cart-pole system.

One of the widely adopted methods for capturing dynamical features in a low-dimensional latent space from high-dimensional data is Variational AutoEncoders (VAEs) [3]. In essence, a VAE comprises an encoder model and a decoder model, both trained on the input data to minimize the reconstruction error between the encoded-decoded data and the input data [3], [4]. Unlike standard autoencoders, which encode input to a single point, VAEs encode inputs as probability distributions across the latent space. VAEs have found applications in various domains, including anomaly detection, image regeneration, data compression [5]–[7].

Notably, VAEs have also emerged as valuable low-dimensional latent models for designing and controlling dynamical systems in recent research endeavors [4], [8]. In this study, state information is represented as images. Then Convolution Neural Networks are used for processing these images and creating latent models. Convolution Neural Networks excel in handling image data, as demonstrated in [9]–[11]. These latent representations are used to acquire an understanding of the dynamics of the environment, executing control tasks similar to the approach in [4], [8], [12], [13]. Diving deeper into this concept reveals that this approach not only helps in predicting short horizons but also longer horizons such as in [14], [15].

None of the above approaches explicitly include controllability constraints in their latent space. We aim to induce controllability constraints in the latent space of the Variational AutoEncoder (VAE) model[3]. The Model Predictive Control techniques [16], [17] are used to evaluate the performance of the models. Since VAEs are trained using the input-output (control-observation) data from the system, the data used for training VAEs may limit the extent to which VAEs exhibit system controllability in response to exogenous inputs. Formally, the VAE's objective [3] is to minimize the prediction error between the next state observation, generated by a dynamics model, given the current state observation and current control action. As we show in this work, this prediction error itself does not guarantee the controllability of a dynamics model, leading to the sub-optimal performance of controllers derived with such a model.

Incorporating a controllability constraint, such as control-

lability Gramian [17], during VAE training could significantly improve the intrinsic controllability of VAEs and lead to the development of efficient control strategies for complex systems. In this work, we conceptualize this idea by developing an approach to enhance the controllability of estimated models by imposing a controllability constraint to the standard VAE objective. In particular, we augment the standard VAE objective for dynamics learning with the *degree of controllability* of the latent dynamics. For our best knowledge, this is the first attempt to explicitly incorporate controllability constraints in designing latent models for dynamical systems estimated from data.

Thus, this work establishes new connections between the state-of-the-art machine learning methods, (VAE), for the dynamics estimation from high-dimensional data (images/Lidar/etc) and the essential control-theoretical properties of dynamical control systems (controllability), leading to more reliable data-driven models and more efficient data-driven controllers. The main contributions of this work are as follows:

- Augmentation of the standard VAE objective for the estimation of unknown dynamics from high-dimensional data with the controllability constraint.
- An efficient algorithm for the optimization of the controllability-constrained VAE objective.
- Demonstration of the effectiveness of the proposed method on the classical dynamical control systems on the tasks of long-term prediction and model-predictive control.

This paper is organized as follows. In Section II, we introduce the notations and necessary definitions for the main components such as state, state observation, action, original dynamics, f , and a dynamics model, f_θ , (latent dynamics), degree of controllability and controllability Gramian, W , and feature extractor and image generator, (encoder, h_ϕ and decoder, g_ψ , respectively). Also, we overview the standard objective for learning a dynamics model from high-dimensional state observation in the formalism of VAE. In Section III we formulate a novel controllability-constrained objective for VAE, which comprises a prediction error term and a penalty term for a reduced degree of controllability of an estimated dynamics model. In Section IV we demonstrate the advantage of the controllability-augmented dynamics models in comparison to the standard (baseline) model. Finally, in Section V we conclude the current work and discuss future research directions and eventual applications. An overview of the method is provided at Fig. 1.

II. PROBLEM FORMULATION

A. Original and Latent Dynamics

Consider a dynamical control system in discrete time, f :

$$f : x_t \times a_t \rightarrow x_{t+1} \quad (\text{Original Dynamics}) \quad (1)$$

with the state and action at time t , denoted by $x_t \in \mathcal{X}$ and $a_t \in \mathcal{A}$, respectively.

We assume the original dynamics function, f , is unknown, and the state, x_t , is partially observed through the state

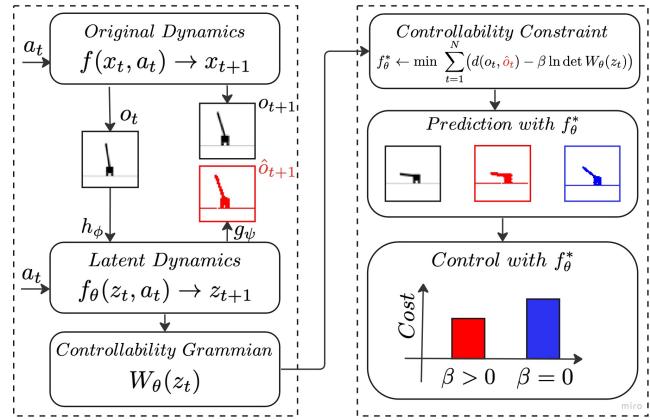


Fig. 1. Method overview: the current and next time states, x_t and x_{t+1} , in the original dynamics are rendered to the corresponding state-observations, given by low resolution images, o_t and o_{t+1} , respectively. The former, o_t , is encoded to the latent state z_t by the encoder, h_ϕ . The next latent state, z_{t+1} , is predicted by the parametrized latent dynamics, f_θ , and the corresponding predicted image, \hat{o}_{t+1} , is generated by the decoder, g_ψ . The prediction error between the decoded and original state-observations, \hat{o}_{t+1} and o_{t+1} , is minimized as in the standard VAE framework. We enhance the standard VAE framework with the controllability-constraint, which explicitly penalizes latent dynamics with low degree-of-controllability. The resulted controllability-constrained latent dynamics (in red) is superior to the standard baseline models (in blue) both in long-term prediction and control tasks. The former is visualized by a larger deviation of the standard model (in blue) from the original dynamics (in black), and the latter is depicted by a large control cost in the original dynamics driven by a controller derived with the standard model for $\beta = 0$ in comparison to a controller derived with the controllability-constrained model for $\beta > 0$.

observation, $o_t \in \mathcal{O}$. The latter is given by a signal from a particular sensor, such as a camera, lidar, proprioceptive sensor, etc. The original dynamics can be examined by input-output triples, $\{o_t, a_t, o_{t+1}\}_{t=1}^N$, which are available or can be collected.

The goal of this work is to improve the quality of dynamics models estimated from data, f_θ , denoted by ‘latent dynamics’ in the sequel, and the efficiency of controllers, π , derived with latent dynamics models.

The latent dynamics is described by the following mapping:

$$f_\theta : z_t \times a_t \rightarrow z_{t+1} \quad (\text{Latent Dynamics}) \quad (2)$$

where $z_t \in \mathcal{Z}$ is the latent state at time t , and $\theta \in \Theta$ is a model parameterizing dynamics in the model class, Θ .

A model, θ , of latent dynamics, f_θ , is estimated from data $\mathcal{D} = \{o_t, a_t, o_{t+1}\}_{t=1}^N$, by standard techniques [4], [8], as overviewed below. In this work we assume that $\forall t : o_t$ is the state observation given by a low-resolution image, rendered from the original state, x_t .

B. Estimation of Latent Dynamics

Deep Neural Networks have been proven as a powerful model class, Θ , for the representation of latent dynamics in

this setting with an appropriate training objective [14], [15]:

$$\ell(\theta, \phi, \psi) = \frac{1}{N} \sum_{t=1}^N d(o_{t+1}, \hat{o}_{t+1}) \quad (3)$$

with $\hat{o}_{t+1} = g_\psi \left(\underbrace{f_\theta(h_\phi(o_t), a_t)}_{z_{t+1}} \right)$,

where N is the number of training input-output pairs, h_ϕ and g_ψ are trainable mappings to and from the latent space, respectively:

$$h_\phi : o_t \rightarrow z_t \quad (\text{Encoder}) \quad (4)$$

$$g_\psi : z_t \rightarrow o_t \quad (\text{Decoder}) \quad (5)$$

and $d(\cdot, \cdot)$ is a given similar function between the original next state observation, $o_t \in \mathcal{O}$ and the reconstructed one, $\hat{o}_t \in \mathcal{O}$:

$$d : o_t \times \hat{o}_t \rightarrow R^+. \quad (6)$$

In our experiments, we used the commonly-used cross-entropy similarity between the original image, o_t , and the reconstructed image, \hat{o}_t , of the original dynamics. There exist various architectures and training methods for the derivation of f_θ , h_ϕ , and g_ψ , [14], [15].

Our method, (Section III) applies to arbitrary architectures and training methods, but in the experiments, we chose a particular latent dynamics model, 'Embed-To-Control', [4] due to its simplicity and transparency.

An optimal solution to the problem in Eq.(3) is derived by solving the following optimization problem [4], [8]

$$\theta^*, \phi^*, \psi^* = \underset{\theta, \phi, \psi}{\operatorname{argmin}} \ell(\theta, \phi, \psi; \mathcal{D}), \quad (7)$$

where $\mathcal{D} = \{o_t, a_t, o_{t+1}\}_{t=1}^N$ is the training data. There exist efficient techniques to solve the optimization problem in Eq.(7) with high-dimensional observation and latent spaces [16], [18]. In particular we used the 'ADAM' optimizer [19].

A solution to Eq.(7) does not guarantee, in general, controllability in estimated latent dynamics. In the next section, we introduce a controllability constraint to the optimization problem in Eq. (7), using *degree of controllability* as a metric [17].

C. Degree of Controllability

In linear dynamics, $x_{t+1} = A_{n \times n}x_t + B_{n \times m}a_t$, controllability is given by a binary test, where the rank of the controllability matrix [17], $C = [B, AB, A^2B, \dots, A^{n-1}B]$, is equal to the state dimension, n , in a controllable system, and it is less than n in an uncontrollable system. A more flexible controllability criterion is the degree of controllability, which is identified by the eigendecomposition of the controllability Gramian, $W_{n \times n} = C \cdot C^\top$ [17]¹ More controllable directions, (eigenvectors of W), correspond to larger eigenvalues of W [17], and the total degree of controllability can be identified by

¹The transpose is denoted by T , e.g., M^\top is the transpose of M .

the controllability volume given by the determinant of the controllability Gramian.

In non-linear dynamics, $x_{t+1} = f(x_t, a_t)$, the local degree of controllability can be estimated by the Gramian of locally-linearized dynamics around the state, x , which gives the local degree of controllability by the eigendecomposition of $W(x) = C(x) \cdot C(x)^\top$. We use this observation in the development of the differentiable controllability constraint in the next section.

III. PROPOSED METHOD FOR CONTROLLABILITY-ENHANCED LATENT DYNAMICS

In this section we propose a novel method to explicitly induce controllability into latent dynamics, f_θ , by augmenting the loss in Eq.(3) with the controllability Gramian of locally-linearized latent dynamics around the latent state, z :

$$W_\theta(z) = C_\theta(z)C_\theta(z)^\top \quad (\text{Controllability Gramian}) \quad (8)$$

$$C_\theta(z) = [B_\theta(z), A_\theta(z)B_\theta(z), A_\theta^2(z)B_\theta(z), \dots, A_\theta^{n-1}(z)B_\theta(z)]$$

where $A_\theta(z) = \nabla_z f_\theta(z)$ and $B_\theta(z) = \nabla_a f_\theta(z)$ are the Jacobians of the latent dynamics, f_θ , with regard to the latent state, z , and real action, a . And, $C_\theta(z)$ is the corresponding parametrized controllability matrix [17].

In general, the pair $(A_\theta(z), B_\theta(z))$ is not necessarily controllable, which may deteriorate the quality of control. In the next section, we augment the standard VAE-type loss for data-driven estimations of dynamics with the parametrized controllability Gramian, which explicitly penalized for a low degree of controllability.

A. Controllability-Enhanced Loss

The parametrized Gramian, Eq. (8), in the latent space, $W_\theta(z)$, allows us to formulate the *controllability-enhanced loss*, $\hat{\ell}$, for learning latent dynamics, f_θ , as follows:

$$\hat{\ell}(\theta, \phi, \psi, \beta; \mathcal{D}) = \ell(\theta, \phi, \psi; \mathcal{D}) - \beta \ell_\omega(\theta; \mathcal{D}). \quad (9)$$

The second term in Eq.(9) explicitly increases the total degree of controllability, (the eigenvalues of W_θ), evaluated by the determinant of the controllability Gramian [20]

$$\ell_\omega(\theta; \mathcal{D}) \doteq \frac{1}{N} \sum_{t=1}^N \ln \det W_\theta(z_t) \quad (\text{Constraint}) \quad (10)$$

The interpretation of this constraint is the time-average controllability-volume of the linearized latent dynamics. Importantly, this constraint is differentiable, which makes it suitable for gradient-based optimization [19], [21].

The solution to Eq.(9) can be derived by the same methods as in Eq. (7), with the essential difference that now f_θ has a desired degree of controllability set by β .

$$\theta_W^*(\beta), \phi^*(\beta), \psi^*(\beta) = \underset{\theta, \phi, \psi}{\operatorname{argmin}} \hat{\ell}(\theta, \phi, \psi, \beta; \mathcal{D}). \quad (11)$$

Notably, in Eq. (11), the optimal parameters for latent dynamics, encoder, and decoder, are parametrized by β , which produces a family of solutions, as shown in Fig. 3, and explained in the next sections.

Pseudo-Code: Summary of the Proposed Method

```

1: Stage I: Training controllability-enhanced models
Require:  $\mathcal{D} = \{o_t, a_t, o_{t+1}\}_{t=1}^N$  - training data;  $\beta$  - value
   of the trade-off parameter;  $\theta$ ,  $\phi$  and  $\psi$  - initial DNN-
   parameters for the dynamics model, the encoder, and the
   decoder, respectively.
2: repeat
3:    $\{\theta, \phi, \psi\} \leftarrow \{\theta, \phi, \psi\} - \nabla \hat{l}(\theta, \phi, \psi; \mathcal{D})$   $\triangleright$  Eq. (9)
4: until convergence
5: return optimal  $f_{\theta^*}, h_{\phi^*}, g_{\psi^*}$ .
6: Stage II: Derivation of the optimal controller
Require:  $f_{\theta^*}, h_{\phi^*}, x, x_g, H$ 
7: // Encoding of original initial and target states
8:  $z = h_{\phi^*}(\text{RenderImage}(x))$   $\triangleright$  initial latent state
9:  $z_g = h_{\phi^*}(\text{RenderImage}(x_g))$   $\triangleright$  target latent state
10: // Main Loop
11: repeat
12:    $a^* = \text{MPC}(\text{COST}, f_{\theta^*}, z, H)$   $\triangleright$  optimal action
13:    $x = f(x, a^*)$   $\triangleright$  original dynamics
14:    $z = f_{\theta^*}(z, a^*)$   $\triangleright$  latent dynamics
15: until  $x = x_g$ 
16: return optimal control trajectory  $\{a_1^*, a_2^*, \dots, a_k^*, \dots\}$ 
17: /* Utility functions: MPC and COST */
18: function MPC(COST,  $f_{\theta^*}$ ,  $z$ ,  $H$ )
19:    $\vec{a} \doteq \{a_1, a_2, \dots, a_{H-1}\} = \text{RandomActionSequence}$ 
20:    $\vec{a}^* = \text{minimum COST}(\vec{a}, f_{\theta^*}, z, H)$ 
21:   return  $\vec{a}_1^*$   $\triangleright$  Return the first action
22: end function // MPC
23: //Inputs: latent state, action sequence, planning horizon
24: function COST( $\{a_1, a_2, \dots, a_{H-1}\}, f_{\theta^*}, z, H$ )
25:    $C = 0$ 
26:   for  $k \in (1, \dots, H-1)$  do
27:      $e = z_g - z$   $\triangleright$  latent space error
28:      $C = C + e^\top \cdot e + a_k^\top \cdot a_k$   $\triangleright$  accumulated cost
29:      $z = f_{\theta^*}(z, a^*)$   $\triangleright$  latent dynamics update
30:   end for
31:   return  $C$ 
32: end function // COST

```

The derivation of the optimal value of β can be done by dual optimization (cf., Eq. (9) in [22] or in general [23]), which we defer to future work (cf., Section V).

B. Controllability-Prediction Trade-off

The trade-off between the degree of controllability of latent dynamics, $z_{t+1} = f_{\theta}(z_t, a_t)$, and the quality of prediction of o_{t+1} from $z_{t+1} = f_{\theta}(h_{\phi}(o_t), a_t)$ is controlled by a non-negative parameter, $\beta \geq 0$. This trade-off characterizes the relative importance between the first term in Eq. (9) and its second term. The former controls the overall prediction quality of the next image, o_{t+1} , from the current image, o_t , and the current action, a_t . The latter directly affects the degree of controllability of the parametrized latent dynamics, f_{θ} .

This way the optimal latent dynamics, $\theta_W^*(\beta)$, in Eq. (11), both possess a desired controllability degree and represents the mapping $\forall t : \{o_t \times a_t \rightarrow o_{t+1}\}$.

This trade-off produces a family of solutions, $f_{\theta^*(\beta)}$ parameterized by β , where the original solution is given by $\beta = 0$. This additional degree of freedom in the optimization problem allows for better solutions with a particular property of interest in this work - controllable latent dynamics. We demonstrate this by numerical simulations of standard control benchmark systems in both tasks: planning with latent dynamics and deriving a more efficient controller for original dynamics.

A typical trade-off, (family of solutions), is demonstrated in Fig. (3), where the solution without the controllability constraint in Eq. (10) appears for $\beta = 0$, and there is 9% of the decrease in the Model-Predictive-Control cost for $\beta = 0.5$. For larger β values the MPC cost increases as expected because the first term in Eq. (10) does not succeed in estimating the dynamics model. The MPC cost is estimated on control of the original dynamics, $x = f(x, a)$, while the optimal control actions are derived with the latent dynamics, $z = f_{\theta}(z, a)$. The full experiment settings are explained in the next section.

C. Pseudo Code of the Method

The proposed method is summarized by the **Pseudo-Code** above, wherein the ‘Stage I’ a model, f_{θ} , is estimated from data with the controllability-constrained objective (cf., Eq. (9)), while in ‘Stage II’, this model is used for the derivation of the optimal controller to control an original dynamics, f , Eq. (1). In both cases, state observations are given by low-resolution images, rendered from the original state.

IV. EXPERIMENTS

In this section, we demonstrate the method represented in Section III on two classical control benchmarks, used for the evaluation of new algorithms for image-based estimation of dynamics and control. We evaluate the controllability-enhanced models, f_{θ}^* , in two important tasks: long-term planning and model predictive control with the estimated model².

This section is organized as follows. Firstly, we explain the data collection and the evaluation metrics, then we provide the numerical simulations for ‘Long-Term Planning’ and ‘Control’ in two different dynamics: Inverted Pendulum and Cart Pole.

1) *Data Collection:* Data for training controllability-constrained models, f_{θ} , is collected with the Open AI simulator for the classical control environments [13], [24], [25]: ‘Inverted Pendulum’ and ‘Cart Pole’.

In data collection, the action a_t is applied to the simulator of an environment, denoted by ‘Step’, which produces (‘Renders’) an image of the resulting state, o_{t+1} :

$$\forall t \in [0, \dots, N] : o_{t+1} = \text{RenderImage}(\text{Step}(a_t)). \quad (12)$$

This way the training data, \mathcal{D} , comprises of action and state-observation trajectories, $\mathcal{D} = \{a_t, o_t\}_{t=0}^N$.

²All the experiments and results can be reproduced by our code repository: <https://github.com/suruchi1997/ControlledVAE.git>

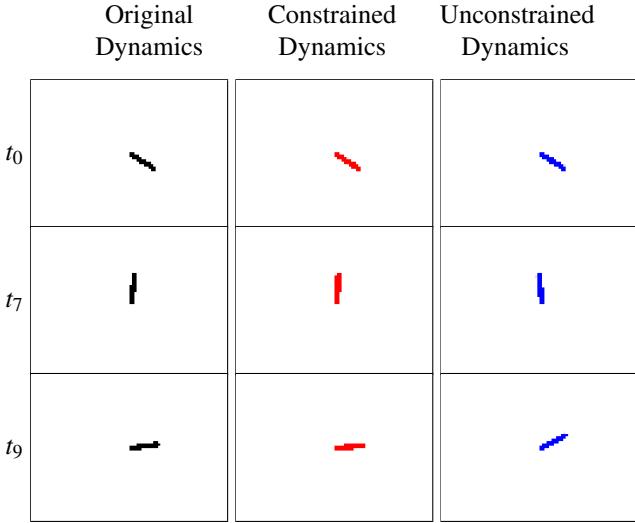


Fig. 2. Qualitative comparison between the long-term prediction by the latent dynamics, f_θ , with the controllability-constraint (the 2nd column in red) and without it (the 3rd column blue). The first column represents the image trajectory by the original dynamics. The pivot point is at the center of the image. The pendulum starts at $\theta(t_0) = \frac{\pi}{4}$ rad from the bottom and $\dot{\theta}(t_0) = 0$ rad s $^{-1}$. The rows correspond to the time steps shown to the left. Visible changes appear after 7 time steps. And, as shown at the bottom row, after 9 time steps, there is a significant deviation of the latent dynamics trajectory for $\beta = 0$ (blue) from the original dynamics trajectory (black); while the latent dynamics trajectory for $\beta = 0.5$ (red) closely follows the original dynamics. The deviation increases in further steps.

2) *Training*: The training procedure is based on the standard stochastic gradient descent (SGD) optimization [19], summarized by ‘Stage I’ in the **Pseudo-Code**.

In the experiments, the encoder, h_ϕ , the latent dynamics, f_θ , and the decoder, g_ψ , are implemented by a similar neural networks’ architecture as in the baseline system [4], which is a SOTA in the field. The architecture details are provided in Appendix V for completeness, and the exact implementations are accessible at the link. Remarkably, our method requires only a small modification, (addition of the controllability constraint), to baseline systems, and can be eventually applied to arbitrary systems.

3) *Evaluation Metrics*: To examine the effectiveness of the controllability constraint Eq. (10) on the quality of the latent dynamics, f_θ , we used two metrics:

-*Qualitative comparison* (cf., Fig. 2 and Fig. 4) between state observation trajectories generated by the ground truth dynamics in Eq. (1) and by the trained latent dynamics, f_θ , for different values of β .

-*Quantitative comparison* (cf., Fig 3 and in the text below) between control cost by a controller derived with the latent dynamics, f_θ^* , trained with the controllability-constraint, $\beta > 0$ and without it $\beta = 0$. The MPC controller [16], [26]

$$\pi : z_t \rightarrow a_t \quad (13)$$

is derived with the quadratic cost defined in the latent space, \mathcal{Z} , as summarized by ‘Stage II’ in the **Pseudo-Code** (cf., lines 27-29). The optimal actions, $a_t^* = \pi(z_t)$, are applied to

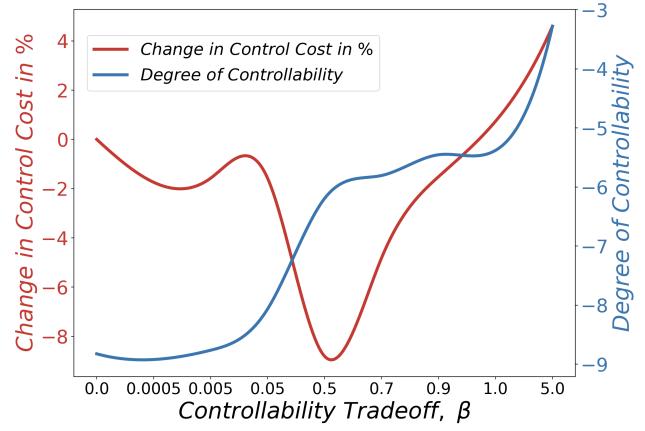


Fig. 3. Relative Change, in the MPC Cost (red) and Degree of Controllability (blue). The left y-axis represents the Percentage change in the MPC cost for $\beta \in [0, \dots, 5.0]$. The MPC cost without controllability constraint is shown for $\beta = 0$. There is 9% decrease in the MPC cost for $\beta \approx 0.5$. The right y-axis shows the minimal eigenvalue of the controllability Gramian, $\ln \lambda_{\min}(W(\beta))$, for the range of β values, which reflects the degree-of-controllability. Each point on the plot represents the average over 20 randomly initialized models, where each model is evaluated on 300 trajectories with 100 steps, starting from a random initial state. There are 3 orders of magnitude in the increase of the degree of controllability between $\beta = 0$ and $\beta \approx 0.5$, which corresponds to the improvement in the efficiency of the MPC controller. As expected, for large values of β , the model predictability term in Eq. (10) is less effective, resulting in a poor dynamical model for fitting the data.

the original dynamics,

$$x_{t+1}^* = f(x_t^*, a_t^*) \quad (14)$$

and the quadratic cost for the optimal action sequence $\{a_1^*, a_2^*, \dots, a_k^*, \dots\}$ is calculated in the original space \mathcal{X} . These evaluation metrics address the two important properties of dynamics: ‘Long-Term Planning’ and ‘Control’, respectively.

A. Inverted Pendulum

1) *Original Dynamics*: Swing up and stabilization of the inverted pendulum is one of the classic problems in control theory [4], [13], [27], [28]. The goal of this experiment is to examine whether the controllability-constrained dynamics reflects better the intrinsic properties (long-term prediction and controllability) of the original dynamics.

In this experiment, we train the latent dynamics with low-resolution images, representing state observation, o_t , and real actions, a_t . This task is challenging because the model f_θ should learn a dynamics model from low-resolution high-dimensional images.

The ‘Inverted Pendulum’ environment is schematically represented in the figure to the left. It comprised of a pivot point (the motor), where the limited control action (torque), $|a_t| < 2(\text{N/m})$, can be applied, and a pole with a point mass at the end.

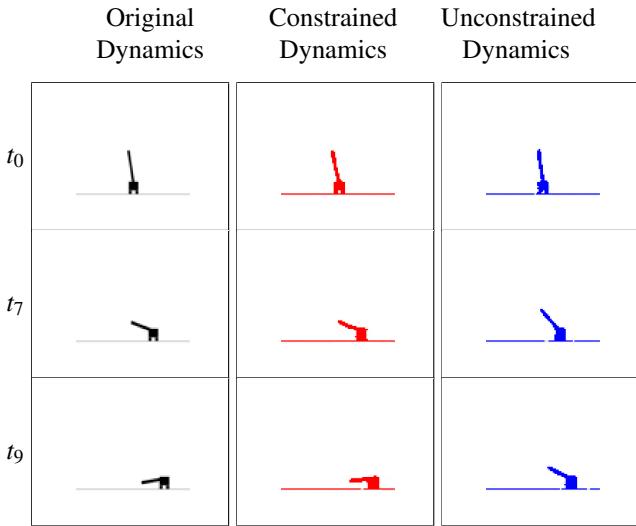


Fig. 4. Qualitative comparison in Cart Pole. The column and row meaning is the same as in Figure (2). The pole starts at $\theta(t_0) = \frac{\pi}{10}$ rad from the top and $\dot{\theta}(t_0) = 0 \text{ rad s}^{-1}$. In this more complicated environment, there is a significant deviation of the baseline solution (blue) from the original dynamics (black) after 7 time steps, while the proposed method (red) closely follows the original dynamics in further steps as well.

The state space of this system is two-dimensional, consisting of angle, $\theta_1 = \theta(\text{rad})$, and angular velocity, $\theta_2 = \dot{\theta}(\text{rad s}^{-1})$. The dynamics of the pendulum is given by:

$$\begin{aligned}\dot{\theta}_1(t) &= \theta_2(t) \\ \dot{\theta}_2(t) &= \frac{g}{l} \sin(\theta_1(t)) + \frac{a(t)}{ml^2}\end{aligned}\quad (15)$$

where $g = 9.8 \text{ m/s}^2$ is gravity, $m = 1 \text{ kg}$ is the mass of the point mass, and $l = 1 \text{ m}$ is the length of the pole.

2) *State Observation*: The dynamics in Eq. (15) is simulated in discrete time with $dt = 0.1 \text{ s}$. The original state, x_t , in the time t is rendered (cf., Eq. (12)) to a corresponding low-resolution image with dimension 48×48 pixels, which represents the state observation o_t , as shown at the top left image in Fig. 2

The qualitative comparison is shown in Fig. 2. The original dynamics (black), the controllability-constrained latent dynamics (red), and the baseline latent dynamics (blue) [4] start from the same initial state. However, in the three last rows, corresponding to the time steps t_7, t_8 and t_9 , it is visible that the controllability-constrained dynamics better predict future states in comparison to the baseline dynamics. Additional details appear in the caption of Fig. 2.

To validate the effectiveness of the proposed method for control, we derived an MPC controller in Eq. (13) with the controllability-constrained latent dynamics for $\beta \in [0, \dots, 5]$. The change in the MPC cost for different values of β is shown in Fig 3. Control cost in the original dynamics by the controller derived with baseline latent dynamics for $\beta = 0$ is the reference. The MPC cost is improved by $\sim 9\%$ by the controller derived with the controllability-constrained latent dynamics for $\beta \approx 0.5$. As expected, the MPC cost increases for larger values of β , because the controllability-constraint becomes dominant in the training objective Eq.

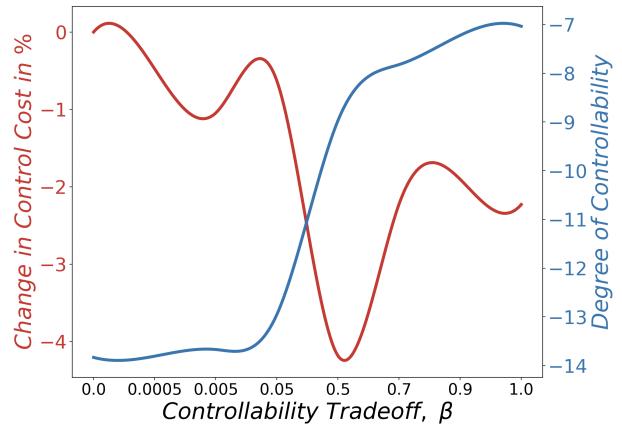


Fig. 5. Relative Change, in the MPC Cost (red) and Degree of Controllability (blue) for 'CartPole'. The meaning of the axes and the evaluation conditions are the same as at Fig. 3. We observe a similar behaviour, where 'Control Cost' with the controllability-enhanced dynamics, $\beta > 0$, is lower in comparison to the baseline dynamics, $\beta = 0$. Noteably, the same number of samples is used for learning dynamics in both cases.

(9), which deteriorates the prediction quality of the latent dynamics. This new *controllability-prediction trade-off* shows that estimated models can be enhanced by explicitly augmenting them with the controllability feature. This trade-off connects machine learning methods for sample-based model estimation with the fundamental concepts in optimal control theory.

B. Cart Pole

To validate the effectiveness of the proposed method in a more complicated system, we conducted the same experiment with the 'Cart Pole' dynamics.

1) *Original Dynamics*: Balancing the pole over the cart is another classic problem in control theory.

The cart pole system consists of a pole attached to a cart that moves over the frictionless track[29]. The state space dimensionality is four. The components of state space include, cart position, x , cart linear velocity, \dot{x} , pole-angle, θ , and pole angular velocity, $\dot{\theta}$.

The action space, $\vec{a}(t) \in \mathcal{A}$, is discrete, comprising of two force values $+1\text{N}$ and -1N . The original dynamics is given by

$$\begin{aligned}\ddot{x}(t) &= \frac{m \sin \theta(t) (\ell \dot{\theta}^2(t) + g \cos \theta(t)) + a(t)}{M + m \sin^2 \theta(t)}, \\ \ddot{\theta}(t) &= -a(t) \cos \theta(t) - \ell \dot{\theta}^2(t) \cos \theta(t) \sin \theta(t) \\ &\quad - (M + m) g \sin \theta(t),\end{aligned}\quad (16)$$

where $x(t), \theta(t), m, M, \ell, g, |a(t)| \leq 1$ are the x coordinate of the center of mass of the cart, the angle of the pole, the pole mass, the cart mass, the pole length, the free-fall acceleration, and the force applied to the cart.

2) *State Observation*: In the same procedure as in the Inverted Pendulum, the dynamics in Eq. (16) is simulated in discrete time, and the state x_t at the time, t , is rendered to a low-resolution image with dimension 80×80 , shown at the top left image in Fig. 4.

Similarly to the ‘Inverted Pendulum’ environment, the performance is evaluated on both long-term prediction and control tasks. Fig. 4 shows the qualitative comparison between the future state predictions by the latent dynamics and the original dynamics. The black images, in the 1st column, represent the image trajectory by the original dynamics. The red images, in the 2nd column, show the predictions by the controllability-constrained models, and the blue images, in the 3rd column, show the predictions by the baseline models. All three dynamics start from the same initial conditions, the 1st row. The images generated by the controllability-constrained model (red) and the baseline model (blue), are similar at the first two steps. In the further steps, it is visible that controllability-constrained models accurately follow the original dynamics, while the baseline models (trained without the controllability constraint) deviate from the original dynamics. This deviation increases in further steps.

The control cost in the original dynamics driven by the controller derived with the controllability-constrained latent dynamics for $\beta = 0.5$ is lower than that derived with the baseline latent model for $\beta = 0.0$.

V. CONCLUSION AND FUTURE WORK

In this work, we conceptualize the idea of the enhancement of data-driven models with the controllability property. The implementation of this idea boils down to adding the controllability constraint to the standard VAE objective for learning a dynamics model from data. The resulting controllability-enhanced VAE objective has the same number of parameters to optimize as in the original VAE. That is because the constraint, (the degree of controllability, expressed via the controllability Grammian), is given with the same parameters as in the original unconstrained model. We discovered the controllability-prediction tradeoff, which is regulated by a scalar parameter β . This proof-of-the-concept study shows that data-driven models may have a small prediction error, which is not enough to guarantee the controllability of the model. The main observation and conclusion are that there exists a positive value of β , when the performance of a derived controller with a learned model is superior to that for $\beta = 0$, as shown in Figure (3).

We defer the optimization of the trade-off parameter β to future work, where the optimal value of β^* will be explored with the dual optimization techniques. Other important venues for future work are i) a rigorous characterization of the solution properties and guarantees, ii) implications of the proposed method in the case of uncontrollable dynamics (or isolated states in dynamics).

A full-fledged method for the derivation of data-driven dynamical models with control-theoretic guarantees for controllability will allow to design of efficient model-based

controllers for various critical mission applications such as, neuro-stimulation feedback, robotic surgery, etc.

ACKNOWLEDGMENT

The authors thank the graduate students at the CI² Lab, CoE, SJSU, Tristan Shah and Himaja Papala for proofreading the manuscript. ST is supported in part by the NSF, grant number: 2246221.

REFERENCES

- [1] D. D. Cox and T. Dean, “Neural networks and neuroscience-inspired computer vision,” *Current Biology*, vol. 24, no. 18, R921–R929, 2014.
- [2] H. A. Pierson and M. S. Gashler, “Deep learning in robotics: A review of recent research,” *Advanced Robotics*, vol. 31, no. 16, pp. 821–835, 2017.
- [3] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.
- [4] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, “Embed to control: A locally linear latent dynamics model for control from raw images,” *Advances in neural information processing systems*, vol. 28, 2015.
- [5] K. Han, H. Wen, J. Shi, K.-H. Lu, Y. Zhang, D. Fu, and Z. Liu, “Variational autoencoder: An unsupervised model for encoding and decoding fmri activity in visual cortex,” *NeuroImage*, vol. 198, pp. 125–136, 2019.
- [6] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, “Variational autoencoder for low bit-rate image compression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2617–2620.
- [7] L. D. Chamain, S. Qi, and Z. Ding, “End-to-end image classification and compression with variational autoencoders,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21916–21931, 2022.
- [8] E. Banijamali, R. Shu, H. Bui, A. Ghodsi, *et al.*, “Robust locally-linear controllable embedding,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 1751–1759.
- [9] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, “Learning to generate chairs with convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1538–1546.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [11] L. Hui and S. Yu-Jie, “Research on face recognition algorithm based on improved convolution neural network,” in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, 2018, pp. 2802–2805.

[12] M. Khalid and S. Omatu, “A neural network based control scheme with an adaptive neural model reference structure,” in *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, IEEE, 1991, pp. 2128–2133.

[13] N. Levine, Y. Chow, R. Shu, A. Li, M. Ghavamzadeh, and H. Bui, “Prediction, consistency, curvature: Representation learning for locally-linear control,” *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[14] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, “Dream to control: Learning behaviors by latent imagination,” *Proceedings of International Conference on Learning Representations*, 2020.

[15] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, “Learning latent dynamics for planning from pixels,” in *International conference on machine learning*, PMLR, 2019, pp. 2555–2565.

[16] Y. Tassa, T. Erez, and W. Smart, “Receding horizon differential dynamic programming,” *Advances in neural information processing systems*, vol. 20, 2007.

[17] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[18] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in *First International Conference on Informatics in Control, Automation and Robotics*, SciTePress, vol. 2, 2004, pp. 222–229.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[20] T. Katayama *et al.*, *Subspace methods for system identification*. Springer, 2005, vol. 1.

[21] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, “Gradient-based optimizer: A new metaheuristic optimization algorithm,” *Information Sciences*, vol. 540, pp. 131–159, 2020.

[22] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, “Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow,” *International Conference on Learning Representations (ICLR)*, 2019.

[23] Y. Nandwani, A. Pathak, and P. Singla, “A primal dual formulation for deep learning with constraints,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[25] J. Arroyo, C. Manna, F. Spiessens, and L. Helsen, “An open-ai gym environment for the building optimization testing (boptest) framework,” in *Building Simulation 2021*, IBPSA, vol. 17, 2021, pp. 175–182.

[26] J. Rawlings, “Tutorial overview of model predictive control,” *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000. DOI: 10.1109/37.845037.

[27] N. Muskinja and B. Tovornik, “Swinging up and stabilization of a real inverted pendulum,” *IEEE transactions on industrial electronics*, vol. 53, no. 2, pp. 631–639, 2006.

[28] M. Bugeja, “Non-linear swing-up and stabilizing control of an inverted pendulum system,” in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, IEEE, vol. 2, 2003, pp. 437–441.

[29] A. G. Barto, R. S. Sutton, and C. W. Anderson, “Neuronlike adaptive elements that can solve difficult learning control problems,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 834–846, 1983. DOI: 10.1109/SMC.1983.6313077.

APPENDIX

A. Network Architectures and Hyperparameters

1) *Pendulum*: Input: Two 48×48 images, 15000 training samples of the form (o_t, a_t, o_{t+1}) , Action Space: 1-dimensional, Latent Space: 2-dimensional, Encoder: 4 layers: Convolution: $16 \times 3 \times 3$; stride (2,2) - Convolution: $32 \times 3 \times 3$; stride (2, 2) - 256 units - 4 units (2 for mean and 2 for variance), Decoder: 5 layers: 256 units - 4608 units - Convolution Transpose: $16 \times 3 \times 3$; stride (2, 2) - Convolution Transpose: $2 \times 3 \times 3$; stride (2, 2) - Sigmoid Layer, Transition Dynamics: 3 layers: 100 units - 100 units - 8 units, MPC Prediction Horizon (H): 10, Control Horizon: 10, Number of Models: 20, Learning Rate: 0.0003, Optimizer: ‘ADAM’ [19], Cost Matrices: $Q = I_{n \times n}$, $R = 0.01$.

2) *CartPole*: The architecture is inspired by. [13] Input: Two 80×80 images. 15000 training samples of the form (o_t, a_t, o_{t+1}) , Action Space: 1-dimensional, Latent Space: 4-dimensional, Encoder: 6 layers: Convolution: $32 \times 5 \times 5$; stride (1,1) - Convolution: $32 \times 5 \times 5$; stride (2, 2)-Convolution: $32 \times 5 \times 5$; stride (2, 2)-Convolution: $10 \times 5 \times 5$; stride (2, 2) - 200 units - 8 units (4 for mean and 4 for variance), Decoder: 7 Layers: 200 units - 1000 units - Convolution Transpose: $32 \times 5 \times 5$; stride (1, 1) - Convolution Transpose: $32 \times 5 \times 5$; stride (1, 1) - Convolution Transpose: $32 \times 5 \times 5$; stride (1, 1) - Convolution Transpose: $2 \times 5 \times 5$; stride (1, 1)-Sigmoid Layer, Transition Dynamics: 3 layers: 100 units - 100 units - 24 units, MPC Prediction Horizon: 10, Control Horizon: 10. Number of Models: 12, Learning Rate: 0.0001, Optimizer: ‘ADAM’ [19], Cost Matrices : $Q = I_{n \times n}$, $R = 1$.