NEST: Tackling Dataset-Level Distribution Shifts via Regime-Oriented Mixture-of-Experts

Lanhao Li[1] , **Bingshu Xie**[1] , **Lijun Sun**[1] , **Xin Xue**[1] , **Haoyi Zhou**[1*] and **Jianxin Li**[1]
[1]Beihang University
{lilanhao, xiebingshu, sunlijun1, xuexin0102, haoyi, lijx}@buaa.edu.cn

Abstract

Accurate long-term forecasting in complex systems is frequently compromised by dataset-level distribution shifts, where diverse underlying behavioral modes and evolving system states drive the dynamic multivariate time-series. While existing methods predominantly focus on local temporal shifts, they fail to explicitly model the global structural challenge where datasets are composites of distinct operational regimes. In this paper, we propose NEST, a specialized framework designed to model and recompose these evolving structures through a two-phase dense MoE architecture. NEST first facilitates structural specialization by partitioning the dataset into distinct operational regimes through unsupervised clustering in a principled moment-entropy space. We introduce a regime-oriented router mechanism that generates initial expert weights based on temporal content, subsequently refined through geometric modulation to regime centroids. Crucially, rather than acting as monolithic predictors, individual experts function as specialized kernels that capture regime-specific dynamics by evolving unique variate-attention patterns. Extensive evaluations on diverse benchmarks, including heterogeneous network traffic and physical phenomena, demonstrate that NEST consistently achieves state-of-the-art performance. Our code and datasets are available at https://github.com/Aaralshin/NEST.

1 Introduction

Time series forecasting serves as a cornerstone for understanding and managing complex systems, ranging from the global web infrastructure [Basu *et al.*, 1996] and power grids [Boehme *et al.*, 2007] to intricate physical phenomena like space weather dynamics [Pulkkinen, 2007]. In these domains, the ability to provide accurate long-term predictions is essential for proactive resource allocation, anomaly detection, and ensuring system stability. However, real-world data streams are challenging to model due to their high non-stationarity and significant distribution shift across datasets.

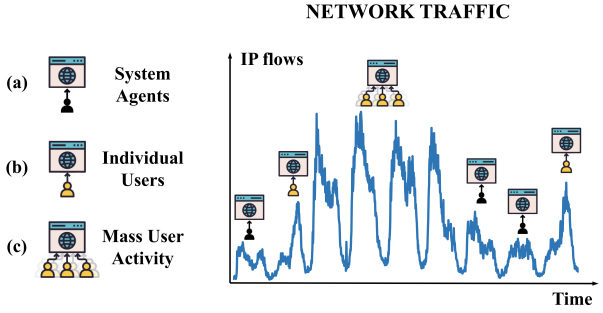

Figure 1: Conceptual illustration of dataset-level distribution shift. Using network traffic as an example, the IP flow volume fluctuates over time according to different underlying behavioral modes.

This distribution shift is not random; it is a manifestation of diverse underlying behavioral modes and evolving system states and can be recurrent during a long period. For example, as illustrated in Figure 1, this heterogeneity is driven by different underlying behavioral modes, each generating a traffic pattern with a distinct statistical signature. Icon (a) illustrates low-volume, baseline traffic might be generated by automated system check-ins or background processes; icon (b) illustrates standard daily peaks, which could correspond to individual users engaging in typical interactive sessions; icon (c) illustrates the highest traffic surges, often reflecting coordinated, large-scale activity, such as many users accessing a popular service simultaneously during a major event. A monolithic forecasting model, when trained on data containing this mixture of behaviors, is forced to learn a single, "averaged" representation that fails to accurately capture the specific dynamics of any individual mode, leading to suboptimal performance.

Despite its prevalence, few existing methods address this dataset-level shift. The predominant focus remains on mitigating the local temporal shift between the historical look-back window and the future prediction window. While valuable, this paradigm primarily corrects for evolving non-stationarity within a small temporal vicinity; it does not explicitly model the global, structural challenge where the dataset itself is a composite of distinct operational regimes.

*Corresponding Author

To address this fundamental problem, we argue for a systematic approach rooted in the identification of distinct operational regimes. We hypothesize that the diverse dynamic behaviors within a dataset can be effectively differentiated by moment-entropy metrics. Specifically, we employ low-order statistical moments (mean and variance) to capture distribution intensity, alongside Singular Value Decomposition Entropy (SVDEn) to quantify structural complexity. By mapping data slices into this moment-entropy space, we apply unsupervised clustering to automatically partition the dataset into functionally distinct categories. This categorization ensures that specialized experts can be trained to master the specific patterns inherent to each regime, effectively capturing the dynamic reorganization of inter-variable dependencies that occurs as the system transitions between different types of behavior.

To this end, we propose NEST: Tackling Dataset-Level Distribution Shifts via Regime-Oriented Mixture-of-Experts. NEST is a two-phase MoE framework designed to address dataset-level distribution shifts (DDS) by explicitly modeling the transitions between underlying system regimes. The process begins with unsupervised regime discovery, where we identify distinct operational modes through clustering in the moment-entropy space. Based on these discovered regimes, we then execute a structured two-phase training protocol to cultivate a heterogeneous expert pool. In *Phase 1*, we facilitate structural specialization within a heterogeneous expert pool; here, individual experts learn to model regime-specific sequence dynamics by evolving unique variate-attention patterns that capture the coupling logic of each mode. In *Phase 2*, the router generates an initial weight based on temporal content, which is then refined through static topological modulation to anchor the decision via geometric proximity to regime centroids. By dynamically recomposing these specialized kernels, NEST effectively transforms the challenge of distribution shifts into a manageable process of regime identification and dependency reconfiguration.

The key contributions of our work are as follows:

- **Structural Regime Modeling:** We propose **NEST**, a two-phase MoE framework that models the evolution of complex system regimes by dynamically recomposing inter-variable dependencies. By decoupling evolving dynamics into specialized kernels, NEST effectively captures the structural transitions between operational modes that drive dataset-level distribution shifts.
- **Principled Discovery and Routing:** We introduce an unsupervised discovery module using moment-entropy metrics to provide a mathematical basis for regime partitioning. Complementarily, we design a regime-oriented router that achieves precise expert orchestration by fusing expert weights initialization with distance-aware geometric modulation.
- **Empirical Validation and Interpretability:** NEST achieves state-of-the-art (SOTA) performance across diverse benchmarks, including heterogeneous network traffic and multi-year physical phenomena. Extensive analysis of attention polymorphism and weight permutation further validates its unique capability in the structural recomposition of variable dependencies.

2 Related Work

2.1 Deep Learning for Time Series Forecasting

Early efforts leveraged RNNs like LSTM [Siami-Namini *et al.*, 2018] and GRU [Weerakody *et al.*, 2021] for temporal gating, though sequential constraints hindered long-horizon efficiency. The transition to Transformers [Wu *et al.*, 2020] introduced global dependencies, yet quadratic complexity prompted sparse or hierarchical variants like Informer [Zhou *et al.*, 2021] and Pyraformer [Liu *et al.*, 2021]. Subsequent innovations focused on domain-specific kernels: Autoformer [Wu *et al.*, 2021] utilized auto-correlation, while FEDformer [Zhou *et al.*, 2022] applied frequency-domain attention via Fourier/wavelet transforms [Torrence and Compo, 1998; Cleveland *et al.*, 1990]. Recent paradigms emphasize structural rethinking: PatchTST [Nie *et al.*, 2022] processes sub-series patches for local semantics, while iTransformer [Liu *et al.*, 2023a] inverts the architecture to model multivariate correlations by treating whole variates as tokens. Concurrently, DLinear [Zeng *et al.*, 2023] demonstrated the efficacy of simple MLP-based decomposition. More recently, unified backbones like UniTS [Gao *et al.*, 2024] have emerged as foundation models, achieving robust zero-shot generalization across diverse forecasting tasks.

2.2 Handling Distribution Shifts

Recent studies address non-stationarity in time series by adapting normalization or learning strategies at the *instance* or *window* level. Representative methods include instance-wise [Ogasawara *et al.*, 2010] learnable input normalization [Passalis *et al.*, 2019], and reversible instance normalization (RevIN [Kim *et al.*, 2021]), which removes non-stationary components during training and restores them during inference. Dish-TS[Fan *et al.*, 2023] proposes a general neural paradigm that mitigates distribution shift by employing a dual framework to separately model the distributions of the input and output spaces. FAN[Ye *et al.*, 2024] performs normalization by leveraging frequency-domain information. Subsequent work extends these ideas through slice-based normalization, such as SAN [Liu *et al.*, 2023b], where each slice is defined within the look-back window and the forecast horizon. However, as these methods focus on modeling distribution shifts between the look-back and forecast windows, they exhibit strong adaptability to short-term fluctuations but limited generalization under prolonged or regime transitions within the dataset, as they overlook the discrepancies induced by variations across training instances.

3 Preliminaries

3.1 Problem Definition

The primary task addressed in this paper is long-term, multivariate time series forecasting. Given a historical multivariate time series, the goal is to learn a forecasting function $f(\cdot)$ that accurately predicts a sequence of future values based on a window of past observations.

Formally, let the input be a look-back window of length L, denoted as $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, where each multivariate observation $x_t \in \mathbb{R}^d$ is a d-dimensional vector at time step t. The model must learn a function f that maps the input window to the subsequent H time steps, where H is the prediction horizon:

$$\hat{\mathbf{y}} = f(\mathbf{x}) \tag{1}$$

Here, $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_H)$, where each $\hat{y}_t \in \mathbb{R}^d$, is the predicted sequence. The objective is to minimize the discrepancy between the prediction $\hat{\mathbf{y}}$ and the ground truth future sequence $\mathbf{y} = (y_1, y_2, \ldots, y_H)$.

The fundamental challenge we address is dataset-level distribution shift (DDS), where a long-term time series is composed of multiple, heterogeneous operational regimes. Unlike local non-stationarity, DDS reflects the structural evolution of the system over extended horizons. For instance, in ionospheric total electronic content(TEC) data, the decadal solar cycles introduce profound shifts in signal characteristics that span years; similarly, in network traffic, the transition from background maintenance to coordinated user surges represents a fundamental change in system behavior. Mathematically, a monolithic function f struggles in this scenario because the mapping between history and future is not invariant. Instead, the underlying inter-variable dependencies are dynamically reorganized as the system transitions between regimes. When a single model is forced to minimize a global objective across this mixture of distinct behaviors, it inevitably learns an "averaged" representation. This averaging effect obscures the specific coupling logic of individual regimes, leading to a loss of fidelity in capturing the precise dynamics required for robust long-term forecasting. To resolve this, it is necessary to identify these regimes and master their unique patterns via specialized kernels.

4 Methodology

4.1 Overall Framework

To address the challenge of dataset-level distribution shift, we propose NEST: Tackling Dataset-Level Distribution Shifts via Regime-Oriented Mixture-of-Experts. The framework is built upon the principle of regime-specific structural modeling, where the heterogeneous operational modes within a long-term time series are first identified and then masterfully captured by a specialized expert pool. Instead of learning a single averaged representation, NEST anchors its predictive logic in the distinct dynamical patterns inherent to different system states.

The NEST architecture comprises three key components:

- **Unsupervised Regime Discovery:** A principled module that partitions the dataset into functionally distinct regimes. By employing **moment-entropy metrics**, it characterizes each data segment through its statistical distribution and structural complexity, ensuring a robust mathematical basis for regime identification.
- **Specialized Expert Pool:** A heterogeneous ensemble consisting of specialized "regime experts" and a "shared expert". Each regime expert is trained to achieve structural specialization, evolving unique variate-attention patterns to capture the specific inter-variable dependencies of its assigned regime, while the shared expert retains global, overarching knowledge.
- **Regime Oriented Router:** A dynamic gating network that orchestrates the expert pool through a functional division of labor. The router achieves precise expert selection by integrating content-based temporal regime alignment with geometric contextual modulation, enabling the model to accurately recompose specialized patterns in response to evolving system dynamics.

The NEST pipeline, illustrated in Figure 2, follows a two-phase process. **Phase 1** consists of an unsupervised regime discovery step followed by the training of a specialized expert pool on the discovered data partitions. In **Phase 2**, these experts are frozen, and the router is trained to adaptively aggregate their predictions for context-aware forecasting.

4.2 Unsupervised Regime Discovery via Data Clustering

The foundational step of our NEST framework is the automated identification of distinct operational regimes within the time series dataset. This is achieved through an unsupervised clustering process performed as a pre-processing step before the main model training.

First, we transform the training time series into a set of overlapping instances, or "slices". Each slice $\mathbf{s}_i$, starting at time index i, is defined as a sequence of length $SL = L + H$, where L is the input window length and H is the prediction horizon:

$$\mathbf{s}_i = \{x_t\}_{t=i}^{i+SL-1} \tag{2}$$

Next, for each slice $\mathbf{s}_i$, we extract a low-dimensional feature vector $\mathbf{v}_i$ designed to capture its core dynamic properties. This vector includes: **Moment Metric:** We compute the mean (μ) and variance (σ^2) of the slice, representing its central tendency and volatility. **Entropy Metric:** To capture intrinsic complexity, we compute the Singular Value Decomposition Entropy (SVDEn). This involves constructing a trajectory matrix $\mathbf{X}$ from the slice $\mathbf{s}_i$ with an embedding dimension d:

$$\mathbf{X} = \begin{pmatrix} s_1 & \ldots & s_{SL-d+1} \\ \vdots & \ddots & \vdots \\ s_d & \ldots & s_{SL} \end{pmatrix} \in \mathbb{R}^{d \times (SL-d+1)} \tag{3}$$

We then perform SVD on $\mathbf{X}$ to obtain its singular values $\{X_{\sigma_j}\}_{j=1}^{k}$. These values are normalized to form a probability distribution $p_j = X_{\sigma_j} / \sum_{l=1}^{k} X_{\sigma_l}$. The SVDEn is the Shannon entropy of this distribution:

$$\text{SVDEn}(\mathbf{s}_i) = -\sum_{j=1}^{k} p_j \log(p_j) \tag{4}$$

The final feature vector for each slice is the concatenation of these metrics: $\mathbf{v}_i = [\mu(\mathbf{s}_i), \sigma^2(\mathbf{s}_i), \text{SVDEn}(\mathbf{s}_i)]$. With the training set represented as a collection of feature vectors $\{\mathbf{v}_i\}$, we apply K-Means clustering to partition the data into M clusters. The objective is to find the set of centroids

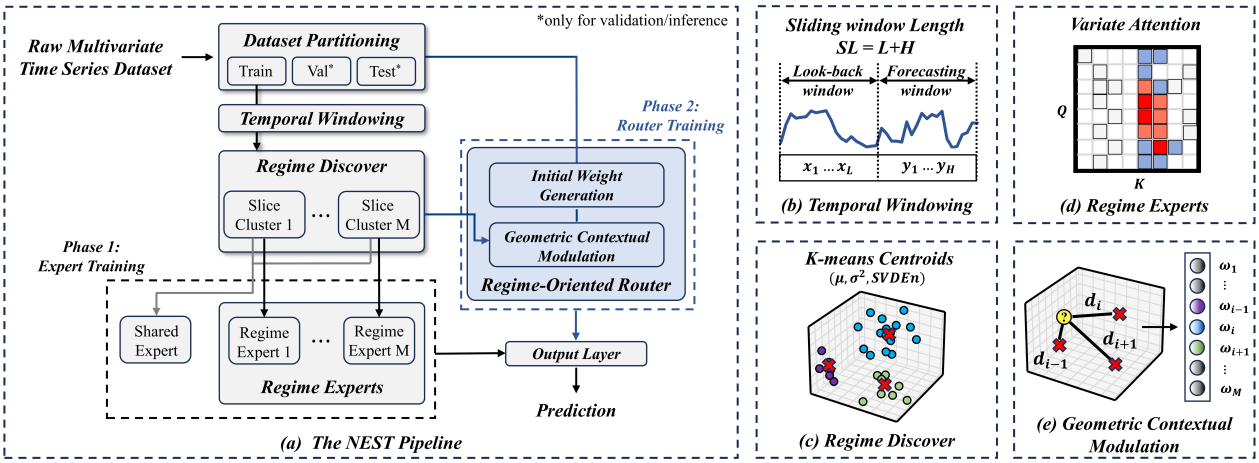

Figure 2: **Overview of the proposed NEST framework and its key components. (a) The NEST Pipeline:** Shows the complete two-phase pipeline. **Phase 1** performs regime discovery and expert training. **Phase 2** freezes the experts and trains the regime-oriented router to adaptively combine their outputs for the final prediction. **(b) Temporal Windowing:** Illustrates the process of creating data slices of length SL from the raw time series. **(c) Regime Discovery:** A conceptual visualization of the regime discovery process, where data slices are clustered in a feature space defined by their statistical and complexity features. **(d) Regime Expert:** Each regime expert consists of a variate-attention mechanism to model inter-variate relations. **(e) Geometric Contextual Modulation:** Modulates expert weights based on the feature-space distance of an input sample to the pre-computed regime centroids after the initialization of the weights.

$C = \{\mathbf{c}_1, \ldots, \mathbf{c}_M\}$ that minimizes the within-cluster sum of squares. Each centroid $\mathbf{c}_m$ serves as a prototypical representation of its regime and is utilized by the router.

A crucial detail of our regime discovery process is the use of data slices spanning both the look-back window and the prediction horizon. This design choice ensures that the discovered regimes are defined by the complete dynamic behavior of a sequence, including its evolution into the future, thus creating more forecasting-oriented clusters. At inference time (i.e., during validation and testing), only the look-back window of length L is available. To assign an incoming sequence to a regime, we compute its feature vector (mean, variance, SVDEn) using solely this available L-length data. We then operate under the assumption that the statistical properties of the look-back window serve as a sufficient proxy to identify the most likely regime for the complete $L+H$ dynamic. That is, from the historical look-back window to the forecasting window, the data distribution remains consistent, which can be ensured by the method discussed in Section 2.2. This procedure strictly avoids data leakage, as the ground truth future values are never used to compute features or determine regime assignment at inference time. The regime centroids are established exclusively from the training set, and the mapping during inference relies only on historical data.

4.3 The Multi-Expert Pool

The predictive core of the NEST framework is a heterogeneous pool of expert models, each designed to capture different facets of the time series dynamics. This pool contains two types of experts: specialized regime expert sets and a single, generalist shared expert.

Regime Experts: Corresponding to the M regimes discovered via moment-entropy metrics, the model contains M specialized regime experts, denoted as $\{E_m\}_{m=1}^{M}$. Each expert E_m is responsible for learning the specific dynamic properties of its assigned regime m. To effectively capture the inherent pattern of the time series, each expert is implemented as a variate-attention module specifically optimized to capture regime-specific inter-variable dependencies. Given an input sequence $\mathbf{x} \in \mathbb{R}^{L \times C}$, where L is the look-back window and C is the number of variables, the expert focuses on the spatial correlations across the channel dimension:

$$E_m(\mathbf{x}) = \text{Attention}_{\text{variate}}(\mathbf{x}) \tag{5}$$

This design ensures that each expert masters the dynamic reorganization of variables within its assigned regime. The resulting attention maps provide an explicit representation of how variable dependencies shift across different operational states.

Shared Expert: To capture global patterns common across all regimes, the pool includes a single shared expert, E_s, which remains active across different data slices throughout Phase 1 training and employs variate attention to provide a stable, foundational forecast.

4.4 Regime-Oriented Router

The core of NEST's adaptability is the regime-oriented router, which dynamically orchestrates the expert pool through a two-step process to ensure stable and precise expert selection under complex distribution shifts.

Initial Weight Generation

The router first dynamically generates a preliminary weight distribution based on the input's temporal content. For the M regime experts, the input embedding $\mathbf{x}_{\text{emb}}$ is processed by a lightweight Transformer encoder and a regime-specific linear

head, yielding the preliminary weight distribution:

$$\mathbf{w}_{\text{init}} = \text{Softmax}(\text{Linear}_{\text{regime}}(\text{Encoder}(\mathbf{x}_{\text{emb}}))) \quad (6)$$

In parallel, the output of the same encoder is passed through a different linear head and a Sigmoid activation function to compute the weight of the shared expert, w_s:

$$w_s = \text{Sigmoid}(\text{Linear}_{\text{shared}}(\text{Encoder}(\mathbf{x}_{\text{emb}}))) \quad (7)$$

Using a Sigmoid function allows the model to treat the shared expert as an independent, additive component, without competing directly with the specialized regime experts for the same weight budget.

Geometric Contextual Modulation

To anchor the content-driven decisions within the global data manifold, we modulate $\mathbf{w}_{\text{init}}$ using the input's proximity to regime centroids in the moment-entropy space. For an input slice $\mathbf{s}_i$, we calculate the Euclidean distance d_m between its moment-entropy metric vector $\mathbf{v}_i$ and each regime centroid $\mathbf{c}_m$:

$$d_m = ||\mathbf{v}_i - \mathbf{c}_m||_2 \quad (8)$$

The distance is converted into a similarity weight $\tilde{\omega}_m$ via an inverse quadratic function. A softening factor $\alpha \in (0, 1]$ is applied to facilitate smooth collaboration during regime transitions:

$$\tilde{\omega}_m = \left(\frac{1}{1 + d_m^2}\right)^{\alpha} \quad (9)$$

The final adaptive weights w'_m are produced by the element-wise modulation of temporal relevance and geometric context:

$$w'_m = w_{\text{init},m} \cdot \tilde{\omega}_m \quad (10)$$

Final Prediction and Training.

The modulated scores are renormalized to produce the final regime expert weights $\mathbf{w}_{\text{regime}}$. A separate gate computes the weight w_s for the shared expert. The final prediction $\hat{\mathbf{y}}$ is the weighted sum of all expert outputs:

$$\hat{\mathbf{y}} = w_s E_s(\mathbf{x}) + \sum_{m=1}^{M} w_m E_m(\mathbf{x}) \quad (11)$$

The router training objective minimizes the Mean Squared Error (MSE) loss between the prediction $\hat{\mathbf{y}}$ and the ground truth $\mathbf{y}$:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 \quad (12)$$

5 Experiments

5.1 Experimental Setup

Datasets. To evaluate NEST's ability to capture evolving variable dependencies across diverse domains, we conduct experiments on several large-scale benchmarks: CESNET-TIMESERIES24 [Koumar *et al.*, 2025]: A high-fidelity network traffic dataset containing 40 weeks of activity from over 275,000 IPs on an ISP network. Its high variability and heterogeneous behavioral modes present an authentic challenge for regime identification. We select five representative IP series for primary evaluation. Ionospheric TEC: A specialized physical dataset characterizing Earth's ionospheric dynamics which features decadal solar cycles and complex spatiotemporal relations. Public Benchmarks: To demonstrate generalizability, we include the Weather and ETT datasets(ETTh1, ETTh2), which are standard benchmarks for evaluating long-term forecasting robustness under varying non-stationarity.

Baselines. We compare our proposed NEST model against a comprehensive suite of SOTA and classical deep learning models for time series forecasting. The baselines include the unified foundation model UniTS[Gao *et al.*, 2024]; Transformer-based models such as iTransformer[Liu *et al.*, 2023a], PatchTST[Nie *et al.*, 2022], FEDformer[Zhou *et al.*, 2022], Autoformer[Wu *et al.*, 2021], and Informer[Zhou *et al.*, 2021]; and the MLP-based DLinear[Zeng *et al.*, 2023]. This diverse set of models allows for a rigorous comparison against different architectural paradigms. We perform hyperparameter search for a fair comparison.

Implementation Details. To ensure a fair comparison, all models were trained and tested on a single NVIDIA V100 32GB GPU. We adopt the same strict chronological split for all benchmarking datasets: 70% for training, 10% for validation, and 20% for testing. We conduct our main experiments with a fixed lookback window of 512 time steps.

5.2 Main Results and Analysis

The main forecasting results are presented in Table **??**, which provides a comprehensive comparison of NEST against all baseline models across nine datasets and four prediction horizons. The analysis of these results highlights the significant and quantifiable advantages of our proposed architecture in handling complex, non-stationary long-span time series datasets.

Superiority Across Diverse Benchmarks. NEST consistently achieves state-of-the-art (SOTA) performance, substantially outperforming all competitive baselines. As indicated in the "1^{st} Count" row of Table **??**, NEST secures the best MSE and MAE results in 32 out of 36 experimental settings across 9 distinct datasets. The performance gains are particularly pronounced on high-variability datasets; for instance, on the CESNET_1 benchmark, NEST surpasses the strong baseline PatchTST by 5.49% and iTransformer by 14.82% in MSE on average across four different prediction horizons. This consistent dominance across multi-year physical cycles and heterogeneous network traffic validates NEST's capability to effectively track and model the dynamic reorganization of inter-variable dependencies, which is a structural challenge that monolithic models and local non-stationarity mitigators fail to address.

Robustness on Heterogeneous Network Traffic. NEST's efficacy is most prominent on the five large-scale **CESNET** benchmarks. Characterized by 40 weeks of real-world ISP traffic, these datasets exhibit extreme variability and regime transitions. NEST's superiority here stems from its systematic handling of non-stationarity: the unsupervised regime discovery module utilizes moment-entropy metrics to categorize data slices into functionally distinct modes, such as high-volatility surges versus stable, low-complexity periods.

Model	NEST(Ours)		iTransformer		UniTS		PatchTST		DLinear		FEDformer		Autoformer		Informer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
CESNET_1	**0.294**	**0.375**	0.345	0.398	0.420	0.427	0.310	0.383	1.143	0.925	0.531	0.553	0.699	0.637	4.901	1.969
CESNET_2	**0.632**	**0.559**	0.676	0.583	0.729	0.607	0.654	0.573	0.749	0.688	0.842	0.695	0.966	0.765	3.119	1.496
CESNET_3	**0.334**	**0.338**	0.341	0.342	0.358	0.351	0.338	0.340	0.343	0.355	0.383	0.380	0.389	0.391	0.865	0.632
CESNET_4	**0.364**	**0.405**	0.406	0.425	0.446	0.439	0.382	0.416	0.771	0.725	0.452	0.487	0.635	0.601	2.222	1.312
CESNET_5	**0.682**	**0.466**	0.693	0.472	0.779	0.511	0.692	0.474	0.691	0.479	0.790	0.531	0.806	0.537	0.746	0.509
Weather	**0.224**	**0.262**	0.231	0.270	0.308	0.343	0.229	0.268	0.242	0.294	0.329	0.370	0.443	0.470	0.704	0.597
ETTh1	**0.515**	**0.515**	0.520	0.524	0.994	0.723	0.523	0.518	0.526	0.524	0.659	0.621	0.817	0.679	1.407	0.918
ETTh2	0.234	**0.334**	0.236	0.338	0.297	0.386	0.251	0.347	**0.229**	0.338	0.315	0.424	0.465	0.511	0.536	0.539
TEC	**0.691**	**0.574**	0.694	0.577	0.803	0.642	0.694	0.577	0.695	0.581	0.735	0.603	0.869	0.667	0.723	0.594
1^{st} **Count**	**32**	**32**	1	3	0	0	3	4	4	0	0	0	0	0	0	0

Table 1: Long term forecasting results of time series models on nine different datasets. The history sequence length is 512. Corresponding prediction lengths include {96, 192, 336, 720}. A lower MSE or MAE indicates a better prediction. Averaged results of four prediction lengths are reported here. 1^{st} Count represents the number of wins achieved by a model under all prediction lengths and datasets. The best result is **bolded** and the second best is underlined.

Dataset	CESNET_1		CESNET_2		CESNET_3		CESNET_4		CESNET_5		Weather		ETTh1		ETTh2		TEC	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NEST	**0.294**	**0.375**	**0.632**	**0.559**	**0.334**	**0.338**	0.364	0.405	**0.682**	**0.466**	**0.224**	**0.262**	**0.515**	**0.515**	**0.234**	0.334	**0.691**	**0.574**
w/o Router	0.315	0.376	0.656	0.572	0.336	0.340	0.400	0.417	0.727	0.492	0.244	0.281	0.699	0.609	0.242	0.343	0.698	0.585
w/o Kmeans	0.307	0.383	0.652	0.570	0.338	0.340	**0.363**	0.406	0.687	0.468	0.225	0.265	0.519	0.519	0.246	0.342	0.692	0.575
w/o R&K	0.330	0.386	0.664	0.578	0.338	0.341	0.380	**0.402**	0.766	0.506	0.268	0.307	0.630	0.579	0.252	0.353	0.700	0.586
Dist. Router	0.309	0.380	0.637	0.561	0.335	0.338	0.392	0.415	0.709	0.480	0.230	0.265	0.522	0.521	0.234	**0.334**	0.699	0.581

Table 2: Ablation study on NEST components across main experiment datasets. The history sequence length is 512. Corresponding prediction lengths include {96, 192, 336, 720}. A lower MSE or MAE indicates a better prediction. Averaged results of four prediction lengths are reported here. The best result is **bolded** and the second best is underlined.

5.3 Ablation Study on NEST Components

To quantify the individual contributions of the core components in NEST, we evaluate five variants: (1) the full **NEST** model; (2) *w/o Router*, which replaces the regime-oriented router with a simple parameter averaging of experts; (3) *w/o Kmeans*, which replaces the moment-entropy based clustering with naive sequential data partitioning; (4) *w/o R&K*, which removes both modules; and (5) *Dist. Router*, which utilizes only the distance-based modulation $\tilde{\omega}_m$ without the temporal initial weight score $\mathbf{w}_{\text{init}}$. The results in Table **??** demonstrate that the full NEST configuration achieves superior performance in nearly all metrics, validating the design of our regime-aware paradigm.

Efficacy of the Regime-Oriented Router: Removing the router(*w/o Router*) leads to a consistent performance drop, with MSE increasing markedly on datasets like ETTh1 (0.515 to 0.699). This underlines that simple expert averaging fails to capture the structural transitions between regimes. Furthermore, the performance gap between NEST and *Dist. Router* confirms the necessity of our routing strategy: while geometric distance provides a global anchor, the initial weight score w_{init} is crucial for identifying the temporal state of the system.

Necessity of Moment-Entropy Based Discovery: The *w/o Kmeans* variant, which lacks principled regime partitioning, underperforms the full model across diverse benchmarks. For instance, on ETTh2, the MSE increases from 0.234 to 0.246. This justifies our use of moment-entropy metrics as a sufficient mathematical basis to discover functionally distinct operational modes, as naive partitioning fails to ensure the structural specialization of experts.

Synergistic Effect and Robustness: The *w/o R&K* variant generally yields the poorest results, highlighting the synergy between regime discovery and adaptive routing. The overwhelming trend across datasets confirms that the integration of principled regime discovery and regime-oriented routing is vital for robust forecasting under structural distribution shifts.

5.4 Hyperparameter Sensitivity Analysis

We evaluate NEST's robustness against key hyperparameters: model dimension (d_{model}), attention heads (n_{heads}), and regime count (M). As shown in Figure 3, NEST maintains remarkable stability across varying architectural configurations. Specifically, performance remains consistent for $d_{model} \in \{64, 128\}$ and $n_{heads} \in \{4, 8, 16\}$, with slight

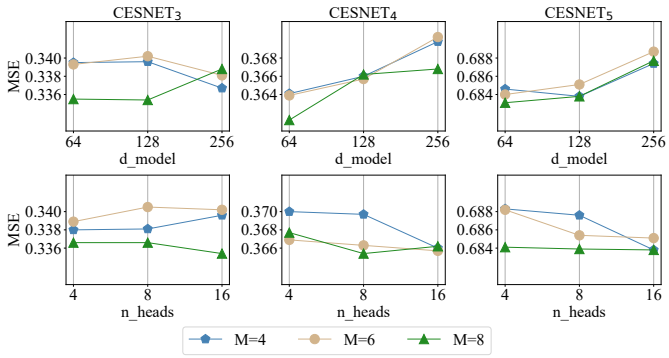

Figure 3: **Hyperparameter sensitivity analysis of NEST.** Parameter sensitivity analysis of NEST across multiple CESNET subsets. The plots illustrate the impact of hidden dimension size (d_{model}), the number of attention heads (n_{heads}), and the number of discovered regimes (M) on forecasting performance (MSE).

Model	Metric	96	192	336	720	Avg.
NEST	MSE	**0.149**	**0.192**	**0.243**	**0.311**	**0.224**
	MAE	**0.199**	**0.242**	**0.279**	**0.327**	**0.262**
CKA-Swap	MSE	0.156	0.223	0.250	0.322	0.237
	MAE	0.206	0.265	0.282	0.336	0.272

Table 3: Expert weight permutation analysis based on CKA similarity. Prediction lengths include {96, 192, 336, 720}.

MSE increases at $d_{model} = 256$. This insensitivity suggests that NEST's efficacy is derived from its regime-aware logic rather than exhaustive parameter tuning. In contrast, the regime count M (Data Slices) exhibits a more data-dependent impact. Empirical results across multiple benchmarks suggest that $M = 8$ serves as a robust and versatile configuration, providing a sufficient hypothesis space for experts to capture the diverse behavioral modes within the moment-entropy space. At the same time, we find that a smaller M (e.g., $M = 4$) can occasionally yield competitive results on datasets with relatively low regime complexity. This indicates that while NEST benefits from a diverse expert pool to handle high dataset-level distribution shifts, it maintains a degree of parameter efficiency, allowing for a pruned architecture when the underlying system transitions are less intricate.

5.5 Study on Regime Experts

To investigate the internal mechanism of NEST, we conduct a multi-level interpretability analysis to verify the structural specialization of the expert pool and the efficacy of the regime-oriented router.

Visualizing Variate-Attention Map. We first visualize the Query-Key (QK) attention matrices of different regime experts to observe their focus on inter-variable dependencies. The attention patterns across experts exhibit heterogeneity, especially in the high attention score QK-pair vicinity. Each expert evolves a distinct variate-attention map that adapts to the input sample, representing a specific coupling logic of variables. This polymorphism confirms that experts have effectively specialized in different system dynamics rather than converging to a redundant averaged representation.

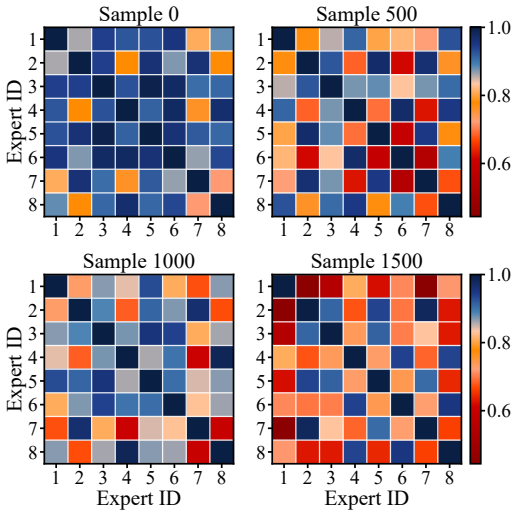

Figure 4: CKA similarity heatmap of specialized experts' QK matrix. Low pairwise similarity scores indicate lower similarity between the experts. Results are sampled from the Weather test set.

Quantifying Expert Dissimilarity via CKA. To further quantify the functional divergence between experts, we employ Centered Kernel Alignment (CKA) to measure the similarity of their attention matrices. As shown in Figure 4, the low pairwise CKA scores across the expert pool indicate significant representational dissimilarity. This high degree of heterogeneity proves that our moment-entropy based regime discovery successfully facilitates the emergence of non-redundant, specialized experts.

Expert Permutation and Router Sensitivity. To validate the necessity of precise expert orchestration, we conduct a weight permutation experiment. For each batch sample, experts with CKA similarity above a predefined threshold are eligible for pairing and routing score swapping, where pairs with higher similarity are prioritized for the swap operation. The results, summarized in Table **??**, show a significant performance degradation across all test samples. Even when two experts appear relatively similar, the router's precise allocation remains critical. This "collapse" upon permutation demonstrates that the efficacy of NEST stems from the exact match between the detected regime and the expert's specialized variate-attention kernel, rather than mere parameter ensemble.

6 Conclusion

In this paper, we addressed the challenge of dataset-level distribution shifts by introducing NEST, a two-phase MoE framework that models the alternation of underlying regimes. By combining moment-entropy based regime discovery, structurally specialized experts, and a regime-oriented router, NEST decomposes heterogeneous temporal dynamics into reusable forecasting kernels and adaptively recomposes them for each input. Extensive evaluations demonstrate that NEST achieves SOTA performance across diverse benchmarks, particularly on highly non-stationary long-term datasets.

Acknowledgments

This work was supported by the grants from the Natural Science Foundation of China (U2541212, 62572035), and Beijing Natural Science Foundation (L248032). Thanks for the computing infrastructure provided by Beijing Advanced Innovation Center for Big Data and Brain Computing. Haoyi Zhou is the corresponding author.

References

[Basu *et al.*, 1996] Sabyasachi Basu, Amarnath Mukherjee, and Steve Klivansky. Time series models for internet traffic. In *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*, volume 2, pages 611–620. IEEE, 1996.

[Boehme *et al.*, 2007] Thomas Boehme, A Robin Wallace, and Gareth P Harrison. Applying time series to power flow analysis in networks with high wind penetration. *IEEE transactions on power systems*, 22(3):951–957, 2007.

[Cleveland *et al.*, 1990] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

[Fan *et al.*, 2023] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529, 2023.

[Gao *et al.*, 2024] Shanghua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

[Koumar *et al.*, 2025] Josef Koumar, Karel Hynek, Tomáš Čejka, and Pavel Šiška. Cesnet-timeseries24: Time series dataset for network traffic anomaly detection and forecasting. *Scientific Data*, 12(1):338, 2025.

[Liu *et al.*, 2021] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[Liu *et al.*, 2023a] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[Liu *et al.*, 2023b] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36:14273–14292, 2023.

[Nie *et al.*, 2022] 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.

[Ogasawara *et al.*, 2010] Eduardo Ogasawara, Leonardo C Martinez, Daniel De Oliveira, Geraldo Zimbrão, Gisele L Pappa, and Marta Mattoso. Adaptive normalization: A novel data normalization approach for non-stationary time series. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[Passalis *et al.*, 2019] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

[Pulkkinen, 2007] Tuija Pulkkinen. Space weather: Terrestrial perspective. *Living Reviews in Solar Physics*, 4(1):1, 2007.

[Siami-Namini *et al.*, 2018] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. Ieee, 2018.

[Torrence and Compo, 1998] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

[Weerakody *et al.*, 2021] Philip B Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021.

[Wu *et al.*, 2020] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[Ye *et al.*, 2024] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37:31350–31379, 2024.

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

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of*

the AAAI conference on artificial intelligence, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] 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*, pages 27268–27286. PMLR, 2022.