SEMI-SUPERVISED LEARNING AND OUTPUT PREDICTION WITH MODE-DEPENDENT DYNAMICAL VARIATIONAL AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting multidimensional time series with hidden switching dynamics is central to safety-critical decision making and data augmentation. However, standard deep generative models rarely represent operating modes explicitly. This paper augments Dynamical Variational Autoencoders (DVAEs) with a sequence of discrete latent variables governing the switching prior dynamics of the continuous latent state. The resulting Mode-Dependent DVAEs (MD-DVAEs) can be trained in a semi-supervised manner. Namely, a labeled data subset anchors discrete latent values to known modes of operation and resolves the permutation ambiguity intrinsic to unsupervised discrete latents. A novel evidence lower bound is derived that includes a term on the categorical mode encoder for supervision. Once trained, MD-DVAEs directly predict future output sequences under either a known or stochastically sampled future mode context. The approach is validated on synthetic datasets, including a comparison with switching-dynamics models and unsupervised baselines.

1 INTRODUCTION

From maneuvering targets to gene regulatory networks, systems rarely follow a single law of motion: they switch between distinct operating modes, and their observations reflect these abrupt regime changes. Switching dynamical systems (SDS) constitute a relevant paradigm, as they exhibit abrupt changes in their state transition or output models due to switching events in operating modes (e.g., a target switching from rectilinear to circular motion (Rong Li & Jilkov, 2005)). While state estimation methods for Markovian Jump Systems (MJS) (Shi & Li, 2015) have been extensively studied, such as the Interacting Multiple Model (IMM) filter (Blom & Bar-Shalom, 1988) or advanced smoothers (Sarıtaş & Orguner, 2023), they primarily focus on inference rather than generation. Consequently, they lack the ability to synthesize future trajectories or augment datasets, capabilities which are essential for tasks involving proactive prediction, simulation, or data augmentation.

Recent advances in deep generative models, particularly Dynamical Variational Autoencoders (DVAEs) (Girin et al., 2021), offer powerful tools for time-series generation. However, standard DVAEs rely solely on continuous latent states to capture dynamics. They do not explicitly represent discrete operating modes, making it difficult to model abrupt regime changes or to condition generation on known future modes. Existing attempts to incorporate discrete latents into variational frameworks rely on marginalizing the discrete variable (Dong et al., 2020). The discrete latent series is learned implicitly, which prevents conditioning output generation on a desired sequence of modes.

In this paper, we propose Mode-Dependent DVAEs (MD-DVAEs), a framework that augments DVAEs with an explicit sequence of discrete latent variables governing the switching prior of the continuous latent state. Unlike traditional MJS where modes follow a Markov chain, our model assumes that the mode sequence is white a priori (i.e., modes at distinct time steps are mutually independent), simplifying the discrete dynamics while retaining the ability to capture switching behaviors. Our key contribution is a semi-supervised learning strategy that anchors these discrete latent values to known operating modes using a labeled subset of the data. This approach resolves the permutation ambiguity and enables direct control over the generation process. By leveraging this anchoring, the trained MD-DVAE can predict future output sequences under a known future mode

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

context. We validate this approach on synthetic benchmarks, the NASCAR dataset (Linderman et al., 2017), and a home-built featuring switching dynamics (governing zeroth, first, and second-order behaviors). The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces VAEs preliminaries. Section 4 details the proposed method. Section 5 presents the experiments, and Section 6 concludes the paper.

2 Related Works

Two largely independent lines of work underpin this paper: the identification and learning of switching dynamical systems (Section 2.1), and deep generative models for time series (Section 2.2). We review each one in turn before positioning our contribution.

2.1 Identification and Learning of Switching Dynamical Systems (SDS) Models

Switching state-space models provide a general framework for dynamical systems driven by a discrete regime, with hidden Markov models and Markov jump linear systems as prominent special cases (Shi & Li, 2015; Rong Li & Jilkov, 2005). This classical literature is estimation-centric: given a model, methods recover the mode and continuous state by EM/Baum-Welch algorithm (Baum et al., 1970), by spectral methods (Hsu et al., 2009), MCMC (Rydén, 2008; Richardson & Green, 1997) or by variational Bayes (Beal, 2003; Cheng et al., 2025). Switching state-space models capture the mode as a latent variable (Murphy, 1998; Ghahramani & Hinton, 2000; Linderman et al., 2017), with recent variants adding explicit-duration or semi-Markov transitions (Ansari et al., 2021). The mode prior is itself a design choice: Markovian in classical SDS, semi-Markovian in the explicit-duration variant above.

A structural obstacle hinders these methods: an SDS is identifiable only up to a permutation of modes, which Balsells-Rodas et al. (2024) formalize as a label-switching ambiguity. Unsupervised training therefore cannot fix mode semantics.

While switching state-space models define a joint distribution over states and modes, their unsupervised modes cannot be steered, so they cannot predict outputs under a chosen regime.

2.2 Deep Learning-Based Methods for Time-series generation

Time-series generation has been approached through several families of models, including recurrent and transformer networks, generative adversarial networks, normalizing flows, diffusion models, and variational autoencoders. We focus here on the three most representative for switching dynamics: recurrent networks, diffusion models, and dynamical variational autoencoders (DVAEs).

Recurrent networks generate trajectories by recursive prediction; switching RNNs (Zhang & Saxena, 2024) introduce a discrete state that selects among different weight sets, but this state is inferred without supervision, so it carries no consistent semantics and cannot be prescribed at generation.

Diffusion-based time-series models such as TimeGrad (Rasul et al., 2021) and CSDI (Tashiro et al., 2021) reach strong predictive performance with sample-based uncertainty—via empirical sample spread rather than a closed-form distribution—but neither provides a stochastic latent regime variable. TimeGrad conditions generation on a deterministic RNN state built from past observations and given covariates, while CSDI models the observation distribution directly; in both cases there is no regime variable to identify, supervise, or prescribe, and iterative sampling costs many denoising steps per forecast step.

DVAEs instead expose an explicit continuous latent state whose transition can be conditioned on a discrete mode, matching the SDS factorization. This family (Girin et al., 2021) builds on Deep Kalman Filters (Krishnan et al., 2015), sequential neural models with stochastic layers (Fraccaro et al., 2016), and deep variational Bayes filters (Karl et al., 2017). We therefore build on this DVAE backbone and augment it with an explicit discrete mode, sampled differentiably via the Gumbel-Softmax relaxion (Jang et al., 2017) and anchored to known regimes by adapting the semi-supervised approach of Kingma et al. (2014) to the dynamic setting. This yields mode-controllable output prediction.

3 VARIATIONAL AUTO-ENCODERS

3.1 STATIC UNSUPERVISED CASE

Given a dataset $\mathcal{D} = \{X^{(i)}\}_{i=1}^{I}$ of i.i.d. samples from an unknown distribution $p_{\mathcal{D}}(X)$, Variational Autoencoders (VAEs) learn a parametric model $p_\theta(X)$ that approximates $p_{\mathcal{D}}(X)$ (Kingma & Welling, 2014). This model minimizes the Kullback-Leibler divergence $D_{\mathrm{KL}}\big(p_{\mathcal{D}}(X)\|p_\theta(X)\big)$ (Mathieu et al., 2019), or, equivalently, maximizes the log-likelihood

$$\mathrm{LL}(\theta;\mathcal{D}) = \sum\nolimits_{X^{(i)}\in\mathcal{D}} \log p_\theta(X^{(i)}) = \sum\nolimits_{i=1}^{I} \log \int p(Z) p_\theta(X^{(i)}|Z)\,\mathrm{d}Z, \tag{1}$$

where Z denotes the latent variable vector with given prior $p(Z)$, and a neural network parametrized by a vector θ defines the decoder $p_\theta(X|Z)$. Direct optimization is intractable, as objective (1) can neither be evaluated nor differentiated w.r.t. θ. To overcome this, the encoder $q_\phi(Z|X)$ is introduced, defined by a neural network parametrized by a variational parameter vector ϕ that approximates the true posterior $p_\theta(Z|X)$. Maximizing the evidence lower bound (ELBO)

$$\mathcal{L}(\theta,\phi;\mathcal{D}) = \sum\nolimits_{i=1}^{I} \Big[\mathbb{E}_{q_\phi(Z|X^{(i)})}\big[\log p_\theta(X^{(i)}|Z)\big] - D_{\mathrm{KL}}\big(q_\phi(Z|X^{(i)})\|p(Z)\big)\Big] \tag{2}$$

with respect to (θ,ϕ) provides a lower bound on $\mathrm{LL}(\theta;\mathcal{D})$. Its first term encourages the accurate reconstruction of the data in expectation over the encoder, while its second term is a Kullback-Leibler divergence penalty that encourages the encoder to remain close to the prior. When the prior and encoder are Gaussian, this Kullback-Leibler divergence admits a closed-form expression. The first reconstruction term can be estimated using the reparameterization trick (Kingma & Welling, 2014), which enables low-variance gradient estimation and training via stochastic gradient ascent.

3.2 STATIC SEMI-SUPERVISED CASE

When the dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ comprises a small subset $\mathcal{D}_l = \{(X^{(i)}, Y^{(i)})\}$ with discrete class labels and a large unlabeled subset $\mathcal{D}_u = \{X^{(j)}\}$, the above static VAE can be extended to the semi-supervised setting (Kingma et al., 2014). In the latent representation (Y, Z), Y captures the discrete class identity while the continuous-valued variable Z captures the remaining label-independent style of the data. The generative model comprises a fixed categorical prior $p(Y)$ over the finite label set, a fixed continuous prior $p(Z)$, and the decoder $p_\theta(X|Y,Z)$ to be learnt. The encoder is factorized accordingly as $q_\phi(Y,Z|X) = q_\phi(Y|X)\, q_\phi(Z|X,Y)$.

Define $\mathcal{H}(q_\phi(Y|X))$ as the entropy of the categorical encoder $q_\phi(Y|X)$, and

$$\ell(\theta,\phi;X,Y) = \mathbb{E}_{q_\phi(Z|X,Y)}\big[\log p_\theta(X|Y,Z)\big] - D_{\mathrm{KL}}\big(q_\phi(Z|X,Y)\,\|\,p(Z)\big) + \log p(Y). \tag{3}$$

On labeled pairs, the ELBO takes the same form as in the static case. On unlabeled data, the discrete unobserved latent Y cannot be sampled via the reparametrization trick, so it is marginalized out exactly. The respective ELBOs are given by, with $|\mathcal{D}|$ the cardinality of $\mathcal{D}$:

$$\mathcal{L}_l(\theta,\phi;\mathcal{D}_l) = \frac{1}{|\mathcal{D}_l|}\sum\nolimits_{(X^{(i)},Y^{(i)})\in\mathcal{D}_l} \ell(\theta,\phi;X^{(i)},Y^{(i)})\,; \tag{4}$$

$$\mathcal{L}_u(\theta,\phi;\mathcal{D}_u) = \frac{1}{|\mathcal{D}_u|}\sum\nolimits_{X^{(j)}\in\mathcal{D}_u} \Big[\sum\nolimits_{Y}\big[q_\phi(Y|X^{(j)})\ell(\theta,\phi;X^{(j)},Y)\big] + \mathcal{H}\big(q_\phi(Y|X^{(j)})\big)\Big]. \tag{5}$$

Since this objective does not directly train $q_\phi(Y|X)$ to predict correctly on labeled data (the reconstruction and prior terms in ℓ do not depend on the encoder's ability to infer Y when it is already given), an explicit classification loss is added, weighted by a hyperparameter α:

$$\mathcal{J}^{\alpha}(\theta,\phi;\mathcal{D}) = \mathcal{L}_u(\theta,\phi;\mathcal{D}_u) + \mathcal{L}_l(\theta,\phi;\mathcal{D}_l) + \alpha\frac{1}{|\mathcal{D}_l|}\sum\nolimits_{(X^{(i)},Y^{(i)})\in\mathcal{D}_l}\big[-\log q_\phi(Y^{(i)}|X^{(i)})\big]. \tag{6}$$

3.3 DYNAMIC CASE

In the dynamic setting, the goal is to learn a generative model $p_\theta(X_{0:T})$ for a dataset $\mathcal{D} = \{X_{0:T}^{(i)}\}_{i=1}^{I}$ of I i.i.d. sequences $X_{0:T}^{(i)} = (X_0^{(i)},\ldots,X_T^{(i)})$ over a time interval $[0,T]$. A latent sequence $Z_{0:T}$,

whose prior $p_\theta(Z_{0:T})$ is to be learnt, is introduced to capture the dynamics of the data. Dynamical Variational Autoencoders (DVAEs) (Girin et al., 2021) provide a unifying framework, accommodating various conditional dependency structures between $X_{0:T}$ and $Z_{0:T}$. The log-likelihood is

$$\mathrm{LL}(\theta;\mathcal{D}) = \sum\nolimits_{X_{0:T}^{(i)}\in\mathcal{D}} \log p_\theta(X_{0:T}^{(i)}) = \sum\nolimits_{i=1}^{I} \log \int p_\theta(Z_{0:T}) p_\theta(X_{0:T}^{(i)}|Z_{0:T})\,\mathrm{d}Z_{0:T}. \quad (7)$$

As in the static case, it is lower-bounded by the ELBO, which extends to the dynamic setting as

$$\mathcal{L}(\theta,\phi;\mathcal{D}) = \sum\nolimits_{i=1}^{I} \Big[\mathbb{E}_{q_\phi(Z_{0:T}|X_{0:T}^{(i)})}\big[\log p_\theta(X_{0:T}^{(i)}|Z_{0:T})\big] - D_{\mathrm{KL}}\big(q_\phi(Z_{0:T}|X_{0:T}^{(i)})\|p_\theta(Z_{0:T})\big)\Big]. \quad (8)$$

Using Bayes chain rule and d-separation arguments (Pearl, 1988), factorization and network structures of the decoder $p_\theta(X_{0:T}|Z_{0:T})$ and encoder $q_\phi(Z_{0:T}|X_{0:T})$ can be exhibited. If the latent process is Markov, $p_\theta(Z_{0:T}) = p_\theta(Z_0)\prod_{k=1}^{T} p_\theta(Z_k|Z_{k-1})$, and the observations are conditionally independent given the latents, $p_\theta(X_{0:T}|Z_{0:T}) = \prod_{k=0}^{T} p_\theta(X_k|Z_k)$, then the model reduces to a Deep Kalman Filter (Krishnan et al., 2015). In this case, the encoder is often factorized as $q_\phi(Z_{0:T}|X_{0:T}) = q_\phi(Z_0|X_{0:T})\prod_{k=1}^{T} q_\phi(Z_k|X_{0:T}, Z_{k-1})$, enabling amortized inference over time.

4 Proposed method: Mode-Dependent DVAE (MD-DVAE)



(a) Labeled case: $C_{0:T}$ is an observed mode sequence (b) Unlabeled case: $C_{0:T}$ is a latent mode sequence

Figure 1: Graphical models of the proposed Mode-Dependent DVAE (MD-DVAE).

A switching system is considered (Figure 1), whose continuous hidden latent state vector sequence $Z_{0:T}$ obeys a Markov transition model $p_\theta(Z_t|Z_{t-1}, C_t)$ conditioned on a mode index $C_t \in \{0,\dots,M-1\}$. The sequence of modes $C_{0:T} = C_0,\dots,C_T$ is assumed i.i.d. from a categorical prior with a given prior probability vector $\pi = [\pi_1,\dots,\pi_M]$, e.g., with $\pi_1,\dots,\pi_M = \frac{1}{M}$ when modes are equiprobable a priori. Each vector X_t in the observation sequence $X_{0:T}$ is assumed to be sampled from an emission model $p_\theta(X_t \mid Z_t)$ common to all modes. Given a dataset of observations, the aim is to learn a generative model, henceforth called Mode-Dependent DVAE (MD-DVAE), that captures both the observation model and mode-conditioned continuous dynamics, and enables the generation of predicted outputs over the subsequent time range $[T+1, T+K]$ under either a known or stochastically sampled future mode sequence $C_{T+1:T+K}$.

4.1 Generative and inference models

In view of the aforementioned Bayes chain rule and d-separation properties, the joint distribution of the latent and observed sequences factorizes as

$$p_\theta(Z_{0:T}, C_{0:T}, X_{0:T}) = \prod_{t=0}^{T} \underbrace{\big[p_\theta(X_t|Z_t)\big]}_{\text{emission}} p(Z_0|C_0) \prod_{t=1}^{T} \underbrace{\big[p_\theta(Z_t|Z_{t-1},C_t)\big]}_{\text{transition}} p(C_0) \prod_{t=1}^{T} \underbrace{\big[p(C_t)\big]}_{\text{mode prior}}. \quad (9)$$

The involved distributions are selected as

$$\begin{aligned} &\text{[Mode priors] } p(C_t) = \mathrm{Cat}(C_t;\pi), \qquad \text{[Decoder] } p_\theta(X_t|Z_t) = \mathcal{N}\big(X_t;\mu_\theta(Z_t), R_\theta(Z_t)\big), \\ &\text{[Mode-conditioned prior dynamics] } p_\theta(Z_t|Z_{t-1},C_t) = \mathcal{N}\big(Z_t;\mu_\theta(Z_{t-1},C_t), Q_\theta(Z_{t-1},C_t)\big). \end{aligned} \quad (10)$$

Depending on whether $C_{0:T}$ is observed or not (labeled/unlabeled data), the encoders are chosen as:

$$q_\phi(Z_{0:T}|C_{0:T}, X_{0:T}) = p(Z_0)\prod\nolimits_{t=1}^{T}\big[q_\phi(Z_t|Z_{t-1}, C_t, X_{t:T})\big]; \quad (11)$$

$$q_\phi(Z_{0:T}, C_{0:T}|X_{0:T}) = q_\phi(Z_{0:T}|C_{0:T}, X_{0:T})p(C_0)\prod\nolimits_{t=1}^{T}\big[q_\phi(C_t|Z_{t-1}, X_{t:T})\big]. \quad (12)$$

The involved distributions are selected as

$$\begin{aligned} q_\phi(C_t|Z_{t-1}, X_{t:T}) &= \mathrm{Cat}\big(C_t; \pi_\phi(Z_{t-1}, X_{t:T})\big)\,; \\ q_\phi(Z_t|Z_{t-1}, C_t, X_{t:T}) &= \mathcal{N}\big(Z_t; \mu_\phi(Z_{t-1}, C_t, X_{t:T}), Q_\phi(Z_{t-1}, C_t, X_{t:T})\big). \end{aligned} \tag{13}$$

Motivations on the selection of the prior and encoder initial distributions and of the encoder structure are explained in Appendix A.1.

4.2 Semi-supervised objective

Defining

$$C_{\mathrm{KL}}(Z_{t-1}, X_{t:T}) = D_{\mathrm{KL}}\big(q_\phi(C_t|Z_{t-1}, X_{t:T}) \,\|\, p(C_t)\big), \tag{14}$$

$$Z_{\mathrm{KL}}(Z_{t-1}, C_t, X_{t:T}) = D_{\mathrm{KL}}\big(q_\phi(Z_t|Z_{t-1}, C_t, X_{t:T}) \,\|\, p_\theta(Z_t|Z_{t-1}, C_t)\big), \tag{15}$$

the ELBOs for the unlabeled and labeled cases can be written as (See Appendix A)

$$\mathcal{L}_u(\theta, \phi; \mathcal{D}_u) = \frac{1}{|\mathcal{D}_u|} \sum_{X^{(i)}_{0:T} \in \mathcal{D}_u} \ell_u(\theta, \phi; X^{(i)}_{0:T}) \text{ and } \mathcal{L}_l(\theta, \phi; \mathcal{D}_l) = \frac{1}{|\mathcal{D}_l|} \sum_{(X^{(i)}_{0:T}, C^{(i)}_{0:T}) \in \mathcal{D}_l} \ell_l(\theta, \phi; X^{(i)}_{0:T}, C^{(i)}_{0:T}) \tag{16}$$

with

$$\begin{aligned} \ell_u(\theta, \phi; X_{0:T}) = \sum_{t=0}^{T} \mathbb{E}_{q_\phi(Z_{0:t}, C_{0:t}|X_{0:T})}\big[\log p_\theta(X_t|Z_t)\big] \\ - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(Z_{0:t-1}, C_{0:t}|X_{0:T})}\big[Z_{\mathrm{KL}}(Z_{t-1}, C_t, X_{t:T})\big] \\ - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(Z_{0:t-1}, C_{0:t-1}|X_{0:T})}\big[C_{\mathrm{KL}}(Z_{t-1}, X_{t:T})\big]\,; \end{aligned} \tag{17}$$

$$\begin{aligned} \ell_l(\theta, \phi; X_{0:T}, C_{0:T}) = \sum_{t=0}^{T} \mathbb{E}_{q_\phi(Z_{0:t}|C_{0:T}, X_{0:T})}\big[\log p_\theta(X_t|Z_t)\big] \\ - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(Z_{0:t-1}|C_{0:T}, X_{0:T})}\big[Z_{\mathrm{KL}}(Z_{t-1}, C_t, X_{t:T})\big] + \sum_{t=0}^{T} \log p(C_t). \end{aligned} \tag{18}$$

The label C_t appears in ℓ_l only as a conditioning variable, so the mode encoder $q_\phi(C_t|Z_{t-1}, X_{t:T})$ is never trained on labeled data. To make the encoder learn from the labels, a mode-prediction term is added that maximizes its likelihood w.r.t. the observed modes, leading to the objective ($\alpha > 0$)

$$\mathcal{J}^\alpha(\theta, \phi; \mathcal{D}) = \mathcal{L}_u(\theta, \phi; \mathcal{D}_u) + \mathcal{L}_l(\theta, \phi; \mathcal{D}_l) + \frac{\alpha}{|\mathcal{D}_l|} \sum_{(X^{(i)}_{0:T}, C^{(i)}_{0:T}) \in \mathcal{D}_l} \sum_{t=1}^{T} \big[\log q_\phi(C_t \mid Z_{t-1}, X^{(i)}_{t:T})\big]. \tag{19}$$

This added cross-entropy (mode-prediction) objective anchors each mode index to a specific mode and resolves the permutation ambiguity of the discrete latent variable.

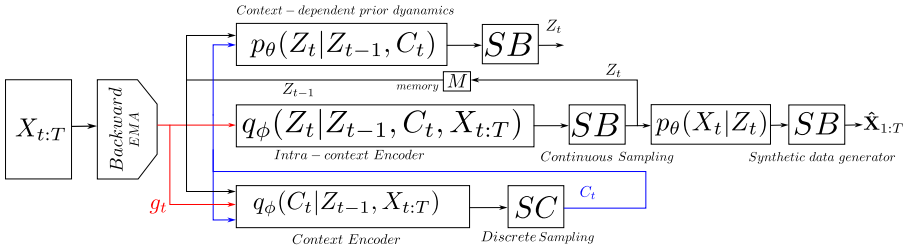
4.3 Major implementation aspects

Importantly, other factorisations would have been possible for (11) and (12). Ours imposes an iterative sampling process during the training loop on the two latent variables. In spite of this computational effort, this solution features better convergence properties.

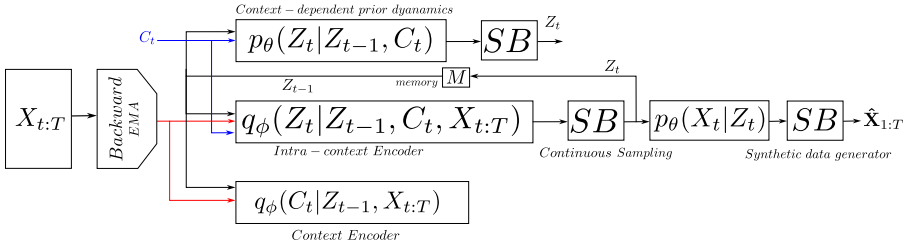
To cope with the conditioning of encoders at time t on the future observations $X_{t:T}$, a fixed-size summary of $X_{t:T}$ is implemented. Although this task usually relies on RNN models, we use an Exponential Moving Average (EMA) filter, which prevents convergence problems. This keeps the encoder input size independent of t. The architecture is summarized in Appendix A.

Figures 2a and 2b respectively showcase the training of the proposed model. The continuous and discrete sampling blocks respectively use the reparameterization trick and the Gumbel-Softmax relaxation (Jang et al., 2017) to guarantee differentiability.

To mitigate posterior collapse, free-bits regularization (Kingma et al., 2016; Bowman et al., 2016) is applied. In other words, KL terms Z_{KL} in (17)–(18) are replaced by $\max(Z_{\mathrm{KL}}, \lambda)$, where λ is the



(a) Unlabeled training paradigm: C_t & Z_t are sampled respectively using a Gumbel-Softmax SC, and the reparameterization trick respectively. The samples are fed back recursively into the encoder at the next step. $X_{t:T}$ is compressed by an EMA filter.



(b) Labeled training paradigm: The observed context C_t conditions the context-dependent prior dynamics.

Figure 2: Summary of the two training paradigms of MD-DVAE.

minimum information budget allocated to the continuous latent. This encourages the model to use the latent space while preserving the overall probabilistic regularization.

Both C_{KL} and Z_{KL} KL terms are annealed during training according to a sigmoid schedule. Neither contributes to the loss at the start: C_{KL} is introduced first and fully annealed, and only after it has reached its full weight does Z_{KL} begin to be introduced.

Because each mode is anchored to a true label, generation can be steered by supplying a desired future mode sequence $C_{T+1:T+K}$. The mode-conditioned prior then the decoder rolls out the corresponding latent dynamics $Z_{T+1:T+K}$, then the decoder $p_\theta(X_t|Z_t)$ produces observations that follow the requested dynamics.

5 Experiments

5.1 Benchmarks, baselines and metrics

5.1.1 NASCAR rSLDS benchmark

As a first baseline, we use the NASCAR recurrent switching linear dynamical system (rSLDS) of (Linderman et al., 2017): a car drives clockwise around an oval whose plane is partitioned into four regions (right turn, left turn, top straight, bottom straight), each defining a linear regime.

We simulate $n = 1000$ independent on-track laps of $m = 126$ steps from a fixed parameter set and record, at every step, the true regime together with a $D_{\mathrm{obs}} = 10$-dimensional projection of the two-dimensional position.

The generative model, the regime partition and the construction protocol are detailed in Appendix C. These one-lap sequences of $m = 126$ steps form the training set. Evaluation uses longer held-out sequences of size $n = 252$, which is equivalent to two-laps per sequence.

5.1.2 Variable order case study

As a second benchmark, we synthesize a fully labeled dataset of $N_{\mathrm{exp}} = 10,000$ independent trajectories of length $T = 200$. Each trajectory is partitioned into five consecutive modes (regimes), and each step carries the context label $C_t \in \{0, \ldots, 4\}$. The two-dimensional ground-truth latent state $Z_t^{\mathrm{GT}} \in \mathbb{R}^2$ follows, within each mode, one of five linear Gaussian stochastic processes: isotropic

white noise, a stationary AR(1) process, white noise with an anisotropic (oriented) covariance, and two non-stationary divergent-mean processes whose drift amplitude grows along the segment. An ablation parameter $\rho = 10^{-1}$ sets the final signal-to-noise ratio of the two non-stationary modes. The regime length is not uniform, each regime lasts between 20 and 50 timesteps.

The 20-dimensional observations are generated from the latent state by a fixed linear Gaussian emission model, $X_t = A^\top Z_t^{\mathrm{GT}} + B + \varepsilon_t$. Each of the 20 measurement channels has its own noise level, calibrated so that three standard deviations equal a fraction $\tau = 10^{-1}$ of that channel's mean value. More details on the generation process are given in Appendix B.

5.1.3 State-space baselines

As baselines we compare against three state-space models built on the `dynamax` library (Linderman et al., 2025): SSM A, an EM-fitted linear Gaussian state space model (LGSSM); SSM B, a physics-informed switching LGSSM using the known emission model $(A, B, \boldsymbol{\sigma}^2)$; and SSM C, a switching linear dynamical system (SLDS) with a Rao-Blackwellized particle filter.

5.1.4 Evaluation protocol and metrics

Both benchmarks follow the same protocol. The model is trained on its training set which is split as 80% training 20% validation and evaluated on ten sequences from a held-out set: NASCAR laps of length $T = 252$ (two laps) and synthetic trajectories of length $T = 200$. The training is semi-supervised with a supervision rate of 50%.

The baseline models differ in the information they receive: SSM A has no mode identification variable and learns a stationary model, SSM B and the MD-DVAE receive the true mode/context sequence, and SSM C infers the regime with a Rao-Blackwellized particle filter in reconstruction and samples it from a fitted Markov chain in generation. We distinguish two applications.

Sequence reconstruction Aimed at reconstructing an exact sequence, it conditions on the full observation sequence (posterior smoothing) and is scored by the standardized observation RMSE together with the regime classification accuracy when a discrete posterior is produced (the MD-DVAE encoder and SSM C; SSM A and SSM B have none).

Unconditional sequence generation Aimed at producing new trajectories that are statistically comparable to the dataset, it compares $M = 200$ prior samples per model with an independent reference ensemble. RMSE would not be a proper metric in this case, as what we aim to assess is whether or not we can produce new data that statistically resembles our data rather than trying to reconstruct a sequence. The work of Stenger et al. (2024) provides an overview of evaluation measures for synthetic time-series generation. We report two metrics: Maximum Mean Discrepancy (MMD) and Jensen-Shannon (JS) distance. MMD compares two distributions directly from samples, without requiring an explicit probability density estimate, by using a Gaussian radial basis function (RBF) kernel. The JS distance, on the other hand, measures the similarity between two probability distributions based on their divergence from their average distribution. Both metrics are lower-is-better. They are computed using benchmark-specific projections, grids, and kernel bandwidths. They are reported against the same benchmark's real-vs-real split-half floor, obtained by splitting the real data into two halves and measuring the discrepancy between them to quantify the intrinsic variability of the real dataset.

5.2 Reconstruction and mode classification

All models are tested on full held-out sequences. For DVAE and SSM B, the true sequence of modes is given to the models. SSM C infers it using a particle filter. SSM A does not use a mode-variable. With known emission model and dynamic, the state-space smoother showcases very good results on both datasets, and the margin is larger on the synthetic data: SSM B and SSM C reconstruct to 0.051, versus 0.203 for the DVAE posterior (Table 1). Both models obtain extremely good classification results with a slight edge for the DVAE model. This is expected given the semi-supervised regime.

	NASCAR		Variable order case	
Model	RMSE ↓	Acc. ↑	RMSE ↓	Acc. ↑
DVAE posterior	0.111	**0.987**	0.203	**0.991**
SSM A	0.094	—	0.125	—
SSM B	0.092	—	**0.051**	—
SSM C	**0.091**	0.872	**0.051**	0.949

Table 1: RMSE reconstruction error and classification accuracy on both datasets.

	NASCAR		Variable order case	
Generator	MMD ↓	JS ↓	MMD ↓	JS ↓
DVAE, reference schedule, mean decode	0.647	0.039	0.780	0.105
DVAE, reference schedule, sampled decode	0.678	**0.017**	**0.402**	**0.005**
SSM A, EM-LGSSM prior	0.758	0.267	0.746	0.373
SSM B, reference schedule	**0.478**	0.071	0.448	0.015
SSM C, sampled Markov chain	0.678	0.170	0.654	0.048
Real-vs-real split-half floor	0.092±0.030	0.001±0.001	0.085±0.016	0.0008±0.0003

Table 2: Unconditional generation on both benchmarks with a reference schedule (only for MD-DVAE and SSM B); lower is better. The MMD feature map is detailed in Appendix D.

5.3 Unconditional generation

Our goal here is to provide our model with the desired mode-sequence, and generate a sequence that statistically resembles our dataset. Similarly to the previous section, SSM-B and DVAE are fed the desired mode sequences, while SSM-C infers. For the DVAE model, we report the metrics for two conditions: the mean output X_t (in which we use the mean of the probability distribution $p_\theta(X_t|Z_t)$); the sampled output, in which we sample from said probability distribution.

With the true mode schedule supplied to both the MD-DVAE and SSM B, the best generator depends on the benchmark: the DVAE with sampled decoding is best on the synthetic data (MMD 0.402, JS 0.005), while on NASCAR the physics SSM B attains the lowest MMD (0.478 vs. 0.678 for the DVAE sampled decode) and the DVAE keeps the lowest occupancy JS (0.017 vs. 0.071). SSM A is the worst on both: being mode-blind, its prior cannot fit the switching data and collapses to a linear model. All generators lie roughly 5-$8\times$ above their respective split-half floors, showcasing the residual gap to real data (Table 2). Occupancy maps expose two failure modes: the stationary SSM A collapses mass onto the origin and misses the divergent regimes (variable order JS 0.373), whereas the DVAE mean decode under-disperses its marginal (variable order JS 0.105) and sampling from the decoder covariance repairs it (0.005). Because the schedule is now oracle, the regime distribution and transition statistics match by construction, so the remaining distances isolate the continuous dynamics and emission rather than the schedule. The aggregate MMD is nevertheless a weak discriminator (all generators occupy a narrow band 0.48-0.76 against a ≈ 0.09 floor), so it should be read jointly with the JS metric.

Variable regime durations. A distinctive property of the synthetic benchmark is that modes persist for a variable number of steps: the five segments of a trajectory have lengths drawn in $[20, 50]$, so a fixed-window or fixed-duration generator cannot represent the data. Because the MD-DVAE conditions its continuous prior on the mode sequence, it yields the best synthetic-benchmark scores (Table 2). Figure 3 overlays the real measurement trajectories with the MD-DVAE and SSM B trajectories generated under the same variable-length schedule: six highest-variance dimensions, the samples were chosen among those where the difference is visible. The MD-DVAE follows the reference across regimes of different durations, whereas SSM B tends to overshoot in the divergent regimes. Across all 200 reference windows, the RMSE is 1.56 for the MD-DVAE *vs* 1.71 for SSM B.

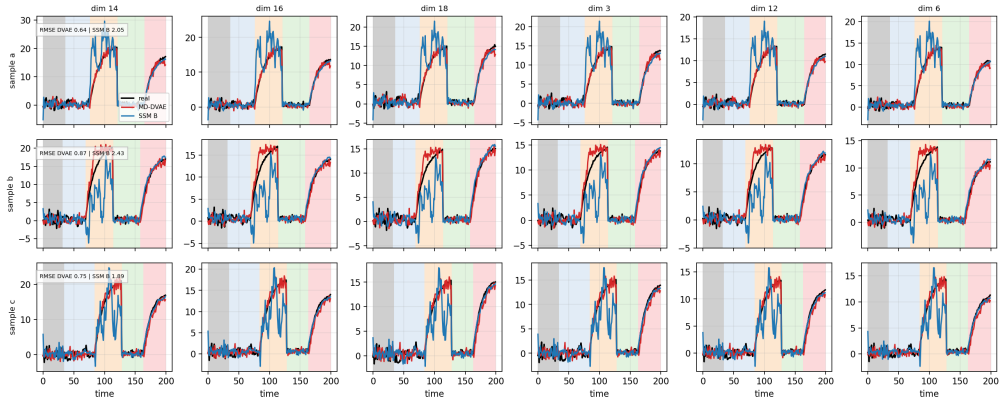


Figure 3: Measurement-space trajectories under the same true (variable-length) mode schedule on the synthetic benchmark (six highest-variance dimensions, one sampled trajectory): real reference (black), MD-DVAE (red) and SSM B (blue), with shaded bands marking the modes; the three samples are randomly chosen.

6 Conclusion

This paper introduces Mode-Dependent Dynamical Variational Autoencoders (MD-DVAEs), a generative framework that augments DVAEs with an explicit sequence of discrete latent variables governing the switching dynamics of the continuous latent state. We derived a novel evidence lower bound for this construction and proposed a semi-supervised training strategy in which a labeled subset anchors the discrete latents to known operating modes. This anchoring resolves the permutation ambiguity. On synthetic benchmarks, the MD-DVAE reconstructs and classifies regimes competitively and generates trajectories whose statistics approach those of the data.

The key benefit of this design is control over generation. The trained model can be driven at generation time by a prescribed future mode schedule: supplying a desired sequence $C_{T+1:T+K}$ rolls out the corresponding mode-conditioned latent dynamics and produces output sequences that follow the requested regimes. This is precisely what standard DVAEs and unsupervised switching models cannot offer, since their inferred modes carry no consistent semantics and cannot be steered. The ability to synthesize trajectories under a chosen regime makes MD-DVAEs particularly valuable for data augmentation and scenario simulation in safety-critical settings.

Several limitations remain and outline directions for future work. Under our current equations factorization, training relies on successive sampling of the discrete and continuous latents, which is computationally heavy and slows the training process. We are currently re-exploring other factorizations to resolve these kind of problems. We also fixed the initial latent priors. This works well with long sequences, in which the contribution of the initial latents is minimal. However, this will probably not generate well on short sequences. Hence, we plan to revisit the decoder equations to remove this hypothesis. Furthermore, we currently limit our study to second-order models or less. It would be interesting to explore how our model would generalize beyond that. Finally, our validation is restricted to synthetic benchmarks in which modes are sharply defined, whereas modes in real data are often fuzzier and only partially observable. Extending the framework to such settings and evaluating it on real-world data remains an important next step.

Acknowledgements

This is a placeholder text. This section was not finalized to avoid breaking double-blind review. This work was funded by the project XXXXXXX.
This work was granted access to HPC resources XXXX.

AI USE STATEMENT

Generative AI tools were used in several supporting capacities during this work. Large language models were consulted for bibliography work, through agents or online LLM portals. They were also used for visualization questions, including figure generation, some minor code automation. LLMs also served as a first review step: we prompted them to summarize our paper and to report what they found most interesting, a phase that proved helpful in organizing the manuscript as a whole. This was particularly the case for Appendices B and C in which the LLMs helped verify that all the results are well in accordance with our generation codes. Finally, LLMs were used as an English polishing tool, both for grammar checking and for suggesting alternative phrasings when we felt an idea was not coming across clearly, as well as to verify that our notation and terminology were used consistently throughout.

REFERENCES

Abdul Fatir Ansari, Konstantinos Benidis, Richard Kurle, Ali Caner Türkmen, Harold Soh, Alexander J. Smola, Yuyang Wang, and Tim Januschowski. Deep explicit duration switching models for time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Carles Balsells-Rodas, Yixin Wang, and Yingzhen Li. On the identifiability of switching dynamical systems. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2639–2672, 2024.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

Matthew J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University College London, 2003.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 1537-274X. doi: 10.1080/01621459.2017.1285773. URL http://dx.doi.org/10.1080/01621459.2017.1285773.

H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. 33(8):780–783, 1988. ISSN 1558-2523. doi: 10.1109/9.1299. URL https://ieeexplore.ieee.org/document/1299/.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning (CoNLL)*, pp. 10–21, 2016.

Cheng Cheng, Jean-Yves Tourneret, and Sinan Yıldırım. A variational bayesian marginalized particle filter for jump markov nonlinear systems with unknown measurement noise parameters. *Signal Processing*, 233:109954, 2025. doi: 10.1016/j.sigpro.2025.109954.

Zhe Dong, Bryan Seybold, Kevin Murphy, and Hung Bui. Collapsed amortized variational inference for switching nonlinear dynamical systems. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2638–2647. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/dong20e.html.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2199–2207, 2016.

Zoubin Ghahramani and Geoffrey E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):831–864, 2000.

Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends® in Machine Learning*, 15(1-2):1–175, 2021. ISSN 1935-8237. doi: 10.1561/2200000089. URL http://dx.doi.org/10.1561/2200000089.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. In *Conference on Learning Theory (COLT)*, pp. 247–260, 2009.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations (ICLR)*, 2017.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3581–3589, 2014.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4743–4751, 2016.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Scott W. Linderman, Andrew C. Miller, Ryan P. Adams, David M. Blei, Liam Paninski, and Matthew J. Johnson. Recurrent switching linear dynamical systems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Scott W. Linderman, Peter Chang, Giles Harper-Donnelly, Aleyna Kara, Xinglong Li, Gerardo Duran-Martin, and Kevin Murphy. Dynamax: A python package for probabilistic state space modeling with JAX. *Journal of Open Source Software*, 10(108):7069, 2025. doi: 10.21105/joss.07069.

Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *36 th International Conference on Machine Learning*, Long Beach, CA, 2019.

Kevin P. Murphy. Switching Kalman filters. Technical report, University of California, Berkeley, 1998.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Elsevier, Morgan Kaufmann, 1988.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning (ICML)*, pp. 8857–8868, 2021.

Sylvia Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59 (4):731–792, 1997.

X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking. part v. multiple-model methods. 41(4):1255–1321, 2005. ISSN 1557-9603. doi: 10.1109/TAES.2005.1561886. URL `https://ieeexplore.ieee.org/document/1561886/`.

Tobias Rydén. Em and mcmc methods for hidden markov models. *Statistics Surveys*, 2:119–145, 2008.

E. Sarıtaş and U. Orguner. Expectation propagation with context adjustment for smoothing of jump markov linear systems. *IEEE Trans. on Aerospace and Electronic Systems*, 59(5), 2023.

Peng Shi and Fanglai Li. A survey on markovian jump systems: Modeling, analysis and filtering. *International Journal of Systems Science*, 46(13):2317–2330, 2015.

Michael Stenger, Robert Leppich, Ian Foster, Samuel Kounev, and André Bauer. Evaluation is key: a survey on evaluation measures for synthetic time series. *Journal of Big Data*, 11(1):66, May 2024. ISSN 2196-1115. doi: 10.1186/s40537-024-00924-7. URL `https://doi.org/10.1186/s40537-024-00924-7`.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 24804–24816, 2021.

Yongxu Zhang and Shreya Saxena. Inference of neural dynamics using switching recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-4178.

A Derivations of the factorized evidence lower bounds

A.1 Selection of the encoder structure and of the prior and encoder initial distributions

Modeling assumptions made in Section 4.1 are explained below.

Mode-conditioned prior and posterior of the latent variables at initial time The prior $p(C_0)$ can be set to the uniform categorical distribution. We assume that $p(Z_0|C_0)$ is independent of C_0 and thus writes as $p(Z_0)$. Typically, it can be set to the standard multivariate Gaussian distribution. Regarding the encoders (11)–(12), we set $q_\phi(C_0|X_{0:T}) = p(C_0)$ and $q_\phi(Z_0|C_0, X_{0:T}) = p(Z_0)$. We believe that this tradeoff between mathematical rigor and simplicity can be justified if dataset sequences are long enough, in that the misidentification of their mode at initial time does not prevent its learning at subsequent times (i.e., contribution of initial conditions is negligible). Furthermore, to learn encoders mimicking the exact posteriors $p(C_0|X_{0:T})$ and $q(Z_0|C_0, X_{0:T})$, specific networks should be implemented, inducing computational overhead and hindering the training convergence.

Selection of the structure of the encoder for the unlabeled and labeled case For the unlabeled case, the exact posterior of the joint discrete-continuous latent sequences factorizes as

$$p_\theta(Z_{0:T}, C_{0:T}|X_{0:T}) = p_\theta(Z_0|C_0, X_{0:T}) \prod_{t=1}^{T} \big[p_\theta(Z_t|Z_{0:t-1}, C_{0:t}, X_{0:T})\big] \\ p_\theta(C_0|X_{0:T}) \prod_{t=1}^{T} \big[p_\theta(C_t|Z_{0:t-1}, C_{0:t-1}, X_{t:T})\big]. \quad (20)$$

It can be readily noticed that, in view of d-seperation properties, the corresponding encoder (12) strictly respects this structure. However, strictly speaking, for the labeled case, the exact posterior of the continuous latent sequence obeys the following structure

$$p_\theta(Z_{0:T}|C_{0:T}, X_{0:T}) = p_\theta(Z_0|C_{0:T}, X_{0:T}) \prod_{t=1}^{T} \big[p_\theta(Z_t|Z_{0:t-1}, C_{0:T}, X_{0:T})\big]. \quad (21)$$

Apart from the above initialization simplifications, the corresponding encoder (11) approximates $p_\theta(Z_t|Z_{0:t-1}, C_{0:T}, X_{0:T}) = p_\theta(Z_t|Z_{t-1}, C_{t:T}, X_{t:T})$ by a model of the form $q_\phi(Z_t|Z_{t-1}, C_t, X_{t:T})$. This assumption of a weak mutual dependence between Z_t and the future modes $C_{t+1:T}$ given $Z_{t-1}, C_t, X_{t:T}$ seems reasonable, all the more if the changes are abrupt and sporadic. In some sense, it constitutes the cornerstone of semi-supervision, in that it avoids training another network. Note that, strictly speaking, the encoder is a *selected* distribution that does not need to replicate the genuine posterior (Blei et al., 2017). Dropping the $C_{t+1:T}$ sequence is similar to (Krishnan et al., 2015) in which only past inputs are considered in the DKF structure.

A.2 Unlabeled data

In this case both $Z_{0:T}$ and $C_{0:T}$ are latent, and the variational posterior is (12). So,

$$\begin{aligned} &\mathcal{D}_{\mathrm{KL}}\big(q_\phi(Z_{0:T}, C_{0:T}|X_{0:T}) \,\big\|\, p_\theta(Z_{0:T}, C_{0:T}|X_{0:T})\big) \\ &\quad = \mathbb{E}_{q_\phi(Z_{0:T}, C_{0:T}|X_{0:T})}\big[\log q_\phi(Z_{0:T}, C_{0:T}|X_{0:T}) - \log p_\theta(Z_{0:T}, C_{0:T}, X_{0:T})\big] \\ &\quad + \log p_\theta(X_{0:T}) \\ &\quad = -\ell_u(\theta, \phi; X_{0:T}) + \log p_\theta(X_{0:T}), \end{aligned} \quad (22)$$

and the unlabeled ELBO is

$$\begin{aligned}\ell_u(\theta,\phi;X_{0:T}) =& \mathbb{E}_{q_\phi(Z_{0:T},C_{0:T}|X_{0:T})}\Big[\\ &\underbrace{\log p(Z_0|C_0)-\log q(Z_0|C_0,X_{0:T})}_{0}+\underbrace{\log p(C_0)-\log q(C_0|X_{0:T})}_{0}\\ &+\sum_{t=0}^{T}\log p_\theta(X_t|Z_t)\\ &+\sum_{t=1}^{T}\log p_\theta(Z_t|Z_{t-1},C_t)+\sum_{t=1}^{T}\log p(C_t)\\ &-\sum_{t=1}^{T}\log q_\phi(Z_t|Z_{t-1},C_t,X_{t:T})-\sum_{t=1}^{T}\log q_\phi(C_t|Z_{t-1},X_{t:T})\Big].\end{aligned} \tag{23}$$

Applying d-separation and suppressing variables that are marginalized in expectations leads to

$$\begin{aligned}&\ell_u(\theta,\phi;X_{0:T})\\ &=\sum_{t=0}^{T}\mathbb{E}_{q_\phi(Z_{0:t},C_{0:t}|X_{0:T})}\big[\log p_\theta(X_t|Z_t)\big]\\ &\quad+\sum_{t=1}^{T}\mathbb{E}_{q_\phi(Z_{0:t-1},C_{0:t}|X_{0:T})}\big[-Z_{\mathrm{KL}}(Z_{t-1},C_t,X_{t:T})\big]\\ &\quad+\sum_{t=1}^{T}\mathbb{E}_{q_\phi(Z_{0:t-1},C_{0:t-1}|X_{0:T})}\big[-C_{\mathrm{KL}}(Z_{t-1},X_{t:T})\big].\end{aligned} \tag{24}$$

A.3 Labeled data

When $C_{0:T}$ is observed, only $Z_{0:T}$ is latent, and the variational posterior is (11). So,

$$\begin{aligned}&\mathcal{D}_{\mathrm{KL}}\big(q_\phi(Z_{0:T}|C_{0:T},X_{0:T})\,\big\|\,p_\theta(Z_{0:T}|C_{0:T},X_{0:T})\big)\\ &=\mathbb{E}_{q_\phi(Z_{0:T}|C_{0:T},X_{0:T})}\big[\log q_\phi(Z_{0:T}|C_{0:T},X_{0:T})-\log p_\theta(Z_{0:T},C_{0:T},X_{0:T})\big]\\ &+\log p_\theta(C_{0:T},X_{0:T})\\ &=-\ell_l(\theta,\phi;X_{0:T},C_{0:T})+p_\theta(C_{0:T},X_{0:T}),\end{aligned} \tag{25}$$

and the labeled ELBO is

$$\begin{aligned}\ell_l(\theta,\phi;X_{0:T},C_{0:T}) =& \mathbb{E}_{q_\phi(Z_{0:T}|C_{0:T},X_{0:T})}\Big[\underbrace{\log p(Z_0|C_0)-\log q(Z_0|C_0,X_{0:T})}_{0}\\ &+\sum_{t=0}^{T}\log p_\theta(X_t|Z_t)+\sum_{t=1}^{T}\log p_\theta(Z_t|Z_{t-1},C_t)+\sum_{t=0}^{T}\log p(C_t)\\ &-\sum_{t=1}^{T}\log q_\phi(Z_t|Z_{t-1},C_t,X_{t:T})\Big]\end{aligned} \tag{26}$$

Applying d-separation and suppressing variables that are marginalized in expectations leads to

$$\begin{aligned}&\ell_l(\theta,\phi;X_{0:T},C_{0:T})\\ &=\sum_{t=0}^{T}\mathbb{E}_{q_\phi(Z_{0:T}|C_{0:T},X_{0:T})}\big[\log p_\theta(X_t|Z_t)\big]\\ &\quad+\sum_{t=1}^{T}\mathbb{E}_{q_\phi(Z_{0:t-1}|C_{0:T},X_{0:T})}\big[-Z_{\mathrm{KL}}(Z_{t-1},C_t,X_{t:T})\big]\\ &\quad+\sum_{t=0}^{T}\log p(C_t).\end{aligned} \tag{27}$$

B SYNTHETIC DATASET GENERATION

This appendix documents the synthetic dataset used throughout the experiments. It is built in two stages: the latent trajectories are first synthesised, and are then raised to the observation space through a fixed emission model.

B.1 LATENT TRAJECTORIES

We generate $N_{\text{exp}} = 10{,}000$ independent two-dimensional latent trajectories $Z_{1:T} \in \mathbb{R}^2$ of length $T = 200$. Each trajectory is divided into five consecutive regimes (contexts), and every step carries the context label $C_t \in \{0, \dots, 4\}$. In the parameterisation every regime has a fixed length; the five lengths are drawn in $[20, 50]$ and sum to $T = 200$. Within a regime the latent state follows a distinct linear Gaussian recursion driven by independent standard normal draws. The five regimes are summarised in Table 3 and detailed below.

Context 0 (isotropic white noise).

$$V_t \sim \mathcal{N}(0, R), \qquad R = \sigma^2 I_2, \qquad \sigma = 0.20, \tag{28}$$

so that $R = \operatorname{diag}(0.09, 0.09)$ originally and $R = \operatorname{diag}(0.04, 0.04)$ currently.

Context 1 (stationary AR(1)). Each component is a stationary first-order process with unit gain,

$$V_{t+1}^{(k)} = a_k V_t^{(k)} + (1 - a_k) b_k \, \eta_t^{(k)}, \qquad V_1^{(k)} \sim \mathcal{N}(0, \sigma_k^2), \tag{29}$$

with $a_1 = 0.70$, $a_2 = 0.05$, and $\sigma_1 = 0.20$, $\sigma_2 = 0.10$, and $b_k = \sigma_k \sqrt{1 - a_k^2}(1 - a_k)$; the stationary covariance is $Q = \operatorname{diag}\big((1 - a_1^2)\sigma_1^2, (1 - a_2^2)\sigma_2^2\big)$, i.e. $Q = \operatorname{diag}(0.020400, 0.009975)$.

Contexts 2 and 4 (non-stationary, divergent mean). The latent noise state $Y_t \in \mathbb{R}^4$ obeys the block recursion

$$Y_{t+1} = A_d Y_t + B_d L_Q \, \xi_t, \qquad A_d = \begin{bmatrix} a & 1-a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & 1-a \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{30}$$

where $a = \exp\big(-1(L_{\text{seg3}})\big)$ with L_{seg3} the length of the third segment, $L_Q = \operatorname{diag}\big(\sqrt{Q_{Y,1}}, \sqrt{Q_{Y,2}}\big)$ and $\xi_t \sim \mathcal{N}(0, I_2)$. Only the first and third components are observed, i.e. $V_t = \big(Y_t^{(1)}, Y_t^{(3)}\big)$. The state starts with standard deviation $\sigma_{\text{ini}} = 10^{-2}$ and drifts towards the final mean μ_{fin}; the final 3σ band is calibrated to the fraction ρ of the drift,

$$\sigma_{\text{fin},k} = \frac{\rho \, \mu_{\text{fin},k}}{3g}, \qquad g = \sqrt{\frac{S_{\text{hidden}}}{L_{\text{seg}} - 1}}, \qquad S_{\text{hidden}} = \sum_{j=1}^{L_{\text{seg}}-2} \big(1 - a^j\big)^2, \tag{31}$$

and the driving covariance is set to $Q_{Y,k} = \sigma_{\text{fin},k}^2 (L_{\text{seg}} - 1)$. Context 2 uses $\mu_{\text{fin}} = (2, 4)$; context 4 uses $\mu_{\text{fin}} = (3, 2)$. In the parameterisation L_{seg} varies, so σ_{fin} and Q_Y follow from the same calibration with the realised regime length.

Context 3 (correlated white noise). A zero-mean white noise whose covariance is aligned with the South-West-North-East axis,

$$R = R_\theta \, \operatorname{diag}\big(\sigma_{\text{SO-NE}}^2, \sigma_{\text{SE-NO}}^2\big) \, R_\theta^\top, \qquad R_\theta = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{32}$$

with $\sigma_{\text{SO-NE}} = 0.10$ and $\sigma_{\text{SE-NO}} = 0.02$, i.e. $R = \begin{bmatrix} 0.0052 & 0.0048 \\ 0.0048 & 0.0052 \end{bmatrix}$.

Context	Regime	Noise type	Parameters
0	1	white isotropic	$R = \sigma^2 I_2$, $\mu = 0$, $\sigma = 0.20$
1	2	stationary AR(1)	$a = (0.70, 0.05)$, $Q = \mathrm{diag}((1 - a_k^2)\sigma_k^2)$, $\sigma = (0.20, 0.10)$
2	3	non-stationary	$a = \exp(-1(L_{\text{seg}}3))$, $\mu_{\text{fin}} = (2, 4)$, $\sigma_{\text{ini}} = 10^{-2}$
3	4	white correlated	$\sigma_{\text{SE-NO}} = 0.02$, $\sigma_{\text{SO-NE}} = 0.10$
4	5	non-stationary	$a = \exp(-1(L_{\text{seg}}3))$, $\mu_{\text{fin}} = (3, 2)$, $\sigma_{\text{ini}} = 10^{-2}$

Table 3: Latent noise regimes. The parameter $\rho = 10^{-1}$ controls the final signal-to-noise ratio of the two non-stationary regimes (contexts 2 and 4). Values are given as .

B.2 Observation model

Each latent trajectory is mapped to a 20-dimensional observation,

$$X_t = A^\top Z_t + B + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}\big(0, \mathrm{diag}\big(\sigma_1^2, \ldots, \sigma_{20}^2\big)\big)\,, \tag{33}$$

with a fixed loading matrix $A \in \mathbb{R}^{2\times 20}$ and a bias $B \in \mathbb{R}^{20}$. The measurement noise is calibrated from the analytical latent means over the whole dataset,

$$\bar{z}_1 = 0.53698899, \qquad \bar{z}_2 = 0.67123624, \tag{34}$$

through the noise-free mean measurement $\bar{y} = A^\top[\bar{z}_1, \bar{z}_2]^\top + B$, so that

$$\sigma_i = \frac{\tau\,|\bar{y}_i|}{3}, \tag{35}$$

i.e. the 3σ band equals the fraction $\tau = 10^{-1}$ of the mean value. The noise is therefore heteroscedastic across dimensions but independent between dimensions. The loading coefficients, biases and noise standard deviations are listed in Table 4; they are identical for the two parameterisations.

i	A_{1i}	A_{2i}	B_i	σ_i	σ_i^2
1	1.000000	0.000000	0.050620	0.019587	3.8365×10^{-4}
2	0.000000	1.000000	0.692770	0.045467	2.0672×10^{-3}
3	3.318434	2.195778	0.126330	0.112739	1.2710×10^{-2}
4	1.693094	1.532779	0.825410	0.092115	8.4851×10^{-3}
5	1.955768	1.416933	0.332420	0.077791	6.0515×10^{-3}
6	1.978317	2.584262	0.437050	0.107801	1.1621×10^{-2}
7	1.700506	1.949929	0.056304	0.075944	5.7675×10^{-3}
8	0.442285	0.054765	0.407860	0.022737	5.1699×10^{-4}
9	0.442384	2.560862	0.900110	0.095220	9.0669×10^{-3}
10	0.583195	2.621194	0.385430	0.081935	6.7133×10^{-3}
11	0.398075	2.349667	0.346980	0.071264	5.0786×10^{-3}
12	2.497933	2.187176	0.197620	0.100236	1.0047×10^{-2}
13	0.735224	1.616092	0.473650	0.065108	4.2390×10^{-3}
14	3.370740	3.659361	0.262010	0.150945	2.2784×10^{-2}
15	0.067023	0.381458	0.256830	0.018296	3.3473×10^{-4}
16	2.561312	3.085273	0.573960	0.134010	1.7959×10^{-2}
17	2.154131	2.373665	0.518940	0.108966	1.1874×10^{-2}
18	3.863281	1.902465	0.423400	0.125831	1.5834×10^{-2}
19	3.621759	0.916072	0.911280	0.115701	1.3387×10^{-2}
20	2.454495	0.799679	0.223430	0.069275	4.7990×10^{-3}

Table 4: Loading matrix A, bias B, and measurement-noise standard deviations σ_i and variances σ_i^2.

B.3 Key constants

The generation is fully determined by the constants collected in Table 5.

Symbol	Value	Meaning
T	200	trajectory length
L_{seg}	$[20, 50]$	steps per regime
N_{exp}	10,000	number of trajectories
ρ	10^{-1}	final-SNR ablation (latent regimes)
τ	10^{-1}	measurement-noise ablation
σ_{ini}	10^{-2}	initial noise std (contexts 2 and 4)
$\bar{z}_1, \bar{z}_2$	$0.53698899,\ 0.67123624$	analytical latent means

Table 5: Key constants of the synthetic dataset

C NASCAR DATASET

This appendix documents the second benchmark, derived from the canonical NASCAR recurrent switching linear dynamical system (rSLDS) of (Linderman et al., 2017).

C.1 GENERATIVE MODEL

Let $C_t \in \{0, \dots, K-1\}$ be the discrete regime and $Z_t \in \mathbb{R}^D$ the continuous latent state (the car's position). The rSLDS is

$$\begin{aligned} C_t \mid C_{t-1}, Z_{t-1} &\sim \mathrm{Cat}\big(\,\mathrm{softmax}(\mathbf{r} + RZ_{t-1})\big), \\ \mathbf{w}_t &\sim \mathcal{N}(\mathbf{0}, \sigma^2 I), \\ Z_t &= A_{C_t} Z_{t-1} + \mathbf{b}_{C_t} + \mathbf{w}_t, \\ \mathbf{v}_t &\sim \mathcal{N}(\mathbf{0}, \eta^2 I_{D_{\text{obs}}}), \\ X_t &= HZ_t + \mathbf{v}_t, \\ Z_1 &\sim \mathcal{N}(\boldsymbol{\mu}_0, \sigma^2 I), \qquad C_1 \sim \mathrm{Cat}(\boldsymbol{\pi}_0), \end{aligned} \tag{36}$$

with an orthonormal emission matrix H ($H^\top H = I_D$). The car drives clockwise around an oval: the discrete regime is a function of the position, which partitions the plane into four regions, as summarised in Table 6.

k	regime	region	dynamics
0	right turn	$z_1 > +2$	rotation about $[+2, 0]$
1	left turn	$z_1 < -2$	rotation about $[-2, 0]$
2	top straight	$\lvert z_1\rvert \le 2,\ z_2 > 0$	drift $[+0.10, 0]$
3	bottom straight	$\lvert z_1\rvert \le 2,\ z_2 < 0$	drift $[-0.25, 0]$

Table 6: Regime partition of the NASCAR rSLDS.

The parameters are $K = 4$, $D = 2$, $D_{\text{obs}} = 10$, $\sigma^2 = 10^{-4}$ and $\eta^2 = 10^{-2}$; the emission matrix H is fixed by a random seed.

C.2 DATASET CONSTRUCTION

The original benchmark is a single long trajectory. Because the soft recurrent transition permits rare off-track excursions over long horizons, we instead draw many independent short trajectories and retain only on-track ones:

1. one parameter set is fixed so that every sequence shares the same emission matrix H, while each sequence uses its own random seed;
2. one lap of $m = 126$ steps is simulated (the limit-cycle period is 125.9 steps); a trajectory whose radius ever exceeds 8 is rejected and redrawn;
3. observations $X_t = HZ_t + \mathbf{v}_t$ are emitted and the true regimes C_t are recorded;

4. the procedure is repeated for $n = 1000$ sequences and stacked into a headerless table with, per sequence, $D_{\text{obs}} = 10$ observation columns followed by one integer regime column used as the context label.

The resulting dataset reproduces the canonical process: the regime shares $[0.18, 0.35, 0.33, 0.14]$ and the near-deterministic cyclic transitions agree with a 10,000-step reference run. The only deviations are sampling choices—many one-lap sequences instead of one long run, and conditioning on remaining on the oval, which lowers the per-dimension observation variance relative to the raw long run.

For evaluation the model is trained on these one-lap ($m = 126$) sequences, while the held-out sequences used for inference span two laps ($T = 252$).

D MMD FEATURE MAP

The feature MMD is a standard Gaussian-RBF two-sample statistic applied to hand-designed per-trajectory summaries. Each trajectory is projected to $\boldsymbol{x} = C^\top \boldsymbol{y}$ and mapped to a feature vector containing:

- an 8×6 normalized occupancy histogram over $x_1 \in [-5, 5]$, $x_2 \in [-4, 4]$ (48 features);
- per-coordinate means and standard deviations (4 features);
- per-coordinate 10th, 50th, and 90th percentiles (6 features);
- per-coordinate lag-one autocorrelations (2 features);
- mean and standard deviation of the step-to-step speed (2 features).

Features are standardized with the reference-ensemble mean and standard deviation, with a 0.03 lower bound on the scale. The RBF kernel is

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\frac{\|\boldsymbol{u} - \boldsymbol{v}\|_2^2}{2\sigma^2}\right), \tag{37}$$

with σ^2 set by the median-distance heuristic on the pooled feature sample, and the reported value is the biased V-statistic

$$\widehat{\mathrm{MMD}}^2(P, Q) = \frac{1}{n_P^2} \sum_{i,i'} k(\boldsymbol{p}_i, \boldsymbol{p}_{i'}) + \frac{1}{n_Q^2} \sum_{j,j'} k(\boldsymbol{q}_j, \boldsymbol{q}_{j'}) - \frac{2}{n_P n_Q} \sum_{i,j} k(\boldsymbol{p}_i, \boldsymbol{q}_j), \tag{38}$$

followed by the square root and clipping at zero.