HYPERKKL: ENABLING NON-AUTONOMOUS STATE ESTIMATION THROUGH DYNAMIC WEIGHT CONDITIONING

Yahia Salaheldin Shaaban, Salem Lahlou
MBZUAI, Abu Dhabi, UAE
{Yahia.Shaaban,Salem.Lahlo}@mbzuai.ac.ae

Abdelrahman Sayed Sayed
Univ Gustave Eiffel, COSYS-ESTAS, F-59657
Villeneuve d'Ascq, France
{abdelrahman.ibrahim}@univ-eiffel.fr

ABSTRACT

This paper proposes HyperKKL, a novel learning approach for designing Kazantzis-Kravaris/Luenberger (KKL) observers for non-autonomous nonlinear systems. While KKL observers offer a rigorous theoretical framework by immersing nonlinear dynamics into a stable linear latent space, its practical realization relies on solving Partial Differential Equations (PDE) that are analytically intractable. Current existing learning-based approximations of the KKL observer are mostly designed for autonomous systems, failing to generalize to driven dynamics without expensive retraining or online gradient updates. HyperKKL addresses this by employing a hypernetwork architecture that encodes the exogenous input signal to instantaneously generate the parameters of the KKL observer, effectively learning a family of immersion maps parameterized by the external drive. We rigorously evaluate this approach against a curriculum learning strategy that attempts to generalize from autonomous regimes via training heuristics alone. The novel approach is illustrated on four nonlinear oscillators including the Duffing, Van der Pol, Lorenz, and Rössler systems.

1 INTRODUCTION

State estimation consider the reconstruction of the full internal state of a dynamical system from partial measurements is a foundational problem in science and engineering. In domains where dynamical models are used, from robotic control (Kang & Xiong, 2025; Nisar et al., 2019) to physiological monitoring (Hussain et al., 2021), only a fraction of the state variables can be directly measured, and the rest must be inferred. The difficulty of this inverse problem scales with the complexity of the underlying dynamics: as systems become more nonlinear, higher-dimensional, and subject to external forcing. As a result, principled estimation methods become increasingly necessary, yet harder to design. A dynamical system is said to be *autonomous* when its state evolution is governed entirely by its current state, without an external driving signal. However, real-world systems are almost never autonomous. Robotic platforms receive motor commands, biological systems respond to external stimuli, and industrial processes are subject to time-varying disturbances (Åström & Murray, 2021). These exogenous inputs fundamentally alter the state evolution and render such systems *non-autonomous*. The interplay between these exogenous inputs and inherent physical nonlinearities yields *non-autonomous nonlinear systems*, a regime that is ubiquitous in real-world applications yet presents the most challenges for observer synthesis.

Linear observers such as the Kalman filter (Kalman, 1960) and the Luenberger observer (Luenberger, 1964) are well studied, but they cannot capture the complexity of such systems. Nonlinear extensions, including the extended Kalman Filter and high-gain observers, exist but carry significant limitations as the former provides only local convergence guarantees Julier & Uhlmann (1997), while the latter exhibits poor transient behavior and high noise sensitivity (Khalil & Praly, 2014). Kazantzis-Kravaris/Luenberger (KKL), another class of observers, offers a robust alternative (Kazantzis & Kravaris, 1998; Andrieu & Praly, 2006). The core idea is to find a higher dimensional transformation that immerses the nonlinear dynamics into a space where the observer dynamics is linear and stable. Under the observability condition known as backward distinguishability, this trans-

formation is guaranteed to exist, allowing the state estimation to converge from any initial condition (Andrieu & Praly, 2006; Bernard & Maghenem, 2024).

The practical bottleneck of KKL observers lies in their numerical realization. The transformation map satisfies a partial differential equation (PDE) that is analytically intractable for general nonlinear systems, and its left-inverse (map to original coordinate) is equally challenging to obtain (Niazi et al., 2023). Recent learning-based approaches have made significant progress by training neural networks to approximate these maps, either through supervised regression on simulated trajectories (Janny et al., 2021), through unsupervised learning objectives (Marani et al., 2025), through physics-informed (PINN) losses that explicitly encode the PDE constraint (Niazi et al., 2025) or neural ordinary differential equations (Miao & Gatsis, 2023).

A critical limitation of current learning-based KKL methods is that they are primarily designed for autonomous systems (Buisson-Fenet et al., 2023). Although the theoretical extension to non-autonomous settings was studied by Bernard & Andrieu (2019), who showed that the KKL framework can accommodate exogenous inputs either by making the transformation maps input-dependent or by augmenting the observer dynamics with an additional injection term. However, no learning-based method has implemented such strategies in practice. On the other hand, meta-learning approaches, such as agnostic meta-learning, has been explored to adapt KKL observers to varying conditions (Trommer & Oksuz, 2023), but this approach requires online gradient updates at test time and has only been validated on autonomous systems with parameter variations rather than exogenous inputs. Extending learning-based KKL observers to handle families of exogenous inputs without retraining remains an open problem, as explicitly noted by Buisson-Fenet et al. (2023) and Niazi et al. (2025).

In this paper, we propose **HyperKKL**, a framework for extending KKL observers to **non-autonomous** systems, and for separating improvements due to **architecture** from those due to **training**. We study two complementary directions.

- **Architecture** We employ hypernetwork conditioning (Ha et al., 2016). A secondary network encodes the exogenous input signal and outputs the observer parameters. This yields an input-adaptive observer that can adjust at inference time without retraining or online gradient updates.
- **Training** We evaluate whether a **fixed-parameter KKL** observer can generalize to non-autonomous dynamics through training alone. We initialize from an observer pretrained in the autonomous setting and fine-tune on non-autonomous trajectories using **curriculum learning** that increases input complexity from simple signals to multi-frequency mixtures.

Contributions. (i) We introduce a **hypernetwork-conditioned KKL** observer that maps input signals to observer parameters, enabling adaptation to varying input conditions without retraining. (ii) We provide a controlled comparison against a training-only baseline and find that **curriculum-based transfer** from autonomous pretraining can **degrade performance** in some regimes. (iii) We present a **systematic empirical evaluation** in four nonlinear systems, **Duffing**, **Van der Pol**, **Lorenz**, and **Rössler**, comparing hypernetwork variants, curriculum schedules, and input encodings under various conditions.

2 Preliminaries and Background

2.1 KKL Observer Theory for Non Linear Systems

2.1.1 Autonomous systems formulation

KKL observer relies on the immersion of a nonlinear system into a higher-dimensional linear system (Luenberger, 1964; 1966; 1971). We consider an autonomous nonlinear systems of the form:

$$\begin{cases} \dot{x} &= f(x), \\ y &= h(x), \end{cases} \tag{1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state at time $t \in \mathbb{R}_{\geq 0}$, $x_0 \in \mathcal{X}$ is the unknown initial condition, and $y \in \mathbb{R}^{n_y}$ is the measured output. The functions $f : \mathcal{X} \to \mathbb{R}^{n_x}$ and $h : \mathcal{X} \to \mathbb{R}^{n_y}$ are assumed smooth.

The KKL framework seeks an injective mapping $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ (with $n_z = n_y(2n_x + 1)$) that lifts the state to a higher-dimensional latent space $z = \mathcal{T}(x)$ governed by linear dynamics. Specifically, $\mathcal{T}$ must satisfy the PDE:

$$\frac{\partial \mathcal{T}}{\partial x}(x)\, f(x) = A\, \mathcal{T}(x) + B\, h(x), \tag{2}$$

where $A \in \mathbb{R}^{n_z \times n_z}$ is Hurwitz, $B \in \mathbb{R}^{n_z \times n_y}$, and (A, B) is controllable. This ensures that $z(t)$ evolves as $\dot{z} = Az + Bh(x)$, and state estimation reduces to simulating a linear observer:

$$\dot{\hat{z}}(t) = A\hat{z}(t) + B\, y(t), \quad \hat{z}(0) = \hat{z}_0. \tag{3}$$

Since A is Hurwitz, the latent error $e(t) = z(t) - \hat{z}(t) \to 0$ asymptotically (Niazi et al., 2025, Remark 4). A left-inverse $\mathcal{T}^* : \mathbb{R}^{n_z} \to \mathcal{X}$ then recovers the state estimate $\hat{x}(t) = \mathcal{T}^*(\hat{z}(t))$, ensuring $\|x(t) - \hat{x}(t)\| \to 0$ as $t \to \infty$.

The existence of such a $\mathcal{T}$ is guaranteed under mild conditions: Brivadis et al. (2023) show that if the system is *backward distinguishable* and *forward complete* on a compact $\mathcal{X}$, then an injective $\mathcal{T}$ satisfying equation 2 exists.

2.1.2 Extension to Non-Autonomous Systems

We consider a non-autonomous nonlinear systems of the form:

$$\begin{cases} \dot{x} &= f(x, u(t)), \\ y &= h(x), \end{cases} \tag{4}$$

where $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state, $u(t) \in \mathbb{R}^m$ is a known external input, and $y(t) \in \mathbb{R}^{n_y}$ is the output. Extending the KKL framework to this setting is non-trivial: unlike the autonomous case, the requisite transformation $\mathcal{T}$ generally becomes *input-dependent* ($\mathcal{T}_u$), often in a causal but implicit manner that hinders practical implementation (Bernard & Andrieu, 2019). To address this, two paradigms exist: the *stationary* approach and the *dynamic* approach.

Stationary Transformation Approach Restricting attention to control-affine systems $\dot{x} = f(x) + g(x)u$, this method seeks a time-invariant diffeomorphism $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ that maps the dynamics to a linear latent form with input injection: $\dot{\hat{z}} = A\hat{z} + By + \bar{\varphi}(\hat{z})u$. The input injection $\bar{\varphi}$ is then defined to satisfy the equivariance condition $\bar{\varphi}(\mathcal{T}(x)) = \frac{\partial \mathcal{T}}{\partial x}(x)g(x)$ (Bernard & Andrieu, 2019). While architecturally simple (requiring no auxiliary states), this method is theoretically brittle. It requires the system to be uniformly instantaneously observable and the drift dynamics to be strongly differentially observable of order n_x. Consequently, it often fails for complex oscillators where inputs induce non-linear phase shifts that a static geometric map cannot capture.

Dynamic Transformation Approach To overcome the rigidity of static maps, the dynamic approach defines the transformation as a time-varying function $\mathcal{T}(x, t)$, or equivalently $\mathcal{T}(x, \theta(t))$, where $\theta(t)$ represents the state of an auxiliary dynamic filter driven by the input u. This transformation solves the time-dependent PDE:

$$\frac{\partial \mathcal{T}}{\partial x}(x, t) f(x, u(t)) + \frac{\partial \mathcal{T}}{\partial t}(x, t) = A\mathcal{T}(x, t) + Bh(x). \tag{5}$$

The key advantage here is universality: Bernard & Andrieu (2019) prove that such a transformation exists under the much milder condition of backward distinguishability. Crucially, the inclusion of the partial time derivative $\frac{\partial \mathcal{T}}{\partial t}$ allows the transformation to "slide" along the solution manifold, effectively compensating for input-induced phase shifts or velocity changes that static maps cannot capture. The trade-off is increased computational complexity, as the observer must now learn to implicitly solve a time-varying PDE, necessitating the use of hypernetworks or recurrent architectures (e.g., LSTMs) to capture the required input history.

2.2 Hypernetworks for Conditional Dynamics

A hypernetwork (Ha et al., 2016; Chauhan et al., 2024) is a neural network that generates, or modulates, the parameters of a *target* (or *base*) network conditioned on a context variable c. Given a target

network g_θ with parameters θ, a hypernetwork $\mathcal{H}_\phi$ parameterized by ϕ produces context-dependent weights:

$$\theta = \mathcal{H}_\phi(c), \tag{6}$$

enabling the target network's behavior to adapt as a function of c, rather than remaining fixed across all conditions. In dynamical-systems settings, this paradigm has been used to build *conditional dynamics models*. CODA (Kirchmeyer et al., 2022) conditions a learned dynamics model on environment-specific context vectors inferred from observed trajectories via a hypernetwork with a low-rank locality constraint. HyperPINN (Belbute-Peres et al., 2021) and HyPINO (Bischof et al., 2025) generate PINN weights conditioned on PDE parameters to solve parameterized families of differential equations without retraining.

Many observer constructions, including the KKL class, are classically formulated for *autonomous* systems. In the *non-autonomous* case, the system evolves under an exogenous input $u(t)$, and the observer must account for input-dependent behavior. A natural way to achieve this is to condition the model on a representation of the input signal, allowing a subset of the model parameters to vary with the input, rather than with explicit physical parameters alone. In this work, we adopt this perspective. The context variable c encodes a representation of the exogenous input, and the hypernetwork modulates the KKL inverse map accordingly. The precise conditioning mechanism and architecture are detailed in Section 3.

2.3 Curriculum Learning

Curriculum learning (Bengio et al., 2009) is a training strategy inspired by human cognitive development. Instead of presenting examples in random order, the training data are organized by increasing difficulty, exposing the model to easier instances before harder ones. Bengio et al. (2009) show that this can be viewed as a form of *continuation method*, a classical strategy in non-convex optimization where a smoothed objective is gradually deformed toward the target objective, yielding both faster convergence and better local minima.

In the *offline* variant of curriculum learning, all training data are pre-generated and partitioned into difficulty levels $\{\mathcal{D}_k\}_{k=1}^{K}$ before training begins. A difficulty metric defines the ordering, and a curriculum schedule governs when the learner advances to harder data. The progression can be *adaptive*. The training procedure moves to the next difficulty level when the loss on the current level plateaus, thereby aligning curriculum transitions with learning progress.

In the context of KKL observer design, the core challenge of extending from autonomous to non-autonomous systems lies in conditioning the mappings or the observer, in our case we choose to extend the training on the inverse map $\mathcal{T}^*$, which must now account for exogenous inputs. We use curriculum learning to train this inverse map in an autoencoder-like fashion, sequentially exposing it to non-autonomous signals of increasing complexity. Here, *difficulty* is associated with the spectral complexity of the exogenous input. Signals with richer high-frequency content produce more challenging system responses and, consequently, harder reconstruction targets for the inverse map. The concrete signal generation process, the difficulty metric, the curriculum schedule, and the plateau-detection rule are specified in Section 3.

3 Methodology

3.1 Problem Formulation

We consider observer design for nonautonomous nonlinear systems as defined in equation 4. Our goal is to estimate $x(t)$ from $y(t)$ and $u(t)$ by immersing the nonlinear dynamics into the linear observer equation 3. As established in Section 2.1.2, the nonautonomous setting requires an input dependent transformation $\mathcal{T}(x,t)$ satisfying the time varying PDE equation 5, restated here for convenience:

$$\frac{\partial \mathcal{T}}{\partial x}(x,t)\, f(x,u(t)) + \frac{\partial \mathcal{T}}{\partial t}(x,t) = A\,\mathcal{T}(x,t) + B\,h(x). \tag{7}$$

Compared to the autonomous PDE equation 2, the additional term $\frac{\partial \mathcal{T}}{\partial t}$ couples the transformation to the temporal evolution of the input. This is precisely what makes static transformations insufficient

for time varying inputs: unless $\mathcal{T}$ adapts over time, the PDE residual grows with the rate of change of u.

Since analytical solutions to equation 7 are intractable for general $f(x, u)$, we cast the observer design as a learning problem. We approximate both $\mathcal{T}$ and its left inverse $\mathcal{T}^*$ using neural networks whose parameters are dynamically generated by a hypernetwork $\mathcal{H}_\psi$ (Section 2.2) conditioned on the input history $u_{[t-w,\,t]}$.

Given a dataset $\mathcal{D}$ of trajectories $\{(x^{(i)}, u^{(i)}, y^{(i)})\}$, the learning objective minimizes the expected violation of the KKL conditions:

$$\min_{\psi} \mathbb{E}_{(x,u)\sim\mathcal{D}}\Big[\underbrace{\big\|x - \hat{\mathcal{T}}^*\big(\hat{\mathcal{T}}(x;\,\theta_u),\,\phi_u\big)\big\|^2}_{\mathcal{L}_{\text{rec}}} + \lambda \underbrace{\left\|\frac{\partial \hat{\mathcal{T}}}{\partial x} f(x,u) + \frac{\Delta \hat{\mathcal{T}}}{\Delta t} - A\hat{\mathcal{T}} - B\,h(x)\right\|^2}_{\mathcal{L}_{\text{PDE}}}\Big], \quad (8)$$

where θ_u, ϕ_u denote the input conditioned parameters of the encoder and decoder respectively, $\frac{\Delta \hat{\mathcal{T}}}{\Delta t}$ is a finite difference approximation of the temporal derivative, detailed in Section 3.3, and $\lambda > 0$ balances reconstruction accuracy against dynamic consistency.

3.2 HyperKKL Architecture

The core idea of HyperKKL is to extend the autonomous KKL observer to nonautonomous systems by conditioning the transformation maps on the exogenous input $u(t)$. We build upon the physics informed autoencoder of Niazi et al. (2025), comprising an encoder $\hat{\mathcal{T}}_\theta$ (the lifting map) and a decoder $\hat{\mathcal{T}}^*_\phi$ (the left inverse map), and propose two architectures that implement the stationary and dynamic paradigms of Section 2.1.2.

3.2.1 Static HyperKKL, Stationary Approach

Following the stationary formulation of Bernard & Andrieu (2019), the Static HyperKKL retains the autonomous transformation $\mathcal{T}(x)$ and augments the observer dynamics with a learned input injection term:

$$\dot{\hat{z}} = A\hat{z} + B\,y + \bar{\varphi}(\hat{z},\,u;\,\xi), \quad (9)$$

where $\bar{\varphi}$ is a small MLP parameterized by ξ that approximates the theoretically required injection $\bar{\varphi}(\mathcal{T}(x)) = \frac{\partial \mathcal{T}}{\partial x}(x)\,g(x)$ from the control affine formulation in Section 2.1.2. An LSTM encoder processes a sliding window $u_{[t-w,\,t]}$ and produces a context embedding, which is concatenated with $\hat{z}$ as input to the injection MLP. The injection network is trained to output zero when $u \equiv 0$, ensuring exact recovery of the autonomous observer on unforced dynamics.

This approach avoids modifying the learned maps $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}^*_\phi$, and instead compensates for the input's effect entirely through the observer dynamics. It is most effective when the input acts as a bounded perturbation that does not fundamentally alter the attractor geometry.

3.2.2 Dynamic HyperKKL, Dynamic Approach

For systems where the input continuously reshapes the attractor, requiring a genuinely time varying transformation $\mathcal{T}(x, t)$ as in equation 5, we employ a residual hypernetwork $\mathcal{H}_\psi$ (Section 2.2) that modulates the weights of both the encoder and decoder. The context variable is the input history $c = u_{[t-w,\,t]}$, and the parameters are decomposed as:

$$\theta_{\text{enc}}(t) = \theta^{\text{base}}_{\text{enc}} + \Delta\theta_{\text{enc}}\big(u_{[t-w,\,t]}\big), \quad (10)$$

$$\phi_{\text{dec}}(t) = \phi^{\text{base}}_{\text{dec}} + \Delta\phi_{\text{dec}}\big(u_{[t-w,\,t]}\big), \quad (11)$$

where $\theta^{\text{base}}_{\text{enc}}$ and $\phi^{\text{base}}_{\text{dec}}$ are the frozen weights obtained from Phase 1 (Section 3.3.1), and $\Delta\theta$, $\Delta\phi$ are input dependent perturbations generated by $\mathcal{H}_\psi$.

The hypernetwork consists of three components. First, a **shared LSTM encoder** processes the input window $u_{[t-w,\,t]}$ and produces a hidden state $h_t \in \mathbb{R}^{d_h}$ summarizing the input history.This hidden state is then passed to **two separate MLP decoder heads**, one for $\Delta\theta_{\text{enc}}$ and one for $\Delta\phi_{\text{dec}}$. Since

directly predicting the full weight perturbation for each target layer would be prohibitively large, each decoder head employs a **chunked prediction** strategy: the target weight matrix $W \in \mathbb{R}^{m \times n}$ is partitioned into smaller blocks, and the MLP predicts each chunk independently from the shared LSTM embedding. This keeps the decoder output dimension manageable while preserving full-rank expressivity within each chunk, avoiding the representational bottleneck of low-rank factorizations (As illustrated in Figure 1).

This residual structure ensures that when $u \equiv 0$, the LSTM hidden state produces $\Delta\theta = \Delta\phi = 0$, exactly recovering the autonomous observer. The low rank constraint further acts as an implicit regularizer, preventing the hypernetwork from overperturbing the well trained base maps.

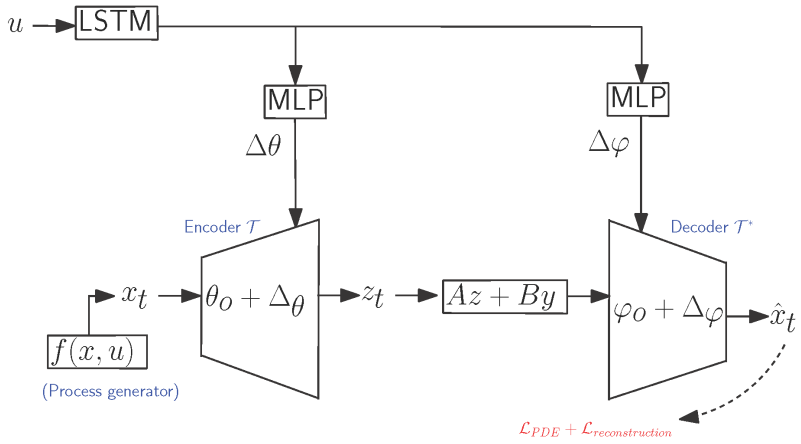


Figure 1: **Dynamic HyperKKL Architecture (Phase 2).** The base encoder $\hat{\mathcal{T}}_{\theta^{\text{base}}}$ and decoder $\hat{\mathcal{T}}^*_{\phi^{\text{base}}}$, pre-trained on autonomous dynamics in Phase 1, are frozen. A shared LSTM encoder processes the input window $u_{[t-w,\,t]}$ and produces a hidden state h_t, which is passed to two separate MLP decoder heads that predict chunked weight perturbations $\Delta\theta_{\text{enc}}$ and $\Delta\phi_{\text{dec}}$. The perturbations are added to the frozen base weights, yielding input-conditioned maps $\hat{\mathcal{T}}(x;\, \theta^{\text{base}} + \Delta\theta)$ and $\hat{\mathcal{T}}^*(z;\, \phi^{\text{base}} + \Delta\phi)$.

3.3 Training Procedure

3.3.1 Two Phase Sequential Training

Following the sequential training paradigm of Niazi et al. (2025) to avoid gradient conflict between the encoder and decoder objectives, we adopt a two phase procedure:

Phase 1, Autonomous Pretraining. We train the base encoder $\hat{\mathcal{T}}_{\theta^{\text{base}}}$ and decoder $\hat{\mathcal{T}}^*_{\phi^{\text{base}}}$ on unforced dynamics ($u \equiv 0$) using the physics informed loss from Niazi et al. (2025), comprising a data fit term $\|z - \hat{\mathcal{T}}(x)\|^2$ on labeled trajectory pairs and the autonomous PDE residual equation 2 evaluated on collocation points. This produces maps that satisfy the autonomous KKL conditions and establishes a high quality initialization. Upon completion, θ^{base} and ϕ^{base} are frozen.

Phase 2, Hypernetwork Training. With the base maps fixed, we train only the hypernetwork parameters ψ on forced trajectories. The loss equation 8 is computed using the input conditioned maps $\hat{\mathcal{T}}(x;\, \theta^{\text{base}} + \Delta\theta)$ and $\hat{\mathcal{T}}^*(z;\, \phi^{\text{base}} + \Delta\phi)$. The spatial gradient $\frac{\partial \hat{\mathcal{T}}}{\partial x}$ is computed via automatic differentiation, while the temporal derivative $\frac{\partial \mathcal{T}}{\partial t}$ in equation 7 is approximated via finite differences over consecutive input windows:

$$\frac{\Delta \hat{\mathcal{T}}}{\Delta t} \approx \frac{\hat{\mathcal{T}}\big(x;\, \theta(u_{[t,\,t+\Delta t]})\big) - \hat{\mathcal{T}}\big(x;\, \theta(u_{[t-\Delta t,\,t]})\big)}{\Delta t}. \tag{12}$$

That is, for the same state x, we evaluate the encoder under two adjacent input windows and take their difference. This captures how the transformation evolves as the input shifts, without requiring explicit differentiation through the LSTM.

3.3.2 ADAPTIVE CURRICULUM LEARNING, BASELINE

As a training strategy baseline, we evaluate the **Adaptive Curriculum** approach described in Section 2.3. This method uses the same static architecture as the Autonomous observer but trains $\hat{\mathcal{T}}^*$ on nonautonomous data of progressively increasing difficulty.

Training trajectories are generated offline and partitioned into difficulty levels $\{\mathcal{D}_k\}_{k=1}^K$ ordered by spectral complexity: $k = 1$ corresponds to constant inputs ($u = c$), $k = 2$ to low frequency sinusoids, and subsequent levels introduce higher frequency components and mixtures. The difficulty metric is defined by the dominant frequency content and rate of change $\|\dot{u}\|$ of each trajectory's input signal. Training proceeds on level $\mathcal{D}_k$ until the loss plateaus, detected when the relative improvement falls below a threshold ϵ over a patience window of p epochs, at which point the scheduler advances to $\mathcal{D}_{k+1}$. This tests whether data diversity alone, without architectural changes, can bridge the gap to nonautonomous observation.

4 EXPERIMENTAL EVALUATION

We evaluate the proposed HyperKKL framework over four nonlinear benchmarks described in detail in Appendices A.1, A.2, A.3, and A.4. Where they are classified into two low-dimensional oscillators (Duffing, Van der Pol), and two chaotic systems (Rossler, Lorenz).

4.1 EXPERIMENTAL SETUP

Datasets and Tasks. For each system, we generate synthetic trajectories via RK4 integration ($\Delta t = 0.05s$, horizon $T = 50s$). Process and measurement noise are added ($\sigma = 0.01$). We evaluate the generalization on four input regimes $u(t)$: (i) **Zero** (autonomous), (ii) **Constant** ($u \sim \mathcal{U}[-1, 1]$), (iii) **Sinusoid** (randomized A, ω, ϕ), and (iv) **Square Wave** (discontinuous jumps).

Baselines. We compare against two standard approaches:(i) **Autonomous KKL**, the observer trained solely on unforced dynamics ($u = 0$), which quantifies the domain shift caused by external inputs; and (ii) **Adaptive Curriculum**, the same static inverse map $\mathcal{T}^*$ retrained on progressively complex input regimes (constant $\rightarrow$ sinusoidal $\rightarrow$ square wave), testing whether data diversity alone can overcome architectural limitations.

Our Framework (HyperKKL). We propose two variants of the HyperKKL architecture:

- **Static HyperKKL:** A stationary transformation $\mathcal{T}(x)$ paired with a learned input injection $\bar{\varphi}(z, u)$ that conditions the observer dynamics on the instantaneous input. This implements the "Stationary Approach" (Sec. 2.1.2).
- **Dynamic HyperKKL:** The full time-varying transformation $\mathcal{T}(x, \theta(t))$ where parameters $\theta(t)$ are generated by an LSTM-based hypernetwork. This explicitly solves the dynamic PDE equation 5 by adapting to the input history.

Implementation Details & Training Strategy. All encoders are 3-layer MLPs (150 units for oscillators, 350 for chaotic systems). The Dynamic HyperKKL employs an LSTM hypernetwork (64 units, window $w = 100$) predicting encoder weights via a low-rank decomposition (Rank: 32 for oscillators, 128 for chaotic systems). Following the physics-informed framework of Niazi et al. (2025), we adopt a **sequential training** scheme to mitigate gradient conflict: we first **warm-start** the base encoder $\mathcal{T}_{\text{base}}$ on autonomous dynamics (Phase 1), then train the hypernetwork on forced dynamics (Phase 2). To prevent loss explosion on large-magnitude chaotic systems (e.g., Lorenz), we normalize the vector field term $f(x)$ in the PDE loss and apply **gradient clipping** (norm 1.0). We adopt the data set generation protocol and autonomous baselines from Niazi et al. (2025), as they represent the current state-of-the-art for autonomous KKL.

Evaluation Metrics. We assess state estimation accuracy using the **Root Mean Squared Error (RMSE)** averaged over test trajectories generated with input parameters distinct from those in the training set, ensuring that we evaluate generalization to unseen forcing regimes. To account for the varying scales of different systems, we also report the **Symmetric Mean Absolute Percentage Error (SMAPE)**. Both metrics are computed on the steady-state response (ignoring the first 5% of the trajectory) to focus on long-term tracking capability rather than initial transient convergence.

4.2 RESULTS AND ANALYSIS

Tables 1 and 2 report RMSE and SMAPE across four systems and four input conditions. We organize our analysis around three key findings.

Hypernetwork Approaches Improve State Estimation on Oscillatory and Mildly Chaotic Systems. On the Duffing oscillator, the **Static HyperKKL** achieves the strongest results, reducing RMSE by up to 62% relative to the Autonomous baseline under sinusoidal inputs ($0.26 \rightarrow 0.10$) and 48% under square wave inputs ($0.33 \rightarrow 0.17$). This is consistent with the theoretical expectation: for low dimensional oscillators whose attractor geometry shifts smoothly with a slowly varying input, a *stationary* transformation $\mathcal{T}(x, u)$ suffices to maintain injectivity. On the Van der Pol oscillator, both hypernetwork variants improve over the autonomous baseline under time varying inputs, with the **Dynamic HyperKKL** achieving the best RMSE on sinusoidal (0.21) and square wave (0.22) forcing. For the chaotic Rössler system, the Dynamic approach yields the lowest errors across all non zero input types (e.g., $1.48 \rightarrow 1.36$ under square wave input), confirming that temporal aggregation of input history becomes increasingly important as attractor complexity grows.

Catastrophic Failure of Curriculum Learning and Static Conditioning on Chaotic Attractors. Curriculum Learning performs dramatically worse than even the naive Autonomous baseline on every system and input condition tested. For instance, on the Van der Pol oscillator RMSE increases from 0.15 to 1.10 under zero input, and from 0.25 to 1.15 under square wave input. On Lorenz, RMSE roughly doubles ($5.55 \rightarrow 11.6$). This demonstrates that the bottleneck is *representational*, not educational: exposing a static architecture to progressively complex non autonomous instances cannot compensate for an inductive bias that is mathematically incapable of solving the required dynamic PDE. Similarly striking is the behavior of the **Static HyperKKL** on the Lorenz system, where conditioning on instantaneous input leads to catastrophic degradation (RMSE ≈ 16 under all non zero inputs, versus 5.5 for the Autonomous baseline). This confirms the theoretical analysis of Bernard (2019): for highly sensitive chaotic systems, a static transformation $\mathcal{T}(x, u(t))$ is insufficient to ensure injectivity, and conditioning on the wrong information can be *actively harmful*, distorting the learned immersion rather than refining it.

Theoretical Consistency and Limitations on the Lorenz Attractor. Across all four systems, the hypernetwork methods correctly recover the autonomous baseline with zero input ($u = 0$), empirically validating the architectural constraint $\Delta\theta \rightarrow 0$ as $u \rightarrow 0$. However, the Lorenz system exposes a fundamental limitation: the **Autonomous** baseline achieves the best overall performance (RMSE ≈ 5.5), and even the Dynamic HyperKKL, while dramatically outperforming Curriculum ($11.6 \rightarrow 6.67$) and Static ($16.2 \rightarrow 6.66$), incurs a modest degradation relative to the input agnostic observer ($5.55 \rightarrow 6.66$). We attribute this to the extreme sensitivity of the Lorenz attractor to perturbations: small errors in the hypernetwork's input conditioned weight modulation propagate exponentially along unstable manifolds, producing residual estimation drift that a conservative autonomous observer avoids by ignoring the input entirely. This suggests that for systems near the edge of observability, the additional representational capacity of hypernetworks must be paired with explicit stability guarantees or regularization to prevent the input conditioning pathway from introducing more noise than signal. Addressing this trade off, potentially through Lyapunov informed training constraints or adaptive gating mechanisms that attenuate modulation under high sensitivity, is an important direction for future work.

Figure 2 illustrates the qualitative behavior on the Duffing oscillator under discontinuous square wave forcing. The Dynamic HyperKKL tracks the ground truth smoothly across input transitions, whereas the Static method exhibits transient spikes at discontinuities, and the Autonomous and Curriculum baselines accumulate persistent phase drift.

Table 1: State Estimation Performance (Duffing and Van der Pol): **RMSE** (SMAPE %). Lower is better.

Method	**Duffing**				**Van der Pol**			
	Zero	Const	Sin	Sqr	Zero	Const	Sin	Sqr
Autonomous	0.04 (5.6)	0.63 (66)	0.26 (26)	0.33 (31)	0.15 (7.0)	0.35 (23.1)	0.23 (9.8)	0.25 (10.5)
Curriculum	0.27 (33)	0.64 (63)	0.44 (41)	0.57 (46)	1.10 (51.4)	1.00 (54.3)	1.15 (51.5)	1.15 (51.7)
Static HyperKKL	**0.04** (5.6)	**0.39**↓ (38)	**0.10**↓ (9.3)	**0.17**↓ (14)	0.12↓ (5.3)	**0.26**↓ (14.1)	0.24 (10.2)	0.25 (10.8)
Dynamic HyperKKL	0.08 (8.2)	0.56↓ (62)	0.24↓ (25)	0.27↓ (28)	**0.12**↓ (5.0)	0.38 (25.6)	**0.21**↓ (8.6)	**0.22**↓ (9.1)

Table 2: State Estimation Performance (Rossler and Lorenz): **RMSE** (SMAPE %). Lower is better.

Method	Rossler (Chaotic)				Lorenz (Chaotic)			
	Zero	Const	Sin	Sqr	Zero	Const	Sin	Sqr
Autonomous	1.14 (6.7)	1.75 (8.6)	1.47 (7.6)	1.48 (8.3)	**5.56** (18)	**5.50** (18)	**5.58** (18)	**5.55** (18)
Curriculum	5.58 (35)	5.55 (35)	5.94 (37)	5.61 (38)	11.5 (41)	11.4 (41)	11.6 (42)	11.6 (42)
Static HyperKKL	1.14 (6.7)	**1.50**↓ (9.5)	1.70 (10)	1.75 (12)	**5.56** (18)	16.0 (51)	16.3 (52)	16.2 (51)
Dynamic HyperKKL	**1.01**↓ (5.1)	1.57↓ (7.7)	**1.38**↓ (6.0)	**1.36**↓ (6.9)	6.67 (22)	6.64 (22)	6.67 (22)	6.66 (22)

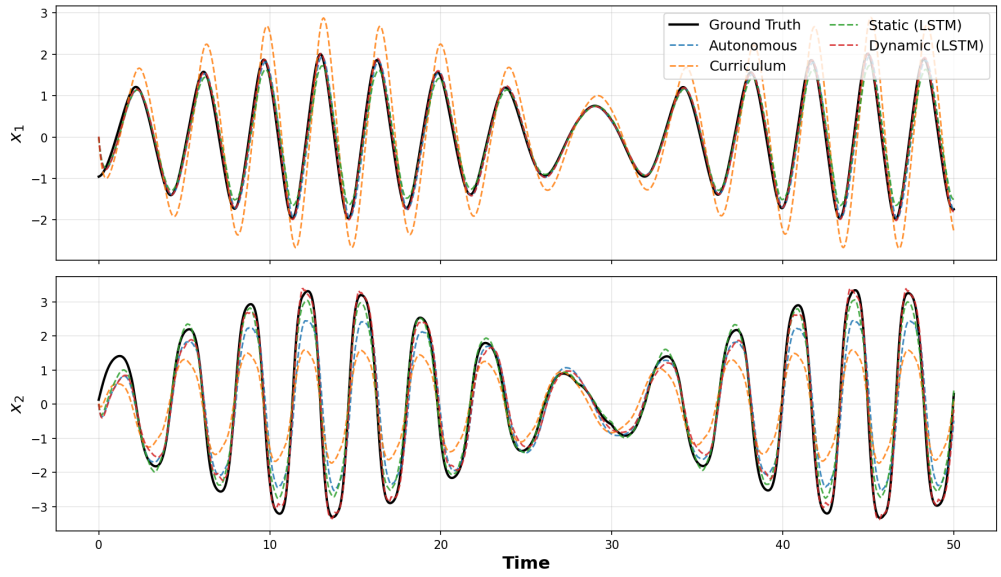


Figure 2: **Duffing (Square Input):** The Dynamic HyperKKL adapts to discontinuous square waves without the transient spikes seen in static methods.

5 Conclusion and Future Work

We introduce **HyperKKL**, a framework that extends physics-informed KKL observer design to non autonomous nonlinear systems by conditioning the observer parameters on input history via hypernetworks. Our experiments reveal that hypernetwork based observers substantially reduce estimation error on oscillatory and mildly chaotic systems under exogenous inputs, while the catastrophic failure of Curriculum Learning across all benchmarks confirms that the challenge is *representational*, not a lack of training on exogenous input dynamics. However, on the highly sensitive chaotic attractors, the autonomous observer remains the most robust, showing an architectural caveat where input conditioning adds a capacity that can improve tracking, but also introduces a modulation pathway through which errors propagate along unstable manifolds. The divergent failure modes, with Curriculum failing everywhere and Static HyperKKL collapsing on Lorenz, underscore that naive input conditioning can be worse than ignoring the input entirely, motivating architectures grounded in the structure of the underlying PDE.

Future Work. The Lorenz failure mode suggests that hypernetwork modulation requires explicit stability constraints, such as Lyapunov informed regularization or adaptive gating that attenuates weight perturbations in high sensitivity regions. Disentangling phase modulation from amplitude scaling within the architecture may further improve robustness. Finally, extending to partially known dynamics would bridge the gap between the idealized benchmarks studied here and the grey box industrial settings with model uncertainty and sparse sensor data.

REFERENCES

Vincent Andrieu and Laurent Praly. On the existence of a kazantzis–kravaris/luenberger observer. *SIAM Journal on Control and Optimization*, 45(2):432–456, 2006.

Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

Filipe de Avila Belbute-Peres, Fei Sha, and Yi-fan Chen. Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks. In *The Symbiosis of Deep Learning and Differential Equations (NeurIPS)*, 2021.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Pauline Bernard. *Observer design for nonlinear systems*, volume 479. Springer, 2019.

Pauline Bernard and Vincent Andrieu. Luenberger observers for nonautonomous nonlinear systems. *IEEE Transactions on Automatic Control*, 64(1):270–281, 2019. doi: 10.1109/TAC.2018.2872202.

Pauline Bernard and Mohamed Maghenem. Reconstructing indistinguishable solutions via a set-valued kkl observer. *Automatica*, 166:111703, 2024. doi: 10.1016/j.automatica.2024.111703.

Rafael Bischof, Michal Piovarci, Michael A. Kraus, Siddhartha Mishra, and Bernd Bickel. Hypino: Multi-physics neural operators via hyperpinns and the method of manufactured solutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Lucas Brivadis, Vincent Andrieu, Pauline Bernard, and Ulysse Serres. Further remarks on kkl observers. *Systems & Control Letters*, 172:105429, 2023. doi: 10.1016/j.sysconle.2022.105429.

Mona Buisson-Fenet, Lukas Bahr, Valery Morgenthaler, and Florent Di Meglio. Towards gain tuning for numerical kkl observers. *IFAC-PapersOnLine*, 56(2):4061–4067, 2023.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *Artificial Intelligence Review*, 57(9):250, 2024. doi: 10.1007/s10462-024-10862-8.

Jean-Marc Ginoux. *Van der Pol's Method: A Simple and Classic Solution*, pp. 275–289. Springer International Publishing, Cham, 2017. ISBN 978-3-319-55239-2. doi: 10.1007/978-3-319-55239-2_10. URL https://doi.org/10.1007/978-3-319-55239-2_10.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Zeshan M. Hussain, Rahul G. Krishnan, and David A. Sontag. Neural pharmacodynamic state space modeling. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4500–4510. PMLR, 2021. URL https://proceedings.mlr.press/v139/hussain21a.html.

Steeven Janny, Vincent Andrieu, Madiha Nadri, and Christian Wolf. Deep kkl: Data-driven output prediction for non-linear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 4376–4381, 2021. doi: 10.1109/CDC45484.2021.9683277.

S. J. Julier and J. K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, 1997.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 1960.

Jiarong Kang and Xiaobin Xiong. Simultaneous ground reaction force and state estimation via constrained moving horizon estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7080–7086, Atlanta, GA, USA, May 2025. doi: 10.1109/ICRA55743.2025.11127398. URL https://doi.org/10.1109/ICRA55743.2025.11127398.

Nikolaos Kazantzis and Costas Kravaris. Nonlinear observer design using lyapunov's auxiliary theorem. *Systems & Control Letters*, 34(5):241–247, 1998.

H. K. Khalil and L. Praly. High-gain observers in nonlinear feedback control. *International Journal of Robust and Nonlinear Control*, 2014.

M. Kirchmeyer, Y. Yin, J. Donà, N. Baskiotis, A. Rakotomamonjy, and P. Gallinari. Generalizing to new physical systems via context-informed dynamics model. *arXiv preprint arXiv:2202.01889*, 2022.

H. J. Korsch and H.-J. Jodl. *The Duffing Oscillator*, pp. 157–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-662-03866-6. doi: 10.1007/978-3-662-03866-6_8. URL https://doi.org/10.1007/978-3-662-03866-6_8.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130 – 141, 1963. doi: 10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml.

D. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, 11 (2):190–197, 1966. doi: 10.1109/TAC.1966.1098323.

D. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6): 596–602, 1971. doi: 10.1109/TAC.1971.1099826.

David G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2):74–80, 1964. doi: 10.1109/TME.1964.4323124.

Y. Marani, I. Filho, T. Al-Naffouri, and T. M. L. Kirati. Unsupervised physics-informed neural network-based nonlinear observer design for autonomous systems using contraction analysis. In *2025 European Control Conference (ECC)*, pp. 2327–2332, 2025. doi: 10.23919/ECC65951.2025.11186841.

Keyan Miao and Konstantinos Gatsis. Learning robust state observers using neural odes. In *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, pp. 208–219, 2023.

M. Umar B. Niazi, John Cao, Matthieu Barreau, and Karl H. Johansson. KKL observer synthesis for nonlinear systems via physics-informed learning. *arXiv preprint arXiv:2501.11655*, 2025.

Muhammad Umar B. Niazi, John Cao, Xudong Sun, Amritam Das, and Karl Henrik Johansson. Learning-based Design of Luenberger Observers for Autonomous Nonlinear Systems, April 2023. URL http://arxiv.org/abs/2210.01476. arXiv:2210.01476 [math].

Barza Nisar, Philipp Foehn, Davide Falanga, and Davide Scaramuzza. VIMO: Simultaneous visual inertial model-based odometry and force estimation. *IEEE Robotics and Automation Letters*, 4(3): 2785–2792, 2019. doi: 10.1109/LRA.2019.2918689. URL https://doi.org/10.1109/LRA.2019.2918689.

Otto E Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

Lukas Trommer and Halil Yigit Oksuz. Adaptive meta-learning-based kkl observer design for nonlinear dynamical systems, 2023.

A SYSTEM DESCRIPTION

Experiment Settings: All experimentss[1] herein are run on a single server equipped with NVIDIA RTX A6000 (50 GB Vram), 48 CPU cores, and 256 GB RAM with PyTorch library V2.9.

A.1 REVERSE DUFFING OSCILLATOR

The Reverse Duffing oscillator is a variant of the classical oscillator studied by Georg Duffing to model structural instability and chaotic behavior in mechanical systems (Korsch & Jodl, 1999). It features a cubic nonlinearity in the stiffness term, making it a standard benchmark for evaluating the observer performance in the presence of strong, non-Lipschitz nonlinearities. We consider the continuous-time dynamics given by:

$$\begin{cases} \dot{x}_1 = x_2^3, \\ \dot{x}_2 = -x_1, \\ y = x_1, \end{cases}$$

where $x = [x_1, x_2]^\top \in \mathbb{R}^2$ is the state vector and $y \in \mathbb{R}$ is the measured output. This system is particularly challenging due to the aggressive cubic growth of the x_2^3 term, testing the observer's ability to handle fast dynamics far from the equilibrium.

A.2 VAN DER POL OSCILLATOR

The Van der Pol oscillator is a 2-dimensional limit cycle oscillator originally by Van der Pol to model electrical circuits with vacuum tubes (Ginoux, 2017). It is characterized by a nonlinear damping that dissipates energy at high amplitudes but generates energy at low amplitudes, resulting in a stable limit cycle, making it valuable for studying self-sustained oscillations in biological and physical systems. We consider the parameterized dynamics with $\mu = 3$:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1, \\ y = x_1, \end{cases}$$

where $x = [x_1, x_2]^\top \in \mathbb{R}^2$ is the state vector, $\mu > 0$ determines the nonlinearity and damping strength, and $y \in \mathbb{R}$ is the measured output. The parameter μ significantly alters the stiffness of the system, making it ideal for testing the hypernetwork ability to handle the varying physical parameters.

A.3 RÖSSLER ATTRACTOR

The Rössler attractor is a 3-dimensional chaotic system introduced by Rössler (1976), designed to be one of the simplest continuous-time systems capable of exhibiting chaotic behavior. It is smoother than the Lorenz attractor A.4 but still produces complex, fractal-like trajectories, serving as a robust testbed for reconstructing higher-dimensional chaotic states from lower-dimensional measurements. The attractor dynamics are defined as:

$$\begin{cases} \dot{x}_1 = -x_2 - x_3, \\ \dot{x}_2 = x_1 + ax_2, \\ \dot{x}_3 = b + x_3(x_1 - c), \\ y = x_2, \end{cases}$$

where $x = [x_1, x_2, x_3]^\top \in \mathbb{R}^3$ is the state vector. We utilize a standard set of chaotic parameters $a = 0.1$, $b = 0.1$, and $c = 14$. The system output is $y = x_2$, requiring the observer to reconstruct the full 3-dimensional chaotic state from a single measurement channel.

[1] Code available in the following repository:
`https://github.com/yehias21/HyperKKl`

A.4 LORENZ SYSTEM

The Lorenz attractor is a seminal 3-dimensional system originally derived by Lorenz (1963) for atmospheric convection. It is arguably the most famous example of deterministic chaos, mainly known for its "butterfly effect" where sensitive dependence on initial conditions makes long-term prediction impossible without accurate state estimation. The system dynamics are given by:

$$\begin{cases} \dot{x}_1 = p(x_2 - x_1) \\ \dot{x}_2 = x_1(q - x_3) - x_2 \\ \dot{x}_3 = x_1 x_2 - r x_3 \\ y = x_2 \end{cases}$$

where $x = [x_1, x_2, x_3]^\top \in \mathbb{R}^3$ is the state vector. We use the classic parameter values $p = 10$, $q = 28$, and $r = 8/3$.

B CONNECTION TO NON-AUTONOMOUS KKL THEORY

In this appendix, we establish the theoretical foundations of the proposed HyperKKL framework. We first visit the KKL observer theory for autonomous systems and its extension to non-autonomous systems via physics-informed learning.

B.1 KKL OBSERVER THEORY FOR NON-AUTONOMOUS SYSTEMS

KKL observer relies on the immersion of a nonlinear system into a higher-dimensional linear system (Luenberger, 1964; 1966; 1971). We consider non-autonomous nonlinear systems of the form equation 4, restated here:

$$\begin{cases} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= h(x(t)) \end{cases} \tag{13}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state at time $t \in \mathbb{R}_{\geq 0}$, $x_0 \in \mathcal{X}$ is the unknown initial condition, $u(t) \in \mathbb{R}^m$ is the external control input, and $y(t) \in \mathbb{R}^{n_y}$ is the measured output. The functions $f : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}^{n_x}$ and $h : \mathcal{X} \to \mathbb{R}^{n_y}$ are assumed to be smooth. The KKL framework involves designing an observer of the following form:

$$\begin{cases} \dot{\hat{z}}(t) &= \Phi(\hat{z}(t), y(t), u(t)), \\ \hat{x}(t) &= \Psi(\hat{z}(t), y(t)) \end{cases} \tag{14}$$

where $\hat{z}(t) \in \mathbb{R}^{n_z}$ is the observer's internal state initialized at $\hat{z}(0)$, which processes the measured output and control input from equation 13 to provide an estimate $\hat{x}(t) \in \mathbb{R}^{n_x}$. Designing the observer requires choosing the continuous functions $\Phi : \mathbb{R}^{n_z} \times \mathbb{R}^{n_y} \times \mathbb{R}^m \to \mathbb{R}^{n_z}$ and $\Psi : \mathbb{R}^{n_z} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_x}$ such that the estimation error:

$$\xi(t) := x(t) - \hat{x}(t) \tag{15}$$

globally asymptotically converges to zero as $t \to \infty$, i.e., $\forall x_0 \in \mathcal{X}, \hat{z}(0) \in \mathbb{R}^{n_z} : \lim_{t\to\infty} \|\xi(t)\| = 0$.

B.2 KKL OBSERVER THEORY FOR AUTONOMOUS SYSTEMS

Originally established for autonomous systems ($\dot{x} = f(x)$), the KKL theory states the existence of an immersion mapping $\mathcal{T}$. According to Brivadis et al. (2023), if the system is *backward distinguishable*[2] and *forward complete* on a compact set $\mathcal{X}$, there exists an injective transformation $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ that maps the nonlinear dynamics into a linear latent system:

$$\dot{z}(t) = Az(t) + Bh(x(t)), \quad z(0) = \mathcal{T}(x_0) \tag{16}$$

[2]A system is backward $\mathcal{O}$-distinguishable on $\mathcal{X}$ if for every distinct pair $x_0^1, x_0^2 \in \mathcal{X}$, there exists $\tau < 0$ such that the backward solutions satisfy $h(x(\tau; x_0^1)) \neq h(x(\tau; x_0^2))$.

where $A \in \mathbb{R}^{n_z \times n_z}$ is a Hurwitz matrix and $B \in \mathbb{R}^{n_z \times n_y}$ is an output injection term, and the pair (A, B) is chosen such that it is controllable[3] and the latent dimension satisfies $n_z = n_y(2n_x + 1)$. The transformation $\mathcal{T}(x)$ is the solution to the PDE equation 2, restated here:

$$\frac{\partial \mathcal{T}}{\partial x}(x)\, f(x) = A\,\mathcal{T}(x) + B\,h(x), \quad \mathcal{T}(0) = 0 \tag{17}$$

Once $\mathcal{T}(x)$ is identified, a state estimate $\hat{x}$ can be recovered using the left-inverse $\mathcal{T}^*$ by simulating the linear observer equation 3:

$$\begin{cases} \dot{\hat{z}}(t) = A\hat{z}(t) + B\,y(t), \\ \hat{x}(t) = \mathcal{T}^*(\hat{z}(t)) \end{cases} \tag{18}$$

Since A is Hurwitz, the error $e(t) = z(t) - \hat{z}(t)$ converges asymptotically to zero (Niazi et al., 2025, Remark 4).

B.3 Extension to Non-Autonomous Systems

Extending the framework to non-autonomous systems (i.e., systems with inputs u) introduces a significant constraint, as the transformation $\mathcal{T}(x)$ must satisfy the immersion condition uniformly for all admissible inputs u. Following the physics-informed learning approach introduced in Niazi et al. (2025), we seek a transformation $\mathcal{T}$ that maps the nonlinear dynamics to a linear latent system. To ensure robust estimation, this target linear system is designed to be Bounded-Input Bounded-State (BIBS) stable. The target dynamics in the latent space are formulated as:

$$\dot{z} = A\,z(t) + B\,h(x(t)) + \bar{\varphi}(u(t)) \tag{19}$$

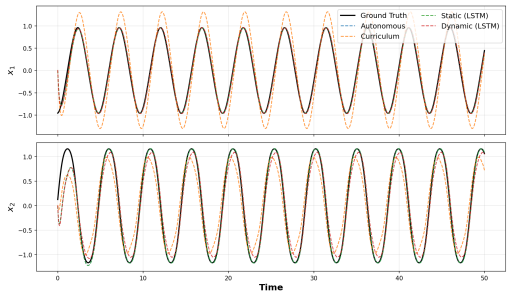
where $\bar{\varphi} : \mathbb{R}^m \to \mathbb{R}^{n_z}$ is a design parameter of choice (typically linear) representing the input injection. In the stationary approach of Section 2.1.2, this corresponds to the injection term $\bar{\varphi}(\hat{z})\,u$. Consequently, the governing PDE for the transformation becomes:

$$\frac{\partial \mathcal{T}}{\partial x}\, f(x, u) = A\,\mathcal{T}(x) + B\,h(x) + \bar{\varphi}(u) \tag{20}$$

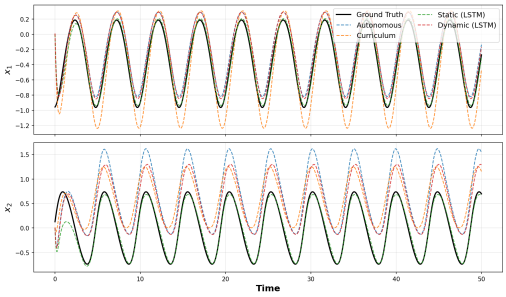
Solving the PDE in equation 20 analytically is often intractable. Recent work by Niazi et al. (2025) utilizes PINNs to approximate $\mathcal{T}(x)$ by minimizing the residual of the PDE directly, effectively turning the observer design problem into a learning problem. The Dynamic HyperKKL framework (Section 3.2.2) extends this further by allowing $\mathcal{T}$ itself to vary with the input history via the time-dependent PDE equation 5.

[3]The pair (A, B) is controllable if $\text{rank}[B\ AB\ \cdots\ A^{n_z-1}B] = n_z$.

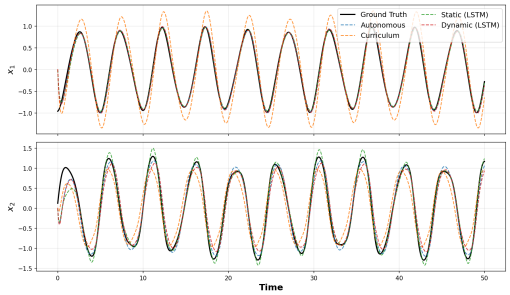
C SIMULATION RESULTS



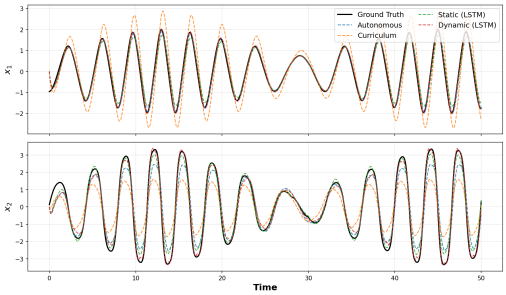
(a) Zero Input



(b) Constant Input

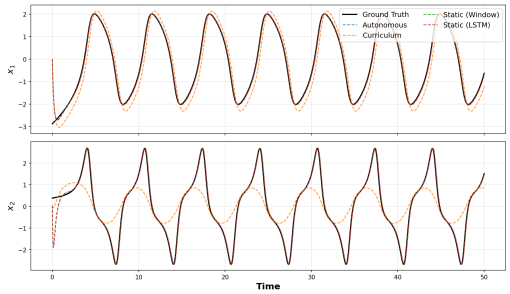


(c) Sinusoid Input

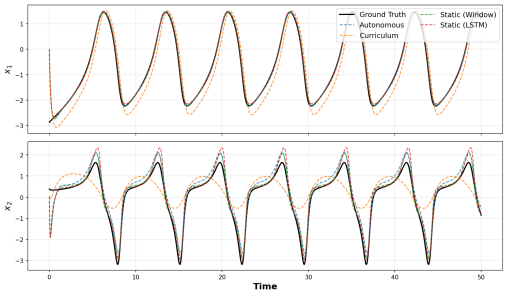


(d) Square Input

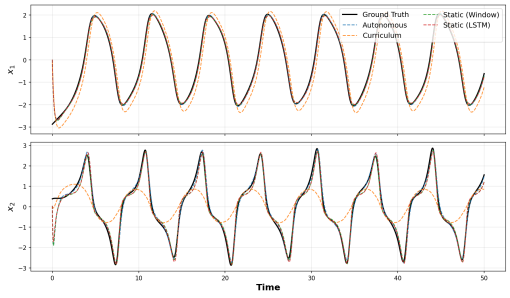
Figure 3: Duffing System: State estimation time-series.



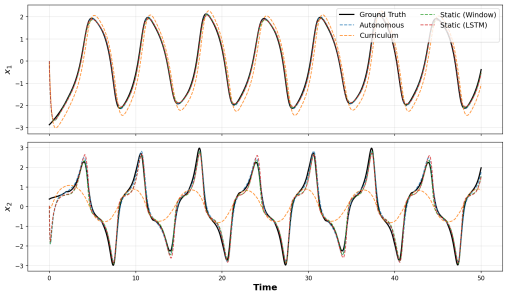
(a) Zero Input



(b) Constant Input



(c) Sinusoid Input



(d) Square Input

Figure 4: Van der Pol System: State estimation time-series.

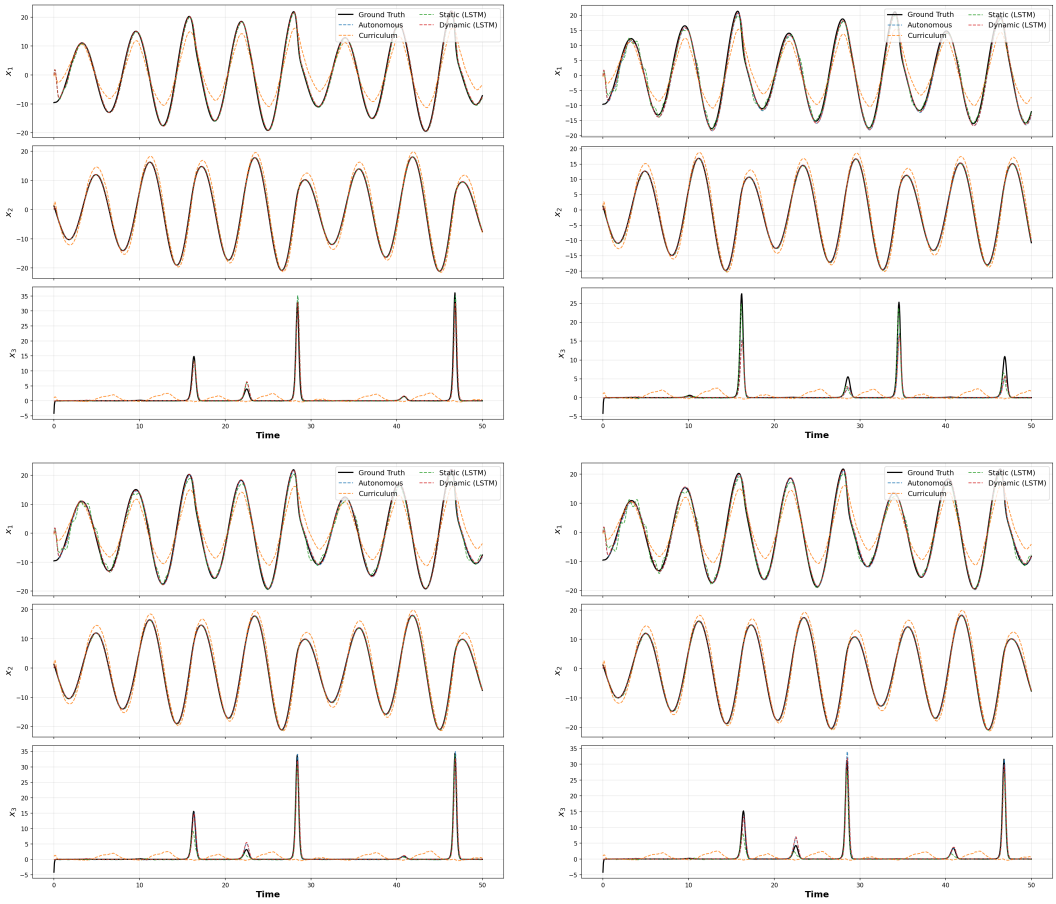


Figure 5: Rossler System (Chaotic): State estimation time-series.

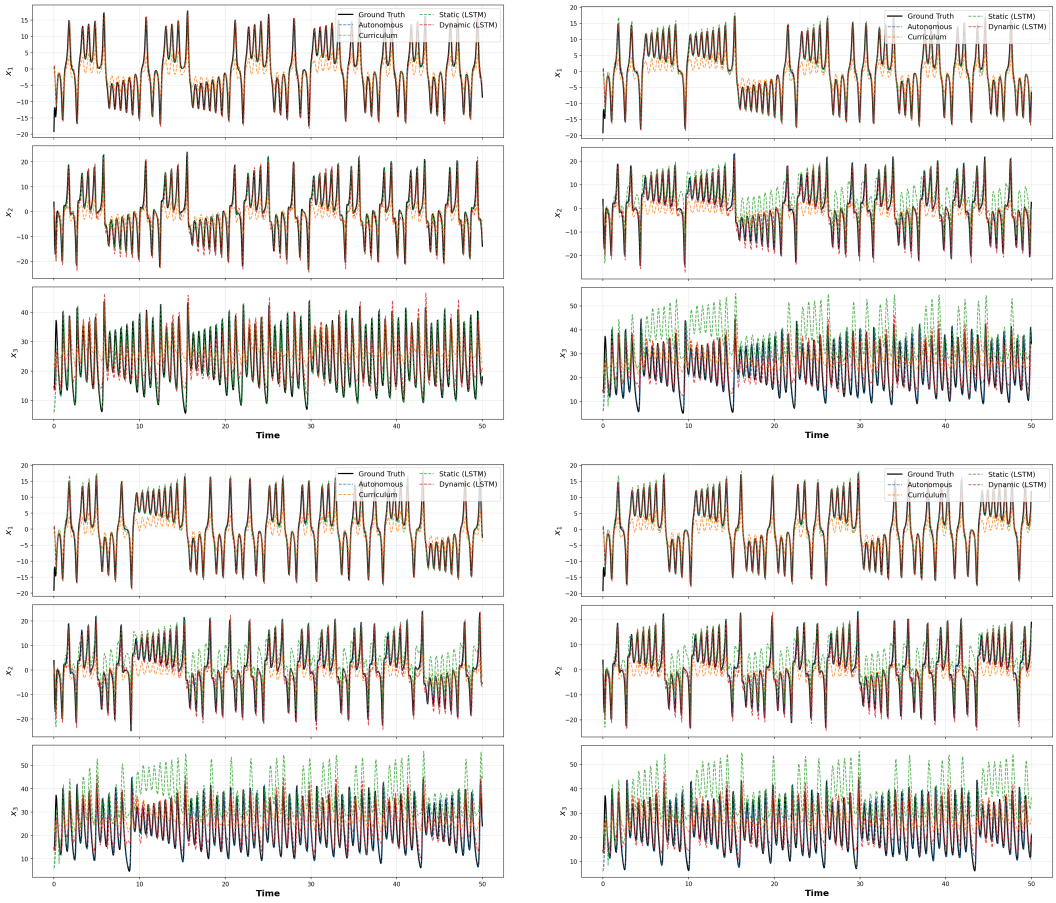


Figure 6: Lorenz System (Chaotic): State estimation time-series.