CHAOSNEXUS: A FOUNDATION MODEL FOR UNIVERSAL CHAOTIC SYSTEM FORECASTING WITH MULTI-SCALE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately forecasting chaotic systems, prevalent in domains including weather prediction and fluid dynamics, remains a significant scientific challenge. The inherent sensitivity of these systems to initial conditions, coupled with a scarcity of observational data, severely constrains traditional modeling approaches. Since these models are typically trained for specific systems, they lack zero-shot or few-shot capabilities on novel or data-limited scenarios. While emerging foundation models address this via pretraining on multiple systems, existing architectures typically operate at a single resolution, often failing to capture the intrinsic multi-scale temporal structures where distinct dynamical patterns unfold. To overcome this limitation, we introduce ChaosNexus, a universal forecasting model driven by our ScaleFormer architecture. It explicitly captures the multi-scale structure of chaotic dynamics with a U-Net-inspired design, enabling the simultaneous modeling of fine-grained fluctuations and coarse-grained trends. Augmented with Mixture-of-Experts layers and a wavelet-based frequency fingerprint, the model can generalizes across heterogeneous dynamical regimes. On a large-scale testbed comprising over 9,000 synthetic chaotic systems, it demonstrates notable improvements in the fidelity of long-term attractor statistics while achieving competitive point-wise forecasting accuracy compared to the leading baseline. This robust performance extends to real-world applications with exceptional data efficiency. For instance, in 5-day global weather forecasting, ChaosNexus achieves a competitive zero-shot mean error below 1°C, a result that further improves with few-shot fine-tuning. Moreover, experiments on the scaling behavior of ChaosNexus provide a guiding principle for scientific foundation models: cross-system generalization stems from the diversity of training systems, rather than sheer data volume.

REVISE

1 INTRODUCTION

Chaotic systems, characterized by their deterministic nature yet high sensitivity to initial conditions, are ubiquitous in the natural world and across diverse scientific and engineering disciplines, including weather forecasting (Shukla, 1998; Rind, 1999), fluid dynamics (Yorke & Yorke, 2005; Najm, 2009), and neural processes (Jia et al., 2023; Vignesh et al., 2025). The intrinsic complexity of such systems renders accurate forecasting both an essential and formidable task, particularly in real-world contexts where data acquisition is resource-intensive and observational records are sparse. While this sensitivity makes precise long-term point-wise prediction impossible, the system's behavior is not entirely random; it is confined to a complex geometric structure known as a strange attractor (Rössler, 1976; Grassberger & Procaccia, 1983), which possesses unique and invariant statistical properties. An effective forecasting model should not only predict the short-term evolution but also reproduce the long-term geometry and statistics of the system's attractor.

The intrinsic difficulty of forecasting chaotic systems is further compounded by the challenge of data sparsity. Traditional system-specific models (Srinivasan et al., 2022; Brenner et al., 2022; Hess et al., 2023) typically require extensive and high-quality observational data from a novel system to accurately infer its underlying dynamics and attractor geometry, creating a significant bottleneck in practical applications. This has motivated a recent paradigm shift toward pretraining a single, universal model (Jiao et al., 2025; Hemmer & Durstewitz, 2025; Lai et al., 2025), based on the proposition

REVISE

that a model exposed to a vast and heterogeneous collection of observational data spanning diverse dynamical systems and operating regimes can learn a rich repertoire of underlying patterns and principles common to chaotic behavior. By leveraging large-scale data during pretraining, such a model can then be applied to a target system with little or no in-distribution data. This strategy is designed to exploit cross-system similarities to compensate for downstream data sparsity, thereby reducing the burden of data acquisition and enhancing out-of-distribution forecasting performance.

Existing works, notably Panda (Lai et al., 2025) and DynaMix (Hemmer & Durstewitz, 2025), instantiate this paradigm through distinct architectural designs. Panda demonstrates its feasibility by pretraining Transformer blocks on a large-scale corpus of synthetic chaotic ODE systems, achieving strong zero-shot forecasts on unseen dynamical systems. DynaMix explores this direction by using a mixture of almost-linear RNN experts with delay- and sinusoidal-based embeddings to reconstruct long-term statistics of novel low-dimensional dynamics. However, individual chaotic systems exhibit multi-scale temporal structure: essential dynamical patterns unfold across a continuum of time scales, and different systems may concentrate energy in widely separated frequency bands. An architecture that operates at a single temporal resolution must either truncate long-range dependencies, oversmooth fast oscillations, or conflate behaviors that live on distinct scales, thereby obscuring system-specific attractor geometries and degrading long-horizon stability. Consequently, although Panda and DynaMix achieve strong zero-shot performance on many benchmarks, their lack of an explicit representation of this intrinsic multi-scale structure may limit out-of-distribution generalization performance when applied to more heterogeneous chaotic dynamics. **REVISE**

To overcome these obstacles, we introduce ChaosNexus, a foundation model for universal chaotic dynamics forecasting. At its core is our proposed ScaleFormer, a U-Net-inspired Transformer architecture designed to master the multi-scale nature of chaotic systems. Its encoder progressively models fine-grained to coarse temporal contexts through hierarchical patch merging, while the symmetric decoder, aided by skip connections, reconstructs fine-grained details via patch expansion. To facilitate robust cross-system generalization, each Transformer block is equipped with a Mixture-of-Experts (MoE) layer that allocates specialized parameters to different dynamical regimes on top of a shared backbone. Furthermore, we condition the model on a frequency fingerprint derived from a wavelet scattering transform, providing a stable spectral signature that captures the system's intrinsic oscillatory and modulatory behavior.

ChaosNexus is pretrained on the chaotic-system corpus introduced by Panda (Lai et al., 2025), consisting of approximately 20,000 synthetically generated ODE systems. Training is guided by a composite objective that jointly enforces short-term predictive accuracy and the preservation of long-term statistical properties. Through extensive experiments, we show that ChaosNexus sets a new state-of-the-art in zero-shot forecasting on chaotic benchmarks. Its remarkable sample efficiency is further highlighted on real-world weather forecasting: ChaosNexus achieves zero-shot temperature MAE below 1°C, outperforming competitive baselines even when they are fine-tuned on more than 470K samples from the target system. Finally, our scaling analysis reveals a key design principle for future chaotic foundation models: generalization benefits more from increasing the diversity of systems in the pretraining corpus than from increasing the number of trajectories per system. Our primary contributions are summarized as follows: **REVISE** **REVISE**

- We propose ChaosNexus, a foundation model for chaotic system forecasting strengthened by explicitly considering the multi-scale structure of chaotic dynamics, enhancing its out-of-distribution generalization performances on diverse systems.
- We design a multi-scale ScaleFormer architecture that couples hierarchical temporal representations with Mixture-of-Experts layers and a wavelet-based frequency fingerprint to capture the multi-scale temporal and spectral structure of chaotic dynamics while allocating specialized parameters to individual systems and dynamical regimes.
- We show that ChaosNexus attains state-of-the-art zero-shot performance on thousands of synthetic chaotic systems and strong zero-shot accuracy on 5-day global weather forecasting.

2 Related Works

Chaotic System Forecasting. Forecasting chaotic systems is a central challenge in science and engineering. Reservoir computing (RC)-based methods (Srinivasan et al., 2022; Gauthier et al., 2021; **REVISE**

Li et al., 2024) represent a key advance: they employ fixed read-in weights to lift inputs into the high-dimensional state space of a randomly initialized reservoir, while training only a linear readout. Concurrently, deep learning models like recurrent neural networks (RNNs) have proven effective, though they often require techniques such as teacher forcing to counteract training instabilities like exploding gradients on chaotic trajectories (Brenner et al., 2022; Hess et al., 2023). More recent works aim to preserve the geometric and statistical properties of system attractors within neural operators. This is achieved through methods like evolution regularization with optimal transport and Maximum Mean Discrepancy (MMD), or by imposing mathematical constraints such as unitarity that leverage system ergodicity (Cheng et al., 2025; He et al., 2025). Despite their success, these frameworks are specialized models, designed and trained for a single, specific system. This inherent lack of generalization renders them impractical for real-world chaotic systems where data is often sparse and systems are unseen, precluding their application in zero-shot or few-shot forecasting.

Out-of-distribution Generalization in Dynamical Systems. Out-of-distribution generalization in dynamical systems is a rapidly growing area of research. Norton et al. (2025) demonstrated that reservoir computers can generalize to unobserved basins of attraction in multistable systems when trained on sufficiently rich transient dynamics, thereby learning a global representation from a single basin. Another prominent strategy involves decomposing system dynamics into shared and specific components, where a base model captures common physical laws and low-dimensional vectors encode system-specific characteristics, leveraging data from multiple regimes to learn fundamental representations of the underlying dynamics (Brenner et al., 2024; Wang et al., 2025; Huang et al., 2023). A complementary paradigm focuses on pretraining foundation models on large synthetic datasets encompassing diverse governing equations, parameter regimes, and initial conditions (Nzoyem et al., 2025; Subramanian et al., 2023; Herde et al., 2024; McCabe et al., 2024; Seifner et al., 2024), and most of these works target PDEs with rich spatiotemporal structure. Within the domain of ODE-based chaotic systems, Panda (Lai et al., 2025) trains Transformer blocks on a large-scale corpus of synthetic chaotic systems and demonstrates strong zero-shot forecasting performance on many unseen systems. DynaMix (Hemmer & Durstewitz, 2025) instead employs a mixture of almost-linear RNN experts with delay- and sinusoidal-based embeddings to reconstruct long-term statistics of chaotic dynamics. Although these works clearly demonstrate the benefits of pretraining for generalization, their architectural designs largely overlook the inherent multi-scale temporal structure of chaotic dynamics. In contrast, we propose a U-Net–inspired multi-scale Transformer backbone, ScaleFormer, equipped with per-scale MoE layers and a wavelet-based frequency fingerprint, which explicitly encodes multi-scale temporal and spectral structure and improves out-of-distribution generalization across thousands of heterogeneous chaotic systems.

REVISE

3 METHODOLOGY

Problem Statement and Model Overview. We address the problem of chaotic system forecasting: given historical observations $\boldsymbol{X}_{1:T} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T) \in \mathbb{R}^{T\times V}$ spanning T times of a chaotic system with V variables, we forecast its successive H steps, *i.e.,* $\hat{\boldsymbol{X}}_{T+1:T+H} = f_\theta(\boldsymbol{X}_{1:T}) \in \mathbb{R}^{H\times V}$, where f_θ denotes the forecasting model. Here, we aim to design a foundation model f_θ that can directly produce faithful forecasting results based on historical observations, with little or no further in-distribution data required for training. We demonstrate the overall architecture of ChaosNexus in Figure 1, which comprises three key components: (i) input dynamics embedding, (ii) the ScaleFormer backbone, and (iii) frequency-enhanced joint scale readout. The details of our framework are shown as follows.

3.1 INPUT DYNAMICS EMBEDDING

In chaotic systems, instantaneous observations are often noisy and insufficient to reveal the governing dynamics. We therefore segment the input trajectory $\boldsymbol{X} \in \mathbb{R}^{T\times V}$ into $S = \lfloor \frac{T}{D} \rfloor + 1$ non-overlapped temporal patches of length D. Each patch $\boldsymbol{P} \in \mathbb{R}^{D\times V}$ encapsulates a short-time trajectory segment, thereby providing essential local dynamical context. Motivated by Koopman theory (Koopman, 1931; Mauroy et al., 2020; Brunton et al., 2021), which posits that nonlinear dynamics can be linearized by lifting them to a suitable high-dimensional space of observables, we first enrich each patch with random polynomial and Fourier features (Appendix C.1), an approach

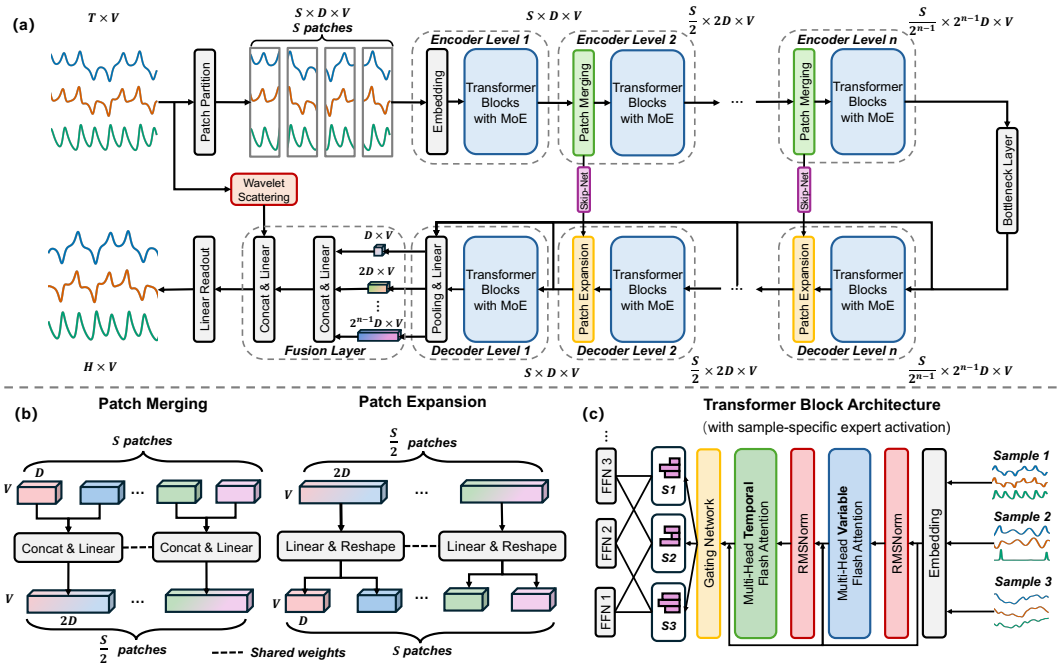

Figure 1: Overview of our ChaosNexus framework, with details of patch merging and expansion operations, and the Transformer block architecture with mixture-of-experts layers.

adopted from recent work (Lai et al., 2025). The augmented patch is then mapped to an embedding $\boldsymbol{u}$ with embedding dimension d_e via a linear layer.

3.2 ScaleFormer Architecture

The patch embeddings are then fed into the ScaleFormer, an encoder-decoder architecture composed of stacked Transformer blocks. Instead of applying standard attention to patches flattened across all dimensions with $\mathcal{O}(S^2V^2)$ complexity, each Transformer block employs dual axial attention. This mechanism factorizes the computation by performing attention sequentially along the variable and temporal axes, reducing the overall complexity to $\mathcal{O}(S^2 + V^2)$. Crucially, the variable attention module can capture the strong coupling between variables—a fundamental property of chaotic dynamics often absent in standard time series. To better accommodate different sequence lengths and enhance generalization, we employ rotary positional embeddings (RoPE) (Su et al., 2024) instead of conventional absolute positional encodings. We also employ pre-normalization to enhance training stability and FlashAttention (Dao et al., 2022) to improve efficiency. Given an input patch embedding $\boldsymbol{u}_p$, the computational flow of our modified Transformer block is:

$$\boldsymbol{h}_p = \mathrm{VA}(\mathrm{RN}(\boldsymbol{u}_p)) + \boldsymbol{u}_p, \qquad \bar{\boldsymbol{h}}_p = \mathrm{TA}(\mathrm{RN}(\boldsymbol{h}_p)) + \boldsymbol{h}_p, \qquad \tilde{\boldsymbol{h}}_p = \mathrm{MoE}(\mathrm{RN}(\bar{\boldsymbol{h}}_p)) + \bar{\boldsymbol{h}}_p, \quad (1)$$

where VA and TA are axial variable and temporal attention operations, respectively. RN denotes the root mean square (RMS) layer normalization (Zhang & Sennrich, 2019). We replace the standard feed-forward network (FFN) with a Mixture-of-Experts (MoE) layer (Dai et al., 2024), which allows a single model to distinguish the dynamics of multiple chaotic systems by enabling different experts to specialize in their unique characteristics. The MoE layer consists of M specialist experts and one shared expert, which are all implemented with standard feed-forward layers. A gating network activates a sparse combination of these experts for each input. Its output is a weighted sum of the

shared expert and the top K specialist experts:

$$\mathrm{MoE}(\bar{\boldsymbol{h}}_p) = \phi_{M+1,p}\mathrm{FFN}_{M+1}(\bar{\boldsymbol{h}}_p) + \sum_{i=1}^{M}(\phi_{i,p}\mathrm{FFN}_i(\bar{\boldsymbol{h}}_p)), \tag{2}$$

$$\phi_{i,p} = \begin{cases} s_{i,p}, & s_{i,p} \in \mathrm{TopK}(\{s_{j,p}\}_{j=1}^{M}, K), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

$$\phi_{M+1,p} = \mathrm{Sigmoid}(\boldsymbol{W}_{M+1}\bar{\boldsymbol{h}}_p), \qquad s_{:,p} = \mathrm{Softmax}(\boldsymbol{W}\bar{\boldsymbol{h}}_p), \tag{4}$$

where $s_{i,p}$ is the score of the i-th specialist expert. $\boldsymbol{W}$s are trainable parameters.

Encoding and Patch Merging. The encoder blocks progressively builds a hierarchy of representations at increasingly coarse resolutions. Following each Transformer block at level i, a *patch merging* layer reduces the temporal resolution by a factor of two while doubling the feature dimension. This down-sampling is achieved by concatenating the features of adjacent temporal patches and applying a learnable linear projection. Given the output of the i-th encoder block, $\boldsymbol{H}_{\text{enc}}^{(i)} \in \mathbb{R}^{\frac{S}{2^{i-1}} \times V \times 2^{i-1}d_e}$, the patch merging is formulated as:

$$\boldsymbol{H}_{\text{enc}}^{'(i)} = \mathrm{Concat}(\boldsymbol{H}_{\text{enc}}^{(i)}[0::2,\dots], \boldsymbol{H}_{\text{enc}}^{(i)}[1::2,\dots])\boldsymbol{W}_{\text{enc}}^{(i)} + \boldsymbol{b}_{\text{enc}}^{(i)}, \tag{5}$$

where the output $\boldsymbol{H}_{\text{enc}}^{'(i)} \in \mathbb{R}^{\frac{S}{2^i} \times V \times 2^i d_e}$ serves as the input to the next encoder level. This allows successive layers to capture features ranging from fine-grained details to coarse, global structures. The hierarchical encoding process culminates in a bottleneck layer positioned at the deepest level of the architecture, which consists of a linear layer that processes the feature representation at the coarsest temporal scale, bridging the transition from the encoding path to the decoding path.

Decoding and Patch Expansion. The decoder blocks reconstructs the high-resolution representation from the low-dimensional features produced by the encoder and a final bottleneck layer. Each decoder block is followed by a patch expansion layer that reverses the merging process. It up-samples the features by doubling the temporal resolution and halving the channel dimension via a linear transformation and a reshape operation. For the i-th decoder level, the input $\boldsymbol{H}_{\text{dec}}^{(i)} \in \mathbb{R}^{\frac{S}{2^i} \times V \times 2^i d_e}$ is expanded, producing an output $\boldsymbol{H}_{\text{dec}}^{'(i)} \in \mathbb{R}^{\frac{S}{2^{i-1}} \times V \times 2^{i-1}d_e}$ as follows:

$$\boldsymbol{H}_{\text{dec}}^{'(i)} = \mathrm{Reshape}(\boldsymbol{W}_{\text{dec}}^{(i)}\boldsymbol{H}_{\text{dec}}^{(i)} + \boldsymbol{b}_{\text{dec}}^{(i)}), \tag{6}$$

Skip Connections. To mitigate the loss of fine-grained information during down-sampling, we introduce skip connections linking encoder and decoder blocks at corresponding resolutions. The output $\boldsymbol{H}_{\text{enc}}^{(i)}$ from the i-th encoder layer is passed through a dedicated skip connection block implemented with 1D convolutions and then fused with the up-sampled features $\boldsymbol{H}_{\text{dec}}^{'(i)}$ from the corresponding decoder layer. This fusion provides the decoder with direct access to high-resolution encoder features, which is crucial for accurate reconstruction of the system's dynamics. Further details are provided in Appendix C.2.

3.3 Frequency-enhanced Joint Scale Readout

The decoder of ScaleFormer produces a set of representations $\{\boldsymbol{H}_{\text{dec}}^{(i)}\}_{i=1}^{L}$ capturing system dynamics at L different temporal scales. To synthesize these into a single, comprehensive representation for forecasting, we first apply temporal mean pooling to each decoder output to obtain system-level features $\bar{\boldsymbol{H}}^{(i)}$ for each scale. These features are then concatenated and projected through a linear fusion layer to produce a unified dynamics representation $\boldsymbol{H}_{\text{uni}} \in \mathbb{R}^{d_e \times V}$ contains integrated multi-scale information:

$$\boldsymbol{H}_{\text{uni}} = \mathrm{Concat}(\bar{\boldsymbol{H}}^{(1)}, \bar{\boldsymbol{H}}^{(2)}, \cdots, \bar{\boldsymbol{H}}^{(L)})\boldsymbol{W}_f + \boldsymbol{b}_f.$$

A robust foundation model must not only model temporal evolution but also identify the underlying dynamical system or its current regime. To this end, we condition our model on frequency-domain information, which serves as a fingerprint for the system's dynamics. We employ the wavelet scattering transform on the historical observations $\boldsymbol{X}$ to extract a stable, multi-scale summary of its spectral content (Appendix C.3). The resulting scattering coefficients, $\boldsymbol{F}_w \in \mathbb{R}^{C \times T' \times V}$, are temporally pooled to yield a single frequency fingerprint, $\bar{\boldsymbol{F}}_w \in \mathbb{R}^{C \times V}$. It distills the system's intrinsic

oscillatory and modulatory behaviors into a fixed-size representation, enhancing the model's ability to distinguish between different dynamical systems. The final multi-step forecast is produced by a linear prediction head that combines the unified dynamics $\boldsymbol{H}_{\text{uni}}$ and the frequency fingerprint $\bar{\boldsymbol{F}}_w$:

$$\hat{\boldsymbol{X}}_{T+1:T+H} = \text{Concat}(\boldsymbol{H}_{\text{uni}}, \bar{\boldsymbol{F}}_w)\boldsymbol{W}_o + \boldsymbol{b}_o, \tag{7}$$

where $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ are learnable parameters. This allows the model to leverage both the learned multi-scale temporal patterns and the intrinsic spectral properties of the system for accurate prediction.

3.4 Training Objective

The total objective function for ChaosNexus is composed of three distinct components: a primary forecasting loss, an auxiliary load balancing loss for the MoE layers, and a distributional regularization term to preserve the system's statistical properties. The primary training objective is the Mean Squared Error (MSE), which measures the point-wise accuracy, formulated as:

$$\mathcal{L}_{\text{mse}} = \frac{1}{B}\sum_{n=1}^{B} ||\hat{\boldsymbol{X}}^n_{T+1:T+H} - \boldsymbol{X}^n_{T+1:T+H}||_2^2, \tag{8}$$

where $\hat{\boldsymbol{X}}^n$ and $\boldsymbol{X}^n$ are the predicted and ground-truth of the n-th trajectory in a batch with size B.

As is standard for Mixture-of-Experts (MoE) models, relying solely on the prediction loss can lead to expert load imbalance, where the gating network disproportionately favors a small subset of experts (Shazeer et al., 2017). This leaves other experts under-trained and limits the model's overall capacity. To mitigate this, we incorporate an auxiliary load balancing loss from Dai et al. (2024):

$$\mathcal{L}_{\text{balance}} = M\sum_{i=1}^{M} f_i r_i, \tag{9}$$

where f_i is the fraction of patches routed to expert i, and r_i is the average routing probability assigned to it. This encourages more uniform expert utilization.

Due to the sensitive dependence on initial conditions in chaotic systems, point-wise accuracy is often insufficient for long-horizon forecasting. A robust forecast must also reproduce the geometric and statistical properties of the system's attractor. To enforce this, we introduce a regularization term based on the Maximum Mean Discrepancy (MMD), which minimizes the divergence between the state distribution of predicted trajectories and that of the ground-truth trajectories (Appendix C.4):

$$\mathcal{L}_{\text{reg}} = \frac{1}{B^2}\sum_{i,j}\kappa(\hat{\boldsymbol{X}}^i, \hat{\boldsymbol{X}}^j) + \frac{1}{B^2}\sum_{i,j}\kappa(\boldsymbol{X}^i, \boldsymbol{X}^j) - \frac{2}{B^2}\sum_{i,j}\kappa(\hat{\boldsymbol{X}}^i, \boldsymbol{X}^j), \tag{10}$$

where $\{\hat{\boldsymbol{X}}^n\}_{n=1}^B$ and $\{\boldsymbol{X}^n\}_{n=1}^B$ represent batches of the full predicted and ground-truth trajectories. Following prior work, we use a mixture of rational quadratic kernels for the kernel function κ (Schiff et al., 2024; Seeger, 2004; Reiss et al., 2019). The final objective function is a weighted sum of these three components: $\mathcal{L} = \mathcal{L}_{\text{mse}} + \lambda_1\mathcal{L}_{\text{balance}} + \lambda_2\mathcal{L}_{\text{reg}}$, where λ_1, λ_2 are hyperparameters that control the relative weights of the auxiliary loss terms.

4 Experiments

In this section, we present comprehensive experiments to evaluate the forecasting capabilities of our proposed model. Due to space constraints, we present the main findings here and provide further in-depth analyses, including supplementary benchmark results, extensive ablation studies, model sensitivity and internal mechanics, as well as visualizations of forecasting cases in Appendix A.

4.1 Zero-shot Forecasting

Setups. We utilize the benchmark dataset consisting of synthetic chaotic systems from Panda (Lai et al., 2025). Its training set contains 20K novel chaotic ODEs, generated synthetically by an evolutionary algorithm that evolved from 129 known systems (Gilpin, 2021; 2023). The data was further

REVISE

diversified with dynamics-preserving augmentations like time-delay embedding (Takens, 2006). The held-out test set, used for evaluation, comprises 9.3K systems derived from a disjoint seed population (Appendix D.1). We use symmetric mean absolute percentage error (sMAPE) (Lai et al., 2025) of 128 and 512 timesteps to evaluate the point-wise forecasting accuracy. We also consider the correlation dimension error (D_{frac}), the Kullback–Leibler (KL) divergence between system attractors (D_{stsp}), the largest Lyapunov exponent error (D_{Lyap}), and the weighted mean energy error (ME_{LRw}) to evaluate the fidelity in key statistical properties of system attractors (Zhang & Gilpin, 2024). These complementary metrics jointly assess both point-wise accuracy and long-term preservation of attractor geometry, which are essential to whether the model has captured the underlying chaotic dynamics. **REVISE** We compare our proposed method against several state-of-the-art time series foundation models with different parameter sizes, including Panda (Lai et al., 2025), Time-MoE (Shi et al., 2024), TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), Moirai-MoE (Liu et al., 2024a), Timer-XL (Liu et al., 2024b), DynaMix (Hemmer & Durstewitz, 2025), Parrot (Zhang & Gilpin, 2025), where '-S', '-B, '-L' refer to small, base, large in parameter size, respectively. To assess the adaptability of general-purpose models to this specific domain, we also include Chronos-S-SFT, a variant of the Chronos-S model that has been fine-tuned on our chaotic systems training corpus. For all other baseline models, we load their officially released pre-trained weights for evaluation. **REVISE** We choose these baselines because they are all foundation models intended for generalization, aligning with our zero-shot evaluation on previously unseen chaotic systems. Details of experimental setups are demonstrated in Appendix D.

Results. We conduct a zero-shot evaluation on the held-out test set of chaotic systems. For a fair comparison, all models use a context length of 512 to autoregressively forecast 512 steps into the future. While ChaosNexus and the Panda baseline are pretrained on the chaotic systems corpus, other baselines are general-purpose time-series foundation models, for which we employ the official pretrained weights. **REVISE** As shown in Figure 2, ChaosNexus demonstrates point-wise accuracy competitive with the baseline, achieving an average sMAPE of 68.901 at 128 steps. Regarding the long-term dynamics, ChaosNexus exhibits superior fidelity. It reduces the average correlation dimension error (D_{frac}) to 0.203. Notably, it attains an average KL divergence of attractors (D_{stsp}) of 1.206. Table 2 in Appendix A.4 further demonstrates the superior performance of ChaosNexus on D_{Lyap} and ME_{LRw}. Given that the sensitive dependence on initial conditions renders any long-term point-wise forecast of a chaotic system ultimately unreliable (Li et al., 2021; Jiang et al., 2023; Schiff et al., 2024), the strong performance of ChaosNexus in long-term statistical metrics is therefore **REVISE** compelling evidence that it can infer intrinsic dynamics of new systems from the contexts rather than superficial pattern memorizing. Notably, leading general-purpose time-series foundation models, despite being pretrained on larger time-series datasets than ours (Appendix D.3), struggle on chaotic system forecasting. We also observe that their generalization capabilities can be improved (from Chronos-SFT-S) after further fine-tuned on chaotic systems corpus. This contrast provides compelling evidence for our claim that chaotic dynamics possess unique differences from general time series. It also validates the necessity of building domain-specific foundation models on chaotic

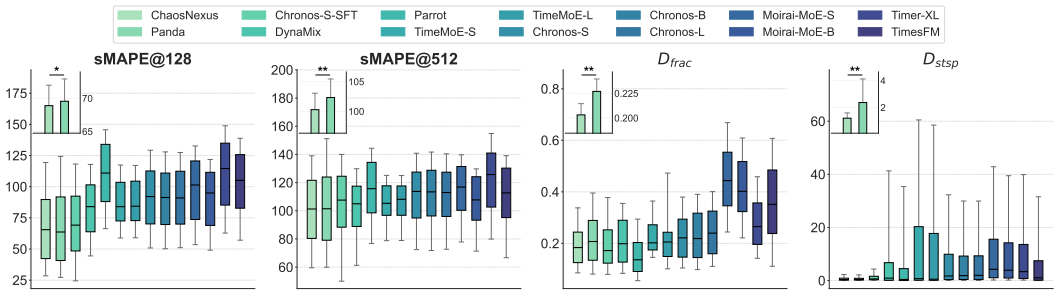

Figure 2: Zero-shot forecasting performances of models on synthetic chaotic systems. Each box shows the median (center line), the middle 50% of results (box), and the overall range (whiskers). The inset plot shows the mean performance with the 95% CI of ChaosNexus and Panda. Asterisks indicate statistically significant differences determined by the Wilcoxon signed-rank test (*: $p < 0.05$, **: $p < 0.01$). **REVISE**

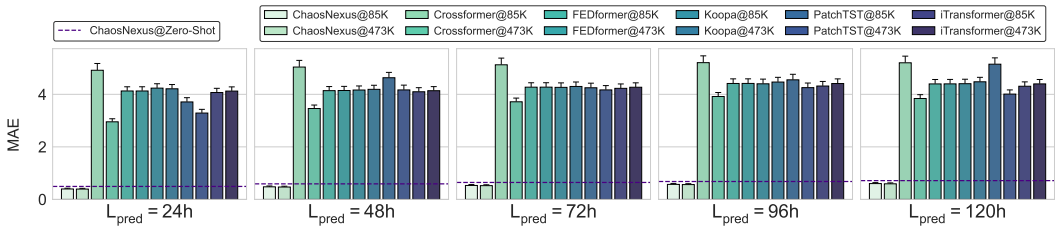

Figure 3: Few-shot forecasting performance for global temperature on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. The zero-shot performance of ChaosNexus is shown as a dashed line for reference.

data and underscores the importance of the specialized architectural designs for multi-scale feature extraction and system disentanglement in ChaosNexus.

4.2 Few-shot Forecasting

Setups. Weather is an inherently chaotic system (Lorenz, 1969; 1982; 2017). For a rigorous evaluation on a real-world chaotic system, we utilize the WEATHER-5K dataset (Han et al., 2024). This dataset comprises hourly meteorological data from 5,672 global weather stations over a 10-year period from 2014 to 2023. It is then chronologically split, with data from 2014 to 2021 used for training, 2022 for validation, and 2023 for testing. Each sample includes five variables: temperature, dew point, wind speed, wind direction, and sea-level pressure. Given the profound real-world importance of forecasting absolute values, we primarily employ the Mean Absolute Error (MAE) to directly measure the discrepancy between predicted and ground-truth observations. MAE is the gold-standard metric in this application, as researchers value the absolute accuracy of these weather-related variables. The forecasting task is to predict the subsequent 120 hours of all variables given 512 hours of historical context. To assess few-shot performance under data-scarce conditions, we fine-tune models on two small subsets of the training data: 0.1% (85K samples) and 0.5% (473K samples). In all few-shot experiments, ChaosNexus is first pretrained on the synthetic chaotic systems corpus and then fine-tuned on exactly the same WEATHER-5K subsets as the baselines, which are trained from scratch without pretraining.Besides foundation models included in Section 4.1, we select several strong deep learning baselines in this benchmark, including FEDformer, CrossFormer, PatchTST, and Koopa. They are widely adopted architectures for time-series forecasting, making them appropriate references for this single-system, real-world benchmark. We also report the performance of our model in a zero-shot setting, without any fine-tuning on the weather dataset. Further details of setups are provided in Appendix F.

REVISE

REVISE

Results. Figure 3 presents the forecasting results for the temperature variable. Remarkably, ChaosNexus in a zero-shot setting—without any fine-tuning—surpasses all baselines in their few-shot configurations. It achieves a mean error strictly below 1°C for 5-day (120-hour) global temperature forecasts. In stark contrast, the baseline models exhibit an MAE of at least 3°C, even when fine-tuned on the same data. The performance of ChaosNexus further improves with few-shot fine-tuning, especially for longer prediction horizons. This suggests that while pre-training endows the model with a robust, universal understanding of chaotic behavior, fine-tuning allows it to adapt these principles to the specific physical constraints and periodicities (e.g., diurnal and seasonal cycles) inherent in meteorological systems. This process grounds the model's abstract dynamical representations in real-world physics, enhancing its ability to generate accurate and stable long-term forecasts. Detailed results of all weather variables and performances of foundation models are shown in the Appendix A.6. We find that foundation models designed for chaotic system forecasting and trained on our corpus of synthetic chaotic dynamics, including ChaosNexus, Panda, and Chronos-S-SFT, perform significantly better than those trained on general time series, even though they use a much larger corpus (see Table 9). It demonstrates that pretraining specifically on chaotic systems provides a more relevant inductive bias for weather forecasting. Moreover, ChaosNexus also outperforms Panda on many variable forecasting tasks, highlighting the contribution of our multi-scale architectural designs.

ADD

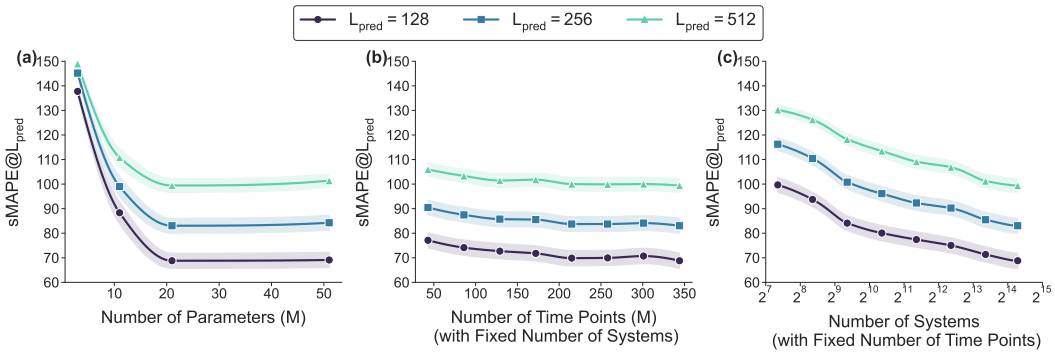

Figure 4: Scaling behavior of ChaosNexus. We demonstrate zero-shot sMAPE on synthetic chaotic systems varying: (a) the number of parameters; (b) the number of time points while holding the system diversity constant; and (c) the number of systems while holding the trajectories per system constant. Lines depict the average value, with shaded regions representing the 95% CI.

4.3 Scaling Behavior

An investigation into scaling behavior is crucial for the development of foundation models, since understanding how model performance scales with key factors such as parameter count and data volume is essential for guiding future research and resource allocation.

Parameter Scaling. We first explored the impact of model size on performance. We generated a suite of models with varying parameter counts, ranging from $2.83M$ to $52.63M$, by systematically adjusting the number of encoder and decoder layers, as well as the dimension d_e of the embedding space. The results demonstrated in Figure 4(a) reveal a consistent trend: increasing the model's parameter count yields steady improvements in performance. For instance, scaling the model from $2.83M$ to $52.63M$ parameters improved the sMAPE@128 by 49.83%, which demonstrates that larger models possess a greater capacity to capture the complex dynamics inherent in the data.

Data Scaling. We further investigated the model's performance as a function of the training data size under two distinct settings. First, we fix the diversity, *i.e.,* the total number, of training systems, while varying the number of trajectories sampled from each system, leading to only different training time points. Second, we increase the diversity of systems while holding the number of training time points constant. From Figure 4(b), we find that merely increasing the number of time points for a fixed set of systems did not lead to a significant enhancement in zero-shot performance. In contrast, Figure 4(c) demonstrates that increasing the number of distinct systems in the training set substantially improved the model's ability to generalize. These findings also support established research (Norton et al., 2025; Lai et al., 2025)on data scaling. While prior work, such as (Lai et al., 2025), establishes the scaling law for system diversity, which our Figure 4(c) corroborates, our complementary analysis in Figure 4(b) provides a refinement. The negligible gain from scaling per-system data volume suggests that effective generalization is driven by corpus-level diversity, i.e., the number of systems rather than by per-system trajectories. **REVISE**

4.4 Multi-scale Feature Analysis

To investigate the inner workings of our multi-scale architecture, we visualize the input signal's patch partitioning alongside the temporal attention maps from shallow and deep layers of both the encoder and decoder. As illustrated in Figure 5 and 8, we select three systems from the test set with progressively weaker regularity (left to right in Figure 5), thus increasing the forecasting difficulty.

Patch Partition Patterns. We find that the shallow layers, which operate on smaller patches, are adept at capturing local, high-frequency fluctuations. In contrast, the deeper layers, processing merged patches that represent longer time intervals, focus on capturing long-term trends and global structures. This is particularly evident in 5(b), where a shallow-layer patch may encompass only a peak or a trough, whereas a deep-layer patch spans an entire peak-valley cycle.

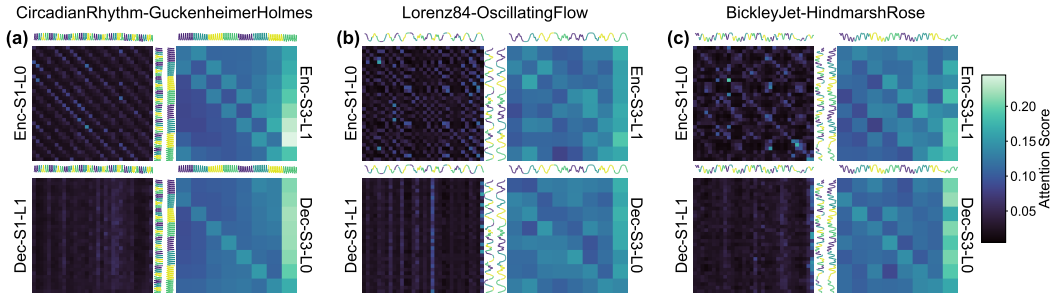

Figure 5: Visualization of input patch partitioning and multi-scale temporal attention for three chaotic systems. Each panel displays attention maps for the shallow (left) and deep (right) layers of the encoder (top) and decoder (bottom).

Temporal Attention Patterns of Encoder Layers. The encoder's attention patterns distinctly reflect this multi-scale processing. The deep encoder layers (upper right of each subfigure) consistently exhibit globalized attention distributions, indicating a focus on synthesizing long-range dependencies. The shallow encoder layers (upper left), however, display system-specific patterns. For the highly regular system in 5(a), the map forms a Toeplitz-like structure (Bajwa et al., 2007), analogous to a convolutional operation, suggesting the model applies fixed-pattern filters to scan the time series. For the more complex system in 5(c), the attention forms distinct blocks, indicating that the model concentrates on specific temporal segments whose interplay is deemed critical for understanding the system's state. The system in 5(b) presents a hybrid pattern, blending the features of 5(a) and 5(c) to capture its intermediate complexity.

Temporal Attention Patterns of Decoder Layers. The decoder's attention mechanisms operate differently, functioning primarily as a selector. This aligns with our architectural design, where the decoder's outputs are mean-pooled over the temporal dimension for the final forecast. The model must therefore learn to select and combine specific patterns from the historical context to support its predictions. The deep decoder layers show a pronounced focus on the final patch, capturing the most recent temporal dependencies crucial for autoregressive prediction. The shallow decoder layers, conversely, appear to anticipate future dynamics; for instance, in 5(b), after observing a descending phase, the model intensifies its attention on historical ascending patterns, selectively weighting the context that is most relevant for the anticipated future trajectory.

5 Conclusions

We introduce ChaosNexus, a foundation model that features a universal, pre-trained approach to chaotic system forecasting, effectively overcoming data sparsity. Its novel multi-scale ScaleFormer architecture, augmented with Mixture-of-Experts layers and a wavelet-based frequency fingerprint, achieves state-of-the-art zero-shot performance by accurately predicting both point-wise evolution and long-term attractor properties. Crucially, our scaling analysis reveals that generalization is driven by the diversity of systems in the pre-training corpus, not the sheer volume of trajectories per system. This key insight provides a clear roadmap for developing powerful, data-efficient models for complex scientific applications.

Ethics Statement

The authors have read and adhered to the ICLR Code of Ethics. The research presented in this paper is foundational and focuses on the modeling of chaotic systems, with primary applications in scientific domains such as meteorology. All data used for training and evaluation is either synthetically generated from mathematical principles or derived from publicly available, non-personal scientific datasets, ensuring no privacy concerns. This work does not involve human subjects, and we do not foresee any direct negative societal impacts or risks of perpetuating social biases. Our aim is to advance the scientific understanding and predictive capabilities for complex physical systems for the benefit of the scientific community.

Reproducibility Statement

We are committed to ensuring the reproducibility of our research. The complete source code for the ChaosNexus model, along with scripts for data processing, training, and evaluation, is publicly available in an anonymous repository at `https://anonymous.4open.science/r/ChaosNexus-C809`. We acknowledge the authors of previous open-source projects (Lai et al., 2025) whose codebases served as a foundation for our implementation. **REVISE** A detailed description of our proposed ScaleFormer architecture, including the patch merging/expansion mechanisms and the Mixture-of-Experts layers, is provided in Section 3. A comprehensive breakdown of implementation details for key components, such as input feature augmentation, skip connections, the wavelet scattering transform, and the MMD regularization term, can be found in Appendix C. Detailed descriptions of the datasets are provided in the appendices: the generation process and augmentations for the synthetic chaotic systems are in Appendix D.1, and the specifics of the WEATHER-5K benchmark are in Appendix F.1. All hyperparameters used for our model variants are explicitly listed in Table 8 in Appendix B. The full experimental protocol, including training procedures and the precise definitions of our evaluation metrics, is detailed in Appendix D.2 and E. All baseline models used in our comparisons are described in Appendix D.3 and F.2.

References

Joakim Andén and Stéphane Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

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*, 2024.

Waheed U Bajwa, Jarvis D Haupt, Gil M Raz, Stephen J Wright, and Robert D Nowak. Toeplitz-structured compressed sensing matrices. In *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, pp. 294–298. IEEE, 2007.

Manuel Brenner, Florian Hess, Jonas M Mikhaeil, Leonard F Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable dendritic rnns for reconstructing nonlinear dynamical systems. In *International conference on machine learning*, pp. 2292–2320. Pmlr, 2022.

Manuel Brenner, Elias Weber, Georgia Koppe, and Daniel Durstewitz. Learning interpretable hierarchical dynamical systems models from time series data. *arXiv preprint arXiv:2410.04814*, 2024.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

Xiaoyuan Cheng, Yi He, Yiming Yang, Xiao Xue, Sibo Cheng, Daniel Giles, Xiaohang Tang, and Yukun Hu. Learning chaos in a linear way. *arXiv preprint arXiv:2503.14702*, 2025.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 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*, 2024.

Daniel J Gauthier, Erik Bollt, Aaron Griffith, and Wendson AS Barbosa. Next generation reservoir computing. *Nature communications*, 12(1):5564, 2021.

William Gilpin. Chaos as an interpretable benchmark for forecasting and modelling. *arXiv preprint arXiv:2110.05266*, 2021.

William Gilpin. Model scale versus domain knowledge in statistical forecasting of chaotic systems. *Physical Review Research*, 5(4):043252, 2023.

Niclas Göring, Florian Hess, Manuel Brenner, Zahra Monfared, and Daniel Durstewitz. Out-of-domain generalization in dynamical systems reconstruction. *arXiv preprint arXiv:2402.18377*, 2024.

Peter Grassberger and Itamar Procaccia. Characterization of strange attractors. *Physical review letters*, 50(5):346, 1983.

Tao Han, Song Guo, Zhenghao Chen, Wanghan Xu, and Lei Bai. Weather-5k: A large-scale global station weather dataset towards comprehensive time-series forecasting benchmark. *arXiv e-prints*, pp. arXiv–2406, 2024.

Yi He, Yiming Yang, Xiaoyuan Cheng, Hai Wang, Xiao Xue, Boli Chen, and Yukun Hu. Chaos meets attention: Transformers for large-scale dynamical prediction. *arXiv preprint arXiv:2504.20858*, 2025.

Christoph Jürgen Hemmer and Daniel Durstewitz. True zero-shot inference of dynamical systems preserving long-term statistics. *arXiv preprint arXiv:2505.13192*, 2025.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pp. IV–317. IEEE, 2007.

Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. *arXiv preprint arXiv:2306.04406*, 2023.

Zijie Huang, Yizhou Sun, and Wei Wang. Generalizing graph ode for learning complex system dynamics across environments. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 798–809, 2023.

Junen Jia, Feifei Yang, and Jun Ma. A bimembrane neuron for computational neuroscience. *Chaos, Solitons & Fractals*, 173:113689, 2023.

Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. *Advances in Neural Information Processing Systems*, 36: 27645–27669, 2023.

Anran Jiao, Haiyang He, Rishikesh Ranade, Jay Pathak, and Lu Lu. One-shot learning for solution operators of partial differential equations. *Nature Communications*, 16(1):8386, 2025.

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

Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for universal representation of chaotic dynamics. *arXiv preprint arXiv:2505.13755*, 2025.

Xin Li, Qunxi Zhu, Chengli Zhao, Xiaojun Duan, Bolin Zhao, Xue Zhang, Huanfei Ma, Jie Sun, and Wei Lin. Higher-order granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction. *Nature communications*, 15(1):2506, 2024.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International conference on machine learning*, pp. 1718–1727. PMLR, 2015.

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning dissipative dynamics in chaotic systems. *arXiv preprint arXiv:2106.06898*, 2021.

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*, 2024a.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024b.

Edward N Lorenz. The predictability of a flow which possesses many scales of motion. *Tellus*, 21 (3):289–307, 1969.

Edward N Lorenz. Atmospheric predictability experiments with a large numerical model. *Tellus*, 34 (6):505–513, 1982.

Edward N Lorenz. Deterministic nonperiodic flow 1. In *Universality in Chaos, 2nd edition*, pp. 367–378. Routledge, 2017.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Alexandre Mauroy, Y Susuki, and Igor Mezic. *Koopman operator in systems and control*, volume 7. Springer, 2020.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems*, 37:119301–119335, 2024.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2):429–443, 1997.

Habib N Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual review of fluid mechanics*, 41(1):35–52, 2009.

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.

Declan A Norton, Yuanzhao Zhang, and Michelle Girvan. Learning beyond experience: Generalizing to unseen state space with reservoir computing. *arXiv preprint arXiv:2506.05292*, 2025.

Roussel Desmond Nzoyem, Grant Stevens, Amarpal Sahota, David AW Barton, and Tom Deakin. Towards foundational models for dynamical system reconstruction: Hierarchical meta-learning via mixture of experts. *arXiv preprint arXiv:2502.05335*, 2025.

Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14):3079, 2019.

David Rind. Complexity and climate. *science*, 284(5411):105–107, 1999.

Michael T Rosenstein, James J Collins, and Carlo J De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1-2): 117–134, 1993.

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

Yair Schiff, Zhong Yi Wan, Jeffrey B Parker, Stephan Hoyer, Volodymyr Kuleshov, Fei Sha, and Leonardo Zepeda-Núñez. Dyslim: Dynamics stable learning by invariant measure for chaotic systems. *arXiv preprint arXiv:2402.04467*, 2024.

Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

Patrick Seifner, Kostadin Cvejoski, and Ramses J Sanchez. Foundational inference models for dynamical systems. *CoRR*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

Jagadish Shukla. Predictability in the midst of chaos: A scientific basis for climate forecasting. *science*, 282(5389):728–731, 1998.

Keshav Srinivasan, Nolan Coble, Joy Hamlin, Thomas Antonsen, Edward Ott, and Michelle Girvan. Parallel machine learning for forecasting the dynamics of complex networks. *Physical Review Letters*, 128(16):164101, 2022.

Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Chapman and Hall/CRC, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in Neural Information Processing Systems*, 36:71242–71262, 2023.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381. Springer, 2006.

D Vignesh, Shaobo He, and Santo Banerjee. A review on the complexities of brain activity: insights from nonlinear dynamics in neuroscience. *Nonlinear Dynamics*, 113(5):4531–4552, 2025.

Yuchen Wang, Hongjue Zhao, Haohong Lin, Enze Xu, Lifang He, and Huajie Shao. A generalizable physics-enhanced state space model for long-term dynamics forecasting in complex environments. *arXiv preprint arXiv:2507.10792*, 2025.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.

James A Yorke and ED Yorke. Chaotic behavior and fluid dynamics. In *Hydrodynamic Instabilities and the Transition to Turbulence*, pp. 77–95. Springer, 2005.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Yuanzhao Zhang and William Gilpin. Zero-shot forecasting of chaotic systems. *arXiv preprint arXiv:2409.15771*, 2024.

Yuanzhao Zhang and William Gilpin. Context parroting: A simple but tough-to-beat baseline for foundation models in scientific machine learning. *arXiv preprint arXiv:2505.11349*, 2025.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Supplementary Experimental Results

A.1 Ablation Studies

To validate the effectiveness of our proposed architecture and training strategy, we conduct a series of ablation studies. Specifically, we evaluate four variants of our model by removing designs of (i) patch merging and expansion operations, (ii) MoE layers, (iii) MMD-based auxiliary regularization, and (iv) frequency fingerprint. The results are shown in Table 1, showing that the full model strikes an effective balance between short-term point-wise accuracy and the preservation of long-term statistical properties.

Patch Merging and Expansion. The removal of the patch merging and expansion modules resulted in a severe degradation of performance. We observed a substantial decline in both short-term predictive accuracy and long-term statistical fidelity, with sMAPE@128 and D_{frac} increasing by 7.8% and 21.70%, respectively. This underscores the critical importance of capturing the multi-scale features inherent in chaotic systems.

MoE Layers. Replacing MoE layers with normal feed-forward layers also leads to the performance drop in both short-term and long-term predictive accuracy. MoE layers enables the model to allocate specialized experts to capture distinct dynamical regimes present across different systems. Otherwise, a single, monolithic network is forced to approximate all behaviors, reducing its capacity and leading to worse performance. The results highlights the vital role of MoE layers in discriminating between diverse dynamics.

MMD-based Auxiliary Regularization. The exclusion of MMD-based auxiliary regularization during training has a particularly pronounced negative impact on long-term forecasting and the preservation of statistical properties, with sMAPE@512 and D_{frac} decreasing by 2.8% and 10.17%, respectively. The auxiliary regularization aligns the state distribution of the learned attractor with that of the ground truth system, which is an invariant measure (Cheng et al., 2025). Its removal decouples the model from this fundamental physical constraint, impairing its ability to generate realistic long-term trajectories.

Frequency Fingerprint. Removing the wavelet transform-based frequency fingerprint results in a noticeable decrease in model performance. The fingerprint provides the model with frequency-domain information of the underlying system, which complements the temporal data by offering a holistic signature of its structural properties. The synergy between these two sources of information allows the model to form a more complete and accurate representation of the dynamics, leading to more robust forecasting.

A.2 Expert Activation Visualization

We visualize the expert activation patterns within the encoder and decoder for selected test systems in Figure 6. We find that systems derived from the same foundation dynamics (Appendix D.1) trigger analogous routing profiles across all layers and scales. This provides direct evidence that the MoE framework has learned to partition the problem space, systematically assigning inputs to specialized experts based on their dynamical properties to effectively process and differentiate between complex systems. We also provide quantitative results in Appendix A.9 to further support our findings. **REVISE**

Table 1: Model performances when removing each of our designs. Reported values represent the mean ± 95% CI. (PME: Patch Merging and Expansion; MoE: Mix-of-Experts Layers; MMD: MMD-based Auxiliary Regularization; FF: Frequency Fingerprint.)

Metrics \ Model	Full	w/o PME	w/o MoE	w/o MMD	w/o FF
sMAPE@128	68.901 ± 3.086	74.161 ± 3.082	69.076 ± 3.069	80.702 ± 3.217	67.699 ± 3.179
sMAPE@512	100.293 ± 2.767	106.542 ± 2.516	100.298 ± 2.694	110.228 ± 2.771	97.002 ± 2.930
D_{frac}	0.203 ± 0.011	0.240 ± 0.010	0.220 ± 0.012	0.220 ± 0.010	0.209 ± 0.010
D_{stsp}	1.206 ± 0.392	1.820 ± 0.620	1.250 ± 0.310	1.460 ± 0.490	1.360 ± 0.440
ME_{LRw}	1.562 ± 0.115	2.218 ± 0.152	1.770 ± 0.122	2.571 ± 0.164	1.771 ± 0.132
D_{Lyap}	0.065 ± 0.025	0.075 ± 0.019	0.065 ± 0.011	0.103 ± 0.032	0.072 ± 0.013

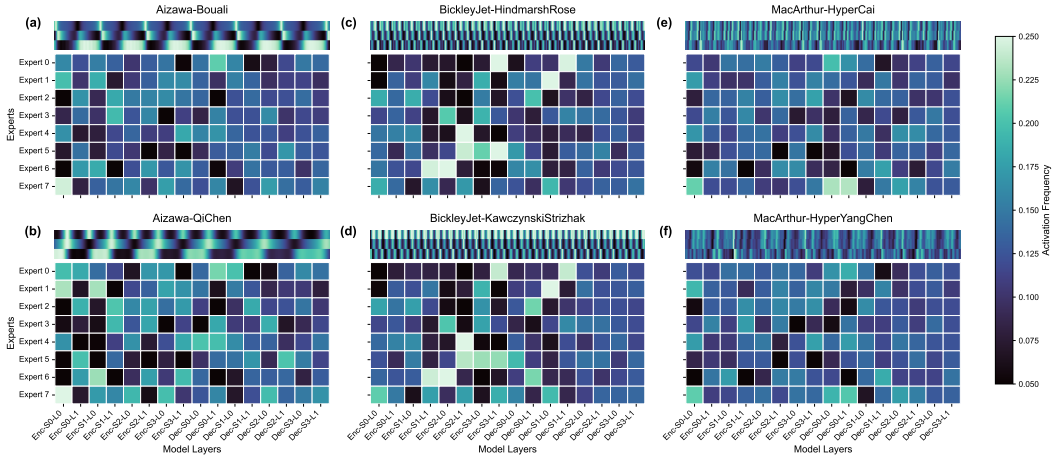

Figure 6: Expert activation visualization for six discovered chaotic systems by the evolutionary framework from three common foundation chaotic systems.

A.3 Performance Sensitivity to Context and Prediction Length

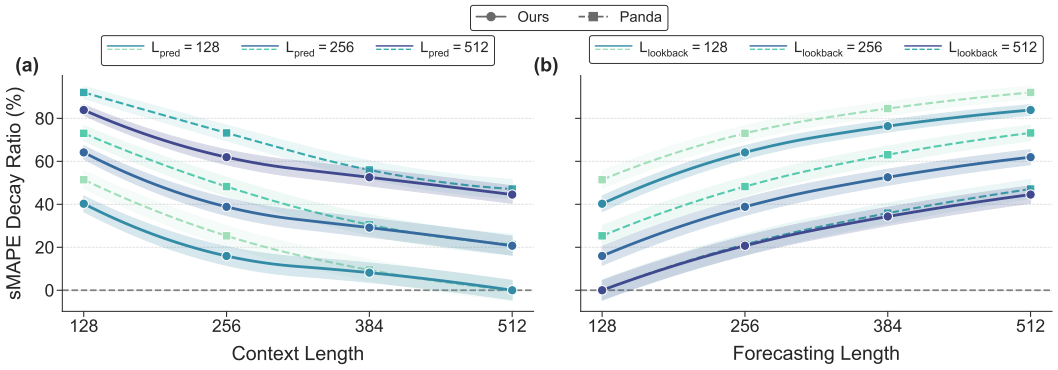

Figure 7: Performance Sensitivity of ChaosNexus and Panda to different (a) context length and (b) forecasting length. Lines depict the average value, with shaded regions representing the 95% CI.

Performance with Different Context Length. We evaluate our model across a range of input context lengths. As shown in Figure 7(a), our model's performance consistently improves with a longer context and consistently surpasses the baseline Panda model. It also shows less sensitivity to the specific context length chosen. These advantages of our model stems from its multi-scale architecture, which effectively leverages information across different temporal scales to build a more stable representation of the system's dynamics.

Performance with Different Prediction Length. Long-horizon forecasting serves as a crucial test of a model's capacity to learn the intrinsic dynamics of a chaotic system. Accordingly, our model's performance advantage over Panda becomes substantially larger at longer prediction horizons, as shown in Figure 7(b). It validates our design philosophy, which prioritizes multi-scale feature extraction and dynamics discrimination to build a more faithful representation of the underlying system.

A.4 Numerical Results on Synthetic Chaotic Systems

We demonstrate detailed numerical results corresponding to Figure 2 in Table 2 for reference.

ADD

Table 2: Detailed numerical results of model performance on synthetic chaotic systems. The best performance of each metric is marked in **bold**, and the second-best performance is underlined. Reported values represent the mean ± 95% CI.

Metric \ Model	ChaosNexus	Panda	Chronos-S-SFT	Chronos-B-SFT	Chronos-L-SFT	Chronos-S	Chronos-B	Chronos-L
sMAPE@128 (↓)	68.901 ± 3.0857	69.567 ± 3.358	70.510 ± 11.356	70.124 ± 12.761	69.765 ± 11.514	86.323 ± 33.031	86.883 ± 33.122	82.730 ± 32.165
sMAPE@512 (↓)	100.293 ± 2.7669	102.333 ± 3.123	101.947 ± 10.226	101.215 ± 13.497	100.824 ± 11.058	104.826 ± 32.191	104.156 ± 31.964	102.967 ± 31.827
D_{frac} (↓)	0.203 ± 0.011	0.227 ± 0.013	0.233 ± 0.165	0.224 ± 0.085	0.210 ± 0.053	0.233 ± 0.135	0.246 ± 0.143	0.219 ± 0.120
D_{stsp} (↓)	1.206 ± 0.392	2.369 ± 1.751	2.391 ± 10.651	2.837 ± 1.978	2.685 ± 1.652	11.498 ± 25.207	11.255 ± 24.561	11.731 ± 27.171
ME_{LRw} (↓)	1.562 ± 2.015	1.649 ± 0.413	1.580 ± 0.350	1.602 ± 0.260	1.571 ± 0.302	2.397 ± 2.698	2.3729 ± 2.8044	2.385 ± 2.871
D_{Lyap} (↓)	0.065 ± 0.025	0.067 ± 0.047	0.072 ± 0.023	0.068 ± 0.021	0.069 ± 0.024	0.082 ± 0.007	0.074 ± 0.008	0.072 ± 0.007

Metric \ Model	Moirai-MoE-S	Moirai-MoE-L	TimeMoE-L	TimeMoE-S	TimerXL	TimesFM	Parrot	DynaMix
sMAPE@128 (↓)	92.223 ± 35.279	95.103 ± 53.000	87.426 ± 13.411	87.186 ± 13.790	105.379 ± 36.289	100.933 ± 15.372	92.084 ± 16.764	70.381 ± 12.148
sMAPE@512 (↓)	108.493 ± 30.777	109.446 ± 31.755	103.489 ± 12.238	103.143 ± 12.757	115.239 ± 34.773	108.211 ± 13.381	114.368 ± 14.724	102.966 ± 14.945
D_{frac} (↓)	0.423 ± 0.204	0.372 ± 0.209	0.230 ± 0.164	0.256 ± 0.310	$\infty \pm$ nan	0.364 ± 0.076	0.106 ± 0.157	0.145 ± 0.182
D_{stsp} (↓)	13.613 ± 27.323	13.581 ± 27.593	10.651 ± 25.348	11.542 ± 28.004	14.534 ± 30.619	9.655 ± 11.048	6.085 ± 17.528	6.904 ± 19.824
ME_{LRw} (↓)	3.181 ± 2.168	6.803 ± 4.842	8.700 ± 1.029	8.965 ± 1.013	3.925 ± 2.648	11.122 ± 0.606	0.654 ± 1.067	1.638 ± 2.372
D_{Lyap} (↓)	0.081 ± 0.012	0.075 ± 0.042	0.072 ± 0.014	0.068 ± 0.002	0.075 ± 0.009	0.069 ± 0.008	0.065 ± 0.012	0.067 ± 0.014

Table 3: Inference time comparison of foundation models when forecasting 512 time steps. Reported values represent the mean ± standard deviation, which are computed based on 1000 runs. **ADD**

Model	Time (s)
ChaosNexus	0.119 ± 0.036
Panda	0.048 ± 0.004
Chronos-S	0.081 ± 0.022
Chronos-B	0.095 ± 0.012
Chronos-L	0.173 ± 0.022
Moirai-MoE-S	1.677 ± 0.377
Moirai-MoE-L	3.124 ± 0.201
TimeMoE-S	0.038 ± 0.019
TimeMoE-L	0.042 ± 0.020
TimesFM	0.143 ± 0.026
Timer-XL	0.005 ± 0.002

A.5 Inference Efficiency

Table 3 demonstrates the computational efficiency of various foundation models in a long-term forecasting scenario. Specifically, we report the inference latency required to generate a prediction horizon of 512 time steps with a context length of 512 time steps. To ensure the statistical reliability of our results, the reported values are the mean and standard deviation derived from 1,000 independent runs. As observed, ChaosNexus exhibits an inference latency approximately 0.017s higher than Panda per forecast. This moderate increase is an expected trade-off adhering to the "no free lunch" principle, attributable to our hierarchical architecture of ScaleFormer, MoE routing, and frequency-domain modeling. Given that the faithful reproduction of complex chaotic dynamics is paramount and the observed latency remains well within practical limits for this task, we consider the computational cost well-justified by the substantial performance gains. Regarding general-purpose baselines, their inference speeds are largely dictated by specific architectural configurations, such as patch granularity and architectural complexity. For instance, Timer-XL achieves high efficiency through large-patch processing (e.g., patch size of 96), whereas Moirai-MoE incurs significant overhead due to its smaller patch size, intricate expert routing and gating clustering mechanisms. However, we emphasize that lower latency cannot compensate for poor generalization. Since these baselines fail to capture chaotic dynamics effectively, their speed advantage offers no practical utility.

A.6 Additional Results on Weather Benchmark

A.6.1 Detailed Results

We demonstrate the detailed forecasting results for all weather variables, including the temperature, dew point, sea level pressure, wind direction, and wind speed in Figure 19-23, respectively. More clear results for ChaosNexus, Panda, Chronos-S-SFT, which are previously trained on the cor- **REVISE**

pus of synthetic chaotic systems, are shown in Figure 24-28. This strong performance paradigm is consistently replicated across the remaining meteorological variables. In the zero-shot setting, ChaosNexus substantially outperforms all baseline models, even when they are fine-tuned on up to 473K samples from the target weather system. The model's forecasting accuracy is further enhanced with few-shot fine-tuning, demonstrating remarkable data efficiency. This advantage is particularly pronounced at longer prediction horizons, highlighting the robustness of the representations learned during pre-training. Collectively, these results validate our central hypothesis: pre-training on a diverse corpus of chaotic systems endows the model with a universal understanding of complex dynamics. This allows ChaosNexus to achieve state-of-the-art performance on real-world forecasting tasks with minimal, or even zero, in-domain fine-tuning, thereby overcoming the critical challenge of data sparsity in scientific applications. Besides comparison with system-specific models in Figure 3 of the main text, we also benchmark the forecasting performance of other foundation models on this dataset. We find that foundation models designed for chaotic system forecasting or trained on our corpus of synthetic chaotic dynamics, including ChaosNexus, Panda, and Chronos-S-SFT, perform significantly better than those trained on general time series, even though they use a much larger corpus (see Table 9). It demonstrates that pretraining specifically on chaotic systems provides a more relevant inductive bias for weather forecasting. Moreover, ChaosNexus also outperforms Panda on many variable forecasting tasks, highlighting the contribution of our multi-scale architectural designs.

ADD

ADD

A.6.2 Temperature Forecasting Performance across Latitudes

We conduct additional analysis and stratify weather stations into three latitude bands: low latitudes (30°N–30°S), mid-latitudes (30°N–60°N, 30°S–60°S), and high latitudes (60°N–90°N, 60°S–90°S). There are 1093, 4000, and 579 stations in low-latitude, mid-latitude, and high-latitude bands, respectively. For each band, we report the MAE on the 5-day temperature forecasting of our model and all baselines. The results are demonstrated in Figure 29-31.

From the results, we can draw the following conclusions:

- **First**, ChaosNexus maintains a zero-shot MAE strictly below 1°C across all latitude bands at the 5-day (120h) horizon. Furthermore, fine-tuning yields consistent performance gains across all stations, for instance, in high-latitude regions, the 120h MAE decreases from 0.8124 to 0.6659 (an $\sim 18\%$ improvement). This confirms that our foundation model serves as a robust universal prior capable of rapid adaptation to local climatic conditions.
- **Second**, the error distribution accurately reflects the inherent complexity of atmospheric dynamics. Zero-shot error is minimized in the tropics (MAE ≈ 0.59) due to lower variability, and increases slightly in mid-to-high latitudes (MAE ≈ 0.74–0.81), regions characterized by chaotic frontal systems and baroclinic instability. Despite these challenges, the error remains tightly bounded.
- **Third**, ChaosNexus consistently outperforms all baselines across every latitude band. It surpasses strong system-specific baselines (e.g., Crossformer, PatchTST) by a substantial margin, avoiding catastrophic errors exceeding 3°C, and reliably outperforms the competing foundation model, Panda, in zero-shot settings. These results establish ChaosNexus as the state-of-the-art solution for chaotic forecasting.

A.7 Additional Results on Multi-scale Feature Analysis

We demonstrate temporal attention map of each encoder and decoder levels of ScaleFormer in Figure 8.

A.8 Forecast Showcases

We demonstrate forecasting showcases of six representative systems in Figure 9.

ADD

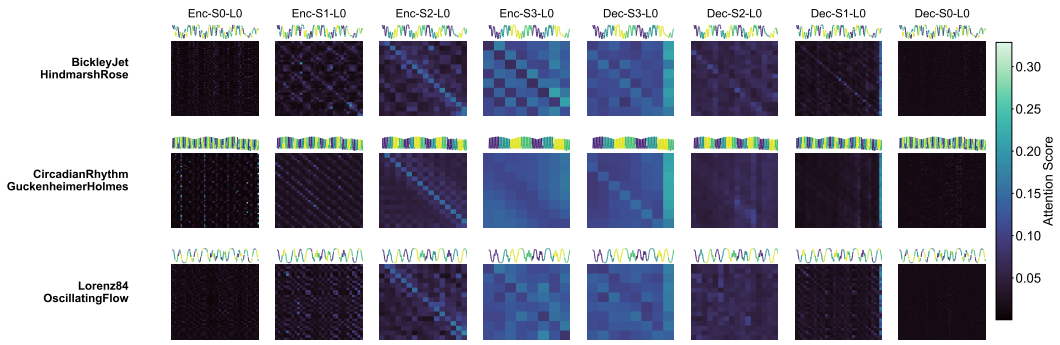

Figure 8: Visualization of input patch partitioning and multi-scale temporal attention for three chaotic systems.

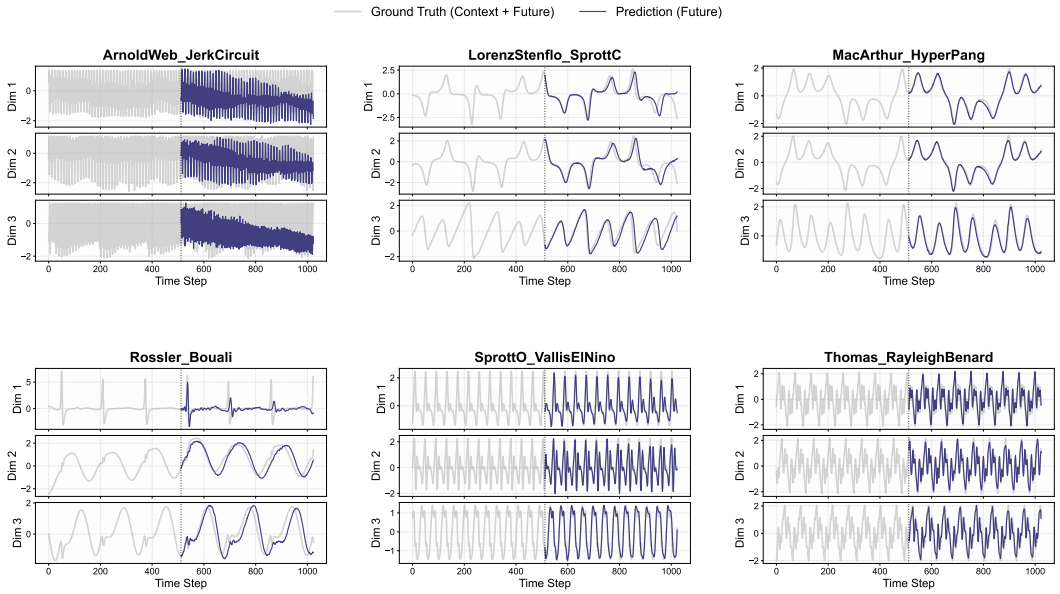

Figure 9: Forecasting showcases of representative chaotic systems.

A.9 Quantitative Analysis on Expert Activation Patterns

A.9.1 Expert Activation Clustering

To investigate the underlying specialization mechanisms within the Mixture of Experts (MoE) architecture, we analyze the gating activation patterns, *i.e.,* expert selection probabilities, across different depths of the network. Specifically, we aggregate the expert activation probabilities of context trajectories from three canonical chaotic dynamical systems, including Lorenz63, Rossler, and Lorenz96 systems, to determine whether the router implicitly learns to distinguish systems based on their governing physical laws.

We employ t-SNE to project the high-dimensional gating distributions from various Encoder and Decoder MoE layers (Depths 1 through 4) into a low-dimensional manifold, demonstrated in Figure 10. To quantify the degree of system-specific specialization in the routing mechanism, we calculate the Adjusted Rand Index (ARI) for each projection, which measures the similarity between the obtained clustering and the ground-truth labels. A score of 1.0 signifies perfect alignment where experts are exclusively specialized for specific systems, whereas a score near 0.0 indicates random assignment.

The visualization reveals that the router's gating decisions are highly structured and system-dependent. In the vast majority of MoE layers, the expert activation patterns form distinct clusters

that correspond precisely to the Lorenz63, Rossler, and Lorenz96 systems. This observation is substantiated by the quantitative metrics, where the ARI scores consistently remain high—exceeding 0.5 in most layers and peaking at 0.9933 in the encoder. These results statistically confirm that the experts exhibit strong system-level specialization, implying that the router implicitly learns to distinguish and dispatch data based on the distinct underlying physical mechanisms of each dynamical system.

Figure 10: Layer-wise expert activation patterns clustered by system type. **ADD**

A.9.2 Entropy of Gating Distribution

Figure 11 depicts the layer-wise evolution of the gating entropy of three canonical systems, including Lorenz63, Rossler, and Lorenz96. Scatter points represent the entropy of the gating distribution from a specific sample, and box plots encapsulate the aggregate statistical dispersion, i.e., the median and interquartile range. The results are summarized as follows:

- **Shallow Encoder.** In the initial encoder layers (Enc-D1 to Enc-D3), the gating distribution exhibits consistently high entropy. This indicates that the router utilizes a diverse mixture of experts to process raw input patches.
- **Bottleneck.** A significant reduction in entropy is observed as the information propagates to the network bottleneck (Enc-D4 and Dec-D4). Here, the entropy minimizes, signifying a regime of

high specialization. The model abstracts the input into core dynamical representations, and the router demonstrates high confidence, assigning specific expert modules to handle distinct underlying patterns. This drop in entropy confirms that the model has successfully disentangled the latent semantics, prioritizing specific experts for specific dynamical behaviors.

- **Shallow Decoder.** In the final decoding stages, entropy rises back to higher levels, which implies collaborative synthesis. To reconstruct accurate continuous trajectories from abstract representations, the decoder must integrate the semantic guidance from both the bottleneck and the high-frequency details retrieved via skip connections. The router therefore employs an ensembling strategy, aggregating outputs from multiple experts to ensure robust, smooth, and precise signal reconstruction.

- **Discussion on Load Balancing Loss.** The results demonstrate that the router establishes a dynamic equilibrium: it yields to the regularization pressure in the shallow layers to maintain generalizability, but prioritizes semantic specialization in the deep layers where distinguishing physical mechanisms is critical. Thus, the load balancing loss serves as a flexible regularizer, preventing mode collapse without suppressing the necessary concentration of attention required to model complex chaotic dynamics.

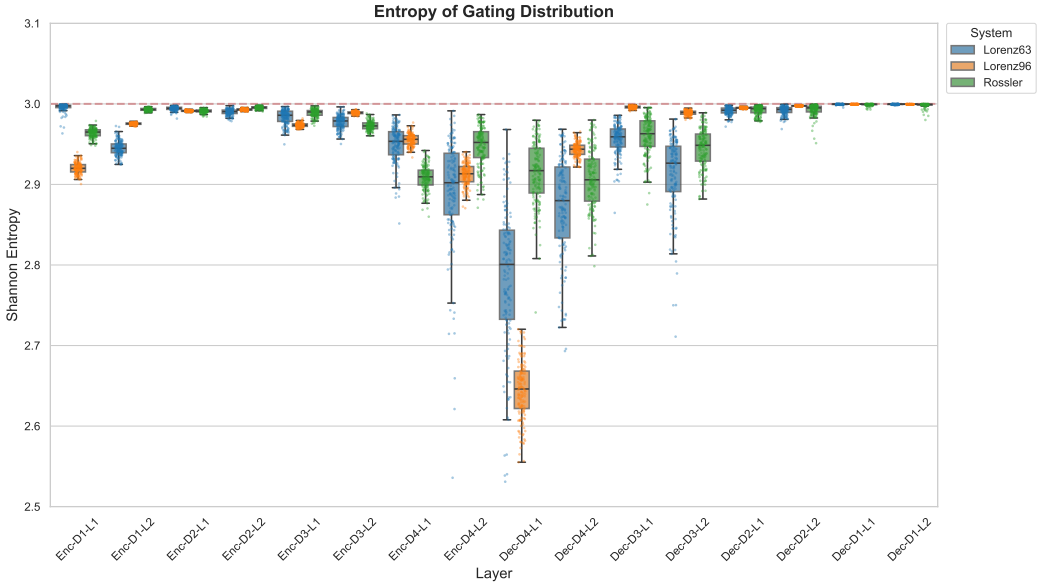

Figure 11: Layer-wise entropy of gating distribution in three canonical systems. **ADD**

A.9.3 Expert Pruning Impact

To validate the distinct functional specialization within our Mixture-of-Experts architecture, we conduct an expert pruning experiment on three canonical chaotic systems, including Lorenz63, Rossler, and Lorenz96. Specifically, we identify the top-2 most frequently activated experts for each system per layer and deactivate them during the inference phase. As evidenced by the results in Table 4, this targeted pruning leads to a consistent degradation across both point-wise forecasting accuracy (sMAPE) and long-term attractor fidelity metrics (D_{frac} and D_{stsp}). This performance drop substantiates that the model relies on specific, specialized experts to capture distinct dynamical regimes, rather than utilizing a generalized ensemble for all inputs.

ADD

Table 4: Expert pruning impact on three canonical chaotic systems. Each reported value indicates the mean ± 95% CI. **ADD**

Experiment	sMAPE@128	sMAPE@512	D_{frac}	D_{stsp}
Lorenz63 w/o Pruning	**62.1053 ± 0.9641**	115.5445 ± 0.6513	0.1316 ± 0.0033	0.2041 ± 0.0187
Lorenz63 w/ Pruning	79.6978 ± 1.0023	123.3420 ± 0.5920	0.1467 ± 0.0032	0.2474 ± 0.0188
Lorenz96 w/o Pruning	**154.1404 ± 0.0912**	**157.5176 ± 0.0697**	**6.0222 ± 0.0139**	**20.5535 ± 0.0488**
Lorenz96 w/ Pruning	154.1597 ± 0.0919	157.5768 ± 0.0697	6.1593 ± 0.0135	20.6266 ± 0.0491
Rossler w/o Pruning	**30.4578 ± 0.5250**	**55.6769 ± 0.5904**	**0.1587 ± 0.0048**	**0.0744 ± 0.0032**
Rossler w/Pruning	37.8179 ± 0.5786	64.8312 ± 0.6044	0.1598 ± 0.0046	0.1022 ± 0.0040

Table 5: Sensitivity analysis to the weighting coefficient λ_2 of MMD regularization. **ADD**

λ_2	sMAPE@128	sMAPE@512	D_{frac}	D_{stsp}
0.01	80.093 ± 3.213	109.596 ± 2.809	0.231 ± 0.012	1.331 ± 0.381
0.05	80.139 ± 3.169	107.743 ± 2.744	0.216 ± 0.012	1.434 ± 0.435
0.10	79.107 ± 3.112	105.665 ± 2.731	0.210 ± 0.012	1.287 ± 0.400
0.50	**68.901 ± 3.086**	**100.293 ± 2.767**	**0.203 ± 0.011**	**1.206 ± 0.392**
1.00	78.474 ± 2.923	102.550 ± 2.412	0.208 ± 0.011	1.329 ± 0.395
5.00	80.928 ± 2.760	103.572 ± 2.320	0.210 ± 0.012	1.385 ± 0.309
10.00	81.280 ± 2.724	103.668 ± 2.319	0.209 ± 0.012	1.318 ± 0.333

A.10 Impact of MMD Regularization

A.10.1 Sensitivity to the Weighting Coefficient

We set $\lambda_2 = 0.5$ in our experiments. Here we demonstrate the sensitivity to the weighting coefficient λ_2. Specifically, we choose λ_2 at different scales: $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. The results are demonstrated in Table 5. From the results, we draw the following conclusions:

- **First**, our observations indicate that $\lambda_2 = 0.5$ represents a robust optimum, effectively balancing the point-wise accuracy required for short-term forecasting with the distributional fidelity needed for long-term stability.
- **Second**, when λ_2 is small (0.01-0.1), we observe a marked degradation in both point-wise accuracy and attractor fidelity. This confirms that explicitly enforcing attractor geometry aids the model in learning the underlying dynamics. Pure MSE minimization is insufficient for chaotic systems as it lacks the global constraints to prevent divergence.
- **Third**, excessively large weights ($\lambda_2 \geq 5.0$) lead to a performance drop on point-wise accuracy, as the distributional constraint begins to dominate the loss landscape, impeding the model's ability to minimize local prediction errors.

A.10.2 Sensitivity to Kernel Function

We conduct additional experiments to compare our default mixture of rational quadratic kernels against three alternatives: a Gaussian kernel, a linear kernel, and a polynomial kernel, which are implemented as follows:

- **Gaussian kernel**. To ensure a fair comparison with the multi-scale nature of our default mixture of rational quadratic kernel, we implemented the Gaussian kernel as a mixture over the same set of length scales $\boldsymbol{\sigma} = \{0.2, 0.5, 0.9, 1.3\}$,

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = \sum_{\sigma \in \boldsymbol{\sigma}} \exp - \frac{||\boldsymbol{u} - \boldsymbol{v}||_2^2}{2\sigma^2}. \tag{11}$$

- **Linear kernel**. The linear kernel captures similarity through a direct dot product in the input space, implying a linear relationship between the governing features of the attractors:

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v}. \tag{12}$$

Table 6: Sensitivity analysis of the kernel function selection of MMD regularization. **ADD**

Kernel	sMAPE@128	sMAPE@512	D_{frac}	D_{stsp}
Mixture of rational quadratic kernel	**68.901 ± 3.086**	**100.293 ± 2.767**	**0.203 ± 0.011**	**1.206 ± 0.392**
Gaussian kernel	80.329 ± 3.198	109.577 ± 2.780	0.227 ± 0.012	1.431 ± 0.515
Linear kernel	82.293 ± 3.145	109.282 ± 2.750	0.217 ± 0.012	1.276 ± 0.313
Polynomial kernel	83.126 ± 3.033	107.908 ± 2.533	0.215 ± 0.011	1.309 ± 0.366

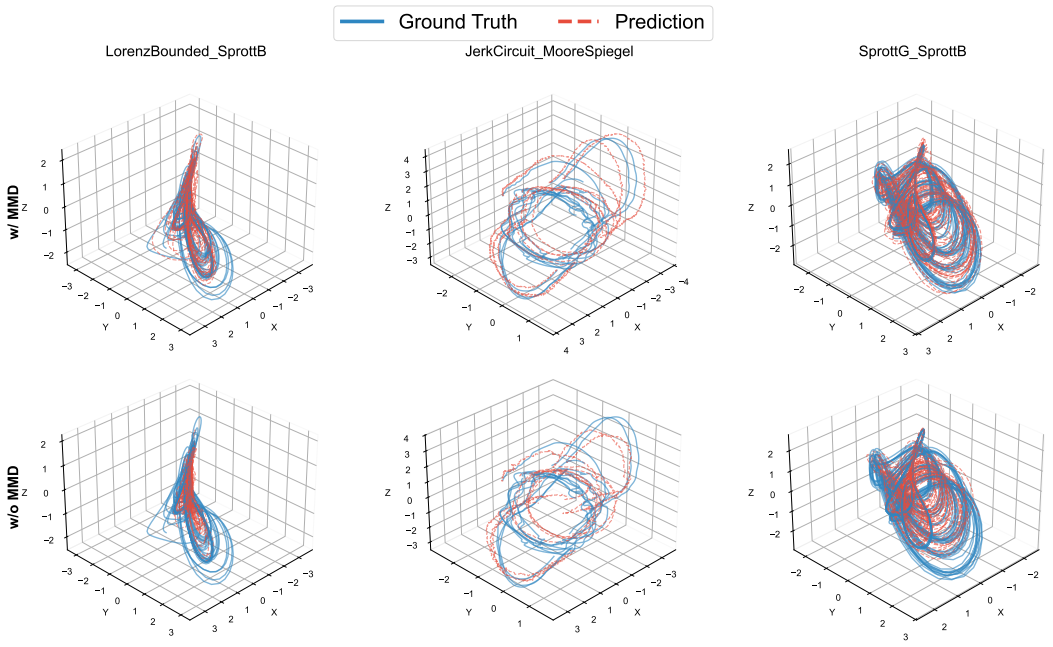

Figure 12: Visualization of the impact of MMD regularization on long-term forecasting. **ADD**

- **Polynomial kernel**. The polynomial kernel projects the inputs into a higher-dimensional feature space determined by the degree d and a bias term c:

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^T \boldsymbol{v} + c)^d, \tag{13}$$

  where we set $d = 2$ and $c = 1$.

The experimental results are shown in Table 6. We find that the mixture of rational quadratic kernels consistently yields superior performance across both short-term forecasting (sMAPE) and long-term attractor reconstruction. It outperforms the Gaussian, linear, and polynomial kernels by a wide margin in both point-wise accuracy and attractor reconstruction fidelity. This aligns with the theoretical motivation in Appendix C.4, a rational quadratic kernel can be viewed as an infinite mixture of Gaussian kernels with varying length scales (Seeger, 2004). This property is crucial for capturing the multi-scale temporal and spectral structures inherent in chaotic systems, which single-scale Gaussian kernels fail to represent adequately.

A.10.3 Visualization Examples

We further provide illustrative forecasting cases that isolate the contribution of the MMD-based auxiliary loss. The results are demonstrated in Figure 12. As observed, the removal of the distributional constraint causes the predicted trajectories to drift significantly from the underlying manifold, failing to reproduce the complex geometry of the strange attractor. In contrast, the MMD-regularized model effectively preserves the attractor structure, ensuring that the forecasted dynamics faithfully align with the ground-truth.

ADD

Table 7: Comparison between alternative spectral representations.

Experiment	sMAPE@128	sMAPE@512	D_{frac}	D_{stsp}
WST (Ours)	**68.9010 ± 3.0857**	**100.293 ± 2.7669**	0.203 ± 0.011	**1.2060 ± 0.3920**
STFT	77.0957 ± 11.5019	102.2048 ± 11.2470	**0.2010 ± 0.0560**	1.3697 ± 1.2395
Learnable	83.5496 ± 11.1222	107.3003 ± 9.9495	0.2152 ± 0.0573	2.0323 ± 1.2871

A.11 Detailed Analysis on Frequency Fingerprint

We explore using the STFT and learnable fourier features as alternative designs for the system fingerprint. Specifically, to implementation STFT, we replace the WST module with an STFT encoding, flattening the time-frequency features into the same dimension as our fingerprint. For learnable fingerprint, we replace the fixed wavelet filters with learnable spectral filters (1D convolutional layer) followed by the same pooling operations, allowing the model to adaptively learn frequency representations. The results are shown in Table 7. From the results, we have the following conclusions:

- **First**, the WST achieves significantly lower point-wise errors and better attractor reconstruction compared to STFT. We attribute this to the fact that chaotic systems exhibit dynamics across a continuum of scales. WST naturally captures multi-scale interactions through its hierarchical cascade, making it more robust for diverse chaotic dynamics. In contrast, STFT suffers from the fixed window size limitation.
- **Second**, Learnable variant performs the worst. Given the vast diversity of our training corpus, learning a single set of spectral filters that generalizes universally is highly difficult. The WST provides a strong inductive bias with its mathematical properties of translation invariance and stability to deformations, offering a stable fingerprint that requires no training, thus enhancing zero-shot generalization.

ADD

A.12 Forecasting Performance on PDE systems

Simulation Setup. We consider the 2D Navier-Stokes equations modeled via the Lattice Boltzmann Method (LBM) using a standard D2Q9 topology. The simulation is configured to generate Von Kármán Vortex Street (VKVS) dynamics past a cylindrical obstacle. The simulation domain is a rectangular channel with dimensions 420×180 lattice units. A cylindrical obstacle with radius $r = 20$ is positioned at $(x, y) = (105, 90)$ to induce flow separation. We impose a parabolic velocity profile at the inlet with a maximum characteristic velocity $u_{LB} = 0.04$, and a standard bounce-back condition on the obstacle surface. The viscosity is adjusted to achieve a Reynolds number (Re) of 450, placing the system in a regime characterized by unsteady, periodic vortex shedding and chaotic turbulence in the wake.

Data Collection. To ensure the flow reaches a statistically stationary state, we discard the initial $90,000$ simulation steps as a burn-in period. Subsequently, we collect a dataset of $T = 4096$ frames, sampled at a temporal interval of $\Delta t = 250$ LBM steps.

Preprocessing. Instead of raw velocity fields, we focus on the vorticity dynamics ($\omega = \partial_x v_y - \partial_y v_x$), computed via central differences, as it better highlights the coherent structures of the fluid. The spatial domain is cropped to remove the laminar inlet region (removing the first 40 columns), resulting in an effective resolution of 380×180. To enable efficient forecasting, we project the high-dimensional vorticity fields into a low-dimensional latent space using Principal Component Analysis (PCA), retaining the top $d = 16$ principal components.

Results. We compare the zero-shot forecasting performance of ChaosNexus with other foundation models on ODE-based chaotic dynamics, including Panda, Parrot (Zhang & Gilpin, 2025), and DynaMix (Hemmer & Durstewitz, 2025). While the forecasting processes operate within a low-dimensional PCA latent space, we apply the inverse transformation to map predictions back to the original observation space for metric evaluation. The context length is 512 steps, and we compute sMAPE at forecasting horizons $\{64, 128, 192, 256, 320, 384, 448, 512\}$, and the results are shown in Figure 13. We also demonstrate illustrative forecasting samples in Figure 14. We find that Chaos-

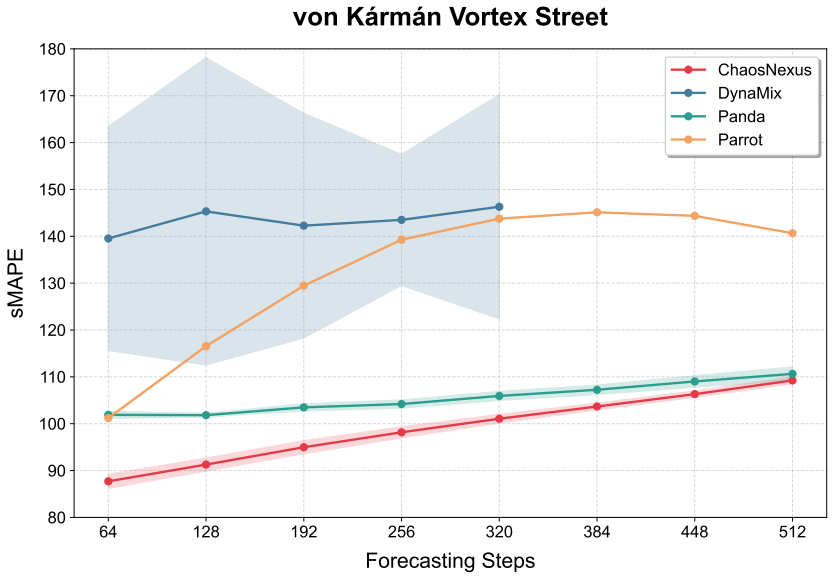

Figure 13: Forecasting performance on Von Kármán Vortex Street (VKVS) dynamics. Lines depict the average value, with shaded regions representing the 95% CI. DynaMix produces NaN values from 320 forecasting steps; therefore, its performance after longer horizons cannot be reported. **ADD**

Table 8: Hyperparameter configurations for ChaosNexus models.

Method	T	H	D	d_e	Blocks	Attention Heads	Skip Depths	M	K	C	J	Q	λ_1	λ_2	Params
ChaosNexus-Mini	512	128	8	24	[1,1,1,1]	[3,6,12,24]	[2,2,2,0]	8	2	48	8	8	0.1	0.5	**2.88M/7.60M**
ChaosNexus-Small	512	128	8	48	[1,1,1,1]	[3,6,12,24]	[2,2,2,0]	8	2	48	8	8	0.1	0.5	**10.88M/29.72M**
ChaosNexus-Base	512	128	8	48	[2,2,2,2]	[3,6,12,24]	[2,2,2,0]	8	2	48	8	8	0.1	0.5	**20.32M/58.01M**
ChaosNexus-Large	512	128	8	64	[3,3,3,3]	[4,8,16,32]	[2,2,2,0]	8	2	48	8	8	0.1	0.5	**52.68M/153.12M**

Nexus achieves superior forecasting performance on this PDE system, despite being trained solely on ODEs. PCA projects spatiotemporal dynamics onto a latent manifold that resembles our ODE training corpus. Crucially, our ScaleFormer architecture excels at modeling the resulting multi-scale temporal dynamics, effectively capturing both the dominant periodic vortex shedding and the fine-grained chaotic fluctuations in the turbulent wake.

B Hyperparameter Settings

Table 8 delineates the hyperparameter configurations for the suite of ChaosNexus models, spanning from Mini to Large scales. Please note that "ChaosNexus" refers to the "ChaosNexus-Base" variant in all analyses, figures, and tables (except for parameter scaling in Section 4.3), if not explicitly stated. For all model variants, we maintain a consistent input context length of $T = 512$ and a prediction horizon of $H = 128$, with the input trajectory segmented into patches of length $D = 8$. The scaling of model capacity is primarily achieved by adjusting the embedding dimension d_e, the number of Transformer blocks at each hierarchical level (Blocks), the corresponding number of attention heads (Heads), and the depth of the convolutional blocks within the skip connections (Skip Depths). Key parameters for our specialized components are kept constant across all scales: each Mixture-of-Experts (MoE) layer consists of $M = 8$ specialist experts, of which the top $K = 2$ are activated for each token, and the wavelet scattering transform produces a frequency fingerprint of dimension $C = 48$. This transform is configured with parameters $J = 8$ and $Q = 8$; as detailed in Appendix C.3, J defines the scale of temporal averaging for the low-pass filter, while Q represents the number of wavelet filters per octave (quality factor). The composite training objective is governed by the weights $\lambda_1 = 0.1$ for the MoE load balancing loss and $\lambda_2 = 0.5$ for the MMD-based distributional regularization. The final column reports both the number of activated and total parameters for each model configuration.

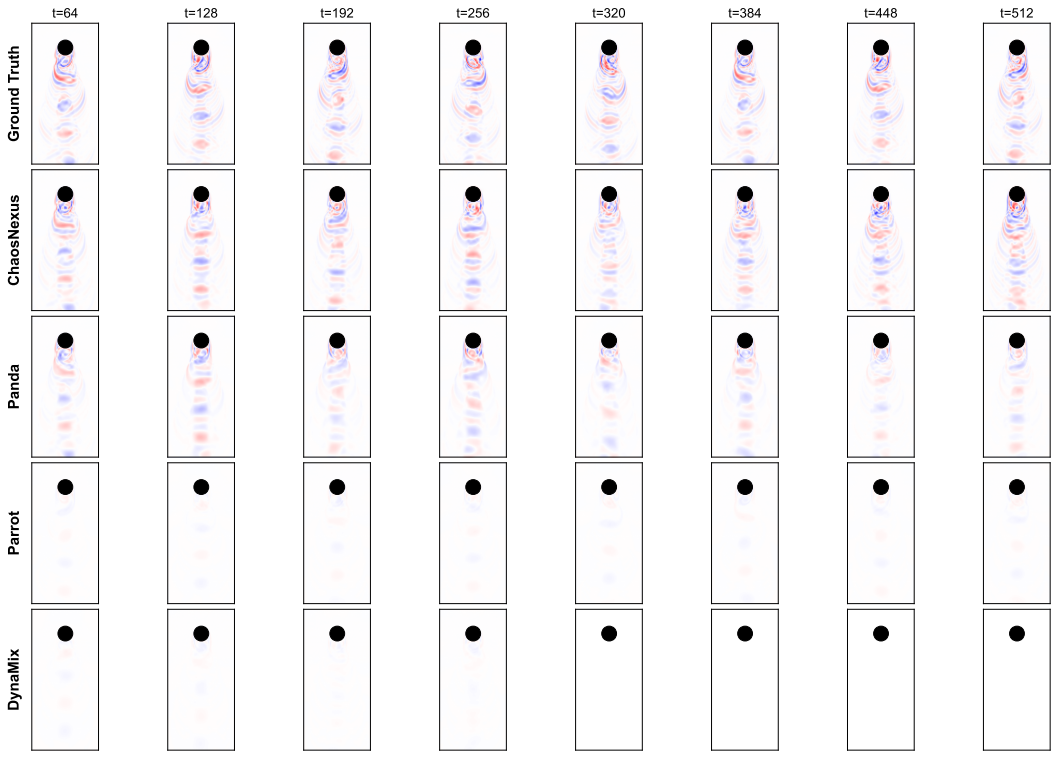

Figure 14: Forecasting visualizations on Von Kármán Vortex Street (VKVS) dynamics. **ADD**

C Implementation Details

C.1 Input Augmentation Features

As stated in the main text, our approach to feature engineering is inspired by Koopman operator theory (Koopman, 1931), which suggests that a complex nonlinear dynamical system can be represented as a linear system in an infinite-dimensional space of observable functions. While this infinite-dimensional space is practically inaccessible, it can be effectively approximated by projecting the system's state into a higher-dimensional feature space. This process of lifting the dynamics is a cornerstone of methods like Extended Dynamic Mode Decomposition (eDMD) (Williams et al., 2015).

Following this principle, and adopting a technique from recent work on pretrained forecast models, we enrich the representation of each time series patch before it is processed by the main architecture. Instead of using the raw patch data alone, we construct an augmented feature vector by concatenating the original patch with two additional sets of randomly generated, nonlinear features.

- **Random Polynomial Features.** To capture nonlinear relationships within each patch, we generate a set of monomial features. For a given polynomial degree, d, this is achieved by first sampling a collection of d-tuples of indices. For each tuple, we compute a new feature by multiplying the patch elements corresponding to those indices. This creates a basis of polynomial observables that can approximate the underlying dynamics. For our model, we use polynomial features of degree $d \in \{2, 3\}$.

- **Random Fourier Features.** To approximate a universal kernel and capture periodic patterns, we employ random Fourier features, a widely-used technique for scaling up kernel methods. This is implemented by projecting a patch onto a set of random vectors, whose components are sampled from a normal distribution. The resulting scalar values are then transformed using both sine and cosine functions, effectively creating a randomized spectral basis.

The final embedding for each patch is formed by concatenating the original patch vector with the generated polynomial and Fourier features. This lifted representation provides a much richer input to the model, allowing it to more easily learn and represent the complex, nonlinear evolution of the dynamical systems.

C.2 Skip Connection Blocks

To mitigate the loss of fine-grained information during the down-sampling operations within the encoder, we employ a skip connection architecture that links encoder and decoder blocks at corresponding resolutions. This mechanism is crucial for providing the decoder with direct access to high-resolution feature maps from the encoder, thereby enhancing the model's ability to reconstruct the system's dynamics with high fidelity.

Our implementation for these skip connections is a specialized 1D residual convolutional block. Its design is inspired by modern convolutional networks that have successfully integrated principles from Transformer architectures, showing high efficiency and performance (Herde et al., 2024). The block operates on different variables independently. The forward pass consists of the following key operations:

- **Depthwise Convolution.** The core of the block is a 1D depthwise convolution with a large kernel size, which is implemented as 7 in our experiments. This operation efficiently captures local spatio-temporal patterns across the patch sequence.
- **Normalization.** Following the convolution, a LayerNorm layer is applied to the features. This standardizes the activations across the feature dimension, ensuring stable training dynamics.
- **Inverted Bottleneck.** The architecture employs an inverted bottleneck design, a hallmark of modern efficient networks. The normalized features are first passed through a point-wise convolution that expands the channel dimension by a factor of 4. This is followed by a GELU activation function, which introduces non-linearity. A second point-wise convolution then projects the features back to the original dimension. This expand-and-contract structure allows the model to learn complex interactions between channels in a higher-dimensional space.
- **Stability and Regularization.** For improved training, two advanced techniques are integrated. First, a learnable, per-channel scaling parameter is applied to the output of the inverted bottleneck. This allows the model to dynamically modulate the contribution of each residual block, which is particularly beneficial in deep architectures. Second, the output of the block is randomly sets to zero during training, effectively bypassing it. This acts as a powerful regularizer, preventing feature co-adaptation and improving model generalization.
- **Residual Connection.** Finally, the output of the processed branch is added to the original input tensor, forming the block's essential residual connection.

By integrating these blocks as skip connections, we ensure that the decoder has access to a rich, multi-scale representation of the input, enabling it to accurately reconstruct detailed system dynamics that might otherwise be lost in the encoder's hierarchical processing.

C.3 Wavelet Scattering Transform

In our work, we employ the Wavelet Scattering Transform (WST) to extract a stable, multi-scale frequency representation from the historical observations $\boldsymbol{X}$. The WST (Mallat, 2012; Bruna & Mallat, 2013; Andén & Mallat, 2014) generates signal representations that are stable to small time shifts and deformations without sacrificing significant information. It achieves this by cascading wavelet convolutions with complex modulus non-linearities, followed by local averaging. This hierarchical structure is analogous to that of a Convolutional Neural Network (CNN), but with fixed, pre-defined wavelet filters instead of learned kernels. The transform is constructed by iteratively applying three fundamental operations: convolution with an analytic wavelet filter $\psi_\lambda(t)$, complex modulus non-linearity $|\cdot|$, and averaging via convolution with a low-pass filter $\phi_J(t)$.

For an input signal $x(t)$, the scattering transform up to the second order, denoted as $S_J x$, is a collection of coefficients from different layers (or orders):

$$S_J x = [S_J^{(0)} x, S_J^{(1)} x, S_J^{(2)} x], \tag{14}$$

where each order is defined as follows:

Zero-Order Coefficients. The zeroth-order coefficients capture the local mean of the signal. They are computed by convolving the input signal $x(t)$ with a wide low-pass filter $\phi_J(t)$, where J defines he scale of temporal averaging, formulated as follows:

$$S_J^{(0)}x(t) = x \star \phi_J(t).$$

This provides the coarsest, most stable representation of the signal's energy.

First-Order Coefficients. The first-order coefficients form the core of the wavelet analysis. The calculation begins by convolving the signal $x(t)$ with a family of first-order analytic wavelets, $\psi_\lambda^{(1)}(t)$, to capture information around specific frequencies λ. The complex modulus of this result is then taken—a crucial step that demodulates the signal and ensures invariance to local phase shifts. Finally, this resulting envelope is smoothed by convolving it with the low-pass filter $\phi_J(t)$, which achieves local time-shift invariance through averaging. The complete operation is summarized by the formula:

$$S_J^{(1)}x(t,\lambda) = |x \star \psi_\lambda^{(1)}| \star \phi_J(t).$$

Second-Order Coefficients. To recover transient information, such as rapid amplitude modulations lost during first-order averaging, the transform recursively applies the wavelet decomposition. This process begins with the modulus envelopes, $|x \star \psi_\lambda^{(1)}|$, generated by the first order. These envelopes are then convolved with a second family of wavelets, $\psi_\mu^{(2)}(t)$, to extract their spectral content, which reveals interactions between the primary frequency bands. Following this, a second modulus operation is applied before the final averaging with the low-pass filter $\phi_J(t)$ stabilizes the representation. The entire cascade is encapsulated by the formula:

$$S_J^{(2)}x(t,\lambda,\mu) = ||x \star \psi_\lambda^{(1)}| \star \psi_\mu^{(2)}| \star \phi_J(t).$$

In our methodology, the collection of all scattering coefficients, $\{S_J^{(0)}, S_J^{(1)}, S_J^{(2)}\}$, forms the feature set $\boldsymbol{F}_w \in \mathbb{R}^{C\times T'\times V}$. Here, C represents the total number of scattering paths (i.e., combinations of λ and μ), T' is the reduced temporal dimension after averaging, and V is the number of variables. To create a single, fixed-size fingerprint for the underlying dynamical system, we apply temporal pooling across the T' dimension. This results in the final representation $\bar{\boldsymbol{F}}_w \in \mathbb{R}^{C\times V}$, which summarizes the intrinsic oscillatory and modulatory characteristics of the system, serving as a robust conditional input for our model.

C.4 Maximum Mean Discrepancy

Forecasting the long-term evolution of chaotic systems necessitates metrics that extend beyond point-wise accuracy. To ensure our model reproduces not just a single trajectory but the system's intrinsic statistical and geometric structure, we employ a distributional loss based on the Maximum Mean Discrepancy (MMD).

As established in prior literature (Schiff et al., 2024), a suitable metric for comparing state distributions of trajectories should exhibit several essential characteristics. Specifically, it must: (i) respect the underlying geometry of the state space and be capable of comparing distributions with non-overlapping supports; (ii) provide an unbiased estimator that can be computed from finite samples; (iii) maintain low computational complexity with respect to both dimensionality and sample size; (iv) act as a true metric on the space of probability measures, ensuring that a vanishing distance implies convergence; and (v) feature parametric estimation rates, such that sample error is independent of the system's dimension.

The family of Integral Probability Metrics (IPMs) (Müller, 1997) provides a general framework that satisfies these desiderata. For any two probability distributions p_1 and p_2, an IPM is defined as the supremum of the difference between expectations over a class of functions $\mathcal{K}$:

$$\text{IPM}(p_1,p_2) = \sup_{\kappa\in\mathcal{K}} |\mathbb{E}_{\boldsymbol{u}\sim p_1}[\kappa(\boldsymbol{u})] - \mathbb{E}_{\boldsymbol{u}'\sim p_2}[\kappa(\boldsymbol{u}')]|. \tag{15}$$

Within this class, we select the Maximum Mean Discrepancy (MMD), which distinguishes itself by defining $\mathcal{K}$ as the unit ball in a Reproducing Kernel Hilbert Space (RKHS), denoted $\mathcal{H}$. The formal

definition of MMD is thus:

$$\text{MMD}(p_1, p_2) := \sup_{||f||_{\mathcal{H}} \leq 1} |\mathbb{E}_{\boldsymbol{u} \sim p_1}[f(\boldsymbol{u})] - \mathbb{E}_{\boldsymbol{u}' \sim p_2}[f(\boldsymbol{u}')]|. \tag{16}$$

By leveraging the reproducing property of the RKHS and the Riesz representation theorem, the squared MMD can be expressed in a convenient analytical form using a kernel function $\kappa(\cdot, \cdot)$ that defines $\mathcal{H}$:

$$\text{MMD}^2(p_1, p_2) = \mathbb{E}_{\boldsymbol{u}, \boldsymbol{u}' \sim p_1}[\kappa(\boldsymbol{u}, \boldsymbol{u}')] + \mathbb{E}_{\boldsymbol{v}, \boldsymbol{v}' \sim p_2}[\kappa(\boldsymbol{v}, \boldsymbol{v}')] - 2\mathbb{E}_{\boldsymbol{u} \sim p_1, \boldsymbol{v} \sim p_2}[\kappa(\boldsymbol{u}, \boldsymbol{v})]. \tag{17}$$

This expression leads directly to the unbiased empirical estimator used in our work as the regularization loss $\mathcal{L}_{\text{reg}}$.

For the kernel function κ, our implementation follows successful precedents (Seeger, 2004; Li et al., 2015; Schiff et al., 2024), employing a mixture of rational quadratic kernels. This choice ensures sensitivity to distributional discrepancies across multiple length scales. The composite kernel is formulated as:

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = \sum_{\sigma \in \boldsymbol{\sigma}} \frac{\sigma^2}{\sigma^2 + ||\boldsymbol{u} - \boldsymbol{v}||_2^2}, \tag{18}$$

where the set of scale parameters is chosen to be $\boldsymbol{\sigma} = \{0.2, 0.5, 0.9, 1.3\}$, consistent with these prior works.

D Details of Experimental Settings for Zero-shot Evaluations

D.1 Details of Synthetic Chaotic System Dataset

The study utilizes the large-scale synthetic dataset of chaotic dynamics introduced by Lai et al. (2025). This dataset is specifically designed to provide a vast and dynamically diverse corpus for pretraining a universal forecasting model, moving beyond reliance on a limited set of well-known systems. For completeness and the reader's convenience, we briefly summarize the methodology used by Lai et al. (2025) to create this dataset. Their generation pipeline is rooted in an evolutionary algorithm that discovers and validates novel chaotic ordinary differential equations (ODEs). **ADD**

Founding Population and Evolutionary Framework. The algorithm begins with a founding population of 129 well-documented, human-curated, low-dimensional chaotic systems (Gilpin, 2021; 2023). For these foundational systems, which include canonical examples like the Lorenz equations, the parameters and initial conditions are meticulously tuned to ensure operation within their chaotic regimes, and their integration timescales are standardized based on invariant mathematical properties such as Lyapunov exponents. From this seed set, the evolutionary framework iteratively generates new candidate systems through a cycle of mutation and recombination. The mutation step introduces variation by randomly sampling pairs of parent systems $\dot{\mathbf{x}} = f_a(\mathbf{x}, t; \theta_a)$ and $\dot{\mathbf{y}} = f_b(\mathbf{y}, t; \theta_b)$ as well as applying a parameter jitter, where random Gaussian noise is added to the default parameters of the selected ODEs ($\tilde{\theta}'_a \sim \mathcal{N}(\theta_a, \sigma)$, $\tilde{\theta}'_b \sim \mathcal{N}(\theta_b, \sigma)$). Subsequently, the recombination step combines the mutated parent systems to form a novel child system using a skew product construction:

$$\begin{cases} \dot{\mathbf{x}} = f_a(\mathbf{x}, t; \theta_a) \\ \dot{\mathbf{y}} = \kappa_b f_b(\mathbf{y}, t; \tilde{\theta}'_b) + \kappa_a f_a(\mathbf{x}, t; \tilde{\theta}'_a) \end{cases}$$

This method is chosen for its propensity to preserve chaotic dynamics under sufficiently weak or strong coupling. The scaling factors, κ_a and κ_b, are determined from the reciprocal of the root mean square (RMS), i.e., $\kappa = 1/\sqrt{\mathbb{E}||f(x,t)||^2}$ of a representative trajectory of the parent system.

Selection for Chaoticity. A critical and computationally intensive stage of the pipeline involves a rigorous, multi-step selection process that filters for genuine and sustained chaotic behavior, culling all other candidates. First, systems exhibiting trivial dynamics are rejected; the numerical integration is automatically terminated for any candidate that converges to a fixed point (indicated by an integration step size falling below 10^{-10}), diverges to infinity (a coordinate value exceeding 10^4), or fails to complete integration within a 5-minute time limit. Surviving candidates are then subjected to the 0-1 test, a standard method for distinguishing between chaotic and periodic or quasiperiodic dynamics. Finally, a further sequence of attractor tests is applied to ensure dynamical complexity. This

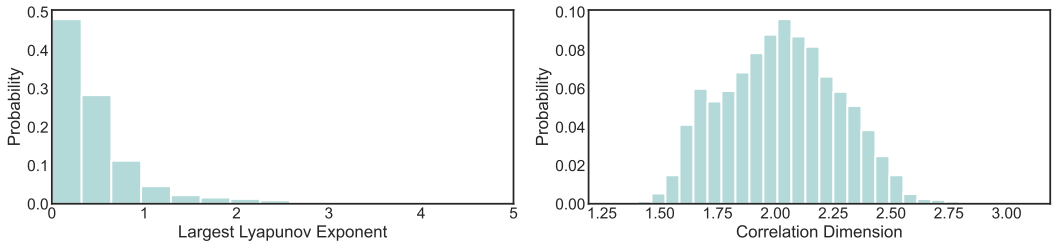

Figure 15: Distributions of the largest Lyapunov exponent and the correlation dimension of synthetic chaotic systems. **ADD**

includes a test based on near-recurrences to reject simple limit cycles, a power spectrum analysis to discard trajectories with only a few dominant frequencies, and an estimation of the largest Lyapunov exponent with the Rosenstein estimator (Rosenstein et al., 1993). This comprehensive discovery and validation process yields a final training corpus of $20K$ unique chaotic dynamical systems.

Data Augmentation and Trajectory Generation. To further expand the dataset's volume, several augmentations are applied to the generated trajectories. These transformations are selected because they preserve the underlying property that the resulting time series originates from a valid nonlinear dynamical system. The augmentations include random time-delay embedding, justified by Takens' embedding theorem (Takens, 2006), convex combinations, and affine transforms. For the final dataset, trajectories of 4096 timesteps are generated for each system using a high-precision numerical integrator with relative and absolute tolerances of 1×10^{-9} and 1×10^{-10}, respectively. Initial conditions are sampled from a preliminary, lower-tolerance integration run to approximate starting on the system's attractor.

Held-Out Test Set. For robust zero-shot evaluation, a distinct held-out test set of 9.3×10^3 systems is created. This set is generated from a reserved subset of 20 systems from the original 129 founding population that are never used in the training set generation. A strict separation is enforced by ensuring that none of these 20 systems, nor any of their mutations, appear as either a driver or a response in the skew product constructions for the training data, thereby preventing any data leakage.

Statistical Properties of Synthetic Systems. We conduct a comprehensive statistical analysis of the generated systems. Specifically, we compute the largest Lyapunov exponent for each system with the Rosenstein estimator (Rosenstein et al., 1993), and estimate the correlation dimension using the Grassberger-Procaccia (GP) algorithm (Grassberger & Procaccia, 1983). The histogram of these two critical invariants across synthetic chaotic systems is visualized in Figure 15. The heavy-tailed distribution of the largest Lyapunov exponent confirms that the dataset encompasses a broad spectrum of dynamical behaviors, ranging from weakly to strongly chaotic regimes. The correlation dimension displays a unimodal broad distribution, demonstrating the diversity of fractal geometries characterizing the synthetic strange attractors. **ADD**

Symbolic Divergence between Training and Held-Out Founding Systems. To quantitatively clarify that our evaluation regime tests for true zero-shot generalization rather than mere parameter-shift adaptation, we analyze the structural distinctness of the held-out founding test systems relative to those used for constructing the training dataset. Specifically, we represent the differential equations of all systems as symbolic expression trees and utilize the Tree Edit Distance (TED) to quantify symbolic structural similarity. It measures the minimum number of node operations (insertions, deletions, or re-labeling) required to transform one symbolic tree into another. Crucially, a TED of zero indicates that two systems share an identical functional topology and differ solely in their numerical coefficients, while any non-zero value implies a difference in the equation's functional terms. We compute the minimum TED for each held-out system against the entire set of founding systems used to construct the training dataset. The resulting distribution shown in Figure 16 is concentrated around a distance of 6. This substantial structural gap confirms that the held-out systems belong to topologically distinct equation families, demonstrating that the model's performance relies on universal dynamical learning rather than parameter interpolation within known structures. **ADD**

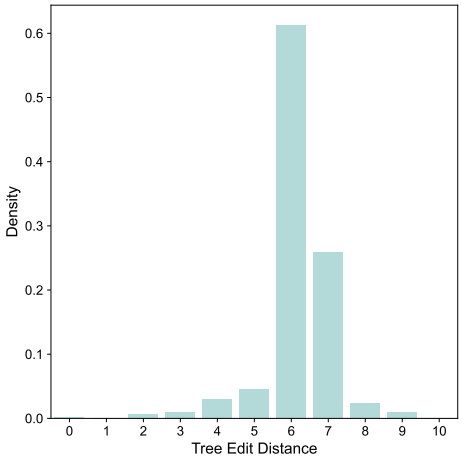

Figure 16: Distribution of minimum tree edit distance for each held-out founding system against the entire set of founding systems used to construct the training dataset. **ADD**

D.2 Details of Evaluation Metrics

To provide a comprehensive assessment of model performance, we employ a suite of evaluation metrics that quantify both short-term, point-wise prediction accuracy and the long-term fidelity of the reconstructed system dynamics. These metrics are designed to evaluate a model's ability to not only forecast the immediate future state but also to reproduce the intrinsic geometric and statistical properties of the chaotic attractor.

sMAPE. For evaluating short-term predictive quality, we utilize the Symmetric Mean Absolute Percentage Error (sMAPE) calculated over a forecast horizon of length T. The sMAPE provides a normalized, point-wise measure of the discrepancy between the predicted trajectory and the ground truth. It is defined as:

$$\text{sMAPE} \equiv \frac{200}{T} \sum_{t=1}^{T} \frac{\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_1}{\|\mathbf{x}_t\|_1 + \|\hat{\mathbf{x}}_t\|_1}, \tag{19}$$

where $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ are the true and forecasted state vectors at time step t, respectively. This metric is particularly well-suited for this task as its percentage-based formulation is robust to the varying scales of different dynamical systems, and it is less sensitive to outliers than the Mean Absolute Error (MAE).

Correlation Dimension Error D_{frac}. To assess a model's ability to replicate the long-term geometric structure, we evaluate its reproduction of the system's strange attractor. In a chaotic dynamical system, long-term trajectories populate a fractal object known as a strange attractor, which possesses a unique and invariant fractal dimension that characterizes its space-filling properties. We use the correlation dimension as a non-parametric method to estimate this fractal dimension directly from the time series data (Grassberger & Procaccia, 1983). This method quantifies how the number of points on the attractor scales with distance by measuring, for each point, the density of neighboring points within a given radius r. The fractal dimension is revealed by the power-law relationship between this point density and the radius r. We compute the correlation dimension for both the ground-truth trajectory and the attractor generated from the model's long-term forecast. The metric D_{frac} is then the root mean square error (RMSE) between these two estimated dimensions. A smaller D_{frac} value signifies that the model's generated dynamics faithfully reproduce the intrinsic geometric complexity of the true system's attractor.

Kullback–Leibler Divergence between System Attractors (D_{stsp}). Beyond geometric structure, a successful long-term forecast must also capture the statistical properties of the attractor. We quantify this using the Kullback-Leibler (KL) divergence (D_{stsp}) between the probability distributions of the true and reconstructed attractors (Hess et al., 2023; Göring et al., 2024). The long-term behavior of a chaotic system can be described by an invariant probability measure over its phase space, which

represents the likelihood of finding the system in a particular state. Operationally, we approximate this invariant measure for both the true and forecasted trajectories by fitting Gaussian Mixture Models (GMMs) to points sampled from each attractor. The D_{stsp} is then the estimated KL divergence between these two GMMs (Hershey & Olsen, 2007). A lower value indicates that the reconstructed attractor more accurately captures the statistical and density profile of the true system's dynamics.

Largest Lyapunov Exponent Error (D_{Lyap}). While geometric and statistical metrics (D_{frac} and D_{stsp}) assess the static shape and density of the attractor, they do not explicitly measure the temporal dynamics of system instability. To verify if the model captures the hallmark of chaos—sensitivity to initial conditions—we evaluate the Largest Lyapunov Exponent (LLE). The LLE quantifies the average exponential rate of divergence of infinitesimally close trajectories. We estimate the LLE for both the ground-truth trajectory and the model's long-term forecast using the Rosenstein estimator (Rosenstein et al., 1993). The metric D_{Lyap} is defined as the absolute difference between these two estimated exponents. A low D_{Lyap} value indicates that the model has successfully internalized the governing physical laws that drive the chaotic evolution, rather than merely memorizing superficial patterns. **ADD**

Weighted Mean Energy Error (ME_{LRw}). To rigorously evaluate the spectral fidelity of the forecasted trajectories, we assess the model's ability to reproduce the system's energy distribution across the frequency domain. While standard time-domain metrics may overlook spectral distortions hidden within smooth predictions, ME_{LRw} explicitly quantifies the deviation in the Power Spectral Density (PSD). To prioritize these dynamically significant components over background noise, we employ a weighted formulation defined as: **ADD**

$$\text{ME}_{\text{LRw}} = \sum_i w_i |\log(\frac{P_{\text{pred}}(f_i)}{P_{\text{true}}(f_i)})|, \tag{20}$$

where $P_{\text{pred}}(f_i)$ and $P_{\text{true}}(f_i)$ epresent the PSD values of the predicted and ground-truth trajectories at frequency f_i, respectively. The weighting coefficient w_i is normalized by the total energy of the ground truth signal:

$$w_i = \frac{P_{\text{true}}(f_i)}{\sum_j P_{\text{true}}(f_j)}. \tag{21}$$

This weighting mechanism ensures that the metric is sensitive to errors in high-energy frequency bands while being robust to negligible fluctuations in low-energy regimes. A lower ME_{LRw} indicates that the model has faithfully reconstructed the intrinsic oscillatory properties and energy profile of the chaotic system.

D.3 Details of Baselines

We compare our proposed method against several state-of-the-art time series foundation models, including Panda (Lai et al., 2025), Time-MoE (Shi et al., 2024), TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), Moirai-MoE (Liu et al., 2024a), and Timer-XL (Liu et al., 2024b). To assess the adaptability of general-purpose models to this specific domain, we also include Chronos-S-SFT, a variant of the Chronos-S model that has been fine-tuned on our chaotic systems training corpus. The key characteristics of each baseline are detailed below.

- **Panda** is a pretrained, encoder-only Transformer model designed for forecasting chaotic dynamics. Based on the PatchTST (Nie et al., 2022) architecture, it introduces interleaved channel and temporal attention layers to capture variable coupling, alongside a dynamics embedding layer that uses polynomial and Fourier features inspired by Koopman operator theory.
- **Time-MoE** is a family of billion-scale, decoder-only Transformer foundation models that utilize a sparse Mixture-of-Experts (MoE) architecture to enhance scalability and computational efficiency. The model tokenizes the input time series point-wise and employs multiple forecasting heads to predict at different resolutions simultaneously through multi-task optimization. Time-MoE is pre-trained on Time-300B, a large-scale collection of over 300 billion time points from diverse domains, to achieve universal forecasting capabilities.
- **TimesFM** is a decoder-only Transformer-based foundation model for zero-shot time series forecasting. It processes time series data by breaking it into patches and is trained autoregressively to predict the next patch based on the preceding context. A key design feature is using an output patch length that is longer than the input patch length to reduce the number of autoregressive

Table 9: The number of time points within the pretraining corpus of different methods.

Method	ChaosNexus	Panda	Time-MoE	TimesFM	Moirai-MoE	Timer-XL
# Time Points	∼0.35B	∼0.35B	∼300B	∼100B	∼231B	∼232B (LOSTA & UTSD)

steps required for long-horizon forecasting. The model is pretrained on a large corpus of approximately 100 billion time points, combining real-world data from Google Trends and Wikipedia with synthetic data.

- **Chronos** is a framework that adapts existing language model architectures, such as the T5 family, for probabilistic time series forecasting. Its core innovation is the tokenization of continuous time series values into a fixed vocabulary using a simple process of mean scaling and uniform quantization. By treating time series as a sequence of discrete tokens, Chronos is trained from scratch using the standard cross-entropy loss objective common to language models. The training corpus consists of a large collection of public datasets, augmented by synthetic data generated via Gaussian processes and a mixup strategy.
- **Moirai-MoE** is a decoder-only Transformer that improves upon its predecessor, Moirai (Woo et al., 2024), by incorporating a sparse Mixture-of-Experts (MoE) architecture. It replaces heuristic-driven, frequency-specific input/output layers with a single projection layer, delegating the task of modeling diverse time series patterns to specialized experts within the MoE layers, thereby enabling automatic token-level specialization. It also introduces a novel gating function that uses cluster centroids from a pretrained model to guide expert assignments. Moirai-MoE is trained on the LOTSA dataset using a decoder-only objective.
- **Timer-XL** is a causal, decoder-only Transformer designed for unified, long-context time series forecasting. It generalizes the next token prediction paradigm to multivariate time series by flattening 2D time series data into a unified context of patch tokens. Its central architectural innovation is TimeAttention, a causal self-attention mechanism that uses a Kronecker product-based mask and specialized position embeddings to effectively model both intra- and inter-series dependencies. Timer-XL is pre-trained on large-scale datasets, such as UTSD and LOTSA, to achieve state-of-the-art zero-shot performance.
- **Chronos-SFT.** To investigate the domain adaptability of general-purpose models, we create a specialized version of Chronos by fine-tuning the publicly available Chronos weights on our chaotic systems training set. This process, referred to as Supervised Fine-Tuning (SFT), allows the model to adapt its learned representations from general time-series data to the specific, complex patterns inherent in chaotic dynamics. This baseline helps to disentangle the effects of model architecture from the benefits of domain-specific training data.
- **DynaMix.** It is a foundation architecture specifically engineered for zero-shot dynamical systems reconstruction (DSR). It employs a Mixture-of-Experts (MoE) framework where the individual experts are Almost-Linear RNNs (AL-RNNs), capable of learning parsimonious dynamical representations. A context-aware gating network dynamically selects experts to generalize across diverse attractors without fine-tuning. To ensure the preservation of long-term invariant statistics, DynaMix is pretrained using sparse teacher forcing on a curated corpus of low-dimensional chaotic and cyclic systems, utilizing delay embeddings to reconstruct the underlying state space geometry. **ADD**
- **Parrot**. It serves as a robust, non-parametric baseline designed to probe the efficacy of learned representations in foundation models. It operates as an efficient in-context nearest-neighbor algorithm: by scanning the provided history for motifs that minimize Euclidean distance to the immediate context, it identifies the closest recurrence and directly copies the subsequent trajectory as the forecast. This approach exploits the determinism and recurrence inherent in strange attractors, demonstrating that simple pattern-matching strategies can often outperform complex deep learning models on chaotic benchmarks. **ADD**

We summarize the number of time points within the pretraining corpus in Table 9 for comparison. We demonstrate the parameter count in Table 10.

REVISE

Table 10: The number of parameters of baseline methods. For methods with mixture-of-experts layers, we demonstrate activated parameter counts/total parameter counts.

Method	ChaosNexus	Panda	Chronos-S	Chronos-B	Chronos-L	Moirai-MoE-S	Moirai-MoE-L	TimeMoE-S	TimeMoE-L	TimerXL	TimesFM
# Parameters	21M/58M	21M	21M	48M	205M	11M/117M	86M/935M	50M/113M	200M/453M	84M	500M

E Details of Training Setup and Computational Infrastructure

Training Setup. We train all ChaosNexus model variants for $100K$ iterations using a global batch size of 1024. The input context length is fixed at 512, and the model forecasts the subsequent 128 time steps. The initial patch size is set to 8. To enable efficient batching across heterogeneous systems, following the existing work (Lai et al., 2025), we randomly sample three channels from each multivariate trajectory to fix the training dimension at $d = 3$. This design aligns with the theoretical minimum of coupled variables required for continuous-time deterministic chaos (Strogatz, 2024). During inference, we process the full multivariate trajectories, since channel attention enables multivariate generalization. The training objective is a weighted sum of MSE, load balancing ($\lambda_1 = 0.1$), and MMD regularization ($\lambda_2 = 0.5$). To ensure convergence stability on chaotic data distributions, we employ the AdamW optimizer. The learning rate is set to 10^{-3} and follows a cosine decay schedule with 10% linear warmup. We also apply gradient norm clipping to 1.0 to mitigate gradient explosion, a common challenge in chaotic system modeling. We provide a detailed hyperparameter setting and discussions in Appendix B. For the Panda baseline, we use the same training setup as ChaosNexus for fair performance comparison. To construct the Chronos-S-SFT baseline, we fine-tune the Chronos model for $300K$ iterations using the AdamW optimizer. The per-device batch size is set to 512. The learning rate is initialized at 10^{-3} and follows a cosine decay schedule with a 10% linear warmup to ensure stable convergence. We apply gradient norm clipping with a threshold of 1.0 to mitigate gradient explosion. Weight decay is set to 0.0. To enhance the model's robustness, we incorporate a diverse set of augmentations during training, including Random Takens Embedding and Random Fourier Series. The implementation utilizes the Hugging Face Trainer framework with 16 dataloader workers to optimize throughput. For system-specific models, we follow the standard training and evaluation protocols provided in the Time-Series-Library [1] to ensure a fair comparison.

Computational Resources. All training experiments are conducted on a node equipped with 8 $\times$ NVIDIA A100 GPUs, each with 80GB memory. The training process requires approximately 10 hours without multi-GPU parallelization. Inference is performed on a single NVIDIA A100 GPU. Our implementation utilizes PyTorch with BF16 to optimize memory usage and throughput.

F Details of Experimental Settings for Few-shot Evaluations

F.1 Details of Weather Dataset

WEATHER-5K is a large-scale, public benchmark dataset designed to advance research in Global Station Weather Forecasting (GSWF) and broader time-series analysis. The dataset derives from the Integrated Surface Database (ISD), a global repository of surface observations managed by the National Centers for Environmental Information (NCEI). While the full ISD contains data from over 20,000 stations, many are unsuitable for machine learning applications due to being non-operational, having inconsistent reporting intervals, or containing significant missing values for key variables. The creation of WEATHER-5K involves a meticulous selection process to curate a high-quality subset of stations that are currently operational and provide long-term, hourly reporting of essential weather elements. After the preprocessing stages, the final dataset contains hourly meteorological data from 5,672 stations worldwide over a 10-year period (2014–2023), providing a rich and extensive resource for developing and benchmarking sophisticated forecasting models. Each station's data includes five primary meteorological variables: Temperature, Dew Point, Wind Speed, Wind Direction, and Sea-Level Pressure.

For reproducibility and standardized evaluation, the WEATHER-5K dataset is chronologically divided into three subsets: a training set, a validation set, and a testing set. The training set consists

[1] https://github.com/thuml/Time-Series-Library

of weather data from 2014 to 2021, the validation set includes data from the year 2022, and the testing set comprises data from 2023. This division follows an 8:1:1 ratio, which allows models to be trained on sufficient historical data, validated on a separate year, and tested on the most recent data for an accurate evaluation. For our experiments under few-shot setting conditions, we use only 0.1% and 0.5% of the training data, respectively.

F.2 Details of Baselines

We compare ChaosNexus against several strong deep learning baselines in this benchmark, including FEDformer (Zhou et al., 2022), CrossFormer (Zhang & Yan, 2023), PatchTST (Nie et al., 2022), and Koopa (Liu et al., 2023). The details are as follows:

- **FEDformer** is a Transformer architecture designed for long-term forecasting that addresses the tendency of standard Transformers to neglect global series properties, such as overall trends. It incorporates a seasonal-trend decomposition framework to disentangle the global profile of the series, which is processed separately from the more detailed components. Its core innovation is the replacement of the standard self-attention mechanism with frequency-domain operations. These Frequency Enhanced Blocks (FEB) and Frequency Enhanced Attention (FEA) modules operate on a randomly selected subset of Fourier or Wavelet basis functions, which not only captures the series' global properties more effectively but also achieves linear computational complexity.
- **CrossFormer** explicitly models the cross-dimension dependencies in multivariate time series, a factor often overlooked by models that focus primarily on temporal relationships. Its architecture is defined by three key components. First, a Dimension-Segment-Wise (DSW) embedding partitions each time series variable into segments, creating a 2D vector array that preserves both temporal and dimensional information. Second, a Two-Stage Attention (TSA) layer processes this array by first applying attention across the time axis and subsequently across the dimension axis. To handle a large number of variables efficiently, the cross-dimension stage uses a router mechanism to achieve linear complexity. Finally, these modules are integrated into a Hierarchical Encoder-Decoder (HED) that processes information at multiple scales to generate the final forecast.
- **PatchTST** introduces an efficient Transformer design centered on two principles: patching and channel-independence. The model first segments each univariate time series into patches, which serve as input tokens. This patching strategy retains local semantic information and quadratically reduces the computational and memory complexity of the attention mechanism, which in turn allows the model to process longer historical sequences. Subsequently, the model employs a channel-independent architecture, where each univariate series (channel) is processed individually by a shared vanilla Transformer encoder, thereby learning temporal patterns without explicit cross-channel mixing in the attention layers.
- **Koopa** is a forecasting model built on Koopman theory, specifically designed to handle non-stationary time series by linearizing their underlying dynamics. The model first employs a Fourier Filter to disentangle the series into time-invariant and time-variant components based on their frequency domain characteristics. It then applies distinct Koopman Predictors (KPs) to each component: a globally learned, parametric operator for the time-invariant dynamics, and locally computed, adaptive operators for the time-variant dynamics. These components are organized into stackable Koopman Blocks within a residual architecture, enabling hierarchical learning and end-to-end optimization of the forecasting objective without a reconstruction loss.

ADD

G Relations to Chaotic System Theories

G.1 Cross-system Generalization

We provide the mathematical intuition for why these components enable generalization across heterogeneous systems:

- **ScaleFormer architecture implements a multi-scale analysis.** Chaotic systems often exhibit multiple distinct timescales, for example, fast oscillations superposed on slow manifolds. the

shallow layers (i.e., fine scales) of ScaleFormer can capture high-frequency dynamics driven by the largest positive Lyapunov exponents, and the deep layers (i.e., coarse scales) capture the global attractor geometry associated with negative exponents. This architecture forces the model to learn the coupling mechanisms between timescales. Since diverse chaotic systems often share similar structural couplings (e.g., relaxational oscillations or bursting patterns) despite differing parameters and equations, explicitly disentangling these scales allows the model to transfer these learned dynamical patterns to unseen systems.

- **MoE layers serve as a basis expansion of local vector fields.** The evolution of a chaotic system can be described by $\dot{\mathbf{x}} = F(\mathbf{x})$. We hypothesize that while global attractors are varied across systems, local vector fields $F(\mathbf{x})$ can be decomposed into a set of local dynamical patterns (e.g., local saddle, spiral, or fold geometries). Mathematically, the MoE layer acts as a functional basis expansion. We view the experts $\{E_k\}_{k=1}^{M}$ as learned basis functions for local dynamics, MoE approximates the unknown vector field $F_{new}(\cdot)$ of an unseen system as:

$$F_{new}(\mathbf{x}) \approx \sum_{k=1}^{M} G_k(\mathbf{x}) \cdot E_k(\mathbf{x}), \tag{22}$$

  where $G_k(\cdot)$ denotes the gating coefficient. Generalization occurs because the model learns a reusable dictionary of experts E_k during training. When encountered a new system, the model performs an online system identification by exploring the optimal combination weights $G_k(\mathbf{x})$ from the inputs, allowing it to reconstruct complex dynamics from these shared basis.

- **Wavelet fingerprints have Lipschitz continuity to diffeomorphisms.** If a novel target x' is a deformed version of a source trajectory x, modeled by a diffeomorphism operator, the distance in our fingerprint Φ satisfies the bound:

$$||\Phi(x) - \Phi(x')|| \leq C||x' - x||. \tag{23}$$

  This bound theoretically guarantees that the mapping from the space of dynamical systems to our conditioning embedding space is stable and continuous. It ensures that structurally related systems, even if never seen during training, are mapped to a compact neighborhood in the feature space. It allows ChaosNexus to treat cross-system generalization as a smooth interpolation problem on a structured manifold.

G.2 Relation to Operator Theory

We discuss the relation of ChaosNexus to operator theory as follows:

- **First**, as detailed in Section 3.1 and Appendix C.1, we pre-process input patches $\mathbf{P}$ using random polynomial and Fourier features. Mathematically, it corresponds to constructing a finite dictionary of observables $\Psi(\mathbf{P})$. This step explicitly mimics the lifting process in extended dynamic mode decomposition (eDMD), projecting the highly nonlinear state evolution onto a higher-dimensional manifold where the dynamics are more amenable to linear approximation.
- **Second**, in the lifted space, the time evolution is governed by the Koopman operator $\mathcal{K}$, such that $\Psi(\mathbf{P}_{t+1}) = \mathcal{K}\Psi(\mathbf{P}_t)$. Our ScaleFormer backbone can be theoretically interpreted as a learnable, finite-dimensional approximation of this operator. Unlike traditional eDMD which approximates $\mathcal{K}$ with a static matrix, our ScaleFormer uses the attention mechanism to learn a state-dependent spectral decomposition. The attention weights effectively perform a dynamic eigenvalue decomposition, attending to the specific eigenmodes most relevant for the current phase space region, thereby handling the continuous spectrum often present in chaotic systems.

G.3 Relation to Invariants

We discuss the relation of ChaosNexus to invariants as follows:

- **First**, chaotic systems are characterized by a spectrum of Lyapunov exponents $\{\lambda_1, \lambda_2, ..., \lambda_d\}$. Positive exponents ($\lambda_i > 0$) drive exponential divergence, while negative negative exponents correspond to dissipative dynamics and attraction to the stable manifold. Our ScaleFormer architecture structurally aligns with this multi-scale dynamical structure. By processing input patches at progressively coarser resolutions, ScaleFormer explicitly disentangles these coupled timescales,

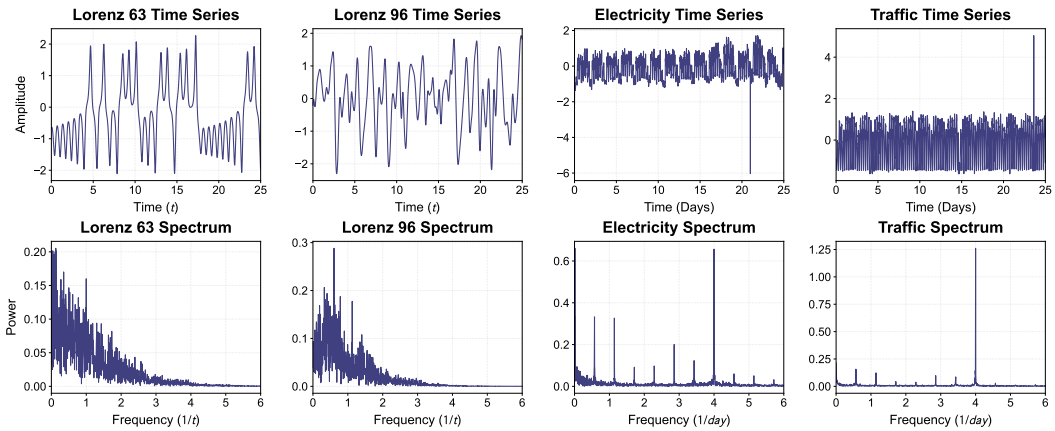

Figure 17: Comparison between chaotic systems and general time series **ADD**

where fine-scale layers capture high-frequency fluctuations and local error growth, corresponding to the dynamics driven by the largest positive Lyapunov exponents, and coarse-scale layers capture long-range dependencies and the global attractor topology, governed by negative Lyapunov exponents. This seperation prevents high-frequency chaotic mixing from obscuring the low-frequency invariant structure.

- **Second**, from the ergodic theory perspective, the long-term behavior of a chaotic system is characterized by an invariant physical measure. Our MMD loss minimizes the integral probability metric (IPM, Appendix C.4) between the predicted and true measures. Crucially, we instantiate the MMD with a mixture of rational quadratic (RQ) kernels. Since the RQ kernel is theoretically equivalent to an infinite-scale mixture of Gaussian kernels, it allows the metric to capture distributional discrepancies across a continuum of length scales. This capability ensures the model effectively learns the multi-scale geometry of the strange attractor, even when point-wise forecasting inevitably diverges.

ADD

H Comparison between Chaotic Systems and General Time Series

To elucidate the fundamental dynamical distinctions between chaotic systems and general real-world time series, we conduct a comparative spectral analysis juxtaposing the Lorenz63 system and the Lorenz96 system, against representative empirical time series of Electricity and Traffic that are considered by system-specific time-series forecasting models such as FEDFormer (Zhou et al., 2022). To ensure rigorous comparability across these disparate physical scales, all time series were standardized and aligned to visualize approximately 25 characteristic cycles, with the chaotic system time units calibrated against the daily periodicity of the empirical data. We then computed the Power Spectral Density (PSD) via Fast Fourier Transform (FFT) to map these temporal evolutions into a unified frequency domain ($1/t$ versus $1/\text{day}$), thereby isolating their underlying structural frequencies.

We demonstrate the results in Figure 17. The analysis reveals a stark topological dichotomy between the two system classes. Chaotic systems exhibit a continuous broadband spectrum, with energy distributed across a continuum of low frequencies without distinct isolated peaks, a hallmark of intrinsic aperiodicity. In contrast, the general time series exhibits a sparse line-spectrum structure, dominated almost entirely by a few fundamental frequencies (the daily cycle), with negligible energy in the intervening bands. This finding demonstrates that while real-world time series are typically governed by sparse, discrete periodic forcing, chaotic systems are fundamentally characterized by a continuous, multi-scale structure, in which dynamic complexity arises from a rich information density distributed across a broad temporal continuum rather than isolated frequencies.

ADD

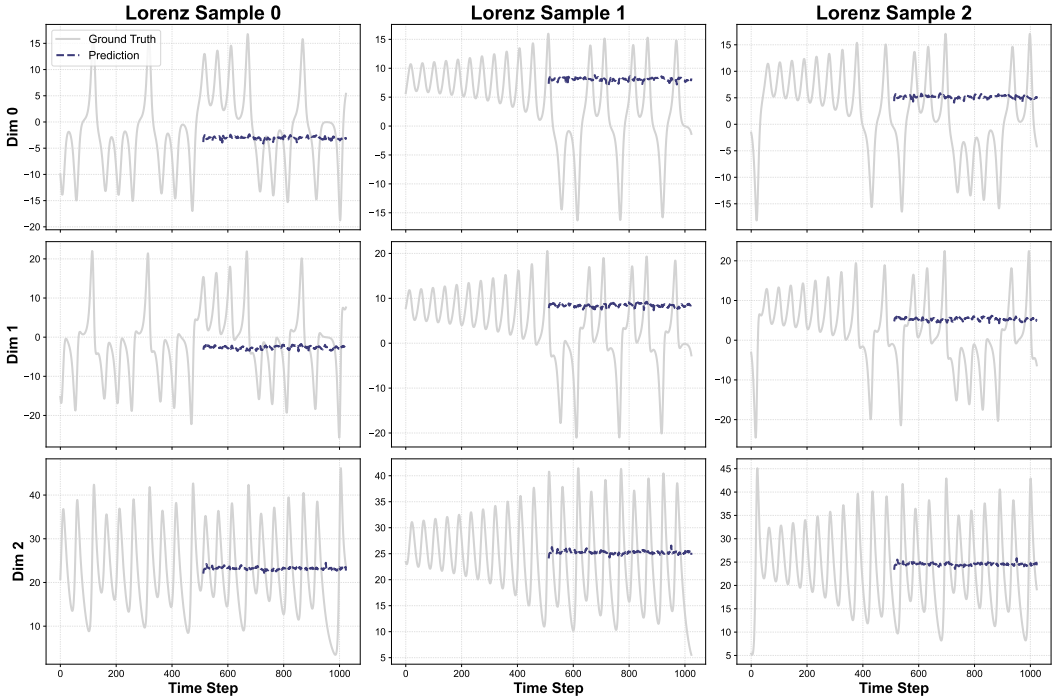

Figure 18: Performance Collapse of FEDFormer in Zero-Shot Forecasting **ADD**

I Performance Collapse of System-Specific Models in Zero-Shot Forecasting

To demonstrate the necessity of designing and training a foundation model for zero-shot chaotic system forecasting, we conduct a controlled experiment where a system-specific model, FEDFormer (Zhou et al., 2022), is trained on the exact training corpus as ChaosNexus. After the training process, we test the model on the canonical Lorenz63 system and demonstrate the results in Figure 18. We find that FEDFormer fails to capture the underlying chaotic dynamics given the context. The phenomenon indicates that without the specific design choices in ChaosNexus, system-specific models suffer from severe underfitting when exposed to highly heterogeneous dynamical systems, rendering them ineffective for zero-shot generalization.

J Usage of Large Language Model Declaration

The authors hereby declare the use of the Large Language Model (LLM) during the preparation of this paper. The role of the LLM is exclusively confined to language polishing and refinement of the manuscript's expression. All foundational and critical aspects of the research, including the formulation of the core ideas, the design of the proposed scheme, the planning of experiments, and the acquisition and analysis of all experimental data, are conducted without the assistance of any AI-based tools and are the sole contribution of the authors.

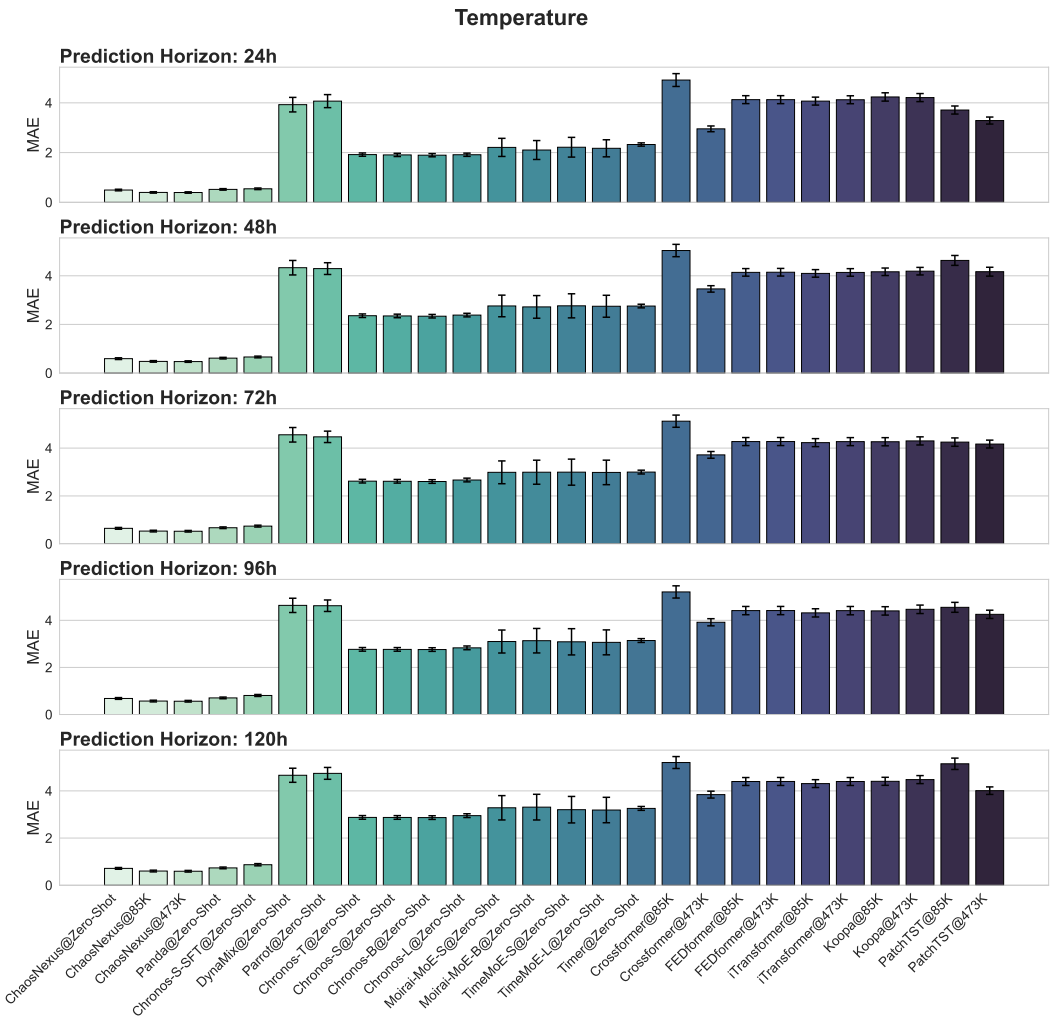

Figure 19: Forecasting performance for temperature on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples.

ADD

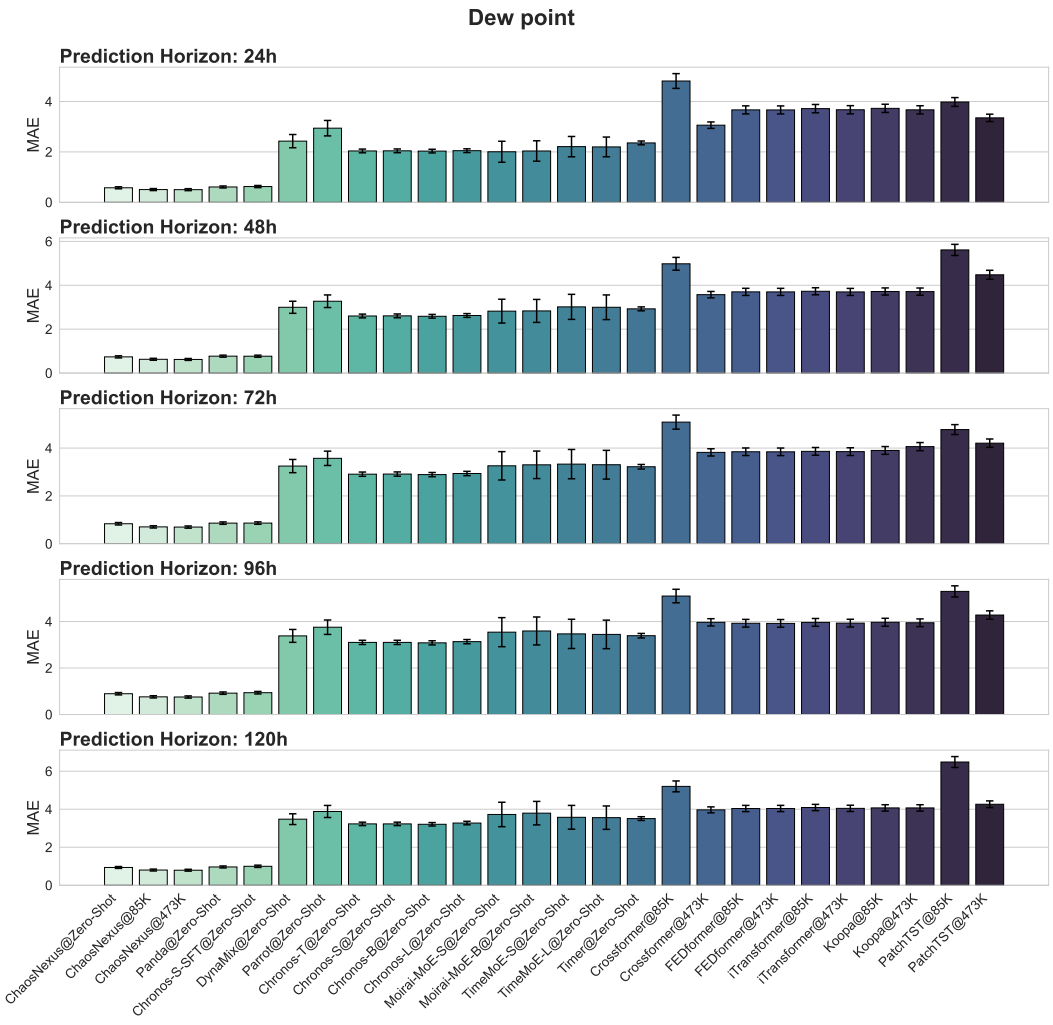

Figure 20: Forecasting performance for dew point on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

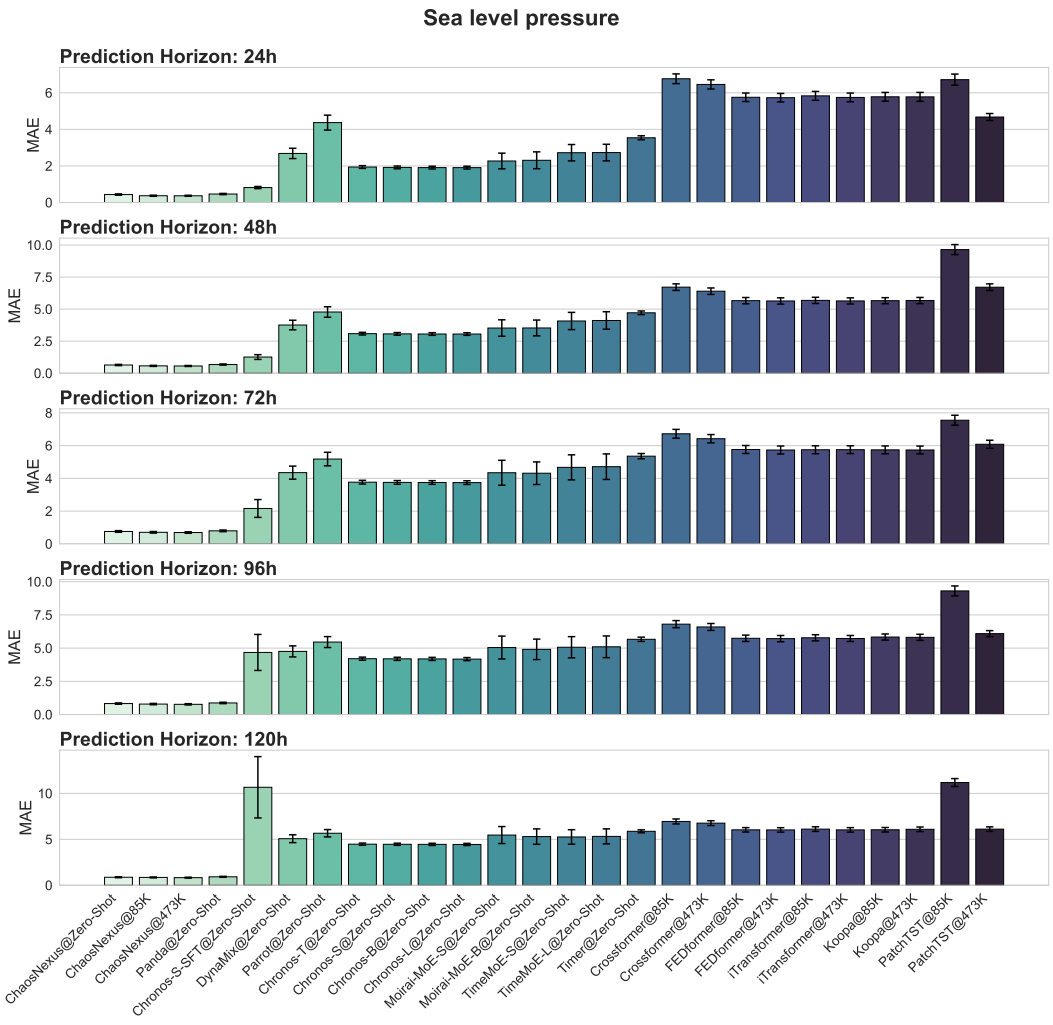

Figure 21: Forecasting performance for sea level pressure on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

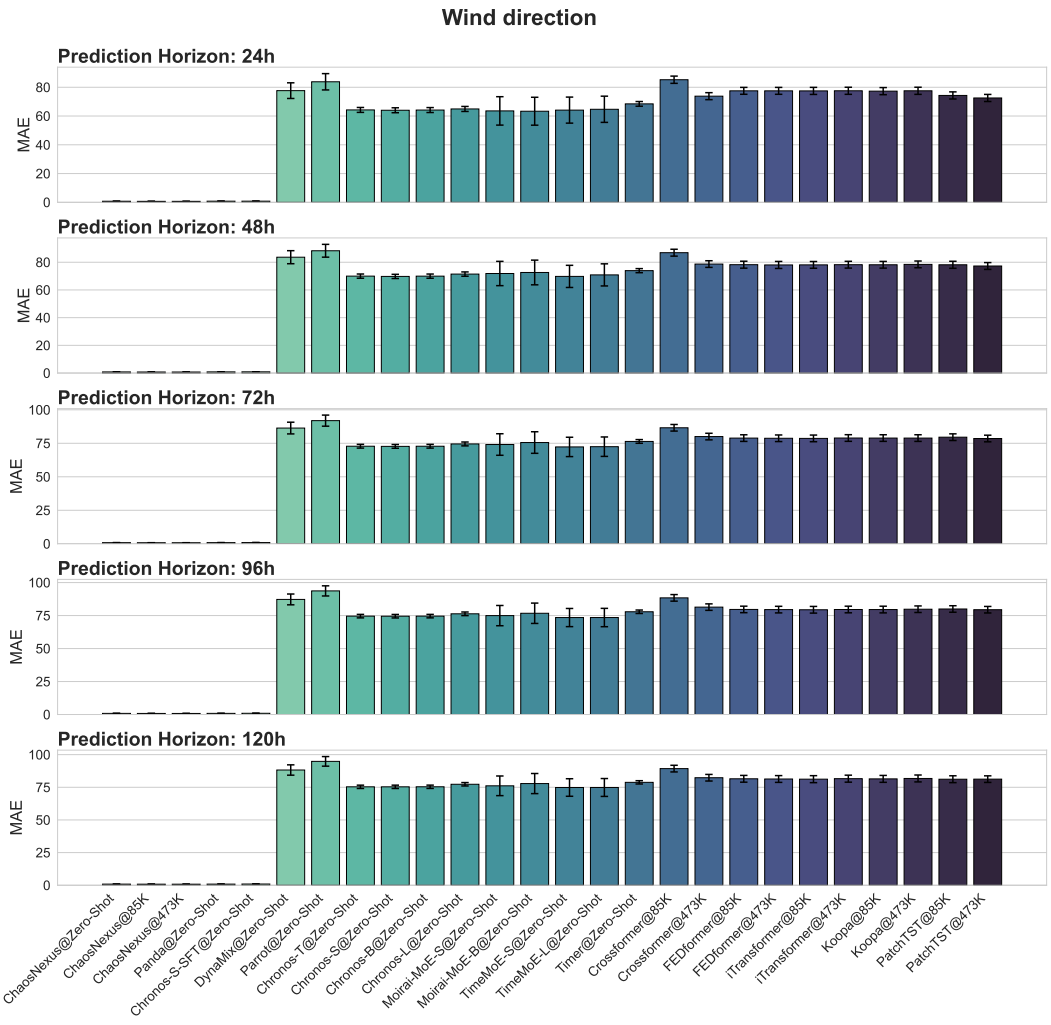

Figure 22: Forecasting performance for wind direction on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

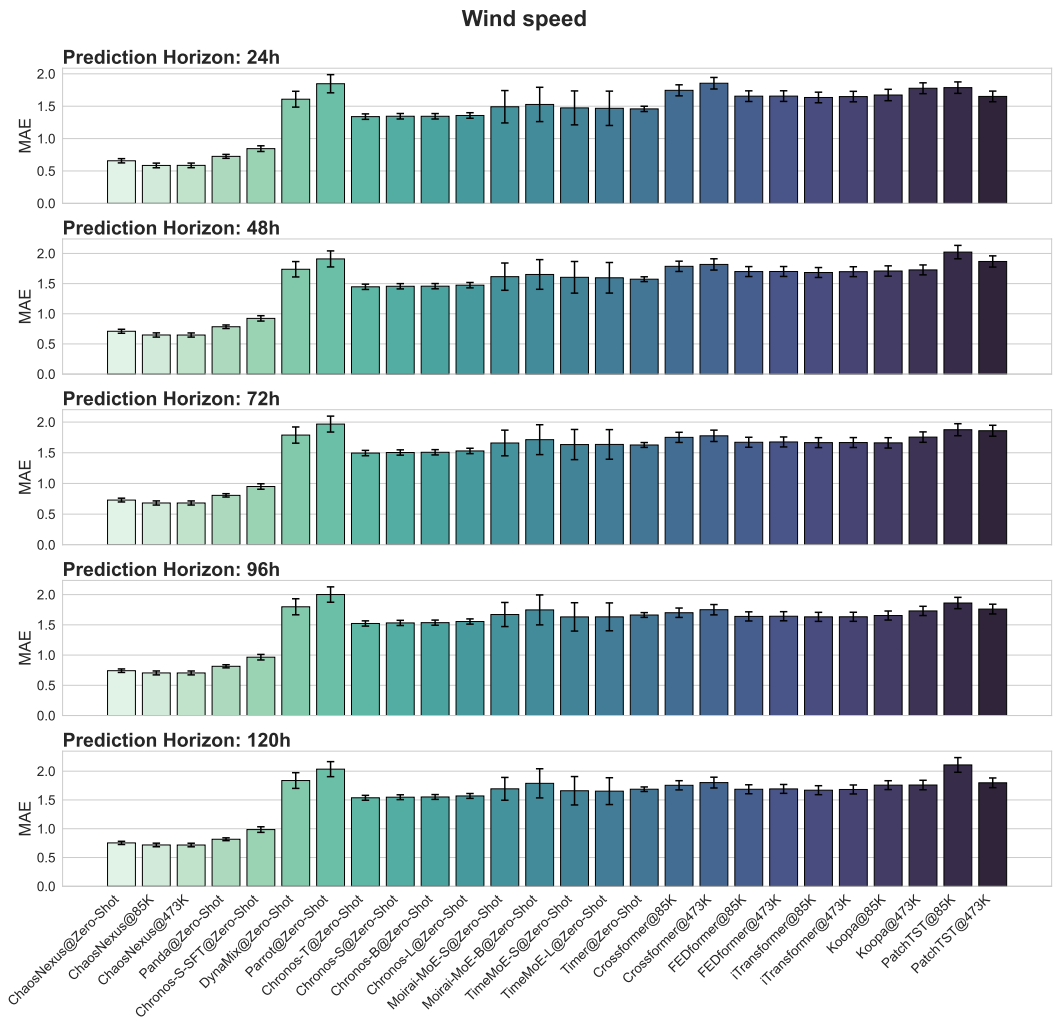

Figure 23: Forecasting performance for wind speed on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

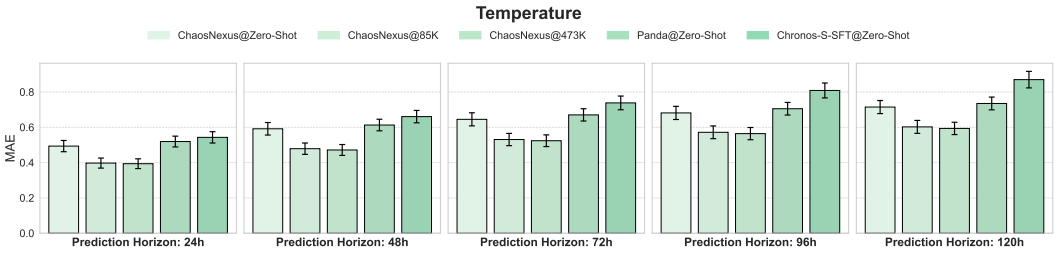

Figure 24: Forecasting performance for temperature on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. Only models previously trained with synthetic chaotic systems are reported. **ADD**

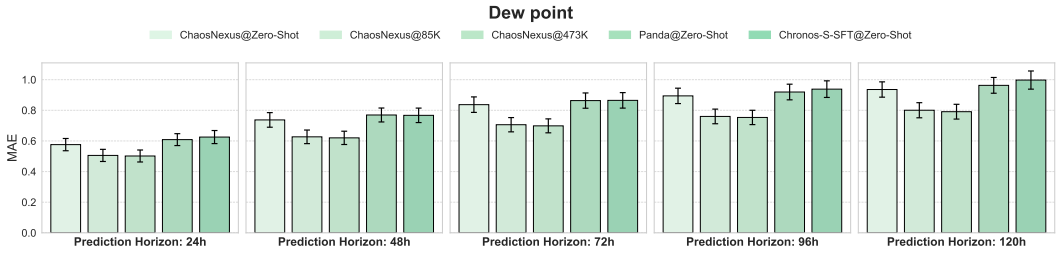

Figure 25: Forecasting performance for dew point on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. Only models previously trained with synthetic chaotic systems are reported. **ADD**

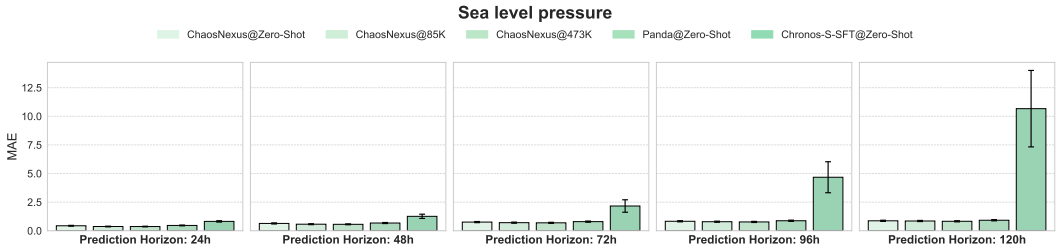

Figure 26: Forecasting performance for sea level pressure on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. Only models previously trained with synthetic chaotic systems are reported. **ADD**

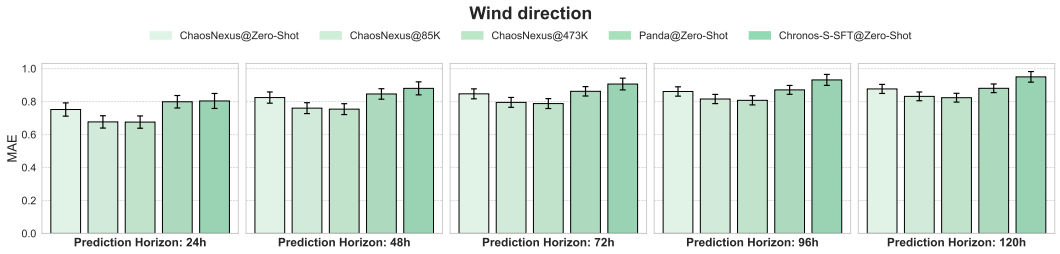

Figure 27: Forecasting performance for wind direction on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. Only models previously trained with synthetic chaotic systems are reported. **ADD**

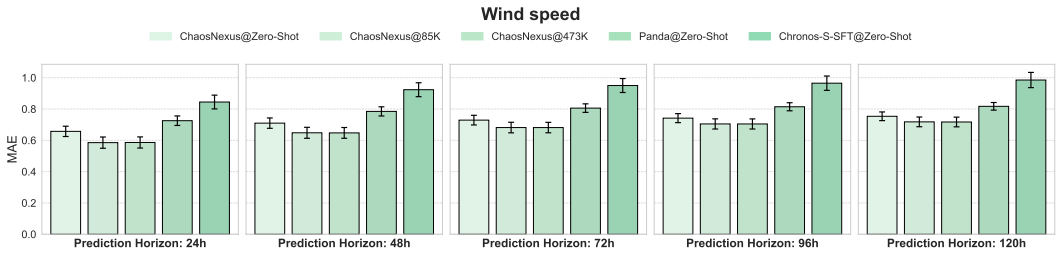

Figure 28: Forecasting performance for wind speed on the WEATHER-5K dataset. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. Only models previously trained with synthetic chaotic systems are reported. **ADD**

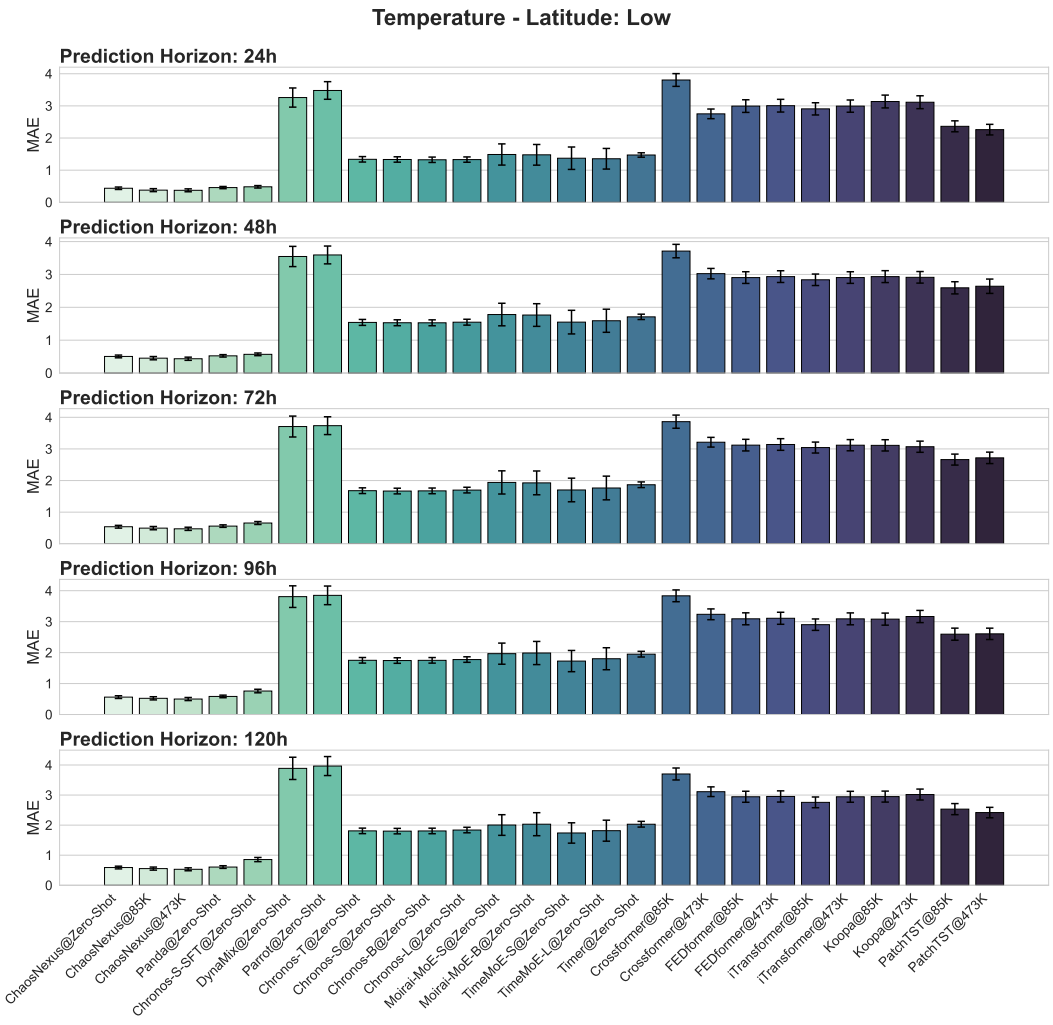

Figure 29: Forecasting performance for temperature of low latitude weather stations. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

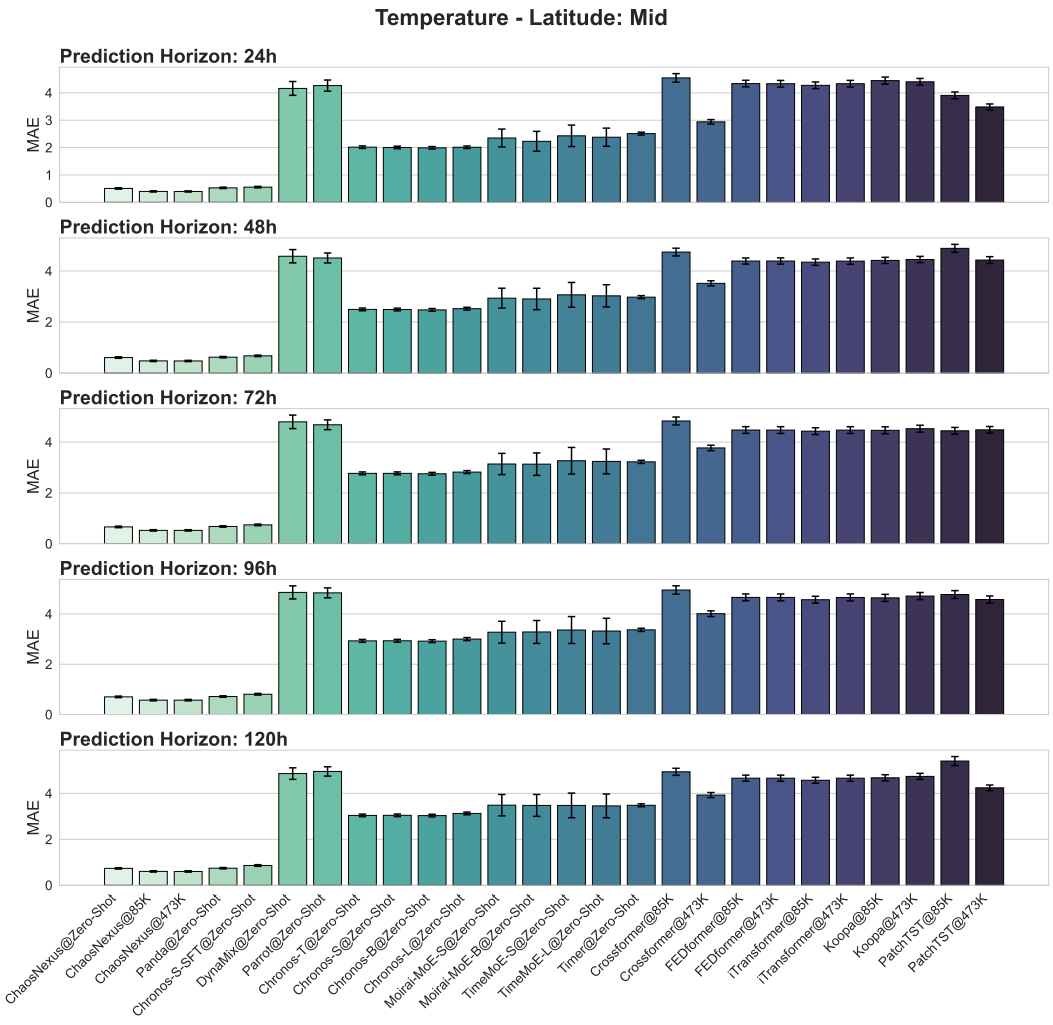

Figure 30: Forecasting performance for temperature of mid-latitude weather stations. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**

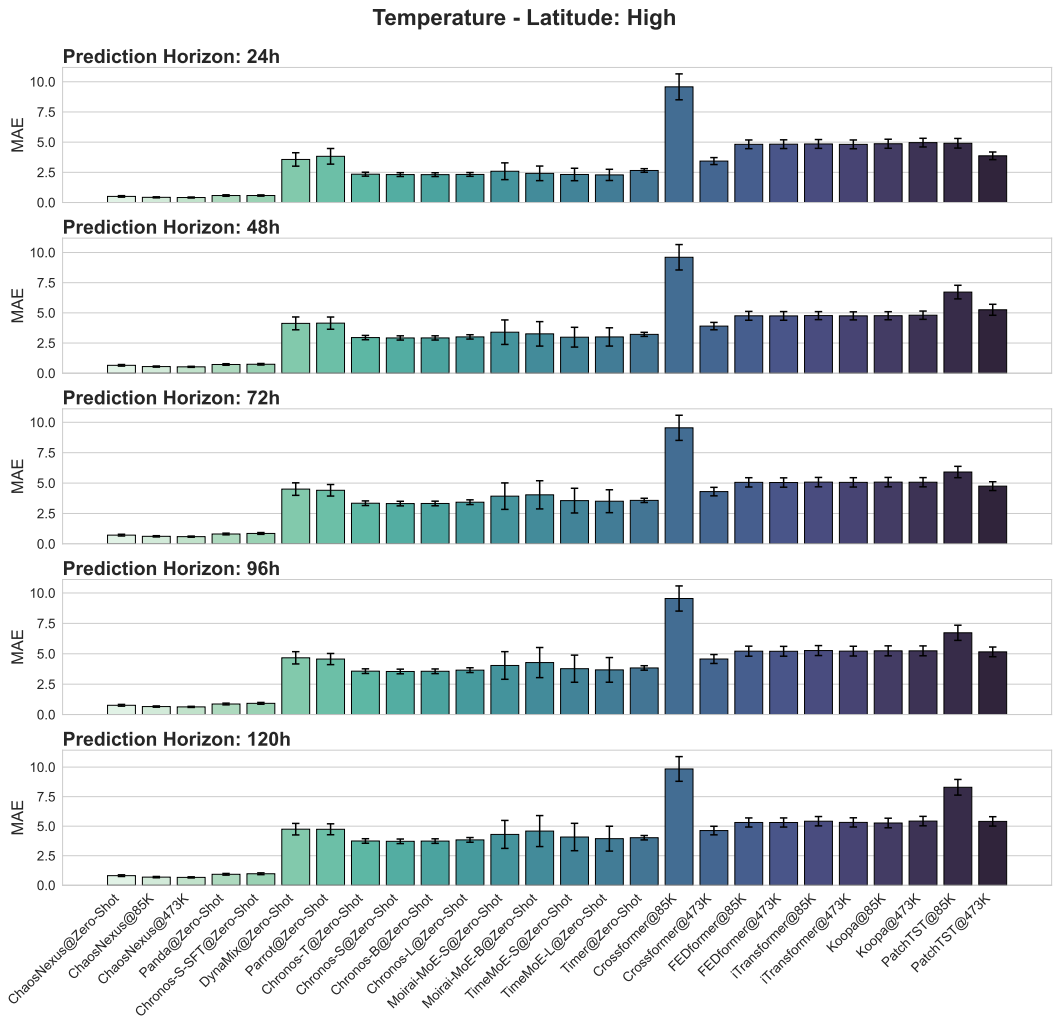

Figure 31: Forecasting performance for temperature of high latitude weather stations. The Mean Absolute Error (MAE) of ChaosNexus and baseline models is compared across multiple prediction horizons after fine-tuning on 85K (0.1%) and 473K (0.5%) samples. **ADD**