Identifying latent state transition in non-linear dynamical systems

Anonymous Authors[1]

Abstract

This work aims to improve generalization and interpretability of dynamical systems by recovering the underlying low-dimensional latent states and their time evolutions. Previous work on disentangled representation learning within the realm of dynamical systems focused on the latent states, possibly with linear transition approximations. As such, they cannot identify nonlinear transition dynamics, and hence fail to reliably predict complex future behavior. Inspired by advances in nonlinear ICA, we propose a state-space modeling framework in which we can identify not just the latent states but also the unknown transition function that maps past states to the present. We introduce a practical algorithm based on variational autoencoders and empirically demonstrate in realistic synthetic settings that we can recover latent state dynamics with high accuracy, and correspondingly achieve high future prediction accuracy.

1. Introduction

We focus on the problem of understanding the underlying states of a target dynamical system from its low-level, high-dimensional sensory measurements. This task is prevalent across various fields, including reinforcement learning (Hafner et al., 2019a) and robotics (Levine et al., 2016). As a running example of such a system, we consider a drone controlled by an autonomous system. Here, the observational data would be a video stream (Figure 1 **(b)**) instead of the full state of the system comprised of absolute position, velocity, and acceleration in $3D$ (Figure 1 **(a)**). This system may have additional variables influencing the state evolution, e.g., the strength and direction of the wind at any time. The main objective of this work is to learn latent representations and state transition functions that would be useful for downstream tasks, e.g., computing control signals for optimal transport of the drone from point A to point B.

Due to the partially observed nature of these problems, learning dynamics directly in the data space (e.g., pixel space) is not feasible, and previous works often focus on learning *latent dynamical systems* (Hafner et al., 2019b). However, such latent models commonly are not guaranteed to recover the true underlying states and transitions (*non-identifiability*), which results in entangled representations, lack of generalization across new domains, and poor interpretability (Schmidhuber, 1992; Bengio et al., 2013). Most existing identifiable methods (Hyvarinen & Morioka, 2016; 2017; Hyvarinen et al., 2019; Khemakhem et al., 2020; Klindt et al., 2020) assume mutually independent components that do not affect each other. This is unrealistic for dynamical systems as the present state of the system depends on the past states, i.e., the transition function propagates the system state by (nonlinearly) mixing the past state components (Morioka et al., 2021; Yao et al., 2021; 2022).

Recently, Yao et al. (2021; 2022) showed that under certain assumptions, it is possible to *identify* or recover the true unobserved latent states in a dynamical system (up to component-wise transformations). Morioka et al. (2021) introduce a framework to estimate the process noise, which represents the stochastic impulses fed to a dynamical process. Yao et al. (2021) utilizes non-stationarity noise to recover latent states from sequential data. Instead, Yao et al. (2022) exploits temporally autocorrelated latent states, while including factors modulating the dynamics and generative functions. These attempts lead to provably identifiable representations, but they only propose non-parametric or linear approximations to the unobserved state transition function. While their nonparametric approximations cannot be unrolled over time, a linear model falls short of predicting future states of complicated systems.

Our contributions. We present the first framework that allows for the *identification of the unknown transition function* alongside latent states and the generative function (see Figure 1). Following previous works (Klindt et al., 2020; Yao et al., 2021; 2022), we first establish the identifiability of latent states (Figure 1: Theorem 1). Different from them, our framework allows estimating the process noise, representing the random impulses fed to a dynamical system. Inspired by Morioka et al. (2021), we show that the estimation of the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SPIGM workshop at ICML 2024. Do not distribute.

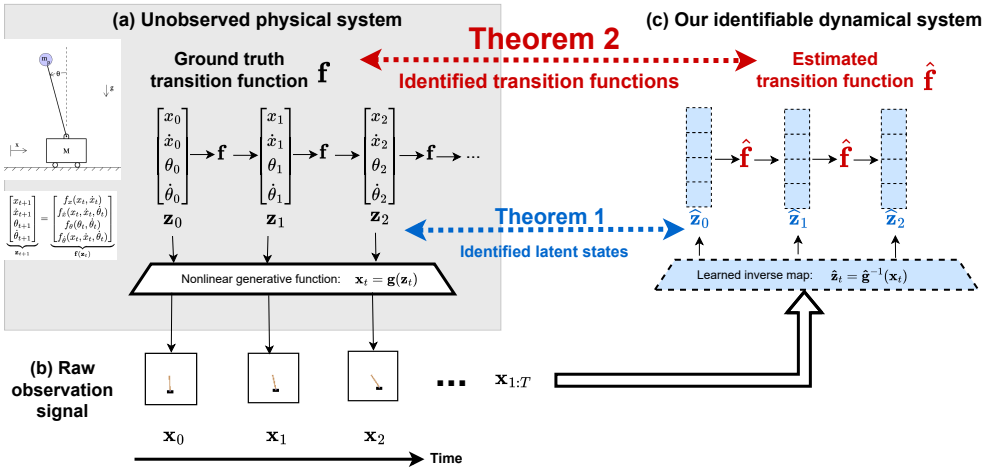

Figure 1. Sketch of our method and main theoretical contribution. **(a)** We assume an underlying *unobserved* dynamical system, e.g., a cartpole, where the full state is composed of the cart position and velocity, and the angle and angular velocity of the pole: $[x, \dot{x}, \theta, \dot{\theta}]$. **(b)** We partially observe the system as a sequence of video frames, which are used as input to our method. **(c)** We learn an inverse generative function that maps the raw observation signals to the latent state variables, as well as a transition function that maps the past latent states to the present latent state. Identifiability of the latent states is ensured by Theorem 1 (Yao et al., 2021). In addition to this, our main contribution is the identifiability of the transition function ensured by Theorem 2.

correct transition function is ensured by restricting the process noise and the transition function (Figure 1: Theorem 2). We propose an evidence lower bound that allows us to recover true underlying factors in the limit of infinite data. Our empirical findings show that our framework manages to predict the future states of an unknown system.

2. Identifiable dynamical system framework

We are interested in inferring latent dynamical systems from high-dimensional sensory observations $\mathbf{x}_{1:T}$, where t is the time index and $\mathbf{x}_t \in \mathbb{R}^D$. We assume a sequence of latent states $\mathbf{z}_{1:T}$, with $\mathbf{z}_t \in \mathbb{R}^K$, are instantaneously mapped to observations via a generative function $\mathbf{g} : \mathbb{R}^K \to \mathbb{R}^D$: $\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t)$. The latent states $\mathbf{z}_{1:T}$ evolve according to Markovian dynamics: $\mathbf{z}_t = \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t)$, where $\mathbf{f} : \mathbb{R}^{2K} \to \mathbb{R}^K$ is an auto-regressive transition function and $\mathbf{s}_t \in \mathbb{R}^K$ represents additional variables influencing the dynamics, e.g., random forces acting on the system or control signals.

Augmented dynamics. We introduce the following generative process with augmented transition and generative functions (Morioka et al., 2021):

$$\mathbf{z}_0 \sim p_{\mathbf{z}_0}(\mathbf{z}_0), \tag{1}$$

$$\mathbf{s}_t \sim p_{\mathbf{s}|\mathbf{u}}(\mathbf{s}_t|\mathbf{u}) = \prod_k p_{s_k|\mathbf{u}}(s_{kt}|\mathbf{u}), \tag{2}$$

$$\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \end{bmatrix} = \mathbf{f}_{\text{aug}}\left(\begin{bmatrix} \mathbf{s}_t \\ \mathbf{z}_{t-1} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t) \\ \mathbf{z}_{t-1} \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{bmatrix} = \mathbf{g}_{\text{aug}}\left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{g}(\mathbf{z}_t) \\ \mathbf{g}(\mathbf{z}_{t-1}) \end{bmatrix}, \tag{4}$$

$\forall t \in 1, \ldots, T$. Above, $\mathbf{u}$ is an auxiliary variable modulating the noise distribution $p_{\mathbf{s}|\mathbf{u}}$. Inspired by Hyvarinen & Morioka (2016; 2017), we consider the following cases: (i) setting $\mathbf{u}$ to an observed regime index leads to a **nonstationary process noise**, as a real-world example, consider a flying drone under different precipitation conditions, which can be observed up to a noise level; and (ii) setting $\mathbf{u} = \mathbf{s}_{t-1}$ implies an **autocorrelated noise process**, as a real-world example, consider a flying drone in a windy environment where the wind speed or direction changes continuously.

2.1. Identifiability theory

Let $\mathcal{M} = (\mathbf{f}_{\text{aug}}, \mathbf{g}_{\text{aug}}, p_{\mathbf{s}|\mathbf{u}})$ denote the ground-truth model. We learn a model $\hat{\mathcal{M}} = (\hat{\mathbf{f}}_{\text{aug}}, \hat{\mathbf{g}}_{\text{aug}}, \hat{p}_{\mathbf{s}|\mathbf{u}})$ by fitting the observed sequences. We make the following assumptions:

(A0) **Distribution matching (Klindt et al., 2020; Yao et al., 2021; 2022)** The learned and the ground-truth observation densities match everywhere.

(A1) **Injectivity and bijectivity (Morioka et al., 2021)** The generator functions $\mathbf{g}$ and $\hat{\mathbf{g}}$ are injective, which implies that the augmented generative functions $\mathbf{g}_{\text{aug}}, \hat{\mathbf{g}}_{\text{aug}}$ are injective. The augmented dynamics functions $\mathbf{f}_{\text{aug}}, \hat{\mathbf{f}}_{\text{aug}}$ are bijective.

Contrary to Morioka et al. (2021), which use an augmented transition model on *observations* $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, our formulation captures the functional dependence between a *latent pair* $(\mathbf{z}_{t-1}, \mathbf{z}_t)$ and the process noise $\mathbf{s}_t$.

(A2) **Decomposed transitions (Klindt et al., 2020; Yao et al., 2021; 2022; Song et al., 2023)** Each dimension of the transition function $\{f_k\}_{k=1}^K$ is influenced by

a single process noise variable s_{kt}. The output is a single latent variable z_{kt}: $z_{kt} = f_k(\mathbf{z}_{t-1}, s_{kt})$, for $k \in 1, \dots, K$ and $t \in 1, \dots, T$.

(A3) **Conditionally independent noise.** Let $q_k(s_{kt}, \mathbf{u}) = \log p(s_{kt}|\mathbf{u})$ denote the conditional density of the noise variable s_{kt}. Let $\eta_k(z_{kt}, \mathbf{u}) = \log p(z_{kt}|\mathbf{z}_{t-1}, \mathbf{u})$ denote the conditional density of the state variable z_{kt}. Conditioned on the auxiliary variable $\mathbf{u}$, we assume:

- Each noise variable $\mathbf{s}_t \in \mathbb{R}^K$ is independent over its dimensions $s_{1t}, \dots, s_{Kt}, \forall t \in 1, \dots, T$: $\log p(\mathbf{s}_t|\mathbf{u}) = \sum_{k=1}^K \log p(s_{kt}|\mathbf{u}) = \sum_{k=1}^K q_k(s_{kt}, \mathbf{u})$.
- the past latent state $\mathbf{z}_{t-1}$ and the present noise $\mathbf{s}_t$ are independent: $\mathbf{s}_t \perp\!\!\!\perp \mathbf{z}_{t-1}|\mathbf{u}$.

Since $\mathbf{s}_t \perp\!\!\!\perp \mathbf{z}_{t-1}|\mathbf{u}$ and each dimension of the transition function $f_k(\mathbf{z}_{t-1}, s_{kt})$ is a function of only a single (conditionally) independent noise variable s_{kt}, the conditional density of the latent pair $(\mathbf{z}_t, \mathbf{z}_{t-1})$ also factorizes: $\log p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}) = \sum_{k=1}^K \log p(z_{kt}|\mathbf{z}_{t-1}, \mathbf{u}) = \sum_{k=1}^K \eta_k(z_{kt}, \mathbf{u})$. The same is assumed for the learned conditional density: $\log p(\hat{\mathbf{z}}_t|\hat{\mathbf{z}}_{t-1}, \mathbf{u}) = \sum_{k=1}^K \log p(\hat{z}_{kt}|\hat{\mathbf{z}}_{t-1}, \mathbf{u})$.

(A4) **Sufficient variability of latent state $\mathbf{z}_t$ (Yao et al., 2021).** For any $\mathbf{z}_t$, there exist some $2K$ values of $\mathbf{u}$: $\mathbf{u}_1, \dots, \mathbf{u}_{2K}$, such that the $2K$ vectors $\mathbf{v}(\mathbf{z}_t, \mathbf{u}_1), \dots, \mathbf{v}(\mathbf{z}_t, \mathbf{u}_{2K})$ are linearly independent for some index l of the auxiliary variable $\mathbf{u}$, where

$$\mathbf{v}(\mathbf{z}_t, \mathbf{u}) = \left(\frac{\partial^2 \eta_1(z_{1t}, \mathbf{u})}{\partial z_{1t} \partial u_l}, \cdots, \frac{\partial^2 \eta_K(z_{Kt}, \mathbf{u})}{\partial z_{Kt} \partial u_l}, \frac{\partial^3 \eta_1(z_{1t}, \mathbf{u})}{\partial z_{1t}^2 \partial u_l}, \cdots, \frac{\partial^3 \eta_K(z_{Kt}, \mathbf{u})}{\partial z_{Kt}^2 \partial u_l} \right) \in \mathbb{R}^{2K}.$$

(A5) **Sufficient variability of process noise $\mathbf{s}_t$.** For any $\mathbf{s}_t$, there exist some $2K$ values of $\mathbf{u}$: $\mathbf{u}_1, \dots, \mathbf{u}_{2K}$, such that the $2K$ vectors $\mathbf{w}(\mathbf{s}_t, \mathbf{u}_1), \dots, \mathbf{w}(\mathbf{s}_t, \mathbf{u}_{2K})$ are linearly independent for some index l of the auxiliary variable $\mathbf{u}$, where

$$\mathbf{w}(\mathbf{s}_t, \mathbf{u}) = (\frac{\partial^2 q_1(s_{1t}, \mathbf{u})}{\partial s_{1t} \partial u_l}, \cdots, \frac{\partial^2 q_K(s_{Kt}, \mathbf{u})}{\partial s_{Kt} \partial u_l}, \frac{\partial^3 q_1(s_{1t}, \mathbf{u})}{\partial s_{1t}^2 \partial u_l}, \cdots, \frac{\partial^3 q_K(s_{Kt}, \mathbf{u})}{\partial s_{Kt}^2 \partial u_l} \in \mathbb{R}^{2K}.$$

2.2. Main theoretical contribution

In this section, we state our main theoretical contribution, that is, the identifiability result for the dynamical function $\mathbf{f}$ (**Theorem 2**). For completeness, we start with a theorem on the identifiability result for the conditionally independent latent states $\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}$ (**Theorem 1**), which Yao et al. (2021) established for the nonstationary noise case. The proofs are detailed in Appendices A.3 and A.4.

Theorem 1. *Under assumptions (A0, A1, A2, A3, A4), latent states $\mathbf{z}_t = \mathbf{h}(\hat{\mathbf{z}}_t)$ are identifiable up to a function composition $\mathbf{h} = \pi_z \circ r_z$ of a permutation $\pi_z : [K] \to [K]$ and element-wise invertible transformation $r_z : \mathbb{R}^K \to \mathbb{R}^K$ (Yao et al., 2021). Or equivalently, the same follows for the generative function $\mathbf{g} \circ \mathbf{h} = \hat{\mathbf{g}}$.*

Theorem 2. *Under assumptions (A0, A1, A2, A3, A4, A5), the process noise $\mathbf{s}_t = \mathbf{k}(\hat{\mathbf{s}}_t)$ is identifiable up to a function composition $\mathbf{k} = \pi_s \circ r_s$ of a permutation $\pi_s : [K] \to [K]$ and an element-wise invertible transformation $r_s : \mathbb{R}^K \to \mathbb{R}^K$. Equivalently, the dynamical function $\mathbf{h}_{aug}^{-1} \circ \mathbf{f}_{aug} \circ \mathbf{k}_{aug} = \hat{\mathbf{f}}_{aug}$ is identifiable up to a function composition $\mathbf{k}_{aug} = [\mathbf{k}, \mathbf{h}] = \pi \circ r$ of a permutation $\pi = [\pi_s, \pi_z] : [2K] \to [2K]$ and an element-wise invertible transformation $r = [r_s, r_z] : \mathbb{R}^{2K} \to \mathbb{R}^{2K}$, where **Theorem 1** already proves that $\mathbf{h}_{aug}^{-1} = [\mathbf{h}^{-1}, \mathbf{h}^{-1}]$ is an invertible element-wise transformation.*

3. Practical implementation using variational inference

We turn our theoretical framework into a practically usable implementation using variational inference. For space considerations, we defer the details to Appendix B. We present the below algorithm for implementation details and Figure 4 for architecture details.

Algorithm 1 Practical learning algorithm

Requires: Variational posterior networks (`ICEncoder` and `NoiseEncoder`) and `Decoder`.

1. Encode initial condition parameters: $\mu_{\mathbf{z}_0}, \log \sigma^2_{\mathbf{z}_0} = \texttt{ICEncoder}(\mathbf{x}_{1:T_{\text{ic}}})$
2. Sample initial condition: $\mathbf{z}_0 \sim \mathcal{N}(\mu_{\mathbf{z}_0}, \sigma^2_{\mathbf{z}_0}\mathbf{I})$
3. For $t \in 1, \dots, T$:
   (a) Encode noise parameters: $\mu_{\mathbf{s}_t}, \log \sigma^2_{\mathbf{s}_t} = \texttt{NoiseEncoder}(\mathbf{x}_{1:t}, \mathbf{z}_{t-1})$
   (b) Sample noise: $\mathbf{s}_t \sim \mathcal{N}(\mu_{\mathbf{s}_t}, \sigma^2_{\mathbf{s}_t}\mathbf{I})$
   (c) Compute the next latent state: $\mathbf{z}_t = \mathbf{f}(\mathbf{s}_t, \mathbf{z}_{t-1})$
   (d) Decode: $\mathbf{x}_t = \texttt{Decoder}(\mathbf{z}_t)$
4. Compute ELBO: $\mathcal{L} = \mathcal{L}_{\text{R}} - \beta\mathcal{L}_{\text{KL}}$. Samples $\{\mathbf{s}_{1:T}, \mathbf{z}_{0:T}\}$ are used to approximate $\mathcal{L}_{\text{KL}}$.
5. Update the parameters $\{\theta, \phi\}$.

4. Synthetic experiment

As common in nonlinear ICA literature (Hyvarinen et al., 2019; Morioka et al., 2021; Yao et al., 2021; 2022), we set

Table 1. Synthetic experiment results (mean, std.dev. across 5 runs). For methods that cannot predict the future, we leave the $\text{MSE}[\bar{\mathbf{x}}_{\text{future}[\cdot]}]$ rows empty (N/A). Likewise, we compute $\text{MCC}[\bar{\mathbf{s}}_{\text{train}}]$ only for LEAP and our method as others do not maintain process noise variables.

	MODELS								
METRICS	β-VAE	PCL	TCL	iVAE	SLOWVAE	KALMANVAE	LEAP-LIN	LEAP-NP	OURS
$\text{MCC}[\bar{\mathbf{z}}_{\text{train}}] \uparrow$	0.60 (±0.05)	0.57 (±0.05)	0.39 (±0.07)	0.58 (±0.06)	0.41 (±0.05)	0.64 (±0.05)	0.68 (±0.03)	0.89 (±0.04)	0.95 (±0.08)
$\text{MCC}[\bar{\mathbf{s}}_{\text{train}}] \uparrow$	N/A	N/A	N/A	N/A	N/A	N/A	0.14 (±0.01)	0.26 (±0.04)	0.66 (±0.09)
$\text{MSE}[\bar{\mathbf{x}}_{\text{future}[2]}] \downarrow$	N/A	N/A	N/A	N/A	N/A	1.27 (±0.19)	0.22 (±0.03)	N/A	0.06 (±0.01)
$\text{MSE}[\bar{\mathbf{x}}_{\text{future}[4]}] \downarrow$	N/A	N/A	N/A	N/A	N/A	1.32 (±0.27)	0.18 (±0.03)	N/A	0.09 (±0.01)
$\text{MSE}[\bar{\mathbf{x}}_{\text{future}[8]}] \downarrow$	N/A	N/A	N/A	N/A	N/A	1.72 (±0.82)	0.59 (±0.13)	N/A	0.21 (±0.03)

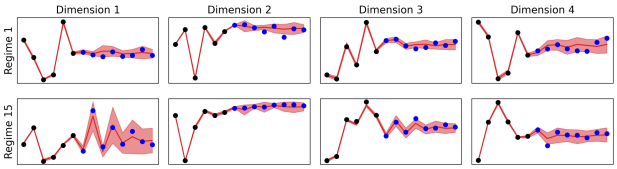

Figure 2. Model predictions (red) in the data space (black and blue dots represent train and future points respectively) with the estimated uncertainties (±2 std.dev). We observe near-perfect predictions and low uncertainty for the input data (the first $T_{\text{train}} = T_0 + T_{\text{dyn}} = 6$ time points) while the uncertainty grows as we unroll over time (the next $T_{\text{future}} = 8$ time points). Further, the uncertainty grows even more when the model predictions are off. Therefore, almost all test points lie in the ±2 std region, reflecting the high calibration level our probabilistic model attains.

up a synthetic experiment to show that our model recovers the latent dynamics, and hence achieves a higher future prediction accuracy.

Dataset. As in Hyvarinen et al. (2019); Yao et al. (2021; 2022), we set up a synthetic data experiment containing multivariate time-series. Same as Yao et al. (2021), we use 2-linear layer random MLPs as generative and transition functions (see Appendix C for details). Each sequence has length $T = T_0 + T_{\text{dyn}} + T_{\text{future}} = 2+4+8 = 14$. We assume a second-order Markov transition (lag=2), and first $T_0 = 2$ states are spared as initial states. The next 4 observations $\mathbf{x}_{1:4}$ are used for training the dynamical model. The last 8 observations $\mathbf{x}_{5:12}$ are used for assessing the performance of future estimation. We choose the future prediction horizon $T_{\text{future}} = 8$ as the double of the training sequence length. If the dynamics are truly identified, the model should predict future states well, even for a longer horizon.

Metrics. We denote the latent states and the process noise for first T_{train} steps by $\bar{\mathbf{z}}_{\text{train}} = \mathbf{z}_{0:T_{\text{train}}}$ and $\bar{\mathbf{s}}_{\text{train}} = \mathbf{s}_{1:T_{\text{train}}}$ respectively. For latent system identification, we measure the validation MCC for the latent sequence $\bar{\mathbf{z}}_{\text{train}}$: $\text{MCC}[\bar{\mathbf{z}}_{\text{train}}]$, and the noise sequence $\bar{\mathbf{s}}_{\text{train}}$: $\text{MCC}[\bar{\mathbf{s}}_{\text{train}}]$. For future prediction performance, we measure the mean squared error (MSE) on the future observations $\text{MSE}[\bar{\mathbf{x}}_{\text{future}}]$. We denote the future observations for the subsequent T_{future} steps by $\bar{\mathbf{x}}_{\text{future}} = \mathbf{x}_{T_{\text{train}}+1:}$. When T_{future} takes different values, e.g., $T_{\text{future}} \in \{2, 4, 8\}$, we denote the future metrics by $\text{MSE}[\bar{\mathbf{x}}_{\text{future}[2]}]$, $\text{MSE}[\bar{\mathbf{x}}_{\text{future}[4]}]$ and $\text{MSE}[\bar{\mathbf{x}}_{\text{future}[8]}]$.

Baseline methods. We compare our method with several nonlinear ICA methods: β-VAE (Higgins et al., 2018), TCL (Hyvarinen & Morioka, 2016), iVAE (Khemakhem et al., 2020), PCL (Hyvarinen & Morioka, 2017), SLOWVAE (Klindt et al., 2020), LEAP (Yao et al., 2021) with its two versions having linear and nonparametric transition functions LEAP-LIN and LEAP-NP; and a disentangled deep state-space model KALMANVAE (Fraccaro et al., 2017).

Main results. We show $\text{MCC}[\bar{\mathbf{z}}_{\text{train}}]$, $\text{MCC}[\bar{\mathbf{s}}_{\text{train}}]$ and $\text{MSE}[\bar{\mathbf{x}}_{\text{future}}]$ results in Table 1. Our model recovers latent states $\mathbf{z}$ and process noise $\mathbf{s}$ better than baselines, as demonstrated by a higher correlation with the true latent states and process noise. This leads to a higher accuracy in future prediction, in terms of $\text{MSE}[\bar{\mathbf{x}}_{\text{future}}]$ with prediction horizons $\{2, 4, 8\}$. We remind that 8-step future prediction corresponds to the double the amount of dynamics steps the models see during training. As the prediction horizon increases, we see that the difference in the future prediction performance between our method and baselines also increases.

5. Discussion

We present the first latent dynamical system that allows for identification of the unknown transition function, and theoretically proved its identifiability based on standard assumptions. We evaluated our approach on synthetic data and showed that (i) the estimated latent states correlated strongly with the ground truth, (ii) our method had the highest future prediction accuracy with calibrated uncertainties. The main limitation stems from identifiability assumptions adopted.

References

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Fraccaro, M., Kamronn, S., Paquet, U., and Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems*, 30, 2017.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

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

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hyvarinen, A. and Morioka, H. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

Hyvarinen, A. and Morioka, H. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pp. 460–469. PMLR, 2017.

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Hyvarinen, A., Sasaki, H., and Turner, R. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019.

Hyvärinen, A., Khemakhem, I., and Morioka, H. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns*, 4(10), 2023.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Klindt, D., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Morioka, H., Hälvä, H., and Hyvarinen, A. Independent innovation analysis for nonlinear vector autoregressive process. In *International Conference on Artificial Intelligence and Statistics*, pp. 1549–1557. PMLR, 2021.

Schmidhuber, J. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

Song, X., Yao, W., Fan, Y., Dong, X., Chen, G., Niebles, J. C., Xing, E., and Zhang, K. Temporally disentangled representation learning under unknown nonstationarity. *arXiv preprint arXiv:2310.18615*, 2023.

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. normflows: A pytorch package for normalizing flows. *arXiv preprint arXiv:2302.12014*, 2023.

Xi, Q. and Bloem-Reddy, B. Indeterminacy in generative models: Characterization and strong identifiability. In *International Conference on Artificial Intelligence and Statistics*, pp. 6912–6939. PMLR, 2023.

Yao, W., Sun, Y., Ho, A., Sun, C., and Zhang, K. Learning temporally causal latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*, 2021.

Yao, W., Chen, G., and Zhang, K. Temporally disentangled representation learning. *Advances in Neural Information Processing Systems*, 35:26492–26503, 2022.

A. Identifiability Theory

In this section, we discuss the identifiability of the latent states and the transition function, and provide the detailed proofs.

We assume a latent dynamical system which is viewed as high-dimensional sensory observations $\mathbf{x}_{1:T}$, where t is the time index and $\mathbf{x}_t \in \mathbb{R}^D$. We assume a sequence of latent states $\mathbf{z}_{1:T}$, with $\mathbf{z}_t \in \mathbb{R}^K$, are instantaneously mapped to observations via a generative function $\mathbf{g} : \mathbb{R}^K \to \mathbb{R}^D$:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t). \tag{5}$$

The latent states $\mathbf{z}_{1:T}$ evolve according to Markovian dynamics:

$$\mathbf{z}_t = \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t), \tag{6}$$

where $\mathbf{f} : \mathbb{R}^{2K} \to \mathbb{R}^K$ is an auto-regressive transition function and $\mathbf{s}_t \in \mathbb{R}^K$ corresponds to process noise.

Our aim is to jointly identify the latent states $\mathbf{z}_{1:T}$, the dynamics function $\mathbf{f}$, and the process noise $\mathbf{s}_{1:T}$. We remind that previous works (Klindt et al., 2020; Yao et al., 2021; 2022; Song et al., 2023) have concentrated on identifying the latent states $\mathbf{z}_{1:T}$, possibly with linear transition approximations, but not a general transition function $\mathbf{f}$. Yet, without a general $\mathbf{f}$, the methods can estimate the underlying states only when corresponding observations are provided or provide simplistic approximations in their absence. Hence, they cannot predict complex future behavior reliably.

Notice that learning a provably identifiable transition function $\mathbf{f} : \mathbb{R}^{2K} \to \mathbb{R}^K$ is not straightforward, since the transition function is not injective. A naive solution can be to simply use a plug-in method (Yao et al., 2021; 2022) for identifying the latents and then fitting a transition function $\mathbf{f}$ on the estimated latents, however, we show empirically in our experiments that it leads to poor prediction accuracy for the future behavior.

A.1. Nonlinear ICA

The nonlinear ICA assumes that the data is generated from independent latent variables $\mathbf{z}$ with a nonlinear generative function $\mathbf{g}$, following Equation (5). It is well-known to be non-identifiable for i.i.d. data (Hyvärinen & Pajunen, 1999; Locatello et al., 2019). Recent seminal works (Hyvarinen & Morioka, 2016; 2017; Hyvarinen et al., 2019) showed that *autocorrelation* and *nonstationarity* existent in non-i.i.d. data can be exploited to identify latent variables in an unsupervised way. Compared to the vanilla ICA that considers independence only along latent dimensions, the idea of these works is to introduce additional independence constraints reflecting the existent structure in the data. These additional constraints are formulated mathematically as *identifiability assumptions*, which restrict the space of the generative function $\mathbf{g}$ and the space of the latent prior $p_{\mathbf{z}}$ (Hyvärinen et al., 2023; Xi & Bloem-Reddy, 2023). The key insight is that, after sufficiently constraining the latent prior $p_{\mathbf{z}}$ using such assumptions, identifying the latent variables $\mathbf{z}_t$ and identifying the injective generative function $\mathbf{g}$ become equivalent tasks (Xi & Bloem-Reddy, 2023).

A.2. Augmented dynamics for identifiable systems

To identify the transition function $\mathbf{f}$ such that $\mathbf{z}_t = \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t)$, we will use the same insight: After sufficiently constraining the noise prior $p_{\mathbf{s}}$; given an identifiable latent pair $(\mathbf{z}_{t-1}, \mathbf{z}_t)$, identifying the noise variables $\mathbf{s}_t$ and identifying the bijective dynamics function $\mathbf{f}$ should be equivalent. Hence, in addition to the identifiability assumptions restricting the space of the generative function $\mathbf{g}$ and the space of the latent prior $p_{\mathbf{z}}$, we will further restrict the space of the dynamics function $\mathbf{f}$, and the space of the noise prior $p_{\mathbf{s}}$.

First, let us note that the identifiability of the process noise $\mathbf{s}_t$ is not trivial since the dynamics function $\mathbf{f} : \mathbb{R}^{2K} \to \mathbb{R}^K$ is not an injective function and hence it does not have an inverse. Following the independent innovation analysis (IIA) framework Morioka et al. (2021), we trivially augment the image space of the transition function and denote the bijective augmented function by $\mathbf{f}_{\texttt{aug}} : \mathbb{R}^{2K} \to \mathbb{R}^{2K}$:

$$\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \end{bmatrix} = \mathbf{f}_{\texttt{aug}} \left(\begin{bmatrix} \mathbf{s}_t \\ \mathbf{z}_{t-1} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t) \\ \mathbf{z}_{t-1} \end{bmatrix} \tag{7}$$

Contrary to Morioka et al. (2021), which use an augmented autoregressive model on *observations* $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, our formulation captures the functional dependence between a *latent pair* $(\mathbf{z}_{t-1}, \mathbf{z}_t)$ and the process noise $\mathbf{s}_t$.

Next, we make the standard assumption in the temporal identifiability literature (Klindt et al., 2020; Yao et al., 2021; 2022; Song et al., 2023) that each dimension of the transition function $\{f_k\}_{k=1}^K$ is influenced by a single process noise variable

s_{kt}. The output is a single latent variable z_{kt}:

$$z_{kt} = f_k(\mathbf{z}_{t-1}, s_{kt}), \quad \text{for } k \in 1, \dots, K \text{ and } t \in 1, \dots, T. \tag{8}$$

Notice that this does not impose a limitation on the generative model, it just creates a segmentation between noise variables and latent variables. For example, if this assumption is violated and there exists a noise variable s_{kt} that affects both z_{it} and z_{jt} with $i \neq j$, then the noise variable s_{kt} can instead be modeled as a latent variable z_{kt}.

We re-state the full generative model for completeness:

$$\mathbf{z}_0 \sim p_{\mathbf{z}_0}(\mathbf{z}_0), \qquad \texttt{\# initial state} \tag{9}$$

$$\mathbf{s}_t \sim p_{\mathbf{s}|\mathbf{u}}(\mathbf{s}_t|\mathbf{u}) = \prod_k p_{s_k|\mathbf{u}}(s_{kt}|\mathbf{u}), \quad \forall t \in 1, \dots, T, \qquad \texttt{\# process noise} \tag{10}$$

$$\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \end{bmatrix} = \mathbf{f}_{\text{aug}}\left(\begin{bmatrix} \mathbf{s}_t \\ \mathbf{z}_{t-1} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{f}(\mathbf{z}_{t-1}, \mathbf{s}_t) \\ \mathbf{z}_{t-1} \end{bmatrix}, \quad \forall t \in 1, \dots, T, \qquad \texttt{\# state transition} \tag{11}$$

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{bmatrix} = \mathbf{g}_{\text{aug}}\left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t-1} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{g}(\mathbf{z}_t) \\ \mathbf{g}(\mathbf{z}_{t-1}) \end{bmatrix}, \quad \forall t \in 2, \dots, T. \qquad \texttt{\# observation mapping} \tag{12}$$

where $\mathbf{u}$ is an auxiliary variable, which modulates the noise distribution $p_{\mathbf{s}|\mathbf{u}}$.

A.3. Proof of Theorem 1: Identifiability of the latent states $\mathbf{z}_t$

This result is already shown in (Yao et al., 2021, Appendix A.3.2). Here, we follow Klindt et al. (2020); Yao et al. (2021; 2022) and repeat their results in our notation as we also make use of this result in Appendix A.4.

The injective functions $\mathbf{g}, \hat{\mathbf{g}} : \mathbb{R}^K \to \mathbb{R}^D$ are bijective between the latent space $\mathbb{R}^K$ and the observation space $\mathcal{X} \subset \mathbb{R}^D$. We denote the inverse functions from the restricted observation space to the latent space by $\mathbf{g}^{-1}, \hat{\mathbf{g}}^{-1}$. This is also implicitly assumed in (Klindt et al., 2020; Yao et al., 2021; 2022; Song et al., 2023).

We have

$$\mathbf{x}_t = \hat{\mathbf{g}}(\hat{\mathbf{z}}_t) = \Big((\mathbf{g} \circ \underbrace{\mathbf{g}^{-1}}) \circ \hat{\mathbf{g}} \Big)(\hat{\mathbf{z}}_t) \implies \hat{\mathbf{g}} = \mathbf{g} \circ \mathbf{h} \implies \mathbf{z}_t = \mathbf{h}(\hat{\mathbf{z}}_t). \tag{13}$$

The function $\mathbf{h} : \hat{\mathbf{z}}_t \mapsto \mathbf{z}_t$ maps the learned latents to the ground-truth latents. To show it is bijective, we need to show it is both injective and surjective. Following Klindt et al. (2020), it is injective since it is a composition of injective functions. Assume it is not surjective, then there exists a neighborhood $\mathbf{U}_\mathbf{z}$ for which $\mathbf{g}(\mathbf{U}_\mathbf{z}) \notin \hat{\mathbf{g}}(\mathbb{R}^K)$. This implies that the neighborhood of images generated by $\mathbf{g}(\mathbf{U}_\mathbf{z})$ has zero density under the learned observation density $p_{\hat{\mathbf{g}}_{\text{aug}}, \hat{\mathbf{f}}_{\text{aug}}, \hat{p}_{\mathbf{s}|\mathbf{u}}}(\mathbf{g}(\mathbf{U}_\mathbf{z})) = 0$, while having non-zero density under the ground-truth observation density $p_{\mathbf{g}, \mathbf{f}_{\text{aug}}}(\mathbf{x})$: $p_{\mathbf{g}_{\text{aug}}, \mathbf{f}_{\text{aug}}, p_{\mathbf{s}|\mathbf{u}}}(\mathbf{g}(\mathbf{U}_\mathbf{z})) > 0$. This contradicts the assumption that the observation densities match everywhere. Then, $\mathbf{h}$ is surjective.

We perform change of variables on the conditional latent density:

$$\log p(\hat{\mathbf{z}}_t|\hat{\mathbf{z}}_{t-1}, \mathbf{u}) = \log p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}) + \log |\mathbf{H}_t|, \tag{14}$$

$$\sum_{k=1}^{K} \underbrace{\log p(\hat{z}_{kt}|\hat{\mathbf{z}}_{t-1}, \mathbf{u})}_{\hat{\eta}_k(\hat{z}_{kt}, \mathbf{u})} = \sum_{k=1}^{K} \underbrace{\log p(z_{kt}|\mathbf{z}_{t-1}, \mathbf{u})}_{\eta_k(z_{kt}, \mathbf{u})} + \log |\mathbf{H}_t| \tag{15}$$

$$\sum_{k=1}^{K} \hat{\eta}_k(\hat{z}_{kt}, \mathbf{u}) = \sum_{k=1}^{K} \eta_k(z_{kt}, \mathbf{u}) + \log |\mathbf{H}_t| \tag{16}$$

where $\mathbf{H}_t = \mathbf{J}_\mathbf{h}(\hat{\mathbf{z}}_t)$ is the Jacobian matrix of $\mathbf{h}$ evaluated at $\hat{\mathbf{z}}_t$. First, we take derivatives of both sides with respect to $\hat{z}_{it}$:

$$\hat{\eta}_i(\hat{z}_{it}, \mathbf{u}) = \sum_{k=1}^{K} \frac{\partial \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt}} \frac{\partial z_{kt}}{\partial \hat{z}_{it}} + \frac{\partial \log |\mathbf{H}_t|}{\partial \hat{z}_{it}} \tag{17}$$

Second, take derivatives with respect to $\hat{z}_{jt}$:

$$0 = \sum_{k=1}^{K} \left(\frac{\partial^2 \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt}^2} \frac{\partial z_{kt}}{\partial \hat{z}_{it}} \frac{\partial z_{kt}}{\partial \hat{z}_{jt}} + \frac{\partial \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt}} \frac{\partial z_{kt}^2}{\partial \hat{z}_{it} \partial \hat{z}_{jt}} \right) + \frac{\partial \log |\mathbf{H}_t|}{\partial \hat{z}_{it} \partial \hat{z}_{jt}} \tag{18}$$

Lastly, take derivatives with respect to u_l:

$$0 = \sum_{k=1}^{K} \left(\frac{\partial^3 \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt}^2 \partial u_l} \frac{\partial z_{kt}}{\partial \hat{z}_{it}} \frac{\partial z_{kt}}{\partial \hat{z}_{jt}} + \frac{\partial^2 \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt} \partial u_l} \frac{\partial z_{kt}^2}{\partial \hat{z}_{it} \partial \hat{z}_{jt}} \right), \tag{19}$$

$$= \sum_{k=1}^{K} \left(\frac{\partial^3 \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt}^2 \partial u_l} [\mathbf{H}_t]_{ki} [\mathbf{H}_t]_{kj} + \frac{\partial^2 \eta_k(z_{kt}, \mathbf{u})}{\partial z_{kt} \partial u_l} \frac{\partial z_{kt}^2}{\partial \hat{z}_{it} \partial \hat{z}_{jt}} \right), \tag{20}$$

since the Jacobian $\mathbf{H}_t$ does not depend on $\mathbf{u}$. Using the sufficient variability assumption *(A4)* for the latent states $\mathbf{z}_t$, we can plug in $2K$ values of $\mathbf{u}_1, \ldots, \mathbf{u}_{2K}$ for which the partial derivatives of the log conditional density $\eta_k(z_{kt}, \mathbf{u})$ form linearly independent vectors $\mathbf{v}(\mathbf{z}_t, \mathbf{u})$. We see that the coefficients of these linearly independent vectors have to be zero: $[\mathbf{H}_t]_{ki}[\mathbf{H}_t]_{kj} = 0$. This implies that the Jacobian matrix $\mathbf{H}_t$ of the transformation $\mathbf{z}_t = \mathbf{h}(\hat{\mathbf{z}}_t)$ has at most 1 nonzero element in its rows. Therefore, the learned latents $\hat{\mathbf{z}}_t$ are equivalent to the ground-truth latents $\mathbf{z}_t$ up to permutations and invertible, element-wise nonlinear transformations.

A.4. Proof of Theorem 2: Identifiability of the latent transition f

Here, we prove our main theoretical contribution, **Theorem 2**. We start by writing the generative process for the observation pair $(\mathbf{x}_{t-1}, \mathbf{x}_t)$:

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{bmatrix} = (\hat{\mathbf{g}}_{\texttt{aug}} \circ \hat{\mathbf{f}}_{\texttt{aug}}) \left(\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\mathbf{z}}_{t-1} \end{bmatrix} \right) = \begin{bmatrix} (\hat{\mathbf{g}} \circ \hat{\mathbf{f}})(\hat{\mathbf{z}}_{t-1}, \hat{\mathbf{s}}_t) \\ \hat{\mathbf{g}}(\hat{\mathbf{z}}_{t-1}) \end{bmatrix} \tag{21}$$

$$= (\mathbf{g}_{\texttt{aug}} \circ \mathbf{f}_{\texttt{aug}}) \circ \underbrace{(\mathbf{g}_{\texttt{aug}} \circ \mathbf{f}_{\texttt{aug}})^{-1} \circ (\hat{\mathbf{g}}_{\texttt{aug}} \circ \hat{\mathbf{f}}_{\texttt{aug}})}_{\mathbf{k}_{\texttt{aug}}} \left(\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\mathbf{z}}_{t-1} \end{bmatrix} \right) \tag{22}$$

The function $\mathbf{k}_{\texttt{aug}} : \mathbb{R}^{2K} \to \mathbb{R}^{2K}$ maps the learned pair $(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})$ to the ground-truth pair $(\mathbf{s}_t, \mathbf{z}_{t-1})$:

$$\begin{bmatrix} \mathbf{s}_t \\ \mathbf{z}_{t-1} \end{bmatrix} = \mathbf{k}_{\texttt{aug}} \left(\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\mathbf{z}}_{t-1} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) \\ \mathbf{k}_2(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) \end{bmatrix}. \tag{23}$$

Similar to the function $\mathbf{h}$ being bijective, it follows that $\mathbf{k}_{\texttt{aug}}$ is also bijective.

Now, we want to show that the augmented function $\mathbf{k}_{\texttt{aug}}$ decomposes into invertible block-wise functions $\mathbf{k}_1, \mathbf{k}_2 : \mathbb{R}^K \to \mathbb{R}^K$ such that (i) $\mathbf{k}_1$ only depends on $\hat{\mathbf{s}}_t$, i.e., $\mathbf{s}_t = \mathbf{k}_1(\hat{\mathbf{s}}_t)$ and (ii) $\mathbf{k}_2$ only depends on $\hat{\mathbf{z}}_{t-1}$, i.e., $\mathbf{z}_{t-1} = \mathbf{k}_2(\hat{\mathbf{z}}_{t-1})$. It is easy to show (ii), since $\mathbf{k}_2 = \mathrm{id}_{\mathbf{z}} \circ \mathbf{g}^{-1} \circ \hat{\mathbf{g}} \circ \mathrm{id}_{\hat{\mathbf{z}}} = \mathbf{h}$ and we have already shown in Appendix A.3 that the function $\mathbf{h} : \hat{\mathbf{z}}_t \mapsto \mathbf{z}_t$ is equal to $\mathbf{h} = \mathbf{g}^{-1} \circ \hat{\mathbf{g}}$ and bijective.

$$\begin{bmatrix} \mathbf{s}_t \\ \mathbf{z}_{t-1} \end{bmatrix} = \mathbf{k}_{\texttt{aug}} \left(\begin{bmatrix} \hat{\mathbf{s}}_t \\ \hat{\mathbf{z}}_{t-1} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) \\ \mathbf{h}(\hat{\mathbf{z}}_{t-1}) \end{bmatrix}. \tag{24}$$

To show (i), we start by performing change of variables for the transformation $\mathbf{k}_{\texttt{aug}} : (\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) \mapsto (\mathbf{s}_t, \mathbf{z}_{t-1})$:

$$\log p(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1} | \mathbf{u}) = \log p(\mathbf{k}_{\texttt{aug}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) | \mathbf{u}) + \log |\mathbf{J}_{\mathbf{k}_{\texttt{aug}}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})|, \tag{25}$$

$$= \log p([\mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}), \mathbf{h}(\hat{\mathbf{z}}_{t-1})] | \mathbf{u}) + \log |\mathbf{J}_{\mathbf{k}_{\texttt{aug}}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})|, \tag{26}$$

where $\mathbf{J}_{\mathbf{k}_{\texttt{aug}}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})$ is the Jacobian matrix for the augmented function $\mathbf{k}_{\texttt{aug}}$ evaluated at $(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})$. As the process noise is temporally independent given $\mathbf{u}$, $\hat{\mathbf{s}}_t \perp\!\!\!\perp \hat{\mathbf{z}}_{t-1} | \mathbf{u}$ and $\mathbf{s}_t \perp\!\!\!\perp \mathbf{z}_{t-1} | \mathbf{u}$, we factorize the densities in Equation (26):

$$\log p(\hat{\mathbf{s}}_t | \mathbf{u}) + \log p(\hat{\mathbf{z}}_{t-1} | \mathbf{u}) = \log p(\mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1}) | \mathbf{u}) + \log p(\mathbf{h}(\hat{\mathbf{z}}_{t-1}) | \mathbf{u}) + \log |\mathbf{J}_{\mathbf{k}_{\texttt{aug}}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})|. \tag{27}$$

The Jacobian $\mathbf{J}_{\mathbf{k}_{\text{aug}}}$ is upper block-diagonal since $\mathbf{z}_{t-1}$ does not depend on $\hat{\mathbf{s}}_t$: $\mathbf{J}_{\mathbf{k}_{\text{aug}}} = \begin{bmatrix} \frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t} & * \\ \mathbf{0} & \mathbf{H}_t \end{bmatrix}$ and its log determinant factorizes $\log|\mathbf{J}_{\mathbf{k}_{\text{aug}}}(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})| = \log|\mathbf{H}_t| + \log|\frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t}|$:

$$\log p(\hat{\mathbf{s}}_t|\mathbf{u}) + \log p(\hat{\mathbf{z}}_{t-1}|\mathbf{u}) = \log p(\mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})|\mathbf{u}) + \log p(\mathbf{h}(\hat{\mathbf{z}}_{t-1})|\mathbf{u}) + \log|\mathbf{H}_t| + \log|\frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t}|. \tag{28}$$

In addition, the noise is conditionally independent over its dimensions given $\mathbf{u}$. Therefore, we can further factorize the densities $p(\hat{\mathbf{s}}_t|\mathbf{u}) = \prod_k p(\hat{s}_{kt}|\mathbf{u})$ and $p(\mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})|\mathbf{u}) = p(\mathbf{s}_t|\mathbf{u}) = \prod_k p(s_{kt}|\mathbf{u})$ with $\mathbf{s}_t = \mathbf{k}_1(\hat{\mathbf{s}}_t, \hat{\mathbf{z}}_{t-1})$:

$$\sum_k \underbrace{\log p(\hat{s}_{kt}|\mathbf{u})}_{\hat{q}_k(\hat{s}_{kt}, \mathbf{u})} + \log p(\hat{\mathbf{z}}_{t-1}|\mathbf{u}) = \sum_k \underbrace{\log p(s_{kt}|\mathbf{u})}_{q_k(s_{kt}, \mathbf{u})} + \log p(\mathbf{h}(\hat{\mathbf{z}}_{t-1})|\mathbf{u}) + \log|\mathbf{H}_t| + \log|\frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t}|, \tag{29}$$

We take the derivative of both sides with respect to $\hat{s}_{it}$:

$$\frac{\partial \hat{q}_i(\hat{s}_{it}, \mathbf{u})}{\partial \hat{s}_{it}} = \sum_k \frac{\partial q_k(s_{kt}, \mathbf{u})}{\partial s_{kt}} \frac{\partial s_{kt}}{\partial \hat{s}_{it}} + \frac{\partial \log|\frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t}|}{\partial \hat{s}_{it}}. \tag{30}$$

Next, we take the derivative with respect to u_l with l being an arbitrary dimension:

$$\frac{\partial^2 \hat{q}_i(\hat{s}_{it}, \mathbf{u})}{\partial \hat{s}_{it} \partial u_l} = \sum_k \frac{\partial^2 q_k(s_{kt}, \mathbf{u})}{\partial s_{kt} \partial u_l} \frac{\partial s_{kt}}{\partial \hat{s}_{it}}, \tag{31}$$

since $|\frac{\partial \mathbf{s}_t}{\partial \hat{\mathbf{s}}_t}|$ does not depend on $\mathbf{u}$. Lastly, take the derivative of both sides with respect to $\hat{z}_{j,t-1}$:

$$0 = \sum_k \left(\frac{\partial^3 q_k(s_{kt}, \mathbf{u})}{\partial s_{kt}^2 \partial u_l} \frac{\partial s_{kt}}{\partial \hat{s}_{it}} \frac{\partial s_{kt}}{\partial \hat{z}_{j,t-1}} + \frac{\partial^2 q_k(s_{kt}, \mathbf{u})}{\partial s_{kt} \partial u_l} \frac{\partial^2 s_{kt}}{\partial \hat{s}_{it} \partial \hat{z}_{j,t-1}} \right). \tag{32}$$

Inspecting the Equation (32), to ensure the sufficient variability assumption *(A5)* for the process noise $\mathbf{s}_t$, the term $\frac{\partial s_{kt}}{\partial \hat{s}_{it}} \frac{\partial s_{kt}}{\partial \hat{z}_{j,t-1}} = 0$. Following a similar reasoning with Morioka et al. (2021), this implies that any dimension of $\mathbf{s}_t$ does not depend on $\hat{\mathbf{s}}_t$ and $\hat{\mathbf{z}}_{t-1}$ at the same time. Since $\mathbf{s}_t \perp\!\!\!\perp \mathbf{z}_{t-1}|\mathbf{u}$ and $\mathbf{z}_{t-1} = \mathbf{h}(\hat{\mathbf{z}}_{t-1})$, $\mathbf{s}_t$ has to depend solely on $\hat{\mathbf{s}}_t$: $\mathbf{s}_t = \mathbf{k}_1(\hat{\mathbf{z}}_{t-1}, \hat{\mathbf{s}}_t) = \mathbf{k}(\hat{\mathbf{s}}_t)$. We conclude that the augmented function $\mathbf{k}_{\text{aug}}$ decomposes into invertible block-wise functions $\mathbf{k}$ and $\mathbf{h}$: $\mathbf{k}_{\text{aug}} = [\mathbf{k}, \mathbf{h}]$

Now, let's get back to Equation (31). Denote the Jacobian matrix of function $\mathbf{k}$ by $\mathbf{J}_{\mathbf{k}}$ and its evaluation at $\hat{\mathbf{s}}_t$ by $\mathbf{J}_{\mathbf{k}}(\hat{\mathbf{s}}_t) = \mathbf{K}_t$. Take the derivative of both sides with respect to $\hat{s}_{mt}$ for some index m:

$$0 = \sum_k \left(\frac{\partial^3 q_k(s_{kt}, \mathbf{u})}{\partial s_{kt}^2 \partial u_l} [\mathbf{K}_t]_{ki} [\mathbf{K}_t]_{km} + \frac{\partial^2 q_k(s_{kt}, \mathbf{u})}{\partial s_{kt} \partial u_l} \frac{\partial^2 s_{kt}}{\partial \hat{s}_{it} \partial \hat{s}_{mt}} \right). \tag{33}$$

Inspecting the Equation (33), we see that to ensure the sufficient variability assumption *(A5)*, the product $[\mathbf{K}_t]_{ki}[\mathbf{K}_t]_{km} = 0$. This implies that each dimension s_{kt} of the true latent state depends only on a single dimension of the learned process noise $\hat{\mathbf{s}}_t$. Hence, the function $\mathbf{k}$ is equal to a composition of permutation and element-wise, invertible nonlinear transformation: $\mathbf{k} = \pi \circ T$.

Following Equation (22), we can write the relationship between the augmented functions as:

$$\mathbf{g}_{\text{aug}} \circ \mathbf{f}_{\text{aug}} \circ \mathbf{k}_{\text{aug}} = \hat{\mathbf{g}}_{\text{aug}} \circ \hat{\mathbf{f}}_{\text{aug}}, \tag{34}$$

$$\underbrace{\hat{\mathbf{g}}_{\text{aug}}^{-1} \circ \mathbf{g}_{\text{aug}}}_{\mathbf{h}_{\text{aug}}^{-1}} \circ \mathbf{f}_{\text{aug}} \circ \mathbf{k}_{\text{aug}} = \hat{\mathbf{f}}_{\text{aug}}, \tag{35}$$

$$\mathbf{h}_{\text{aug}}^{-1} \circ \mathbf{f}_{\text{aug}} \circ \mathbf{k}_{\text{aug}} = \hat{\mathbf{f}}_{\text{aug}}, \tag{36}$$

$$\tag{37}$$

where $\mathbf{h}_{\text{aug}}^{-1} = [\mathbf{h}^{-1}, \mathbf{h}^{-1}]$. We have shown that both $\mathbf{h}$ and $\mathbf{k}$ are compositions of permutations and element-wise invertible transformations. Hence, the augmented transition function $\hat{\mathbf{f}}_{\text{aug}}$ is equal to the true augmented transition function $\mathbf{f}_{\text{aug}}$ up to compositions of permutations and element-wise transformations.

A.5. Alternative Versions of Sufficient Variability Assumption

If the variable $\mathbf{u}$ is an observed categorical variable (e.g., domain indicator), the assumptions *(A4, A5)* can be written in an alternative form without partial derivatives with respect to u_l, similar to Hyvarinen et al. (2019); Yao et al. (2021). For example, for the latent states $\mathbf{z}_t$, the alternative version of the *(A4)* takes the form:

- **Sufficient variability of latent states for a categorical u (Yao et al., 2021).** For any $\mathbf{z}_t$, there exist some $2K+1$ values for $\mathbf{u}$: $\mathbf{u}_1, \ldots, \mathbf{u}_{2K}$, such that the $2K$ vectors $\mathbf{v}(\mathbf{z}_t, \mathbf{u}_{j+1}) - \mathbf{v}(\mathbf{z}_t, \mathbf{u}_j)$ with $j = 0, 1, \ldots, 2K$, are linearly independent where

$$\mathbf{v}(\mathbf{z}_t, \mathbf{u}) = \left(\frac{\partial \eta_1(z_{1t}, \mathbf{u})}{\partial z_{1t}}, \cdots, \frac{\partial \eta_K(z_{Kt}, \mathbf{u})}{\partial z_{Kt}}, \frac{\partial^2 \eta_1(z_{1t}, \mathbf{u})}{\partial z_{1t}^2}, \cdots, \frac{\partial^2 \eta_K(z_{Kt}, \mathbf{u})}{\partial z_{Kt}^2} \right) \in \mathbb{R}^{2K}. \tag{38}$$

A similar categorical version is provided in (Hyvarinen et al., 2019, Assumption 3), while the continuous version is provided in the same work (Hyvarinen et al., 2019, Appendix D).

B. Variational inference

Similar to previous works (Yao et al., 2021; 2022), we want to maximize the marginal log-likelihood $\log p(\mathbf{x}_{1:T}|\mathbf{u})$ that is obtained by marginalizing over the latent states $\mathbf{z}_{1:T}$ and process noise $\mathbf{s}_{1:T}$:

$$\log p_\theta(\mathbf{x}_{1:T}|\mathbf{u}) = \log \int_{\mathbf{z},\mathbf{s}} p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{u}) \, \mathrm{d}\mathbf{z}_{0:T} \, \mathrm{d}\mathbf{s}_{1:T}, \tag{39}$$

where we decompose the joint distribution as follows:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{u}) = p_\theta(\mathbf{z}_0) \prod_{t=1}^{T} p_\theta(\mathbf{s}_t|\mathbf{u}) \underbrace{p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t)}_{\delta(\mathbf{z}_t - \mathbf{f}(\mathbf{s}_t, \mathbf{z}_{t-1}))} p_\theta(\mathbf{x}_t|\mathbf{z}_t). \tag{40}$$

Note that the state transitions $p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t)$ are assumed to be deterministic. The above integral is intractable due to non-linear dynamics $\mathbf{f}$ and observation $\mathbf{g}$ functions. As typically done with the deep latent variable models, we approximate the log marginal likelihood by a variational lower bound, i.e., we introduce an amortized approximate posterior distribution $q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u})$ that decomposes as follows:

$$q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u}) = q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u}) \prod_{t=1}^{T} \underbrace{q_\phi(\mathbf{s}_t|\mathbf{z}_{0:t-1}, \mathbf{s}_{1:t-1}, \mathbf{x}_{1:T}, \mathbf{u})}_{q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:t})} \underbrace{q_\phi(\mathbf{z}_t|\mathbf{z}_{0:t-1}, \mathbf{s}_{1:t}, \mathbf{u})}_{q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t)} \tag{41}$$

We simplify the variational posterior $q(\mathbf{s}_t|\cdot)$ as $q_\phi(\mathbf{s}_t|\mathbf{z}_{0:t-1}, \mathbf{s}_{1:t-1}, \mathbf{x}_{1:T}, \mathbf{u}) = q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:t})$, corresponding to a filtering distribution. As in the generative model, we choose $q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t) = p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t) = \delta\left(\mathbf{z}_t - \mathbf{f}(\mathbf{s}_t, \mathbf{z}_{t-1})\right)$:

$$q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u}) = q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}) \prod_{t=1}^{T} q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:t}) p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t), \tag{42}$$

where the functional forms of the densities $q_\phi(\mathbf{z}_0|\cdot)$ and $q_\phi(\mathbf{s}_t|\cdot)$ are chosen as diagonal Gaussian distributions whose parameters are computed by recurrent neural networks. The variational lower bound takes the following form:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})} \left[\log p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) + \log \frac{p_\theta(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{u})}{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u})} \right] \tag{43}$$

$$= \underbrace{\sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{z}_t)}[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t)]}_{\text{Reconstruction term, } \mathcal{L}_R} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})} \left[\log \frac{p_\theta(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{u})}{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u})} \right]}_{\text{KL term, } \mathcal{L}_{KL}}. \tag{44}$$

The reconstruction term can easily be computed in a variational auto-encoder framework. Below, we provide the derivation of of the KL term:

$$\mathcal{L}_{KL} = \mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})}\left[\log \frac{p_\theta(\mathbf{z}_0)}{q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u})} + \log \frac{p_\theta(\mathbf{z}_{1:T}, \mathbf{s}_{1:T}|\mathbf{u})}{q_\phi(\mathbf{z}_{1:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u})}\right] \tag{45}$$

$$= -D_{KL}(q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u})||p_\theta(\mathbf{z}_0)) + \mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})}\left[\log \frac{p_\theta(\mathbf{z}_{1:T}, \mathbf{s}_{1:T}|\mathbf{u})}{q_\phi(\mathbf{z}_{1:T}, \mathbf{s}_{1:T}|\mathbf{x}_{1:T}, \mathbf{u})}\right] \tag{46}$$

$$= -D_{KL}(q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u})||p_\theta(\mathbf{z}_0)) + \mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})}\left[\sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{s}_t|\mathbf{u})p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t)}{q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:T}, \mathbf{u})q_\phi(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{s}_t)}\right] \tag{47}$$

$$= -D_{KL}(q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u})||p_\theta(\mathbf{z}_0)) + \sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{z}_{0:T}, \mathbf{s}_{1:T})}\left[\log \frac{p_\theta(\mathbf{s}_t|\mathbf{u})}{q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:T}, \mathbf{u})}\right] \tag{48}$$

$$= -D_{KL}(q_\phi(\mathbf{z}_0|\mathbf{x}_{1:T}, \mathbf{u})||p_\theta(\mathbf{z}_0)) + \sum_{t=1}^{T} \underbrace{\mathbb{E}_{q_\phi(\mathbf{s}_t, \mathbf{z}_{t-1}|\mathbf{z}_{t-2}, \mathbf{s}_{t-1}, \mathbf{x}_{1:T}, \mathbf{u})}\left[\log \frac{p_\theta(\mathbf{s}_t|\mathbf{u})}{q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:T}, \mathbf{u})}\right]}_{-\mathbb{E}_{q_\phi(\mathbf{z}_{t-1}|\mathbf{z}_{t-2}, \mathbf{s}_{t-1})}[D_{KL}(q_\phi(\mathbf{s}_t|\mathbf{z}_{t-1}, \mathbf{x}_{1:T}, \mathbf{u})||p_\theta(\mathbf{s}_t|\mathbf{u}))]} \tag{49}$$

Initial encoding. We map each observation $\mathbf{x}_t$ to an initial embedding $\mathbf{r}_t$ via an MLP or CNN depending on the input modality. Using the first $T_{\text{ic}} = 4$ initial embeddings $\mathbf{r}_{1:T_{\text{ic}}}$, an initial condition encoder (MLP) outputs the parameters of the variational posterior $q(\mathbf{z}_0|\mathbf{x}_{1:T_{\text{ic}}})$ for the initial condition $\mathbf{z}_0$. Our ablations showed that the model is rather insensitive to T_{ic}.

Sequence prediction. We start by sampling from the initial value distribution $\mathbf{z}_0 \sim q(\mathbf{z}_0|\mathbf{x}_{1:T_{\text{ic}}})$. Next, a forward sequential layer (RNN + MLP) takes the initial embeddings $\mathbf{r}_{1:t}$ up to time t and the (sampled) previous latent state $\mathbf{z}_{t-1}$ as input $[\mathbf{r}_{1:t}, \mathbf{z}_{t-1}]$, and outputs the parameters of the variational posterior $q(\mathbf{s}_t|\mathbf{x}_{1:t}, \mathbf{z}_{t-1})$ for the noise variables $\mathbf{s}_{1:T}$. For example, for the first noise variable $\mathbf{s}_1$, the variational posterior is of the form $q(\mathbf{s}_1|\mathbf{x}_1, \mathbf{z}_0)$. Subsequently, given a sample from the noise variable $\mathbf{s}_1 \sim q(\mathbf{s}_1|\mathbf{z}_0, \mathbf{x}_1)$ and the sampled initial state $\mathbf{z}_0$, we predict the next state $\mathbf{z}_1 \equiv \mathbf{f}(\mathbf{z}_0, \mathbf{s}_1)$, or more specifically: $z_{k1} \equiv f_k(\mathbf{z}_0, s_{k1})$ for $k \in 1, \ldots, K$. We model each output k of the transition function f_k as a separate MLP, to encourage the conditional independence of the latent states. We recursively compute the trajectory $\mathbf{z}_{2:T}$ of future latent states and noise posteriors $q(\mathbf{s}_{2:T})$.

Priors and ELBO computation. We assume a standard Gaussian prior for the initial condition $p(\mathbf{z}_0) = \mathcal{N}(0, I)$. The prior $p(\mathbf{s}_t|\mathbf{u}) = \prod_k p(s_{kt}|\mathbf{u})$ over the noise variables are $1D$ trainable conditional flows. To allow for multi-modal prior distributions for $1D$ noise variables s_{kt}, we choose neural spline flows as the flow architecture (Durkan et al., 2019; Stimper et al., 2023). For the computation of the KL divergence, the terms containing a conditional flow do not have closed form solutions. We compute them by a Monte Carlo approximation using the sequence samples $\{\mathbf{z}_{0:T}, \mathbf{s}_{1:T}\}$. Finally, the decoder $\mathbf{d}$ outputs the mean of our Gaussian observation model with fixed variance: $p(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{d}(\mathbf{z}_t), I)$.

C. Synthetic Data Generation

Similar to Yao et al. (2021; 2022), we set up a synthetic data experiment containing multivariate time-series. We set the dimension of $\mathbf{s}$, $\mathbf{z}$ and $\mathbf{x}$ to $K = 8$. Same as Yao et al. (2021; 2022), we use a 2-linear layer random MLP as the generative function $\mathbf{g}$, 2-linear layer random MLP as the transition function $\mathbf{f}$ and we choose $\mathbf{z}_{1:T}$ to be a second-order Markov process, i.e., $\mathbf{z}_t = \mathbf{f}(\mathbf{z}_{t-2}, \mathbf{z}_{t-1}, \mathbf{s}_t)$. The number of environments is $R = 20$. For each environment, we generate 7500/750/750 sequences as train/validation/test data following our generative model. The distribution of the process noise $\mathbf{s}_t$ is conditioned on the environment index $\mathbf{u}$. Each sequence has length $T = T_0 + T_{\text{dyn}} + T_{\text{future}} = 2 + 4 + 8 = 14$. As we have a second-order Markov process, first $T_0 = 2$ states are spared as initial states. The next 4 observations $\mathbf{x}_{1:4}$ are used for training the dynamical model. The last 8 observations $\mathbf{x}_{5:12}$ are used for assessing the performance of future estimation. We choose the future prediction horizon $T_{\text{future}} = 8$ as the double of the training sequence length. If the dynamics are truly identified, the model should predict future states well, even for a longer horizon.

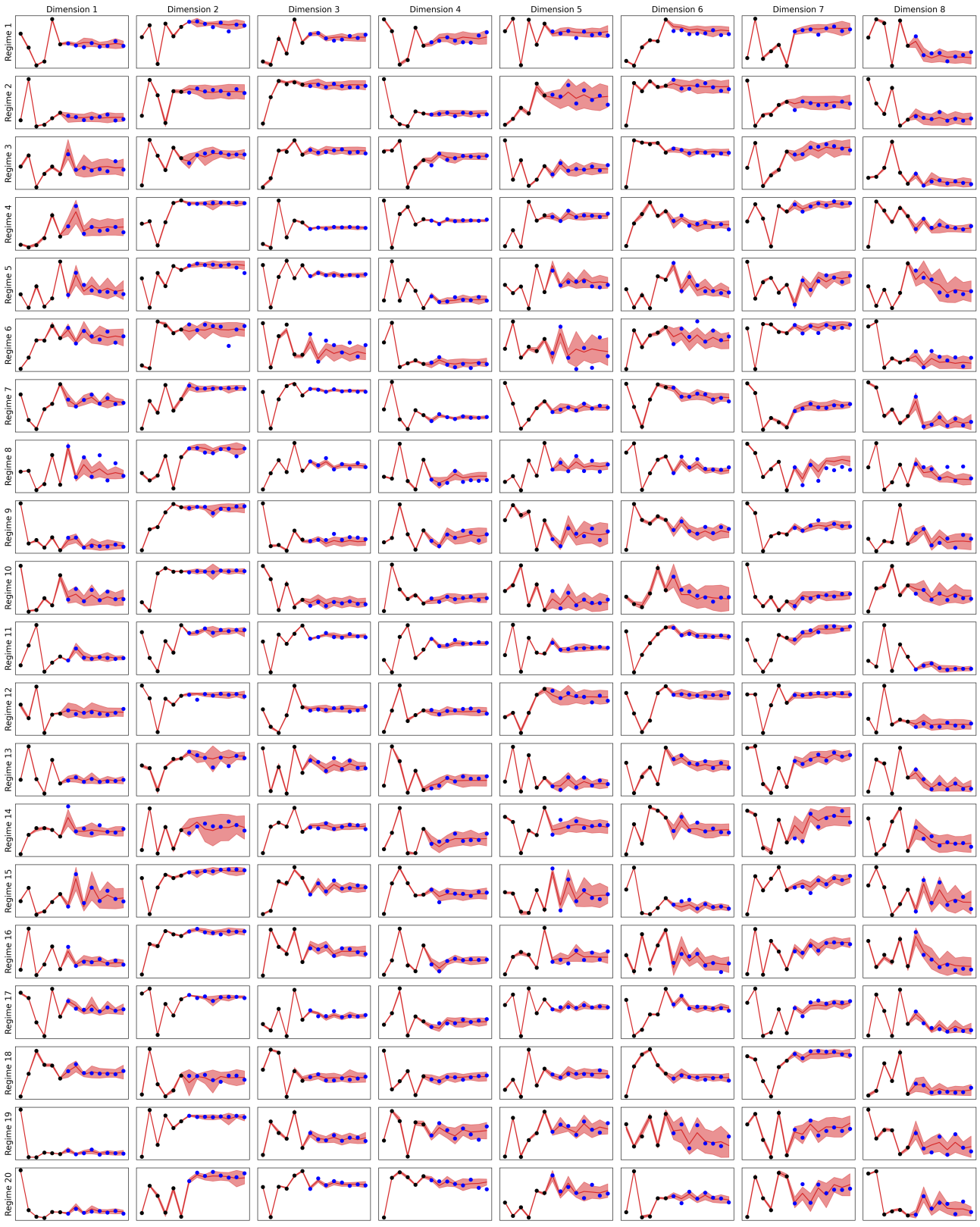

Figure 3. Extended version of Figure 2

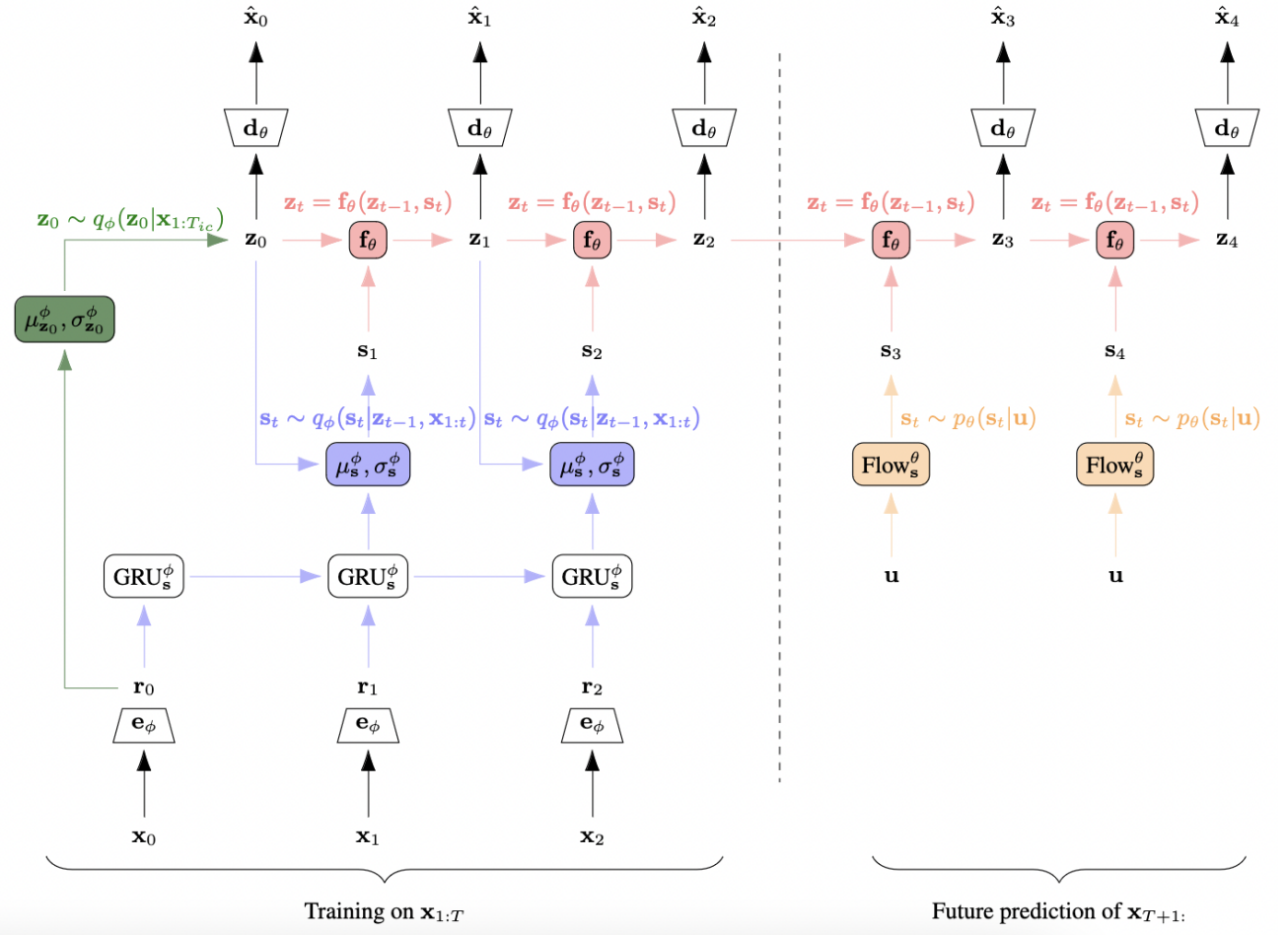

Figure 4. Diagram of the model architecture: In training, the observation $\mathbf{x}$ is passed through the encoder $\mathbf{e}_\phi$ to get the representation $\mathbf{r}$. We learn the distribution over the initial latent state $\mathbf{z}_0$ conditional on the representations of the first T_{ic} observations (in the diagram, we show $T_{ic} = 1$; in our experiments, we use $T_{\text{ic}} = 2$). The latent state is decoded by the decoder $\mathbf{d}_\theta$ to produce the predicted observation $\hat{\mathbf{x}}$ (which is trained to match the corresponding actual observation). The next value of the latent state is computed by the transition function $\mathbf{f}_\theta$, which depends both on the previous state and on the process noise $\mathbf{s}$. In training, the process noise $\mathbf{s}$ is sampled from the variational posterior that depends on the previous state as well as on the representation created by a recurrent neural network (GRU) that has received up to the current observation. In future prediction, the process noise $\mathbf{s}$ is sampled from the prior, which is a learned normalizing flow.

D. Architecture and optimization details

We optimize our model with Adam optimizer. We chose all hyperparameters for our method, two versions of LEAP and KalmanVAE with cross-validation. In particular, we performed random search as well as Bayesian optimization over learning rate, weight regularization, the number of layers in all MLPs, and latent dimensionality.

D.1. Synthetic data experiments

- **Encoder**: Below, the output of encoder_base_layer goes into encoder_rnn_layer and s_encoder.
  - (encoder_base_layer (MLP)):
    - `Linear(in_features=8, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2)`
    - `Linear(in_features=64, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2)`
    - `Linear(in_features=64, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2)`
    - `Linear(in_features=64, out_features=64, bias=True)`
  - (encoder_rnn_layer): `GRU(in=64, hidden_dim=64, output_size=64)`
  - (ic_encoder):
    - `Linear(in_features=256, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2)`
    - `Linear(in_features=64, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2)`
    - `Linear(in_features=64, out_features=16, bias=True)`
  - (s_encoder (MLP)):
    - `Linear(in_features=80, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2),`
    - `Linear(in_features=64, out_features=64, bias=True)`
    - `LeakyReLU(negative_slope=0.2),`
    - `Linear(in_features=64, out_features=16, bias=True)`
- **Decoder (MLP)**
  - `Linear(in_features=8, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2),`
  - `Linear(in_features=64, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2)`
  - `Linear(in_features=64, out_features=8, bias=True))`
- **Transition function**: Here, we consider 8 different MLPs, each of which has the following architecture:
  - `Linear(in_features=17, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2),`
  - `Linear(in_features=64, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2)`
  - `Linear(in_features=64, out_features=1, bias=True))`

D.2. Cartpole experiments

- **Encoder**
  - `Conv2d(in_channels=3, num_filter=32, kernel=3, stride=2, pad=1)`
  - `GeLU()`

  - `Conv2d(in_channels=32, num_filter=32, kernel=3, stride=2, pad=1)`
  - `GeLU()`
  - `Conv2d(in_channels=32, num_filter=64, kernel=3, stride=2, pad=1)`
  - `GeLU()`
  - `Conv2d(in_channels=64, num_filter=64, kernel=3, stride=2, pad=1)`
  - `GeLU()`
  - `Flatten()`
  - `Linear(1024,8)`

- **Decoder**
  - `Linear(8,1024)`
  - `UnFlatten(4x4x64)`
  - `ConvTranspose2d(in_channels=64, num_filter=64, kernel=3, stride=2, pad=1, output_padding=1)`
  - `GeLU()`
  - `ConvTranspose2d(in_channels=64, num_filter=32, kernel=3, stride=2, pad=1, output_padding=1)`
  - `GeLU()`
  - `ConvTranspose2d(in_channels=32, num_filter=32, kernel=3, stride=2, pad=1, output_padding=1)`
  - `GeLU()`
  - `ConvTranspose2d(in_channels=32, num_filter=1, kernel=3, stride=2, pad=1, output_padding=1)`
  - `sigmoid()`

- **Transition function**: Here, we consider 8 different MLPs, each of which has the following architecture:
  - `Linear(in_features=17, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2)`
  - `Linear(in_features=64, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2)`
  - `Linear(in_features=64, out_features=1, bias=True))`

- **Normalizing Flow:** As the nonstationary prior for the noise variables, we use $1D$ conditional normalizing flows, which are 1-layer neural spline flows conditioned on the auxiliary variable $\mathbf{u}$. Before taken as, the auxiliary variable $\mathbf{u}$ is embedded. This is done by a single linear layer with 32 dimensions in the synthetic experiments, and an MLP with the following architecture in the cartpole experiment:
  - `Linear(in_features=7, out_features=64, bias=True)`
  - `LeakyReLU(negative_slope=0.2)`
  - `Linear(in_features=64, out_features=32, bias=True)`