LEARNING INVERTIBLE OBSERVABLES FOR STRUCTURALLY CONSISTENT KOOPMAN MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the intricate dynamics of systems, from molecular interactions to climate patterns, remains a central challenge in science and engineering. The Koopman operator provides a powerful mathematical framework by translating nonlinear dynamics into a tractable linear form. However, current methods face significant limitations, including lack of invertibility, inadequate handling of system input, and inability to represent measurement distributions. We introduce an invertible, Koopman-consistent framework, built on principles of invertible normalizing flows, that addresses these issues. Our approach provides a structurally sound architecture that supports invertible observable functions, is explicitly conditioned on exogenous inputs, and captures the probabilistic nature of dynamic systems. This modular and scalable framework enables efficient learning of Koopman representations across diverse systems. Ablation studies confirm the necessity of each component. Experiments on simulated and real-world data demonstrate resilience to missing information, infrequent measurements, and noise, highlighting its potential for constructing structurally consistent, accurate models of real-world phenomena.

1 INTRODUCTION

Nonlinear dynamical systems are widespread across natural and engineered domains, but their complexity poses persistent challenges for modeling, analysis, and control. The Koopman operator offers a principled means of connecting nonlinear and linear perspectives by representing nonlinear evolution in an observable space with linear dynamics. In practice, however, Koopman-based approaches encounter three structural modeling challenges: theoretical non-invertibility, input non-closure and distributional non-identifiability. These limitations motivate a structurally consistent framework that simultaneously overcomes all three while preserving accuracy and physical interpretability.

The Koopman operator provides a global linear representation of nonlinear dynamics by embedding them in a higher-dimensional function space (Koopman, 1931). In contrast to local linearization near equilibria, this formulation enables spectral analysis in which eigenvalues and eigenfunctions describe stability, periodicity, and long-term behavior (Budišić et al., 2012; Surana, 2016; Giannakis, 2019; Klus et al., 2020). Since its introduction, the theory has been extensively advanced (Mezić, 2005; Rowley et al., 2009; Mezić, 2013; Mauroy et al., 2020; Brunton et al., 2021). Practical computation remains difficult because it requires expressive observables and finite-dimensional approximations of the infinite operator. To address this challenge, a variety of numerical methods have been proposed, such as dynamic mode decomposition (DMD) and its variants (Schmid, 2010; Tu, 2013; Williams et al., 2015; Klus et al., 2015; Proctor et al., 2016; Takeishi et al., 2017b; Mauroy & Goncalves, 2019; Colbrook et al., 2023) , many of which have demonstrated empirical success (Arbabi & Mezic, 2017; Korda & Mezić, 2018b).

Despite recent advances, Koopman-based modeling remains constrained by three structural challenges that limit precision and interpretability. First, when the observable is not bijective (*theoretical non-invertibility*), the latent representation cannot recover the underlying dynamics, which weakens prediction and undermines physically meaningful modal analysis. Second, *input non-closure*: modeling inputs by operator decomposition breaks functional closure and invalidates spectral analysis; simple concatenation with measurements reduces expressivity under fixed model complexity, and this limitation intensifies as input dimension grows. Third, *distributional non-identifiability*: de-

terministic mappings or fixed uncertainty models cannot represent the true conditional law on the measurement space, blurring intrinsic dynamics and stochastic effects. These issues are pervasive in real-world systems, and no existing approach resolves all three within a single framework, limiting the practical reach of Koopman modeling for complex systems.

Invertible neural networks have been introduced for Koopman learning, enabling bijective observables with exact reconstruction and thus more faithful identification of invariant subspaces (Jin et al., 2023; Meng et al., 2024; Jin et al., 2024; Hou et al., 2024). However, current applications remain mainly limited to autonomous systems. More generally, these architectures belong to the family of normalizing flows, which are designed for bijective, conditional, and probabilistic mappings (Behrmann et al., 2019; Winkler et al., 2019; Kobyzev et al., 2020; Papamakarios et al., 2021), making normalizing flows a natural foundation to address the central challenges of Koopman modeling.

Building on the strengths of normalizing flows, we propose a structurally consistent Koopman framework that constructs globally linear representations through invertible transformations. This guarantees exact reconstruction while automatically identifying observables, avoiding the problem-specific designs required by traditional approaches. Our method integrates an invertible feature extractor with a normalizing-flow component in a modular manner, allowing flexible adaptation across diverse dynamical systems. The design ensures structurally consistent modeling and simultaneously resolves the challenges of theoretical non-invertibility, input non-closure, and distributional non-identifiability.

2 Method: Koopman Normalizing Invertible Flow Framework

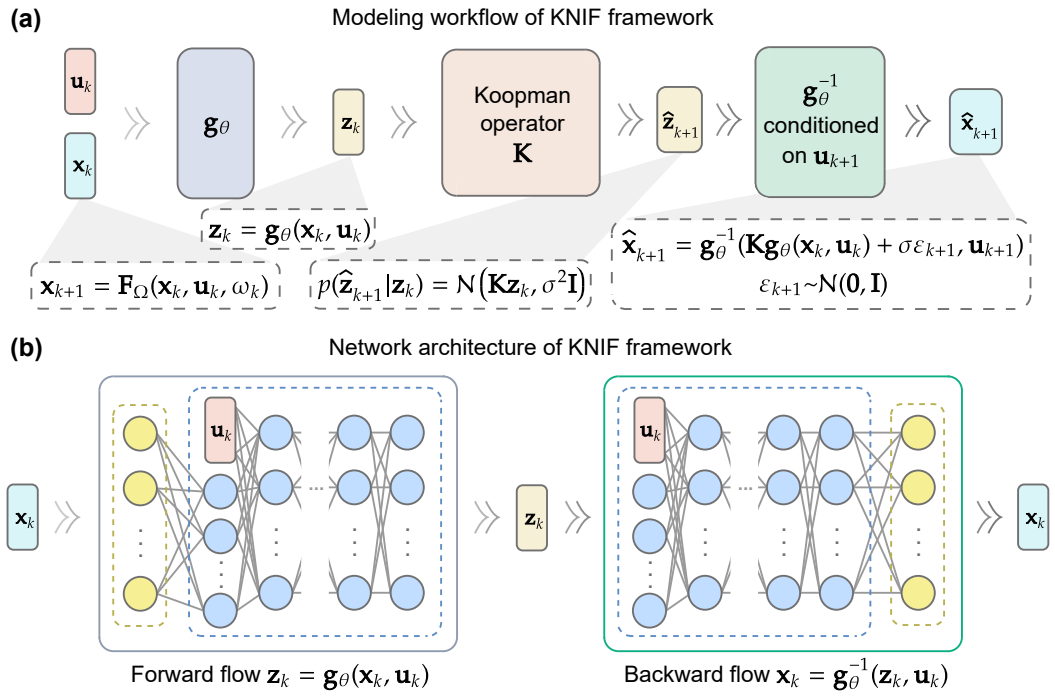

Figure 1: Overview of the KNIF framework. (a), The modeling workflow of KNIF framework, where the Koopman operator only presents the single-step recursive process to maintain consistency in the mathematical form of the modeling. (b), The neural network architecture of KNIF framework, where the yellow part represents the invertible extension module and the blue part represents the normalizing flow module.

2.1 MODELING

In real-world applications, dynamical systems are shaped by intrinsic dynamics, external inputs, and random disturbances; accordingly, we include input and stochasticity and write the nonlinear dynamics as follows (Takeishi et al., 2017b)

$$\mathbf{x}_{k+1} = \mathbf{F}_\Omega(\mathbf{x}_k, \mathbf{u}_k, \omega_k), \tag{1}$$

where the measurement $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m$, the exogenous input $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^p$, the random disturbance $\omega \in \Omega$, and $\mathbf{F}_\Omega$ is a nonlinear mapping. Here, the exogenous input $\mathbf{u}$ denotes externally specified setpoints/commands or environmental drivers and is treated as given, while ω collects all randomness (process and measurement noise); we take $\mathbb{E}_\Omega[\cdot]$ over ω. In this case, the linear Koopman operator $\mathcal{K}_\Omega$ is defined on the observable function g as

$$\mathcal{K}_\Omega g(\mathbf{x}_k, \mathbf{u}_k) := \mathbb{E}_\Omega[g(\mathbf{F}_\Omega(\mathbf{x}_k, \mathbf{u}_k, \omega_k), \mathbf{u}_{k+1})], \tag{2}$$

where $\mathbb{E}_\Omega[\cdot]$ denotes the expectation in the sample space Ω. The Koopman operator is typically infinite-dimensional but can be numerically approximated by a finite-dimensional matrix $\mathbf{K}$ based on a vector-valued observable function $\mathbf{g}$, and the spectral characteristics of $\mathcal{K}_\Omega$ can be approximated by computing the eigenvalues and eigenvectors of $\mathbf{K}$. Then challenges can be described as

Definition 1 (Theoretical invertibility/non-invertibility for $\mathbf{x}$ given $\mathbf{u}$)**.** *For each $\mathbf{u}$ with $\mathcal{S}_\mathbf{u} := \{\mathbf{x} : (\mathbf{x}, \mathbf{u}) \in \mathcal{S} \subseteq \mathcal{X} \times \mathcal{U}\} \neq \emptyset$, define the section $\mathbf{g}_\mathbf{u} : \mathcal{S}_\mathbf{u} \to Z_\mathbf{u}$ by $\mathbf{g}_\mathbf{u}(\mathbf{x}) := \mathbf{g}(\mathbf{x}, \mathbf{u})$, where $Z_\mathbf{u} := \{\mathbf{g}(\mathbf{x}, \mathbf{u}) : (\mathbf{x}, \mathbf{u}) \in \mathcal{S}\}$. We say $\mathbf{g}$ is* invertible for $\mathbf{x}$ given $\mathbf{u}$ on the support *if $\mathbf{g}_\mathbf{u}$ is a bijection onto $Z_\mathbf{u}$ and there exists a conditional back-mapping $\mathbf{r}_\mathbf{u} : Z_\mathbf{u} \to \mathcal{S}_\mathbf{u}$ such that*

$$\mathbf{r}_\mathbf{u}\big(\mathbf{g}_\mathbf{u}(\mathbf{x})\big) = \mathbf{x} \quad \textit{and} \quad \mathbf{g}_\mathbf{u}\big(\mathbf{r}_\mathbf{u}(\mathbf{z})\big) = \mathbf{z} \quad \textit{for all } \mathbf{x} \in \mathcal{S}_\mathbf{u},\ \mathbf{z} \in Z_\mathbf{u}.$$

Theoretical non-invertibility (for $\mathbf{x}$ given $\mathbf{u}$) *holds when the above fails for some $\mathbf{u}$.*

Definition 2 (Input non-closure)**.** *The model is* closed *under the Koopman update iff there exists a constant linear operator $\mathbf{K}$ such that equation (2) holds for all trajectories on the support (with observable $\mathbf{g}(\mathbf{x}, \mathbf{u})$). Otherwise it is* not closed*.*

Definition 3 (Distributional non-identifiability on the measurement space)**.** *Within the chosen modeling family, there exists no parameter θ such that the modeled conditional $q_\theta(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$ coincides on the data support with the true conditional law on the measurement space induced by equation (2), i.e. $\mathrm{Law}_\Omega(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$.*

Remark 1 (Minimal illustrations of three Koopman modeling challenges)**.** *(i) **Theoretical non-invertibility:** $\mathbf{g}(\mathbf{x}, \mathbf{u}) = \|\mathbf{x}\|_2^2$ maps $\mathbf{x}$ and $-\mathbf{x}$ to the same value, so no global conditional inverse $r_\mathbf{u}(\mathbf{z})$ recovers $\mathbf{x}$ everywhere on $\mathcal{S}_\mathbf{u}$.*
*(ii) **Input non-closure:** Let $\boldsymbol{\psi}(\mathbf{x})$ be a dictionary and express the Dynamic Mode Decomposition with control (DMDc, (Proctor et al., 2016)) form as $\boldsymbol{\psi}(\mathbf{x}_{k+1}) \approx \mathbf{A}\,\boldsymbol{\psi}(\mathbf{x}_k) + \mathbf{B}\,\mathbf{u}_k$ with $\mathbf{z}_k = \boldsymbol{\psi}(\mathbf{x}_k)$. Then $\mathbf{z}_{k+1} \approx \mathbf{A}\mathbf{z}_k + \mathbf{B}\mathbf{u}_k$ cannot be written as a single $\mathbf{K}\,\mathbf{z}_k$. DMDc is a prototypical operator-splitting formulation; henceforth we focus on single-operator, input-closed methods.*
*(iii) **Distributional non-identifiability:** If the modeling family cannot represent the true conditional law $\mathrm{Law}_\Omega(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$ (e.g., deterministic or fixed-form under non-Gaussian/multimodal effects), then $\nexists\, \theta$ such that $q_\theta(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k) = \mathrm{Law}_\Omega(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$ on the support.*

To address the three structural challenges, we use normalizing flows as Koopman-consistent observables. Normalizing flows are generative models that map a simple base distribution to a complex one via differentiable and bijective transformations. Formally, we set $\mathbf{z} = \mathbf{g}(\mathbf{x}, \mathbf{u})$, where $\mathbf{g}$ is bijective in $\mathbf{x}$ and is conditioned on $\mathbf{u}$, allowing exact recovery $\mathbf{x} = \mathbf{g}^{-1}(\mathbf{z}, \mathbf{u})$ and density estimation

$$p(\mathbf{x} \mid \mathbf{u}) = p(\mathbf{z} \mid \mathbf{u}) \left| \det \left(\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right) \right|, \tag{3}$$

where $\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}$ is the Jacobian matrix of $\mathbf{g}$, and $\det$ denotes the determinant of the Jacobian.

By adopting a normalizing flow as the observable function, we can obtain

Proposition 1 (Time-invariant conditional linear evolution)**.** *Let $\mathbf{z}_k := \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)$ be an observable that is bijective in $\mathbf{x}$ for each fixed $\mathbf{u}$. Assume* (i) *$\mathbf{u}_k$ is non-anticipative and depends only on $(\mathbf{x}_{0:k}, \mathbf{u}_{0:k-1})$, not on future $(\mathbf{x}_{k+1:\infty}, \omega_{k:\infty})$,* (ii) *$(\mathbf{x}_k, \mathbf{u}_k)$ is first-order Markov, and* (iii) *the transition kernel $\mathbb{P}(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{u}_k)$ is time-invariant on the data window. Then there exists a constant matrix $\mathbf{K}$ such that*

$$\mathbb{E}_\Omega[\mathbf{z}_{k+1} \mid \mathbf{z}_k] = \mathbf{K}\,\mathbf{z}_k.$$

Normalizing flows naturally fulfill the structural requirements of Koopman modeling: invertibility ensures accurate recovery; conditional input preserves operator closure without additional augmentation; and flexible distribution modeling allows non-Gaussian uncertainty capture. Building on these, we propose the Koopman Normalizing Invertible Flow (KNIF) framework, which integrates modular components to realize these theoretical advantages in practical modeling scenarios.

Given the measurement data $\{\tilde{\mathbf{x}}_i\}$ and the exogenous input data $\{\tilde{\mathbf{u}}_i\}$ directly measured from the system, the KNIF framework models the dynamics by lifting the original measurement into an observable space through a forward flow $\mathbf{g}_\theta$ (Fig. 1b), where θ denotes the shared parameters of the model. The forward flow consists of two functionally integrated modules: the invertible extension module, which lifts the original measurement $\tilde{\mathbf{x}}$ into a higher-dimensional space with spectral expressiveness and a guaranteed reconstruction, and the normalizing flow module, which models conditional input and enables density transformation into a tractable Gaussian form. Since the normalizing flow maintains the same input-output dimensionality by construction, the invertible extension module plays a critical role in dimension expansion and enhancing the spectral expressiveness of the Koopman operator.

The Koopman operator is approximated in the lifted space using numerical methods such as Ordinary Least Squares (OLS), Dynamic Mode Decomposition (DMD (Schmid, 2010)), or Identification of Hybrid Dynamical Systems (IHYDE (Yuan et al., 2019)), leveraging the structured latent space generated by the KNIF architecture to ensure that the learned operator captures the globally linear evolution under complex nonlinear dynamics. The evolution of observables is computed as

$$\hat{\mathbf{g}}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \mathbf{K}^t \mathbf{g}_\theta(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0), \tag{4}$$

where $\mathbf{K}^t$ denotes t successive applications of $\mathbf{K}$ (i.e., t-step propagation / iteration), and the approximation error arises from the linearization (Fig. 1a).

Sampling is performed from a zero-centered Gaussian distribution shifted by the observable space mean. The Koopman-predicted mean and the Gaussian standard deviation are jointly used to form a unified latent representation, which is then mapped back to the original measurement space through the inverse flow $\mathbf{g}_\theta^{-1}$, thereby achieving the reconstruction of the original system (Fig. 1a). This reconstruction mechanism completes the closed-loop modeling pipeline, ensuring consistency between learned dynamics and original measurements.

2.2 Network Structure

The invertible extension module $\mathbf{g}_{\theta_1}$ and the normalizing flow module $\mathbf{g}_{\theta_2}$ form a cohesive observable function $\mathbf{g}_\theta = \mathbf{g}_{\theta_2} \circ \mathbf{g}_{\theta_1}$, jointly encoding dynamics and uncertainties in a Koopman-consistent manner. This modular composition yields an expressive observable function that jointly supports invertibility, input sensitivity, and density modeling, providing a coherent foundation for Koopman-based representations under complex structural conditions. This subsection introduces the implementation of two modules.

2.2.1 Structure of the Invertible Extension Module

Any theoretical invertible module can be used here, and we introduce a Chebyshev layer to extend the dimension, which can leverage its superior function approximation capability and reduce the number of parameters (Xu et al., 2024). The Chebyshev feature function $\mathbf{\Phi}_{\theta_1}$ is formulated as

$$\mathbf{\Phi}_{\theta_1}(\mathbf{x}) = \cos(\mathbf{W}_{\theta_1} \arccos(\mathbf{x})), \tag{5}$$

where $\mathbf{x} \in [-1, 1]^m$ and $\mathbf{W}_{\theta_1} \in \mathbb{R}^{d \times m}$ is a fully connected (FC) layer with parameter θ_1 and without bias.

Proposition 2 (Recoverability of the invertible extension)**.** *If $\mathbf{W}_{\theta_1}$ includes m standard-basis rows, then the Chebyshev feature $\mathbf{\Phi}_{\theta_1}(\mathbf{x})$ explicitly contains the raw state $\mathbf{x}$. Let F_θ be any bijective module acting on $\mathbf{\Phi}_{\theta_1}(\mathbf{x})$, and define $\mathbf{y} = F_\theta(\cdot)$. Then $\mathbf{x}$ can be exactly recovered from $\mathbf{z}$ by*

$$\mathbf{x} = \Pi_{\mathbf{x}}\big(F_\theta^{-1}(\mathbf{y})\big),$$

where $\Pi_{\mathbf{x}}$ selects the coordinates of $\mathbf{x}$ embedded in $\mathbf{\Phi}_{\theta_1}(\mathbf{x})$.

The domain of Chebyshev features function is $[-1, 1]^m$, therefore, a normalization layer $\mathbf{\Phi}_{\text{norm}}$ is used along with the Chebyshev layer with its inverse $\mathbf{\Phi}_{\text{norm}}^{-1}$. Therefore, the whole module consists of $\mathbf{g}_{\theta_1} = \mathbf{\Phi}_{\theta_1} \circ \mathbf{\Phi}_{\text{norm}}$ with its inverse $\mathbf{g}_{\theta_1}^{-1} = \mathbf{\Phi}_{\text{norm}}^{-1} \circ \mathbf{\Phi}_{\theta_1}^{-1}$.

2.2.2 STRUCTURE OF THE NORMALIZING FLOW MODULE

The normalizing flow module is composed of multiple normalizing flow blocks stacked together. Several kinds of normalizing flows are introduced in (Kobyzev et al., 2020), and here we use the residual flow (Behrmann et al., 2019) for its convenience and ability to model complex distributions. The input $\mathbf{u}$ in the system (1) is considered a condition, and the formulation of the residual flow is

$$\boldsymbol{\psi}_i(\mathbf{y}) = \mathbf{y} + \boldsymbol{\phi}_{\theta_{2i}}(\mathbf{y}, \mathbf{u}), \tag{6}$$

where $i = 1, 2, \cdots, L$, L is the number of blocks and $\boldsymbol{\phi}_{\theta_{2i}}$ is a multilayer perceptron (MLP) with parameter θ_{2i}. If the Lipschitz constant of $\boldsymbol{\phi}_{\theta_{2i}}$ is less than 1, then the residual flow is theoretically invertible. This can be realized by setting the activation function as ReLU etc. and restricting the 2-norm of FC layers to less than 1 (Behrmann et al., 2019). Furthermore, we apply spectral normalization to every linear layer in the residual flow. The inverse of residual flow can be computed by a fixed-point iteration in this case, which is described in Algorithm 1.

Algorithm 1 Inverse of residual flow (Behrmann et al., 2019)

1: **Input:** output from residual flow $\mathbf{z}$, condition $\mathbf{u}$, contractive MLP $\boldsymbol{\phi}_{\theta_{2i}}$, number of fixed-point iterations k
2: **Init:** $\mathbf{y}^0 \leftarrow \mathbf{z}$
3: **for** $j = 0, \ldots, k$ **do**
4: $\quad \mathbf{y}^{j+1} \leftarrow \mathbf{z} - \boldsymbol{\phi}_{\theta_{2i}}(\mathbf{y}^j, \mathbf{u})$
5: **end for**

The Jacobian of $\boldsymbol{\psi}_i$ is equal to $\mathbf{I} + \frac{\partial \boldsymbol{\phi}_{\theta_{2i}}(\mathbf{y})}{\partial \mathbf{y}}$. We can compute the Jacobian eigenvalues of $\boldsymbol{\phi}_{\theta_{2i}}$ as λ_{ij}, $j = 1, 2, \cdots, d$, then the Log-Determinant of the Jacobian (LDJ) of $\boldsymbol{\psi}_i$ can be formulated as

$$\sum_{j=1}^{d} \log(1 + \lambda_{ij}) \tag{7}$$

Finally, the normalizing flow module using residual flows can be represented as $\mathbf{g}_{\theta_2} = \boldsymbol{\psi}_L \circ \boldsymbol{\psi}_{L-1} \circ \cdots \circ \boldsymbol{\psi}_1$ with inverse $\mathbf{g}_{\theta_2}^{-1} = \boldsymbol{\psi}_1^{-1} \circ \boldsymbol{\psi}_2^{-1} \circ \cdots \circ \boldsymbol{\psi}_L^{-1}$, and its LDJ can be calculated as $\sum_{i=1}^{L} \sum_{j=1}^{d} \log(1 + \lambda_{ij})$. Equation (7) is provided for intuition; in implementation our tasks are non-image with observable dimension $d \leqslant 50$. For each residual block, we explicitly form the Jacobian and compute the exact log-determinant through the LU decomposition (`slogdet`). The cost is $O(d^3)$ per block and negligible at this scale.

2.3 TRAINING

We model the observable as Gaussian with its *mean* given by the Koopman prediction (defined via Proposition 1) and its *variance* estimated under an isotropic Gaussian assumption, i.e.,

$$\mathbf{g}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) \;\sim\; \mathcal{N}\big(\,\hat{\mathbf{g}}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t),\; \sigma^2 \mathbf{I}\,\big).$$

Combining the mean-fitting and density terms, we seek $\mathbf{g}_{\theta^*}$ by (proven in Appendix Section A.2)

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L} = \mathcal{L}_{\text{lin}} + \lambda\, \mathcal{L}_{\text{reg}}, \\ & \mathcal{L}_{\text{lin}} = \sum_{t=1}^{n} \left\| \mathbf{g}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) - \hat{\mathbf{g}}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) \right\|^2, \\ & \mathcal{L}_{\text{reg}} = -\sum_{t=1}^{n} \log \left| \det \frac{\partial \mathbf{g}_{\theta_2}(\mathbf{y}, \tilde{\mathbf{u}}_t)}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\tilde{\mathbf{y}}_t} \right|, \end{aligned} \tag{8}$$

where $\mathbf{y} = \mathbf{g}_{\theta_1}(\mathbf{x})$ and θ^* solves equation 8. Under the Gaussian model above, maximum-likelihood scaling yields $\lambda = 2\sigma^2$. During training, $\mathbf{K}$ is synchronously updated via DMD.

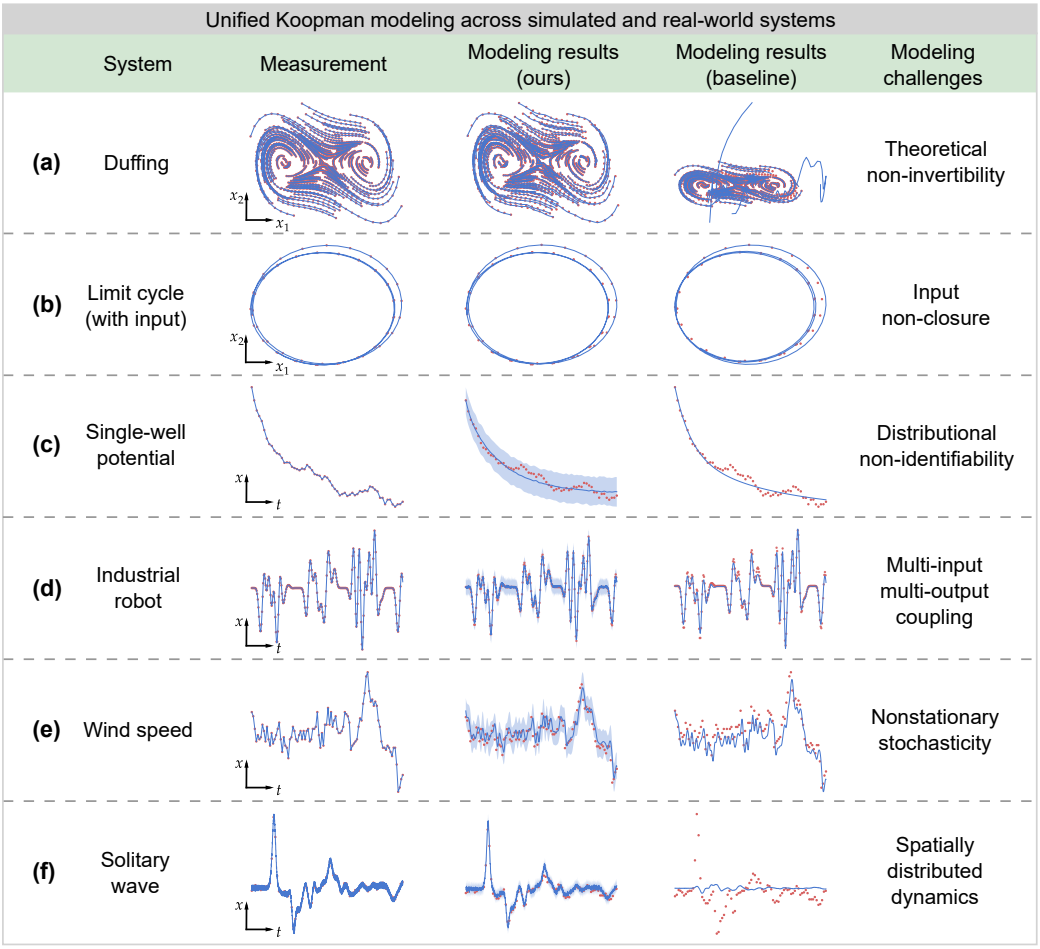

Figure 2: Modeling different types of systems. (a–c), Simulated systems representing autonomous, system with input, and stochastic systems. (d–f), Real-world systems used to validate universal applicability. Blue solid lines show the mean, blue shaded regions the 95% confidence interval, and red dots represent the measurement points. Baseline: methods discussed in Subsection 3.1.3.

3 Experiments

3.1 Modeling Nonlinear Systems with Different Dynamical Properties

We first evaluate the KNIF framework on six representative systems, divided into two groups: three simulated systems designed to highlight key Koopman structural challenges, and three real-world systems selected to demonstrate universal applicability. In Koopman-based modeling, all cases involve the fundamental problem of learning invertible observables, while each class additionally presents specific challenges.

3.1.1 Simulated Systems

Simulated systems include the Duffing system, the limit cycle system with exponential input, and the single-well potential system (Fig.2a–c). The Duffing system is a highly nonlinear autonomous system in which small modeling errors quickly lead to trajectory divergence, making it especially sensitive to theoretical non-invertibility. In baseline methods, non-bijective observables collapse distinct states and the absence of a proper inverse introduces ambiguity during reconstruction, producing divergent predictions. KNIF avoids this by using an invertible flow that preserves state distinctions and enables precise recovery (Fig.2a).

The limit cycle system has a time-dependent input, which raises the input non-closure challenge. Its time-varying input prevents a time-invariant, input-closed Koopman operator: Approaches such as DMDc fail to yield a complete Koopman operator, deterministic basis expansions suffer from combinatorial growth (Appendix equation (23)), and other neural observables with the same dimension are less expressive. KNIF directly conditions the flow on the input, enabling closed operator construction (Fig. 2b).

The single-well potential system is driven by high-intensity Gaussian process noise, creating distributional non-identifiability. Baseline models capture only deterministic mean trajectories and fail to represent uncertainty. KNIF transforms the complex latent distribution into a Gaussian, allowing both mean and variance to be modeled linearly and reconstructed probabilistically (Fig. 2c).

As shown in Fig. 2a–c, baseline Koopman methods fail in each case, while KNIF successfully reconstructs the trajectories and captures the underlying dynamics.

3.1.2 Real-world Systems

Real-world systems include the industrial robot, the wind speed, and the solitary wave system (Fig. 2d–f). These examples validate the broad applicability of KNIF to high-dimensional and noisy physical systems, where standard Koopman methods fail due to non-invertibility and distributional non-identifiability, with the industrial robot further exhibiting input non-closure.

The industrial robot system features multi-input multi-output coupling and strong mechanical nonlinearities, making it an ideal testbed. We use the Industrial Robot Identification Benchmark dataset (Weigand et al., 2023), generated from a 6-DOF robot under torque control. This dataset captures real joint dynamics affected by backlash, nonlinear friction, and hydraulic compensation, recorded with physical sensors and controllers. Baseline methods show large errors near abrupt transitions, while KNIF maintains accurate predictions and probabilistic estimation (Fig. 2d).

The wind speed system presents nonstationary stochasticity, where fluctuating spatiotemporal patterns introduce uncertainty. Wind is a key renewable energy source but difficult to model due to turbulence. We use data from (of Water & Limited, 2021; Wang et al., 2022). Baselines deviate severely under strong stochasticity, whereas KNIF achieves accurate reconstruction and uncertainty capture (Fig. 2e).

The solitary wave system exhibits spatially distributed dynamics governed by partial differential equations. We use the Laboratory Conical Island dataset from NOAA (for Tsunami Research, 1995), which contains controlled measurements of wave run-up and propagation around a conical island, recorded at eight spatial measurement points. This dataset reflects nonlinear wave–bathymetry interactions and serves as a tsunami modeling benchmark. Baselines fail to capture the spatiotemporal dynamics, while KNIF infers continuous behavior from sparse measurements (Fig. 2f).

Although the underlying systems exhibit significant nonlinearities and stochastic behaviors that cause baseline methods to break down, KNIF accurately models each system as shown in Fig. 2d-f, thus demonstrating its strong generalizability across complex dynamical regimes.

3.1.3 Benchmarks

We further compare KNIF with state-of-the-art Koopman approaches grouped as autoencoder-based (LKIS (Takeishi et al., 2017a), DeepKoopman (Lusch et al., 2018)), dictionary-learning-based (EDMD-DL (Li et al., 2017)), and invertible network-based (IKN (Jin et al., 2023), FlowDMD (Meng et al., 2024)), plus the non-Koopman baseline Neural ODE (NODE (Chen et al., 2018b)). For fairness, we primarily compare against methods with a globally linear closed Koopman operator; DeepKoopman is the sole exception, included to illustrate the structural benefit of the normalizing flow observable. We report results only for DeepKoopman, EDMD-DL, FlowDMD, and NODE, since these represent the most widely used baselines; a qualitative analysis of other approaches is provided in the Appendix Table 2. Parameter counts are matched across methods, and test set reconstruction MSEs are reported. Datasets include Duffing, Van der Pol, a limit cycle with exponential input, and a nonpolynomial vector field, as well as two real benchmarks: the SISO Electro-Mechanical Positioning System (EMPS (Janot et al., 2019)), which exhibits asymmetric friction and integrator effects, and the aforementioned industrial robot with MIMO dynamics under torque control. The results, summarized in Table 1, demonstrate that FlowDMD fails on strongly nonlin-

Dataset	**DeepKoopman**	**EDMD-DL**	**FlowDMD**	**NODE**	**KNIF (ours)**
Duffing	1.053	1.865×10^{-2}	–	6.533×10^{-2}	$\mathbf{7.673 \times 10^{-5}}$
Van der Pol	5.855	8.334×10^{-2}	–	7.383×10^{-2}	$\mathbf{1.636 \times 10^{-3}}$
Limit cycle	–	4.770×10^{-2}	2.657×10^{-3}	5.708×10^{-2}	$\mathbf{2.280 \times 10^{-4}}$
Nonpolynomial	–	5.867×10^{-3}	3.028×10^{-3}	8.985×10^{-3}	$\mathbf{6.523 \times 10^{-4}}$
EMPS	4.444×10^{-2}	7.476×10^{-6}	4.161×10^{-8}	2.346×10^{-5}	$\mathbf{4.548 \times 10^{-9}}$
Industrial robot	4.463×10^{2}	5.971	1.180×10^{-2}	2.455	$\mathbf{4.663 \times 10^{-3}}$
Average rank	4.667	3.000	3.000	3.333	**1.000**

Table 1: Comparison of test set reconstruction Mean Squared Errors (MSEs) across different methods on simulated and real systems. For methods with abnormal numerical results, they are ranked last in the corresponding datasets. Here, "–" indicates that no valid result was obtained. *Protocol:* capacity/budget matched; baselines tuned from authors' recommended ranges. Long-horizon rollouts accumulate small per-step errors, explaining the large MSE gaps.

ear Duffing and Van der Pol systems but is competitive elsewhere due to its invertibility. NODE performs similarly to EDMD-DL, reflecting the limited expressiveness of dictionary learning approaches. DeepKoopman performs worst due to the three structural issues and overfitting problem of autoencoder (Kingma et al., 2013), with its non-invertibility causing state collapse and its inability to handle time-varying inputs leading to divergent predictions. KNIF achieves the best accuracy across all benchmarks by jointly addressing non-invertibility, input non-closure, and distributional non-identifiability.

3.1.4 Koopman Modal Analysis

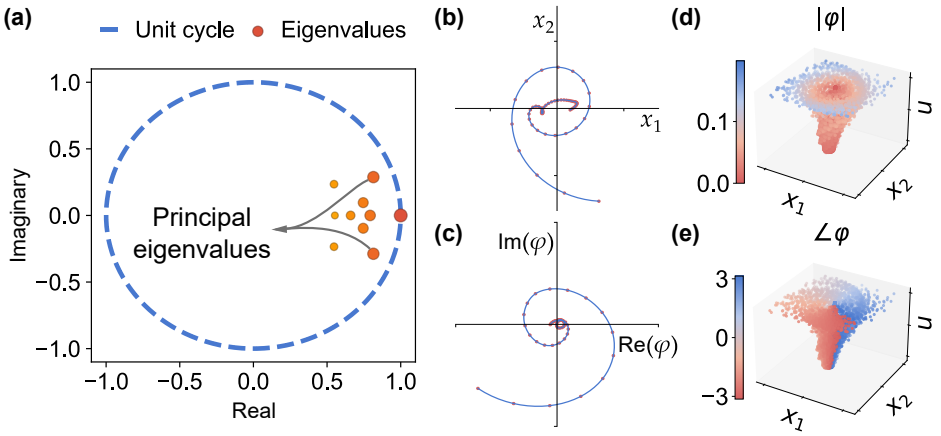

Figure 3: Koopman spectral analysis of simulated and real-world systems. (a–e), Analysis of the nonpolynomial vector field system: (a), Koopman spectrum from the KNIF framework; (b, c), A representative trajectory in phase space and its principal eigenfunction in the complex plane; (d, e), Amplitudes and phases of the principal eigenfunction across all trajectories.

Finally, we show that KNIF achieves superior modeling accuracy while preserving meaningful spectral interpretations. On the nonpolynomial vector field system, we retrain with Laplacian measurement noise to test whether interpretable Koopman representations persist under simultaneous structural challenges. The learned Koopman spectrum (Fig. 3a) remains informative despite observation noise, including the trivial eigenvalue $\lambda_0 = 1$ and a principal pair $\lambda_{1,2} = 0.8145 \pm 0.2878j$ cap-

turing the dominant dynamics (Mauroy & Mezić, 2016). Since similarity transforms preserve the spectrum and modal coordinates, the reported content is not representation-dependent. The principal pair yields conjugate eigenfunctions; by symmetry we analyze one. Under a cosine input, a representative phase–plane trajectory rotates around the equilibrium at $(0, 0)$ (Fig. 3b), and the principal eigenfunction maps this rotation to the complex plane with the equilibrium at $0 + 0j$ (Fig. 3c). Aggregated magnitude and phase visualizations reveal contraction (Fig. 3d) and coherent rotation (Fig. 3e), indicating that KNIF extracts interpretable structure from noisy and complex systems.

3.2 Modeling Capability under Structural and Practical Challenges

The KNIF framework adopts a modular design that addresses three key structural challenges in Koopman modeling: non-invertibility, input non-closure, and distributional non-identifiability. Real-world systems further introduce practical difficulties such as low sampling rates, noise, and partial observability. To evaluate both aspects, we design two experiments: an ablation study isolating non-invertibility and input non-closure to validate KNIF's components, and a robustness study assessing resilience to practical issues and distributional uncertainty. Together, these results confirm that KNIF improves both accuracy and interpretability under structural and practical challenges.

3.2.1 Ablation

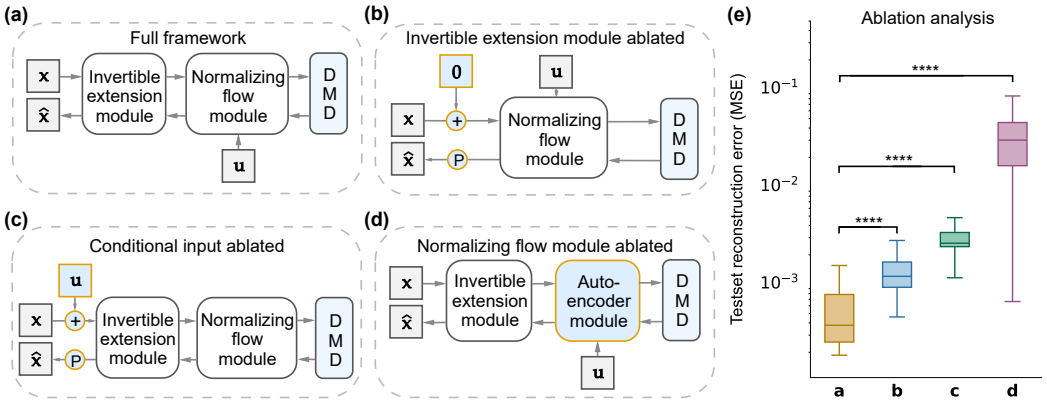

Figure 4: Ablation study of the KNIF framework. (a–d) show the full model (a) and three variants (b-d). Highlighted areas mark modifications, where "+" denotes concatenation and "P" projection to the measurement space. (e) shows reconstruction errors on the nonpolynomial system with Mann–Whitney U tests. Asterisks indicate statistical significance ($p < 10^{-n}$).

To verify that each component of KNIF is necessary, we conduct ablation experiments by replacing key modules (Fig. 4a–d). Three simplified variants are considered: (b) removing the invertible extension and applying zero padding, (c) excluding conditional input and concatenating it with the measurement (as in DeepKoopman, EDMD-DL, and FlowDMD), and (d) replacing the normalizing flow with a standard autoencoder of comparable size.

As shown in Fig. 4e, the full framework achieves the lowest reconstruction error. Removing the invertible extension moderately increases error, removing conditional input causes further degradation, and replacing the normalizing flow yields the largest drop, confirming the importance of explicitly modeling the input dependency and bijective transformations for capturing nonlinear dynamics.

We also perform the Mann–Whitney U test (Gao & Yan, 2022) between the full framework and each variant, all yielding $p < 10^{-4}$, which verifies that the observed improvements are statistically significant and that each component contributes meaningfully to the framework's effectiveness.

3.2.2 Robustness

To evaluate robustness, we study the Kraichnan–Orszag (KO) system with cyclic symmetry and fully coupled nonlinear dynamics. Three experiments test low sampling rates, strong observation noise, and partial observability.

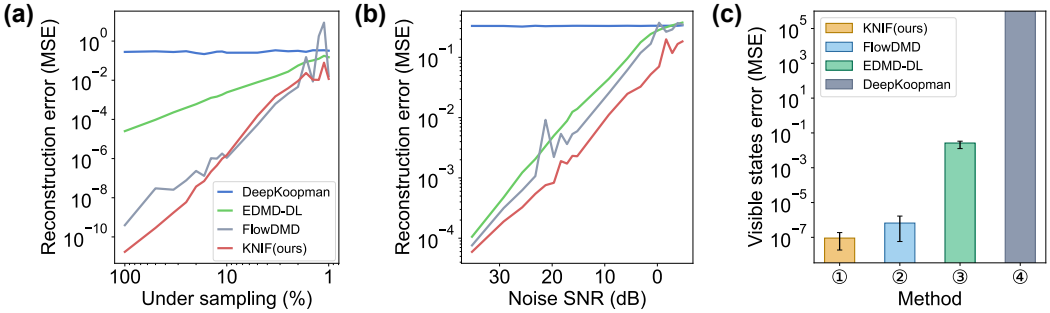

Figure 5: Robustness experiments in challenging cases. (a-c), Comparison of reconstruction errors on the test set under different sampling rates, noise strengths and three partially observable settings.

For low sampling and noise, KNIF is compared with EDMD-DL, DeepKoopman, and FlowDMD using only five observables. With sufficient data (sampling rate $> 10\%$), KNIF clearly outperforms baselines, remains competitive in low-data regimes, and under strong noise achieves about an order of magnitude higher accuracy. Importantly, it sustains this advantage even when noise exceeds 0 dB, showing its ability to disentangle signal from noise and handle distributional non-identifiability.

For partial observability, we test cases in which only two of the three KO states are measured. To recover the full dynamics, we use a unified delay-embedding front-end shared by all methods and details are deferred to the Appendix Section A.4. As shown in Fig. 5c, KNIF achieves significantly better reconstruction accuracy on the visible states compared to baselines.

Overall, these results demonstrate that KNIF is robust to limited data, high noise, and partial observability, confirming its effectiveness under both structural and practical challenges.

4 Discussion

Real-world dynamical systems are shaped by nonlinear interactions, external forcing, and stochastic disturbances, leading to measurements that are noisy, incomplete, and often too intricate to model explicitly. We propose a unified, invertible Koopman framework that learns globally linear representations directly from raw measurements. To our knowledge, no approaches address the combined structural challenges of theoretical non-invertibility, input non-closure, and distributional non-identifiability within a single architecture while remaining practical for real systems. This design supports structurally consistent, robust, and interpretable modeling and offers a strong baseline for real-world analysis.

Our implementation is intentionally simple to emphasize scalability and modularity, but it admits richer components when needed. In the invertible extension, more expressive feature extractors can be concatenated with measurements, for example Neural Basis Expansion (NBE (Oreshkin et al., 2019)), Sinusoidal Representation Networks (SIREN (Sitzmann et al., 2020)), and Kolmogorov–Arnold Networks (KAN (Liu et al., 2024)). In the flow module, autoregressive flows (AR (Kingma et al., 2016)), neural ordinary differential equations (NODE (Chen et al., 2018b)), and Langevin flows (Chen et al., 2018a) may further increase flexibility. For Koopman estimation we default to exact dynamic mode decomposition (exact DMD) for clarity, and sparsity promoting DMD (SpDMD (Jovanović et al., 2014)), optimized DMD (optDMD (Askham & Kutz, 2018)), and residual DMD (ResDMD (Colbrook et al., 2023)) are compatible alternatives. A brief investigation of these variants appears in (Appendix Section A.10).

Several extensions merit study. The current loss is theory driven without empirical tuning, and adding terms such as higher-order derivative regularization may improve temporal smoothness. Moreover, finite-dimensional global linear fits can be inadequate for some systems; time-varying Koopman operators (Guého et al., 2021) and techniques tailored to continuous spectra (Lusch et al., 2018) are promising next steps, which trade interpretability for engineering performance.

LLM USAGE STATEMENT

During manuscript preparation, we used general-purpose large language models (e.g., the GPT family) for grammar polishing, minor rephrasing, and limited code assistance (such as generating boilerplate, suggesting refactorings, and offering debugging hints). We did not rely on LLMs to generate research ideas, methods, experimental designs, analyses, or conclusions. All technical content, experiments, and claims were designed, implemented, and verified by the authors, who take full responsibility for the paper. We did not embed any executable instructions, hidden prompts, or other mechanisms intended to influence the peer-review process in the manuscript or its supplementary materials.

REPRODUCIBILITY STATEMENT

We include a self-contained ZIP artifact in the supplementary materials with executable source code, processed data, pretrained checkpoints, and notebooks to reproduce all main tables/figures or to train from scratch. The Appendix details data acquisition, dataset statistics (value ranges, horizons/lengths, sampling rates, dimensions) and splits, exact network architectures and implementation details, and per-experiment model/training hyperparameters (learning-rate schedules, batch sizes, regularization, epochs). We also provide environment files and adopt deterministic evaluation settings, so a single run with our checkpoints reproduces the reported metrics. During double-blind review, the artifact is shared as supplementary material; upon acceptance, it will be de-anonymized and archived.

REFERENCES

Hassan Arbabi and Igor Mezic. 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.

Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: a python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International conference on machine learning*, pp. 573–582. PMLR, 2019.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern Koopman theory for dynamical systems, 2021. Preprint at `http://arxiv.org/abs/2102.12086`.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. DANCo: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern recognition*, 47(8):2569–2581, 2014.

Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science*, 2(7):433–442, 2022.

Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin Duke. Continuous-time flows for efficient inference and density estimation. In *International Conference on Machine Learning*, pp. 824–833. PMLR, 2018a.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018b.

Matthew J Colbrook, Lorna J Ayton, and Máté Szőke. Residual dynamic mode decomposition: robust and verified Koopmanism. *Journal of Fluid Mechanics*, 955:A21, 2023.

Nicola Demo, Marco Tezzele, and Gianluigi Rozza. PyDMD: Python dynamic mode decomposition. *Journal of Open Source Software*, 3(22):530, 2018.

NOAA Center for Tsunami Research. Solitary wave on a conical island, 1995. URL https://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_ConicalIsland.

Ting-Ting Gao and Gang Yan. Autonomous inference of complex network dynamics from incomplete and noisy data. *Nature Computational Science*, 2(3):160–168, 2022.

Dimitrios Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Applied and Computational Harmonic Analysis*, 47(2):338–396, 2019.

Damien Guého, Puneet Singla, and Manoranjan Majji. Time-Varying Koopman Operator Theory for Nonlinear Systems Prediction. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2294–2299. IEEE, 2021.

Xiao Hou, Jin Zhang, and Le Fang. Invertible neural network combined with dynamic mode decomposition applied to flow field feature extraction and prediction. *Physics of Fluids*, 36(9), 2024.

Sara M Ichinaga, Francesco Andreuzzi, Nicola Demo, Marco Tezzele, Karl Lapo, Gianluigi Rozza, Steven L Brunton, and J Nathan Kutz. PyDMD: A Python package for robust dynamic mode decomposition, 2024. Preprint at http://arxiv.org/abs/2402.07463.

Alexandre Janot, Maxime Gautier, and Mathieu Brunot. Data set and reference models of EMPS. In *Nonlinear system identification benchmarks*, 2019.

Yuhong Jin, Lei Hou, Shun Zhong, Haiming Yi, and Yushu Chen. Invertible Koopman network and its application in data-driven modeling for dynamic systems. *Mechanical Systems and Signal Processing*, 200:110604, 2023.

Yuhong Jin, Lei Hou, and Shun Zhong. Extended dynamic mode decomposition with invertible dictionary learning. *Neural networks*, 173:106177, 2024.

Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2), 2014.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Stefan Klus, Péter Koltai, and Christof Schütte. On the numerical approximation of the Perron-Frobenius and Koopman operator, 2015. Preprint at http://arxiv.org/abs/1512.05997.

Stefan Klus, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, and Christof Schütte. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43 (11):3964–3979, 2020.

Bernard O Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018a.

Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28:687–710, 2018b.

Bian Li, Yian Ma, J Nathan Kutz, and Xiu Yang. The adaptive spectral Koopman method for dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 22(3):1523–1551, 2023.

Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10), 2017.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. KAN: Kolmogorov-Arnold networks, 2024. Preprint at http://arxiv.org/abs/2404.19756.

Peter Y Lu, Joan Ariño Bernad, and Marin Soljačić. Discovering sparse interpretable dynamics from partial observations. *Communications Physics*, 5(1):206, 2022.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Alexandre Mauroy and Jorge Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6):2550–2565, 2019.

Alexandre Mauroy and Igor Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016.

Alexandre Mauroy, Y Susuki, and Igor Mezic. *Koopman operator in systems and control*, volume 484. Springer, Cham, 2020.

Yuhuang Meng, Jianguo Huang, and Yue Qiu. Koopman operator learning using invertible neural networks. *Journal of Computational Physics*, 501:112795, 2024.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

Igor Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual review of fluid mechanics*, 45(1):357–378, 2013.

National Institute of Water and Atmospheric Research Limited. Cliflo: Niwa's national climate database, 2021. URL http://cliflo.niwa.co.nz/.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, 2019. Preprint at http://arxiv.org/abs/1905.10437.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Amit Surana. Koopman operator based observer synthesis for control-affine nonlinear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6492–6499. IEEE, 2016.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017a.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, 96(3):033310, 2017b.

Floris Takens. Dynamical systems and turbulence. *Warwick, 1980*, pp. 366–381, 1981.

Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

Yasen Wang, Huazhen Fang, Junyang Jin, Guijun Ma, Xin He, Xing Dai, Zuogong Yue, Cheng Cheng, Hai-Tao Zhang, Donglin Pu, et al. Data-driven discovery of stochastic differential equations. *Engineering*, 17:244–252, 2022.

Jonas Weigand, Julian Götz, Jonas Ulmen, and Martin Ruskowski. Dataset and baseline for an industrial robot identification benchmark. 2023.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows, 2019. Preprint at `http://arxiv.org/abs/1912.00042`.

Zhongshu Xu, Yuan Chen, and Dongbin Xiu. Chebyshev feature neural network for accurate function approximation, 2024. Preprint at `http://arxiv.org/abs/2409.19135`.

Ye Yuan, Xiuchuan Tang, Wei Zhou, Wei Pan, Xiuting Li, Hai-Tao Zhang, Han Ding, and Jorge Goncalves. Data driven discovery of cyber physical systems. *Nature communications*, 10(1): 4894, 2019.

Ye Yuan, Xiuting Li, Liang Li, Frank J Jiang, Xiuchuan Tang, Fumin Zhang, Jorge Goncalves, Henning U Voss, Han Ding, and Jürgen Kurths. Machine discovery of partial differential equations from spatiotemporal data: A sparse Bayesian learning framework. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(11), 2023.

Zhexuan Zeng, Zuogong Yue, Alexandre Mauroy, Jorge Gonçalves, and Ye Yuan. A sampling theorem for exact identification of continuous-time nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, 2024.

Kevin Zhang and Hod Lipson. Aligning AI-driven discovery with human intuition, 2024. Preprint at `http://arxiv.org/abs/2410.07397`.

A Appendix

A.1 Challenges in Koopman representation

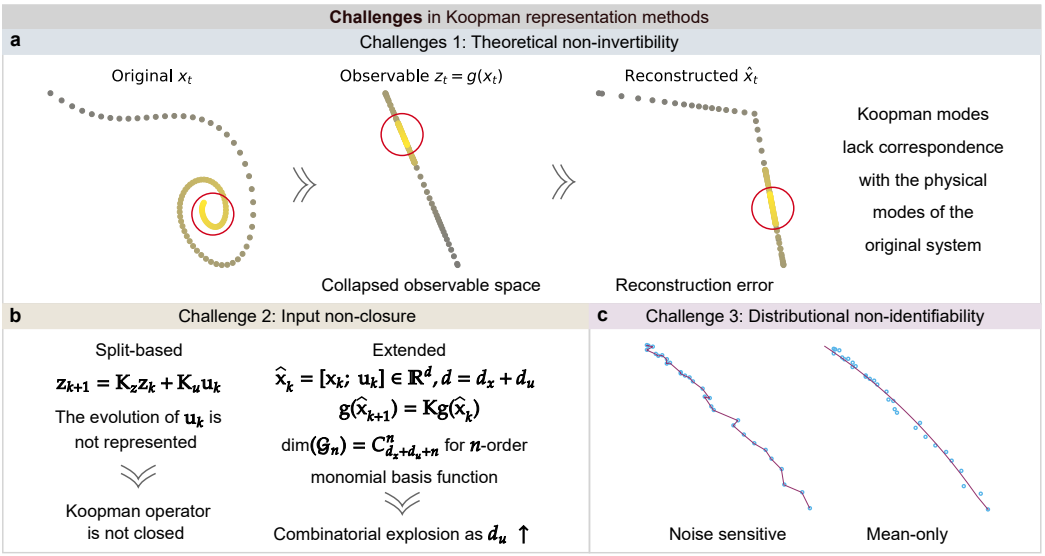

Figure 6: The challenges in Koopman representation learning. (a-c), Existing Koopman methods face three key challenges: theoretical non-invertibility, where points with the same color correspond to the same time step in autoencoder-based Koopman modeling; input non-closure, where in the extended approach, monomial basis functions are used as examples of observable functions; and distributional non-identifiability.

Koopman operator theory enables a linear representation of nonlinear dynamical systems by lifting the measurement into a higher-dimensional observable space, where the dynamics evolve linearly. In data-driven contexts, existing methods can be broadly divided into two categories. The first includes classical approaches such as extended dynamic mode decomposition (EDMD) Williams et al. (2015), DMD with control (DMDc) Proctor et al. (2016), and subspace DMD (subDMD) Takeishi et al. (2017b), which employ fixed basis functions to approximate the observables and estimate the Koopman operator via linear regression. These methods are analytically tractable but rely heavily on manual basis selection, which limits their adaptability. The second family replaces fixed dictionaries with neural networks, as in EDMD with dictionary learning (EDMD-DL) Li et al. (2017), or employs encoder-decoder architectures, as in Learning Koopman invariant subspaces (LKIS) Takeishi et al. (2017a) and DeepKoopman Lusch et al. (2018), to jointly learn the lifting and approximate inverse.

However, existing methods are unable to simultaneously address three key structural challenges that commonly arise in modeling real-world dynamical systems (Fig. 6). The first is *theoretical non-invertibility*: most approaches do not enforce a bijective mapping between the observable space and the original measurement, limiting reconstruction fidelity and undermining physical interpretability. The second is *input non-closure*: inputs are often incorporated through concatenation or heuristic structures, which disrupt the functional closure of the operator and lead to inconsistent latent evolution. The third is *distributional non-identifiability*: current models are mainly designed for deterministic prediction and lack the capacity to capture uncertainty propagation or model non-Gaussian measurement distributions.

One of the key structural challenges in Koopman representation learning is theoretical non-invertibility, particularly prominent in learning-based approaches. These methods typically implement the observable function as a neural network encoder without enforcing bijectivity. This leads to three structural consequences. First, distinct system measurements may be mapped to the same latent representation, making it impossible to resolve the correspondence between latent trajectories and original system measurements. Second, the learned Koopman modes lose their physical

interpretability, as spectral components cannot be consistently associated with interpretable system behavior. Third, the prediction–reconstruction cycle becomes structurally invalid: Even if latent dynamics evolve linearly, the absence of an exact inverse precludes an accurate recovery in the original measurement space. These issues are illustrated in the top panel of Fig. 6a. As a result, the learned representation breaks the alignment between latent evolution and true system dynamics, limiting its applicability to reconstruction, control, and interpretation tasks. Such inversion failures are especially critical in applications where measurement interpretability and trajectory reconstruction are essential, such as physical simulation, control, and diagnostics.

Another major challenge in Koopman representation learning is input non-closure, arising from the lack of a unified mechanism to incorporate external inputs into the lifted space. Unlike measurement variables, input signals are externally generated and not governed by the system dynamics, yet are treated as known sequences during training. This externality makes it challenging to consistently embed inputs into the Koopman-invariant observable space. As a result, many methods treat inputs as exogenous forcing terms. For example, DMDc and similar approaches employ two decoupled linear mappings: one for the lifted measurement and another for the input, which violates the closure assumption of the Koopman operator, which requires all observables to evolve under a single linear operator and consequently renders spectral analysis of the full system theoretically invalid. Other methods attempt to extend the Koopman framework by augmenting the measurement with inputs Korda & Mezić (2018a), effectively treating input signals as additional coordinates in the observable space in order to include them within the lifted dynamics. However, such measurement augmentation leads to a combinatorial expansion of the observable space as the input dimensionality increases, resulting in overparameterized models with high computational cost and limited scalability. These issues are illustrated in the bottom left panel of Fig. 6b. Both approaches either violate closure or introduce excessive complexity, failing to represent input-driven dynamics in a unified and interpretable manner. This limitation is especially problematic in real-world systems with structured input-output interactions, including autonomous agents, robotics, and process control.

A third core challenge in Koopman representation learning is distributional non-identifiability, referring to the inability to capture uncertainty propagation and distributional dynamics. First, classical methods such as EDMD and DMDc rely on least-squares estimation under deterministic assumptions, resulting in high sensitivity to noise and poor uncertainty modeling. Second, many learning-based approaches predict only the conditional mean of the next measurement, neglecting the underlying uncertainty. As a result, they fail to capture the full distributional structure of stochastic systems. Third, these methods are typically restricted to unimodal and light-tailed distributions, lacking the capacity to represent multimodal, heavy-tailed, or otherwise non-Gaussian behaviors commonly observed in practice. These issues are illustrated in the bottom right panel of Fig. 6c. Such limitations undermine the applicability of Koopman-based models to real-world systems with intrinsic stochasticity, sensor noise, human interaction, and environmental stochasticity, which require distributionally reliable modeling and control.

A.2 Derivation of the loss function

We can regard the measurement at time t as a random variable $\mathbf{X}_t$, $\mathbf{X}_t$ has the distribution $p_{\mathbf{X}_t}$. We assume that the distribution $p_{\mathbf{X}_t}$ is unknown and utilize the ability of normalizing flows to estimate arbitrarily complex distributions. Since the invertible extension module $\mathbf{g}_{\theta_1}$ is a feature extraction module that includes the original measurement, we do not change its probabilistic distribution. Consider $\mathbf{Y}_t = \mathbf{g}_{\theta_1}(\mathbf{X}_t)$, the normalizing flow module $\mathbf{g}_{\theta_2}$ can transform $\mathbf{Y}_t$ into a new random variable $\mathbf{Z}_t = \mathbf{g}_{\theta_2}(\mathbf{Y}_t)$ with tractable distribution function, such as Gaussian distribution. We give the following proposition

Proposition 3. *Consider the system (1) and the normalizing flow module $\mathbf{g}_{\theta_2}$. Assume that $\mathbf{Z}_t$ follows the distribution $p_{\mathbf{Z}_t}(\mathbf{z}) = \mathcal{N}(\tilde{\mathbf{z}}_t, \sigma^2\mathbf{I})$, where $\tilde{\mathbf{z}}_t = \mathbf{g}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)$ and $\tilde{\mathbf{y}}_t = \mathbf{g}_{\theta_1}(\tilde{\mathbf{x}}_t)$ with $t = 0, 1, \cdots, n$ are the observable data. A negative log-likelihood objective function is as following*

$$\mathcal{L}_{NLL} = \frac{1}{2\sigma^2}\sum_{t=1}^{n}\|\mathbf{g}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t) - \hat{\mathbf{g}}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)\|^2 - \sum_{t=1}^{n}\log\left|\det\frac{\partial \mathbf{g}_{\theta_2}(\mathbf{y}, \tilde{\mathbf{u}}_t)}{\partial \mathbf{y}}\bigg|_{\mathbf{y}=\tilde{\mathbf{y}}_t}\right|. \tag{9}$$

Proof. Consider a sample at time t, our objective is to optimize the normalizing flow $\mathbf{g}_{\theta_2}$ so that the reconstructed data by the DMD algorithm follow the same distribution as the original data. Accord-

ing to the definition of the Koopman operator (2), the reconstructed $\hat{\mathbf{g}}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)$ is expected to closely approximate the mean $\mathbf{g}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)$ of $p_{\mathbf{Z}_t}$. Furthermore, the distribution $p_{\hat{\mathbf{Z}}_t}(\mathbf{z}) = \mathcal{N}(\hat{\mathbf{z}}_t, \sigma^2\mathbf{I})$ is expected to closely approximate the distribution $p_{\mathbf{Z}_t}$, and this can be achieved by maximizing the likelihood function of the sample

$$\begin{aligned}
\mathcal{L}_{\text{MLE}}(t) &= p_{\hat{\mathbf{Y}}_t}(\tilde{\mathbf{y}}_t) \\
&= p_{\hat{\mathbf{Z}}_t}(\mathbf{g}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)) \left| \det \frac{\partial \mathbf{g}_{\theta_2}(\mathbf{y}, \tilde{\mathbf{u}}_t)}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\tilde{\mathbf{y}}(t)} \right| \\
&= \frac{N(t)}{\sqrt{(2\pi)^m \det(\sigma^2\mathbf{I})}} \exp\left(-\frac{1}{2}\mathbf{M}(t)^T (\sigma^2\mathbf{I})^{-1} \mathbf{M}(t) \right) \\
&= \frac{N(t)}{\sqrt{(2\pi\sigma^2)^m}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{M}(t)\|^2 \right),
\end{aligned} \tag{10}$$

where the vector $\mathbf{M}(t) = \mathbf{g}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t) - \hat{\mathbf{g}}_{\theta_2}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{u}}_t)$ and the scalar $N(t) = \left| \det \frac{\partial \mathbf{g}_{\theta_2}(\mathbf{y}, \tilde{\mathbf{u}}_t)}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\tilde{\mathbf{y}}_t} \right|$.

Apply a negative logarithmic transformation to the likelihood function and neglect the constants

$$-\log \mathcal{L}_{\text{MLE}}(t) = \frac{1}{2\sigma^2} \|\mathbf{M}(t)\|^2 - \log N(t). \tag{11}$$

Because for every time t, the sample $\mathbf{x}(t)$ follows an independent distribution and the product of probabilities is equivalent to the sum of their logarithms, the final objective can be obtained through a sum of the single sample result

$$\mathcal{L}_{\text{NLL}} = \sum_{t=1}^{n} -\log \mathcal{L}_{\text{MLE}}(t). \tag{12}$$

□

By the formulation of the objective function in Proposition 3, an intuitive loss function can be designed as in main text for $g_{\theta_2}(\mathbf{y}, \mathbf{u}) = g_\theta(\mathbf{x}, \mathbf{u})$ and $\lambda = 2\sigma^2$ is the custom parameter.

A.3 Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) is an algorithm for approximating the Koopman operator, and we use the exact DMD algorithm Tu (2013) which includes a Singular Value Decomposition (SVD) trick. Consider the snapshot matrices sampled from the system (1)

$$\begin{aligned}
\mathbf{G}_\mathbf{X} &= [\mathbf{g}_\theta(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0), \mathbf{g}_\theta(\tilde{\mathbf{x}}_1, \tilde{\mathbf{u}}_1), \cdots, \mathbf{g}_\theta(\tilde{\mathbf{x}}_{n-1}, \tilde{\mathbf{u}}_{n-1})], \\
\mathbf{G}_\mathbf{Y} &= [\mathbf{g}_\theta(\tilde{\mathbf{x}}_1, \tilde{\mathbf{u}}_1), \mathbf{g}_\theta(\tilde{\mathbf{x}}_2, \tilde{\mathbf{u}}_2), \cdots, \mathbf{g}_\theta(\tilde{\mathbf{x}}_n, \tilde{\mathbf{u}}_n)],
\end{aligned} \tag{13}$$

the DMD algorithm seeks a best-fit linear matrix operator $\mathbf{K}$ that approximates the Koopman operator $\mathcal{K}_\Omega$

$$\mathbf{K} = \arg\min_{\mathbf{K}} \|\mathbf{G}_\mathbf{Y} - \mathbf{K}\mathbf{G}_\mathbf{X}\|_F^2 = \mathbf{G}_\mathbf{Y}\mathbf{G}_\mathbf{X}^\dagger. \tag{14}$$

If $\mathbf{K}$ is too large, it will increase the computational burden, so we use SVD to reduce the dimensionality. By decomposing $\mathbf{G}_\mathbf{X}$ as

$$\mathbf{G}_\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{15}$$

we retain only the leading r singular values and truncate the matrices as

$$\mathbf{U}_r = \mathbf{U}(:, 1:r), \quad \mathbf{\Sigma}_r = \mathbf{\Sigma}(1:r, 1:r), \quad \mathbf{V}_r = \mathbf{V}(:, 1:r). \tag{16}$$

This gives a reduced representation

$$\tilde{\mathbf{K}} = \mathbf{U}_r^* \mathbf{G}_\mathbf{Y} \mathbf{V}_r \mathbf{\Sigma}_r^{-1}. \tag{17}$$

The eigenvalues and eigenvectors of $\tilde{\mathbf{K}}$ define the DMD eigenvalues and modes. The DMD modes are computed as

$$\mathbf{\Phi} = \mathbf{G}_{\mathbf{Y}}\mathbf{V}_r\mathbf{\Sigma}_r^{-1}\mathbf{W}, \tag{18}$$

where $\mathbf{W}$ contains the eigenvectors of $\tilde{\mathbf{K}}$: $\tilde{\mathbf{K}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$. These modes describe coherent structures in the data that evolve at rates determined by the corresponding eigenvalues.

Then the reconstruction of the observable data is

$$\hat{\mathbf{g}}_\theta(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \mathbf{\Phi}\mathbf{\Lambda}^t\mathbf{\Phi}^{-1}\mathbf{g}_\theta(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0). \tag{19}$$

For detailed implementation, the exact DMD algorithm is performed using `pyDMD` library Demo et al. (2018) Ichinaga et al. (2024).

A.4 Partially observable state reconstruction

The partially observable state reconstruction consists mainly of two parts: first, the visible state sequence is used to construct a time-delay matrix, and then an intrinsic dimension estimation method is applied to estimate the total dimensionality of the original system. Second, an encoder is used to encode the hidden states from the time-delayed data. The visible and encoded hidden states are then concatenated and fed into the KNIF framework for joint optimization.

Consider the measurement data $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n]$, the time-delayed matrix $\tilde{\mathbf{H}}$, also known as the Hankel matrix is defined as

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{x}}_0 & \tilde{\mathbf{x}}_1 & \cdots & \tilde{\mathbf{x}}_{n-h+1} \\ \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_{n-h+2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{x}}_{h-1} & \tilde{\mathbf{x}}_h & \cdots & \tilde{\mathbf{x}}_n \end{bmatrix}, \tag{20}$$

where h is the delay number.

Then we implement the Dimensionality from Angle and Norm Concentration (DANCo) algorithm Ceruti et al. (2014) on the Hankel matrix $\tilde{\mathbf{H}}$ to estimate the intrinsic dimension of the system. The DANCo algorithm is performed by the `skdim` library Bac et al. (2021). The dimension of the hidden states can then be determined by the difference between the total system dimension and the dimension of the visible states.

Finally, an encoder is used to encode the hidden states. 1D convolution is used to encode the time-delay information into hidden states, a process that is theoretically supported by Takens' embedding theorem. The hidden and visible states are then jointly used to train the KNIF framework combining with the encoder. After training, the encoder can be used to reconstruct the hidden states, while the KNIF framework can be used to reconstruct the visible states.

A.5 Comparison of Koopman modeling methods

We use a table to compare various Koopman modeling methods.

A.6 Data preparation

For the simulation data of autonomous systems and systems with input, we use MATLAB's `ode45` solver. For stochastic systems, we employ the Euler–Maruyama method. For the Sine-Gordon system, the data is generated using the analytical solution. For the KdV system, we use the split-step Fourier method Jin et al. (2024) to solve. All generated data are divided into training, validation, and test sets with a ratio of $10 : 1 : 1$. For the real-world system, we divide the provided train set into segments of the same length, resulting in a total of N_{traj} trajectories for training, and then use the provided test set for evaluation.

A.7 Neural network architecture

All the neural networks that we use are simple multilayer perceptrons (MLPs) with two hidden layers. The layer dimensions are $(input\_dim, hidden\_dim)$, $(hidden\_dim, hidden\_dim)$, and $(hidden\_dim, output\_dim)$. All layers use ReLU as the activation function.

Table 2: Comparison of Koopman modeling methods across key structural properties. *Input modeling* strategies are categorized as: *None* — no input considered; *Split-based* — input added separately as a linear term; *Extended/Conditional* — input integrated via concatenation or as a conditional variable. *Koopman interpretability* is indicated using three stars: ★ denotes (1) global linearity, (2) bijectivity between the original and observable spaces (i.e., one-to-one mapping), and (3) unified modeling of input and measurement dynamics. Filled stars (★) indicate satisfaction of the corresponding property; hollow stars (☆) indicate otherwise.

Method	Invertible mapping	Input modeling	Noise robustness	Distribution modeling	Koopman interpretability
EDMD	✓	None	✗	✗	★★☆
DMDc	✓	Split-based	✗	✗	★★☆
subDMD	✓	Split-based	✓	✗	★★☆
EDMD-DL	✓	Extended	✗	✗	★★★
LKIS	✗	Extended	✓	✗	★☆★
DeepKoopman	✗	Extended	✗	✗	☆☆★
IKN	✓	Extended	✗	✗	☆★★
FlowDMD	✓	Extended	✗	✗	★★★
KNIF (Ours)	✓	Conditional	✓	✓	★★★

For residual flows, we apply spectral normalization to the linear layers after each training epoch to ensure the invertibility of the network. For conditional flows, we concatenate the condition $\mathbf{u}$ directly with the input $\mathbf{x}$, or first extract features from $\mathbf{u}$ using a Chebyshev layer, and then concatenate the result with $\mathbf{x}$. For encoder in partially observable state reconstruction, we use a 1D convolutional network. The first convolutional layer has a kernel size of $kernel\_size$ to construct time delays, while the latter two layers use 1×1 convolutions Lu et al. (2022), which function similarly to an MLP.

A.8 Experiments parameters

The total parameters used in our experiments are the following: sampling period T_s, sampling time of a single trajectory T, the number of train set trajectories N_{traj}, the range of random initial points R. The number of trajectories in the test set and the valid set is maintained at 10% of the train set. For neural network, all residual flow blocks use MLPs with one hidden layer, and the parameters are: the dimension of the Chebyshev layer D_c, the dimension of the hidden layer D_h, the number of residual blocks N_{res}, the SVD truncation order r, the number of training epochs N_{epo}, the learing rate lr, the weight of log-determinant Jacobian term in loss function λ. Table 3 shows the detailed setup of each system.

A.9 Supplementary experimental details

- **Challenge 1: theoretical non-invertibility.** To illustrate the issue of theoretical non-invertibility, we take the autoencoder architecture on the Duffing system as an example in Fig. 6a. An autoencoder consists of an encoder $\mathbf{f}_{\text{enc}}$ which maps the input measurement $\mathbf{x}$ to a latent representation $\mathbf{z}$, and a decoder $\mathbf{f}_{\text{dec}}$ which aims to approximate the inverse of the encoder by reconstructing $\mathbf{x}$ from $\mathbf{z}$.

  The autoencoder is typically trained by minimizing the loss

$$\mathcal{L} = \sum_{t=1}^{n} \|\mathbf{f}_{\text{enc}}(\tilde{\mathbf{x}}_t) - \hat{\mathbf{f}}_{\text{enc}}(\tilde{\mathbf{x}}_t)\|^2 + \sum_{t=1}^{n} \|\tilde{\mathbf{x}}_t - \mathbf{f}_{\text{dec}} \circ \hat{\mathbf{f}}_{\text{enc}}(\tilde{\mathbf{x}}_t)\|^2, \tag{21}$$

  where $\hat{\mathbf{f}}_{\text{enc}}(\cdot)$ denotes the reconstructed result of $\mathbf{f}_{\text{enc}}(\cdot)$ using the DMD algorithm.

  However, minimizing the loss does not guarantee that the encoder is invertible. In other words, there may not exist a function $\mathbf{f}_{\text{dec}} = \mathbf{f}_{\text{enc}}^{-1}$ that is globally valid. As shown in the results of Fig. 6a, due to the non-invertibility of the mapping, the encoder $\mathbf{f}_{\text{enc}}$ may project distinct measurements onto the same point in the observable space. This leads to a collapsed

Dynamical system	T_s [s]	T [s]	N_{traj}	R	D_c	D_h	N_{res}	r	N_{epo}	lr	λ
a) Van der Pol	0.05	0.5	50000	$[-5, 5]$	50	50	3	8	100	10^{-3}	–
b) Duffing	0.1	1	50000	$[-2, 2]$	50	50	5	10	100	10^{-3}	–
c) Nonpolynomial vector field	0.1	6	10000	$[-1, 1]$	10	20	3	5	100	10^{-3}	–
d) Limit cycle	0.1	6	10000	$[-2, 2]$	20	20	10	20	100	10^{-3}	–
e) Single-well potential	0.01	0.6	10000	$[-5, -3] \cup [3, 5]$	6	10	3	3	100	10^{-3}	10^{-3}
f) Periodic potential	0.01	6	1000	$[-3, 3]$	10	20	20	5	100	10^{-3}	5×10^{-3}
g) Sine-Gordon	0.01	0.5	2000	$[-5, 5]$	50	100	3	10	100	10^{-3}	–
h) Korteweg–de Vries	0.01	0.5	10000	$[-\pi, \pi]$	50	100	3	10	100	10^{-3}	–
i) Kraichnan–Orszag (hidden)	0.01	0.6	10000	$[-1, 1]$	5	10	3	3	100	10^{-3}	–
j) Kraichnan–Orszag (sample)	Δt	$10\Delta t$	10000	$[-1, 1]$	5	40	3	3	100	10^{-3}	–
k) Kraichnan–Orszag (noise)	0.01	0.1	10000	$[-1, 1]$	5	40	3	3	100	10^{-3}	10^{-3}
l) Industrial robot	0.1	3.5	1110	–	20	80	3	20	1000	10^{-3}	10^{-3}
m) Wind speed	Δt	$10\Delta t$	2620	–	20	80	3	20	1000	10^{-3}	10^{-3}
n) Solitary wave	Δt	$31\Delta t$	48	–	20	80	3	20	1000	10^{-3}	10^{-3}
o) EMPS	0.001	0.04	621	–	20	80	3	20	1000	10^{-3}	10^{-5}

Table 3: Detailed setup of numerical experiments for dynamical systems

observable space where information about the original measurements is lost. Consequently, even if the reconstruction within the observable space appears accurate, mapping back to the original space via the decoder $\mathbf{f}_{\text{dec}}$ non-invertibility introduces reconstruction errors, again stemming from the lack of a truly invertible correspondence.

- **Challenge 2: input non-closure.** To analyze the issue of input non-closure, we consider two commonly used strategies: the Split-based approach and the Extended approach (Fig. 6b), and discuss their respective limitations.

  In Koopman analysis involving external inputs $\mathbf{u}_t$, the input sequence is typically known but not predictable. Therefore, the Koopman observable should include $\mathbf{u}_t$ to form a parameterized function space. However, in the Split-based method, the input $\mathbf{u}_t$ is excluded from the evolution equation. This exclusion prevents the Koopman operator from being well-defined, making the Koopman analysis inapplicable under this formulation. In contrast, the Extended method attempts to address this by explicitly incorporating the input into the observable space. As a representative example, consider the use of n-order monomial basis functions, where the extended observable is defined as Zeng et al. (2024)

$$g_j(\hat{\mathbf{x}}) \in \mathcal{G}_n = \left\{ \hat{x}_1^{s_1} \cdots \hat{x}_d^{s_d} | (s_1, \cdots, s_d) \in \mathbb{N}^d : s_1 + \cdots + s_d \leqslant n \right\}, \tag{22}$$

  where $\hat{\mathbf{x}} = [\mathbf{x}; \mathbf{u}]$, $\dim(\mathbf{x}) = d_x$, $\dim(\mathbf{u}) = d_u$ and $\dim(\hat{\mathbf{x}}) = d = d_x + d_u$.

  According to the rules of combinatorics, the dimension of $\mathcal{G}_n$ can be easily calculated as follows

$$\dim(\mathcal{G}_n) = C_{d_x+d_u+n}^n = \frac{(d_x + d_u + n)(d_x + d_u + n - 1) \cdots (d_x + d_u + 1)}{n!}, \tag{23}$$

  which is evident from the expression that increasing d_u leads to a curse of dimensionality, as the number of basis functions grows rapidly with the input dimension.

  For Extended combining with learning-based method, the combinatorial explosion is typically avoided. However, the model dimension still scales significantly with the input dimension d_u , resulting in a substantial increase in the number of trainable parameters. In contrast, the KNIF framework does not suffer from this issue, as its architecture decouples the input dimension from the growth of the latent space.

- **Challenge 3: distributional non-identifiability.** To illustrate this challenge, we consider a single trajectory of a system with real Koopman eigenvalues corrupted by additive observation noise

$$\begin{aligned} \dot{x}_1 &= \mu x_1 \\ \dot{x}_2 &= \lambda(x_2 - x_1^2), \end{aligned} \tag{24}$$

  where $\mu = -0.5, \lambda = -0.6$.

  The autoencoder-based structure exhibits sensitivity to noise, while the EDMD algorithm merely captures the mean behavior, failing to reconstruct the original distribution. The EDMD algorithm uses manually selected basis functions as observables. Here, we use the observable function $\left[x_1, x_2, x_1^2\right]$.

  In fact, when trained on sufficiently large datasets, learning-based methods tend to exhibit mean-seeking behavior, focusing on minimizing expected reconstruction loss rather than modeling the full distribution. On the other hand, classical methods, due to their closed-form nature, are often highly sensitive to noise and thus also struggle to recover the true underlying dynamics.

- **Reconstruction: Van der Pol system.** In the reconstruction experiments presented in the main text, we employ the KNIF framework based on the DMD algorithm to learn an observable function $\mathbf{g}_\theta$ that lifts the original state into a linear observable space. However, due to the limitations of batch-wise training, DMD-based reconstruction is performed independently for each trajectory. To enable global Koopman analysis on multiple trajectories, we propose an alternating optimization algorithm (Algorithm 2), similar to EDMD-DL Li et al. (2017), which fine-tunes a parametric Koopman matrix $\mathbf{K}$ jointly with the pretrained observable function $\mathbf{g}_\theta$. We emphasize that spectral normalization of $\mathbf{g}_\theta$ is essential when using residual flows, but may not be necessary for other types of normalizing flows. This

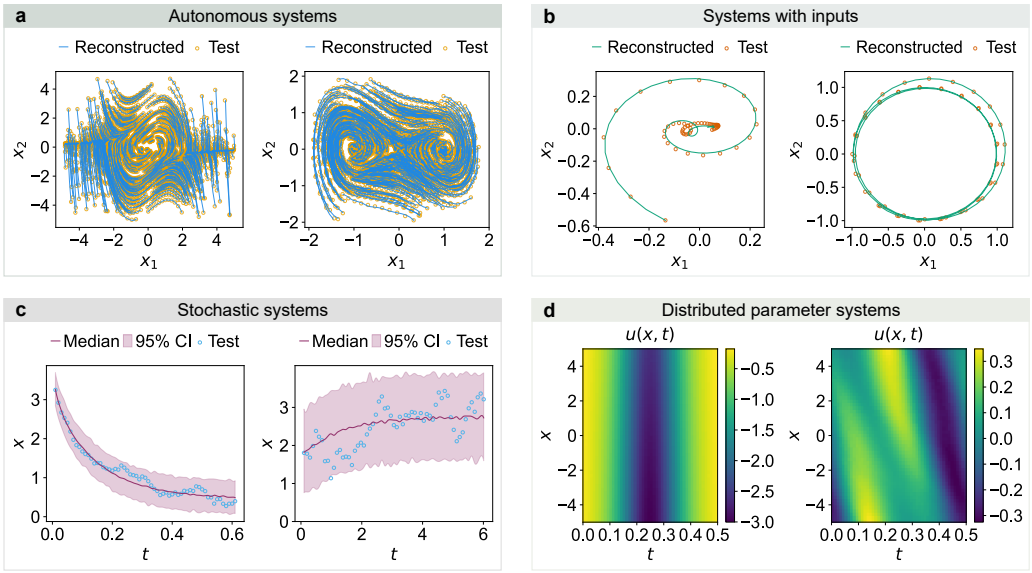

Figure 7: The reconstruction results of simulated systems. (a), Reconstruction experiment on autonomous systems composed of the Van der Pol and Duffing systems. Solid lines represent reconstructed trajectories starting from different initial conditions, and hollow circles indicate test data points. (b), A single test trajectory from the reconstruction experiment on systems with input composed of systems nonpolynomial vector field system and limit cycle system. (c), Reconstruction of stochastic single-well potential system and periodic potential system, with the median prediction shown as a solid line, the 95 percent confidence interval as a shaded region, and the test data points as hollow circles. (d), Reconstruction results of two distributed parameter systems, where the left and right panels show the spatiotemporal evolution of $u(x,t)$ in the Sine-Gordon system and the Korteweg–de Vries system, respectively.

results in a globally consistent Koopman operator. We adopt this algorithm in all subsequent experiments that require global Koopman analysis. The simulation results of all reconstruction experiments are shown in the Fig .7, including Duffing, limit cycle, and single-well potential cases for completeness.

The Van der Pol system describes an oscillator with nonlinear damping and is commonly used to analyze self- excited oscillations. Mathematically, it is governed by the equation

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -x_1 + \mu x_2(1 - x_1^2), \end{aligned} \tag{25}$$

where μ is chosen as 1 in our experiment.

In terms of nonlinear behavior, Van der Pol system exhibits a stable limit cycle changing with μ.

The reconstruction MSE is 1.636×10^{-3} and the Koopman mode analysis is shown in Fig. 8. The Koopman spectrum contains a trivial eigenvalue at 1; aside from that, the principal eigenvalues (i.e., the one with the largest magnitude excluding the trivial mode) are $0.9760 \pm 0.0175i$. The principal Koopman eigenvalues capture the oscillatory behavior of the system. Its amplitude peaks in regions of the phase space where the oscillation amplitude is large, particularly in the upper and lower sections of the limit cycle, accurately characterizing the spatial strength distribution of the mode across the state space. The phase varies smoothly along the trajectory, forming a circular pattern that reflects the rotational nature of the Van der Pol system.

- **Reconstruction: Duffing system.** The Duffing system exhibits bistability and nonlinear oscillations. Due to the cubic restoring force, the system has a double-well potential. Math-

Algorithm 2 Alternating optimization for finetuning the Koopman operator

Initialize $\mathbf{g}_\theta$ from a pretrained model; initialize $\mathbf{K}$
Set learning rates $\delta_\mathbf{g} > 0$, $\delta_\mathbf{K} > 0$, the weight of LDJ term λ, number of epochs N, Koopman update steps M
Construct dataset $\{(\mathbf{x}_t, \mathbf{x}_{t+1})\}_{t=0}^{n-1}$ and mini-batch loader
Estimate initial $\mathbf{K} \leftarrow \arg\min_{\hat{\mathbf{K}}} \sum_{t=0}^{n-1} \|\hat{\mathbf{K}}\mathbf{g}_\theta(\mathbf{x}_t) - \mathbf{g}_\theta(\mathbf{x}_{t+1})\|^2$
for epoch $= 1$ to N **do**
    Step 1: Fix K, update $\mathbf{g}_\theta$
    for each mini-batch $(\mathbf{x}_b, \mathbf{y}_b)$ **do**
        $\mathbf{g}_x \leftarrow \mathbf{g}_\theta(\mathbf{x}_b)$
        $\mathbf{g}_y \leftarrow \mathbf{g}_\theta(\mathbf{y}_b)$
        $\hat{\mathbf{g}}_y \leftarrow \mathbf{K}\mathbf{g}_x$
        $\mathcal{L} \leftarrow \text{MSE}(\hat{\mathbf{g}}_y, \mathbf{g}_y) + \lambda \text{LDJ}(\mathbf{g}_x)$
        $\theta \leftarrow \theta - \delta_\mathbf{g} \nabla_\theta \mathcal{L}$
        Normalize spectral norm of $\mathbf{g}_\theta$
    end for
    Step 2: Fix $\mathbf{g}_\theta$**, update K**
    Evaluate $\mathbf{g}_x \leftarrow \mathbf{g}_\theta(\mathbf{x})$, $\mathbf{g}_y \leftarrow \mathbf{g}_\theta(\mathbf{y})$
    for $i = 1$ to M **do**
        $\mathbf{K} \leftarrow \mathbf{K} - \delta_\mathbf{K} \nabla_\mathbf{K} \|\mathbf{K}\mathbf{g}_x - \mathbf{g}_y\|^2$
        Normalize spectral norm of $\mathbf{K}$
    end for
    Save model if current loss improves
end for
return $\mathbf{g}_\theta$, $\mathbf{K}$

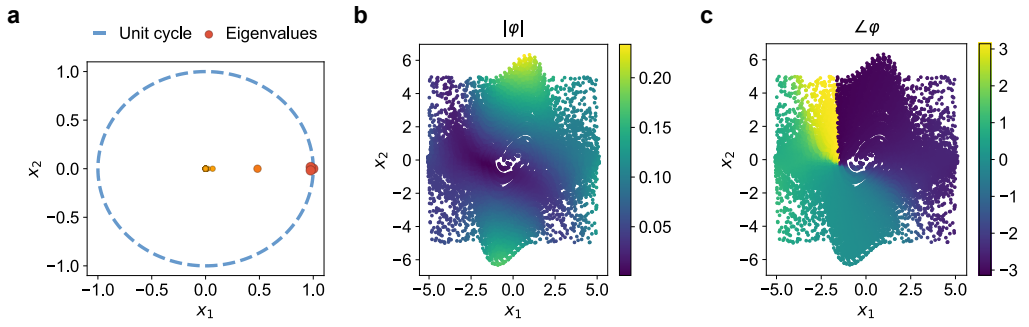

Figure 8: Koopman analysis of Van der Pol system using KNIF framework. (a), The Koopman operator spectrum. (b, c), The amplitude and phase of the principal Koopman eigenfunction.

ematically, it is governed by the equation

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\delta x_2 - \alpha x_1 - \beta x_1^3, \end{aligned} \tag{26}$$

where $\delta = 0.5, \alpha = -1, \beta = 1$ are chosen in our experiment.

In terms of nonlinear behavior, Duffing system exhibits two stable equilibrium points $(\pm 1, 0)$.

The reconstruction MSE is 7.673×10^{-5} and the Koopman mode analysis is shown in Fig. 9. The principal eigenvalues are $0.9568 \pm 0.0973i$, indicating weakly damped oscillations. The corresponding Koopman eigenfunction captures this behavior through both its amplitude and phase structure. The amplitude increases gradually from the center outward and does not peak near the equilibria $(-1, 0)$ or $(1, 0)$, reflecting a global modulation pattern across the state space rather than localized behavior near the stable points. The phase varies smoothly within each attraction basin, indicating continuous spatial organization. However, a sharp phase discontinuity occurs near $x_1 = 0$, suggesting that the Koopman eigenfunction also captures a topological transition between the two basins.

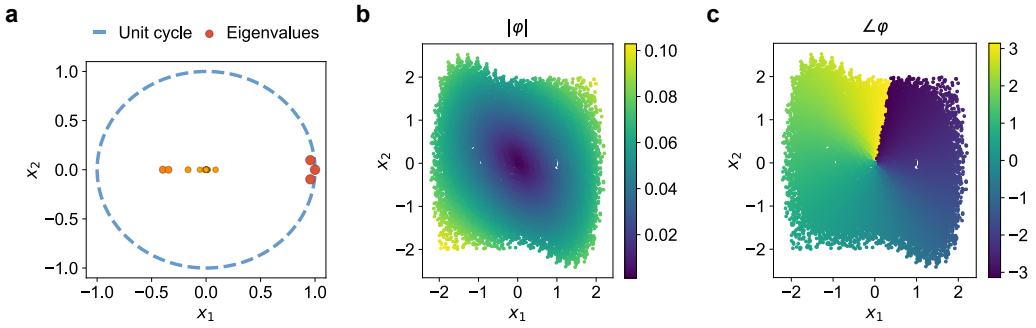

Figure 9: Koopman analysis of Duffing system using KNIF framework. (a), The Koopman operator spectrum. (b, c), The amplitude and phase of the principal Koopman eigenfunction.

- **Reconstruction: Nonpolynomial vector field system.** The nonpolynomial vector field system exhibits nonpolynomial dynamics due to the rational terms in the vector field. Mathematically, it is governed by the equation Zeng et al. (2024)

$$\begin{aligned} \dot{x}_1 &= -x_1 + \frac{4x_2}{1 + x_2^2}, \\ \dot{x}_2 &= -x_2 - \frac{4x_1}{1 + x_2^2} + u, \end{aligned} \tag{27}$$

  where $u = 0.3\cos(1.2t)$ is chosen in our experiment.

  In terms of nonlinear behavior, nonpolynomial vector field system has a stable equilibrium point at the origin $(0, 0)$.

  The reconstruction MSE is 6.523×10^{-4} and the Koopman mode analysis is shown in the main text.

- **Reconstruction: Limit cycle system.** The limit cycle system exhibits nonlinear oscillatory behavior, where trajectories spiral outward for small amplitudes and inward for large amplitudes, eventually settling into periodic motion. Mathematically, it is governed by the equation

$$\begin{aligned} \dot{x}_1 &= 3x_2 - x_1(x_1^2 + x_2^2 - 1), \\ \dot{x}_2 &= -3x_1 - x_2(x_1^2 + x_2^2 - 1) + u, \end{aligned} \tag{28}$$

  where $u = \exp(-t)$ is chosen in our experiment.

  In terms of nonlinear behavior, limit cycle system has a stable limit cycle $x_1^2 + x_2^2 = 1$.

The reconstruction MSE is 2.280×10^{-4} and the Koopman mode analysis is shown in Fig. 10. The principal eigenvalues are $0.4840 \pm 0.0671i$, indicating weakly damped oscillatory dynamics. The amplitude of the corresponding Koopman eigenfunction reaches its minimum along the limit cycle $x_1^2 + x_2^2 = 1$, and increases both inside and outside the cycle. This pattern suggests that the dominant mode is least activated once the trajectory reaches the limit cycle, consistent with the notion that the system stabilizes upon entering the periodic orbit. The phase exhibits a smooth rotational pattern centered around the origin, capturing the angular structure of the sustained oscillation.

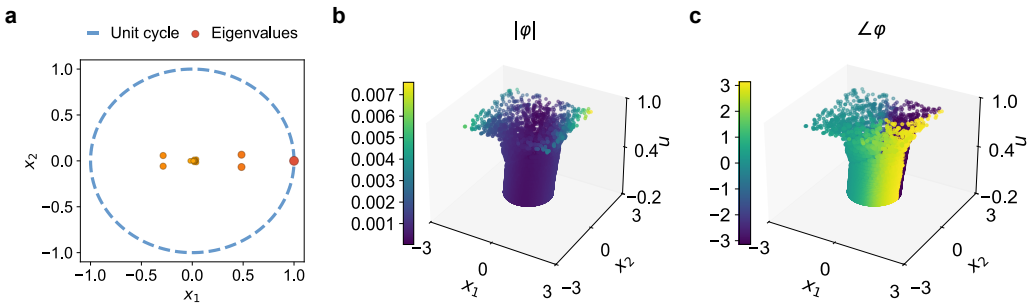

Figure 10: Koopman analysis of limit cycle system using KNIF framework. (a), The Koopman operator spectrum. (b, c), The amplitude and phase of the principal Koopman eigenfunction.

- **Reconstruction: Single-well potential system.** The single-well potential system exhibits stochastic stability, where the state is naturally attracted toward the origin, creating a stable equilibrium. However, random disturbances induced by noise cause the system to deviate, with the magnitude of these deviations depending on the noise intensity. Mathematically, it is governed by the equation

$$dx = -x^3 dt + \sigma dW, \tag{29}$$

  where W denotes the standard Brownian motion or Wiener process and $\sigma = \sqrt{0.5}$ is selected.

  In terms of nonlinear behavior, single-well potential system has a stable equilibrium point at the origin $(0, 0)$.

  The Koopman mode analysis is shown in Fig. 11. The principal eigenvalue is 0.9561 and the complex eigenvalues are $0.0875 \pm 0.2406i$. The principal eigenfunction exhibits large amplitude away from $x = 0$, while reaching a minimum at $x = 0$, where a phase transition also occurs. This indicates that the system reaches an equilibrium at this location, which effectively separates the state space into two branches. The complex-conjugate eigenfunctions display similar patterns in both amplitude and phase, although the transition is slightly shifted from $x = 0$, due to strong system noise.

- **Reconstruction: Periodic potential system.** The periodic potential system exhibits a spatially repeating structure of alternating stable and unstable equilibrium, due to the sinusoidal drift term. The system features an infinite number of equilibrium points arising from the periodic nature of the sine function. Random noise causes transitions between these potential wells, leading to rich stochastic behavior. Mathematically, it is governed by the equation

$$dx = \sin(x) dt + \sigma dW, \tag{30}$$

  where W denotes the standard Brownian motion or Wiener process and $\sigma = \sqrt{0.5}$ is selected.

  In terms of nonlinear behavior, the system possesses an infinite sequence of equilibrium points at $x = n\pi$ ($n \in \mathbb{Z}$), with alternating stability. The points $x = 2n\pi$ are unstable, while $x = (2n + 1)\pi$ are locally stable, forming a chain of periodic potential wells along the real line.

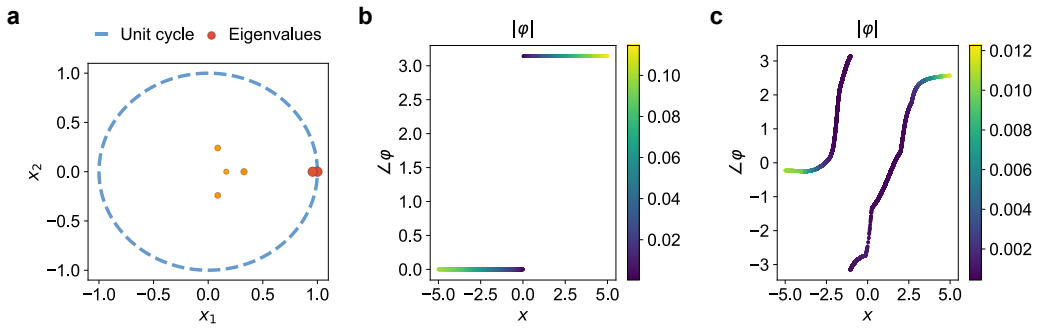

Figure 11: Koopman analysis of single-well potential system using KNIF framework. (a), The Koopman operator spectrum. (b), The amplitude and phase of the principal Koopman eigenfunction. (c), The amplitude and phase of the Koopman eigenfunction that captures oscillations.

The Koopman mode analysis is shown in Fig. 12. The principal eigenvalue is 0.9981 and the complex eigenvalues are $0.9012 \pm 0.0025i$. The principal eigenfunction exhibits large amplitude away from $x = 0$, while reaching a minimum at $x = 0$, where a phase transition also occurs. This indicates that the system reaches an equilibrium at this location. The complex-conjugate eigenfunctions exhibit large amplitudes in the far-field region, reflecting convergent behavior. The phase undergoes transitions near $\pm\pi, 0$, corresponding to the periodic potential wells of the system, although the transition is slightly shifted due to strong system noise.

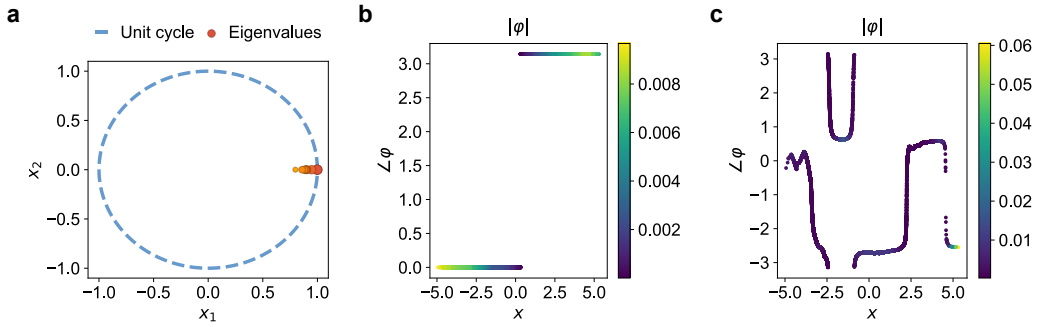

Figure 12: Koopman analysis of periodic potential system using KNIF framework. (a), The Koopman operator spectrum. (b), The amplitude and phase of the principal Koopman eigenfunction. (c), The amplitude and phase of the Koopman eigenfunction that captures oscillations.

- **Reconstruction: Sine-Gordon system.** The Sine-Gordon system is a fundamental nonlinear wave equation that arises in various physical contexts, including solid-state physics, nonlinear optics, and field theory. Mathematically, it is governed by the Sine-Gordon equation Yuan et al. (2023)

$$u_{tt} = u_{xx} - \sin(u), \tag{31}$$

where the term $\sin(u)$ leads to complex wave interactions such as kink-antikink solitons and localized oscillatory breather states.

We consider a breather solution to equation (31)

$$u(x,t) = 4\arctan\left(\frac{\sin(t/\sqrt{2})}{\cosh(x/\sqrt{2})}\right). \tag{32}$$

The reconstruction MSE is 2.875×10^{-5} and the Koopman mode analysis is shown in Fig. 13, where the amplitudes and phases of all 50 eigenfunctions are visualized along a single trajectory. The system contains only a small number of complex eigenvalues, and thus primarily exhibits decaying behavior with weak oscillatory characteristics.

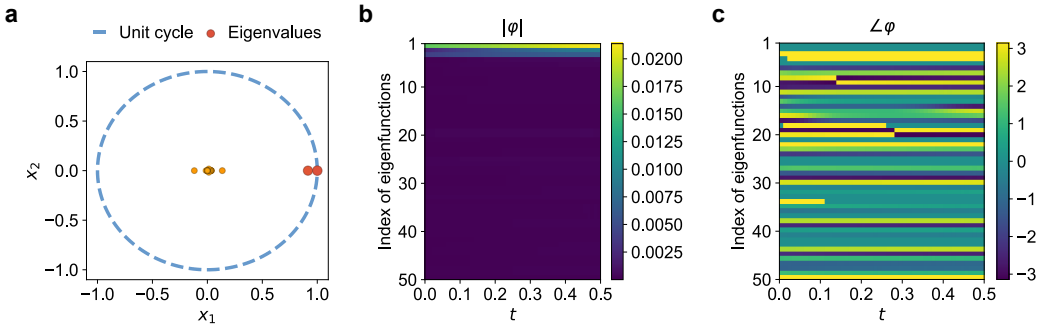

Figure 13: Koopman analysis of Sine-Gordon system using KNIF framework. (a), The Koopman operator spectrum. (b, c), The amplitude and phase of the Koopman eigenfunctions on a single trajectory.

- **Reconstruction: Korteweg–de Vries system.** The Korteweg–de Vries (KdV) system is a fundamental nonlinear dispersive wave equation that arises in modeling the unidirectional propagation of long waves in shallow water or acoustic waves in plasma. It captures the balance between nonlinear steepening and linear dispersion. Mathematically, it is governed by the KdV equation

$$u_t + u_{xxx} + uu_x = 0, \tag{33}$$

where the nonlinear convection term uu_x and the dispersive third-order derivative u_{xxx} interact to produce complex wave phenomena such as solitary waves and dispersive shock waves.

We consider a variety of initial conditions generated by random convex combinations of localized and oscillatory profiles Jin et al. (2024)

$$\begin{aligned} IC_1 &= \exp\left(-\left(x-\frac{\pi}{2}\right)^2\right), \\ IC_2 &= -\sin\left(\frac{x}{2}\right)^2, \\ IC_3 &= \exp\left(-\left(x+\frac{\pi}{2}\right)^2\right). \end{aligned} \tag{34}$$

The reconstruction MSE is 3.003×10^{-5} and the Koopman mode analysis is shown in Fig. 14, where the amplitudes and phases of all 50 eigenfunctions are visualized along a single trajectory. Apart from the two conjugate principal eigenvalues, the remaining eigenvalues are nearly arranged in a small circular cluster, indicating rich oscillatory behavior in the system and frequent phase variations over time.

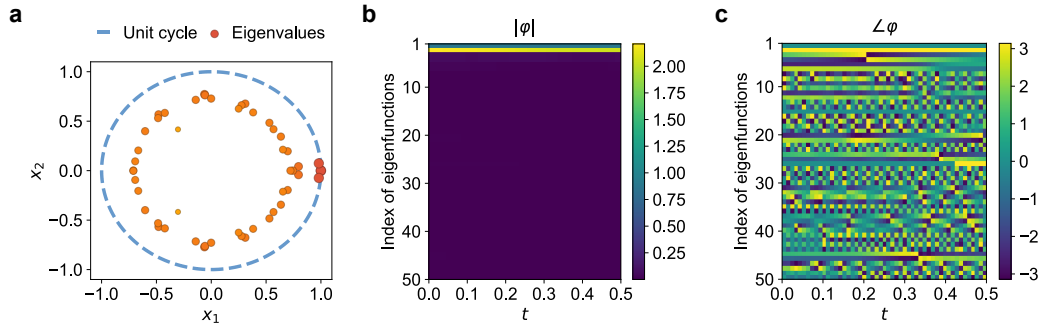

Figure 14: Koopman analysis of Korteweg–de Vries system using KNIF framework. (a), The Koopman operator spectrum. (b, c), The amplitude and phase of the Koopman eigenfunctions on a single trajectory.

- **Robustness.** For the robustness experiments, we use the Kraichnan–Orszag (KO) system. The KO system is a classical nonlinear dynamical system that arises from the study of statistical mechanics and turbulence modeling. It serves as a canonical example in the analysis of nonlinear interactions and energy-conserving dynamics. Mathematically, it is governed by the following set of ordinary differential equations Li et al. (2023)

$$\begin{aligned} \dot{x}_1 &= x_2 x_3, \\ \dot{x}_2 &= x_1 x_3, \\ \dot{x}_3 &= -2 x_1 x_2, \end{aligned} \tag{35}$$

  which defines a three-dimensional, autonomous, and nonlinear vector field.

  In terms of nonlinear behavior, the KO system exhibits energy-conserving quasi-periodic dynamics. The system has a conserved quantity $x_1^2 + x_2^2 + x_3^2$, which constrains the trajectories to evolve on closed manifolds. As a result, the system features bounded and oscillations.

  In the partially observable experiments, we select only two dimensions from the train (Fig. 15a) and validation datasets for each run, while the full-dimensional test set is used during evaluation. This setting simulates limited observability during training and assesses the generalization capability of the learned model. Our method leverages Takens' embedding theorem Takens (1981), which ensures that the full system states can be reconstructed from time-delayed embeddings of the observable states. We first construct time-delay embedded measurement data from the visible states and then apply an intrinsic dimension estimation method Chen et al. (2022); Zhang & Lipson (2024) to infer the dimensionality of the complete system, as shown in Fig. 15b. The results indicate that in all three cases, the algorithm correctly identifies the intrinsic dimension of the system as 3, implying the existence of a hidden state. We then employ an encoder to estimate the hidden state from the delay-embedded visible states Lu et al. (2022), and concatenate the estimated hidden state with the visible ones to form the full states for our framework, which is trained via joint optimization. As shown in Fig. 15c, the observable states in the test set are accurately reconstructed, while the estimation error for the hidden state is relatively larger but remains within an acceptable range.

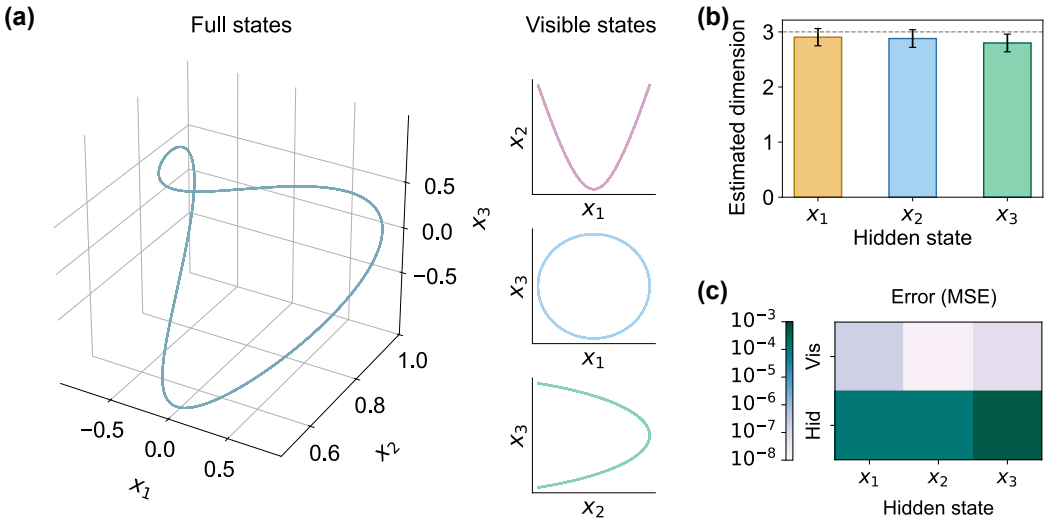

Figure 15: Hidden states reconstruction. (a), Spatial trajectories of the Kraichnan–Orszag system. The left plot shows the three-dimensional trajectory under full observation, while the three smaller plots on the right show two-dimensional trajectories when only two of the three dimensions are visible. (b), KNIF-estimation results of the original system dimension in the presence of one hidden state, where the height of each bar indicates the estimated mean and the error bars represent one standard deviation. (c), KNIF-reconstruction errors of the visible and hidden states.

For the low-sampling-rate experiments, we first generate a high-resolution dataset and then downsample it using various sampling intervals to create the experimental data. This allows us to investigate the impact of temporal sparsity on model performance.

In high-noise-strength experiments, additive Gaussian noise with varying levels of strength is added to the train, validation, and test sets simultaneously. During evaluation, the reconstruction error is computed with respect to the clean (noise-free) data using MSE, in order to assess the robustness of different methods to observation noise.

- **Ablation.** In the ablation experiments, we ensured that the number of model parameters remained approximately the same across different methods. For example, the full framework contains 2.7k parameters, while the variant replacing the observable module with a standard autoencoder uses 2.8k parameters. After ablating different components, we applied each model to reconstruct the original trajectories and compared their reconstruction errors. Then, treating each trajectory as an independent sample, we performed a Mann–Whitney U test on the reconstruction results across different trajectories to assess statistical significance.
- **Industrial robot benchmark.** The industrial robot studied in this work is governed by a nonlinear second-order differential equation that captures the dominant effects of rigid-body dynamics, actuator friction, and hydraulic compensation. The system dynamics can be expressed as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_{\mathbf{F}}(\dot{\mathbf{q}}) + \boldsymbol{\tau}_{\mathbf{H}}(\mathbf{q}) = \mathbf{U}\boldsymbol{\tau}_{\mathbf{M}}(\mathbf{q},\dot{\mathbf{q}}), \tag{36}$$

where $\mathbf{q} \in \mathbb{R}^6$ denotes the joint angles, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ contains Coriolis and centrifugal effects, and $\mathbf{g}(\mathbf{q})$ accounts for gravity. The term $\boldsymbol{\tau}_{\mathbf{F}}(\dot{\mathbf{q}})$ models nonlinear joint friction, while $\boldsymbol{\tau}_{\mathbf{H}}(\mathbf{q})$ represents the hydraulic weight counterbalance torque. The motor torques $\boldsymbol{\tau}_{\mathbf{M}}(\mathbf{q},\dot{\mathbf{q}})$ are transmitted through a diagonal gearbox matrix $\mathbf{U} = \mathrm{diag}(u_1,\ldots,u_6)$.

To evaluate model accuracy in accordance with the benchmark recommendations Weigand et al. (2023), we adopt the normalized root mean square error (NRMSE) and the coefficient of determination (R^2) as the primary performance metrics. Let $q_{\mathrm{meas},n,k}$ and $q_{\mathrm{est},n,k}$ denote the measured and predicted joint positions of joint n at time step k, respectively, and let K be the total number of time steps in the evaluation trajectory. Then, the NRMSE for joint n is defined as

$$\mathrm{NRMSE}_n = \sqrt{\frac{1}{K\sigma_{q,n}^2}\sum_{k=1}^{K}\left(q_{\mathrm{meas},n,k} - q_{\mathrm{est},n,k}\right)^2}, \tag{37}$$

where $\sigma_{q,n}^2$ is the empirical variance of the measured trajectory. The corresponding R^2 score is computed as

$$\mathrm{R}_n^2 = 100 \cdot \left(1 - \frac{\sum_{k=1}^{K}\left(q_{\mathrm{meas},n,k} - q_{\mathrm{est},n,k}\right)^2}{\sum_{k=1}^{K}\left(q_{\mathrm{meas},n,k} - \bar{q}_{\mathrm{est},n}\right)^2}\right), \tag{38}$$

with $\bar{q}_{\mathrm{est},n} = \frac{1}{K}\sum_{k=1}^{K} q_{\mathrm{est},n,k}$ denoting the predicted mean. These two metrics are evaluated separately for all six joints, and the average values across joints are also reported.

We test the trained model on the test set, which is driven by six-dimensional motor torque inputs applied to the robot joints, as shown in Fig. 16a. The one-step predicted results are presented in Fig. 16b, showing that the linear model learned by the framework still generalizes well to real data. The NRMSEs for the six joints on the test set are 0.0937, 0.0727, 0.0666, 0.0880, 0.1173, and 0.1407, respectively. The corresponding R^2 scores are 99.1224%, 99.4711%, 99.5560%, 99.2254%, 98.6244%, and 98.0234%, respectively. The average NRMSE and R^2 across all joints are 0.0965 and 99.0038%.

- **Wind speed and solitary wave datasets.** Both datasets are web datasets and do not provide test sets; therefore, we only performed reconstruction and analysis on the training sets in the main text.
- **Real-world systems model analysis.** The extracted principal Koopman eigenfunctions (solid lines) exhibit distinct behaviors across the three real-world systems in Fig. 17a-c. In the industrial robot system (Fig. 17a), the principal eigenvalue is 0.9938, and the eigenfunction captures coordinated transitions across six joint measurements under strong nonlinear

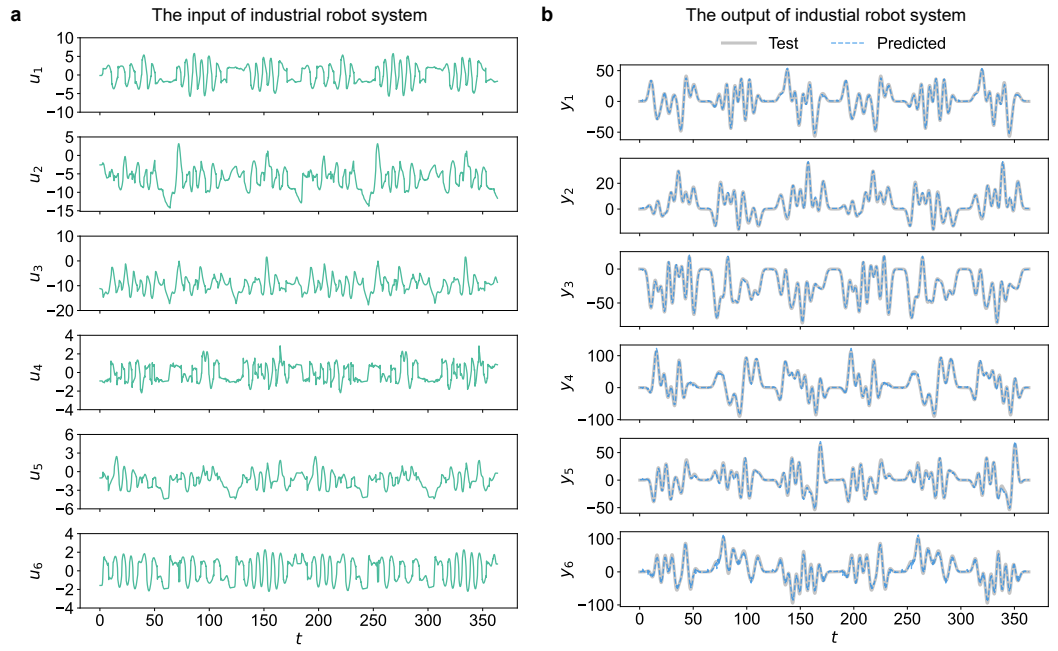

Figure 16: Test results on the industrial robot benchmark using the KNIF framework. (a), The motor torques data of the six joints serves as the input to the system. (b), The prediction results of the KNIF framework for the joint positions as output data correspond to the input. The solid lines represent the model predictions, and the hollow circles indicate the test data points.

actuation. In the wind speed system (Fig. 17b), the principal eigenvalue is 0.9162. The Koopman spectrum displays a continuous component near the origin, indicating a simple spectral structure and suggesting that the system's nonlinearity is relatively weak. The principal eigenfunction closely follows the measurement trend and can be approximated as an affine transformation. In the solitary wave system (Fig. 17c), the principal eigenvalues are $0.9690 \pm 0.0139j$; the real (blue) and imaginary (red) parts of the eigenfunction represent oscillatory spatiotemporal modes across eight spatial points, capturing both the propagating wavefront and its phase dynamics.

A.10 Exploration of modular extensions

To evaluate the extensibility and robustness of the proposed KNIF framework, we conducted a series of modular substitution experiments on its three core components. Each experiment isolates one module while keeping the rest of the framework unchanged and measures the impact on trajectory reconstruction accuracy using MSE at each time point. We note that the effects of module replacements may vary across different examples; here, we present results using the nonpolynomial vector field system as a representative case.

- **DMD variants.** The original KNIF framework uses the simplest Exact DMD, and we consider the variants Sparse-promoting DMD (SpDMD) Jovanović et al. (2014) and optimized DMD (optDMD) Askham & Kutz (2018).

  The SpDMD algorithm builds upon the original DMD by adding an ℓ_1 regularization term on the modal amplitudes, enabling automatic selection of the most representative modes

  $$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{lin}} + \lambda \|\boldsymbol{\alpha}\|_1, \tag{39}$$

  where $\boldsymbol{\alpha}$ is the amplitudes of DMD modes and λ is the weight. In our experiments, we chose $\lambda = 10^{-3}$.

  The OptDMD algorithm formulates DMD as a global nonlinear least-squares problem by jointly optimizing the modal coefficients and their corresponding continuous-time eigen-

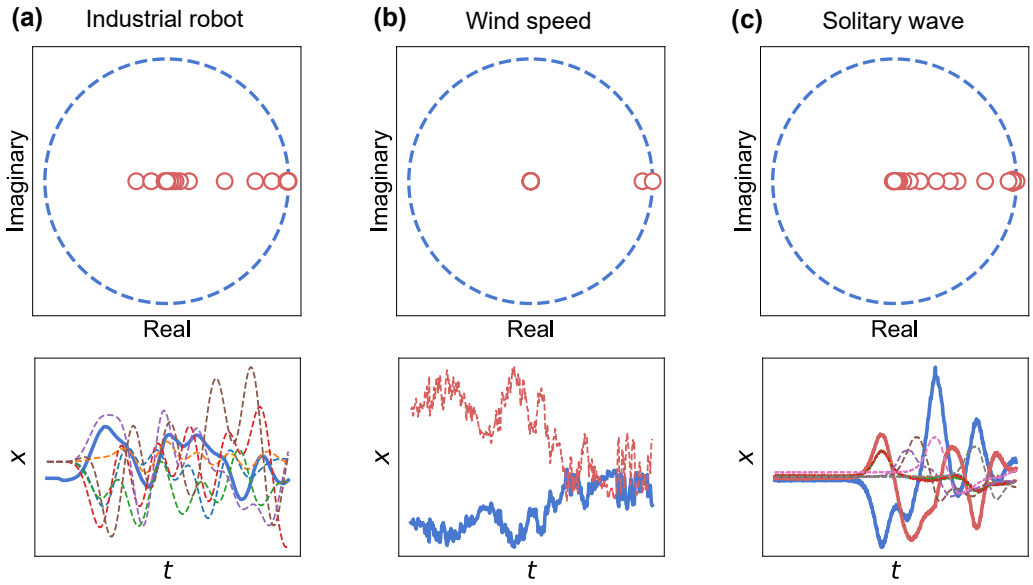

Figure 17: Real-world systems model analysis. (a-c), Analysis of real-world systems: Trajectories are normalized before plotting. Thick solid lines denote principal eigenfunctions (real or imaginary part), dashed lines indicate measurements.

values

$$\mathcal{L}_{\text{total}} = \|\mathbf{X}^* - \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B}\|_F^2, \tag{40}$$

where $\mathbf{X} = [\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n]$, $\mathbf{\Phi}(\boldsymbol{\alpha})_{ij} = \exp(\alpha_j t_i)$ encodes the temporal evolution of each mode, $\boldsymbol{\alpha}$ are the continuous-time eigenvalues, and $\mathbf{B}$ contains the modal contributions.

The comparison results are shown in Fig. 18a. Overall, both SpDMD and OptDMD outperform Exact DMD. OptDMD achieves the best fit during the middle time segment but performs worse at the beginning and end. In contrast, SpDMD maintains consistently low errors across all time points, although it is slightly inferior to OptDMD in the middle segment. However, both SpDMD and OptDMD impose significantly higher CPU overhead compared to Exact DMD.

- **Invertible extension module variants.** The original KNIF framework uses a Chebyshev layer to extract features and maintain invertibility, and we consider the neural basis expansion (NBE) Oreshkin et al. (2019) and the sinusoidal representation network (SIREN) Sitzmann et al. (2020) concatenated with the original measurements.

  The NBE uses an MLP as the basis function and learns it automatically. We set two layers $(input\_dim, lifted\_dim)$, $(lifted\_dim, lifted\_dim)$ in our experiments.

  The SIREN uses a sine layer as the basis function

  $$\phi(\mathbf{x}) = \sin(\mathbf{W}\mathbf{x} + \mathbf{b}). \tag{41}$$

  The comparison results are shown in Fig. 18b. Overall, NBE outperforms Chebyshev, while SIREN exhibits larger errors. This may be attributed to the broader range of functional forms that NBE can learn, giving it an advantage when the data size is large, whereas Chebyshev has the benefit of fewer parameters.

- **Normalizing flow module variants.** The original KNIF framework uses a residual flow, and we consider the autoregressive (AR) flow Kingma et al. (2016) and the neural ODE (NODE) flows Chen et al. (2018b).

  The AR flow uniquely applies a dimension-wise sequential transformation, where each output dimension z_i depends only on the preceding inputs $x_{<i}$. Specifically, we use the affine form

  $$z_i = \mu_i(x_{<i}) + \sigma_i(x_{<i}) \cdot x_i, \tag{42}$$

where μ_i and σ_i are neural networks similar to residual flow, and $\sigma_i > 0$ to ensure monotonicity. The inverse of AR flow is formulated as

$$x_i = \frac{z_i - \mu_i(x_{<i})}{\sigma(x_{<i})}. \tag{43}$$

The NODE flow defines a continuous transformation from input $\mathbf{x}$ to latent variable $\mathbf{z}$ by solving an initial value problem

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{f}(\mathbf{z}(t), t), \mathbf{z}(0) = \mathbf{x}, \tag{44}$$

where $\mathbf{f}$ is a neural network similar to residual flow. Its inverse is formulated as

$$\frac{d\mathbf{z}(t)}{dt} = -\mathbf{f}(\mathbf{z}(t), 1-t), \mathbf{z}(1) = \mathbf{z}. \tag{45}$$

The comparison results are shown in Fig. 18c. Overall, the AR flow and the residual flow achieve similar performance, while the NODE performs worse. However, the AR flow involves significantly more model parameters than the residual flow.

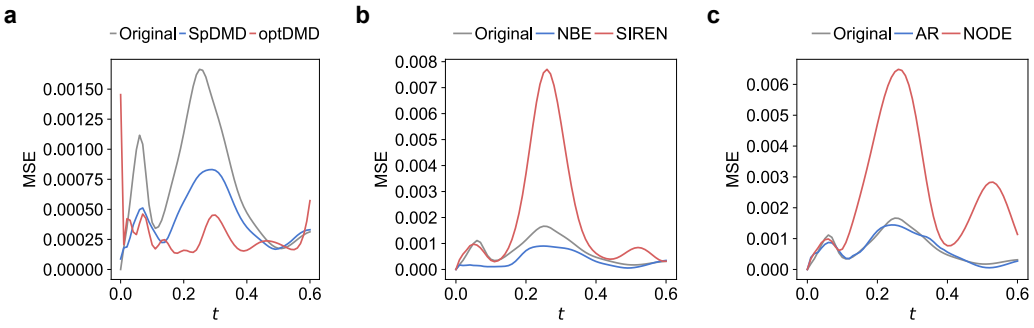

Figure 18: Comparison of reconstruction results for each module extension. (a), Comparison of Original, SpDMD and optDMD. (b), Comparison of Original, NBE and SIREN. (c), Comparison of Original, AR and NODE.