Deep Spectral Learning of Embedded Latent Transfer Operators for Stochastic Dynamical Systems

Ryogo Tanaka[1,2] **Yoshinobu Kawahara**[1,2]

[1]Graduate School of Information Science and Technology, The University of Osaka , Osaka , Japan
[2]Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan

Abstract

We propose a spectral learning method for stochastic nonlinear dynamical systems represented with embedded latent transfer operators in deep feature spaces. We instantiate the method as Deep Spectral Encoder (DSE), an operator-based latent state-space model in which a time-invariant neural encoder implements learnable nonlinear feature maps from observations, and these features define Markovian latent states whose temporal evolution and observation mapping are described by the transfer and observation operators, respectively. Functional canonical correlation analysis in a learnable Galerkin-projected feature space provides state coordinates from past and future observations, and the two linear operators are estimated on the state coordinates as ridge-regularized closed-form solutions that coincide with Galerkin projections of the associated covariance operators. On this representation, we generalize sequential Bayesian filtering and Koopman spectral mode decomposition in feature space. Experiments on several scenarios show stable and superior performance with sequential Bayesian filtering and dynamic mode decomposition baselines even under noise and partial observability.

1 INTRODUCTION

Understanding and forecasting the evolution of stochastic nonlinear dynamics using time-series data is a central yet technically demanding problem across science and engineering. Real-world time series data often appear non-stationary in the observation space due to nonlinear evolution, observational noise, and partial observability, so that simple linear models fail to capture the underlying structure.

A classical approach to forecasting in dynamical systems is sequential Bayesian filtering, typified by the Kalman filter [Kalman, 1960] and its nonlinear extensions, such as the extended and unscented Kalman filters, and kernel Kalman filters [Anderson and Moore, 1979, Doucet et al., 2001, Gebhardt et al., 2019]. When state space models are learned from data, common strategies infer latent state trajectories using expectation maximization or variational approximations, or learn deep latent state models with amortized inference [Krishnan et al., 2015, Becker et al., 2019]. Continuous-time latent stochastic dynamics have also been modeled through parametric or variational latent SDE formulations [Hasan et al., 2022, Duncker et al., 2019].Another line of work adopts *spectral learning* to identify a latent Markov representation directly from empirical moments of past and future observations, avoiding iterative latent state inference. In statistics and control, stochastic realization and subspace identification construct predictive state coordinates via canonical correlation analysis (CCA) between past and future blocks and estimate dynamics by regression [Akaike, 1975, Desai et al., 1985, Katayama, 2005]. In machine learning, spectral algorithms were developed for hidden Markov models [Hsu et al., 2009] and extended to nonlinear settings through kernel-based stochastic realization and Hilbert space embeddings [Kawahara et al., 2006, Song et al., 2010, Ke et al., 2025]. These estimators are closed-form and statistically well behaved, but their effectiveness depends on feature representations that are usually fixed in advance.

We propose Deep Spectral Learning (DSL), a method for stochastic dynamical systems that learns embedded latent transfer operators in finite-dimensional deep feature spaces while retaining an operator-based state space formulation with closed-form operator estimation. As a concrete instantiation of DSL, we introduce Deep Spectral Encoder (DSE), in which a time-invariant neural encoder maps observation windows to block-wise features and functional CCA in a learnable Galerkin-projected feature space yields predictive state coordinates. On these coordinates, the transfer and observation operators are estimated as ridge-regularized closed-form solutions via a deep-feature two-stage proce-

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:6610–6630.

dure [Xu et al., 2021], and these solutions coincide with Galerkin projections of associated conditional covariance operators onto the subspaces spanned by the learned features. This representation generalizes sequential Bayesian filtering in feature space by propagating and updating embedded state laws with the learned operators, and it also generalizes Koopman spectral analysis via eigen-decomposition of the learned transfer operator, respectively in closed-forms. Experiments on several scenarios demonstrate robust forecasting and spectrum recovery under noise and partial observability, with superior performance to the baseline methods.

In this framework, Hilbert-space embeddings and conditional covariance operators serve as the standard population-level operator view used to interpret the estimators, while the role of DSL is to make the finite-dimensional representation spaces used by spectral realization learnable. DSE specifies these spaces by neural encoder and dictionary maps: the encoder and block-feature maps define the past–future CCA problem that yields predictive state coordinates, while the state and observation dictionaries define the spaces spanned by these functions for closed-form transfer and observation operator estimation. It differs from the kernel-based realization in ELTO [Ke et al., 2025], whose empirical representation is expressed through kernel evaluations over samples in an RKHS; DSE instead learns explicit finite-dimensional Galerkin coordinates and performs operator regression in these realized coordinates. The feature-space filtering and Koopman-style spectral analysis developed below are downstream uses of this learned operator representation.

The remainder of this paper is organized as follows. Section 2 reviews the mean embeddings, covariance operators, and deep-feature-based two-stage estimation of linear operators. Section 3 develops the operator-based state-space formulation and stochastic realization with deep features, and Section 4 presents the DSE model and its learning algorithm. We develop sequential state estimation and spectral mode decomposition with our operator-based representation, respectively, in Section 5 and in Section 6, with experimental results, and Section 7 concludes the paper. The code is available at `https://github.com/uosaka-mlsyslab/DeepSpectralEncoder`.

2 BACKGROUND

2.1 HILBERT SPACE EMBEDDING OF CONDITIONAL DISTRIBUTIONS

Embedding probability laws, including conditional ones, into Hilbert spaces allows us to realize expectations and conditional expectations as inner products and linear operators on feature spaces. This subsection recalls the corresponding Hilbert-space embedding framework for random variables, including mean elements that represent marginal laws and the covariance, cross-covariance, and conditional covariance operators [Baker, 1973, Hsing and Eubank, 2015]. These constructions are used as standard background and provide the population-level operator view for the finite-dimensional estimators introduced later.

Let $\mathbf{x}$ and $\mathbf{y}$ be random variables taking values $\boldsymbol{x} \in \mathbb{X}$ and $\boldsymbol{y} \in \mathbb{Y}$, where $\mathbb{X}$, $\mathbb{Y}$ are real vector spaces, with corresponding marginal distributions $P_{\mathbf{x}}$, $P_{\mathbf{y}}$ and joint distribution $P_{\mathbf{xy}}$. Let $\mathbb{H}$ and $\mathbb{G}$ be separable Hilbert spaces, and consider feature maps $\phi_x : \mathbb{X} \to \mathbb{H}$, $\psi_y : \mathbb{Y} \to \mathbb{G}$ such that $\mathbb{E}\left[\|\phi_x(\mathbf{x})\|^2\right] < \infty$ and $\mathbb{E}\left[\|\psi_y(\mathbf{y})\|^2\right] < \infty$. We represent the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$ in feature spaces by their mean elements. The mean element of $\mathbf{x}$ is defined by

$$\mu_{\mathbf{x}} := \mathbb{E}\big[\phi_x(\mathbf{x})\big] = \int \phi_x(\boldsymbol{x})\, dP_{\mathbf{x}}(\boldsymbol{x}) \in \mathbb{H},$$

and $\mu_{\mathbf{y}} \in \mathbb{G}$ is defined analogously [Smola et al., 2007]. The square-integrability assumptions ensure that these expectations are well-defined as Hilbert space elements. This construction is called a mean embedding and it satisfies $\mathbb{E}[f(\mathbf{x})] = \mathbb{E}[\langle f, \phi_x(\mathbf{x})\rangle_{\mathbb{H}}] = \langle f, \mu_{\mathbf{x}}\rangle_{\mathbb{H}}$ for all $f \in \mathbb{H}$. Under suitable conditions on the feature map, the mean embedding is injective and therefore characterizes the underlying distribution [Sriperumbudur et al., 2008].

Mean embeddings provide a first-order summary of the marginal distributions in the feature spaces. To describe second-order relationships in the feature space, we introduce covariance and cross-covariance operators, defined as expectations of rank-one operators induced by the feature maps [Baker, 1973].

Definition 1 *Recall that $\phi_x(\mathbf{x}) \in \mathbb{H}$ and $\psi_y(\mathbf{y}) \in \mathbb{G}$ denote the feature maps of $\mathbf{x}$ and $\mathbf{y}$ introduced above, and assume they are square-integrable. Then, the covariance operator $\mathcal{C}_{\mathbf{x}} : \mathbb{H} \to \mathbb{H}$ and cross-covariance operator $\mathcal{C}_{\mathbf{yx}} : \mathbb{H} \to \mathbb{G}$ are defined by*

$$\mathcal{C}_{\mathbf{x}} := \int \phi_x(\boldsymbol{x}) \otimes \phi_x(\boldsymbol{x}) dP_{\mathbf{x}}(\boldsymbol{x}),$$
$$\mathcal{C}_{\mathbf{yx}} := \int \psi_y(\boldsymbol{y}) \otimes \phi_x(\boldsymbol{x}) dP_{\mathbf{xy}}(\boldsymbol{x}, \boldsymbol{y}),$$

where the rank-one operator $a \otimes b : \mathbb{H} \to \mathbb{G}$ is defined by $(a \otimes b)w := \langle w, b\rangle_{\mathbb{H}}\, a$ for all $b,\ w \in \mathbb{H}$ and $a \in \mathbb{G}$. For any $f \in \mathbb{H}$ and $g \in \mathbb{G}$,

$$\mathbb{E}_{\mathbf{xy}}[f(\mathbf{x})g(\mathbf{y})] = \langle \mathcal{C}_{\mathbf{yx}}\, f, g\rangle_{\mathbb{G}} = \left\langle f, \mathcal{C}^*_{\mathbf{yx}}\, g\right\rangle_{\mathbb{H}},$$

which is the standard representation of cross-covariance operators for Hilbert-space-valued random variables.

In the integrable function space, conditional expectation can be viewed as the orthogonal projection onto the closed subspace generated by $\phi_x(\mathbf{x})$ [Hsing and Eubank, 2015]. Restricting to linear predictors of the form $\mathcal{A}\,\phi_x(\mathbf{x})$ with $\mathcal{A} \in \mathcal{L}(\mathbb{H}, \mathbb{G})$, where $\mathcal{L}(\mathbb{H}, \mathbb{G})$ denotes the space of bounded

linear operators from $\mathbb{H}$ to $\mathbb{G}$, leads to the regularized least-squares problem with a regularization parameter $\lambda > 0$

$$\mathcal{A}^* := \underset{\mathcal{A}\in\mathcal{L}(\mathbb{H},\mathbb{G})}{\operatorname{argmin}} \mathbb{E}\left[\|\psi_y(\mathbf{y}) - \mathcal{A}\phi_x(\mathbf{x})\|^2_{\mathbb{G}}\right] + \lambda\|\mathcal{A}\|^2_{\mathrm{HS}},$$

where $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert-Schmidt norm. The corresponding normal equations yield the unique solution as $\mathcal{A}^* = \mathcal{C}_{\mathbf{yx}}(\mathcal{C}_{\mathbf{x}} + \lambda\mathcal{I})^{-1}$, where $\mathcal{I}$ is the identity operator. The operator $\mathcal{A}^*$ coincides with the population optimal linear predictor of $\psi_y(\mathbf{y})$ from $\phi_x(\mathbf{x})$ in the mean-squared sense [Grünewälder et al., 2012]. Motivated by this characterization and by the operator-theoretic treatment of conditional covariance based on $\mathcal{C}_{\mathbf{x}}$, $\mathcal{C}_{\mathbf{yx}}$ in [Fukumizu et al., 2004, Song et al., 2009], this leads to the following definition:

Definition 2 *We define $\mathcal{C}_{\mathbf{y}|\mathbf{x}} := \mathcal{C}_{\mathbf{yx}}(\mathcal{C}_{\mathbf{x}} + \lambda\mathcal{I})^{-1}$ and refer to $\mathcal{C}_{\mathbf{y}|\mathbf{x}} : \mathbb{H} \to \mathbb{G}$ as the (regularized) conditional covariance operator.*

This operator is the ridge-regularized population-optimal linear predictor of $\psi_y(\mathbf{y})$ from $\phi_x(\mathbf{x})$, and thus serves as a linear representation of conditional expectations in feature space. In practice, such conditional covariance operators are approximated on some finite-dimensional subspaces.

2.2 TWO-STAGE ESTIMATION WITH DEEP FEATURES

In the previous section, we introduced operators such as covariance and conditional covariance operators acting on feature spaces. To obtain learnable finite-dimensional representations of these operators, we choose neural feature maps that span finite subspaces and estimate the corresponding operators on these subspaces. For this estimation we adopt the two-stage deep feature procedure [Xu et al., 2021], in which two variables $\mathbf{x}$ and $\mathbf{z}$ are related through a nontrivial conditional distribution $P(\mathbf{x} \mid \mathbf{z})$, and estimation proceeds by first recovering the conditional expectation of features of $\mathbf{x}$ given $\mathbf{z}$, and then fitting a target function on the predicted features.

Let $\psi_{\theta_X} : \mathbb{X} \to \mathbb{R}^{d_1}$ and $\phi_{\theta_Z} : \mathbb{Z} \to \mathbb{R}^{d_2}$ denote neural feature maps, parameterized by θ_X and θ_Z, for a random variable $\mathbf{x}$ taking the value $\boldsymbol{x} \in \mathbb{X}$ and $\mathbf{z}$ taking the value $\boldsymbol{z} \in \mathbb{Z}$, and let $y \in \mathbb{R}$ be a scalar outcome. We model the structural function as $f(\boldsymbol{x}) := r_\xi\left(\psi_{\theta_X}(\boldsymbol{x})\right)$ using preimage map $r_\xi : \mathbb{R}^{d_1} \to \mathbb{R}$ parameterized by ξ. Then the objective is

$$\min_{\xi,\,\theta_X,\,\theta_Z} \mathbb{E}\left[\left(y - r_\xi\left(\mathbb{E}[\psi_{\theta_X}(\mathbf{x}) \mid \mathbf{z}]\right)\right)^2\right]. \tag{1}$$

This casts learning as estimating a conditional covariance operator and then fitting a preimage model on its prediction. In Xu et al. [2021], r_ξ is restricted to be linear, in which case the second stage admits a closed-form ridge solution.

The first stage estimates $\mathbb{E}[\psi_{\theta_X}(\mathbf{x}) \mid \mathbf{z}]$ in the span of ϕ_{θ_Z} by optimizing the linear map $\mathbf{V}$ and the feature parameters θ_Z, while keeping θ_X fixed:

$$\min_{\mathbf{V},\,\theta_Z} \frac{1}{N}\sum_{i=1}^{N} \|\psi_{\theta_X}(\boldsymbol{x}_i) - \mathbf{V}\phi_{\theta_Z}(\boldsymbol{z}_i)\|_2^2 + \lambda_1\|\mathbf{V}\|_F^2, \tag{2}$$

where $\lambda_1 > 0$ is the regularization term. For any fixed θ_Z, the minimizer in $\mathbf{V}$ has the closed form $\hat{\mathbf{V}}(\theta_X, \theta_Z) = \Psi\Phi^\top(\Phi\Phi^\top + \lambda_1\mathbf{I})^{-1}$; where $\Psi \in \mathbb{R}^{d_1\times N}$ and $\Phi \in \mathbb{R}^{d_2\times N}$ are the feature matrices whose i-th columns are $\psi_{\theta_X}(\boldsymbol{x}_i)$ and $\phi_{\theta_Z}(\boldsymbol{z}_i)$, respectively, and $\mathbf{I}$ is the identity matrix. The parameters θ_Z are then updated by gradient descent on the loss evaluated at $\mathbf{V} = \hat{\mathbf{V}}(\theta_X, \theta_Z)$, treating the closed-form solution as a differentiable function of ϕ_{θ_Z}. The linear map $\hat{\mathbf{V}}$ thus obtained serves as a finite-dimensional estimate of the conditional covariance operator on the subspaces spanned by the neural feature maps.

The second stage fits the structural function on the predicted conditional mean features. Given $\hat{\mathbf{V}}$ from Stage 1 with $\hat{\theta}_Z$ now fixed, we learn r_ξ and refine θ_X by minimizing:

$$\min_{\xi,\theta_X} \frac{1}{N}\sum_{i=1}^{N}\left(y_i - r_\xi\left(\hat{\mathbf{V}}\phi_{\theta_Z}(\boldsymbol{z}_i)\right)\right)^2 + \lambda_2\,\Omega(\xi), \tag{3}$$

where $\lambda_2 > 0$ is also the regularization term, and $\Omega(\cdot)$ denotes a suitable regularizer on ξ. The resulting predictor is obtained as $\hat{f}(\mathbf{x}) = r_{\hat{\xi}}(\psi_{\hat{\theta}_X}(\mathbf{x}))$. When the feature maps are fixed, Stage 1 is convex and admits a unique closed-form solution, and Stage 2 is a standard supervised learning problem. Furthermore, if the preimage map is restricted to be linear, Stage 2 similarly admits a closed-form solution [Xu et al., 2021].

Joint optimization of all parameters $(\xi, \theta_X, \theta_Z)$ with respect to the final prediction loss can lead to degenerate representations. As observed by Xu et al. [2021], without isolating the two stages, the model may bypass the conditional expectation structure, allowing ϕ_{θ_Z} and r_ξ to fit a direct mapping from $\mathbf{z}$ to y while rendering the intermediate features ψ_{θ_X} uninformative. To mitigate this issue, the training typically alternates between two stages. In Stage 1, θ_X is fixed, ensuring that $\hat{\mathbf{V}}$ properly estimates the conditional expectation $\mathbb{E}[\psi_{\theta_X}(\mathbf{x}) \mid \mathbf{z}]$ rather than propagating the outcome gradient directly to θ_Z. In Stage 2, θ_Z is fixed, and the prediction network is optimized on the projected features. This alternating scheme stabilizes the representation learning.

3 DEEP SPECTRAL LEARNING

3.1 STATE SPACE MODEL IN OPERATOR FORM

Section 2.1 introduced the mean embedding and the conditional covariance operator that provide linear representations of expectations and conditional expectations in Hilbert

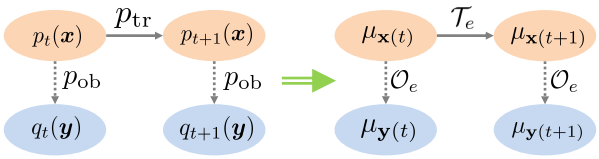

Figure 1: Overview of SSM in Operator Form

feature spaces. We apply this to a latent state space model (SSM) for a discrete-time stochastic dynamical system and relate a distribution-level description to an operator-level one.

Let $\{\mathbf{x}(t)\}_{t\in\mathbb{T}}$ be a latent stationary and ergodic Markov process on a state space $\mathbb{X}$, and $\{\mathbf{y}(t)\}_{t\in\mathbb{T}}$ be an observation process on an observation space $\mathbb{Y}$. The latent dynamics are characterized by a Markov transition kernel from $\mathbf{x}(t)$ to $\mathbf{x}(t+1)$. When this conditional law admits a density with respect to a reference measure on $\mathbb{X}$, we denote it by $p_{\mathrm{tr}}(\boldsymbol{x}_{t+1} \mid \mathbf{x}(t) = \boldsymbol{x}_t)$. The observation mechanism is characterized by the conditional law of $\mathbf{y}(t)$ given $\mathbf{x}(t)$. When this conditional law admits a density with respect to a reference measure on $\mathbb{Y}$, we denote it by $p_{\mathrm{ob}}(\boldsymbol{y}_t \mid \mathbf{x}(t) = \boldsymbol{x}_t)$. We do not assume that these densities are known, and we do not model them parametrically. They are introduced to describe how the latent dynamics and the observation channel induce linear maps on probability laws.

At the level of probability densities, p_{tr} induces an evolution of state distributions. If p_t denotes the density of $\mathbf{x}(t)$, then the density at the next step satisfies

$$p_{t+1}(\boldsymbol{x}) \;=\; \int_{\mathbb{X}} p_{\mathrm{tr}}(\boldsymbol{x} \mid \mathbf{x}(t) = \boldsymbol{z})\, p_t(\boldsymbol{z})\, d\boldsymbol{z}. \tag{4}$$

Similarly, p_{ob} induces a map from state distributions to observation ones. If q_t denotes the density of $\mathbf{y}(t)$, then

$$q_t(\boldsymbol{y}) \;=\; \int_{\mathbb{X}} p_{\mathrm{ob}}(\boldsymbol{y} \mid \mathbf{x}(t) = \boldsymbol{x})\, p_t(\boldsymbol{x})\, d\boldsymbol{x}. \tag{5}$$

The left side of Figure 1 summarizes these distribution-level transitions through p_{tr} and p_{ob}.

On the other hand, we now express the same model at the level of embedded distributions. We represent the marginal laws of $\mathbf{x}(t)$ and $\mathbf{y}(t)$ by their mean embeddings $\mu_{\mathbf{x}(t)} \in \mathbb{H}$ and $\mu_{\mathbf{y}(t)} \in \mathbb{G}$. The conditional covariance operators provide linear maps that propagate these embedded laws. In particular, we denote by $\mathcal{T}_e$ and $\mathcal{O}_e$ the operators associated with the one-step latent transition and the observation channel, respectively:

$$\mathcal{T}_e := \mathcal{C}_{\mathbf{x}(t+1)|\mathbf{x}(t)}, \quad \mathcal{O}_e := \mathcal{C}_{\mathbf{y}(t)|\mathbf{x}(t)}. \tag{6}$$

This yields an operator-level state-space representation in which the mean embeddings evolve linearly, as $\mu_{\mathbf{x}(t+1)} = \mathcal{T}_e\, \mu_{\mathbf{x}(t)}$, and $\mu_{\mathbf{y}(t)} = \mathcal{O}_e\, \mu_{\mathbf{x}(t)}$. The right side of Figure 1 depicts this operator form and its correspondence to the distribution-level description. In [Ke et al., 2025], $\mathcal{T}_e$ and $\mathcal{O}_e$ are termed the embedded latent transfer operator (ELTO) and embedded observation operator, respectively, and a kernel-based instance of this formulation is proposed. The DSL construction below keeps this operator-level state-space view, but realizes the predictive state coordinates and operator regressions in explicit finite-dimensional neural Galerkin spaces rather than through RKHS kernel evaluations over samples.

3.2 STOCHASTIC REALIZATION WITH DEEP FEATURES

We adopt the framework of stochastic realization to construct latent state variables directly from observed data. Classical stochastic realization methods, such as balanced stochastic realization based on CCA between past and future observations [Akaike, 1975, Desai et al., 1985], estimate a state-space representation by identifying a subspace that yields optimal least-squares predictions. We build on this idea and generalize it to Hilbert spaces and deep features by employing functional CCA for our setting.

Assume the observation process is zero-mean and square-integrable. Fix a Hilbert space $\mathbb{F}$ and a feature map $\phi : \mathbb{R}^d \to \mathbb{F}$. Let $\mathbb{T}$ denote the set of time indices of a trajectory. For each $t \in \mathbb{T}$, we define the feature process $u_t = \phi(\boldsymbol{y}_t) \in \mathbb{F}$, which we regard as an $\mathbb{F}$-valued random variable. We view $\{u_t\}_{t\in\mathbb{T}}$ as elements of the Hilbert space $L^2(\,\Omega, \mathbb{F}\,)$ of square-integrable $\mathbb{F}$-valued random variables. For any $u,\, v \in L^2(\Omega;\mathbb{F})$, the inner product is defined by $\langle u, v\rangle_{L^2(\Omega;\mathbb{F})} = \mathbb{E}\big[\langle u, v\rangle_{\mathbb{F}}\big]$, where $\langle\cdot,\cdot\rangle_{\mathbb{F}}$ denotes the inner product in $\mathbb{F}$. We then define the past and future information as the closed linear spans

$$\mathbb{H}^- := \overline{\mathrm{span}}\{u_{t-1}, u_{t-2}, \dots\} \subset L^2(\Omega;\mathbb{F}), \tag{7}$$

$$\mathbb{H}^+ := \overline{\mathrm{span}}\{u_t, u_{t+1}, \dots\} \subset L^2(\Omega;\mathbb{F}). \tag{8}$$

For arbitrary linear combinations $v^- = \sum_{k=1}^{K} \alpha_k u_{t-k}$ and $w^- = \sum_{\ell=1}^{L} \gamma_\ell u_{t-\ell} \in \mathbb{H}^-$, the inner product in $L^2(\Omega;\mathbb{F})$ can be expressed, under weak stationarity, in terms of the lag-covariance function $\Lambda(r) := \mathbb{E}\big[\langle u_t, u_{t-r}\rangle_{\mathbb{F}}\big]$ for any time $t \in \mathbb{T}$, as

$$\langle v^-, w^-\rangle_{L^2(\Omega;\mathbb{F})} = \sum_{k=1}^{K}\sum_{\ell=1}^{L} \alpha_k \gamma_\ell\, \Lambda(\ell - k). \tag{9}$$

An analogous expression holds for $\mathbb{H}^+$ and for past–future cross terms.

For the feature process $u_t \in \mathbb{F}$, define the lag-r covariance operator ($r \in \mathbb{Z}$) as

$$\mathcal{C}_r : \mathbb{F} \to \mathbb{F}; \quad \mathcal{C}_r\, x := \mathbb{E}[\langle u_t, x\rangle_{\mathbb{F}}\, u_{t+r}], \tag{10}$$

where $x \in \mathbb{F}$, and $\mathcal{C}_r$ is the unique bounded linear operator satisfying $\langle \mathcal{C}_r\, x,\, h\rangle_{\mathbb{F}} = \mathbb{E}\,[\langle u_t,\, x\rangle_{\mathbb{F}} \langle u_{t+r},\, h\rangle_{\mathbb{F}}]$ for all $h \in$

$\mathbb{F}$. Equivalently, $\mathcal{C}_r$ is denoted as $\mathbb{E}[u_{t+r} \otimes u_t]$. We can define the following vectors as the past and future information at t:

$$\boldsymbol{p}_t = [u_{t-1},\, u_{t-2}, \dots]^\top,\ \boldsymbol{f}_t = [u_t,\, u_{t+1}, \dots]^\top. \quad (11)$$

Then the covariance for past process $\mathbb{E}[\boldsymbol{p}_t \otimes \boldsymbol{p}_t]$ and for future process $\mathbb{E}[\boldsymbol{f}_t \otimes \boldsymbol{f}_t]$ are considered as block Toeplitz operators; the $(i,\, j)$-block of past is $\mathcal{C}_{j-i}$, and that of future is $\mathcal{C}_{i-j}$. Also, the covariance between future and past $\mathbb{E}[\boldsymbol{f}_t \otimes \boldsymbol{p}_t]$ are considered as the block Hankel operator; the $(i,\, j)$-block is $\mathcal{C}_{i+j-1}$.

The orthogonal projection of $\mathbb{H}^+$ onto $\mathbb{H}^-$ defines the projected space $\mathbb{X}_t := \mathcal{P}(\mathbb{H}^+ \mid \mathbb{H}^-)$, consisting of all minimum-variance linear predictions of future features based on the past. Because the block Toeplitz covariance operators admit Cholesky factorizations [Chui et al., 1982], one can whiten the past and future coordinates, and the projection of $\mathbb{H}^+$ onto $\mathbb{H}^-$ in the whitened basis factors through the block Hankel operator. When the block Hankel operator has finite rank r, the image of this projection is therefore r-dimensional, and $\mathbb{X}_t$ is the minimal splitting subspace between $\mathbb{H}^-$ and $\mathbb{H}^+$: any subspace between $\mathbb{D} \subset \mathbb{H}^-$ satisfying $\mathcal{P}(\mathbb{H}^+ \mid \mathbb{D}) = \mathcal{P}(\mathbb{H}^+ \mid \mathbb{H}^-)$ contains $\mathbb{X}_t$. In our setting, the covariance operators on $\mathbb{H}^-$ and $\mathbb{H}^+$ inherit block Toeplitz and Hankel structure from weak stationarity of the feature process u_t as established above, so the results of [Lindquist and Picci, 1991, Katayama, 2005] apply directly. A basis for $\mathbb{X}_t$ therefore provides the minimal information from the past needed to predict the future optimally in the least-squares sense, and can serve as latent state coordinates for a Markov model of u_t.

To find such a basis we employ CCA between $\mathbb{H}^-$ and $\mathbb{H}^+$ [Desai et al., 1985]. That is, we consider the following functional CCA:

$$\max_{v^- \in \mathbb{H}^- \setminus \{0\},\, v^+ \in \mathbb{H}^+ \setminus \{0\}} \frac{\mathrm{Cov}(v^-, v^+)}{\sqrt{\mathrm{Var}(v^-)}\,\sqrt{\mathrm{Var}(v^+)}}, \quad (12)$$

where for $v^- = \sum_{i \geq 1} \alpha_i u_{t-i}$ and $v^+ = \sum_{j \geq 0} \beta_j u_{t+j}$, $\mathrm{Cov}(v^-, v^+) = \sum_{i,j}\, \alpha_i \beta_j \mathbb{E}[\langle u_{t-i},\, u_{t+j} \rangle_{\mathbb{F}}]$, $\mathrm{Var}(v^-) = \mathrm{Cov}(v^-, v^-)$ and $\mathrm{Var}(v^+) = \mathrm{Cov}(v^+, v^+)$. Under weak stationarity, these reduce to Hankel and Toeplitz block operators via $\Lambda(r) = \mathbb{E}[\langle u_t, u_{t-r} \rangle_{\mathbb{F}}]$ for all time t. The top r canonical directions from this CCA provide an orthonormal basis for the prediction space $\mathbb{X}_t$ [Eubank and Hsing, 2008, Katayama, 2005]. Specifically, if $\sum_{i \geq 1} \alpha_{l,i}\, u_{t-i}$ denotes the l-th canonical direction for $\mathbb{H}^-$ with canonical correlation s_l, then the coordinates $x_l(t) := s_l^{1/2} \sum_{i \geq 1} \alpha_{l,i}\, u_{t-i}$ $(l = 1, \dots, r)$ form an orthonormal basis of $\mathbb{X}_t$. Collecting them into $\mathbf{x}(t) := [x_1(t), \dots, x_r(t)]^\top \in \mathbb{R}^r$, one can characterize the one-step evolution of these state coordinates: since the block Hankel operator in our setting admits the same rank-r factorization and block-row-shift property, the construction in Katayama [2005], Ke et al. [2025] determines a unique transition matrix $\mathbf{A} \in \mathbb{R}^{r \times r}$ on the state coordinates as

$$\mathbf{x}(t+1) = \mathbf{A}\, \mathbf{x}(t) + \mathbf{w}(t), \quad (13)$$

where $\mathbf{w}(t) := \mathbf{x}(t+1) - \mathcal{P}(\mathbf{x}(t+1) \mid \mathbf{x}(t))$ is the innovation noise satisfying $\mathbf{w}(t) \perp \mathbb{H}^-$. This Markov representation gives the optimal minimum-variance evolution of the latent state along time.

In practice, the infinite past and future are truncated to a finite window of length ℓ, and the population functional CCA is approximated in finite-dimensional Galerkin subspaces. Given a finite number of samples, we define the past and future delay vectors as $\tilde{\boldsymbol{p}}_t := (\boldsymbol{y}_{t-1}^\top, \dots, \boldsymbol{y}_{t-\ell}^\top)^\top$ and $\tilde{\boldsymbol{f}}_t := (\boldsymbol{y}_t^\top, \dots, \boldsymbol{y}_{t+\ell-1}^\top)^\top$, and denote by $I_{\mathrm{SR}} \subset \mathbb{T}$ the time index set for which both windows are well-defined. By applying a block feature map $\tilde{\phi}$ and centering the outputs, we arrange them into feature matrices Φ_p and Φ_f over $t \in I_{\mathrm{SR}}$.

Weak stationarity is used to express the past and future covariance blocks through lag-covariances, giving the Toeplitz and Hankel structures used in the CCA realization. The finite-rank condition gives the exact case in which the predictive state space is r-dimensional; in finite samples, retaining the top r CCA directions corresponds to an approximate low-rank past–future predictive structure. The function-space assumptions are implemented through the learned block-feature space and the dictionary spaces used for subsequent operator regression.With the corresponding sample covariance blocks $\mathbf{C}_-$ and $\mathbf{C}_+$, as well as the cross-covariance $\mathbf{H}$, the CCA problem reduces to a singular value decomposition of the normalized cross-covariance block $\mathbf{T} := \mathbf{C}_+^{-1/2}\, \mathbf{H}\, \mathbf{C}_-^{-1/2}$. The singular values and vectors of $\mathbf{T}$ provide finite-dimensional estimates of the canonical correlations and directions, which are then used to extract the state coordinates $\mathbf{x}(t)$. Further details of the data-driven algorithm are provided in Appendix A.1.

4 MODEL ARCHITECTURE AND LEARNING ALGORITHM

4.1 DESIGN PRINCIPLES OF DEEP SPECTRAL ENCODER

In this section, we instantiate the deep spectral learning framework developed in Section 3 in finite-dimensional deep feature spaces. We introduce the Deep Spectral Encoder, an architecture specifically designed to strictly separate representation learning from dynamics modeling. Jointly optimizing all neural components and operators often leads to degenerate solutions where the encoder absorbs temporal structures, rendering the learned operators trivial. To prevent this, our model adopts a modular design. It sequentially integrates a time-invariant encoder for point-wise feature extraction, a stochastic realization module for state coordinate construction, a closed-form operator estimation

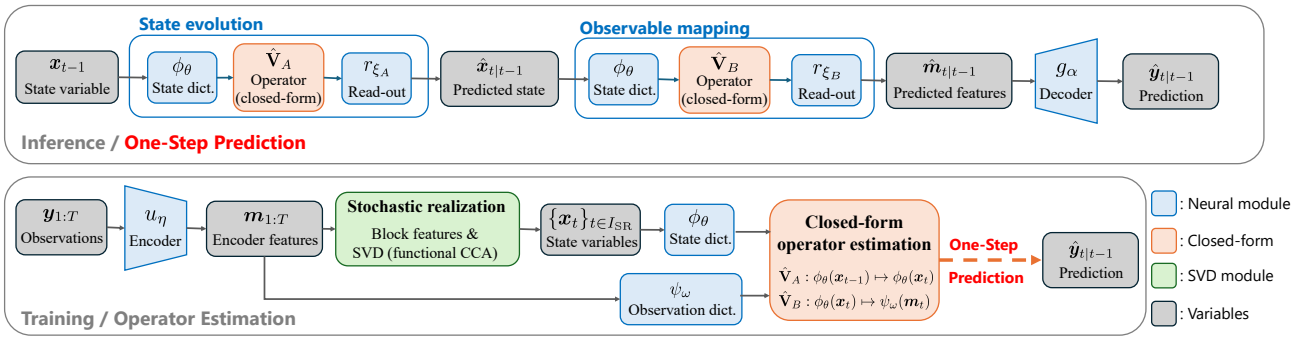

Figure 2: Overall Architecture of Deep Spectral Encoder

step on the feature space, and a time-invariant decoder for observation reconstruction. Figure 2 summarizes the computational graph of this architecture. The detailed empirical algorithms are provided in Appendix A.

Deep Feature Extraction and State Realization: To extract predictive latent states from the observation sequence, we first map each observation $\boldsymbol{y}_t$ to a feature representation $\boldsymbol{m}_t = u_\eta(\boldsymbol{y}_t)$ via a time-invariant neural encoder $u_\eta : \mathbb{R}^d \to \mathbb{R}^m$. To construct the past and future block features required for functional CCA without feeding excessively large windows into a single network, we apply a shallow head network $h : \mathbb{R}^\ell \to \mathbb{R}$ to the delay vectors of each feature dimension. Specifically, for the k-th dimension of the encoder output, we form the past delay vector $\tilde{\boldsymbol{v}}_{k,t} = (u_\eta^{(k)}(\boldsymbol{y}_{t-1}), \ldots, u_\eta^{(k)}(\boldsymbol{y}_{t-\ell}))^\top$ and obtain the scalar block feature $\tilde{\phi}^{(k)}(\tilde{\boldsymbol{p}}_t) := h(\tilde{\boldsymbol{v}}_{k,t})$. Stacking these scalars yields the past feature $\tilde{\phi}(\tilde{\boldsymbol{p}}_t) \in \mathbb{R}^m$, and the future feature $\tilde{\phi}(\tilde{\boldsymbol{f}}_t)$ is constructed analogously.

Given these block features for $t \in I_{\mathrm{SR}}$, we directly apply the stochastic realization procedure described in Section 3.2. By computing the SVD of the normalized cross-covariance $\mathbf{T}$, we obtain the top r canonical directions. These directions yield the r-dimensional latent state coordinates $\boldsymbol{x}_t \in \mathbb{R}^r$, providing a finite-dimensional Markovian representation of the sequence.

Closed-Form Operator Estimation: On the estimated latent state space, we introduce a neural dictionary $\phi_\theta : \mathbb{R}^r \to \mathbb{R}^{d_A}$, and on the observation feature space, another dictionary $\psi_\omega : \mathbb{R}^m \to \mathbb{R}^{d_B}$. These dictionaries induce finite-dimensional subspaces on which we approximate the transfer operator by $\mathbf{V}_A : \mathbb{R}^{d_A} \to \mathbb{R}^{d_A}$ and the observation operator by $\mathbf{V}_B : \mathbb{R}^{d_A} \to \mathbb{R}^{d_B}$.

Let $\Phi_- = [\phi_\theta(\boldsymbol{x}_{t-1})]_{t\in I_{\mathrm{SR}}}$, $\Phi_+ = [\phi_\theta(\boldsymbol{x}_t)]_{t\in I_{\mathrm{SR}}}$, and $\Psi = [\psi_\omega(\boldsymbol{m}_t)]_{t\in I_{\mathrm{SR}}}$ be the feature matrices. Following the deep-feature two-stage procedure, we estimate the operators as ridge-regularized closed-form solutions:

$$\hat{\mathbf{V}}_A = \Phi_+\Phi_-^\top(\Phi_-\Phi_-^\top + \lambda_A\mathbf{I})^{-1}, \tag{14}$$

$$\hat{\mathbf{V}}_B = \Psi\widehat{\Phi}_+^\top(\widehat{\Phi}_+\widehat{\Phi}_+^\top + \lambda_B\mathbf{I})^{-1}, \tag{15}$$

where $\widehat{\Phi}_+ := [\phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})]_{t\in I_{\mathrm{SR}}}$ is the matrix of one-step predicted state features computed via the state read-out map $\hat{\boldsymbol{x}}_{t|t-1} := r_{\xi_A}(\hat{\mathbf{V}}_A\phi_\theta(\boldsymbol{x}_{t-1}))$. These closed-form solutions coincide with the Galerkin projections of the underlying conditional covariance operators $\mathcal{T}_e$ and $\mathcal{O}_e$, enabling stable estimation without constructing sample-dependent Gram matrices. The read-out maps r_{ξ_A} and r_{ξ_B} are then learned by minimizing the regularized reconstruction errors on the operator-projected features $\hat{\mathbf{V}}_A\Phi_-$ and $\hat{\mathbf{V}}_B\widehat{\Phi}_+$. This step corresponds to the second stage of the deep feature procedure, taking the form of the objective in Eq. (3).

4.2 STAGED TRAINING STRATEGY

To successfully train the architecture and enforce the modular design described in Section 4.1, we employ a staged training schedule. At the outset, we freeze the parameters of the encoder and the decoder, and we optimize the dictionaries and read-out maps. The computational cost and the differentiable CCA/SVD implementation are summarized in Appendix A.3. This step alternates between closed-form updates of the operators $\hat{\mathbf{V}}_A$ and $\hat{\mathbf{V}}_B$ and gradient-based updates of the neural parameters for the dictionaries and read-out maps.

Once the operators and read-outs have stabilized, we unfreeze the encoder and decoder to perform end-to-end refinement along the test-time causal path. We compute the one-step-ahead prediction $\hat{\boldsymbol{y}}_{t|t-1} = g_\alpha(r_{\xi_B}(\hat{\mathbf{V}}_B\phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})))$ and optimize the stepwise reconstruction loss $\sum_{t\in I_{\mathrm{SR}}} \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}\|^2$, augmented with a regularization term that maximizes the sum of the canonical correlations obtained from the stochastic realization. This staged schedule separates representation learning from operator estimation and mitigates degenerate solutions observed in joint end-to-end training.

5 SEQUENTIAL STATE ESTIMATION

Sequential state estimation, often formulated as sequential Bayesian filtering, propagates information about a latent state in a dynamical system from sequentially observed data. Classical Kalman filtering realizes this recursion through posterior mean and covariance updates, alternating between a prediction step that propagates the belief state in time and an innovation step that incorporates new observations [Kalman, 1960, Anderson and Moore, 1979]. The idea of expressing Kalman-type updates at the level of operators on feature spaces has been explored in the kernel Kalman rule proposed by [Fukumizu et al., 2013, Gebhardt et al., 2019]. In this section we follow this operator-theoretic viewpoint in a general Hilbert-space setting and then describe its finite-dimensional implementation in the feature space learned by DSE.

At time t, denote the mean embedding and covariance operator for the prior by $\mu^-_{\mathbf{x}(t)}$ and $\mathcal{C}^-_{\mathbf{x}(t)}$, respectively. Let $\mathcal{R} \in \mathcal{L}(\mathbb{G})$ and $\mathcal{Q} \in \mathcal{L}(\mathbb{H})$ be self-adjoint, positive semi-definite, bounded noise covariance operators at time t.

Then the innovation Kalman update in Hilbert space is

$$\mu^+_{\mathbf{x}(t)} = \mu^-_{\mathbf{x}(t)} + \mathcal{K}_t \left(\psi_y(\boldsymbol{y}_t) - \mathcal{C}_{\mathbf{y}(t)|\mathbf{x}(t)}\mu^-_{\mathbf{x}(t)}\right), \quad (16)$$

$$\mathcal{C}^+_{\mathbf{x}(t)} = \mathcal{C}^-_{\mathbf{x}(t)} - \mathcal{K}_t\, \mathcal{C}_{\mathbf{y}(t)|\mathbf{x}(t)}\, \mathcal{C}^-_{\mathbf{x}(t)}, \quad (17)$$

where $\mathcal{K}_t = \mathcal{C}^-_{\mathbf{x}(t)}\, \mathcal{C}^*_{\mathbf{y}(t)|\mathbf{x}(t)} \left(\mathcal{C}_{\mathbf{y}(t)|\mathbf{x}(t)}\mathcal{C}^-_{\mathbf{x}(t)}\mathcal{C}^*_{\mathbf{y}(t)|\mathbf{x}(t)} + \mathcal{R}\right)^{-1}$, and $\psi_y(\boldsymbol{y}_t) \in \mathbb{G}$ is the feature map of the observation. These are the operator-level Kalman updates.

For the time update of state process, we employ the conditional covariance operator for state process $\mathcal{C}_{\mathbf{x}(t+1)|\mathbf{x}(t)}$, and its Kalman update formulated as

$$\mu^-_{\mathbf{x}(t+1)} = \mathcal{C}_{\mathbf{x}(t+1)|\mathbf{x}(t)}\mu^+_{\mathbf{x}(t)}, \quad (18)$$

$$\mathcal{C}^-_{\mathbf{x}(t+1)} = \mathcal{C}_{\mathbf{x}(t+1)|\mathbf{x}(t)}\mathcal{C}^+_{\mathbf{x}(t)}\mathcal{C}^*_{\mathbf{x}(t+1)|\mathbf{x}(t)} + \mathcal{Q}. \quad (19)$$

Applying these updates sequentially yields an operator-level forward filter for the latent state.

In practice, the operators $\mathcal{C}_{\mathbf{x}(t+1)|\mathbf{x}(t)}$ and $\mathcal{C}_{\mathbf{y}(t)|\mathbf{x}(t)}$ are represented by the matrices $\hat{\mathbf{V}}_A$ and $\hat{\mathbf{V}}_B$ estimated in Section 4.1. The process noise covariance $\mathcal{Q}$ and observation noise covariance $\mathcal{R}$ are estimated from the empirical residuals of these operators on the training trajectory. Then the filtering recursion is performed entirely in the feature space by propagating and updating the moments through the learned operators without explicit Gram matrix constructions. This feature-space recursion corresponds to the operator-level Kalman-type update proposed in Gebhardt et al. [2019]. The detailed estimation of noise covariances and the recursive filtering algorithm are provided in Appendix A.4. Finally, the latent state is recovered via the read-out map as $\hat{\boldsymbol{x}}_t = r_{\hat{\xi}_A}(\boldsymbol{\mu}^+_t)$.

Table 1: Quad-Link Pendulum Images

Method	length	Without noise	With noise
RKN	1.5k	0.2198 ± 0.0132	0.2944 ± 0.0236
LSTM	1.5k	0.2527 ± 0.0152	0.3167 ± 0.0223
ELTO-KF	1.5k	0.2175 ± 0.0130	0.2942 ± 0.0215
DSE (ours)	1.5k	$\mathbf{0.2006} \pm 0.0228$	$\mathbf{0.2593} \pm 0.0274$
RKN	15k	0.2838 ± 0.0190	0.3258 ± 0.0241
LSTM	15k	0.3050 ± 0.0183	0.3158 ± 0.0244
ELTO-KF	15k	0.2924 ± 0.0175	0.2936 ± 0.0207
DSE (ours)	15k	$\mathbf{0.2615} \pm 0.0211$	$\mathbf{0.2683} \pm 0.0254$

Numerical Results: We evaluate on quad-link pendulum image sequences of size 48×48 under a clean regime and a corrupted regime with occlusion and geometric distortion. Initial joint angles are sampled uniformly from $[-\pi, \pi]$ with zero initial velocities, viscous friction is 0.1, the simulation time step is 10^{-4} seconds, frames are recorded every 0.05 seconds, and the first five frames in each sequence are kept clean. One-step prediction errors are reported as mean squared error for training–test sequence lengths of 1.5×10^3 and 1.5×10^4. For the one-step comparison in Table 1, baselines include a Recurrent Kalman Network (RKN) [Becker et al., 2019] and an LSTM predictor, as well as ELTO-Kalman filtering (ELTO-KF) [Ke et al., 2025]. We keep the encoder resolution, data splits, preprocessing, simulator and noise parameters fixed across methods, including the corrupted setting with $\rho = 0.2$ and thresholds $[0, 0.25, 0.75, 1.0]$.

Table 1 reports the mean and standard deviation of the one-step prediction MSE for all methods under the clean and corrupted regimes. Across all training lengths and noise settings, DSE attains the lowest average error and consistently improves upon the RKN and the LSTM, while also outperforming the ELTO-based predictor.

Results on Multi-Step Prediction: To evaluate rollout stability beyond one-step prediction, we further consider multi-step prediction on the 1.5×10^3 corrupted quad-link benchmark. At each time t, each method is first conditioned on the observations up to t using its own state-update mechanism and then predicts $\hat{\boldsymbol{y}}_{t+H|t}$ for $H \in \{1, 5, 10, 20, 50\}$ without using observations after t. We also include Neural Continuous-Discrete State Space Model (NCDSSM) [Ansari et al., 2023] as an additional neural latent state-space baseline, adapting its image encoder and decoder to the 48×48 grayscale observations. Table 2 reports the resulting prediction MSEs. In this setting, DSE attains the lowest mean MSE across the evaluated horizons, and the multi-step results complement the one-step comparison by measuring how prediction errors compound after the filtering update is no longer conditioned on new observations.

Table 2: Multi-step prediction on Quad-Link benchmark

Method	$H = 1$	$H = 5$	$H = 10$	$H = 20$	$H = 50$
ELTO-KF	0.2942 ± 0.0215	0.3104 ± 0.0256	0.3309 ± 0.0304	0.3068 ± 0.0403	0.3953 ± 0.0503
RKN	0.2944 ± 0.0236	0.3209 ± 0.0308	0.3452 ± 0.0408	0.3758 ± 0.0501	0.4057 ± 0.0607
LSTM	0.3167 ± 0.0233	0.3557 ± 0.0350	0.3852 ± 0.0451	0.4109 ± 0.0554	0.4307 ± 0.0658
NCDSSM	0.2804 ± 0.0358	0.3005 ± 0.0406	0.3259 ± 0.0457	0.3607 ± 0.0555	0.4004 ± 0.0702
DSE (ours)	$\mathbf{0.2593} \pm 0.0274$	$\mathbf{0.2754} \pm 0.0308$	$\mathbf{0.2952} \pm 0.0357$	$\mathbf{0.3256} \pm 0.0456$	$\mathbf{0.3805} \pm 0.0551$

6 MODE DECOMPOSITION

Operator-theoretic analysis describes nonlinear dynamics via linear operators acting on observables, which enables spectral decompositions and mode interpretations [Brunton et al., 2022]. Let $\mathcal{K}$ denote the Koopman operator and let $\{(\lambda_i, \varphi_i)\}_{i\geq 1}$ be its eigenpairs. For an observable $f : \mathbb{X} \to \mathbb{C}^d$ that lies in the span of these eigenfunctions, the t-step conditional expectation admits the expansion

$$(\mathcal{K}^t f)(\boldsymbol{x}) = \sum_i \lambda_i^t \, \varphi_i(\boldsymbol{x}) \, \boldsymbol{v}_i, \tag{20}$$

where $\boldsymbol{v}_i$ are Koopman modes and each mode evolves with a single complex rate. This Koopman mode decomposition separates oscillatory frequencies from growth or decay, and motivates finite-dimensional approximations such as dynamic mode decomposition (DMD) [Schmid, 2010].

We analyze the learned dynamics through the Koopman operator associated with the latent Markov process in Section 3.1. For an observable function $f : \mathbb{X} \to \mathbb{C}^d$, the Koopman operator $\mathcal{K}$ maps f to its conditional expectation under the transition density p_{tr}, defined as [Klus et al., 2019]:

$$(\mathcal{K}f)(\boldsymbol{x}_t) = \int_{\mathbb{X}} p_{\text{tr}}(\boldsymbol{x}_{t+1} \mid \mathbf{x}(t) = \boldsymbol{x}_t) \, f(\boldsymbol{x}_{t+1}) \, d\boldsymbol{x}_{t+1} \tag{21}$$

$$= \mathbb{E}[f(\mathbf{x}(t+1)) \mid \mathbf{x}(t) = \boldsymbol{x}_t] \, . \tag{22}$$

For the observation model p_{ob} defined in Section 3.1, we consider the conditional mean observation function f_{ob} given by

$$f_{\text{ob}}(\boldsymbol{x}_t) = \int_{\mathbb{Y}} p_{\text{ob}}(\boldsymbol{y}_t \mid \mathbf{x}(t) = \boldsymbol{x}_t) \, \boldsymbol{y}_t \, d\boldsymbol{y}_t, \tag{23}$$

so that $(\mathcal{K} f_{\text{ob}})(\boldsymbol{x}_t) = \mathbb{E}[\mathbf{y}(t+1) \mid \mathbf{x}(t) = \boldsymbol{x}_t]$.

We define observables in the Hilbert space $\mathbb{H}$ and write $\mathcal{K}_{\mathbb{H}}$ for the Koopman operator projected onto $\mathbb{H}$. Under the assumption that $\mathbb{E}[f_{\text{ob}}(\mathbf{x}(t+1)) \mid \mathbf{x}(t) = \cdot] \in \mathbb{H}$, the projected operator satisfies the identity [Fukumizu et al., 2004, Ke et al., 2025]:

$$\mathcal{C}_{\mathbf{x}(t)} \, (\mathcal{K}_{\mathbb{H}} f_{\text{ob}}) = \mathcal{C}_{\mathbf{x}(t)\mathbf{x}(t+1)} \, f_{\text{ob}}. \tag{24}$$

Since the inverse of $\mathcal{C}_{\mathbf{x}(t)}$ may be ill-posed, we work with the ridge-regularized projected Koopman operator

$$\mathcal{K}_{\mathbb{H},\lambda} = (\mathcal{C}_{\mathbf{x}(t)} + \lambda \mathcal{I})^{-1} \mathcal{C}_{\mathbf{x}(t)\mathbf{x}(t+1)},$$

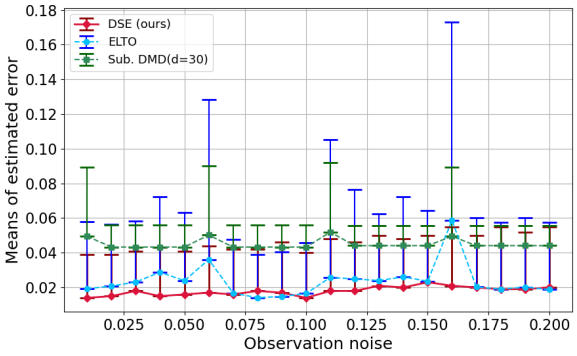

Figure 3: Results for VDP Oscillator

which can take the adjoint and yield

$$\mathcal{K}^*_{\mathbb{H},\lambda} = \mathcal{C}_{\mathbf{x}(t+1)\mathbf{x}(t)} \, (\, \mathcal{C}_{\mathbf{x}(t)} + \lambda \mathcal{I} \,)^{-1} = \mathcal{T}_e, \tag{25}$$

where $\mathcal{T}_e$ is the conditional covariance operator defined in Section 3.1 that propagates embedded state laws forward in time. As discussed in Section 4, $\mathbf{V}_A$ is the matrix representation of the Galerkin projection of $\mathcal{T}_e$ onto the subspace spanned by the learned feature dictionaries and is estimated by the closed-form normal equations in Eq. (14). This adjoint relation connects Koopman spectral analysis to the learned transfer operator.

To compute the spectrum and modes, we perform eigendecomposition induced by learned $\hat{\mathbf{V}}_A$. When the sampling interval Δt is known, we convert to continuous-time eigenvalues via $\mu_i = \Delta t^{-1} \log \lambda_i$, so that the real part of μ_i gives the decay rate and imaginary part gives the angular frequency. In the numerical experiments below, we quantify spectrum recovery by comparing the estimated eigenvalues to a reference spectrum computed from the ground-truth dynamics, and we compare the results with several baseline methods described later.

Numerical Results: To evaluate the robustness of DSE in recovering Koopman spectra under noise, we consider two standard nonlinear oscillators: the Van der Pol (VDP) oscillator and Stuart-Landau (SL) oscillator with observation and process noises.

We first examine the VDP oscillator against zero-mean Gaussian observation noise. By varying the noise variance σ^2_{obs}

Table 3: Results at Representative Noise Levels (VDP)

noise	DSE (ours)	ELTO	sDMD	hDMD
0.05	0.016±0.025	0.024±0.039	0.043±0.013	0.076±0.042
0.10	0.014±0.026	0.016±0.029	0.043±0.013	0.109±0.072
0.15	0.023±0.028	0.024±0.041	0.044±0.012	0.129±0.082
0.20	0.020±0.035	0.019±0.039	0.044±0.012	0.143±0.091
Avg.	**0.019**±0.029	0.024±0.046	0.045±0.020	0.104±0.074

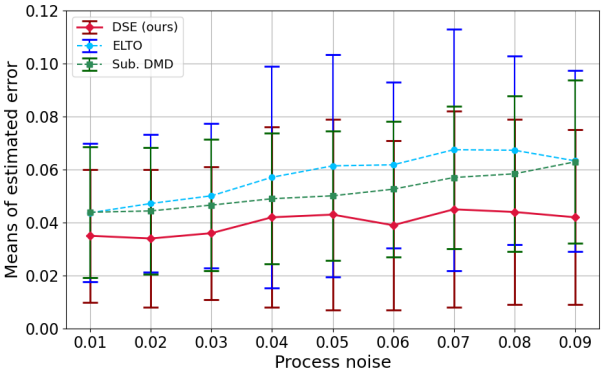

Figure 4: Results for SL Oscillator

between 0.01 and 0.20, we generate independent noisy trajectories and compare the estimated eigenvalues with a reference spectrum derived from the noise-free limit cycle. The estimation error is defined as the absolute difference between the estimated and reference eigenvalues averaged over multiple trials. Next, we evaluate the SL oscillator to assess robustness against dynamic process noise. Following the same evaluation protocol, we fix the observation noise variance at 0.01 and vary the process noise variance ϵ_{proc} between 0.01 and 0.09. For this system, the estimated eigenvalues are compared with an analytical reference spectrum derived from the asymptotic theory under weak process noise [Bagheri, 2014]. The explicit governing equations, numerical integration schemes, and procedures for computing the reference spectra for both systems are detailed in Appendix C.2.1 and C.2.2, while their data generation hyperparameters are summarized in Tables 10 and 12.

Baselines: DSE is compared with several DMD-based approaches: extended DMD (eDMD) [Williams et al., 2015], Hankel DMD (hDMD) [Arbabi and Mezić, 2017], subspace DMD (sDMD) [Takeishi and Kawahara, 2021], and ELTO [Ke et al., 2025], whose learned transfer operator spectrum serves as the Koopman spectral estimator.

For VDP, hDMD and sDMD are applied directly to noisy scalar observations with length-30 Hankel delay coordinates. ELTO and DSE first infer a latent state trajectory via the realization step, then estimate the operator, and report the eigenvalues for the Koopman spectrum. For SL, ELTO and DSE again infer latent states from the observation process.

Table 4: Results at Representative Noise Levels (SL)

noise	DSE (ours)	ELTO	sDMD	eDMD
0.01	0.035±0.025	0.044±0.026	0.044±0.025	0.325±0.129
0.05	0.043±0.036	0.061±0.042	0.050±0.024	0.318±0.132
0.09	0.042±0.033	0.063±0.034	0.063±0.031	0.323±0.126
Avg.	**0.041**±0.032	0.058±0.034	0.052±0.026	0.321±0.130

eDMD and sDMD lift observations using complex exponentials based on the polar Fourier feature map.

Figures 3 and 4 plot mean eigenvalue estimation errors with one-standard-deviation bars, and Tables 3 and 4 report the same metric at representative noise levels; methods with substantially larger errors are omitted for clarity. On VDP, DSE achieves the lowest mean error across noise levels; ELTO is the closest baseline, matching DSE at several levels, but DSE exhibits smaller standard deviations. The DMD baselines incur larger errors, with hDMD degrading notably as noise increases. On SL, DSE again achieves the lowest mean error, with sDMD and ELTO close behind, and remains more stable than ELTO.

7 CONCLUSION

This work developed a deep spectral learning framework for stochastic dynamical systems and instantiated it as DSE, an operator-based latent state space model that learns time-invariant encoders and linear transfer and observation operators in finite-dimensional deep feature spaces. Using stochastic realization with covariance and conditional covariance operators, DSE computes predictive state coordinates and ridge-regularized closed-form operator estimates via Galerkin regression, enabling Kalman-style sequential state estimation and Koopman spectral mode decomposition in feature space. Experiments on nonlinear oscillators and image sequences show robust forecasting and spectrum recovery under noise and partial observability, with performance comparable to or better than baselines.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers JP22H05106 and JP26H02498, JST CREST Grant Number JPMJCR1913, and JST SPRING Grant Number JPMJSP2138.

References

H. Akaike. Markovian representation of stochastic processes by canonical variables. *SIAM Journal on Control*, 13(1): 162–173, 1975.

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

A. F. Ansari, A. Heng, A. Lim, and H. Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 926–951, 23–29 Jul 2023.

H. Arbabi and I. Mezić. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

S. Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, floquet modes, and Koopman spectrum. *Physics of Fluids*, 26(9):094104, 2014.

C. R. Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186: 273–289, 1973.

P. Becker, H. Pandya, G. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann. Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 544–552. PMLR, 2019.

S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz. Modern Koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022.

C. K. Chui, J. D. Ward, and P. W. Smith. Cholesky factorization of positive definite bi-infinite matrices. *Numerical Functional Analysis and Optimization*, 5(1):1–20, 1982.

U. B. Desai, D. Pal, and R. D. Kirkpatrick. A realization approach to stochastic model reduction. *International Journal of Control*, 42(4):821–838, 1985.

A. Doucet, N. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.

L. Duncker, G. Bohner, J. Boussard, and M. Sahani. Learning interpretable continuous-time models of latent stochastic dynamical systems. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1726–1734, 09–15 Jun 2019.

R. L. Eubank and T. Hsing. Canonical correlation for stochastic processes. *Stochastic Processes and their Applications*, 118(9):1634–1661, 2008.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14(118):3753–3783, 2013.

G. H. W. Gebhardt, A. Kupcsik, and G. Neumann. The kernel Kalman rule: Efficient nonparametric inference by recursive least-squares and subspace projections. *Machine Learning*, 108:2113–2157, 2019.

S. Grünewälder, G. Lever, L. Baldassarre, S. Patterson, A. Gretton, and M. Pontil. Conditional mean embeddings as regressors. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

A. Hasan, J. M. Pereira, S. Farsiu, and V. Tarokh. Identifying latent stochastic differential equations. *IEEE Transactions on Signal Processing*, 70:89–104, 2022.

T. Hsing and R. Eubank. *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. Wiley, 2015.

D. J. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *22nd Annual Conference on Learning Theory - COLT 2009, Montreal, Quebec, Canada, June 18-21*, 2009.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

T. Katayama. *Subspace Methods for System Identification*. Communications and Control Engineering. Springer Nature, 2005.

Y. Kawahara, T. Yairi, and K. Machida. A kernel subspace method by stochastic realization for learning nonlinear dynamical systems. In *Proceedings of the 19th Conference on Neural Information Processing Systems*, pages 665–672. MIT Press, 2006.

N. Ke, R. Tanaka, and Y. Kawahara. Learning stochastic nonlinear dynamics with embedded latent transfer operators. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4861–4869. PMLR, 2025.

S. Klus, I. Schuster, and K. Muandet. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. *Journal of Nonlinear Science*, 30(1):283–315, 2019.

R. G. Krishnan, U. Shalit, and D. Sontag. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

A. Lindquist and G. Picci. A geometric approach to modelling and estimation of linear stochastic systems. *Journal of Mathematical Systems, Estimation and Control*, 1(3): 241–333, 1991.

P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656: 5–28, 2010.

A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory*, volume 4754 of *Lecture Notes in Computer Science*, pages 13–31, 2007.

L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968, 2009.

L. Song, B. Boots, S. M. Siddiqi, G. Gordon, and A. Smola. Hilbert space embeddings of hidden Markov models. In *Proceedings of the 27th International Conference on Machine Learning*, pages 991–998, 2010.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, G. R. G. Lanckriet, and B. Schölkopf. Injective Hilbert space embeddings of probability measures. In *21st Annual Conference on Learning Theory*, pages 111–122, 2008.

N. Takeishi and Y. Kawahara. Learning dynamics models with stable invariant sets. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 9782–9790, 2021.

M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

L. Xu, Y. Chen, S. Srinivasan, N. de Freitas, A. Doucet, and A. Gretton. Learning deep features in instrumental variable regression. In *International Conference on Learning Representations*, 2021.

Deep Spectral Learning of Embedded Latent Transfer Operators for Stochastic Dynamical Systems (Supplementary Material)

Ryogo Tanaka[1,2] **Yoshinobu Kawahara**[1,2]

[1]Graduate School of Information Science and Technology, The University of Osaka , Osaka , Japan
[2]Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan

A DETAILS OF NUMERICAL ALGORITHMS

This appendix provides the formal pseudo-code for the training and inference procedures of the Deep Spectral Encoder (DSE). The following algorithms instantiate the operator-theoretic framework described in the main text into implementable steps.

A.1 EMPIRICAL STOCHASTIC REALIZATION

Algorithm 1 details the computational procedure for the empirical stochastic realization introduced in Sections 3.2 and 4. This procedure extracts the finite-dimensional Markovian state coordinates from the encoder features.

Block Feature Construction. To capture temporal dependencies over the window length ℓ without excessively increasing the parameter space, the sequence of encoder features $\boldsymbol{m}_t = u_\eta(\boldsymbol{y}_t)$ is rearranged into delay vectors dimension-wise. A shared shallow network h maps these delay vectors to scalar features, which are then concatenated to form the past and future block features $\tilde{\phi}(\tilde{\boldsymbol{p}}_t)$ and $\tilde{\phi}(\tilde{\boldsymbol{f}}_t)$.

Subspace Identification. The covariance and cross-covariance operators on the Galerkin subspace are empirically estimated as the matrices $\mathbf{C}_-$, $\mathbf{C}_+$, and $\mathbf{H}$. The functional canonical correlation analysis (CCA) reduces to the singular value decomposition (SVD) of the normalized cross-covariance matrix $\mathbf{T}$, where diagonal loading is applied before whitening for numerical stability. The top r canonical directions extracted from the right singular vectors identify the predictive coordinates $\boldsymbol{x}_t$ for the latent process.

A.2 DEEP SPECTRAL ENCODER TRAINING

The training process follows a staged schedule to ensure the separation of representation learning and dynamics modeling. Phase 1 employs an alternating optimization with block cross-fitting, while Phase 2 performs end-to-end refinement. The procedure is summarized in Algorithm 2. While the theoretical formulation defines the covariance operators without centering, the empirical implementation centers the extracted feature matrices by subtracting their respective sample means before computing the sample covariance blocks and performing the stochastic realization.

A.3 COMPUTATIONAL COST AND SVD IMPLEMENTATION

We summarize the DSE-specific computational costs of stochastic realization and closed-form operator estimation. Let $N := |I_{\mathrm{SR}}|$ be the number of valid past–future windows, ℓ the window length, m the block-feature dimension, r the retained CCA state dimension, and d_A, d_B the output dimensions of the state and observation dictionaries. Neural feature extraction, block-feature construction, and decoder evaluation are architecture-dependent; Table 5 therefore focuses on the computational steps specific to DSE after the relevant features have been evaluated.

Algorithm 1 Empirical Stochastic Realization via Functional CCA

Require: Encoder features $\{\boldsymbol{m}_t\}_{t=1}^{T} \subset \mathbb{R}^m$, window length ℓ, state dimension r, head network h.
Ensure: Latent state coordinates $\{\boldsymbol{x}_t\}_{t \in I_{\mathrm{SR}}}$.

Feature Extraction:
$\tilde{\boldsymbol{v}}_{k,t}^{p} = (m_{t-1}^{(k)}, \ldots, m_{t-\ell}^{(k)})^{\top}, \quad \tilde{\boldsymbol{v}}_{k,t}^{f} = (m_{t}^{(k)}, \ldots, m_{t+\ell-1}^{(k)})^{\top}$ for $k = 1 \ldots m$.
$\tilde{\phi}^{(k)}(\tilde{\boldsymbol{p}}_t) = h(\tilde{\boldsymbol{v}}_{k,t}^{p}), \quad \tilde{\phi}^{(k)}(\tilde{\boldsymbol{f}}_t) = h(\tilde{\boldsymbol{v}}_{k,t}^{f})$ for $k = 1 \ldots m$.
Stack into block features: $\tilde{\phi}(\tilde{\boldsymbol{p}}_t), \tilde{\phi}(\tilde{\boldsymbol{f}}_t) \in \mathbb{R}^m$.
$\Phi_p = \mathrm{Center}([\tilde{\phi}(\tilde{\boldsymbol{p}}_t)]_{t \in I_{\mathrm{SR}}}), \quad \Phi_f = \mathrm{Center}([\tilde{\phi}(\tilde{\boldsymbol{f}}_t)]_{t \in I_{\mathrm{SR}}})$.
Subspace Identification:
$\mathbf{C}_{-} = \frac{1}{|I_{\mathrm{SR}}|}\Phi_p\Phi_p^{\top}, \quad \mathbf{C}_{+} = \frac{1}{|I_{\mathrm{SR}}|}\Phi_f\Phi_f^{\top}, \quad \mathbf{H} = \frac{1}{|I_{\mathrm{SR}}|}\Phi_f\Phi_p^{\top}$.
$\mathbf{T} = \mathbf{C}_{+}^{-1/2}\mathbf{H}\mathbf{C}_{-}^{-1/2}$.
Compute rank-r SVD: $\mathbf{T} \approx \mathbf{U}_{l,r}\boldsymbol{\Sigma}_r\mathbf{U}_{r,r}^{\top}$.
State Construction:
Canonical direction matrix: $\mathbf{A}_{can} = \mathbf{C}_{-}^{-1/2}\mathbf{U}_{r,r} \in \mathbb{R}^{m \times r}$.
Past canonical variates: $\boldsymbol{z}_{p,t} = \mathbf{A}_{can}^{\top}\tilde{\phi}(\tilde{\boldsymbol{p}}_t)$ for $t \in I_{\mathrm{SR}}$.
$\boldsymbol{x}_t = \boldsymbol{\Sigma}_r^{1/2}\boldsymbol{z}_{p,t}$.

Table 5: DSE-specific computational cost.

Step	Time	Memory
CCA covariance blocks $\mathbf{C}_{-}, \mathbf{C}_{+}, \mathbf{H}$	$\mathcal{O}(Nm^2)$	$\mathcal{O}(m^2)$
CCA whitening and SVD	$\mathcal{O}(m^3)$	$\mathcal{O}(m^2)$
CCA projection	$\mathcal{O}(Nmr)$	$\mathcal{O}(Nr)$
Transfer solve $\hat{\mathbf{V}}_A$	$\mathcal{O}(Nd_A^2 + d_A^3)$	$\mathcal{O}(d_A^2)$
Observation solve $\hat{\mathbf{V}}_B$	$\mathcal{O}(N(d_A^2 + d_A d_B) + d_A^3 + d_A^2 d_B)$	$\mathcal{O}(d_A^2 + d_A d_B)$

The main bottlenecks are the covariance formation and the CCA whitening/SVD when the block-feature dimension m is large. The window length ℓ affects the architecture-dependent block-feature construction cost and enters the computational cost through the chosen block-feature dimension m. The operator-estimation steps work with feature- and dictionary-dimensional matrices and do not construct an $N \times N$ sample-indexed Gram matrix. If intermediate features are cached, the additional storage is linear in N.

The SVD in Algorithm 1 solves the finite-dimensional CCA problem for the normalized past–future cross-covariance matrix. The retained rank-r singular directions define the predictive state coordinates used by stochastic realization. During representation learning, this CCA/SVD block is differentiable: gradients can be propagated through the rank-r SVD to the normalized cross-covariance matrix and then to the block-feature and encoder parameters. This differentiation is used for learning the representation; the transfer and observation operators are still obtained by the ridge-regression updates in Eq. (14) and Eq. (15) rather than treated as unconstrained recurrent parameters. Numerically, the whitening step is regularized by diagonal loading, and only a fixed rank-r subspace is retained, which truncates directions associated with small or unstable singular values.

A.4 SEQUENTIAL STATE ESTIMATION (FILTERING)

While the underlying dynamical system is nonlinear and stochastic, DSE enables sequential inference via linear recursions within the learned Hilbert space embeddings. Algorithm 3 summarizes the noise statistics estimation and the recursive filtering process.

Empirical Noise Estimation. The filtering process requires the process and observation noise covariances, $\mathcal{Q}$ and $\mathcal{R}$. In our framework, these are estimated empirically from the residuals of the learned operators on the training data. This approach accounts for the approximation error of the Galerkin projection as a form of "modeling noise." The state residuals $\boldsymbol{e}_t^x$ and observation residuals $\boldsymbol{e}_t^y$ are computed from the difference between the one-step operator predictions and the actual encoded features.

Algorithm 2 Training of Deep Spectral Encoder

Require: Observations $\{\boldsymbol{y}_t\}_{t=1}^T$, window ℓ, state dim r, dictionaries d_A, d_B, folds K.
Ensure: Parameters $(\eta, \alpha, \theta, \omega, \xi_A, \xi_B)$, operators $\{\hat{\mathbf{V}}_A, \hat{\mathbf{V}}_B\}$.

Phase 1: Staged Training with Cross-Fitting
Initialize η, α (frozen), $\theta, \omega, \xi_A, \xi_B$. Partition I_{SR} into $\{B_k\}_{k=1}^K$.
while not converged **do**
  $\{\boldsymbol{x}_t\} \leftarrow$ Stochastic Realization$(\{u_\eta(\boldsymbol{y}_t)\}, \ell, r)$ (see Algorithm 1).
  for $k = 1$ **to** K **do**
    $I_{-k} = I_{\mathrm{SR}} \setminus B_k$.
    $\hat{\mathbf{V}}_A^{(-k)} = \Phi_{-k}^+ (\Phi_{-k}^-)^\top \big(\Phi_{-k}^- (\Phi_{-k}^-)^\top + \lambda_A \mathbf{I}\big)^{-1}$.
    $\hat{\mathbf{V}}_B^{(-k)} = \Psi_{-k} (\widehat{\Phi}_{-k}^+)^\top \big(\widehat{\Phi}_{-k}^+ (\widehat{\Phi}_{-k}^+)^\top + \lambda_B \mathbf{I}\big)^{-1}$.
    $\hat{\boldsymbol{h}}_{A,t}^{(\mathrm{cf})} = \hat{\mathbf{V}}_A^{(-k)} \phi_\theta(\boldsymbol{x}_{t-1})$, $\hat{\boldsymbol{h}}_{B,t}^{(\mathrm{cf})} = \hat{\mathbf{V}}_B^{(-k)} \phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})$ for $t \in B_k$.
  end for
  Update $(\theta, \omega, \xi_A, \xi_B)$ via Adam on $\mathcal{L}_{\mathrm{rec}}(\hat{\boldsymbol{h}}_A^{(\mathrm{cf})}, \hat{\boldsymbol{h}}_B^{(\mathrm{cf})})$.
end while
Phase 2: End-to-End Refinement
Unfreeze η, α.
while not converged **do**
  Compute $\hat{\boldsymbol{y}}_{t|t-1} = g_\alpha(r_{\xi_B}(\hat{\mathbf{V}}_B \phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})))$ for all t.
  Update $(\eta, \alpha, \theta, \omega, \xi_A, \xi_B)$ via Adam on $\sum_t \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}\|^2 - \lambda_{\mathrm{cca}} \sum_{l=1}^r s_l$.
end while

Feature-Space Recursion. The recursion implemented in Algorithm 3 propagates the first and second moments of the embedded state distribution. Unlike standard Kalman filters that operate on the observation space, this filter acts on the feature coordinates $\boldsymbol{\phi}_t$. The "Innovation" step incorporates new information by projecting the observation into the feature space via $\psi_\omega(u_\eta(\boldsymbol{y}_t))$, allowing for nonlinear updates to the latent state while maintaining a linear computational structure.

Algorithm 3 Feature-Space Sequential State Estimation

Require: Learned DSE, new observations $\{\boldsymbol{y}_t\}_{t=1}^T$, regularization $\gamma_{\mathcal{Q}}, \gamma_{\mathcal{R}}$.
Ensure: State estimates $\{\hat{\boldsymbol{x}}_t\}_{t=1}^T$.

Noise Estimation:
$\boldsymbol{e}_t^x = \phi_\theta(\boldsymbol{x}_t) - \hat{\mathbf{V}}_A \phi_\theta(\boldsymbol{x}_{t-1})$, $\boldsymbol{e}_t^y = \psi_\omega(\boldsymbol{m}_t) - \hat{\mathbf{V}}_B \phi_\theta(\boldsymbol{x}_t)$.
$\mathcal{Q} = \frac{1}{|I_{\mathrm{SR}}|-1} \sum_t \boldsymbol{e}_t^x (\boldsymbol{e}_t^x)^\top + \gamma_{\mathcal{Q}} \mathbf{I}$, $\mathcal{R} = \frac{1}{|I_{\mathrm{SR}}|} \sum_t \boldsymbol{e}_t^y (\boldsymbol{e}_t^y)^\top + \gamma_{\mathcal{R}} \mathbf{I}$.
Recursion:
Initialize $\boldsymbol{\mu}_0^+, \boldsymbol{\Sigma}_0^+$ from training ensemble.
for $t = 1$ **to** T **do**
  $\boldsymbol{\mu}_t^- = \hat{\mathbf{V}}_A \boldsymbol{\mu}_{t-1}^+$, $\boldsymbol{\Sigma}_t^- = \hat{\mathbf{V}}_A \boldsymbol{\Sigma}_{t-1}^+ \hat{\mathbf{V}}_A^\top + \mathcal{Q}$.
  $\hat{\boldsymbol{\psi}}_t^- = \hat{\mathbf{V}}_B \boldsymbol{\mu}_t^-$, $\mathbf{S}_t = \hat{\mathbf{V}}_B \boldsymbol{\Sigma}_t^- \hat{\mathbf{V}}_B^\top + \mathcal{R}$.
  $\mathbf{K}_t = \boldsymbol{\Sigma}_t^- \hat{\mathbf{V}}_B^\top \mathbf{S}_t^{-1}$.
  $\boldsymbol{\psi}_t = \psi_\omega(u_\eta(\boldsymbol{y}_t))$.
  $\boldsymbol{\mu}_t^+ = \boldsymbol{\mu}_t^- + \mathbf{K}_t(\boldsymbol{\psi}_t - \hat{\boldsymbol{\psi}}_t^-)$.
  $\boldsymbol{\Sigma}_t^+ = (\mathbf{I} - \mathbf{K}_t \hat{\mathbf{V}}_B) \boldsymbol{\Sigma}_t^-$.
  $\hat{\boldsymbol{x}}_t = r_{\xi_A}(\boldsymbol{\mu}_t^+)$.
end for

B TRAINING DETAILS

B.1 CROSS FITTING

To prevent self-fitting and temporal leakage when estimating the finite-dimensional transfer operators and learning the read-out maps, we adopt block cross-fitting over contiguous time segments. We partition the time index set $\{1, \ldots, T\}$ into

K disjoint, contiguous blocks $\{B_k\}_{k=1}^{K}$ and define the out-of-fold time index set for block B_k by $I_{-k} := \{1, \ldots, T\} \backslash B_k$. In all experiments, we use $K = 5$ contiguous blocks with a minimum block size of 100 time steps. All stage-1 estimators are fitted on I_{-k}, and the corresponding out-of-fold features are then computed on the held-out block B_k. This construction preserves the test-time causal path and removes information flow from B_k into its own predictors. The same scheme is applied to the estimation of both the state process transfer operator and the observation operator.

Let $\phi_\theta : \mathbb{R}^r \to \mathbb{R}^{d_A}$ be the state feature map, $\mathbf{V}_A : \mathbb{R}^{d_A} \to \mathbb{R}^{d_A}$ the transfer operator in the feature space, and $r_{\xi_A} : \mathbb{R}^{d_A} \to \mathbb{R}^r$ the state read-out map. For each fold k, stage-1 fits an out-of-fold operator $\mathbf{V}_A^{(-k)}$ by ridge-regularized multi-target least squares problem on I_{-k}:

$$\mathbf{V}_A^{(-k)} = \Phi_{-k}^{+}(\Phi_{-k}^{-})^\top \big(\Phi_{-k}^{-}(\Phi_{-k}^{-})^\top + \lambda_A \mathbf{I}\big)^{-1}, \tag{26}$$

where $\Phi_{-k}^{-} = [\phi_\theta(\boldsymbol{x}_{t-1})]_{t \in I_{-k}} \in \mathbb{R}^{d_A \times |I_{-k}|}$ and $\Phi_{-k}^{+} = [\phi_\theta(\boldsymbol{x}_t)]_{t \in I_{-k}} \in \mathbb{R}^{d_A \times |I_{-k}|}$. Out-of-fold features on B_k are then computed as

$$\widehat{\mathbf{H}}_{A,B_k}^{(\mathrm{cf})} := \mathbf{V}_A^{(-k)}\, \Phi_{B_k}^{-} \tag{27}$$

where $\Phi_{B_k}^{-} = \big[\phi_\theta(\boldsymbol{x}_{t-1})\big]_{t \in B_k}$. Let $\widehat{\mathbf{H}}_A^{(\mathrm{cf})} \in \mathbb{R}^{d_A \times T}$ denote the out-of-fold design matrix obtained by stacking the block-wise matrices $\widehat{\mathbf{H}}_{A,B_k}^{(\mathrm{cf})}$ column-wise over all folds, such as

$$\widehat{\mathbf{H}}_A^{(\mathrm{cf})} = [\widehat{\mathbf{H}}_{A,B_1}^{(\mathrm{cf})}, \ldots, \widehat{\mathbf{H}}_{A,B_K}^{(\mathrm{cf})}]. \tag{28}$$

Then stage-2 learns the state read-out map parameters ξ_A only from the out-of-fold design:

$$\widehat{\xi}_A = \operatorname*{argmin}_{\xi_A} \frac{1}{|\, I_{\mathrm{SR}} \,|} \sum_{t \in I_{\mathrm{SR}}} \Big\| \boldsymbol{x}_t - r_{\xi_A}\Big(\widehat{\boldsymbol{h}}_{A,t}^{(\mathrm{cf})}\Big) \Big\|_2^2 + \lambda_{r,A}\Omega(\xi_A), \tag{29}$$

where $\widehat{\boldsymbol{h}}_{A,t}^{(\mathrm{cf})}$ denotes the t-th column of $\widehat{\mathbf{H}}_A^{(\mathrm{cf})}$; $\Omega(\cdot)$ denotes a regularizer on the read-out parameters and $\lambda_{r,A}$ is the corresponding regularization coefficient. This cross-fitted two-stage estimator preserves the computation graph at the inference one-step prediction $\hat{\boldsymbol{x}}_{t\,|\,t-1}$ and mitigates co-adaptation between operators and neural networks as encoder/decoder.

For the observation operator we use the same block cross-fitting protocol as for the state operator. Let $\psi_\omega : \mathbb{R}^m \to \mathbb{R}^{d_B}$ be the observation feature map, $\mathbf{V}_B : \mathbb{R}^{d_A} \to \mathbb{R}^{d_B}$ the observation operator in feature space, and $r_{\xi_B} : \mathbb{R}^{d_B} \to \mathbb{R}^m$ the observation read-out map. For each fold k we consider the out-of-fold index set I_{-k} and the corresponding held-out block B_k defined above. On I_{-k} we form the feature matrix for state process as $\Phi_{-k} = [\phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})]_{t \in I_{-k}} \in \mathbb{R}^{d_A \times |I_{-k}|}$ and the feature matrix for observations as $\Psi_{-k} = [\psi_\omega(\boldsymbol{m}_t)]_{t \in I_{-k}} \in \mathbb{R}^{d_B \times |I_{-k}|}$. The first stage then estimates an out-of-fold matrix representation for the observation operator $\mathbf{V}_B^{(-k)}$ by ridge-regularized least squares,

$$\mathbf{V}_B^{(-k)} = \Psi_{-k}\, \Phi_{-k}^{\top} \big(\Phi_{-k}\Phi_{-k}^{\top} + \lambda_B \mathbf{I}\big)^{-1}, \tag{30}$$

which has the same closed form as in the main paper but is fitted only on indices in I_{-k}. On the held-out block B_k we construct the corresponding out-of-fold design by propagating the one-step state predictors through $\mathbf{V}_B^{(-k)}$, that is,

$$\Phi_{B_k} = \big[\phi_\theta(\hat{\boldsymbol{x}}_{t|t-1})\big]_{t \in B_k}, \qquad \widehat{\mathbf{H}}_{B,B_k}^{(\mathrm{cf})} = \mathbf{V}_B^{(-k)}\, \Phi_{B_k}. \tag{31}$$

Arranging the block-wise matrices $\widehat{\mathbf{H}}_{B,B_k}^{(\mathrm{cf})}$ column-wise over all folds yields the global out-of-fold design $\widehat{\mathbf{H}}_B^{(\mathrm{cf})} \in \mathbb{R}^{d_B \times T}$ used in Stage-2. Let $\mathbf{M} = [\boldsymbol{m}_t]_{t \in I_{\mathrm{SR}}}$ denote the encoder feature matrix whose columns are the encoder features $\boldsymbol{m}_t$. The observation read-out map parameters ξ_B are learned only from the out-of-fold design:

$$\widehat{\xi}_B = \operatorname*{argmin}_{\xi_B} \frac{1}{|I_{\mathrm{SR}}|} \sum_{t \in I_{\mathrm{SR}}} \Big\| \boldsymbol{m}_t - r_{\xi_B}\Big(\widehat{\boldsymbol{h}}_{B,t}^{(\mathrm{cf})}\Big) \Big\|_2^2 + \lambda_{r,B}\Omega(\xi_B), \tag{32}$$

where $\widehat{\boldsymbol{h}}_{B,t}^{(\mathrm{cf})}$ denotes the t-th column of $\widehat{\mathbf{H}}_B^{(\mathrm{cf})}$ and $\lambda_{r,B}$ is the corresponding regularization coefficient. By this construction, each column of $\widehat{\mathbf{H}}_B^{(\mathrm{cf})}$ is generated by an operator $\mathbf{V}_B^{(-k)}$ that has never been fitted on its own time index, so the features entering the second-stage regression are genuinely out-of-fold.

This separation helps to mitigate co-adaptation between the encoder and the observation operator and encourages $\mathbf{V}_B$ to act as a stable, time-invariant linear map from the state feature subspace induced by ϕ_θ into the observation feature subspace induced by ψ_ω, rather than drifting toward a nearly trivial transfer matrix.

Table 6: Ablation Study on Training Schedules Using Quad-Link Pendulum Dataset

	Length 1.5k		Length 15k	
Schedule	**No noise**	**With noise**	**No noise**	**With noise**
Staged (Proposed)	$\mathbf{0.2006} \pm 0.0228$	$\mathbf{0.2593} \pm 0.0274$	$\mathbf{0.2615} \pm 0.0211$	$\mathbf{0.2683} \pm 0.0254$
Joint	0.2248 ± 0.0378	0.2977 ± 0.0512	0.2835 ± 0.0344	0.3202 ± 0.0474

Table 7: Ablation Study on Components Using the 1.5×10^3 Corrupted Quad-Link Benchmark

Variant	**One-step MSE**	**Multi-step MSE at $H = 10$**
Full DSE	$\mathbf{0.2593} \pm 0.0274$	$\mathbf{0.2952} \pm 0.0357$
w/o CCA-based state construction	0.2903 ± 0.0455	0.3406 ± 0.0551
w/o closed-form operator estimation	0.2758 ± 0.0405	0.3203 ± 0.0505
joint training instead of staged training	0.2977 ± 0.0512	0.3553 ± 0.0752

B.2 ABLATION STUDY

We report two ablations on the quad-link pendulum image benchmark used in Section 5. The first ablation compares the staged optimization schedule with joint end-to-end training, and the second isolates the contribution of the main DSE components on the 1.5×10^3 corrupted regime. The data generation, encoder–decoder architectures, and regularization coefficients are kept identical to the main experiment unless otherwise stated.

Training Schedule. In the staged schedule, operator estimation is decoupled from representation learning. We first fix the encoder u_η and decoder g_α, estimate the operators $\mathbf{V}_A$ and $\mathbf{V}_B$ by solving regularized least-square problems corresponding to the Galerkin projections on the Hilbert spaces spanned by the feature maps ϕ_θ and ψ_ω, and learn the read-out maps r_{ξ_A} and r_{ξ_B} by minimizing the corresponding regularized objectives on the operator-projected features. After updating the operators and the read-out maps in the first phase, we further optimize the neural components along the same training objective as in the main experiment, namely the reconstruction loss augmented by a regularization term given by the sum of canonical correlation coefficients obtained from the operator-based stochastic realization. In this refinement phase, $\mathbf{V}_A$ and $\mathbf{V}_B$ may be kept fixed or recomputed from the current features, and the read-out maps may either be fixed or fine-tuned depending on the selected implementation. Under this schedule the temporal structure is primarily encoded through $\mathbf{V}_A$ and $\mathbf{V}_B$, whereas the encoder and decoder are encouraged to produce feature coordinates and reconstructions that are compatible with the latent Markov process induced by the staged operator estimation rather than directly absorbing sequence-level dynamics.

In the joint schedule, all modules $(\eta, \theta, \omega, \alpha, \xi_A, \xi_B)$ and the operators are trained simultaneously from scratch with the same reconstruction loss augmented with a canonical-correlation regularizer. At each gradient step, the closed-form solutions for $\mathbf{V}_A$ and $\mathbf{V}_B$ are recomputed from the current features and treated as differentiable functions of ϕ_θ and ψ_ω. This tight coupling tends to increase gradient interference between representation learning and operator estimation, so that the encoder absorbs much of the temporal structure into its own representation.

Table 6 reports the one-step prediction MSE of DSE under the staged and joint schedules on quad-link pendulum images, for training sequence lengths 1.5×10^3 and 1.5×10^4 and for both the clean and corrupted regimes. Across these settings, the staged schedule attains lower or comparable error, with the largest gap typically observed for long corrupted sequences where joint training frequently converges to degenerate operator solutions. These results support the use of the staged schedule as the default optimization strategy in the main experiments.

Component Ablation. We further isolate the contribution of the main DSE components on the 1.5×10^3 corrupted regime. The variant without CCA-based state construction uses the encoder features directly as state coordinates, bypassing the stochastic realization step. The variant without closed-form operator estimation replaces the ridge-regression solutions for $\mathbf{V}_A$ and $\mathbf{V}_B$ with three-layer neural networks parameterizing the transition and observation maps, trained by gradient-based optimization. The joint-training variant is the same as the joint schedule above and removes the staged optimization procedure. The role of learned deep features is assessed by the ELTO-KF comparison in Table 1, which follows the same broad operator-based filtering framework but does not learn the explicit neural feature and dictionary spaces used by DSE.

Table 8: Quad-Link Data Generation and Corruption Setting.

Item	Setting and Values
Simulation / observation	
Image resolution	48×48 grayscale
Initial joint angles	$q(0) \sim \mathrm{Unif}([-\pi, \pi]);\ \dot{q}(0) = 0$
Friction coefficient	0.1
Simulator time step	$dt = 10^{-4}\,\mathrm{s}$
Frame interval	$\Delta t_{\mathrm{frame}} = 0.05\,\mathrm{s}$
Sequence / split	
Sequence lengths evaluated	1.5×10^3 and 1.5×10^4 frames
# trajectories (train / val)	$1200/200$
# trajectories for decoder (if applicable)	300
Corruption setting	
Correlation parameter	$\rho = 0.2$
Thresholds	$[0,\ 0.25,\ 0.75,\ 1.0]$
Clean prefix	First 5 frames are kept clean
Evaluation	
Metric	Prediction mean squared error
Repeats	5 independent runs (mean $\pm$ std.)

Table 7 reports the one-step prediction MSE and the multi-step prediction MSE at horizon $H = 10$ for these variants. Removing any of the main components degrades prediction relative to the full DSE. The degradation under the variant without CCA-based state construction is more pronounced for multi-step prediction, indicating that the predictive state coordinates obtained by stochastic realization contribute to rollout stability. Replacing the closed-form operator estimation with neural-network parameterization increases the variability across runs, while joint training gives the largest degradation among the tested variants.

C EXPERIMENTAL SETTING

This section specifies the data generation procedures, noise settings, baseline configurations, and evaluation protocols used in the experiments.

C.1 SEQUENTIAL STATE ESTIMATION

We evaluate sequential prediction on simulated quad-link pendulum trajectories observed as 48×48 grayscale image sequences. Initial joint angles are sampled uniformly from $[-\pi,\ \pi]$ with zero initial velocities, the viscous friction coefficient is set to 0.1, the simulator time step is 10^{-4} seconds, and frames are recorded every 0.05 seconds. In the corrupted regime, we use the fixed noise configuration with $\rho = 0.2$ and thresholds $[0,\ 0.25,\ 0.75,\ 1.0]$, and we keep the first five frames in each sequence clean. We report prediction mean squared error for the two sequence-length settings 1.5×10^3 and 1.5×10^4, and Table 1 summarizes the mean and standard deviation over 5 independent runs.

All methods use the same image resolution, preprocessing, simulator configuration, and data split. The quad-link sequences are generated using the simulator distributed in the RKN reference repository at `https://github.com/ALRhub/rkn-share`, and the remaining simulator parameters follow the simulator defaults.

Simulation and Corruption Setting. Table 8 summarizes the quad-link data generation, corruption process, and evaluation protocol. We follow the simulator configuration provided in the RKN reference implementation [Becker et al., 2019] and adopt the same data split protocol as in the setting of ELTO-based method [Ke et al., 2025]. Specifically, we use $1,200$ trajectories to train our method, 300 trajectories to train the decoder, and 200 trajectories for validation and evaluation.

Parameters for Our Method. Table 9 reports the architecture and training hyperparameters of our method used in the quad-link experiment. We use the staged training schedule described in the main text. To ensure numerical stability when computing the normalized cross-covariance $\tilde{\mathbf{T}} = (\mathbf{C}_+ + \delta_{\mathrm{cca}}\mathbf{I})^{-1/2}\,\mathbf{H}\,(\mathbf{C}_- + \delta_{\mathrm{cca}}\mathbf{I})^{-1/2}$, we apply diagonal loading with $\delta_{\mathrm{cca}} > 0$ as the covariance regularization. Furthermore, in our experiments, we set $\gamma_{\mathcal{Q}} = \gamma_{\mathcal{R}} = 10^{-6}$ in the clean regime and $\gamma_{\mathcal{Q}} = \gamma_{\mathcal{R}} = 10^{-3}$ in the corrupted regime.

Table 9: Hyperparameters for Our Method in Quad-Link Experiment.

Component / hyperparameter	Architectures and Values
Encoder u_η	
Convolution 1	12 filters, 5×5 kernel, stride 2×2, ReLU; 2×2 max-pooling with stride 2×2
Convolution 2	12 filters, 3×3 kernel, stride 2×2, ReLU; 2×2 max-pooling with stride 2×2
Fully connected	200 units, ReLU
Output heads	Mean head: linear; variance head: $\mathrm{ELU}(\cdot) + 1$ to enforce nonnegative variance
Padding / normalization	Zero-padding is chosen so that each convolution preserves the spatial resolution; layer normalization is applied in all convolutional layers
Feature dimension m	100
Decoder g_α	
Decoder architecture	Fully connected: 144, ReLU; transposed conv: 16 filters, 5×5 kernel, stride 4×4, ReLU; transposed conv: 12 filters, 3×3 kernel, stride 4×4, ReLU; transposed conv output: 1 channel, stride 1×1, Sigmoid
Output type	Sigmoid output (image intensity in $(0, 1)$)
Deep spectral learning / state model	
Latent state dimension r	20
SR window length ℓ	30
Dictionary dimensions d_A, d_B	$50 \,/\, 50$
State dictionary ϕ_θ (DF-A feature net)	MLP: $20 \to 128 \to 64 \to 50$, ReLU, dropout 0.1
Observation dictionary ψ_ω (DF-B obs net)	MLP: $100 \to 64 \to 32 \to 50$, ReLU, dropout 0.1
Feature mapping MLP h (encoder output $\to$ block feature)	1-hidden-layer MLP: $\ell(=30) \to 32 \to 1$, ReLU
Read-out map r_{ξ_A}	2-hidden-layer MLP: $50 \to 32 \to 32 \to 20$, ReLU
Read-out map r_{ξ_B}	2-hidden-layer MLP: $50 \to 32 \to 32 \to 100$, ReLU
Regularization	
Ridge regularization λ_A, λ_B	$10^{-2} \,/\, 10^{-3}$
Covariance regularization δ_{cca} (diagonal loading)	10^{-3}
CCA regularization weight λ_{cca}	10^{-4}
Optimization (staged schedule)	
Optimizer	Adam
Learning rate $(u_\eta, g_\alpha, \phi_\theta, \psi_\omega)$	10^{-3}
Batch size	150
Stage-1 epochs	25
Stage-2 epochs	100
Cross-fitting (contiguous blocks)	$K = 5$, min_block_size $= 100$
Compute / reproducibility	
Framework / hardware	Tesla V100S-PCIE-32GB GPU

C.2 MODE DECOMPOSITION

We evaluate Koopman spectrum recovery on the Van der Pol (VDP) oscillator with additive observation noise and the Stuart-Landau (SL) oscillator with additive process noise, defined in Eq. (33) and (34). For each noise level, we compute the absolute eigenvalue error between the estimated spectrum and a reference spectrum, and report mean $\pm$ standard deviation over 50 independent trials. To ensure reproducibility, we summarize the detailed simulation configurations and the hyperparameters of the proposed method in the following tables. For the oscillator experiments, although the simulator outputs scalar observations, we use time-delay embedding to construct vector-valued inputs to the encoder and decoder.

C.2.1 Van der Pol Oscillator with Observation Noise

Governing Equations and Simulation. This section details the data generation process for the VDP oscillator evaluated in Section 6. The deterministic continuous-time system is governed by

$$\frac{dx}{dt} = y, \quad \frac{dy}{dt} = \mu(1 - x^2)y - x, \tag{33}$$

Table 10: Simulation Setting for VDP Oscillator Experiment

Item	Setting and Values
Parameter	$\mu = 2.0$
Integrator / step size	4th-order Runge–Kutta, $\Delta t = 0.1$
Initial condition	$(x_0, y_0) = (2.0, 0.0)$
Trajectory length	$T = 3{,}500$
Train/validation split	first 3,000 / last 500 samples
Observation	scalar observation with additive noise
Observation noise	i.i.d. Gaussian, mean 0, variance σ^2_{obs}
Noise sweep	$\sigma^2_{\text{obs}} \in \{0.01, 0.02, \ldots, 0.20\}$
Trials per noise level	50 independent trajectories
Reference spectrum	integer harmonics of the frequency (noise-free); $\omega = 0.823498$
Metric / reporting	absolute eigenvalue error; mean $\pm$ std over 50 trials

Table 11: Parameter Setting for Our Method in VDP Oscillator Experiment.

Component / hyperparameter	Architectures and values
Encoder u_η	
Encoder architecture	4-layer MLP with $\tanh$: $30 \to 128 \to 128 \to 64 \to 50$
Feature dimension m	50
Decoder g_α	
Decoder architecture	4-layer MLP with $\tanh$: $50 \to 64 \to 128 \to 128 \to 30$ (linear output)
Deep spectral learning / state model	
Latent state dimension r	9
SR window length ℓ	20
Dictionary dimensions d_A, d_B	30 / 30
State dictionary ϕ_θ (DF-A feature net)	MLP: $9 \to 128 \to 64 \to 30$, ReLU, dropout 0.1
Observation dictionary ψ_ω (DF-B obs net)	MLP: $50 \to 64 \to 32 \to 30$, ReLU, dropout 0.1
Feature mapping MLP h (encoder output $\to$ block feature)	1-hidden-layer MLP: $\ell(=20) \to 32 \to 1$, ReLU
Read-out map r_{ξ_A}	2-hidden-layer MLP: $30 \to 32 \to 32 \to 9$, ReLU
Read-out map r_{ξ_B}	2-hidden-layer MLP: $30 \to 32 \to 32 \to 50$, ReLU
Regularization	
Ridge regularization λ_A, λ_B	$10^{-2}/10^{-3}$
Covariance regularization δ_{cca} (diagonal loading)	10^{-3}
CCA regularization weight λ_{cca}	10^{-3}
Optimization (staged schedule)	
Optimizer	Adam
Learning rate (u_η, g_α)	10^{-4}
Learning rate $(\phi_\theta, \psi_\omega)$	10^{-3}
Batch size	100
Stage-1 epochs	25
Stage-2 epochs	100
Cross-fitting (contiguous blocks)	$K = 5$, min_block_size $= 100$
Compute / reproducibility	
Framework / hardware	Tesla V100S-PCIE-32GB GPU

where the nonlinearity parameter μ is defined in Table 10. Following the simulation configuration used in the ELTO supplementary setting [Ke et al., 2025], we integrate this system numerically using the fourth-order Runge-Kutta method. The precise simulation parameters, including the initial conditions, time step Δt, trajectory lengths, and training splits, are summarized in Table 10.

Noise Setting and Evaluation Protocol. To rigorously evaluate robustness, we corrupt the observations with independent and identically distributed zero-mean Gaussian noise having variance σ^2_{obs}. As outlined in Section 6, we sweep σ^2_{obs} from 0.01 to 0.20 with an increment of 0.01. The number of independent trials generated per noise level is provided in Table 10. For the evaluation protocol, the reference spectrum is constructed using the integer harmonics of the noise-free limit-cycle frequency $\omega = 0.823498$, which corresponds to extracting the discrete-time eigenvalues at the sampling interval Δt.

Table 12: Simulation Setting for SL Oscillator Experiment

Item	Value
Parameters	$\mu = 1.0,\ \gamma = 0.9,\ \beta = 0.3$
Integrator / step size	4th-order Runge–Kutta, $\Delta t = 0.1$
Initial condition	$(r_0, \theta_0) = (0.1, 0.0)$
Trajectory length	$T = 3{,}500$
Train/validation split	first 3,000 / last 500 samples
Observation	$y_t = e^{i\theta_t}$
Observation noise	i.i.d. Gaussian, variance fixed to 0.01
Process noise sweep	variance $\epsilon_{\text{proc}} \in \{0.01, 0.02, \ldots, 0.09\}$
Trials per noise level	50 independent trajectories
Reference spectrum	asymptotic theory under weak process noise [Bagheri, 2014]; $\omega = 0.6$, $\kappa = 3.0$
Metric / reporting	absolute eigenvalue error; mean $\pm$ std over 50 trials

Table 13: Parameter Setting for Our Method in SL Oscillator Experiment.

Component / hyperparameter	Architectures and values
Encoder u_η	
Encoder architecture	4-layer MLP with $\tanh$: $20 \to 128 \to 128 \to 64 \to 50$
Feature dimension m	50
Decoder g_α	
Decoder architecture	4-layer MLP with $\tanh$: $50 \to 64 \to 128 \to 128 \to 20$ (linear output)
Deep spectral learning / state model	
Latent state dimension r	13
SR window length ℓ	20
Dictionary dimensions d_A, d_B	30 / 30
State dictionary ϕ_θ (DF-A feature net)	MLP: $13 \to 128 \to 64 \to 30$, ReLU, dropout 0.1
Observation dictionary ψ_ω (DF-B obs net)	MLP: $50 \to 64 \to 32 \to 30$, ReLU, dropout 0.1
Feature mapping MLP h (encoder output $\to$ block feature)	1-hidden-layer MLP: $\ell(=20) \to 64 \to 1$, ReLU
Read-out map r_{ξ_A}	2-hidden-layer MLP: $30 \to 32 \to 32 \to 13$, ReLU
Read-out map r_{ξ_B}	2-hidden-layer MLP: $30 \to 32 \to 32 \to 50$, ReLU

Parameters for our method. Table 11 reports the architecture and training hyperparameters of our method used in the VDP experiment. As introduced in Appendix C.1, we use the same diagonal loading $\delta_{\text{cca}} > 0$ for the covariance regularization to ensure numerical stability.

C.2.2 Stuart-Landau Oscillator with Process Noise

Governing Equations and Observation. This section details the explicit complex-valued differential equations and observation functions for SL oscillator evaluated in Section 6. The stochastic continuous-time system is governed by

$$\frac{\partial z}{\partial t} = (\mu + i\gamma)z - (1 + i\beta)|z|^2 z + \sqrt{\epsilon_{\text{proc}}}\eta, \tag{34}$$

where the parameters μ, γ, and β govern the limit-cycle behavior, and ϵ_{proc} scales the complex-valued system noise η. The specific values for these parameters are defined in Table 12. The system is partially observed via the observation function $y_t = e^{i\theta_t}$, where θ_t denotes the phase of the complex state z at time t.

Simulation, Noise Setting, and Evaluation Protocol. Table 12 summarizes the data generation, noise settings, and evaluation protocol for the SL oscillator. Following the simulation configuration used in the ELTO supplementary setting [Ke et al., 2025], we integrate this system numerically using the fourth-order Runge-Kutta method. To assess robustness against dynamic stochasticity, we fix the observation noise variance to 0.01 and sweep the process noise variance ϵ_{proc} from 0.01 to 0.09 with an increment of 0.01. Because process noise inherently alters the Koopman spectrum, the reference spectrum used for evaluation is analytically derived from the asymptotic theory of the SL oscillator under weak process noise [Bagheri, 2014]. Specifically, the reference eigenvalues are computed using the fundamental frequency $\omega = 0.6$ and the decay rate parameter $\kappa = 3.0$.

Parameters for Our Method. Table 13 reports the architecture hyperparameters of our method used in the SL experiment. The regularization, optimization stages, and reproducibility settings are identical to those in the VDP experiment detailed in Table 11. As introduced in Appendix C.1, we apply the same diagonal loading for the covariance regularization to ensure numerical stability.