

LASS-ODE: WHEN LARGE FOUNDATION MODELS MEET SMALL UNIFIED ODE REPRESENTATIONS

Anonymous authors

Paper under double-blind review

ABSTRACT

Foundation models have transformed language and vision through large-scale attention over discrete tokens, yet progress on continuous-time dynamical signals remains limited. A core challenge is the absence of a natural token-based representation for ODE trajectories, which evolve continuously, span multiple temporal resolutions, and are often partially observed. We introduce the Tokenized ODE Representation (TOR), which maps trajectories into latent tokens governed by local and linear Neural ODEs, leveraging their linearity for efficient scaling. To capture both temporal context and shared structure across systems, we design a hybrid attention architecture that alternates intra-system self-attention, modeling dependencies within each trajectory, and inter-system cross-attention, supported by a Dynamic ODE Hub (DOH) that serves as a shared repository for inter-system knowledge exchange. These components form LASS-ODE (Large Scale Small ODE), a foundation model with strong capacity for interpolation, extrapolation, probabilistic inference, and zero-shot generalization across diverse ODE systems.

1 INTRODUCTION

Foundation models have achieved transformative progress across modalities such as text, vision, time series, and graphs by leveraging large-scale attention-based architectures (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Dosovitskiy et al., 2020; Kaplan et al., 2020; Das et al., 2024a; Xiaoming et al., 2025; Liu et al., 2025b; Mao et al., 2024; Liu et al., 2025a). At their core is the attention mechanism, which enables global token interaction and flexible context aggregation, supported by predictable scaling laws with model and data size (Kaplan et al., 2020). Yet a central paradox remains: while foundation models thrive on *discrete tokens*, many scientific and engineering domains are governed by *continuous-time dynamics* described by Ordinary Differential Equations (ODEs). These dynamics underpin simulations, forecasting, digital twins, and control in engineered systems (Vallado, 2001; Rasheed et al., 2019; Li et al., 2025b; Mai et al., 2025), Earth and environmental processes (Lorenz, 2017), and life and health systems (Qian et al., 2021). Bridging the discrete-token paradigm with continuous-time dynamical systems remains an open challenge.

Existing approaches extend Transformers to temporal data by discretizing trajectories into patches and treating them as tokens. Recent time-series foundation models have advanced this strategy by learning local short-term dynamics within each patch and using attention to capture long-range dependencies (Nie et al., 2022; Das et al., 2024a; Ansari et al., 2024a). While effective for regularly sampled signals, these models remain tied to fixed-interval updates and cannot faithfully represent the underlying ODEs. As a result, they lack guarantees of continuity, struggle with irregular or multi-resolution data, and fail to expose the physical semantics of dynamical systems. Techniques such as imputation and positional encoding can partially address irregularity (Wen et al., 2022; Zeng et al., 2023), but they do not resolve the fundamental gap: patch tokenization does not provide a principled way to encode continuous evolution.

Compared with patch-based foundation models, small-scale system-specific approaches can directly model continuous-time dynamics within a single system. Symbolic regression methods infer explicit governing equations from data, producing closed-form ODEs that can be integrated at arbitrary temporal resolutions (Brunton et al., 2016; Li et al., 2022). Physics-informed neural networks embed physical constraints into coordinate-based neural fields, allowing the learned solution to be queried at any time (Cuomo et al., 2022). The Neural ODE family instead learns the underlying vector

field and integrates it in continuous time (Chen et al., 2018b; Rubanova et al., 2019; Kidger et al., 2020b; Li et al., 2025a). These approaches can achieve high accuracy on targeted systems, yet often require case-specific tuning and lack scalability. More importantly, they do not generalize across heterogeneous systems, even though many systems share recurring ODE structures such as energy-conserving Hamiltonian forms (Greydanus et al., 2019), damping and forcing patterns, or oscillatory dynamics (Krishnapriyan et al., 2021; Rackauckas et al., 2020). This tension highlights the need for architectures that preserve the fidelity of continuous-time modeling while supporting transfer across diverse dynamical domains.

To address this gap, we introduce a new philosophy, *LASS (Large-Scale Small)*, for scalable modeling of ODE-governed systems. In contrast to conventional strategies that focus solely on scaling up large attention architectures, our philosophy emphasizes first *scaling down* to capture local physical information in small, physics-aware tokens, and then *scaling up* with global attention to integrate these tokens across trajectories and systems. Each token preserves physical constraints and provides an efficient representation of local dynamics, often through simple linear forms, while attention layers support flexible long-range interactions. This dual design—small physics-aware representations combined with large-scale token interaction—aims to achieve both fidelity to underlying physical laws and scalability across diverse datasets.

Building on this philosophy, we propose *LASS-ODE*, a scalable architecture that unifies continuous-time modeling with foundation-style generalization. Central to LASS-ODE is the *Tokenized ODE Representation (TOR)*, which encodes state trajectories as piecewise linear ODE flows in the latent space. Unlike patch tokenizers, TOR preserves continuity and interpretable eigenstructure. Moreover, the linear format brings fast ODE integration, suitable for large-scale computation. To facilitate cross-system generalization, we introduce the *Dynamic ODE Hub (DOH)*, a shared, trainable matrix that is dynamically updated from token messages and serves as a hub for inter-system knowledge exchange. Finally, we propose a novel architecture that alternates between intra- and inter-attention, enabling deep, dynamic mutual enhancement between local dynamics and shared global structures.

Our contributions are threefold:

- **Continuous-time tokenization.** We introduce the Tokenized ODE Representation (TOR), which encodes latent flows into piecewise linear tokens with guaranteed continuity, interpretable eigenstructure, and fast computations.
- **Scalable cross-system modeling.** We design LASS-ODE, a foundation model that blends physics-aware tokenization with scalable intra- and inter-attention mechanisms.
- **Empirical validation.** Across chaotic physics, biological networks, and power systems, LASS-ODE achieves consistent improvements over state-of-the-art time-series foundation models and Neural ODE-based methods.

2 PRELIMINARIES

Problem Setup. We consider families of continuous-time dynamical systems governed by ordinary differential equations (ODEs) or differential-algebraic equations (DAEs). For a given system with state $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ denotes the system’s state space, its evolution is described by a vector field $f : \mathcal{X} \rightarrow \mathcal{X}$ as $\dot{\mathbf{x}} = f(\mathbf{x})$. Our setting includes sequential state observations $\{\mathbf{x}_i(t_j^i)\}_{i,j}$ from multiple systems indexed by i , each with distinct state spaces $\{\mathcal{X}_i\}$, potentially varying in dimension, temporal resolution, and observability. For the i^{th} system, the timestamps of all observed points $\mathcal{T}_{\text{obs}}^i := \{t_j^i\}_j$ are also available. Our goal is to build a foundation model that estimates f for each system. The model supports the following downstream tasks:

- **Interpolation:** Reconstruct missing data at arbitrary times within the observed time span.
- **Extrapolation:** Predict system trajectories at future times beyond the observation interval.
- **Probabilistic trajectory inference:** Estimate distributions over trajectories to enable uncertainty-aware predictions under partial observability, noise, etc.
- **Zero-shot inference:** The model reconstructs trajectories for previously unseen systems by conditioning only on a short prefix of observed states, without access to any additional input-output pairs from the same system or parameter fine-tuning (Liu et al., 2025b).

In the following, we introduce the preliminaries for the learning models and omit the system index for notational simplicity, as all models are applied independently to each system in practice.

Latent ODE for multi-dimensionality, partial-observability, and multi-resolution. Recall that different ODE systems can be multi-dimensional, partially observed, and sampled at multiple resolutions. We first *replicate certain channels to enforce a fixed input dimensionality across systems*, but this alone does not address the underlying redundancy or missing information. We therefore *construct a latent space* that captures the core dynamical factors to reduce redundancy (tackling multi-dimensionality) and introduce a latent random variable to represent a distribution over trajectories (tackling partial observability). We then evolve this latent state using a Neural ODE to accommodate multi-resolution sampling. Together, these components motivate the use of Latent ODE (Rubanova et al., 2019). Specifically, in latent ODE, an encoder f_{enc} processes N observations $\{\mathbf{x}(t_i)\}_{i=0}^N$ and $\mathcal{T}_{\text{obs}} := \{t_i\}_{i=0}^N$ to infer an initial latent state $\mathbf{z}(t_0) \in \mathcal{Z}$ that summarizes past information. The posterior distribution over this state is parameterized as:

$$\mathbf{z}(t_0) \sim \mathcal{N}(f_{\text{enc}}(\{\mathbf{x}(t_i)\}, \mathcal{T}_{\text{obs}})). \quad (1)$$

Choice of the encoder f_{enc} for large-scale training. We adopt a GRU encoder (the orange box in the left of Fig. 1) that processes the concatenated input $[\mathbf{x}(t), \Delta t]$, where Δt denotes the sampling interval of the specific ODE system. This design enables the encoder to handle multi-resolution datasets by incorporating temporal resolution as an input. Specifically, we have:

$$f_{\text{enc}}(\{\mathbf{x}(t_i), \Delta t_i\}_{i=0}^N) = g(\mathbf{h}), \mathbf{h} = \text{GRU}_{\phi}(\{[\mathbf{x}(t_i); \Delta t_i]\}_{i=0}^N), \quad (2)$$

where g is a multilayer perceptron (MLP) to convert $\mathbf{h}$ into the mean and variance of $\mathbf{z}(t_0)$. Other options for irregularly sampled data, such as RNN- Δ_t (Che et al., 2018) and ODE-RNN (Rubanova et al., 2019), incur heavier computational costs and prevent the use of highly optimized batched GRU kernels, which leads to significantly lower GPU throughput.

Given $\mathbf{z}(t_0)$, the decoder evolves the latent trajectory via an ODE parameterized by a neural network h_{θ} , and reconstructs/predicts the observed states using a likelihood model p_{θ} :

$$\mathbf{z}(t_i) = \text{ODESolve}(h_{\theta}, \mathbf{z}(t_0), t_0, t_i), \quad \mathbf{x}(t_i) \sim p_{\theta}(\mathbf{x}(t_i) | \mathbf{z}(t_i)), \quad (3)$$

Here, $\text{ODESolve}(h_{\theta}, \mathbf{z}(t_0), t_0, t_i)$ solves the initial value problem (IVP) defined by the neural ODE $\dot{\mathbf{z}}(t) = h_{\theta}(\mathbf{z}(t))$ with initial condition $\mathbf{z}(t_0)$, and returns the integrated latent state at time t_i . The decoder thus consists of both the ODE function h_{θ} and the probabilistic head p_{θ} . Training maximizes the variational lower bound:

$$\text{ELBO} = \mathbb{E}_{\mathbf{z}(t_0)} [\log p_{\theta}(\{\mathbf{x}(t_i)\})] - \text{KL} [\mathcal{N}(f_{\text{enc}}(\{\mathbf{x}(t_i)\}, \mathcal{T}_{\text{obs}})) \parallel p(\mathbf{z}(t_0))], \quad (4)$$

where $p(\mathbf{z}(t_0))$ is typically a standard Gaussian prior. In deterministic settings, the likelihood term can be replaced with a squared-error loss.

Multi-head attention for scalable interactions. As the core computation underlying modern foundation models, multi-head attention enables the contextual interaction between embedding vectors. Given query Q , key K , and value V , the attention output is computed as:

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O, \quad \text{head}_i = \text{softmax} \left(\frac{Q W_i^Q (K W_i^K)^{\top}}{\sqrt{d_k}} \right) V W_i^V, \quad (5)$$

where d_k is the dimension of each query/key vector.

3 LASS-ODE: LARGE-SCALE SMALL ODE

3.1 TOKENIZED ODE REPRESENTATION TO SCALE DOWN LATENT ODES

Latent ODEs provide a principled way to model continuous-time dynamics, but they remain computationally expensive due to the need for numerical ODE integration at every forward pass. Moreover, the neural network h_{θ} that parameterizes the latent derivative in equation 3 is typically unstructured and difficult to interpret, which limits its scalability and generalizability across systems. In foundation models, tokens are the essential elements that support large-scale representation learning and

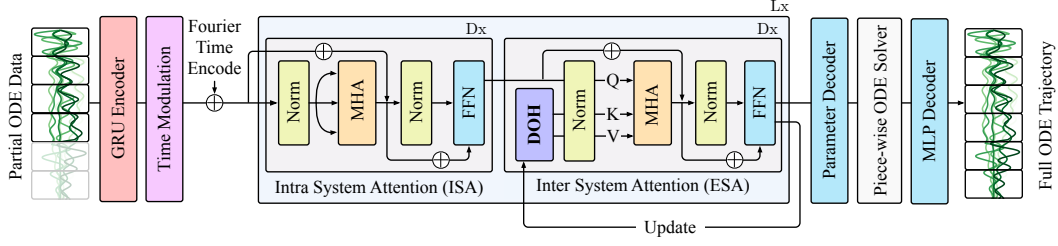


Figure 1: The architecture of LASS-ODE.

compositional reasoning. Hence, before scaling up latent ODEs to the foundation scale, we first scale them down by converting continuous latent dynamics into a small set of discrete ODE tokens. This Tokenized ODE Representation (TOR) reduces integration overhead and imposes structural regularity on each token’s dynamics, making them both interpretable and reusable as fundamental units for large-scale modeling.

Given the observations $\{x(t_i)\}_{i=0}^N$, we define that the k^{th} token is a small temporal segment of the observed trajectory over the interval $[t_k^{start}, t_k^{end}]$. The set of tokens forms a collection of non-overlapping patches of the full time-series trajectory (Nie et al., 2022) from t_0 to t_N . We now formally define the Tokenized ODE Representation (TOR), which models each trajectory patch using a token-specific latent ODE flow.

Definition 1 (Tokenized ODE Representation (TOR)). Given tokenized segments, TOR models each token’s latent evolution in a continuous space $\mathcal{Z}$ as:

$$\text{TOR}_k := \{z_k(t) \mid t \in [t_k^{start}, t_k^{end}]\}, \quad \text{where } z_k(t) = \text{ODESolve}(\dot{z}_k(t), z_k(t_k^{start}), t_k^{start}, t), \quad (6)$$

with the token-specific linear dynamics defined by:

$$\dot{z}_k(t) = A_k z_k(t) + b_k, \quad (7)$$

where $z_k(t_k^{start}) \in \mathcal{Z}$ is the initial latent state, and $A_k \in \mathbb{R}^{d \times d}$, $b_k \in \mathbb{R}^d$ are token-specific parameters. The latent segment TOR_k can be decoded via $p_\theta(\text{Token}_k \mid \text{TOR}_k)$ to accurately reconstruct the original observation patch, where p_θ and the decoding process is defined in equation 3.

The linear latent dynamics in equation 7 provide sufficient expressiveness over small token intervals, particularly when composed with the nonlinear map p_θ . TOR retains the key advantages of latent ODEs: it provides a unified representation that naturally supports multi-resolution sampling, continuous evolution, and principled uncertainty modeling through the variational latent variable framework. In addition, because of the linearity, TOR is highly efficient for large-scale computation. During decoding, the initial latent state $z(t_0)$ is encoded once using a GRU-based encoder, as depicted in equation 1 and equation 2. Then, with the shared $z(t_0)$, off-the-shelf ODE solvers such as `torchdiffeq.odeint` (Chen et al., 2018b) can efficiently solve the following piecewise linear IVP, which guarantees the continuity of the ODE trajectories.

$$\begin{aligned} h(t) &:= A(t)z(t) + b(t), \quad \text{where } A(t) := A_k, b(t) := b_k \quad \text{if } t \in [t_k^{start}, t_k^{end}], \\ z(t) &= \text{ODESolve}(h(t), z(t_0), t_0, t \in (t_0, t_{N_{\max}}]), \end{aligned} \quad (8)$$

where we define the piecewise linear function $h(t)$ to take place of the nonlinear ODE function h_θ in equation 3. Moreover, we utilize $t_{N_{\max}}$ rather than t_N to denote the user-defined maximum ending time across all ODE systems. The white box in the right part of Fig. 1 is the introduced ODE solver. After obtaining the flow $z(t)$, we can utilize an MLP decoder (see the blue box in the right of Fig. 1), namely, p_θ in equation 3, to draw samples of $x(t)$.

The introduced physical prior in TOR not only facilitates efficient modeling and computation, but also naturally motivates the use of global attention in a foundation model: each token embedding should attend to others to exchange and refine dynamic patterns, such as oscillatory modes and damping behaviors, reflected in the eigenstructure of A_k and the offset b_k (Callier & Desoer, 2012). To support TOR-based foundation models, two key questions naturally arise:

1. How can we construct an informative embedding for each token?

2. How should we design intra- and inter-ODE system attention?

The first question is addressed below, while the second is discussed in Section 3.2.

To represent each token, we summarize its temporal evolution using the hidden state in equation 2. Compared to the latent initial condition $z(t_0)$, the hidden state $\mathbf{h}$ provides a richer and higher-dimensional summary, as also noted in Rubanova et al. (2019). To further encode token-specific dynamics such as A_i and $\mathbf{b}_i$, we adopt a time modulation, inspired by (Perez et al., 2018):

$$\mathbf{e}_i = \gamma(t_i^{\text{start}}) \odot f_{\text{embed}}(\mathbf{h}) + \beta(t_i^{\text{start}}), \quad (9)$$

where $\mathbf{e}_i \in \mathbb{R}^{d_{\text{model}}}$ is the embedding vector for token i , and f_{embed} is an MLP applied to the global summary vector $\mathbf{h}$. The modulation functions $\gamma(\cdot)$ and $\beta(\cdot)$ are two independent MLPs that take the token’s starting time t_i^{start} as input and output vectors in $\mathbb{R}^{d_{\text{model}}}$. The operator $\odot$ denotes Hadamard multiplication. By modulating the shared representation $f_{\text{embed}}(\mathbf{h})$ with time-specific scaling and shifting functions $\gamma(t_i^{\text{start}})$ and $\beta(t_i^{\text{start}})$, the model can adaptively adjust the embedding for each token to reflect its unique temporal context. The time modulation is presented as the pink box to the left of Fig. 1.

Since the modulation is conditioned only on the token’s starting time t_i^{start} , we can generate the full trajectory of embedding vectors, spanning from the beginning t_0 to the defined maximum time $t_{N_{\text{max}}}$, which may extend beyond the end of the observational window t_N . As long as the local dynamics A_k and $\mathbf{b}_k$ can be accurately estimated from the token embedding (and the subsequent attention), the entire trajectory can be reconstructed by equation 8. This design enables full trajectory reconstruction in a sequence-to-sequence (seq-to-seq) manner, eliminating the need for auto-regressive (AR) next-token prediction (Rubanova et al., 2019).

Remark: Seq-to-seq vs. AR. AR generation and next-token prediction have proven highly effective in LLMs. However, in the context of latent dynamics, applying AR introduces a significant computational drawback. Specifically, if we predict the next token in an AR fashion and feed it back as input to the GRU encoder, the latent initial condition z_0 in equation 1 may change at each step. As a result, the ODE in equation 8 must be re-integrated repeatedly from a new initial condition. In contrast, our seq-to-seq design enables one-pass ODE integration in each forward round.

3.2 INTRA- AND INTER-SYSTEM ATTENTION TO SCALE UP TORS

Intra-system attention. We begin by modeling intra-system attention over the full trajectory of tokens. Let K_{max} denote the number of tokens spanning from the initial time t_0 to the extrapolated horizon $t_{N_{\text{max}}}$. Let $E \in \mathbb{R}^{K_{\text{max}} \times d_{\text{model}}}$ denote the matrix of token embeddings $\{\mathbf{e}_i\}$, constructed via time modulation as described in equation 9. To further enrich temporal representations, we augment each token embedding with a Fourier feature encoding of its starting time, denoted by $\text{FF}(t_i^{\text{start}})$ (Tancik et al., 2020). Let $\text{FF} \in \mathbb{R}^{K_{\text{max}} \times d_{\text{model}}}$ denote the matrix formed by stacking these encodings for all tokens. These Fourier features enable the model to capture both low- and high-frequency temporal patterns, which are essential for modeling complex and potentially oscillatory system dynamics. The resulting embeddings are processed through a multi-head self-attention (MHA) mechanism, as defined in equation 5, where each token attends to all others within the same trajectory to capture intra-system dependencies. To improve training stability, we adopt the pre-normalization (pre-LN) Transformer (Xiong et al., 2020). Formally, the *intra-system attention module* is defined as:

$$E^{(1)} = E + \text{FF} + \text{MHA}(\text{LN}(E + \text{FF}), \text{LN}(E + \text{FF}), \text{LN}(E + \text{FF})), \quad (10)$$

$$\tilde{E} = E^{(1)} + \text{FFN}(\text{LN}(E^{(1)})), \quad (11)$$

where LN denotes layer normalization, and the FFN is a position-wise feedforward network. $\tilde{E} \in \mathbb{R}^{K_{\text{max}} \times d_{\text{model}}}$ is the output. In general, the module is denoted as $\tilde{E} = \text{ISA}(E)$.

Inter-system attention. Unlike intra-system attention, inter-system interaction lacks an inherent sequential structure. To enable cross-trajectory communication in a permutation-invariant manner, we draw inspiration from Set Attention (Lee et al., 2019) and Slot Attention (Locatello et al., 2020).

Specifically, we introduce a *Dynamic ODE Hub (DOH)* as the central mechanism. The DOH serves two key roles: (1) it interacts with token embeddings from all systems to extract shared structure and dependencies, and (2) it provides a higher-level abstraction that dynamically responds to incoming token information, enabling cross-system coordination.

Let $H \in \mathbb{R}^{S \times d_{\text{model}}}$ denote a trainable parameter matrix (i.e., a hub) for DOH, where S is a hyperparameter to control the size of H . We perform cross-attention from tokens to the DOH: each token embedding queries the hub to extract information. Formally, the inter-system attention module is defined as:

$$E^{(1)} = E + \text{MHA}(\text{LN}(E), \text{LN}(H), \text{LN}(H)), \quad (12)$$

$$\tilde{E} = E^{(1)} + \text{FFN}(\text{LN}(E^{(1)})), \quad (13)$$

where $\text{LN}(\cdot)$ denotes layer normalization, $\text{MHA}(\cdot, \cdot, \cdot)$ is multi-head attention, and $\text{FFN}(\cdot)$ is a position-wise feedforward network. We denote the overall module as: $\tilde{E} = \text{ESA}(E, H)$. While the static H stores useful global patterns, its expressiveness becomes limited as the ISA and ESA modules are stacked deeper. To capture higher-level abstractions, we promote dynamic updates to H . Similar to (Locatello et al., 2020), we allow each hub to evolve based on interactions with token embeddings via cross-attention, and update the hub using a GRU mechanism:

$$\tilde{H} = \text{GRU}(\text{MHA}(\text{LN}(H), \text{LN}(E), \text{LN}(E)), H), \quad (14)$$

where the GRU treats H as the hidden state and updates it using the cross-attention output as input. This cross-attention allows each hub to selectively aggregate information from all token embeddings. We denote this GRU-based update as $\tilde{H} = \text{Update}(H, E)$.

Alternating ISA, ESA, and Hub Update. With the intra-system attention module (**ISA**), inter-system attention module (**ESA**), and the dynamic hub update mechanism (**Update**), we construct the fundamental building block of our architecture. The core idea is to alternate between ISA and ESA to enable bidirectional information flow between local temporal structures and global coordination. We then apply a gating mechanism $G = \text{Gate}(E, E_{\text{inter}})$ to adaptively fuse the intra-attended embeddings E with the DOH-modulated context E_{inter} , enabling selective integration. Finally, the DOH is updated via the **Update** operation to reflect higher-level abstractions and to better support subsequent attention layers. This ISA-ESA-Update pattern forms the core iterative structure of our model, as summarized in Algorithm 1. The middle of Fig. 1 visualizes this process.

Algorithm 1 ISA-ESA-Update

Require: Token embeddings $E \in \mathbb{R}^{K_{\text{max}} \times d_{\text{model}}}$, initial DOH $H \in \mathbb{R}^{S \times d_{\text{model}}}$

Require: Number of blocks L , number of attention layers per block D

```

1: for block  $\ell = 1$  to  $L$  do
2:   for layer  $d = 1$  to  $D$  do
3:      $E \leftarrow \text{ISA}(E)$  ▷ Intra-system self-attention
4:   end for
5:   for layer  $d = 1$  to  $D$  do
6:      $E_{\text{inter}} \leftarrow \text{ESA}(E, H)$  ▷ Inter-system attention from DOH
7:   end for
8:    $G \leftarrow \text{Gate}(E, E_{\text{inter}})$  ▷ Gating function over token and hub context
9:    $E \leftarrow E + G \odot E_{\text{inter}}$  ▷ Gated fusion
10:   $H \leftarrow \text{Update}(H, E)$  ▷ Update hub via GRU
11: end for
12: return Final token features  $\tilde{E} \leftarrow E$ 

```

3.3 TRAINING WITH PREFIXES TO SUPPORT ZERO-SHOT INFERENCE

As discussed in the last paragraph of Section 3.1, we abandon the AR formulation due to the intractability of repeatedly integrating large-scale ODEs from updated initial conditions. Instead, we adopt a seq-to-seq approach. However, this design choice may compromise the model’s ability to generalize in zero-shot settings, where only a limited temporal prefix is available at test time. To address this, we introduce *prefix-based training* to explicitly teach the model to operate under variable-length observation windows.

Concretely, we simulate zero-shot scenarios during training by varying the prefix length of the input trajectory. This affects only the encoder, i.e., the GRU- Δt module, which processes the sequence up to the specified prefix endpoint. We extract the hidden state $\mathbf{h}$ at the final prefix time as the summary representation. Since the GRU is causal and recurrent, we can compute all prefix representations in a single forward pass, ensuring efficiency. Formally, let $\{t_m^{\text{prefix}}\}_{m=1}^M$ denote a set of prefix times, where each $t_m^{\text{prefix}} \in [t_0, t_N]$. The encoder GRU- Δt processes the full input sequence $\{[x(t_i); \Delta t_i]\}_{i=0}^N$, and we collect the hidden states at these prefix times as $\{\mathbf{h}_m\}_{m=1}^M$. Correspondingly, the initial latent states for decoding are defined as $\{z^{(m)}(t_0)\}_{m=1}^M$, computed through equation 1 and equation 2.

Given the prefix-specific GRU hidden state $\mathbf{h}_m$ and its corresponding latent initialization $z^{(m)}(t_0)$, we generate token embeddings E using the time-modulated encoding defined in equation 9. These embeddings are then refined through intra- and inter-system attention using the core attention block described in Algorithm 1, resulting in the final token representations $\tilde{E}$.

Next, we introduce a parameter decoder f_{param} that maps each refined token embedding $\tilde{e}_i$ to a local linear dynamic operator (A_i, b_i) , representing the parameters of a piecewise latent ODE:

$$(A_i, b_i) = f_{\text{param}}(\tilde{e}_i), \quad \text{for } i = 1, \dots, K_{\text{max}}. \quad (15)$$

The parameter decoder is shown as the blue box in Fig. 1. The latent trajectory $z^{(m)}(t)$ is then obtained by integrating the piecewise TOR defined in equation 8, generating the full latent evolution over the interval $[t_0, t_{N_{\text{max}}}]$. Finally, the latent trajectory is decoded into the observable space using equation 3, producing the reconstructed trajectory $x^{(m)}(t)$. This process enables *probabilistic interpolation and extrapolation*, where the model generates full trajectory reconstructions conditioned on latent dynamics. To train the model, we maximize ELBO defined in equation 4, which is evaluated over the complete time range from t_0 to $t_{N_{\text{max}}}$.

4 EXPERIMENTS

4.1 DATASETS

We evaluate LASS-ODE on a curated dataset across diverse real-world ODE systems, as summarized in Figure 2. Representative training trajectories are provided in Appendix A.2.

For power systems, we study transient stability by integrating swing-equation and higher-order generator-network ODEs using ANDES for scripted nonlinear dynamics under perturbations (Huang et al., 2021), and Dynawo for utility-grade short- and long-term studies (Guironnet et al., 2018). To probe nonlinear control, we adopt benchmark tasks from the `neural_clbf` library (Amos et al., 2022), which emphasize stability and safety under Lyapunov-Barrier constraints. Beyond engineered systems, we source mechanistic models from the BioModels repository (Malik-Sheriff et al., 2020), a curated collection of literature-based ODE models in systems biology, and run time-course simulations for reproducibility. We also evaluate on the “Datasets for learning of unknown characteristics of dynamical systems” (Szczęśna et al., 2023), which comprises 33,000 standardized time series from 15 chaotic and non-chaotic systems with randomized initial conditions. Finally, we include general nonlinear signals from (Augustyn, 2020), covering diverse dynamical systems with partially known or unknown governing equations.

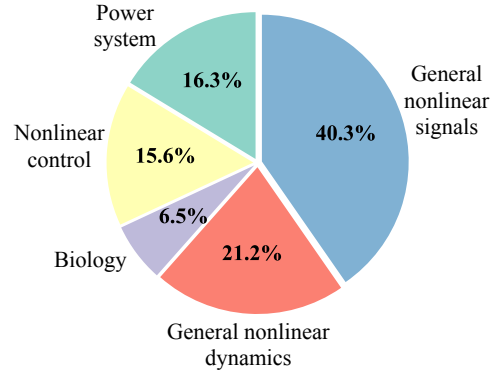


Figure 2: Ratios of data sources in our collected ODE benchmark.

4.2 IN-DOMAIN INTERPOLATION AND EXTRAPOLATION

To evaluate in-domain inference, we construct test prefixes from ODE trajectories generated using the same underlying equations as the training data, but with different initial conditions and parameter values. Each prefix randomly spans between 10% and 90% of the full trajectory duration, covering a subinterval of $[t_0, t_{N_{\max}}]$. The computed average Mean Square Error (MSE) is reported in Table 1. We compare our method with both foundation time-series models (TimeFM (Das et al., 2024a), TimeGPT (Garza et al., 2023a), and Chronos (Ansari et al., 2024a)) and Neural ODE-based small-scale methods (Latent ODE (Rubanova et al., 2019) and ContiFomer (Chen et al., 2023)).

Across diverse datasets, LASS-ODE achieves a 50–90% reduction in error compared to competing methods. Firstly, we observe that time-series foundation models are not suitable for inferring ODE measurements since ODE trajectories often exhibit non-periodic behaviors, sharp spikes, oscillatory trends, and irregular patterns (see Appendix A.2 and A.3) that cannot be captured by fixed patch-based tokenization or discrete updates. Secondly, Neural ODE-based methods achieve relatively strong performance within individual systems, highlighting the potential of latent spaces with continuous ODE flows, including our proposed TOR. However, strong in-domain accuracy does not necessarily translate to inter-system generalization. In the next subsection, we therefore evaluate the zero-shot generalization capacity of all methods.

Table 1: Test MSE ($\times 10^{-2}$) for in-domain systems.

Data	TimesFM		TimeGPT		Chronos		ContiFomer		Latent ODE		LASS-ODE	
	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra
Synthetic Nonlinear ODE	-	0.74	-	0.72	-	0.90	0.48	0.62	0.38	0.19	0.08	0.09
Power System Swing Equation	-	7.45	-	6.70	-	8.94	1.45	1.63	1.81	1.90	0.67	0.74
Linear Fading System	-	1.49	-	1.24	-	2.23	1.46	1.62	1.13	1.14	0.11	0.12
Harmonic Oscillator	-	1.93	-	1.55	-	2.26	0.82	0.34	0.45	0.36	0.12	0.14
Ueda Oscillator	-	2.53	-	1.92	-	2.88	0.91	0.47	0.71	0.43	0.14	0.16
Lorenz Attractor	-	5.44	-	3.96	-	6.41	0.96	0.38	0.53	0.67	0.27	0.30
Rucklidge Attractor	-	8.02	-	5.39	-	9.67	0.63	1.17	0.82	0.86	0.35	0.38
Rössler Attractor	-	15.24	-	7.62	-	19.05	1.85	1.74	1.69	1.77	0.56	0.63

4.3 ZERO-SHOT GENERALIZATION TO UNSEEN SYSTEMS

Table 2: Test MSE ($\times 10^{-2}$) for unseen systems.

Data	TimesFM		TimeGPT		Chronos		ContiFomer		Latent ODE		LASS-ODE	
	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra	Interp	Extra
Spiral	-	0.38	-	0.17	-	0.29	0.35	0.47	0.17	0.16	0.03	0.05
Glycolytic	-	15.77	-	23.34	-	22.79	36.39	46.98	24.15	26.90	2.35	4.08
Lotka	-	13.37	-	17.74	-	13.87	14.09	16.28	13.89	14.97	2.78	3.65
SIR Epidemic Model	-	2.67	-	2.14	-	2.77	3.55	3.98	3.08	3.17	0.77	1.06
Double Pendulum	-	57.98	-	55.09	-	62.89	97.82	95.69	74.63	82.23	8.23	10.22
FitzHugh–Nagumo Model	-	23.96	-	25.55	-	28.63	31.77	35.25	20.03	18.65	5.82	7.63

To evaluate zero-shot generalization, we use the first 30% of each trajectory as input for interpolation and extrapolation tasks. Results are reported in Table 2. Under this setting, ContiFomer and Latent ODE perform poorly on several complex systems, such as Glycolytic dynamics, the double pendulum, and the FitzHugh–Nagumo model. This limitation arises from their lack of scale-down capacity: they process trajectories sequentially with fixed patterns, making it difficult to adapt when trajectory segments differ across systems. In contrast, LASS-ODE decomposes trajectories into TORs and leverages attention-based interactions between them. This design allows LASS-ODE to maintain low MSE even in few-shot scenarios.

4.4 ABLATION STUDY

To assess the contribution of each component, we conduct an ablation study. We test the zero-shot results for different systems and compute the average MSE. As shown in Fig. 3, inter-attention plays a central role in enabling information exchange across different ODE systems, while intra-attention is essential for token contextualization within a trajectory. In addition, removing the time modulation layer leads to a notable increase in MSE. This is because that time modulation in equation 9 creates proper embedding by effectively integrating the temporal feature h with time.

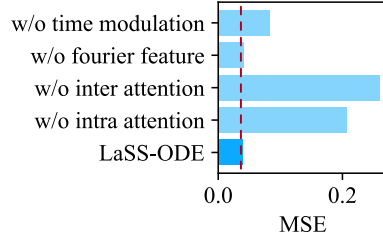


Figure 3: Results in ablation studies.

5 RELATED WORK

Foundation models for time series. Large-scale pretraining has recently enabled time-series models with cross-domain generalization. TIMEGPT introduced a GPT-style decoder for zero-shot forecasting across diverse corpora (Garza et al., 2023b), and LAG-LLAMA scaled decoder-only transformers with lagged covariates for probabilistic prediction (Rasul et al., 2023). MOIRAI unified training across frequencies and variables with any-variate attention (Liu et al., 2024), while sparse MoE extensions reduce reliance on handcrafted frequency heuristics. Tokenization-based approaches such as CHRONOS and TIMESFM leverage scaling-and-quantization or flexible horizon objectives to achieve strong zero-shot accuracy across benchmarks (Ansari et al., 2024b; Das et al., 2024b). Collectively, these models demonstrate the promise of foundation-style architectures for time series, but they almost exclusively cast sequences into symbolic tokens, overlooking domain-specific inductive biases such as ODE/DAE structure. Our work addresses this gap by developing a foundation-style model that explicitly incorporates continuous-time ODE dynamics and enable transfer across heterogeneous dynamical systems.

Neural ODE/CDE/SDE families for learning dynamics. Continuous-time neural architectures offer a complementary line of work. Neural ODEs (Chen et al., 2018a) introduced black-box differential solvers as network layers, enabling flexible vector-field learning. Extensions such as Latent ODEs and ODE-RNNs (Rubanova et al., 2019) infer latent initial states from irregular observations; Neural CDEs generalize recurrent networks to pathwise control signals (Kidger et al., 2020a); and Neural RDEs extend to rough paths (Morrill et al., 2021). These families establish the feasibility of learning dynamics directly in continuous time, but they are typically trained as narrow, system-specific models without cross-domain scalability. Our contribution is to integrate this expressivity into a foundation model for ODEs/DAEs, combining universal pretraining and retrieval with continuous-time architectures to achieve systematic generalization across dynamical regimes under partial observability and noise.

6 CONCLUSION, LIMITATION, AND FUTURE WORK

We introduced LASS-ODE, a new foundation model paradigm that unifies large-scale attention with small-scale ODE representations. By tokenizing continuous-time trajectories into piecewise linear latent ODEs, our Tokenized ODE Representation (TOR) enables efficient, interpretable, and reusable modeling units. Together with intra-system self-attention and inter-system cross-attention via a Dynamic ODE Hub, LASS-ODE achieves strong performance across interpolation, extrapolation, probabilistic inference, and zero-shot generalization tasks. Experiments on diverse ODE and DAE datasets demonstrate that our approach bridges the gap between task-specific continuous-time models and general-purpose foundation models. Despite these advances, our work has limitations. Currently, we focus on autonomous systems. On the application side, extending LASS-ODE to digital twins, control-theoretic tasks, and multi-modal settings (e.g., coupling ODE dynamics with text or sensor metadata) offers promising avenues. More broadly, we envision LASS-ODE as a stepping stone toward foundation models that combine the universality of large-scale pretraining with the fidelity of continuous-time physical modeling. For example, our next step is to provide products of LASS-SDE or LASS-PDE.

REFERENCES

- Brandon Amos, Ivan Rodriguez, Ching-An Li, and J Zico Kolter. Learning neural control lyapunov-barrier functions with a class of neural networks. In *Proceedings of the 61st IEEE Conference on Decision and Control (CDC)*, 2022.
- Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024a.
- Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024b.
- Dariusz Augustyn. Datasets for learning of unknown characteristics of dynamical systems. <https://gitlab.com/draugustyn/signal-data>, 2020.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- Frank M Callier and Charles A Desoer. *Linear system theory*. Springer Science & Business Media, 2012.
- Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018a.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018b.
- Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.
- Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3):88, 2022.
- Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024a.
- Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024b.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023a.

- Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023b.
- Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.
- Adrien Guironnet, Marianne Saugier, Sebastien Petitrenaud, Florent Xavier, and Patrick Panciatucci. Towards an open-source solution using modelica for time-domain simulation of power systems. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, oct 2018. doi: 10.1109/isgteurope.2018.8571872. URL <https://doi.org/10.1109%2Fisgteurope.2018.8571872>.
- Chenshuang Huang, Kai Wang, Chaojie Lin, and Javad Khazaei. Andes: a python library for power system modeling and simulation. *SoftwareX*, 15:100747, 2021.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020a.
- Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020b.
- Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosior, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019.
- Haoran Li, Yang Weng, and Hanghang Tong. Console: Convex neural symbolic learning. *Advances in Neural Information Processing Systems*, 35:5766–5774, 2022.
- Haoran Li, Muhao Guo, Yang Weng, and Hanghang Tong. Neural predictive control to coordinate discrete-and continuous-time models for time-series analysis with control-theoretical improvements. *arXiv preprint arXiv:2508.01833*, 2025a.
- Haoran Li, Zhihao Ma, Yang Weng, Haiwang Zhong, and Xiaodong Zheng. Low-dimensional ode embedding to convert low-resolution meters into “virtual” pmus. *IEEE Transactions on Power Systems*, 40(2):1439–1451, 2025b. doi: 10.1109/TPWRS.2024.3427637.
- Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025a.
- Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024.
- Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025b.
- Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.
- Edward N Lorenz. Deterministic nonperiodic flow 1. In *Universality in Chaos, 2nd edition*, pp. 367–378. Routledge, 2017.

- Lihao Mai, Haoran Li, Yang Weng, Erik Blasch, and Xiaodong Zheng. Guaranteed conversion from static measurements into dynamic ones based on manifold feature interpolation. *IEEE Transactions on Power Systems*, 40(5):3763–3777, 2025. doi: 10.1109/TPWRS.2025.3540724.
- Rahuman S. Malik-Sheriff, Mihai Glont, Tung V. N. Nguyen, Krishna Tiwari, Matthew G. Roberts, Ashley Xavier, and et al. Biomodels—15 years of sharing computational models in life science. *Nucleic Acids Research*, 48(D1):D407–D415, 2020. doi: 10.1093/nar/gkz1055.
- Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Forty-first International Conference on Machine Learning*, 2024.
- James Morrill, Patrick Kidger, James Foster, and Terry Lyons. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pp. 7589–7599. PMLR, 2021.
- Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.
- Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- Zhaozhi Qian, William Zame, Lucas Fleuren, Paul Elbers, and Mihaela van der Schaar. Integrating expert odes into neural odes: pharmacology and disease progression. *Advances in Neural Information Processing Systems*, 34:11364–11383, 2021.
- Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.
- Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers. *arXiv preprint arXiv:1910.01719*, 2019.
- Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.
- Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.
- Agnieszka Szczesna, Dariusz Augustyn, Katarzyna Hareźlak, Henryk Josiński, Adam Świtoński, and Paweł Kasprowski. Datasets for learning of unknown characteristics of dynamical systems. *Scientific Data*, 10(79), 2023. doi: 10.1038/s41597-023-01978-7.
- Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.
- David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.
- Shi Xiaoming, Wang Shiyu, Nie Yuqi, Li Dianqi, Ye Zhou, Wen Qingsong, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *ICLR 2025: The Thirteenth International Conference on Learning Representations*. International Conference on Learning Representations, 2025.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pp. 10524–10533. PMLR, 2020.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

A APPENDIX

A.1 MODEL HYPERPARAMETERS AND LEARNABLE PARAMETERS

Hyperparameters (default run). Unless otherwise stated, our experiments use the following configuration:

Table 3: Model and training hyperparameters.

Category	Name	Value
Architecture	Latent dim	20
	Embed dim	256
	GRU hidden	256
	GRU layers	2
	Tokens per horizon	50
	Primitives	30
	Attention heads	8
	Attention depth (per stack)	10
	MLP ratio (FFN)	4
	Blocks (interleaved)	2
	ODE solver	dopri5
Training	Optimizer	AdamW
	Learning rate	5×10^{-4}
	Betas	(0.9, 0.95)
	Weight decay	0.05
	Batch size	32
	Epochs	200
	# Scenarios	15
	Time points	100-400

Parameter count (by component). For the default configuration above (and a small output dimension placeholder $D=8$ for the reconstruction MLP), the learnable parameter counts are:

Module	# Params	Share
Encoder (GRU- Δt)	372,260	1.98%
Time modulation embedder	381,440	2.02%
Fourier positional encoding	0	0.00%
<i>Intra</i> token self-attention stack (depth 10)	9,871,360	52.38%
<i>Inter</i> token-primitive cross-attn stack (depth 10)	7,902,720	41.93%
Parameter decoder (to A, b)	173,732	0.92%
Reconstruction MLP ($d \rightarrow D$)	3,760	0.02%
Total (LASS-ODE)	18,845,304	100%

Notes. (i) The total is dominated by the attention stacks; changing E , heads, or depth has the largest effect. (ii) The reconstruction MLP depends on D (data dimension). If D differs from 8, only that row and the total change slightly. (iii) The primitive library $H \in \mathbb{R}^{K \times E}$ is learnable and contributes linearly in $K \cdot E$.

A.2 EXAMPLES OF TRAINING TRAJECTORIES

We present representative trajectories from three dynamical systems used in our experiments. These examples highlight the diversity of behaviors in the dataset, spanning chaotic attractors, oscillatory dynamics, and converging swing equations, and provide context for the types of signals on which LASS-ODE is trained.

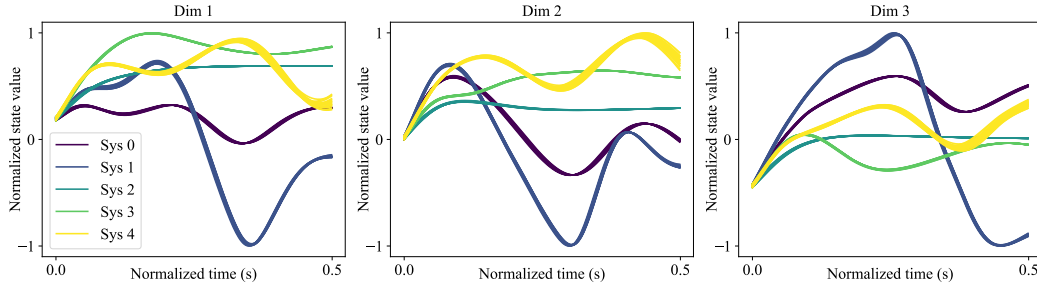


Figure 4: Dataset of general nonlinear signals (Augustyn, 2020).

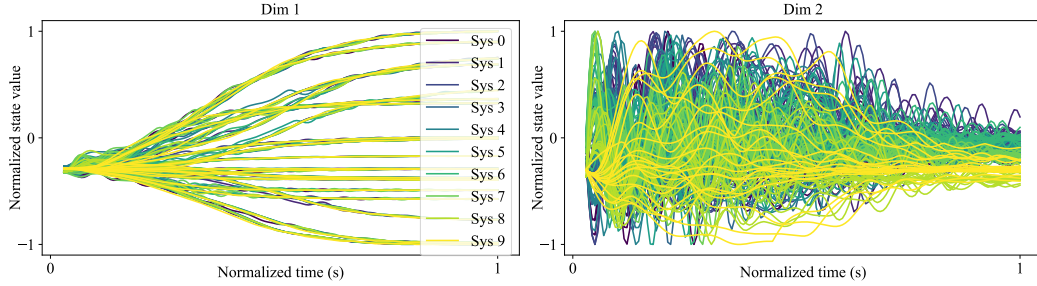


Figure 5: Dataset of power system dynamics governed by swing equations (Huang et al., 2021).

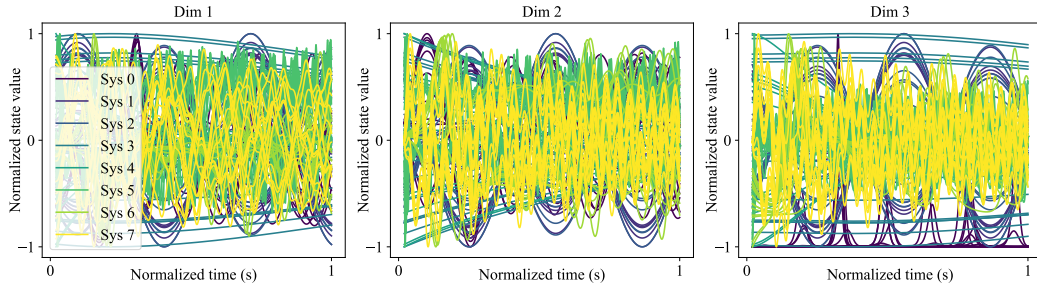


Figure 6: Dataset of general nonlinear dynamics Szczesna et al. (2023), comprising canonical examples such as the Lorenz attractor, Ueda oscillator, etc.

A.3 EXAMPLES OF FORECASTING RESULTS

To complement the training trajectories shown in Appendix A.2, we present forecasting results for the same three dynamical systems. These examples demonstrate how LASS-ODE captures distinct dynamical behaviors ranging from chaotic to oscillatory and convergent regimes, highlighting its effectiveness across diverse nonlinear ODE systems.

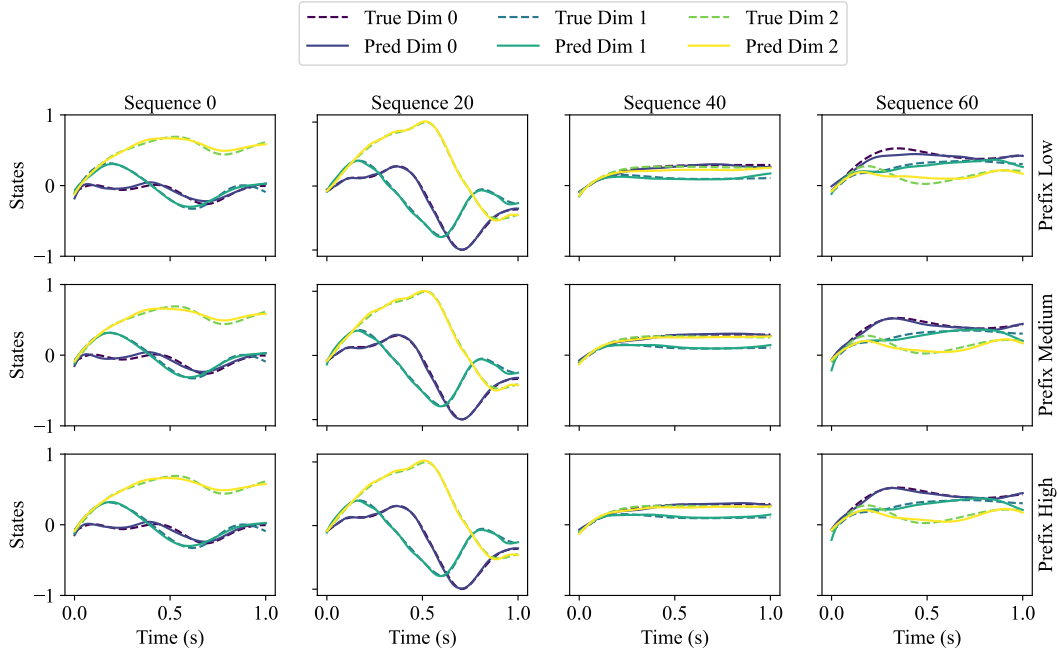


Figure 7: Forecasting results on general nonlinear signals (Augustyn, 2020).

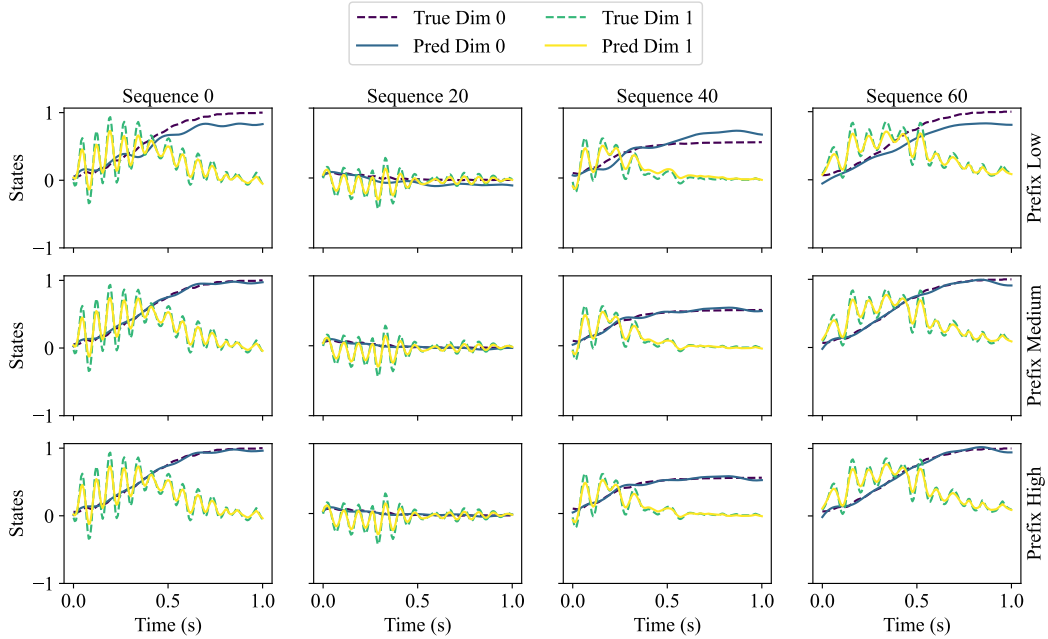


Figure 8: Forecasting results on swing-equation dynamics (Huang et al., 2021), demonstrating accuracy in convergent regimes of power systems.

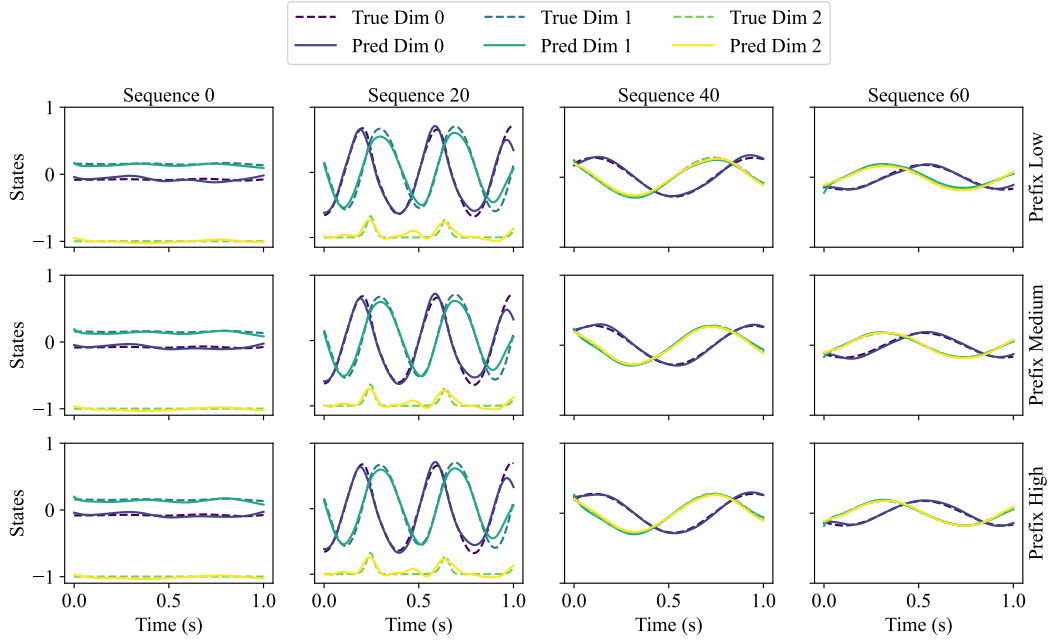


Figure 9: Forecasting results on general nonlinear dynamics (e.g., Lorenz attractor and Ueda oscillator) (Szczęsna et al., 2023; Augustyn, 2020).

A.4 FORECASTING COMPARISON ON SWING-EQUATION DATASET

To further evaluate forecasting performance, we compare LASS-ODE against recent time-series models, including TimeGPT, TimeFM, and Chronos, on the swing-equation dataset. The models are trained on system trajectories from 0–0.5 s and evaluated on the forecasting horizon 0.5–1 s. This comparison highlights the advantages of physics-informed modeling in capturing power system dynamics relative to general-purpose forecasting baselines.

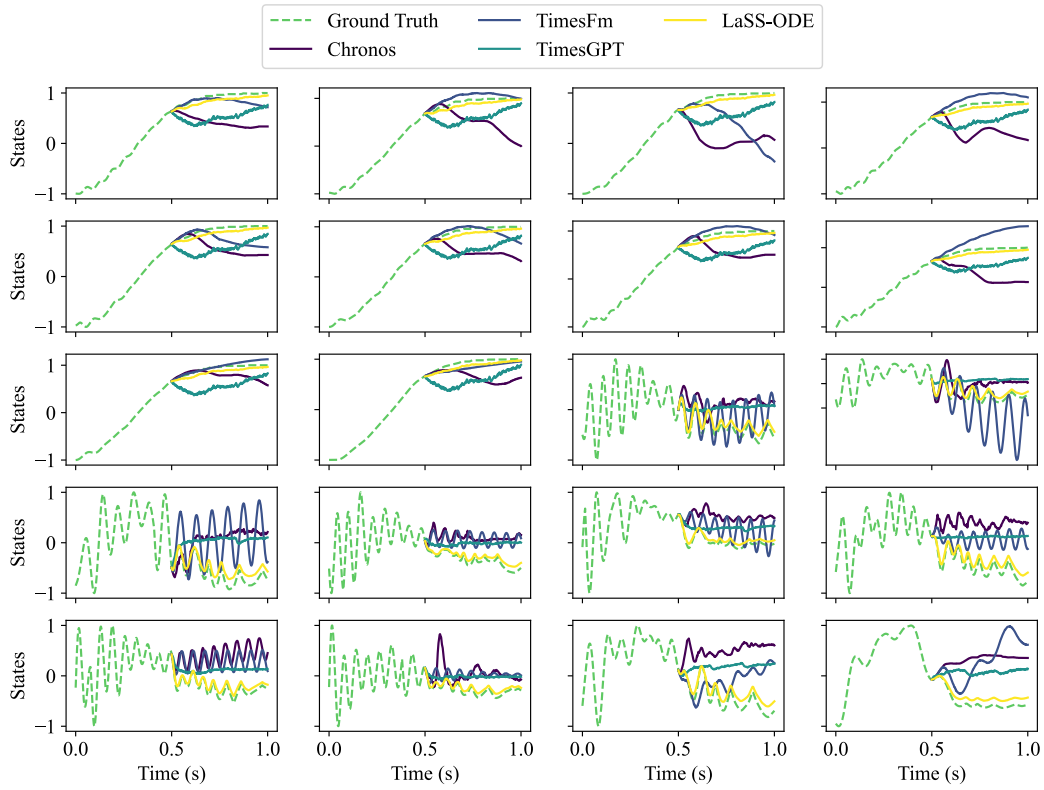


Figure 10: Forecasting comparison on the swing-equation dataset of 20 representative dimensions. Each subplot corresponds to one dimension, with ground-truth trajectories shown as dashed lines, and forecasts from LASS-ODE, TimeGPT, TimeFM, and Chronos shown as solid curves.