

NewtonianVAE: Proportional Control and Goal Identification from Pixels via Physical Latent Spaces

Miguel Jaques
University of Edinburgh
Edinburgh, UK
m.a.m.jaques@sms.ed.ac.uk

Michael Burke
Monash University
Melbourne, AU
michael.burke1@monash.edu

Timothy Hospedales
University of Edinburgh
Edinburgh, UK
t.hospedales@ed.ac.uk

Abstract

Learning low-dimensional latent state space dynamics models has proven powerful for enabling vision-based planning and learning for control. We introduce a latent dynamics learning framework that is uniquely designed to induce proportional controllability in the latent space, thus enabling the use of simple and well-known PID controllers. We show that our learned dynamics model enables proportional control from pixels, dramatically simplifies and accelerates behavioural cloning of vision-based controllers, and provides interpretable goal discovery when applied to imitation learning of switching controllers from demonstration. Notably, such proportional controllability also allows for robust path following from visual demonstrations using Dynamic Movement Primitives in the learned latent space.

1. Introduction

Vision-based control is highly desirable across numerous industrial applications, both in robotics and process control. At present, much practical vision-based control relies on supervised learning to build bespoke perception modules, prior to downstream dynamics modelling and controller design. This can be expensive and time consuming, and as a result there is growing interest in developing model-based approaches for direct vision-based control.

Model-based approaches for visual control tend to learn latent dynamics models that are subsequently used within suitable planning or model predictive control (MPC) frameworks, or to train policies for later use. We argue that this decoupling of dynamics and control is computationally expensive and often unnecessary. Instead we learn a structured latent dynamical model that directly allows for simple proportional control to be applied. Proportional-Integral-Derivative (PID) feedback control produces commands that are proportional to an

error or cost term between current system state \mathbf{x} and a (potentially dynamic) target state \mathbf{x}^{goal} :

$$\mathbf{u}_t = K_p (\mathbf{x}_t^{goal} - \mathbf{x}_t) + K_i \sum_{t'} (\mathbf{x}_{t'}^{goal} - \mathbf{x}_{t'}) + K_d \frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{\Delta t} \quad (1)$$

Gain terms (K_p, K_i, K_d) shape the controller response to errors. PID control is ubiquitous in industry, and broadly applicable across numerous domains, providing a simple and reliable off-the-shelf mechanism for stabilising systems. PID control is also the basis of a wide range of more powerful control strategies, including the more flexible dynamic movement primitives [24, 43] that augment PD control laws with a forcing function for trajectory following. Essentially we learn the state encoding $\mathbf{x}(I)$ from images I for which robots can be trivially controlled from pixels according to Eq 1.

We structure latent dynamics so that that PID control can be applied to move between latent states, to remove the requirement for complex planning or reinforcement learning strategies. Moreover, we show that imitation learning from demonstrations becomes a simple goal inference problem under a proportional control model in this latent space, and can even be extended to sequential tasks comprising multiple sub-goals.

Imitation learning from high dimensional visual data is particularly challenging [2]. Behaviour cloning, which seeks to reproduce demonstrations, is particularly vulnerable to generalisation failures for high dimensional visual inputs, while inverse reinforcement learning (IRL) [38] strategies are hard to train and extremely sample inefficient. By learning a structured dynamics model, we allow for more robust control in the presence of noise and simplify the inverse reward inference process. In summary, the primary contributions of this work are:

Embedding for proportional controllability We induce a latent space where taking an action in the direction between the current position and some target

position, $\mathbf{u} \propto \mathbf{x}^{target} - \mathbf{x}$, moves the system towards the target position. Uniquely, this enables simple proportional control from pixels.

Imitation learning using latent switching proportional control laws We leverage the properties of this embedding to frame imitation learning as a goal inference problem under a switching proportional control law model in the structured latent space for sequential goal reaching problems. This enables one-shot interpretable imitation learning of switching controllers from high-dimensional pixel observations.

Imitation learning using dynamic movement primitives (DMPs) We also leverage the properties of our embedding to fit dynamic movement primitives in the structured latent space for trajectory tracking problems. This enables one-shot imitation learning of trajectory following controllers from pixels.

Results show that embedding for proportional controllability produces more interpretable latent spaces, allows for the use of simple and efficient controllers that cannot be applied with less structured latent dynamical models, and enables one-shot learning of control and interpretable goal identification in sequential multi-task imitation learning settings.

2. Related Work

This paper takes a model-based approach to visual control, using variational autoencoding (VAE) [28]. Latent dynamical systems modelling using autoencoding is widely used [32], and has been proposed for Bayesian filtering [15, 27, 31], and as inverse graphics for improved video prediction and vision-based control [25]. Ha and Schmidhuber [21] train a latent dynamics model using a variational recurrent neural network (VRNN) in the latent space of a VAE, and then learn a controller that acts in this space using a known reward model. Hafner et al. [22] extend this approach to allow planning from pixels. Unfortunately, because these approaches decouple dynamics modelling and control, they place an unnecessary computational burden on control, either requiring sampling-based planning or further RL policy optimisation. We argue that this burden can be alleviated by imposing additional structure on the latent space such that proportional control becomes feasible.

In doing so, we build on the control hypothesis advocated by Full and Koditschek [17], which seeks to model complex phenomena and systems through simple template models and controllers, using anchor networks to abstract the complexity away from control. This also simplifies the challenges of imitation learning, allowing for sequential task composition [7].

The addition of structural inductive biases into neural models has become increasingly important for gener-

alisation. Injecting knowledge of known physical equations [20, 25] has been shown to improve dynamics modelling, while the inclusion of structured transition matrices was essential to learn Koopman operators [1] that model dynamical systems with compositional properties [35]. Here, a block-wise structure with shared blocks was used to learn transition dynamics, which highlighted the importance of added structure in linear state space models, but this was not applied to visual settings. Models like embed to control (E2C) [46] or deep variational Bayes filters (DVBF) [27] recover structured conditionally linear latent spaces which can be used for control, but, as will be demonstrated later, are still unsuitable for direct proportional control. PVEs [26] learn an explicit positional representation, but do so by minimizing a combination of several heuristic loss functions. Since these models do not use a decoder, it is not possible to visually inspect the learned representations in image space.

NewtonianVAE not only provides latent space interpretability, but also simplifies imitation learning. Inverse reinforcement learning (IRL) strategies for imitation learning typically struggle to learn from high dimensional observation traces as they tend to be based on the principle of feature counting and observation frequency matching [38], as in maximum entropy IRL [47]. Maximum entropy IRL has been extended to use a deep neural network feature extractor [47], but this is highly vulnerable to overfitting and has extensive data requirements. Recent adversarial IRL approaches [16, 18, 23] avoid the challenge of learning a global reward function by training policies directly, but these have yet to be successfully scaled to high dimensional problems. As a result, most imitation learning approaches tend to assume access to low dimensional states, avoiding the challenge of learning from pixels.

Behaviour cloning approaches using dynamic movement primitives (DMP) [24, 43] have proven particularly powerful for trajectory following control, but are typically applied to low-dimensional proprioceptive states directly as they require proportionally controllable state spaces. Deep DMPs [40] learn visually task parametrised DMPs, but the DMP itself still requires low dimensional state measurements. Chen et al. [8] propose VAE-DMPs, which impose DMP dynamics in the latent space of a variational auto-encoder, allowing for direct imitation learning. In contrast, this work learns dynamics models independently of tasks, which allows for more flexible downstream applications, including DMP fitting for trajectory following and switching multi-goal imitation learning from pixels (unlike Chen et al. [8], which use proprioception observations).

Standard imitation learning strategies can

fail in multi-goal settings or on more complex tasks. In order to address this, many approaches frame the problem of imitation learning from these lower level states as one of skill or options [30, 44] learning using switching state space models. These switching models include linear dynamical attractor systems [11], conditionally linear Gaussian models [9, 33], Bayesian non-parametrics [39, 41], and neural variational models [29]. Kipf et al. [29] learn task segmentations to infer compositional policies, but the model uses environment states directly instead of images. Burke et al. [5, 6] use a switching controller formulation for control law identification from image, proprioceptive state and control action observations. This work applies a similar strategy for goal inference, but, unlike the approaches above, makes use of a learned latent state representation and does not require proprioceptive or low level state information.

Despite this reliance on proprioceptive state information, there is a growing interest in direct visual imitation learning and control. Nair et al. [37] train a variational autoencoder (VAE) on image observations of an environment, and subsequently sample from this latent space in order to train goal-conditioned policies that can be used to move between different goal states. In contrast, we propose a latent dynamics model that allows for latent proportional controllability and eliminates the need to train a policy to move between goal states.

In addition to the works discussed above, a research area in the unsupervised learning literature of particular interest is that of learning physically plausible representations (from video) by enforcing temporal evolution according to explicit or implicit physical dynamics [4, 19, 25, 45]. Though promising, these approaches have only been applied to very simple toy environments where dynamics are well known, and are still to be scaled up to real world scenes.

3. Variational models for visual control

In order to learn a compact latent representation of videos that can be used for planning and control we use the variational autoencoder framework (VAE) [28, 42] and its recurrent formulation (VRNN), [10]. In this section we briefly present a general formulation of the VRNN, of which many recent models are particular cases or variations [15, 22, 27, 31, 46]. For derivation details please refer to [10].

Given a sequence of T images, $\mathbf{I}_{1:T}$, and actuations $\mathbf{u}_{1:T} \in \mathbb{R}^{d_u}$ and the corresponding latent representations, $\mathbf{z}_{1:T} \in \mathbb{R}^{d_z}$, the marginal image likelihood is given by:

$$p(\mathbf{I}_{1:T}|\mathbf{u}_{1:T}) = \int p(\mathbf{I}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T})p(\mathbf{z}_{1:T}|\mathbf{u}_{1:T})d\mathbf{z}_{1:T} \quad (2)$$

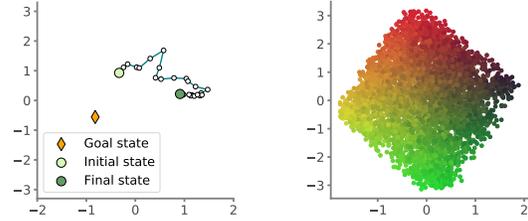


Figure 1: Trajectory of a point mass actuated using $\mathbf{u}_t \propto (\mathbf{x}^{goal} - \mathbf{x}_t)$ (left) in the latent space learned by an E2C model (right).

where we factorize the terms above as:

$$p(\mathbf{I}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = \prod p(\mathbf{I}_t|\mathbf{z}_t)$$

$$p(\mathbf{z}_{1:T}|\mathbf{u}_{1:T}) = \prod p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}_{t-1}),$$

with an approximate posterior given by:

$$q(\mathbf{z}_{1:T}|\mathbf{I}_{1:T}) = \prod q(\mathbf{z}_t|\mathbf{I}_t, \mathbf{z}_{t-1}, \mathbf{u}_{t-1}). \quad (3)$$

The model components are trained jointly by maximizing the lower bound on (2):

$$\mathcal{L} = \sum_t \mathbb{E}_{q(\mathbf{z}_t|\mathbf{I}_t, \mathbf{z}_{t-1}, \mathbf{u}_{t-1})} [p(\mathbf{I}_t|\mathbf{z}_t) + \text{KL}(q(\mathbf{z}_{t+1}|\mathbf{I}_{t+1}, \mathbf{z}_t, \mathbf{u}_t)||p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t))], \quad (4)$$

via the reparametrization trick, by drawing samples from the posterior distributions, $q(\mathbf{z}_t|\mathbf{I}_t, \mathbf{z}_{t-1}, \mathbf{u}_{t-1})$. Under this framework, the various desired inductive biases are usually built into the structure of the transition prior $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t)$. In this work we will build on the formulation that uses a linear dynamical system as latent dynamics:

$$p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t) = A(\mathbf{z}_t) \cdot \mathbf{z}_t + B(\mathbf{z}_t) \cdot \mathbf{u}_t + \mathbf{c}(\mathbf{z}_t) \quad (5)$$

which has been studied extensively in the context of deep probabilistic models [3, 15, 27, 31, 36].

4. Newtonian Variational Autoencoder

Motivation To motivate our model, we begin by examining the properties of an existing latent variable model used for control. We train an E2C model [46], since it applies a locally linear latent transition as in (5) and is highly representative of properties obtained in these types of model. We use a simple point mass system that can move in the $[x, y]$ plane and train the model on random transitions in image space (more details in the experiments section). Since the environment is 2D with 2D controls, we use a 4D latent space (2 dimensions for position and 2 for velocity). Our goal is to explore how the E2C model behaves when a basic proportional

control law $\mathbf{u}_t \propto (\mathbf{x}^{goal} - \mathbf{x}_t)$ is applied, where \mathbf{x} is the latent system configuration.

An immediate problem is that even though the latent coordinates corresponding to position are correctly learned (Fig. 1(right)), it is necessary to plot *every coordinate pair* and their correlation with ground truth positions in order to visually determine which 2 coordinates correspond to the position \mathbf{x} . Having determined such \mathbf{x} , we can use a random target position \mathbf{x}^{goal} and see if successively applying an action $\mathbf{u}_t \propto (\mathbf{x}^{goal} - \mathbf{x}_t)$ will guide the system towards \mathbf{x}^{goal} (which we term *proportional controllability*). Note that PID control is trivially achievable given a P-controllable system, so we focus on P-control for simplicity of exposition, without loss of generality. Fig. 1(left) shows that this simple control law fails to guide the system towards the goal state, even though the latent space is seemingly well structured. These problems are present in existing variational models for controllable systems, including E2C [46], DVBF [27] and the Kalman VAE [15].

To avoid the need for ground truth data and visual inspection, we construct a model that explicitly treats position and velocity as separate latent variables \mathbf{x} and \mathbf{v} . To ensure correct behaviour under a proportional control law¹ the change in position and velocity should be directly related to the force applied. I.e. given an external action \mathbf{u} representing the force (=acceleration) acting on a system, \mathbf{x} and \mathbf{v} should follow Newton’s second law, $d^2\mathbf{x}/dt^2 = \mathbf{F}/m$. Although this might seem like a trivial statement from a physical standpoint, this type of behaviour is not built into existing neural models, where the relationship between action and latent states can be arbitrary. This arbitrary relationship in turn complicates control, and it becomes necessary to learn downstream controllers or policies to compensate for these dynamics while meeting a control objective.

We make one additional observation: in many cases the external action \mathbf{u} is applied along disentangled dimensions of the system. For example, for a 2-arm robot, actions correspond to torques on the angles of each arm relative to its origin². These action dimensions correspond to the polar coordinates $[\theta_1, \theta_2]$, which are the ideal disentangled coordinates to describe such a robot. We use this fact to formulate a model that not only provides an interpretable and P-controllable latent space, but also the correct disentanglement by construction.

Formulation We now formulate a model satisfying the above desiderata. For an actuated rigid body systems with D degrees of freedom, we model the system

¹For further analysis of the convergence and stability of PID controllers, see [12, 13].

²The example also applies more generally to any robot actuated with torques along its joints.

configuration (positions or angles) by a set of coordinates $\mathbf{x} \in \mathbb{R}^D$ with double integrator dynamics, inspired by Newton’s equations of motion:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = A(\mathbf{x}, \mathbf{v}) \cdot \mathbf{x} + B(\mathbf{x}, \mathbf{v}) \cdot \mathbf{v} + C(\mathbf{x}, \mathbf{v}) \cdot \mathbf{u} \quad (6)$$

To build a discrete form of (6) into a VAE formulation, we use the instantaneous system configuration (or position) \mathbf{x} as the stochastic variable that is inferred by the approximate posterior, $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{I}_t)$, with velocity a deterministic variable that is simply the finite difference of positions, $\mathbf{v}_t = (\mathbf{x}_t - \mathbf{x}_{t-1})/\Delta t$. The generative model is now given by

$$p(\mathbf{I}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u}_{1:T}) = \prod p(\mathbf{I}_t|\mathbf{x}_t) \quad (7)$$

$$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = \prod p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}; \mathbf{v}_t) \quad (8)$$

where the transition prior is:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}; \mathbf{v}_t) = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_{t-1} + \Delta t \cdot \mathbf{v}_t, \sigma^2) \quad (9)$$

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \Delta t \cdot (A\mathbf{x}_{t-1} + B\mathbf{v}_{t-1} + C\mathbf{u}_{t-1}) \quad (10)$$

with $[A, \log(-B), \log C] = \text{diag}(f(\mathbf{x}_t, \mathbf{v}_t, \mathbf{u}_t))$, where f is a neural network with linear output activation. Using diagonal transition matrices encourages correct coordinate relations between \mathbf{u} , \mathbf{x} and \mathbf{v} , since linear combinations of dimensions are eliminated. In order to obtain the correct directional relation between \mathbf{u} and \mathbf{x} , required for interpretable controllability, we set C to be strictly positive (in addition to diagonal). B is strictly negative to provide a correct interpretation of the term in \mathbf{v} as friction, which aids trajectory stability. During inference, \mathbf{v}_t is computed as $\mathbf{v}_t = (\mathbf{x}_t - \mathbf{x}_{t-1})/\Delta t$, with $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{I}_t)$ and $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{I}_{t-1})$. This inference model provides a principled way to infer velocities from consecutive positions, similarly to [26]. We use Gaussian $p(\mathbf{I}_t|\mathbf{x}_t)$ and $q(\mathbf{x}_t|\mathbf{I}_t)$ parametrized by a neural network throughout.

We train all model components using the following ELBO (full derivation in Appendix A):

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{I}_t)q(\mathbf{x}_{t-1}|\mathbf{I}_{t-1})}[\mathbb{E}_{p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t; \mathbf{v}_t)}p(\mathbf{I}_{t+1}|\mathbf{x}_{t+1}) + \text{KL}(q(\mathbf{x}_{t+1}|\mathbf{I}_{t+1})\|p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t; \mathbf{v}_t))] \quad (11)$$

A crucial component of this ELBO is performing future-rather than current-step reconstruction through the generative process (first term above). This is known to encourage the use of the transition prior when learning the latent representation [22, 27, 46].

Further considerations Another key difference between a simple LDS and our Newtonian model is the fact that we consider velocity to be a deterministic latent variable that is uniquely determined by the stochastic

positions. In contrast, independent inference through \mathbf{z} means that position and velocity might not have the direct relation that is present in the physical world (velocity as the derivative of position). Both of these contribute to a lack of physical plausability, in the Newtonian sense, in existing models. Though technically our transition prior is a special case of the LDS (5), these added structural constraints are crucial in order to induce a Newtonian latent space that directly allows for PID control of latent image states.

5. Efficient Imitation with P-Control

A key benefit of the Newtonian latent space is that it dramatically simplifies image-based imitation learning. Given a visual demonstration sequence $D_{\mathbf{I}} = \{(\mathbf{I}_1, \mathbf{u}_1), \dots, (\mathbf{I}_T, \mathbf{u}_T)\}$, we encode the frames using the inference network $q(\mathbf{x}|\mathbf{I})$ described above in order to produce demonstrations in latent space, $D_{\mathbf{x}} = \{(\mathbf{x}_1, \mathbf{u}_1), \dots, (\mathbf{x}_T, \mathbf{u}_T)\}$.

5.1. Learning Vision-Driven Switching P-Control

We can fit a switching P-controller³ to a set of demonstration sequences in latent space using a Mixture Density Network (MDN), where the action likelihood given a state is a mixture of N proportional controllers:

$$P(\mathbf{u}_t|\mathbf{x}_t) = \sum_{n=1}^N \pi_n(\mathbf{x}_t) \mathcal{N}(\mathbf{u}_t | K_n(\mathbf{x}_n^{goal} - \mathbf{x}_t), \sigma_n^2) \quad (12)$$

where K_n , \mathbf{x}_n^{goal} and σ_n^2 , $\forall n \in 1..N$, are learnable parameters, and $\pi(\mathbf{z})$ is a parametric function like a neural network. Intuitively, fitting this MDN to the latent demonstrations splits the demonstrations into regions where a specific proportional controller would correctly fit that part of the trajectory. If the latent space is P-controllable (such as the one produced by the NewtonianVAE), the vectors \mathbf{x}_n^{goal} will correspond to the intermediate goals or bottleneck states in the demonstration sequence. As an added benefit, we can pass the learned goals through NewtonianVAE’s decoder in order to obtain their visual representation, providing an interpretable control policy.

Learning a finite-state machine Having identified the latent vectors corresponding to the goals, we determine the order in which they must be reached by analysing their visits during the demonstrations, directly extracting initiation sets and termination conditions. This produces a simple finite-state machine (FSM) that determines goal state transitions. The FSM and extracted P-controllers can then be used to reproduce demonstrated behaviours by driving the robot to

³We use a P-controller instead of a PID-controller for simplicity of exposition and without loss of generality.

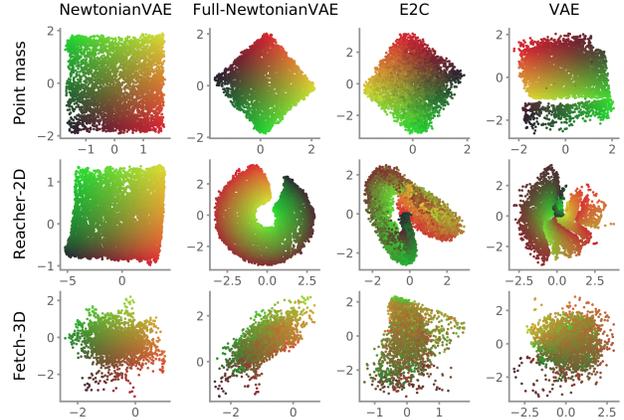


Figure 2: Latent spaces of various models in the point mass, reacher-2D and fetch-3D environments. Each dot corresponds to the latent representation of a test frame, and the red-to-green color coding encodes the true 2D position/angle values. For E2C [46], we plot the two latent dimensions that best correlated with the true positions. Since the configuration space of the fetch-3D env is 4D, we visualize only the first two coordinates. Only for our NewtonianVAE does latent space (position) and true space (color) correlate perfectly.

each goal in succession, but could also be used within an options framework [44] for reinforcement learning.

5.2. Learning Visual Path Following with DMPs

It is clear that the latent space of a NewtonianVAE can be used for switching goal-based imitation learning, but proportionality is also a precursor for trajectory following using DMPs. A DMP [24] is a proportional-derivative controller with a learned forcing function

$$\tau \ddot{\mathbf{x}} = \alpha (\beta (\mathbf{x}^{goal} - \mathbf{x}) - \dot{\mathbf{x}}) + \mathbf{f}. \quad (13)$$

Here, τ is a time scaling constant, and α, β are proportional control gain terms. The forcing function

$$\mathbf{f}_t = \sum_{i=1}^N \frac{\Phi(t) \mathbf{w}_i}{\sum_{i=1}^N \Phi(t)} (\mathbf{x} - \mathbf{x}^{goal}) \quad (14)$$

captures trajectory dynamics, using a weighted linear combination of radial basis functions, $\Phi(t) = \exp(-\frac{1}{\sigma_i^2} (y - c_i)^2)$, with centres c_i and variances σ_i^2 .

The canonical system $\dot{y} = -\alpha_y y$ gently decays over time, smoothly modulating the forcing function until reaching an end goal, \mathbf{x}^{goal} . Basis functions and parameters are fit to demonstration trajectories using weighted linear regression. Since the NewtonianVAE embeds for proportionality, DMPs can be fit directly to the latent space from demonstration data, allowing for vision-based trajectory control and path following.

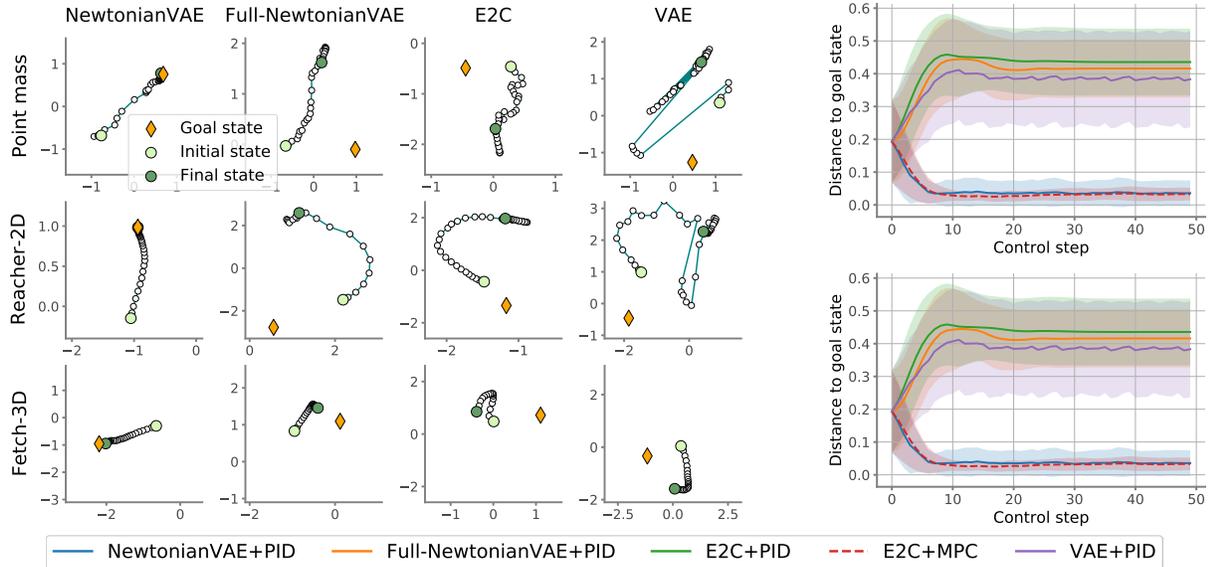


Figure 3: **Left:** P-control trajectories for point mass, reacher-2D and fetch-3D environments. Plots are in the latent space of Fig. 2. We can see that only NewtonianVAE produces a latent space where a P-controller correctly leads the systems from the initial to goal state. **Right:** Convergence rates of PID control using various latent embeddings for the point mass (left) and reacher-2D (right) systems, over 50 episodes. We use gain parameters $K_p = 8$, $K_i = 2$, $K_d = 0.5$. For contrast, we show Model Predictive Control (MPC, using CEM planning [22]).

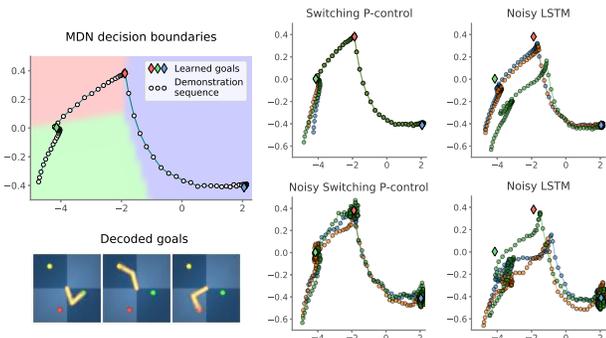


Figure 4: **Left:** Demonstration sequence and learned mixture of P-controllers (MDN). Each background color and corresponding diamond correspond to a component $\pi_n(\mathbf{x})$ and \mathbf{x}_n^{goal} , $\forall n \in \{1, 2, 3\}$, respectively. **Right:** Rollouts after imitation learning using switching P-controllers and LSTM policy, with a single demonstration sequence. In the noisy regime each action has an added noise $\mathcal{N}(0, 0.25^2)$. All plots are in the NewtonianVAE’s latent space.

6. Experiments

We validate our model on 3 simulated continuous control environments, to allow for better evaluation and ablations, and on data collected from a real PR2 robot.

Point mass A simple point mass system adapted from the `PointMass` environment from `dm_control`. The mass is linearly actuated in the 2D plane and

its movement bounded by the edges of the frame.

Reacher-2D A 2D reacher robot adapted from the `Reacher` environment in `dm_control` and inspired by [29]. We alter the environment so that the robot’s middle joint can only bend in one direction, in order to prevent the existence of two possible arm configurations for every end effector position. We also limit the origin joint angle range to $[-160, 160]$ so that the system configuration can be described in polar coordinates by two variables corresponding to the angle of each arm, avoiding a discontinuity in case of full circular motion.

Fetch-3D The 3D reacher environment `FetchReachEnv` from OpenAI Gym. We use this to show that our model learns the desirable representations even in visually rich 3D environments of multi-joint robots with partial occlusions.

To train the models, we generate 1000 random sequences with 100 time-steps for the point mass and reacher-2D systems, and 30 time-steps for the fetch-3D system. More implementation details for each of the environments can be found in Appendix B.

Baseline models We compare our model to E2C⁴ and a static VAE (each frame encoded individually). Additionally, in order to better understand the effect of diagonality and positivity of the transition matrices in (10), we test Full-NewtonianVAE, where the matrices

⁴DBVF [27] and E2C learn similar latent spaces, as both rely on an unstructured conditionally linear dynamical system.

A, B, C are unbounded and full rank. Architecture and training details can be found in Appendix B.

6.1. Visualizing latent spaces and P-controllability

In this section we compare the latent space and P-controllability properties of the NewtonianVAE and baseline models on the simulated environments: point mass, reacher-2D and fetch-3D.

Comparing latent spaces We start by visualizing the latent spaces learned by each models on all the environments. Fig. 2 shows that only the NewtonianVAE is able to learn a representation corresponding to the natural disentangled coordinates in both environments (e.g. $[x, y]$ in the point mass and $[\theta_1, \theta_2]$ in the reacher-2D), and that these are correctly correlated with ground-truth values, coded in the red-green spectrum. This shows that the structure imposed on the transition matrices in (10) is key to learning correct latent spaces in both Cartesian and polar coordinates.

P-controllability Even though the models above produce different latent spaces, most are well structured and show a clear correlation with the ground truth state (color coded). Although their structure is visually appealing, we are primarily interested in verifying is whether they satisfy P-controllability. To do this, we sample random starting and goal states, and successively apply the control law $\mathbf{u}_t \propto (\mathbf{x}(\mathbf{I}^{goal}) - \mathbf{x}_t(\mathbf{I}_t))$. A space is deemed P-controllable if the system moves to $\mathbf{x}(\mathbf{I}^{goal})$ in the limit of many time-steps. For reference, we also apply model-predictive control to E2C.

Convergence curves in the true state space are shown in Fig. 3, along with example rollouts in the learned latent space (more examples in Appendix C). We can see that only NewtonianVAE produces P-controllable latent states, as all the remaining models diverge under a P-controller. This highlights the fact that even though the latent spaces learned by the Full-NewtonianVAE and E2C are seemingly well structured for the point mass system, they fail to provide P-controllability. While these systems can still be stabilised using more complex control schemes such as MPC, this is entirely unnecessary with a P-controllable latent space, where trivial control laws can be applied directly.

6.2. MDN goal and boundary visualization

Having trained a NewtonianVAE on a dataset of random transitions we can use the learned representations to fit the mixture of P-controllers in (12) to the few-shot demonstration sequences.

Reacher-2D In this environment there are three colored balls in the scene and the task is reaching the three balls in succession, where the arm’s starting location varies across demonstration sequences. We used the

true reacher model with a custom controller to generate demonstration images. A full demonstration sequence is shown in Appendix B. For this experiment we use a linear $\pi(\mathbf{x})$, though a MLP yields similar results.

After fitting (12) on a *single* demonstration sequence, we visualize the goals \mathbf{x}^{goal} and the decision boundaries of the switching network $\pi(\mathbf{x})$ in Fig. 4(left). The figure shows that goal states are correctly identified (diamond markers), and that the three sub-task regimes are correctly segmented. Decoding \mathbf{x}^{goal} , confirms that the goals are correctly represented in image space, adding a layer of interpretability to an upstream control policy.

Imitation learning performance We now compare various imitation learning methods in the simulated task described above. A reward of 1.0 is given when the system reaches a neighborhood of each target (as measured in the true system state), but the targets must be reached in sequence. A more detailed description of the task can be found in Appendix B. Our method (switching P-controller) uses a finite-state machine inferred from the MDN trained on latent demonstrations (Fig. 4(left)). We compare it to behaviour cloning with an LSTM with 50 recurrent units, in the NewtonianVAE’s latent space, and GAIL [23], a state-of-the-art IRL method trained on ground truth proprioceptive states. Table 1 shows the imitation efficiency for increasing numbers of demonstration sequences, with example rollouts shown in Fig. 4(right). The results show that goal-driven P-control in a hybrid control policy is significantly more data efficient and robust to noise than a standard behaviour cloning policy. Additionally, switching controllers dramatically outperform GAIL⁵, even though this was trained on 5 times the number of environment interactions used by the NewtonianVAE.

Real multi-object reacher We now apply our model to real robot data. Here, we record a 7-DoF PR2 robot arm that moves between 6 objects in succession in a hexagon pattern. A full sequence comprises approximately 100 frames. We use 636 frames to train the NewtonianVAE and an additional 100 held-out frames to train the MDN. Further model and dataset details can be found in Appendix B.

Fig. 5 shows the image representations of the learned goals (left) and the mode $\pi(\mathbf{x})$ that is active for each frame in the demonstration sequence (right). We can see that the six goals are correctly identified by the MDN, and that segmentations are correct in the sense that a frame is assigned to the learned goal to which the robot is moving at that time step. Note that the model is able to recover correct goals and segmentations even

⁵Maximum Entropy IRL performed equally poorly, failing to reach a single goal. This is unsurprising, due to the connections between this and adversarial imitation learning [14].

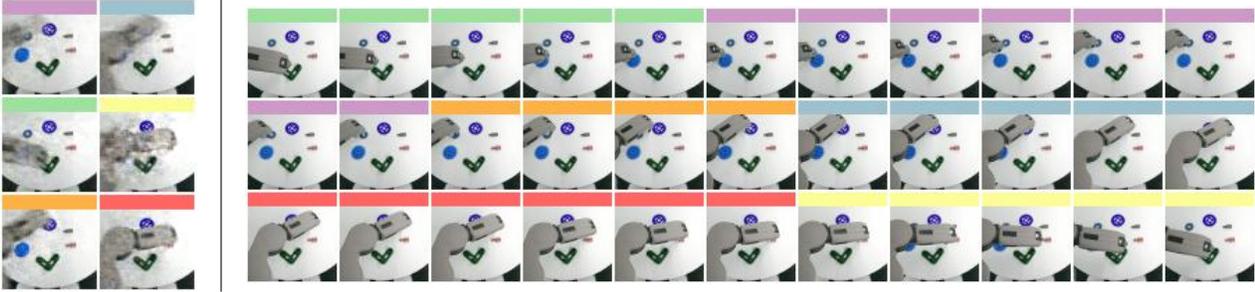


Figure 5: Decoded goals (left) and sequence segmentation (right) learned for a 6-goal visual trajectory of a PR2 robot. The sequence shows 33 equally spaced frames of a 100-frame demonstration.

| Demonstration sequences | Switching P-controller | | LSTM | | GAIL from proprioception |
|-------------------------|------------------------|-----------------|-----------------|-----------------|--------------------------|
| | Clean | Noisy | Clean | Noisy | |
| 1 | 3.0 ± 0.0 | 2.17 ± 0.32 | 0.81 ± 0.35 | 0.27 ± 0.20 | — |
| 10 | 3.0 ± 0.0 | 2.01 ± 0.34 | 3.00 ± 0.00 | 1.42 ± 0.34 | — |
| 100 | 3.0 ± 0.0 | 2.06 ± 0.30 | 3.00 ± 0.00 | 1.23 ± 0.30 | 0.62 |

Table 1: Efficiency of imitation learning methods for vision-based sequential multi-task control. Metric: Environment Reward (max = 3.0). The NewtonianVAE is used to encode the frames. ‘Noisy’: Added action noise $\mathcal{N}(0, 0.25^2)$ during the rollouts. Error ranges: 95% confidence interval across 100 rollouts. GAIL is trained for 5000 episodes.

though not all of the joints are visible in every frame.

6.3. Fitting DMPs for path following in latent space

We show how the NewtonianVAE can be used to enable a robot to learn a vision-driven controller to follow a demonstration trajectory, using the fetch-3D environment. To this end, we draw a ‘G’-shaped trajectory in the first 2 dimensions of the latent space and fit a DMP. The DMP runs in 100 time-steps, spanning 4 seconds of execution, where we feed the acceleration output by the DMP as the action to the environment, and the new state and velocity is inferred by the NewtonianVAE.

Fig. 6 shows that the robot correctly follows the demonstration trajectory, showing that the latent space induced by the NewtonianVAE enables path following using a DMP just by virtue of its P-controllability property, without needing to be explicitly trained to perform well under a DMP, as done by [8].

7. Discussion

Limitations and Future Work This work assumes that underlying systems are proportional controllable, and follow Newtonian dynamics. Moreover, it should be noted that vision-based torque control of high dimensional robot manipulators requires high speed vision. However, in our opinion, the most notable limitation is the fact that the imitation learning model only learns a fixed set of goals. Ideally, the agent would learn a semantic goal, which would represent a command “fetch the yellow ball”, for a variable position of the yellow ball and not a fixed state. However, this would require demonstration data with substantially more

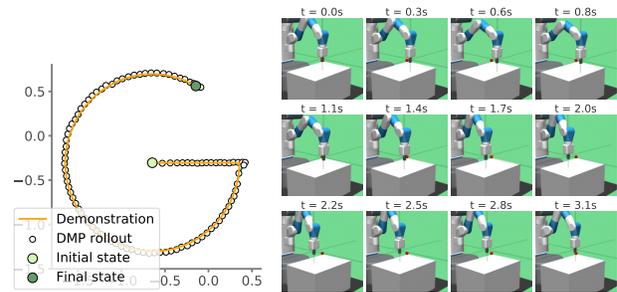


Figure 6: **Left:** Overhead view of demonstration and trajectory produced by the DMP in the fetch-3D environment. The first 2 dimensions of the NewtonianVAE’s latent space are shown. **Right:** Frames seen by the NewtonianVAE during this rollout.

variety than considered here. We have also avoided multi-modal demonstrations for simplicity, though we believe it would be of interest to integrate our method with approaches like InfoGAIL [34].

Conclusion We introduced NewtonianVAE, a structured latent dynamics model designed to allow P-controllability from pixels. Results show that this structured latent space allows for trivial, robust control in the presence of noise and dramatically simplifies and improves imitation learning, which can be framed either as a switching goal-inference or as a path following problem in the latent space. Additionally, our model provides visually interpretable goal discovery and task segmentation under both simulated and real environments, without any labelled or proprioception data.

References

- [1] I. Abraham, G. De, L. Torre, and T. D. Murphey. Model-Based Control Using Koopman Operators. In *RSS*, 2017.
- [2] J. A. D. Bagnell. An invitation to imitation. Technical Report CMU-RI-TR-15-08, Carnegie Mellon University, Pittsburgh, PA, March 2015.
- [3] P. Becker-Ehmck, J. Peters, and P. Van Der Smagt. Switching Linear Dynamics for Variational Bayes Filtering. In *ICML*, 2019.
- [4] F. D. A. Belbute-Peres, K. A. Smith, K. R. Allen, J. B. Tenenbaum, and J. Z. Kolter. End-to-End Differentiable Physics for Learning and Control. In *NIPS*, 2018.
- [5] M. Burke, Y. Hristov, and S. Ramamoorthy. Hybrid system identification using switching density networks. In *CoRL*, 2019.
- [6] M. Burke, S. Penkov, and S. Ramamoorthy. From explanation to synthesis: Compositional program induction for learning from demonstration. In *RSS*, 2019.
- [7] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.
- [8] N. Chen, M. Karl, and P. Van Der Smagt. Dynamic movement primitives in latent space of time-dependent variational autoencoders. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 629–636. IEEE, 2016.
- [9] S. Chiappa and J. R. Peters. Movement extraction by detecting dynamics switches and repetitions. In *NIPS*, 2010.
- [10] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A Recurrent Latent Variable Model for Sequential Data. *Advances in Neural Information Processing Systems*, 2015.
- [11] K. R. Dixon and P. K. Khosla. Trajectory representation using sequenced linear dynamical systems. In *ICRA*, 2004.
- [12] R. C. Dorf and R. H. Bishop. *Modern control systems*. Pearson, 2011.
- [13] L. Duc, A. Ilchmann, S. Siegmund, and P. Taraba. On stability of linear time-varying second-order differential equations. *Quarterly of applied mathematics*, 64(1):137–151, 2006.
- [14] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.
- [15] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *NIPS*, 2017.
- [16] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. 2018.
- [17] R. J. Full and D. E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of experimental biology*, 202(23):3325–3332, 1999.
- [18] S. K. S. Ghasemipour, R. Zemel, and S. Gu. A divergence minimization perspective on imitation learning methods. *CoRL*, 2019.
- [19] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian Neural Networks. In *NeurIPS*, 2019.
- [20] V. L. Guen and N. Thome. Disentangling Physical Dynamics from Unknown Factors for Unsupervised Video Prediction. In *CVPR*, 2020.
- [21] D. Ha and J. Schmidhuber. World models. In *NeurIPS*, 2018.
- [22] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019.
- [23] J. Ho and S. Ermon. Generative adversarial imitation learning. In *NIPS*, 2016.
- [24] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [25] M. Jaques, M. Burke, and T. Hospedales. Physics-as-Inverse-Graphics: Unsupervised Physical Parameter Estimation from Video. In *ICLR*, 2020.
- [26] R. Jonschkowski, R. Hafner, J. Scholz, and M. Riedmiller. PVEs: Position-Velocity Encoders for Unsupervised Learning of Structured State Representations. *CoRR*, abs/1705.09805, 2017.
- [27] M. Karl, M. Soelch, J. Bayer, and P. Van Der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *ICLR*, 2018.
- [28] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.
- [29] T. Kipf, Y. Li, H. Dai, V. Zambaldi, A. Sanchez-Gonzalez, E. Grefenstette, P. Kohli, and P. Battaglia. CompILE: Compositional Imitation Learning and Execution. In *ICML*, 2019.
- [30] G. Konidaris and A. G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *NIPS*, 2009.

- [31] R. G. Krishnan, U. Shalit, and D. Sontag. Deep Kalman Filters. *arXiv preprint arXiv:1511.05121*, 2015.
- [32] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filiat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.
- [33] S. Levine and P. Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *NIPS*, 2014.
- [34] Y. Li, J. Song, and S. Ermon. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations. In *NIPS*, 2017.
- [35] Y. Li, H. He, J. Wu, D. Katabi, and A. Torralba. Learning compositional koopman operators for model-based control. In *ICLR*, 2020.
- [36] S. W. Linderman, M. J. Johnson, A. C. Miller, R. P. Adams David M Blei Liam Paninski Harvard, and G. Brain. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *AISTATS*, 2017.
- [37] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual reinforcement learning with imagined goals. In *NeurIPS*, 2018.
- [38] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *ICML*, 2000.
- [39] S. Niekum and A. G. Barto. Clustering via dirichlet process mixture models for portable skill discovery. In *NIPS*. 2011.
- [40] A. Pervez, Y. Mao, and D. Lee. Learning deep movement primitives using convolutional neural networks. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 191–197. IEEE, 2017.
- [41] P. Ranchod, B. Rosman, and G. Konidaris. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *IROS*, 2015.
- [42] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.
- [43] S. Schaal. Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.
- [44] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.
- [45] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins. Hamiltonian Generative Networks. In *ICLR*, 2020.
- [46] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *NIPS*, 2015.
- [47] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.