Disentangled Generative Models for Robust Prediction of System Dynamics

Stathi Fotiadis [1] **Mario Lino** [2] **Shunlong Hu** [1] **Stef Garasto** [3] **Chris D Cantwell** [2] **Anil Anthony Bharath** [1]

Abstract

The use of deep neural networks for modelling system dynamics is increasingly popular, but long-term prediction accuracy and out-of-distribution generalization still present challenges. In this study, we address these challenges by considering the parameters of dynamical systems as factors of variation of the data and leverage their ground-truth values to disentangle the representations learned by generative models. Our experimental results in phase-space and observation-space dynamics, demonstrate the effectiveness of latent-space supervision in producing disentangled representations, leading to improved long-term prediction accuracy and out-of-distribution robustness.

1. Introduction

The robust prediction of the behaviour of dynamical systems remains an open question in machine learning, and engineering in general. The ability to make robust predictions is important not only for forecasting systems of interest like weather (Garg et al., 2021), but also because it supports innovations in fields like system control, autonomous planning (Hafner et al., 2019) and computer-aided engineering (Brunton et al., 2020). In this context, the use of deep generative models has recently gained significant traction for sequence modelling (Girin et al., 2020). The robustness of machine learning models can be considered along two axes: (1) long-term and (2) out-of-distribution (OOD) performance. Accurate long-term prediction can be notoriously difficult in many dynamical systems because error accumulation can cause divergence in finite time (Zhou et al., 2020; Raissi et al., 2019), a problem that even traditional solvers can suffer from. At the same time, machine learning techniques are known to suffer from poor OOD performance (Goyal & Bengio, 2020), when they are employed in a setting they had not encountered in their training phase.

When it comes to modelling dynamics, training data contain both the dynamics themselves and the dynamical system parameters. However, many approaches to learning fail to distinguish between the two, which could result in entangled representations, leading to overfitting and thus poorer forecasts (Bengio et al., 2013). Here, our aim is to investigate generative models whose latent space is disentangled in such a way that the parameters and the dynamics are distinctly represented. More specifically, we explore dynamical systems modelled by ordinary differential equations (ODEs) and their respective parameters.

Our method builds on two elements. First, the inherent ability of Variational Autoencoders (VAEs) (Kingma & Welling, 2014) to produce disentangled representations in an unsupervised way (Higgins et al., 2017), a feature that has been applied in the context of image and scene modelling (Kim & Mnih, 2018). Second, latent space supervision with ground-truth factors has been found to produce more disentangled representations in image modelling (Locatello et al., 2019). We motivate the use of disentangled representations through a theoretical analysis of the emission process through the scope of dynamical systems. In practice, we treat the parameters of a dynamical system as factors of variation of the data distribution and use the ground-truth values of these parameters to improve the latent space disentanglement. While various assumptions, like domain stationarity, have been used to improve the disentanglement in the prediction of dynamical systems in an unsupervised way (Li & Mandt, 2018; Miladinović et al., 2019), to the best of our knowledge, this is the first attempt to use supervised disentanglement for system dynamics. Furthermore, contrary to system-identification techniques that require knowledge of the full underlying system to be computationally effective (Ayyad et al., 2020), our technique only needs to be aware of the system parameters.

Contributions Our work is the first, to the best of our knowledge, that uses ground-truth information of the dynamical system parameters to disentangle the latent space of generative models. We provide a theoretical motivation for disentangled representations in dynamical system prediction and,

[1]Department of Bioengineering, Imperial College London, UK [2]Department of Aeronautics, Imperial College London, UK [3]School of Computing and Mathematical Sciences, University of Greenwich, London, UK. Correspondence to: Stathi Fotiadis <s.fotiadis19@imperial.ac.uk>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

practically, encourage latent space disentanglement through supervision. We conduct experiments with VAEs trained with noisy observations of the phase-space of 3 dynamical systems. We also apply our method to a state-of-the-art generative model (Recurrent State Space Model (Hafner et al., 2019)) trained on image sequences of a swinging pendulum. We propose a definition of OOD data in the context of system dynamics and evaluate the performance of models in- and out-of-distribution. We demonstrate that models with a disentangled latent space can better capture the variability of dynamical systems and produce more accurate long-term predictions, both in- and out-of-distribution. However, the practical importance of our method is currently limited by the labelling cost. It would be worth assessing the model in the semi-supervised setting, as that would be better suited for real-world application. All the necessary code to reproduce our experiments is provided at `https://github.com/stathius/sd-vae`.

2. Related Work

VAEs and disentanglement. Disentanglement aims to produce representations where each latent variable captures a different factor of variation of the data distribution. This can also be seen as identifying the true causal model of the data-generating process (Schölkopf, 2019). While supervised disentanglement is a long-standing idea (Mathieu et al., 2016), information-theoretic properties can be leveraged to allow unsupervised disentanglement in VAEs (Higgins et al., 2017; Kim & Mnih, 2018). Recent findings (Locatello et al., 2020a) emphasize the vital role of inductive biases from models or data for useful disentanglement, leading to semi- and weakly-supervised disentanglement approaches (Locatello et al., 2019; 2020b). In the field of physical sciences, hierarchical priors have been proposed to learn disentangled representations of high-dimensional spatial fields (Jacobsen & Duraisamy, 2022). To assess the strength of disentanglement, simulated datasets are usually used, because simulations give access to the ground-truth factors of variation (i.e., color or shape of an object in image data). Various metrics have been proposed to quantify disentanglement, both predictor-based (Eastwood & Williams, 2018; Kumar et al., 2018) and information-theoretic ones (Chen et al., 2018) but the task still presents challenges (Carbonneau et al., 2020).

Disentanglement in sequence modelling. While disentanglement methods are often tested in a static (image) setting, there is a growing interest in applying disentanglement to sequence dynamics. Using a bottleneck corresponding to the degrees of freedom of the physical system, Iten et al. (2018) learn an interpretable representation using a VAE. However, their model gives physically inconsistent predictions in OOD data (Barber et al., 2021). Disentangling content from dynamics has also been tried in deep state-space models (SSMs) (Fraccaro et al., 2017; Li & Mandt, 2018), but these methods focus mostly on modelling variations in the appearance of moving objects, failing to take dynamics into account. Unsupervised techniques have also been proposed. Assuming domain stationarity, Miladinović et al. (2019) separate the dynamics from sequence-wide properties in dynamical systems like Lotka-Volterra but they do not fully evaluate the OOD performance of their model. Yeo et al. (2021) suggest that learning hierarchy of semantic concepts leads to feature abstraction and enhanced disentanglement, while (Li et al., 2023) propose a model for time-series generation whose representation is disentangled by minimizing the pairwise total correlation between the latent variables. While unsupervised methods have their advantages, they also dismiss a wealth of information that can be cheaply collected from simulated data, a gap that our method tries to fill.

VAEs for sequence modelling. Dynamical VAEs (Girin et al., 2020) have long been used to model sequence dynamics. Combining VAEs with physics-informed neural networks (Raissi et al., 2019) can also be used to model stochastic differential equations (Zhong & Meidani, 2022). Feed-forward VAEs have also attracted a lot of interest in modelling physical systems. There are two main motivations for this. First, VAEs offer various ways to incorporate the inductive biases obtained from prior knowledge of the physical system. Second, since their latent space is relatively simple, one can easily assess if those inductive biases result in more interpretable representations. Methods to incorporate inductive biases include (i) bottlenecks based on the degrees of freedom of the physical system (Iten et al., 2018), (ii) the use of geometric and topological information of the dynamical system responses to shape the manifold of the latent representations (Lopez & Atzberger, 2022), and (iii) using physics-informed priors (Takeishi & Kalousis, 2021). Furthermore, feed-forward VAEs can be combined with recurrent neural networks (RNN) to improve accuracy while at the same time learning highly-disentangled representations of dynamical systems (Yeo et al., 2021).

3. Problem formulation

3.1. Dynamical systems

Let $\mathbf{u} \in \mathbb{R}^d$ be the state of a system. We consider system dynamics that are governed by a set of differential equations (DEs):

$$\frac{d\mathbf{u}}{dt} = \mathcal{F}(\mathbf{u}, \boldsymbol{\xi}) \tag{1}$$

where $\mathcal{F}$ describes the governing equations and $\boldsymbol{\xi} \in \mathbb{R}^{N_\xi}$ denotes the parameters of these DEs. While these equations describe how the system state evolves over time, there is a limited number of real-world problems where they can be

solved analytically. Hence, most often, the time evolution of a system is acquired by numerical methods, given the governing equations and some initial state. In experimental data, observations $\tilde{\mathbf{u}}_t$ contain some noise: $\tilde{\mathbf{u}}_t = \mathbf{u}_t + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \in \mathbb{R}^d$ is the stochastic uncorrelated noise. In our computational experiments, the data are corrupted with white Gaussian noise to simulate observation noise.

In the experimental section, we are concerned with dynamical systems governed by Ordinary DEs (ODEs) but our methods could in principle apply to Partial DEs as well. The three dynamical systems we study are the swinging pendulum, the Lotka-Volterra system used to model prey-predator populations, and the planar 3-body system. The governing equations are the following:

$$\textbf{Simple pendulum: } \ddot{\theta} = -\frac{g}{\ell}\sin\theta \tag{2}$$

$$\textbf{Lotka-Volterra: } \dot{x} = \alpha x - \beta xy$$
$$\dot{y} = \delta xy - \gamma y \tag{3}$$

$$\textbf{3-body system: } \dot{\mathbf{v}}_i = \frac{K_1}{m_i}\sum_j \frac{m_i m_j}{|\mathbf{r}_{ij}|^3}\mathbf{r}_{ij}$$
$$\dot{\mathbf{x}}_i = K_2 \mathbf{v}_i \tag{4}$$
$$\mathbf{v}_i, \mathbf{x}_i \in \mathbb{R}^2, i \in [1,2,3]$$

Where θ is the length of the pendulum, g is the acceleration due to gravity and ℓ is its length. Since the two parameters appear in ratio we keep gravity constant and only vary the length of the pendulum, i.e., $\boldsymbol{\xi} = [\ell]$. In Lotka-Volterra, x, y are the prey and predator populations while the 4 parameters $\boldsymbol{\xi} = [\alpha, \beta, \gamma, \delta]$ describe the interaction of the two species. In the 3-body, $\mathbf{x}_i, \mathbf{v}_i$ are the positions and velocities of the bodies and the 4 parameters $\boldsymbol{\xi} = [K_1, m_1, m_2, m_3]$ represent the gravity constant and masses. Overall, these systems are characterized by a varied number of degrees of freedom, governing equations and number of parameters. We also refer to Appendix B.2 for more details.

3.2. Theoretical motivation for disentanglement

The problem setup that involves inferring the evolution of a system state, up to some time in future, $t+n$, given a number of previous (observed) states up to a point in time t. The system dynamics are defined by the form of the differential equation (DE), the parameters of it $\boldsymbol{\xi}$, and the initial conditions $\boldsymbol{I}$. Since the DE is deterministic, if parameters and current state is known then next step can be calculated using numerical methods with a high precision (bound by the numerical precision of the computational method). We can consider the simplified setting where the conditional distribution of the next state $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t}; \boldsymbol{\xi}_C, \boldsymbol{I}_C)$ is characterized only by the noise of the observations, assuming there is no other type of uncertainty. In the absence of noise the distribution becomes Dirac's delta function. In practice

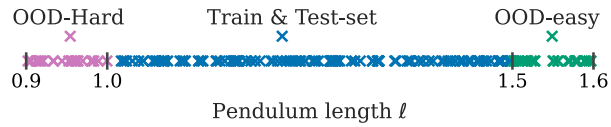


Figure 1: Samples from the parameter distributions of pendulum length ℓ. Each trajectory in the train, validation & test sets is simulated with length drawn from $\ell \sim \mathcal{U}(1.0, 1.5)$, while OOD-Easy and OOD-Hard have disjoint distributions. Note that predicting the trajectory of a shorter pendulum is harder for our models because it swings faster.

we often do not have access to $\boldsymbol{\xi}, \boldsymbol{I}$. There are two options in this case a) assign priors on $\boldsymbol{\xi}_C, \boldsymbol{I}_C$, and marginalize over them to obtain an estimate of the marginal, and b) estimate $\boldsymbol{\xi}_C$ and $\boldsymbol{I}_C$ and directly model the conditional. Given the wide nature of divergence in the trajectories of a system for different $\boldsymbol{\xi}_C, \boldsymbol{I}_C$, it is hard to both assign a proper prior and efficiently marginalize. On the other hand if we can derive good estimates for $\boldsymbol{\xi}_C, \boldsymbol{I}_C$, then the second modelling choice becomes more appealing and this is where disentanglement can be beneficial. For simplicity, we consider the pendulum where its length $\boldsymbol{\xi}_C = l$ varies between trajectories, and all other parameters and initial conditions are constant and known. In this case, the marginal $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t})$ remains unknown, the conditional, $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t}, l)$ is just characterized by the observational noise, as described earlier. In VAEs this procedure is modelled by the decoder as $P(\mathbf{x}_{t+n}|\mathbf{z}_{<t})$. Disentanglement allows the separation of the latent vector in two parts i) $\mathbf{z}_{<t}$ that captures the dynamics and ii) $\mathbf{z}_l$ that encodes the pendulum length. This leads to a conditional distribution $P(\mathbf{x}_{t:t+n}|\mathbf{z}_{<t}, \mathbf{z}_l)$ which better resembles the functional structure of the real conditional distribution above.

3.3. Definition of OOD dataset

The evolution of a dynamical system is defined by far and foremost by the form of $\mathcal{F}$ in Equation (1). Considering $\mathcal{F}$ given, the next most important factor that characterizes the distribution of trajectories in the system is the values of the parameters $\boldsymbol{\xi}$. If these parameters come from a distribution $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$, the trajectories of states that the system can follow will form another distribution $\mathbf{u}_{\leq t} \sim P(\mathbf{u}_{\leq t} \mid \boldsymbol{\xi})$, where $\mathbf{u}_{\leq t}$ is the evolution of the system states from the start-up to time t. Given the nature of dynamical systems, different parameters can produce widely different trajectories in state space (Lai & Winslow, 1994) so it is reasonable to assume that changes in the parameter distribution will affect the trajectory distribution as well. For our systems, we additionally verify this by visually inspecting the trajectories produced by each parameter distribution (see Appendix A.1). Here, we define an OOD dataset as a dataset comprising a set of trajectories derived from a parameter distribution that is

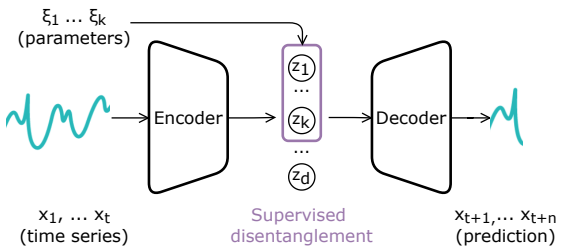


Figure 2: **The SD-VAE model** Taking as input t observations of the phase space, the model predicts the state in future time-steps in one pass. The model is trained with an additional loss term over some of the latent, which makes the representation more disentangled.

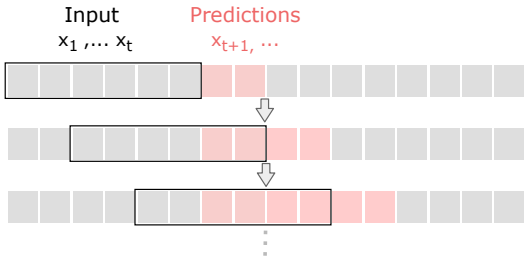


Figure 3: **Autoregressive prediction** By using model autoregressively on its own predictions we can derive an arbitrarily long horizon.

disjoint from the one used to generate the trajectories of the training dataset. For each system, we draw the parameters $\boldsymbol{\xi}$ from a uniform distribution which is the same for the training, validation and test sets. These three datasets are considered in-distribution. Furthermore, we create two additional datasets using different parameter distributions. The support of these distributions is disjoint from the previous distribution and with each other. We name these datasets OOD-Easy and OOD-Hard. Figure 1 illustrates the distribution of lengths of the pendulum datasets.

Capturing the whole distribution of trajectories in a single training set is unrealistic (Fotiadis et al., 2020) and for learning models with robust OOD prediction, some extra inductive biases are needed (Bird & Williams, 2019; Barber et al., 2021). In our method, this inductive bias comes by leveraging the ground-truth parameters to disentangle the latent representation. For the observation-space pendulum experiments, we extend the notion of parameters to include the initial conditions (boundary conditions could also be added). A detailed description of the datasets is provided in Table 3 of the Appendix.

4. Methods

4.1. Variational Autoencoders (VAEs)

VAEs (Kingma & Welling, 2014) offer a principled approach to latent variable modeling. It combines an encoder $Q_\phi(z|\boldsymbol{x})$ which takes the data x as input and infers the latent representation z, with a generative decoder $P_\theta(\boldsymbol{x}|z)$ that project the representation back to the data space. The encoder and decoder are parameterized by neural networks which makes the computation of the marginal likelihood prohibitively expensive. Training is, thus, done with approximate inference, i.e., maximizing the evidence lower bound (ELBO) of the marginal over the data.

$$\begin{aligned}\mathcal{L}_{\phi,\theta}(\mathbf{x}) = &\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log P_\theta(\mathbf{x} \mid \mathbf{z})\right] \\ &- D_{KL}\left(Q_\phi(\mathbf{z} \mid \mathbf{x})||P(\mathbf{z})\right)\end{aligned} \tag{5}$$

In the standard formulation, the ELBO consists of a reconstruction loss and the Kullback-Leibler divergence between the posterior distribution $Q_\phi(\mathbf{z} \mid \mathbf{x})$ and a prior $P(\mathbf{z})$.

4.2. Disentangling VAEs for modelling dynamics

In theory, disentanglement in VAEs can be also achieved in an unsupervised way. Choices include using a prior with uncorrelated variables like the standard normal, adding a weighting factor on the KL divergence term of the loss (Higgins et al., 2018) or constraining the size of the latent space to coincide with the factors of variation in the data (Iten et al., 2020). As Locatello et al. (2020a) has shown, unsupervised disentanglement only works if there are biases in the data to exploit. Supervised disentanglement is possible when information about the factors of variation of the data is available. Using the ground-truth values of simulated images to disentangle VAE representations improves image generation quality (Locatello et al., 2019).

We extend the idea of supervised disentanglement to the context of modelling dynamics. In our datasets, each trajectory is accompanied by the parameters $\boldsymbol{\xi}$ of the ODE that were used to produce it. We treat those parameters as factors of variation in the data and use them to enforce a structure on the latent space using constrained optimization. Under the Karush-Kuhn-Tucker conditions, we can rewrite the constraint in the Langragian form and obtain the regularization term $\mathcal{L}_{\boldsymbol{\xi}}(\mathbf{z}_{1:N_\xi}, \boldsymbol{\xi})$, between the ground truth parameters $\boldsymbol{\xi} \in \mathbb{R}^{N_\xi}$ and the output of the first N_ξ latents of the VAE, $\mathbf{z}_{1:N_\xi}$. We discuss the choice of the regression term $\mathcal{L}_{\boldsymbol{\xi}}$ in Section 5. Extending the original VAE to generate predictions instead of reconstructions, is also needed. To accommodate for this, we change the reconstruction term in Equation (5) to a prediction term $\log P_\theta\left(\mathbf{x}_{t<,\leq t+n} \mid \mathbf{z}\right)$, leading to the the final training objective:

$$\begin{aligned}\mathcal{L}_{\phi,\theta}(\mathbf{x}_{\leq \mathbf{t}}) = &\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x}_{\leq t})}\left[\log P_\theta(\mathbf{x}_{t<,\leq t+n} \mid \mathbf{z})\right. \\ &\left.+ \delta\ \mathcal{L}_{\boldsymbol{\xi}}(\mathbf{z}_{1:N_\xi}, \boldsymbol{\xi})\right] \\ &- \beta D_{KL}\left(Q_\phi(\mathbf{z} \mid \mathbf{x}_{\leq t})||P(\mathbf{z})\right)\end{aligned} \tag{6}$$

Where t is the length of the input and n of the predicted output and we drop the dependence of $\mathbf{z}$ on $\mathbf{t}$ to simplify the notation. To allow more flexibility between prediction and disentanglement, both the KLD and regression terms are weighted by tunable parameters (β and δ respectively). Weighting the prediction term can also be seen as tuning the decoder variance. A more elaborate derivation of the objective can be found in Appendix C.1 We refer to this model as **SD-VAE**. A schematic of the architecture can be seen in Figure 2.

Both the VAE and SD-VAE can produce arbitrarily-long predictions by re-feeding the model predictions back as input (Figure 3). This autoregressive approach has been shown to work well in problems like wave propagation and weather forecasting (Fotiadis et al., 2020; Lam et al., 2022).

4.3. Disentanglement of dynamics in observation-space

We investigate how disentanglement affects modelling of dynamics when the state of the system is not accessible directly but it inferred from high-dimensional observations like image sequences. In this case, the state of the system $\mathbf{u}_t \in R^d$ is mapped to a high-dimensional rendering $\mathbf{f}_t$. In our dataset $\mathbf{f}_t \in \mathbb{R}_{\geq 0}^{64\times 64}$ and $t \in \mathbb{N}$. The model for this dataset is the Recurrent State Space Model (RSSM) (Hafner et al., 2019). RSSM has been successfully used for planning from pixels and is considered state-of-the-art model in long-term spatiotemporal prediction (Saxena et al., 2021). Furthermore, RSSM is a hybrid model combining deterministic and stochastic components, and this allows us to assess disentanglement outside VAEs. We use the same formulation of the loss function as in the original paper, with the addition of the supervised disentanglement loss, similarly to what we do in Equation (6). Since the RSSM has latent variables for each time-step, we apply a disentanglement loss on all of them. The SD-RSSM loss function can be found in Appendix C.3.

5. Disentangling for system dynamics

In this section we compare models trained to predict the evolution of dynamical systems. The main goal of our experiments is to assess whether supervised disentanglement of VAEs improves the prediction accuracy and if the improvement also transfers to OOD data. To achieve this we compare VAEs with our proposed SD-VAE. Additionally, using quantitative and qualitative approaches, we analyze how latent space supervision affects the representation of VAEs. Next, we try to see if supervised disentanglement works also in deterministic autoencoders (AEs). Lastly, we conduct experiments with LSTMs, a popular recurrent method for low dimensional sequence modelling (Yu et al., 2019). Overall, we train and compare VAE, SD-VAE, AE, SD-AE and LSTM models.

5.1. Datasets

To create the datasets, we use an adaptive Runge-Kutta integrator with a timestep of 0.01 seconds. For every simulated sequence we draw a different combination of parameters. For the pendulum simulations we randomly draw the initial angle θ from a uniform distribution $10° - 170°$, the angular velocity ω is always 0. For the Lotka-Volterra and 3-body system, the initial conditions are always the same to avoid pathological trajectories. Dataset details can be found in Appendix A.1.

5.2. Models and training

Choices for the VAE models We use the same model choices for both VAE and SD-VAE. Our prior is an isotropic Gaussian $P(z) = \mathcal{N}(z \mid \mathbf{0}, \boldsymbol{I})$ which helps to disentangle the learned representation (Higgins et al., 2017). To get a closed form KL-divergence term, we use a Gaussian with diagonal covariance as the approximate posterior distribution $q_\phi(z \mid \boldsymbol{x}) = \mathcal{N}(z \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, a common practical choice (Kingma & Welling, 2014). The decoder has a Laplace distribution $p_\theta(\boldsymbol{x} \mid z) = \text{Laplace}(\boldsymbol{x} \mid \boldsymbol{\mu}_x, \boldsymbol{I})$ which is equivalent to using a L_1 prediction loss. Preliminary experiments showed that L_1 loss works better than L_2. This is not unexpected, since L_1 is known to provide crisper results in image modelling (Mathieu et al., 2019) and has also been used in time-series forecasting (Tang & Matteson, 2021). The covariance of the decoder is constant and isotropic. No scaling of the covariance is needed since we weight the KLD term in Equation (6).

Choices for the supervised disentanglement term For the regression loss we chose the L_1 loss, corresponding to a Laplacian prior with mean $\boldsymbol{\xi}_i$ and unitary covariance. This choice was driven by preliminary experiments with various loss functions. Using a standard normal prior pulls the latents to be close to 0 but this comes at odds with the disentanglement loss term because the target parameters can have larger values. To alleviate this issue we scale the parameters $\boldsymbol{\xi} \in [0, 1]$. We find that linear scaling offers some small improvement and we use it throughout our experiments (details in Appendix C.2).

Training Early experiments revealed significant variance in the performance of the models, depending on hyperparameters. With this in mind, we take various steps to make model comparisons as fair as possible. Firstly, all models have similar capacity of neurons. Both the VAE and AE have an encoder with two hidden layers of sizes 400 and 200 respectively and a reverse decoder. The LSTM model has two stacked LSTM cells with a hidden size of 100, which results in an equivalent number of learned parameters. We tune the hyperparameters of each method using grid-search and train the same number of models for each method to avoid favouring one over the others by chance. To further reduce

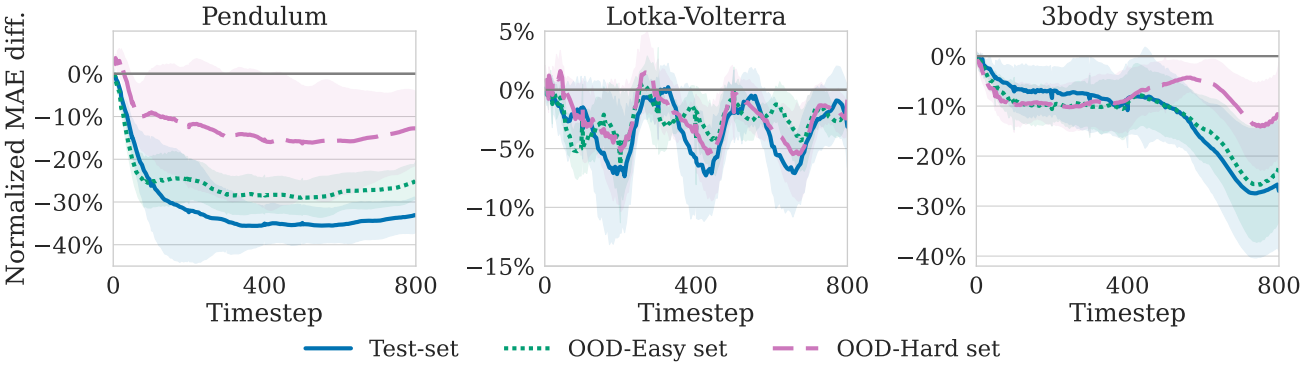


Figure 4: **Disentangled VAE (SD-VAE) vs VAE**. Normalized MAE difference between the SD-VAE and the plain VAE. Negative values indicate that SD-VAE has lower error. We plot the mean and one standard deviation interval of the best 5 models (selected based on the cummulative MAE over a validation set)

Table 1: **MAE averaged over 800 steps.** Mean of the best 5 models that were selected by validation MAE. SD-VAE outperforms VAE and the other models. LSTM diverged during testing on Lotka-Volterra.

	Pendulum			Lotka-Volterra			3-body system		
	Test-set	OOD-Easy	OOD-Hard	Test-set	OOD-Easy	OOD-Hard	Test-set	OOD-Easy	OOD-Hard
LSTM	0.829	1.318	1.985	–	–	–	0.061	0.082	0.099
MLP	0.635	1.097	1.420	0.103	0.140	0.157	0.064	0.079	0.093
SD-MLP	0.687	1.088	1.442	0.104	0.141	0.157	0.053	0.067	0.084
VAE	0.673	1.128	1.386	0.104	0.142	0.159	0.060	0.075	0.089
SD-VAE	**0.443**	**0.819**	**1.185**	**0.100**	**0.138**	**0.155**	**0.048**	**0.062**	**0.080**

statistical chance, we conduct large-scale experiments training overall 1200 models which required more than 5,000 CPU-hours. Details for the hyperparameters and number of experiments can be found in Appendix D.1.

5.3. Long-term and OOD prediction accuracy

We compare the prediction accuracy of VAE and SD-VAE on the three dynamical systems described in Section 3.1 and for each system we compare on three datasets: the in-distribution test-set, which shares the same parameter distribution with the training set, and the OOD-Easy and OOD-Hard sets which represent an increasing distribution shift from the training data. Models are compared using the Mean Absolute Error (MAE) between prediction and ground truth, a widely used metrics for sequence prediction problems (Girin et al., 2020), that was also used for training. Models are used in an autoregressive manner (Section 4.2) to produce long-term predictions of 800 timesteps. We consider this to be sufficiently long-term since it is 20 times longer than the output of a forward pass. We predict up to 800 timesteps because the simulated trajectories are of 1000 steps long and we reserve the first 200 timesteps to randomly select a starting point for the input. To account for the variability in model training, we provide the mean and standard deviation computed for the 3 best models of each method. The best models are selected based on average validation MAE.

Results (Figure 4 & Table 1) indicate that SD-VAE offers a substantial and consistent improvement over the VAE across all 3 dynamical systems and datasets with a reduction in error that surpasses 30% in the pendulum system. In both the pendulum and 3-body system the improvement is mostly increasing for long-term predictions indicating that SD-VAE captures better the system dynamics. While the accuracy of both models deteriorates in the OOD-Easy and OOD-Hard set (details in Figure 13 of the Appendix), SD-VAE still outperforms the VAE. This is an indication that the disentanglement of domain parameters can be a useful inductive bias for OOD generalization. Overall, results show that SD-VAE forecasts more accurately both long-term and OOD, indicating that supervised disentanglement helps the model to better capture the system dynamics.

5.4. Disentanglement of representations

We want to understand if latent space supervision leads to differences in the learned representations of VAEs and SD-VAEs. For this, we use various metrics to quantify disentanglement. Measuring disentanglement is a challenging task;

Table 2: **SD-VAE exhibits stronger disentanglement properties than the plain VAE according to widely used metrics**. Scores are averages over the best 3 models (selected by validation accuracy).

	Pendulum		Lotka-Volterra		3 body system	
	VAE	SD-VAE	VAE	SD-VAE	VAE	SD-VAE
Disentanglement	-	-	0.27	**0.53**	0.20	**0.90**
Completeness	0.17	**0.90**	0.20	**0.57**	0.13	**0.90**
Informativeness	0.94	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**
SAP	0.03	**0.87**	0.04	**0.21**	0.01	**0.67**
MIG	0.01	**0.17**	0.00	**0.03**	0.00	**0.08**

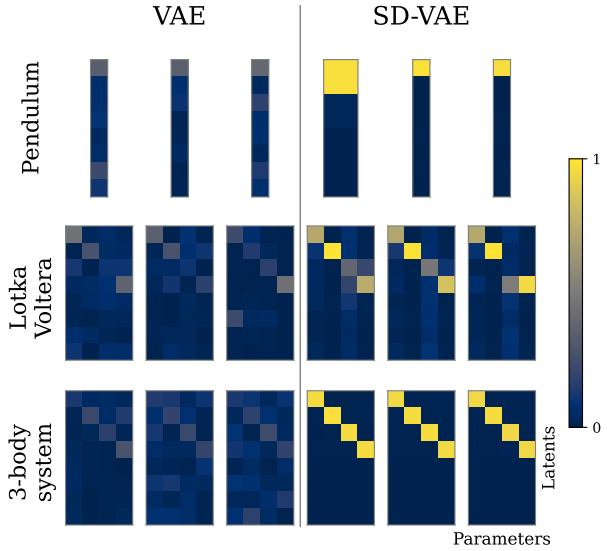


Figure 5: **Disentanglement of representations.** The x-axis corresponds to the parameters and the y-axis to the latent various. The color scale denotes the value of the importance weights. These values were extracted from the weights of a regressor trained to predict the parameters from the latent values. High values (yellow) indicate that the latent variable that has high predictive power over the ground-truth value. We present the top 3 models for each method and dataset. The latent space of the SD-VAE is disentangled in highly-predictive and non-predictive parts, while the VAE encoding exhibits no such characteristic.

many metrics have been proposed that do not always correlate well with each other (Locatello et al., 2020a). In this work we use *Disentanglement*, *Completeness*, *Informativeness* (DCI) (Eastwood & Williams, 2018), a predictor-based measuring frameworks that analyses three different aspects of disentanglement. Briefly, *Disentanglement* measures how well the factors of variation are factorized in the representation, *Completeness* indicates if each factor is captured by a single latent variable, and *Informativeness* quantifies the amount of information a representation captures about the factors of variation. Recent studies suggest that in practice DCI with random forests is "the best all-around metric" (Zaidi et al., 2020). For completeness, we additional include Mutual Information Gap (MIG) (Chen et al., 2018) an information-theoretic metric that quantifies disentanglement as the difference between the mutual information (MI) between the top two latent-factor pairs, and the Attribute Predictability Score (SAP) (Kumar et al., 2018) a metric that works similarly to MIG but uses the importance weight of a learned predictor instead of MI.

Metrics in Table 2 indicate that SD-VAE produces more disentangled representations than the VAE in all the systems. Specifically, we observe a significant increase in *Disentanglement*, *Completeness* and SAP scores and a more modest increase in MIG. We also observe that the *Informativeness* of both VAE and SD-VAE is close to the maximum (1), this suggests that the representation of the VAE also captures information about the parameters but this is spread across the latent dimensions. *Disentanglement* can not be computed for the pendulum since there is only one factor of variation (length).

To compute the DCI metrics, we train a boosted trees regressor to predict the parameters from the latent codes (on the training dataset). The importance weights of the learned regressor demonstrate the predictive power of each latent for each parameter. We visualize the weights of the best SD-VAE and VAE models in Figure 5. We use the best 3 models as before (selected by validation MAE). To allow better visual inspection we keep the first 8 latents. To further facilitate the comparison for the VAE we place the highest value of each column at the top diagonal positions ([1,1], [2,2] etc). We provide visualizations of the full latent space with importance weights. We observe that the supervised latents of the SD-VAE have very high predictive power for the system parameters, while the other latents are not significant. In the case of VAE, the predictive power is spread across the whole latent code. In conjunction with the disentanglement metrics, these findings demonstrate that latent space supervision produces highly disentangled representations.

5.5. Linearity of correlation

The importance weights in the previous section denote the strong correlation between latents $\mathbf{z}$ and parameters $\boldsymbol{\xi}$. Since trees can capture both linear and non-linear dependencies, the nature of the relationship remains, unclear. To quantify the linearity, we fit linear regression models for each z_i, ξ_j pair and compute the absolute Pearson correlation coefficient between the two variables. Pearson r values are visualized in Figure 15. Results (see Appendix E.1) indicate that the relationship between supervised latents and parameters and is strongly linear in most cases. This aligns with our experimental findings that linear scaling works best for the disentanglement loss (see Appendix C.2). We exploit this linearity to perform traversals of the latent space in the next section.

5.6. Latent space traversals

Being able to traverse between two points in the latent space is a property that indicates meaningful representations. Interpolation in latent space can produce meaningful images in properly disentangled VAEs (Higgins et al., 2017). While images have easily recognizable visual components, traversals of dynamical systems are harder to portray. Here we study whether interpolating between two points in the latent space of SD-VAE can produce meaningful trajectories. For this, we create a new pendulum dataset containing 100 trajectories with linearly spaced length in the range $l \in [1.0 - 1.5]$. The initial conditions are kept constant ($\theta = \frac{pi}{2}, \omega = 0$) for all the trajectories to facilitate comparisons . We use the encoder of our best SD-VAE model to extract the latent variables for each trajectory. For each trajectory, the encoder produces 4 latent variables $z_1 \dots z_4$. We linearly interpolate between the latents of the two extreme trajectories ($l = 1.0$ and $l = 1.5$), driven by our findings in Appendix E.1 that latents and parameters have a higly linear correlation. Next, we feed the real and interpolated latents to the decoder and predict up to 1000 timesteps. We find that the total mean absolute error between prediction and ground truth is 0.29 with the real latents and 0.33 with the interpolated one. These results indicate that linear latent space interpolation produces meaningful latent codes. This is further corroborated by plotting the real and interpolated latents together (Figure 3). The relationship between the real latents z_i and pendulum length l is highly linear, which further explains with the linear interpolation method works well.

5.7. Disentangling AEs and stability

We pose the question of whether supervised disentanglement can also be applied to (deterministic) AEs. For this, we train both AE and SD-AE models and compare them with VAE and SD-VAE models (Table 1). Results indicate that disentangling AEs does not offer much if any improvement in prediction accuracy. Probabilistic models seem better suited to capture the variation in the data distribution. It also illustrates that latent space disentanglement is not trivial and more work is needed to help us understand what works in practice and why.

We also trained LSTMs and found that their prediction accuracy is subpar compared to the other models. In fact for the Lotka-Volterra system, LSTMs proved to be very unstable: none of the 72 trained LSTMs could predict long-term (800 steps) without diverging. On the note of model stability, this is something to take into consideration when using supervised disentanglement in practice. In AEs supervised disentanglement resulted in a higher percentage of unstable models. SD-VAE was the most stable model in the pendulum and 3-body systems with more than 90% of the models being stable, but in the Lotka-Volterra systems the VAE training produced more stable models.

6. Modelling dynamics in observation-space

We extend the idea of supervised disentanglement to models that infer the state from high-dimensional observations such as image sequences.

6.1. Datasets

The dynamical system we use in this experiment is the swinging pendulum, a common benchmark for modelling dynamics in image sequences (Brunton et al., 2020; Barber et al., 2021). The data set contains sequences of images of a moving pendulum. The positions of the pendulum are first computed by a numerical simulator and then rendered in image space as frames of dimension 64×64. The length of the pendulum ℓ, the strength of gravity g and the initial conditions (position θ, angular velocity ω) are set to different values in each trajectory so they differ from each other. Parameters are drawn from a uniform distribution. For the OOD sets we change the distributions of length l, and gravity g but keep the same distribution of θ and ω as in training. More details about the data set are illustrated in Appendix A.3. For the simulations, we use an adaptive Runge-Kutta integrator with a timestep of 0.05 seconds.

6.2. Model and Training

In this experiment, we use RSSM described in Section 4.3. RSSM is a generative model including both stochastic and deterministic latents. We use supervised disentanglement on the stochastic part, and term that model SD-RSSM. The RSSM model we use follows the architecture as described in (Hafner et al., 2019) & (Saxena et al., 2021). Disentanglement is applied to all four parameters (length ℓ, gravity g, initial position θ and velocity ω), but only length and gravity

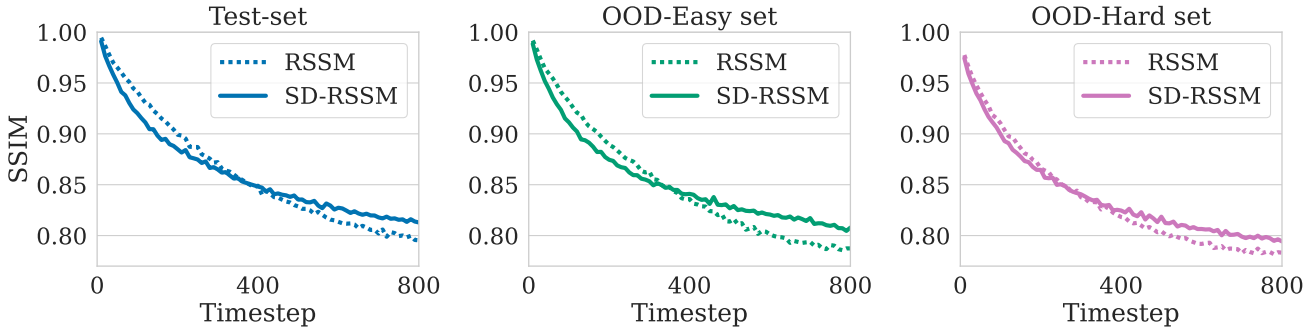


Figure 6: **Prediction quality on the observation-space pendulum.** Structural Similarity (SSIM) as a function of the predicted timestep. The disentangled SD-RSSM model seems more robust on long-term predictions.

vary between datasets. For training, we use sequences of 100 frames and batch size 100. We use an L_2 loss for the disentanglement term because preliminary results showed it performs better than L_1 and BCE. During testing the model uses 50 frames as context. We train 24 RSSM and 24 SD-RSSM models (detailed hyperparameters in Appendix D.2).

6.3. Results

We compare the predictions of RSSM and SD-RSSM using structural similarity (SSIM), a metric that takes into account the qualitative characteristics of the image, something that pixel-wise metrics like MSE and MAE fail to do (Wang et al., 2004). We select the best RSSM and SD-RSSM models based on the average SSIM over the validation set and plot the SSIM as a function of the timestep (Figure 6) and is widely used for dynamical system prediction from spatiotemporal data (Pant & Farimani, 2020). Results show that while for the short-term predictions, the RSSM has higher SSIM, in long-term prediction after about 350 steps SD-RSSM is performing better, in all 3 datasets. Furthermore, as we move from the test-set to the OOD sets, we observe that the SD-RSSM model closes the performance gap in the short-term prediction. Specifically, in the OOD-Hard for predictions up to around 350 step it is almost equivalent to RSSM while at the same time maintaining its long-term (>350) advantage. We hypothesize that SD-RSSM has better long-term performance due to less overfitting to the relatively short training horizon. Similarly, the robustness in increasing distribution shifts could also be explained by less overfitting on the parameters of the training data. We also compared using the peak signal-to-noise ratio (PSNR), drawing similar conclusions (details in Appendix F). Qualitative results show that both models produce accurate short time predictions and also accurately capture the appearance of the pendulum even in long-term predictions. Where they differ is in how well they capture the long-term dynamics indicating that latent space disentanglement is helpful for long-term prediction. Overall, results suggest that supervised disentanglement can be used to model dynamical systems in observation-space sequences, resulting in improved long-term and OOD performance.

7. Conclusion

We have shown that using ground-truth parameters to supervise the latent space of VAEs encourages them to learn more disentangled and interpretable representations while at the same time increasing their prediction accuracy and OOD generalization in three dynamical systems. We have, further, shown that supervised disentanglement improves generative models like RSSM trained on observation-space data of a swinging pendulum and leads to better long-term forecasting performance and robustness to OOD shifts. These results make supervised disentanglement an attractive choice for the generative modelling of system dynamics. In practice, VAE and SD-VAEs should be preferred over their deterministic counterparts. Using simulated data makes the label collection cheap but this is not always possible. Extending our method to the semi-supervised setting, i.e., supervising with few labels, is important for real-world applications where the collection of labels is more expensive but robust prediction of system dynamics remains critical. Further analysis of the method using systems with more complex dynamics is also an important avenue for future work.

Acknowledgements

M.L. and S.F. acknowledge financial support from the Departments of Aeronautics and Bioengineering respectively.

References

Ayyad, A., Chehadeh, M., Awad, M. I., and Zweiri, Y. Real-time system identification using deep learning for linear processes with application to unmanned aerial vehicles. *IEEE Access*, 8:122539–122553, 2020. doi: 10.1109/ACCESS.2020.3006277.

Barber, G., Haile, M. A., and Chen, T. Joint Parameter

Discovery and Generative Modeling of Dynamic Systems. *ArXiv preprint*, abs/2103.10905, 2021. URL https://arxiv.org/abs/2103.10905.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35: 1798–1828, 08 2013. doi: 10.1109/TPAMI.2013.50.

Bird, A. and Williams, C. K. Customizing sequence generation with multitask dynamical systems. *arXiv*, 2019. ISSN 23318422.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020. ISSN 0066-4189. doi: 10.1146/annurev-fluid-010719-060214.

Carbonneau, M.-A., Zaidi, J., Boilard, J., and Gagnon, G. Measuring Disentanglement: A Review of Metrics. *ArXiv preprint*, abs/2012.09276, 2020. URL https://arxiv.org/abs/2012.09276.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating Sources of Disentanglement in Variational Autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/1ee3dfcd8a0645a25a35977997223d22-Paper.pdf.

Eastwood, C. and Williams, C. K. I. A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Fotiadis, S., Pignatelli, E., Lino Valencia, M., Cantwell, C. D., Storkey, A., and Bharath, A. A. Comparing recurrent and convolutional neural networks for predicting wave propagation. In *ICLR 2020 Workshop on Deep Differential Equations*, 2 2020. URL http://arxiv.org/abs/2002.08981.

Fraccaro, M., Kamronn, S., Paquet, U., and Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3601–3610, 2017.

Garg, S., Rasp, S., and Thuerey, N. Weatherbench probability: Medium-range weather forecasts with probabilistic machine learning methods. In *EGU General Assembly Conference Abstracts*, pp. EGU21–11448, 2021.

Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., and Alameda-Pineda, X. Dynamical Variational Autoencoders: A Comprehensive Review. *ArXiv preprint*, abs/2008.12595, 2020. URL https://arxiv.org/abs/2008.12595.

Goyal, A. and Bengio, Y. Inductive Biases for Deep Learning of Higher-Level Cognition. *ArXiv preprint*, abs/2011.15091, 2020. URL https://arxiv.org/abs/2011.15091.

Hafner, D., Lillicrap, T. P., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Higgins, I., Sonnerat, N., Matthey, L., Pal, A., Burgess, C. P., Bosnjak, M., Shanahan, M., Botvinick, M., Hassabis, D., and Lerchner, A. SCAN: learning hierarchical compositional visual concepts. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rkN2Il-RZ.

Iten, R., Metger, T., Wilming, H., del Rio, L., and Renner, R. Discovering physical concepts with neural networks. *ArXiv preprint*, abs/1807.10300, 2018. URL https://arxiv.org/abs/1807.10300.

Iten, R., Metger, T., Wilming, H., del Rio, L., and Renner, R. Discovering Physical Concepts with Neural Networks. *Physical Review Letters*, 124(1), 2020. ISSN 0031-9007. doi: 10.1103/physrevlett.124.010508.

Jacobsen, C. and Duraisamy, K. Disentangling Generative Factors of Physical Fields Using Variational Autoencoders. *Frontiers in Physics*, 10:536, 2022. ISSN 2296424X. doi: 10.3389/FPHY.2022.890910/BIBTEX.

Kim, H. and Mnih, A. Disentangling by factorising. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2654–2663. PMLR, 2018. URL http://proceedings.mlr.press/v80/kim18b.html.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

Lai, Y.-C. and Winslow, R. L. Extreme sensitive dependence on parameters and initial conditions in spatio-temporal chaotic dynamical systems. *Physica D: Nonlinear Phenomena*, 74(3-4):353–371, 1994.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Pritzel, A., Ravuri, S., Ewalds, T., Alet, F., Eaton-Rosen, Z., et al. Graphcast: Learning skillful medium-range global weather forecasting. *ArXiv preprint*, abs/2212.12794, 2022. URL https://arxiv.org/abs/2212.12794.

Li, Y. and Mandt, S. Disentangled sequential autoencoder. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5656–5665. PMLR, 2018. URL http://proceedings.mlr.press/v80/yingzhen18a.html.

Li, Y., Lu, X., Wang, Y., and Dou, D. Generative time series forecasting with diffusion, denoise, and disentanglement. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23009–23022. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/91a85f3fb8f570e6be52b333b5ab017a-Paper-Conference.pdf.

Li, Y., Lu, X., Wang, Y., and Dou, D. Generative Time Series Forecasting with Diffusion, Denoise, and Disentanglement. *ArXiv preprint*, abs/2301.03028, 2023. URL https://arxiv.org/abs/2301.03028.

Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., and Bachem, O. Disentangling Factors of Variation Using Few Labels. *ArXiv preprint*, abs/1905.01258, 2019. URL https://arxiv.org/abs/1905.01258.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. A sober look at the unsupervised learning of disentangled representations and their evaluation. *Journal of Machine Learning Research*, 21:1–62, 2020a. ISSN 15337928.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359. PMLR, 2020b. URL http://proceedings.mlr.press/v119/locatello20a.html.

Lopez, R. and Atzberger, P. J. GD-VAEs: Geometric Dynamic Variational Autoencoders for Learning Nonlinear Dynamics and Dimension Reductions. *ArXiv preprint*, abs/2206.05183, 2022. URL https://arxiv.org/abs/2206.05183.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. Disentangling disentanglement in variational autoencoders. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4402–4412. PMLR, 2019. URL http://proceedings.mlr.press/v97/mathieu19a.html.

Mathieu, M., Zhao, J., Sprechmann, P., Ramesh, A., and LeCun, Y. Disentangling factors of variation in deep representations using adversarial training. *ArXiv preprint*, abs/1611.03383, 2016. URL https://arxiv.org/abs/1611.03383.

Miladinović, Ä., Gondal, M. W., Schölkopf, B., Buhmann, J. M., and Bauer, S. Disentangled State Space Representations. *ArXiv preprint*, abs/1906.03255, 2019. URL https://arxiv.org/abs/1906.03255.

Pant, P. and Farimani, A. B. Deep learning for efficient reconstruction of high-resolution turbulent dns data. *arXiv*, 2020. doi: 10.48550/ARXIV.2010.11348. URL https://arxiv.org/abs/2010.11348.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 10902716. doi: 10.1016/j.jcp.2018.10.045.

Saxena, V., Ba, J., and Hafner, D. Clockwork Variational Autoencoders. *ArXiv preprint*, abs/2102.09532, 2021. URL https://arxiv.org/abs/2102.09532.

Schölkopf, B. Causality for Machine Learning. *ArXiv preprint*, abs/1911.10500, 2019. URL https://arxiv.org/abs/1911.10500.

Takeishi, N. and Kalousis, A. Physics-Integrated Variational Autoencoders for Robust and Interpretable Generative Modeling. *Advances in Neural Information Processing Systems*, 34:14809–14821, 2021.

Tang, B. and Matteson, D. S. Probabilistic Transformer For Time Series Analysis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 23592–23608. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c68bd9055776bf38d8fc43c0ed283678-Paper.pdf.

Tschannen, M., Bachem, O., and Lucic, M. Recent Advances in Autoencoder-Based Representation Learning. *ArXiv preprint*, abs/1812.05069, 2018. URL https://arxiv.org/abs/1812.05069.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yeo, K., Grullon, D. E., Sun, F.-K., Boning, D. S., and Kalagnanam, J. R. Variational inference formulation for a model-free simulation of a dynamical system with unknown parameters by a recurrent neural network. *SIAM Journal on Scientific Computing*, 43(2):A1305–A1335, 2021.

Yu, Y., Si, X., Hu, C., and Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7):1235–1270, 2019. ISSN 0899-7667. doi: 10.1162/neco_a_01199. URL https://doi.org/10.1162/neco_a_01199.

Zaidi, J., Boilard, J., Gagnon, G., and Carbonneau, M.-A. Measuring Disentanglement: A Review of Metrics. *ArXiv preprint*, abs/2012.09276, 2020. URL https://arxiv.org/abs/2012.09276.

Zhong, W. and Meidani, H. PI-VAE: Physics-Informed Variational Auto-Encoder for stochastic differential equations. *ArXiv preprint*, abs/2203.11363, 2022. URL https://arxiv.org/abs/2203.11363.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. *arXiv*, 2020. ISSN 23318422.

Software and Data

We provide all the necessary code to reproduce our experiments at https://github.com/stathius/sd-vae. The repository contains code and instructions for generating all the datasets and training all the models presented in this work using the hyperparameters that are clearly presented in the paper. This should significantly help others reproduce our experiments. For any further clarifications, readers are encouraged to contact the corresponding author(s).

Accessibility

We have used vector-based figures to increase clarity for zooming-in, a color palette that is easily distinguishable by colorblind people and different line styles. We have also curated arxiv citations to refer to the corresponding conference or journal publications where possible.

A. Datasets

A.1. Phase space

For simulations, we use an adaptive Runge-Kutta integrator with a timestep of 0.01 seconds. Each simulated sequence has a different combination of parameters. Simulation of the pendulum uses an initial angle θ which is randomly between $10^\circ - 170^\circ$ while the angular velocity ω is 0. For the other two systems the initial conditions are always the same to avoid pathological configurations.

A.2. Visualizing dataset shift

Visualizing the distribution shift between datasets is not always straightforward. Especially in cases like dynamical system trajectories where there is usually not much familiarity with their visual representation in comparison for example to natural images. To facilitate qualitative comparisons we we depict the datasets from the three dynamical systems. We provide plots for all the dynamical systems and each dataset in separate figures so that the differences become more apparent. Apart from the phase space diagrams we also provide trajectories across time, offering another way to discern the difference in dynamics.

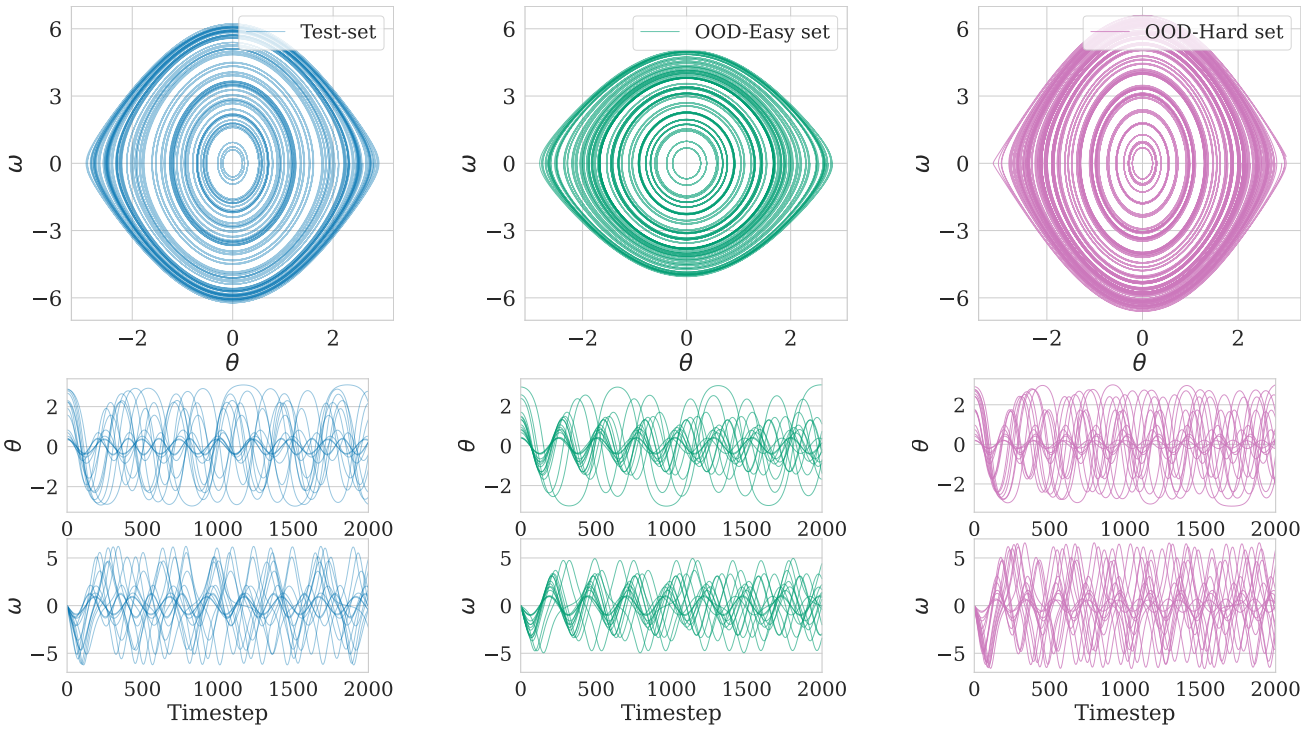


Figure 7: Phase space diagrams **(top)** and evolution over time (**bottom**) for random samples from the pendulum datasets. The OOD-Hard test set exhibits higher variation in the trajectories of θ, ω as can be seen in the bottom row.

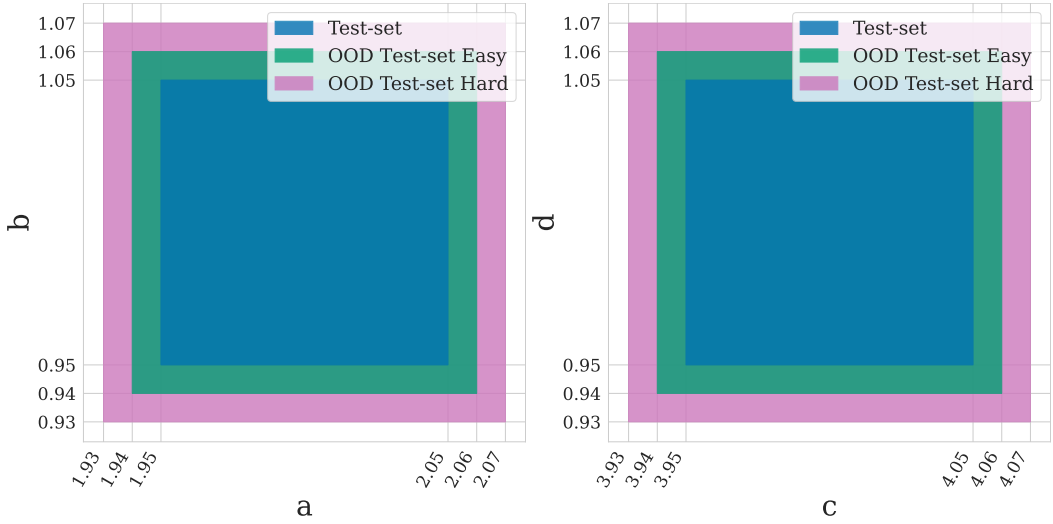


Figure 8: Example illustration of the parameter distribution for the LV test sets. The regions do not overlap, colors represent regions not boundaries. The OOD-Easy test (green) set does not include any of the parameter configurations of the training and original test set (blue). Respectively, the OOD-Hard dataset (magenta) does not include none of the OOD-Easy or the original test set configurations. The parameter space of the blue region is almost half as big at the green area (again without any overlap), signifying a significant OOD shift).

Table 3: **Datasets.** In L-V and 3-body OOD test sets, at least one domain parameter is outside of the parameter range used for training.

	Pendulum	Lotka-Volterra	3-Body
ODEs	$\ddot{\theta} + \frac{g}{\ell}\sin\theta = 0$	$\dot{x} = \alpha x - \beta xy$ $\dot{y} = \delta xy - \gamma y$	$\bar{m}_i \frac{d\vec{v}_i}{dt} = K_1 \sum_j \frac{\bar{m}_i \bar{m}_j}{\bar{r}_{ij}^3} \overrightarrow{r_{ij}}$ $\frac{d\vec{x}_i}{d\bar{t}} = K_2 \vec{v}_i$
Number of ODEs	1	2	6
Independent Variables	θ, ω	x(prey), y(predator)	$\vec{x}_i, \vec{v}_i, i = 1, 2, 3$
Initial values	$\theta \in [10^o - 170^o]$ $\omega = 0$	$x = 5, y = 3$	$\vec{x_1} = (-1, -1), \vec{v_1} = (0.0, 0.5)$ $\vec{x_2} = (1, -1), \vec{v_2} = (0.5 - 0.5)$ $\vec{x_3} = (0, 1), \vec{v_3} = (-0.5, 0.0)$
Timestep	0.01	0.01	0.01
Sequence length	2000	1000	1000
Noise σ^2	0.05	0.05	0.01
Parameters	l(length)	$\alpha, \beta, \gamma, \delta$	K_2, m_1, m_2, m_3
Train/Val/Test	$l \in [1.0 - 1.5]$	$A = \{\alpha \in [1.95, 2.05]\}$ $B = \{\beta \in [0.95, 1.05]\}$ $C = \{\gamma \in [3.95, 4.05]\}$ $D = \{\delta \in [1.95, 2.04]\}$ $\Omega_{\text{train}} = (A \times B \times C \times D)$	$K = \{K_2 \in [1.95, 2.05]\}$ $M1 = \{m_1 \in [1.95, 2.05]\}$ $M2 = \{m_2 \in [1.95, 2.05]\}$ $M3 = \{m_3 \in [1.95, 2.05]\}$ $\Omega_{\text{OOD-Hard}} =$ $(K \times M1 \times M2 \times M3)$
OOD Test Set Easy	$l \in [1.5 - 1.6]$	$A = \{\alpha \in [1.94, 2.06]\}$ $B = \{\beta \in [0.94, 1.06]\}$ $C = \{\gamma \in [3.94, 4.06]\}$ $D = \{\delta \in [1.94, 2.06]\}$ $\Omega_{\text{OOD-Easy}} =$ $(A \times B \times C \times D) \setminus \Omega_{\text{train}}$	$K = \{K_2 \in [1.94, 2.06]\}$ $M1 = \{m_1 \in [1.94, 2.06]\}$ $M2 = \{m_2 \in [1.94, 2.06]\}$ $M3 = \{m_3 \in [1.94, 2.06]\}$ $\Omega_{\text{OOD-Hard}} =$ $(K \times M1 \times M2 \times M3) \setminus \Omega_{\text{train}}$
OOD Test Set Hard	$l \in [0.9 - 1.0]$	$A = \{\alpha \in [1.93, 2.07]\}$ $B = \{\beta \in [0.93, 1.07]\}$ $C = \{\gamma \in [3.93, 4.07]\}$ $D = \{\delta \in [1.93, 2.07]\}$ $\Omega_{\text{OOD-Hard}} =$ $(A \times B \times C \times D) \setminus$ $(\Omega_{\text{train}} \cup \Omega_{\text{OOD-Easy}})$	$K = \{K_2 \in [1.93, 2.07]\}$ $M1 = \{m_1 \in [1.93, 2.07]\}$ $M2 = \{m_2 \in [1.93, 2.07]\}$ $M3 = \{m_3 \in [1.93, 2.07]\}$ $\Omega_{\text{OOD-Hard}} =$ $(K \times M1 \times M2 \times M3) \setminus$ $(\Omega_{\text{train}} \cup \Omega_{\text{OOD-Easy}})$
Number of sequences			
Train/Val/Test		8000/1000/1000	
OOD Test Set Easy		1000	
OOD Test Set Hard		1000	

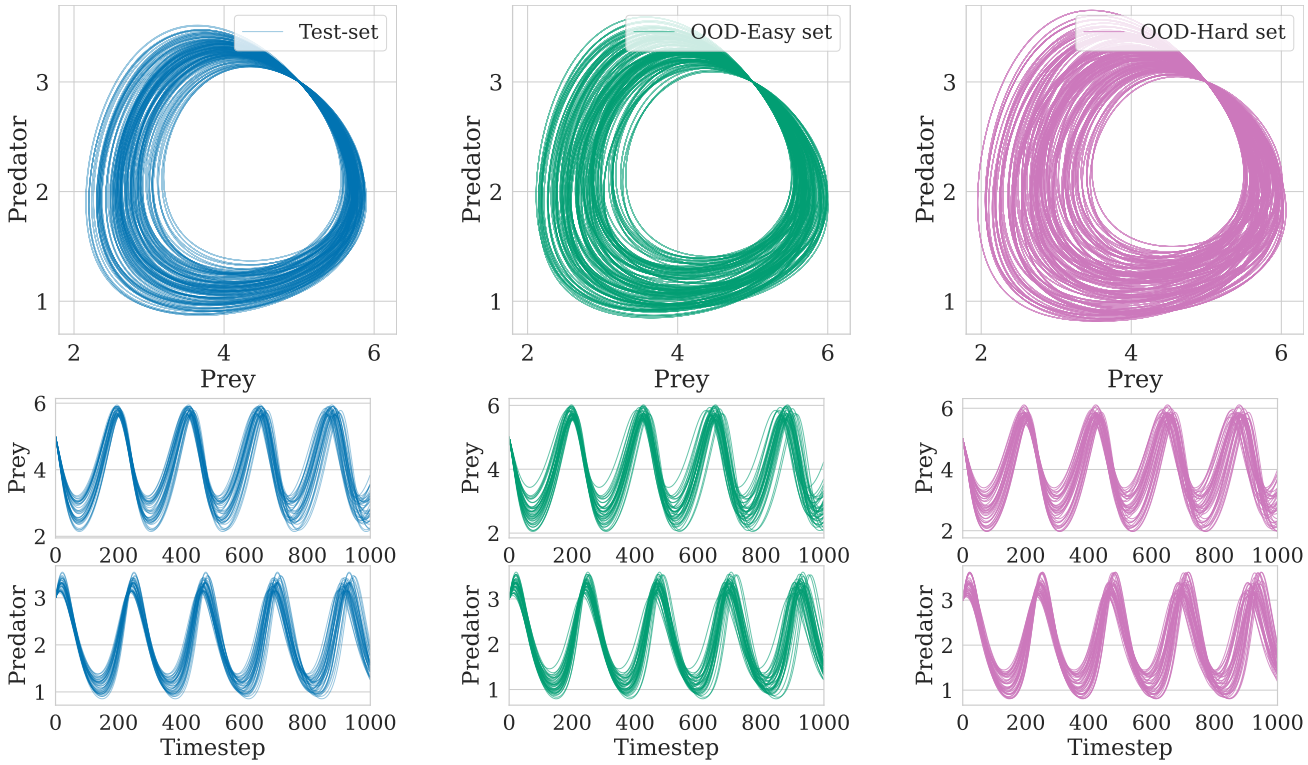


Figure 9: Phase space diagrams **(top)** and evolution over time (**bottom**) for random samples from the Lotka-Volterra datasets. The OOD test sets have an increasingly wider coverage of the domain in the phase-space and time.

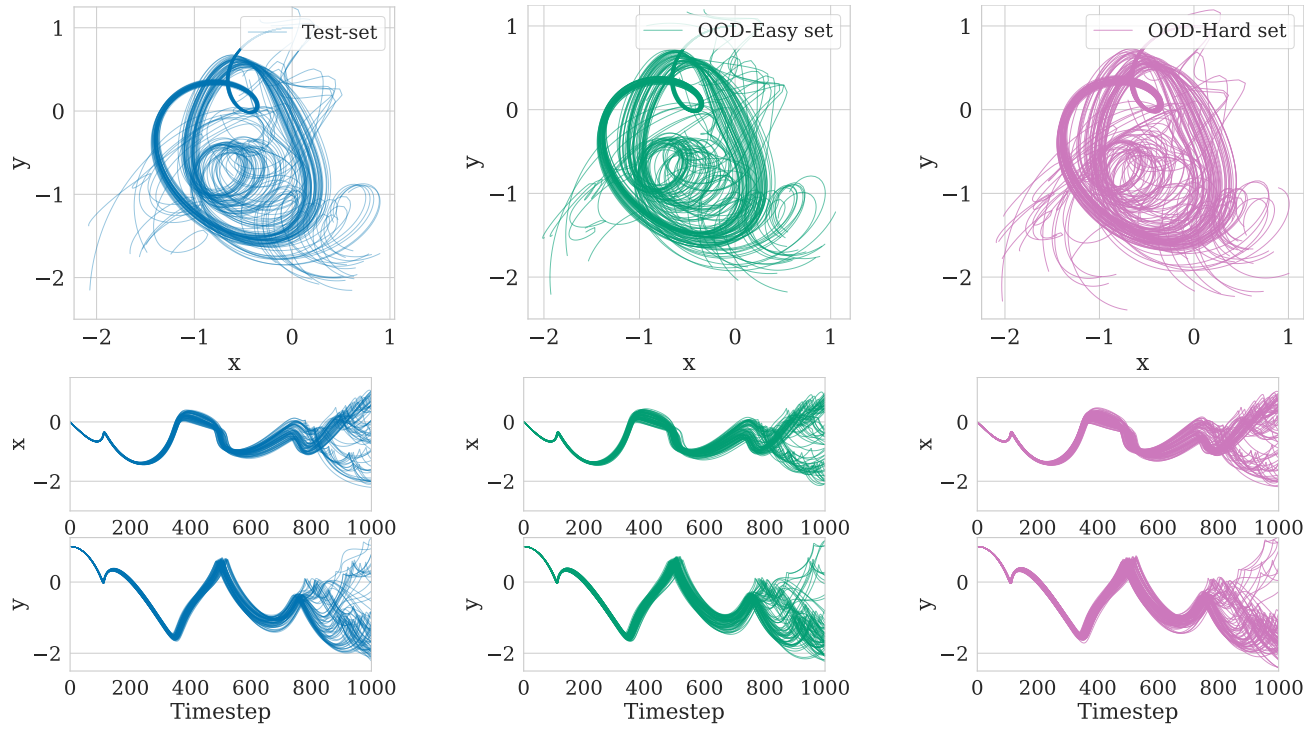


Figure 10: Cartesian coordinates **(top)** and evolution over time (**bottom**) for random samples from the 3-body system datasets. We only plot the first body to avoid cluttering. The OOD test sets include a wider range of possible trajectories. This is evident by the higher coverage of the domain in the cartesian coordinates plot.

A.3. Observation-space

This data set contains image sequences of a moving pendulum under different conditions. The positions of the pendulum are first computed by a numerical simulator and then rendered in pixel space as frames of dimension 64×64. An example image sequence is shown in Figure 11. For the simulations, we use an adaptive Runge-Kutta integrator with a timestep of 0.05 seconds. The length of the pendulum, the strength of gravity and the initial conditions (position, momentum) are set to different values so that each trajectory slightly differs from the others. The initial angle and initial velocity are drawn from the same uniform distribution for all data sets. The initial angle ranges from 30° to 170° and the initial velocity ranges from -2 to 2 rad/s. For training, validation and in-distribution testing set, the gravity fall in the range $8.0 - 12.0$ m^2/s , and the pendulum length lies between $1.20 - 1.40$ m. In the easy OOD testing set, the gravity is between $12.0 - 12.5$ m^2/s and the pendulum length is between $1.40 - 1.45$ m, while in the hard OOD testing set, the gravity is $12.5 - 13.0$ m^2/s and the pendulum length is $1.45 - 1.50$ m. The distributions of these parameters are shown in Figure 12.

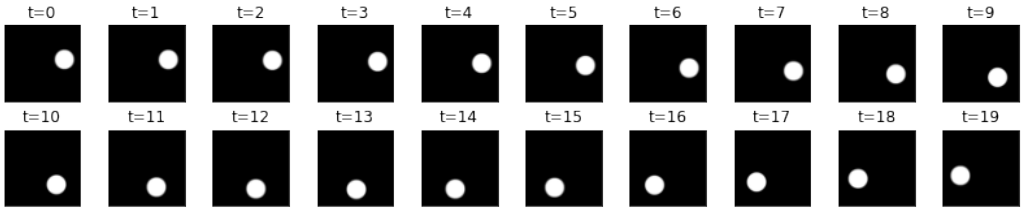


Figure 11: Example image sequence from the observation-space pendulum data set

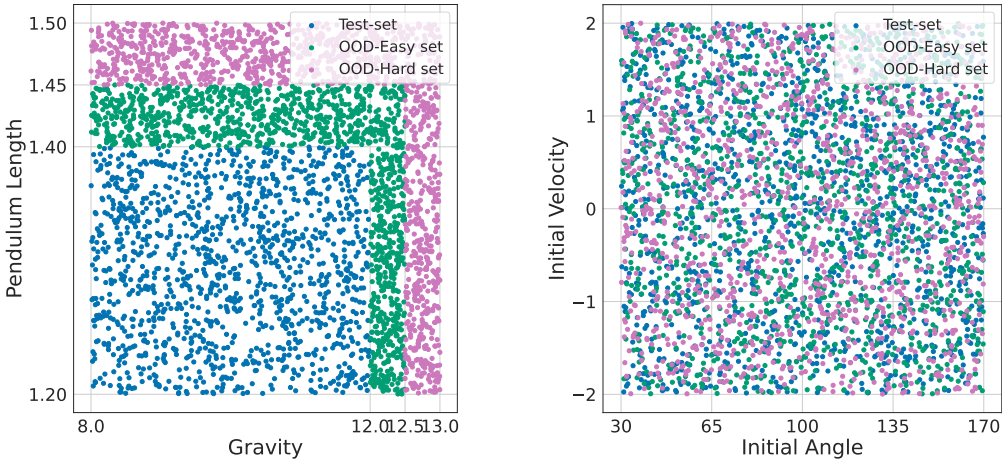


Figure 12: **Parameter distribution for the observation-space pendulum test-sets. Left** For the in-distribution test-set we draw the pendulum length and gravity from the same distribution as during training. The OOD test-sets represent distribution shifts of increasing magnitude, where parameters are drawn from totally different space which has zero overlap with the training and in-distribution test-set. **Right** The initial angle and angular velocity are drawn from the same uniform distribution for all test-sets.

B. Motivating disentanglement

Disentanglement for dynamical system prediction is motivated both from previous experimental results and theoretically.

B.1. Experimental motivation

It is well established that disentangled representations can improve downstream tasks performance and are less prone to overfitting (Bengio et al., 2013). For example, in image generation, disentangled representations enable more controlled synthesis of images with desired attributes while in image reconstruction or inpainting, they can help fill in missing parts of an image while preserving the existing attributes (Higgins et al., 2017; Locatello et al., 2020a).

Disentangled representations can also be beneficial for time-series prediction by separating appearance from the underlying dynamics (Li & Mandt, 2018) or by separating trends, seasonal patterns, noise, and other relevant factors (Li et al., 2022).

Disentanglement for dynamical systems has not been studied as extensively but previous research demonstrated VAEs can fully recover the parameters of a physical system in their latent space (Iten et al., 2018). This is something we also corroborate in this work, where we see that the latent space of the plain VAE contains almost all the information of the system parameters (as indicated by Informativeness on Table 2). This is a strong indication that parameter inference is very important for prediction and models learn it implicitly. Latent space supervision acts as a regularizer, making the parameter inference task explicit. If this leads to disentangled representations, and results in Table 2 suggests it does, the increased prediction performance should be expected.

B.2. Theoretical motivation

Disentanglement can also be motivated by a probabilistic view of the evolution of dynamical systems. We assume a class, C, of a deterministic dynamical system, S_C, parameterized by unobserved parameters $\boldsymbol{\xi}_C$. Our aim is to predict the evolution of the system state, up to some time in future, $t+n$, given a number of observations of the system state up to some point in time t. The initial conditions of the dynamical system, $\boldsymbol{I}_C$, are also unobserved at inference, and constitute a form of uncertainty. Given this form of uncertainty, we can consider the inference problem under a probabilistic framework as estimating the distribution $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t};\boldsymbol{\xi}_C,\boldsymbol{I}_C)$ where $\boldsymbol{\xi}_C,\boldsymbol{I}_C$ are not observed. Our best options for solving the prediction problem are:

- Assign priors on $\boldsymbol{\xi}_C,\boldsymbol{I}_C$, and marginalize over them to obtain an estimate of the marginal $P(\mathbf{x}_{t:t+n} \mid \mathbf{x}_{<t})$
- Estimate $\boldsymbol{\xi}_C$ and $\boldsymbol{I}_C$ and directly model the conditional

Both of these two approaches can be modelled with neural networks, but given the wide nature of divergence in the trajectories of a system for different $\boldsymbol{\xi}_C,\boldsymbol{I}_C$ (they may be considered quasi-chaotic), it is hard to both assign a proper prior and efficiently marginalize. If, on the other hand, the model can identify $\boldsymbol{\xi}_C,\boldsymbol{I}_C$ correctly, we'd be better of with the second modelling choice. Disentanglement can help with better system identification.

To see how disentanglement can be beneficial in this case, we consider the nature of the probability distribution $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t})$ and illustrate it with an example on the simple pendulum. For this example, we assume that $\boldsymbol{\xi}_C = l$ the length of the pendulum, while all other parameters and initial conditions are constant. The marginal $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t})$ remains unknown, but if we condition on pendulum length, $P(\mathbf{x}_{t:t+n}|\mathbf{x}_{<t},l)$ is a Gaussian distribution (since the model is deterministic and we assume Gaussian observation noise). In VAE terms this procedure is modelled by the decoder as $P(\mathbf{x}_{t+n}|\mathbf{z}_{<t})$. With supervised disentanglement we can separate the latent vector in two parts (i) $\mathbf{z}_{<t}$, which captures the dynamics, and (ii) $\mathbf{z}_l$, which captures the information about the pendulum length. This leads to a conditional distribution $P(\mathbf{x}_{t:t+n}|\mathbf{z}_{<t},\mathbf{z}_l)$ which better resembles the functional structure of the real conditional distribution. Assuming that the model is able to capture well the deterministic dynamics after training, this should be a better modelling choice and increase prediction performance.

C. Training objective

C.1. Derivation of SD-VAE loss

VAEs are trained by maximizing the Evidence Lower Bound over the dataset:

$$\text{ELBO} := \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right)\right]$$

We can enforce a structure on the latent space of VAEs using constrained optimization. Rewriting the objective in the Langragian form, under the Karush-Kuhn-Tucker conditions, the constrains become regularization terms. The majority of the methods using this approach can be subsumed in the following objective (see Tschannen et al. (2018) for a comprehensive review):

$$\text{ELBO}(\phi,\theta) + \beta\mathbb{E}_{\mathbf{x}}R_1\left(q_\phi(\mathbf{z} \mid \mathbf{x})\right) + \delta\mathbb{E}_{\mathbf{x},\mathbf{z}}R_2\left(q_\phi(\mathbf{z} \mid \mathbf{x}),\mathbf{z}\right)$$

We use $R_1 = D_{\text{KL}}\left(q_\phi(z \mid x) \| p(z)\right)$ a common choice for enabling unsupervised disentanglement that was originally proposed in beta-VAE (Higgins et al., 2017). Contrary to many other approaches for the second regularizer we use a

supervised term $R_2 = \mathcal{L}_{\boldsymbol{\xi}}(\mathbf{z}_{1:N_\xi}, \boldsymbol{\xi}) = \|\mathbf{z}_{1:N_\xi} - \boldsymbol{\xi}\|_2$ where ξ the real ODE parameters as described in Section 4.2. While many dynamical VAE methods use a different latent for each time step (Li & Mandt, 2018), our model can be seen as performing multi-step prediction from a single latent vector. Putting the above together we arrive at the formulation of Equation (6):

$$\mathcal{L}_{\phi,\theta}(\mathbf{x}_{\leq \mathbf{t}}) = \mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x}_{\leq t})}\left[\log P_\theta(\mathbf{x}_{t<,\leq t+n} \mid \mathbf{z}) + \delta \mathcal{L}_{\boldsymbol{\xi}}(\mathbf{z}_{1:N_{\boldsymbol{\xi}}}, \boldsymbol{\xi})\right] - \beta D_{KL}\left(Q_\phi(\mathbf{z} \mid \mathbf{x}_{\leq t}) || P(\mathbf{z})\right)$$

C.2. Scaling of the parameter

For our experiments (both in phase and observation space) we scale the ground-truth parameter in the $[0, 1]$ range:

$$\hat{\xi}_i = \frac{\xi - \min(\xi_i)}{\max(\xi_i) - \min(\xi_i)} \tag{7}$$

where ξ_i are the domain parameters and their corresponding minimum and maximum values of domain parameters from the training set. During training we use the output of $\hat{\xi}$ as the target for the regression loss.

C.3. SD-RSSM loss

The SD-RSSM is built upon the original RSSM with the addition of regression loss term which enhances the latent space disentanglement. Since the RSSM has latent variables for each time-step, we apply a disentanglement loss on all of them.

$$\mathcal{L}_{SD-RSSM}(\boldsymbol{o}_{\leq t}) = \sum_{t=1}^{T} \Bigg(\underbrace{\mathbb{E}_{q(\boldsymbol{s_t}|\boldsymbol{o_{\leq t}})}[\ln p(\boldsymbol{o_t} \mid \boldsymbol{s_t}))]}_{\text{reconstruction}} - \underbrace{\mathbb{E}_{q(\boldsymbol{s_{t-1}}|\boldsymbol{o_{\leq t-1}})}\left[\mathrm{KL}[q(\boldsymbol{s_t} \mid \boldsymbol{o_{\leq t}}) \| p(\boldsymbol{s_t} \mid \boldsymbol{s_{t-1}})]\right]}_{\text{prediction}}$$
$$+ \underbrace{\delta \mathbb{E}_{q(\boldsymbol{s_t}|\boldsymbol{o_{\leq t}})}\left[\left\|\boldsymbol{\xi} - \boldsymbol{s_t}^{(1:N_\xi)}\right\|_2\right]}_{\text{supervised disentanglement loss}} \Bigg)$$

Where $\boldsymbol{o_t}$ is the observations, $\boldsymbol{s_t}$ the stochastic latent variables at time t, $\boldsymbol{\xi}$ are the k dimensional domain parameters and δ tunes the supervised disentanglement strength.

D. Training and hyperparameters

D.1. Phase Space Experiments

Typically our training sequences are at least 1000 steps long. As a form of data augmentation, for each batch we select a random starting point t within the sequence.

An Adam optimizer with $b_1 = 0.9$ and $b_2 = 0.999$ and a scheduler for the learning rate are employed. The maximum number of epochs was set to 2000 but we also do early stopping using a validation set which led to significantly less epochs.

Table 4: **Pendulum hyperparameters**

	AE	SD-AE	VAE	SD-VAE	LSTM
Input Size			10, 50		
Output Size			1, 10		
Hidden Layers			[400, 200]		50,100,200
Latent Size			4, 8, 16		-
Nonlinearity			Leaky ReLU		Sigmoid
Num. Layers	-	-	-	-	1,2,3
Learning rate			10^{-3}		
Batch size	16, 32	16	16, 32	16	16, 64
Sched. patience	20, 30, 40	20,30	20	20	30
Sched. factor	0.3	0.3	0.3	0.3	0.3
Gradient clipping	No	1.0	1.0		
Layer norm (latent)	No	No	Yes	Yes	No
Teacher Forcing	-	-	-	-	Partial
Decoder γ	-	-	$10^{-3}, 10^{-4}, 10^{-5}$	$10^{-3}, 10^{-4}$	-
Sup. scaling	-	Linear	-	Linear	-
Supervision δ	-	0.1, 0.2, 0.3	-	0.01, 0.1, 0.2	-
# of experiments	72	72	72	72	72

Table 5: **Lotka-Volterra hyperparameters**

	AE	SD-AE	VAE	SD-VAE	LSTM
Input Size			50		
Output Size			10		
Hidden Layers			[400, 200]		50,100
Latent Size			8, 16, 32		-
Nonlinearity			Leaky ReLU		Sigmoid
Num. Layers	-	-	-	-	1,2,3
Learning rate	$10^{-3}, 10^{-4}$	$10^{-3}, 10^{-4}$	$10^{-3}, 10^{-4}$	10^{-3}	10^{-3}
Batch size	16, 32, 64	16, 32	16, 32	16	10, 64, 128
Sched. patience	20, 30	20, 30	20	20	20, 30
Sched. factor	0.3, 0.4	0.3	0.3	0.3	0.3
Gradient clipping	No	No	0.1, 1.0	0.1, 1.0	No
Layer norm (latent)	No	No	No	No	No
Teacher Forcing	-	-	-	-	Partial, No
Decoder γ	-	-	$10^{-4}, 10^{-5}, 10^{-6}$	$10^{-4}, 10^{-5}, 10^{-6}$	-
Sup. scaling	-	Linear	-	Linear	-
Supervision δ	-	0.1, 0.2, 0.3	-	0.01, 0.1, 0.2, 0.3	-
# of experiments	72	72	72	72	72

Table 6: **3-body system hyperparameters**

	AE	SD-AE	VAE	SD-VAE	LSTM
Input Size			50		
Output Size			10		
Hidden Layers			[400, 200]		50,100
Latent Size			8, 16, 32		-
Nonlinearity			Leaky ReLU		Sigmoid
Learning rate	$10^{-3}, 10^{-4}$	$10^{-3}, 10^{-4}$	$10^{-3}, 10^{-4}$	10^{-3}	
Batch size	16, 32	16	16	16	16, 64, 128
Sched. patience	30, 40, 50, 60	30, 40, 50, 60	30, 40, 50, 60	30, 40, 50, 60	20, 30
Sched. factor	0.3, 0.4	0.3	0.3, 0.4	0.3, 0.4	0.3
Gradient clipping	No	No	No	No	No
Layer norm (latent)	No	No	No	No	No
Decoder γ	-	-	$10^{-5}, 10^{-6}$	$10^{-5}, 10^{-6}$	-
Sup. scaling	-	Linear	-	Linear, Sigmoid	-
Supervision δ	-	0.05, 0.1, 0.2, 0.3	-	0.1, 0.2	-
# of experiments	96	96	96	96	96

Table 7: **Number of experiments with phase space data.** Each experiment corresponds to a distinct configuration of hyperparameters.

	AE	SD-AE	VAE	SD-VAE	LSTM	Total
Pendulum	72	72	72	72	72	360
L-V	72	72	72	72	72	360
3-body	96	96	96	96	96	480
					Total experiments	**1200**

D.2. Observation-space experiments

Table 8: **Hyperparameters for the RSSM and SD-RSSM models**

	RSSM	SD-RSSM
Batch Size	50, 100	
Decoder std.	1.0, 2.0	
Train Input Length	50, 100	
Supervision δ	-	0.01, 0.1, 1
Seeds	3	1
# of experiments	24	24

E. Additional Results for Phase Space Experiments

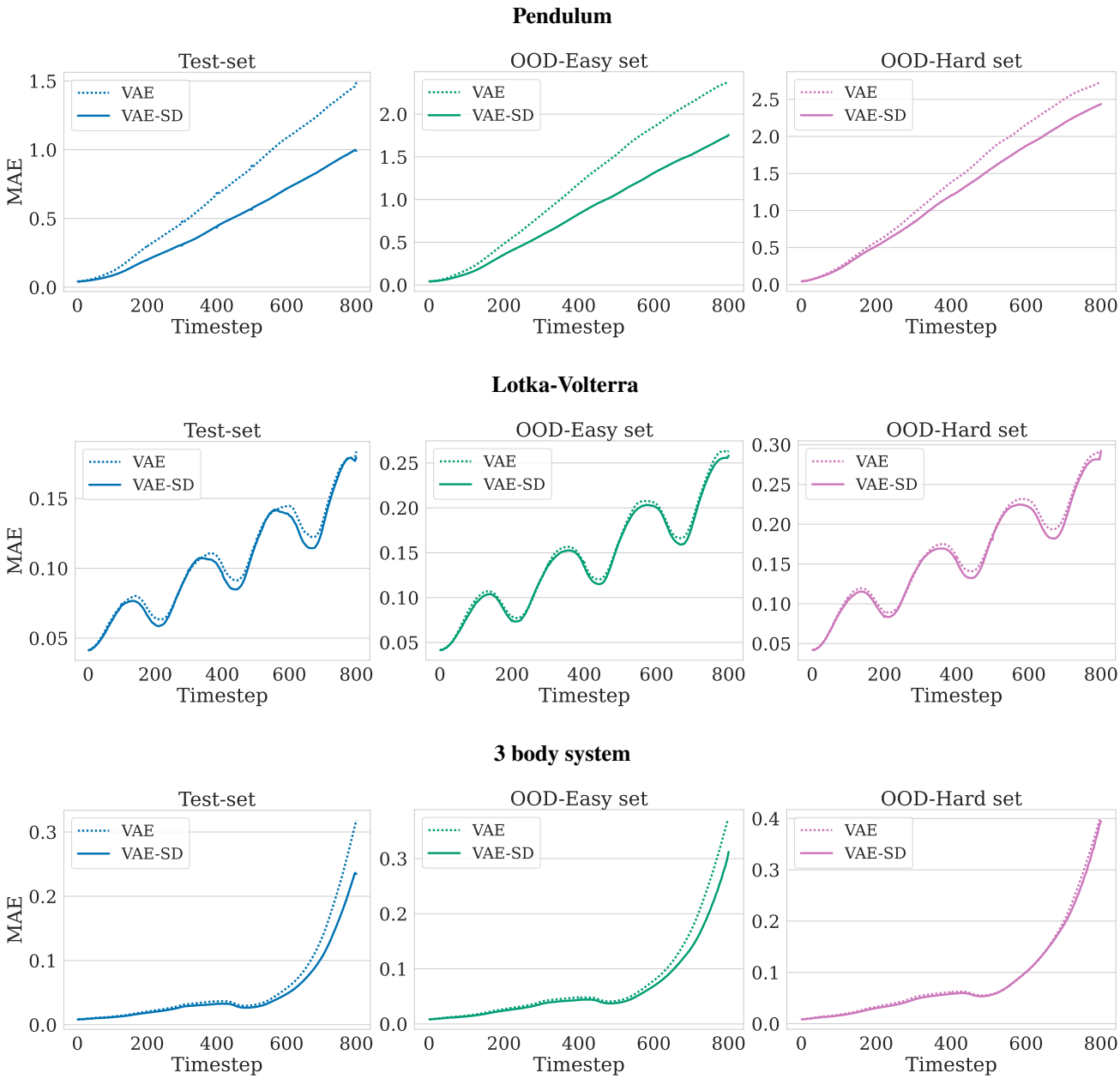
Table 9: **SD-VAE exhibits stronger disentanglement properties than the plain VAE according to many common metrics**. Figures are computed over the best 3 models.

	Pendulum		Lotka-Volterra		3 body system	
	VAE	SD-VAE	VAE	SD-VAE	VAE	SD-VAE
Disentanglement	-	-	0.27 ± 0.06	**0.53 ± 0.06**	0.20 ± 0.00	**0.90 ± 0.00**
Completeness	0.17 ± 0.06	**0.90 ± 0.00**	0.20 ± 0.00	**0.57 ± 0.06**	0.13 ± 0.06	**0.90 ± 0.00**
Informativeness	0.94 ± 0.01	**0.99 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**
SAP	0.03 ± 0.04	**0.87 ± 0.02**	0.04 ± 0.01	**0.21 ± 0.04**	0.01 ± 0.01	**0.67 ± 0.04**
MIG	0.01 ± 0.00	**0.17 ± 0.01**	0.00 ± 0.00	**0.03 ± 0.00**	0.00 ± 0.00	**0.08 ± 0.01**

Table 10: **Model stability** Percentage of models that diverge during testing in all some trajectories (out of all the trained models)

	Pendulum	Lotka-Volterra	3-body system
LSTM	86%	100%	53%
AE	42%	14%	50%
SD-AE	69%	29%	58%
VAE	3%	10%	15%
SD-VAE	2%	48%	9%

Figure 13: **Mean Absolute Error (MAE)** between model predictions and ground-truth trajectories.



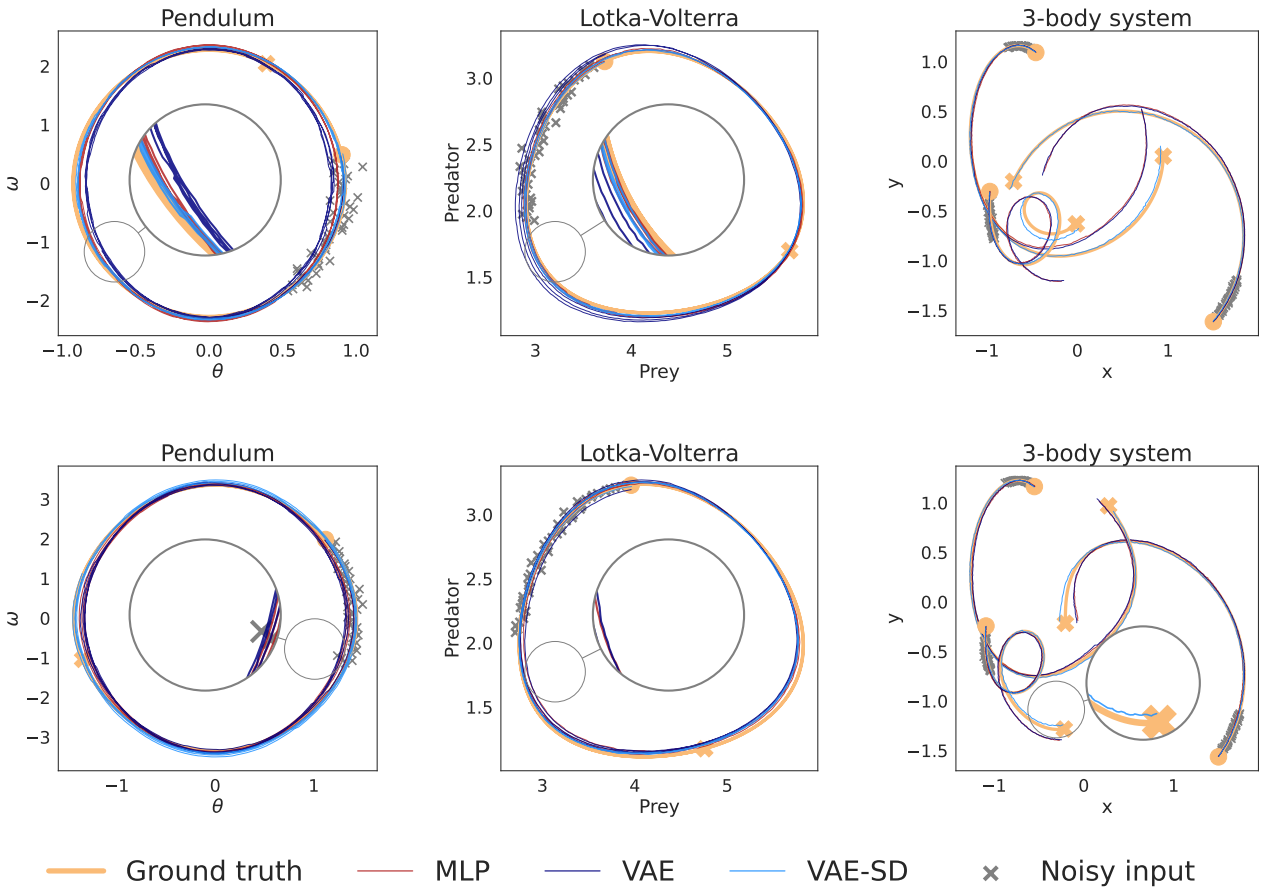


Figure 14: **Model predictions in phase space**. Trajectories are taken from the OOD-Hard set of each system. The trajectory observations are noisy, denoted by the grey '×' markers. The orange circle and the orange and bold '×' markers denote the start and end of the ground-truth trajectories respectively

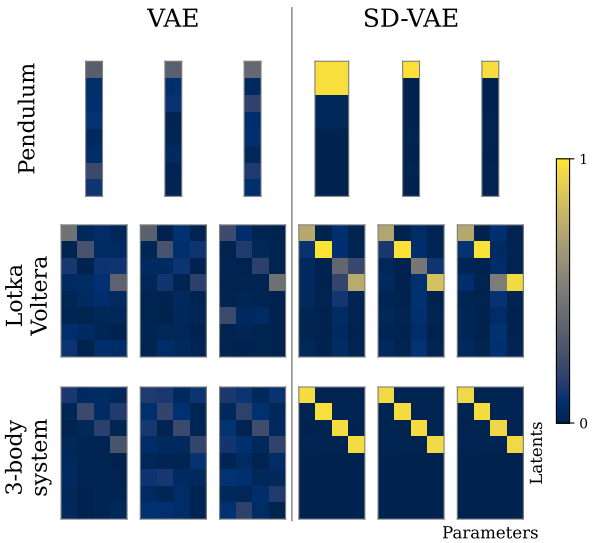
E.1. Linear correlation between z and ξ

To capture the relationship between latents z and parameters ξ we train boosted trees regressors with z as predictors and ξ as targets. The weights of the trained trees indicate whether there is a strong correlation between latents and real parameters. These results, depicted in Figure 5 and Figure 15 (a,b), show that supervised latents have very high predictive power for their respective real parameters. Nevertheless, trees can capture both linear and non-linear dependencies. It remains, unclear from this analysis if the relationship is linear or not. To futher clarify the relationship, we fit linear regression models between all z_i, ξ_j pairs. For each fit we compute the absolute Pearson correlation coefficient r that captures the linearity between the two variables. A value of r close to 1 denotes a highly linear correlation. Pearson r values are visualized in Figure 15 (c,d). We also provide numerical values of r for the supervised latents:

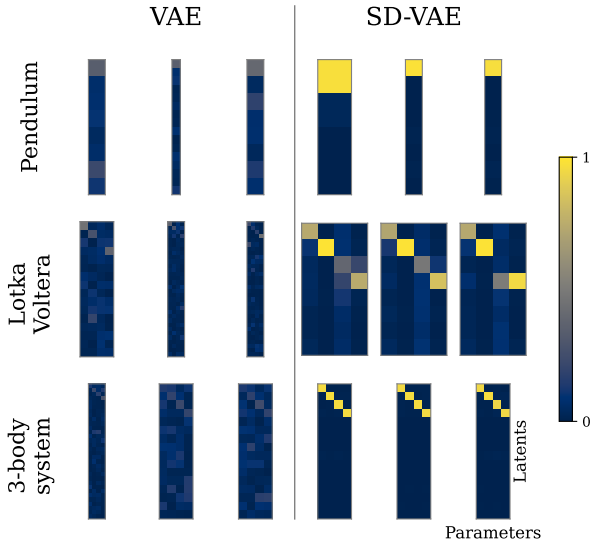
- **Pendulum**: $r_l = 0.94$
- **Lotka-Volterra** : $r_a = 0.52, r_b = 0.91, r_c = 0.23, r_d = 0.82$
- **3-Body system**: $r_K = 0.84, r_{m_1} = 0.87, r_{m_2} = 0.87, r_{m_3} = 0.85$

These results show that the relationship between supervised latents and is highly linear in most cases. This aligns with our experimental findings that linear scaling works best for the disentanglement loss (see Appendix C.2). We furthermore, exploit this linearity to perform traversals of the latent space in Appendix E.2.

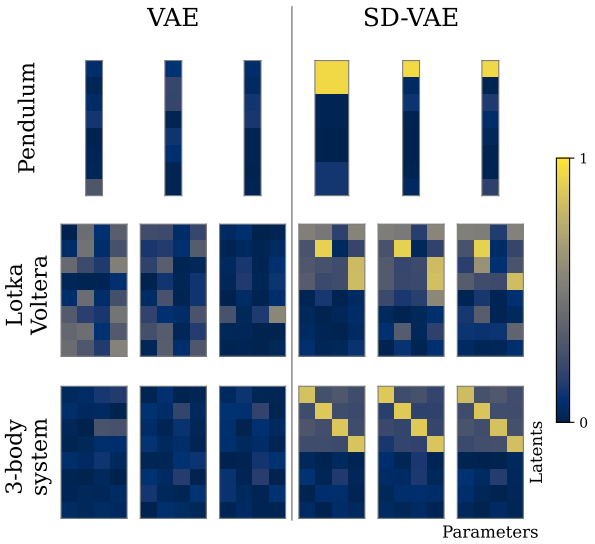
Figure 15: **Correlation between latent values and system parameters. Top** The importance weights of a random forest regressor trained to predict the ground-truth parameter values from the latents computed on the training set. High importance weights indicate a latent variable that has high predictive power over the ground-truth value. **Bottom** Pearson correlation between (absolute) between the system paramters and the latent variables.



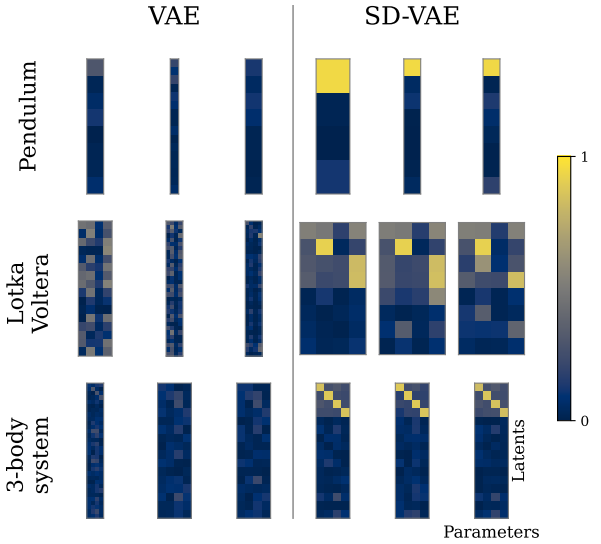
(a) Importance weights. Ordered and truncated latents.



(b) Importance weights. All latents.



(c) Pearson correlation. Ordered and truncated latents.



(d) Pearson correlation. All latents.

E.2. Latent space traversals

Being able to traverse between two point in the latent space and obtain a meaningful representation is a highly desirable property. Simple linear interpolation in latent space can produce meaningful images in properly disentangled VAEs (Higgins et al., 2017). While images have easily recognizable visual components, similar traversals for dynamical systems are harder to portray. Here we study whether interpolating between two points in the latent space of SD-VAE can produce meaningful trajectories. First, we create a new pendulum dataset containing 100 trajectories with linearly spaced pendulum length in the range $l \in [1.0 - 1.5]$. The initial conditions are kept constant ($\theta = \frac{pi}{2}, \omega = 0$) for all the trajectories to facilitate comparisons and we use the same noise level as in the training dataset. We use the encoder of our best SD-VAE model to extract the latent variables for each trajectory. For each trajectory the encoder produces 4 latent variables $z_1 \ldots z_4$. We interpolate between the latents of the the two extreme trajectories ($l = 1.0$ and $l = 1.5$). The interpolation we use is linear, driven by our

findings that latents and parameters have a strongly linear correlation (Section 5.5 and Appendix E.1). Next, we feed the real and interpolated latents to the decoder and predict up to 1000 timesteps. We find that the total mean absolute error between prediction and ground truth is 0.29 with the real latents and 0.33 with the interpolated one. These results indicate that linear latent space interpolation produces meaningful latent codes. This is further corroborated by plotting the real and interpolated latents together. As we can see in Figure 3 the relationship between the real latents z_i and pendulum length l is highly linear, which further explains with the linear interpolation method works well.

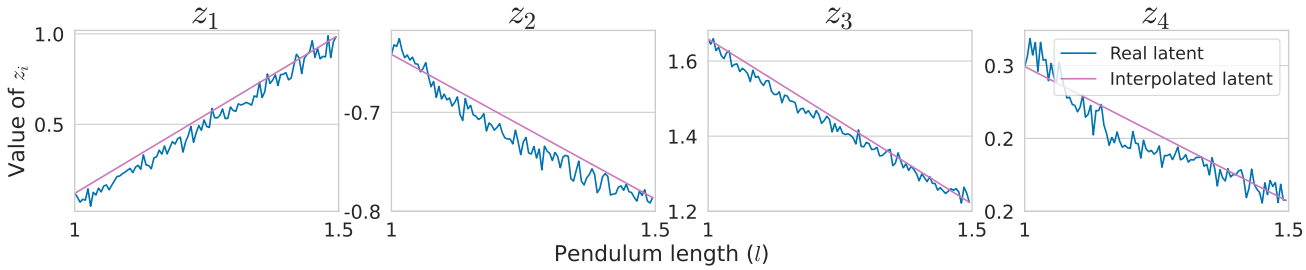


Figure 16: **Traversals of the latent space of SD-VAE** on the pendulum system. We pass trajectories corresponding to different pendulum length through the encoder to obtain the real values of z_i. The interpolation is done between the latent of minimum(l = 1.0) and maximum(l = 1.5) length. For this experiment the initial conditions were kept constant $\theta = \frac{\pi}{2}, \omega = 0$

F. Additional Results for Phase Space Experiments

Table 11: **Model comparison in observation-space pendulum.** Metrics are calculated between ground truth and prediction of the models at exactly 800 timesteps in the future.

	SSIM			PSNR		
	Test-set	OOD-Easy	OOD-Hard	Test-set	OOD-Easy	OOD-Hard
RSSM	0.795	0.787	0.783 2	13.36	12.71	12.26
SD-RSSM	**0.813**	**0.808**	**0.794**	**14.16**	**13.82**	**12.90**

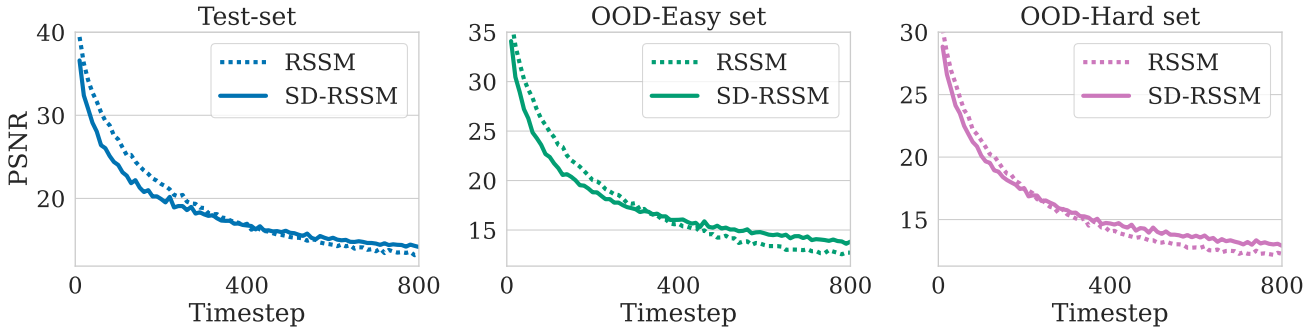


Figure 17: **Prediction quality on the observation-space pendulum.** PSNR as a function of the distance predicted into the future (x axis)

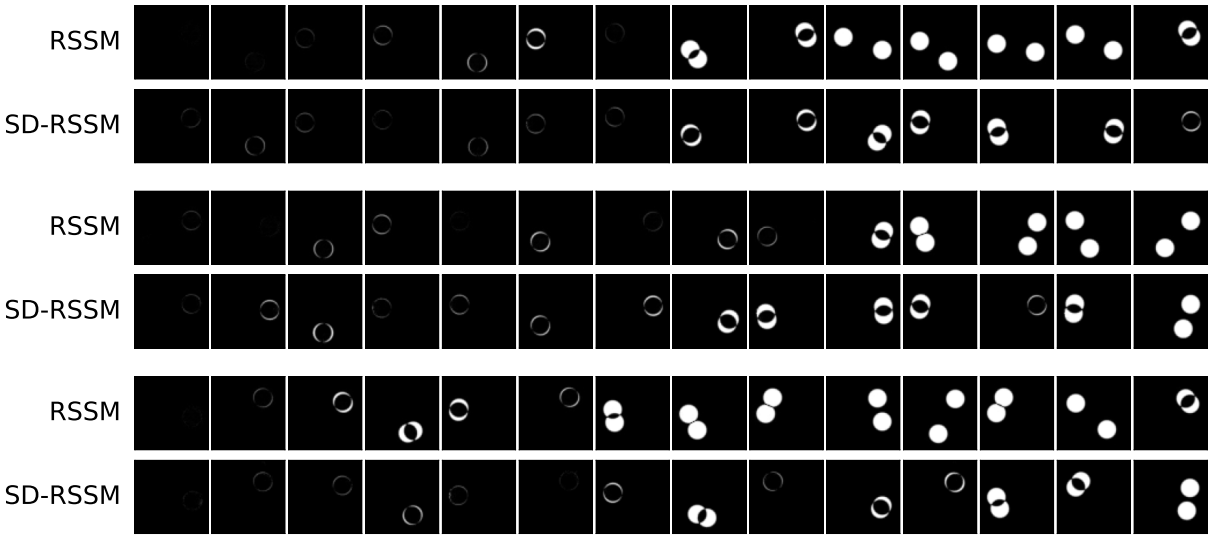


Figure 18: **Samples of predicted trajectories.** Absolute difference between ground truth and predictions on the test-set of the pendulum data set.