

KAIROS: Unified Training for Universal Non-Autoregressive Time Series Forecasting

Kuiye Ding
dingkuiye@ict.ac.cn
Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China

Hongxiao Li, Yifan Wang
lihongxiao19@mails.ucas.ac.cn
wangyifan2014@ict.ac.cn
Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China

Fanda Fan[†]
fanfanda@ict.ac.cn
Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China

Lei Wang, Chunjie Luo
{wanglei, luochunjie}@ict.ac.cn
Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China

Zheyang Wang
zheyang.wang@durham.ac.uk
Department of Mathematical Sciences,
Durham University
Durham, UK

Jianfeng Zhan
zhanjianfeng@ict.ac.cn
Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China
University of Chinese Academy of
Sciences
Beijing, China

Abstract

In the World Wide Web, reliable time series forecasts provide the forward-looking signals that drive resource planning, cache placement, and anomaly response, enabling platforms to operate efficiently as user behavior and content distributions evolve. Compared with other domains, time series forecasting for Web applications requires much faster responsiveness to support real-time decision making. We present KAIROS, a non-autoregressive time series forecasting framework that directly models segment-level multi-peak distributions. Unlike autoregressive approaches, KAIROS avoids error accumulation and achieves just-in-time inference, while improving over existing non-autoregressive models that collapse to over-smoothed predictions. Trained on the large-scale corpus, KAIROS demonstrates strong zero-shot generalization on six widely used benchmarks, delivering forecasting performance comparable to state-of-the-art foundation models with similar scale, at a fraction of their inference cost. Beyond empirical results, KAIROS highlights the importance of non-autoregressive design as a scalable paradigm for foundation models in time series. Code is available at: <https://github.com/Day333/Kairos>.

CCS Concepts

• Mathematics of computing → Time series analysis.

Keywords

Time Series Forecasting; Multi-peak Distributions

1 Introduction

The World Wide Web functions as a large-scale, dynamic socio-technical infrastructure in which services must anticipate and respond to shifting content, demand, and user behavior [22]. Time series forecasting provides the forward-looking signals required to operate under such non-stationary conditions, leveraging historical telemetry to anticipate future patterns and trends [23, 24, 26, 32, 38,

42, 62]. Strong forecasting performance improves user experience and underpins intelligent web services, powering personalized recommendation and multi-peak representation learning [41, 57], capacity planning and predictive auto-scaling for microservices [19], web-scale behavioral and economic modeling [7], and event-driven financial forecasting [62]. These capabilities position time series forecasting as a foundational component for building adaptive, data-driven platforms across the World Wide Web [43]. Recently, time series foundation models have emerged [3, 8, 15, 36, 51, 58], inspired by the pre-training paradigm of large language models, and have demonstrated strong potential in zero-shot forecasting, cross-domain transfer, and long-range dependency modeling.

The current landscape of time series modeling exhibits a clear polarization. On one side, direct predictors built upon simple linear layers generate the entire forecast sequence [5, 40, 63] in a single forward pass. This *non-autoregressive* (NAR) strategy [16, 18] ensures extremely high inference efficiency, but its constrained representational power makes it inadequate for capturing the complex temporal dynamics of real-world data. On the other side, in pursuit of higher predictive performance, emerging *time series foundation models* (TSFMs) have largely reverted to the *autoregressive* (AR) generative paradigm [3, 36, 51]. While AR models demonstrate strong modeling power, they also re-expose three intrinsic weaknesses of this paradigm that have long been discussed in sequence generation: (i) *slow inference speed*, as inference time grows linearly with sequence length [28], which becomes a critical bottleneck in long-horizon scenarios requiring rapid responses; (ii) *exposure bias*, since models rely on teacher forcing with ground-truth labels during training but must depend on their own potentially erroneous predictions at inference, leading to a mismatch that harms generalization [16, 46]; and (iii) *error accumulation* [46], as small mistakes made early in the inference process can be amplified across subsequent steps, causing predictions to drift away from the ground truth sequence. We illustrate the key difference between AR and NAR forecasting in Figure 1, where AR decoders generate outputs

[†]Corresponding author.

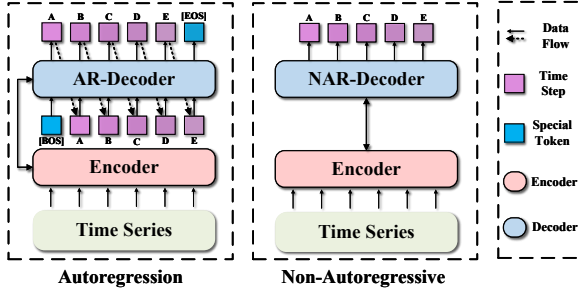


Figure 1: Autoregressive (AR) vs. non-autoregressive (NAR) forecasting. Left: the AR decoder generates each future token/segment sequentially, conditioning on previously produced outputs, which creates strict left-to-right dependencies and inference time that grows with the horizon. Right: the NAR decoder predicts all future segments in parallel from the encoder states in a single pass, removing the sequential dependency and enabling faster decoding.

sequentially while NAR decoders predict all future segments in parallel.

These shortcomings motivate us to revisit the NAR paradigm, which has been widely regarded as a promising path toward models that combine efficient inference with accurate forecasting, and has already achieved notable success in applications such as recommender systems [49] and machine translation [16, 18]. However, applying the NAR paradigm to time series forecasting requires addressing its most fundamental challenge, namely the *Multi-peak Distribution* of target data [16]. In time series, this phenomenon is often described as *Temporal Distribution Shift* [4, 11, 45], where similar historical inputs may lead to multiple, equally plausible futures. Unlike a global drift, such variability typically exhibits a *local* nature: certain forecast segments diverge significantly while others remain relatively close, as illustrated in Fig. 2. We therefore characterize this as a segment-level multi-peak distribution problem, and argue that the term multi-peak distribution is more appropriate. This segment-level diversity provides the key motivation for adopting *segment-wise forecasting* in our NAR design. When trained with point-estimation objectives such as MSE, conventional NAR models often suffer from *mode collapse* [36], producing averaged outcomes across possible futures and leading to forecasts that are excessively *over-smooth* [36] and of limited practical value.

To address the above challenges, we propose **KAIROS**, a high-performance non-autoregressive framework specifically designed to resolve the issue of multi-peak forecasting. Unlike conventional single-model approaches, KAIROS is a complete solution driven by three synergistic mechanisms: (i) **Scenario-Aware Generative Experts (SAGE)**: for future segments with multi-peak distributions, a single predictor tends to collapse different plausible outcomes into an averaged forecast, leading to mode collapse and over-smoothing. SAGE equips each segment with a mixture-of-experts (MoE) prediction head, where different experts specialize in distinct plausible scenarios, and a dynamic gating network routes each prediction to the most relevant expert combination. This design explicitly alleviates the multi-peak problem while maintaining parallel generation. (ii) **Learnable Exogenous Vectors**: the multi-peak nature

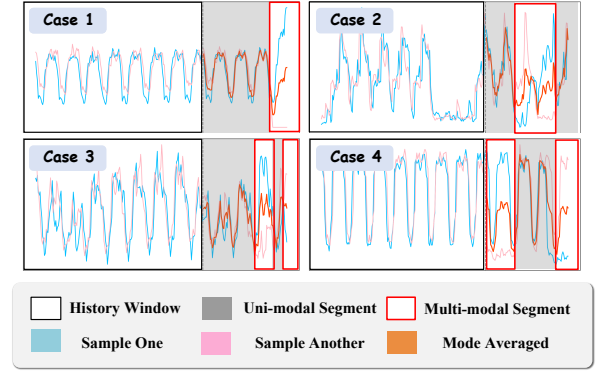


Figure 2: Illustration of four representative cases where the history windows exhibit high similarity, yet the corresponding prediction horizons differ across segments. Within the forecast, some regions (grey) are relatively uni-modal with consistent sequence, while others (red) are multi-peak with large divergence among plausible futures. In the multi-peak segments, models trained with point-estimation losses tend to produce mode-averaged predictions (orange), leading to over-smoothing or partial mode collapse. Importantly, this effect does not occur uniformly across the horizon but varies segment by segment, motivating our design of segment-wise forecasting to explicitly capture such local variability. These four cases are all from the ECL dataset [64].

of time series often arises from hidden external factors (e.g., environmental or contextual shifts) that are not directly observable in the input. KAIROS introduces a novel mechanism that combines statistical summaries of the input with a set of learnable *exogenous noise* vectors, effectively approximating these latent external variables to provide unique conditional information for each segment, thereby mitigating the multi-peak challenge. (iii) **Segment Causal Residual Noise (SCRN)**: to mitigate discontinuities across independently generated segments, KAIROS introduces lightweight learnable noise embeddings in a causal residual form. SCRn refines each segment’s prediction by injecting structured noise derived from its immediate predecessor, ensuring that every segment can only condition on past segments while remaining blind to the future. In this way, SCRn inherits the benefits of autoregressive dependency modeling with far lower computational cost, enabling forecasts that evolve smoothly and consistently across segments without reintroducing the inefficiencies of AR models.

Our contributions lie in four aspects:

- We propose KAIROS, a non-autoregressive framework for time series forecasting that explicitly addresses the multi-peak challenge through a mixture-of-experts design.
- We design learnable exogenous vectors to capture latent external factors and SCRn to causally link successive segments, improving temporal coherence without compromising NAR efficiency.

- Experimentally, KAIROS achieves state-of-the-art zero-shot performance on multiple forecasting benchmarks with substantially faster inference than autoregressive baselines, and shows clear advantages on long-horizon tasks.

2 Related Work

2.1 Time Series Foundation Models

Time series foundation models (TSFMs) have achieved remarkable progress in recent years. Most existing TSFMs adopt an AR paradigm [3, 8, 35, 36, 39, 48, 51], which provides strong modeling capacity and has led to significant performance gains. However, AR models also inherit the drawbacks of sequential decoding, including slow inference and exposure bias. A few NAR variants have been explored [15, 58], but these are primarily encoder-only architectures without a dedicated NAR decoder, and thus cannot fully exploit the advantages of parallel generation. In contrast, our model is designed with an explicit NAR decoding mechanism to address these limitations. Beyond modeling paradigms, current state-of-the-art TSFMs typically rely on either point-wise encoding [3, 39, 48, 51] or fixed-size patch encoding [8, 15, 35, 36, 58]. Point-wise representations often lead to prohibitive parameter counts, while fixed patches struggle to capture sufficient temporal variation. To address these limitations, we adopt the adaptive granularity patch proposed in [9], which enables variable-granularity representations better suited for capturing temporal dynamics in TSFMs.

Time-MoE is a decoder-only *autoregressive* time-series model that scales computation with sparse mixture-of-experts inside the backbone and generates forecasts step by step with point-wise tokenization [51]. *KAIROS* is *non-autoregressive*: it predicts future segments in parallel. Its MoE is placed in scenario-aware segment heads (SAGE) to separate alternative futures rather than to scale an AR backbone; it introduces learnable exogenous vectors to capture latent external drivers; and it employs Segment Causal Residual Noise (SCRN) to refine each segment using only preceding segments, improving temporal coherence without reverting to AR decoding. In short, Time-MoE targets scalable AR pretraining, whereas KAIROS targets segment-level multi-peak and cross-segment linkage within a parallel decoder.

Other related efforts that combine large language models (LLM) with time series forecasting have also shown promising results [10, 25, 30, 31, 34, 54], but their formulations differ substantially from ours. These approaches typically treat time series as natural language tokens and rely on LLM, whereas our work focuses on designing a dedicated non-autoregressive forecasting framework. Such approaches are therefore not considered TSFMs in this paper, and we do not provide a detailed discussion.

2.2 Non-Autoregressive Models

AR models remain the important choice for time series forecasting [2, 27, 29, 52], and recent TSFMs largely adopt the same formulation [3, 37, 51]. While effective, AR decoding introduces exposure bias and stepwise error accumulation [46], and its latency scales with the forecast horizon, treating short and long horizons uniformly despite different uncertainty and complexity.

NAR generation removes stepwise dependencies and enables parallel prediction. In neural machine translation, NAR substantially

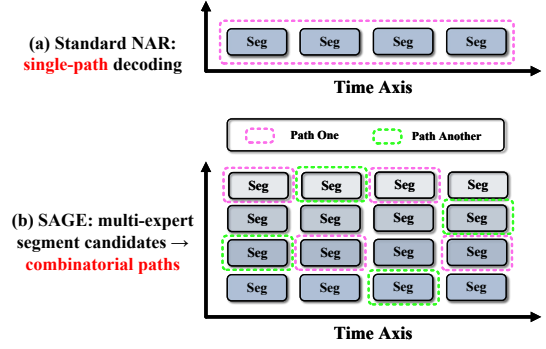


Figure 3: Illustration of the prediction space with and without Scenario-Aware Generative Experts (SAGE). (i) Standard non-autoregressive decoding outputs one deterministic segment per step, resulting in a single prediction path that often collapses to the mean and produces over-smoothed forecasts. (ii) SAGE trains multiple experts to generate diverse candidate segments for each future step. By composing alternative segments along the time axis, the model can explore multiple plausible prediction paths, mitigating mode collapse and improving the fidelity of segment-level forecasts.

reduces latency but faces multimodality and alignment (fertility) issues, latent-variable formulations, knowledge distillation, and iterative refinement [16]. Similar ideas have been explored beyond text, e.g., recommendation and reranking [49], where NAR achieves competitive accuracy with significant efficiency gains [18]. Most task-specific full-shot time series forecasting models [6, 21, 40, 44, 45, 53, 56, 63] adopt a non-autoregressive paradigm, typically relying on a single linear projection, whereas the non-autoregressive formulation of time series foundation models (TSFMs) remains largely underexplored. In time series, however, NAR remains underexplored: existing attempts are predominantly encoder-only or imputation-oriented and lack a dedicated generative NAR decoder, leaving segment-level multimodality and cross-segment dependencies insufficiently addressed. Our work situates a NAR decoder within a forecasting framework that explicitly targets these two gaps: handling multimodal futures at the segment level and preserving temporal coherence without reverting to AR decoding.

3 Problem Definition

Time series forecasting is the task of predicting future observations given a history window of past values. For a sequence $\mathbf{x}_{1:T}$, the model takes the most recent L points $\mathbf{x}_{T-L+1:T}$ as context and outputs predictions for the next H steps $\mathbf{x}_{T+1:T+H}$. In the foundation model setting, a TSFM with parameters θ is pretrained on large-scale corpora to learn a general conditional mapping $p_{\theta}(\mathbf{x}_{T+1:T+H} | \mathbf{x}_{T-L+1:T})$. Once trained, the same model can be directly applied in a zero-shot manner on unseen datasets or tasks, i.e., generating forecasts $\hat{\mathbf{x}}_{T+1:T+H} = f_{\theta}(\mathbf{x}_{T-L+1:T})$ without task-specific finetuning. Objective: learn a universal forecasting model that minimizes the expected loss between predictions and true futures across diverse domains and horizons.

Table 1: Key modeling differences between KAIROS and representative TSFMs.

Method	Kairos (Ours)	Sundial (2025)	Time-MoE (2024)	Timer-XL (2024)	Moirai (2024)	Moment (2024)	LLMTIME (2024)	Chronos (2024)	Lag-Llama (2023)	TimesFM (2023)
Non-autoregressive Decoding	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
Adaptive Multi-granularity Patch	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Modeling Multi-peak Distribution	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗

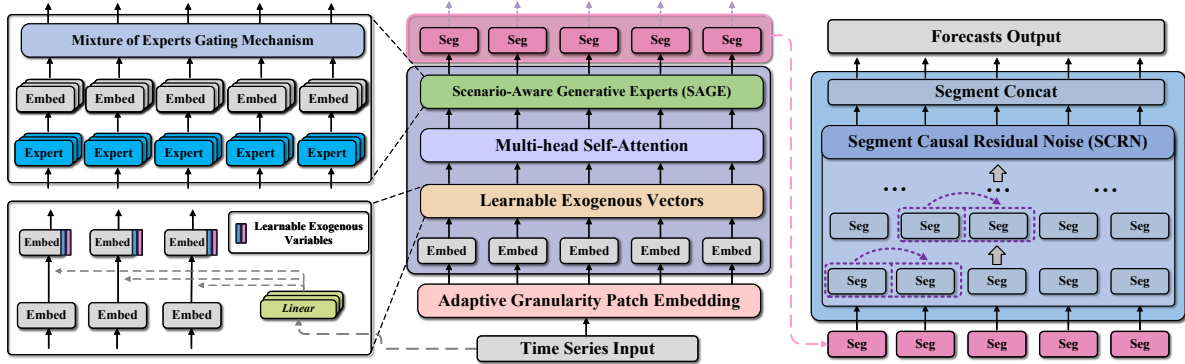


Figure 4: Overview of the proposed KAIROS framework. The model takes time series input and encodes it with adaptive granularity patch embeddings, augmented by learnable exogenous vectors. Scenario-Aware Generative Experts with a mixture-of-experts gating mechanism generate each future segment in parallel. Finally, the Segment Causal Residual FiLM refines segment outputs in a causal manner, linking past and future segments while preserving the efficiency of non-autoregressive decoding.

4 Methodology

We propose KAIROS, a non-autoregressive framework designed to address segment-level multi-peak in time series forecasting. As shown in Figure 4, KAIROS follows an encoder-decoder architecture with adaptive patch embeddings and learnable tokens to represent historical contexts, while segment-wise mixture-of-experts heads generate multiple future segments in parallel. A causal residual FiLM module then refines predictions across segments, combining the efficiency of non-autoregressive decoding with the temporal awareness of autoregressive models. In the following subsections, we detail each component of the framework, including the encoder design, the scenario-aware generative experts, the learnable exogenous vectors, and the causal refinement module.

4.1 Scenario-Aware Generative Experts

A central challenge of non-autoregressive forecasting is to handle multi-peak: for a given historical context, multiple plausible futures may exist. Direct training with pointwise losses often leads to mode averaging and over-smoothed predictions. To mitigate this issue, we introduce Scenario-Aware Generative Experts (SAGE), a segment-wise mixture-of-experts head designed to disentangle alternative futures and generate diverse segment predictions in parallel. As shown in Figure 3, standard non-autoregressive decoding produces a single path that often averages plausible futures, whereas SAGE supplies diverse segment candidates at each step, enabling combinatorial path composition and reducing over-smoothing.

Input representation. Let $\mathbf{H} \in \mathbb{R}^{B \times C \times D \times P}$ denote the encoder output for a batch of size B , with C channels, hidden dimension D , and P patch tokens. For the k -th forecast segment of length S , we flatten each (D, P) block into $\mathbf{z}_n \in \mathbb{R}^d$ where $d = DP$, resulting in $N = BC$ instances.

Routing mechanism. Each instance is routed to experts through

$$\mathbf{s}_n = \text{softmax}(\mathbf{W}_g \mathbf{z}_n) \in \mathbb{R}^E, \quad (1)$$

where $\mathbf{W}_g \in \mathbb{R}^{E \times d}$ and E is the number of experts. A sparse gate \mathbf{g}_n is formed by retaining the top- K entries of \mathbf{s}_n , which reduces computation and encourages specialization.

Expert predictions. Expert e produces a segment prediction by

$$\hat{\mathbf{y}}_{n,e} = \mathbf{W}^{(e)} \mathbf{z}_n + \mathbf{b}^{(e)} \in \mathbb{R}^S, \quad (2)$$

where $\mathbf{W}^{(e)} \in \mathbb{R}^{S \times d}$ and $\mathbf{b}^{(e)} \in \mathbb{R}^S$. The final output combines gated experts with a shared lightweight path:

$$\hat{\mathbf{y}}_n = \sum_{e=1}^E g_{n,e} \hat{\mathbf{y}}_{n,e} + \sigma(\mathbf{w}_s^\top \mathbf{z}_n) \mathbf{W}_s \mathbf{z}_n, \quad (3)$$

where $\mathbf{W}_s \in \mathbb{R}^{S \times d}$ and $\mathbf{w}_s \in \mathbb{R}^d$ parameterize the shared predictor, and σ is the sigmoid activation.

Load balancing. To prevent expert collapse, we introduce an auxiliary loss

$$\mathcal{L}_{\text{aux}} = \lambda E \sum_{e=1}^E r_e f_e, \quad (4)$$

where $r_e = \frac{1}{N} \sum_{n=1}^N s_{n,e}$ is the average routing probability and $f_e = \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{e \in \text{Top-}K(s_n)\}$ is the selection frequency. The coefficient λ controls the strength of balancing.

Segment loss. For the k -th segment, the training loss is

$$\mathcal{L}^{(k)} = \frac{1}{N} \sum_{n=1}^N \ell(\mathbf{y}_n^{(k)}, \hat{\mathbf{y}}_n), \quad (5)$$

where $\mathbf{y}_n^{(k)} \in \mathbb{R}^S$ is the ground truth segment and $\ell(\cdot)$ is a forecast loss. The overall objective for SAGE is $\mathcal{L}^{(k)} + \mathcal{L}_{\text{aux}}$.

By assigning alternative futures to different experts and regularizing expert usage, SAGE alleviates mode collapse and enables the model to generate sharper, more diverse segment-level predictions while retaining the parallelism of non-autoregressive decoding.

4.2 Learnable Exogenous Vectors

Non-autoregressive forecasting often suffers from multi-peak: similar historical contexts may correspond to different plausible futures. A key reason is the influence of hidden external factors that are not explicitly recorded in the data (e.g., market shocks, policy interventions, or user behavior changes). To capture this variability, we introduce Learnable Exogenous Vectors (LEV) as a parametric mechanism that injects controllable noise into the prediction process.

For each forecast segment $k \in \{1, \dots, K\}$, we associate a set of E learnable vectors

$$\mathbf{v}^{(k)} = \{\mathbf{v}_1^{(k)}, \dots, \mathbf{v}_E^{(k)}\}, \quad \mathbf{v}_e^{(k)} \in \mathbb{R}^D, \quad (6)$$

where D is the hidden dimension. These vectors are initialized randomly and updated during training. They play the role of *latent exogenous variables*, offering additional conditioning signals beyond the encoder representation.

Integration with encoder outputs. Given encoder output $\mathbf{h}^{(k)} \in \mathbb{R}^{N \times D}$ for segment k ($N = BC$ instances), the input to the decoder head is augmented as

$$\tilde{\mathbf{h}}^{(k)} = [\mathbf{h}^{(k)}; \mathbf{v}^{(k)}], \quad (7)$$

where $[\cdot; \cdot]$ denotes concatenation along the token dimension. This augmentation provides the mixture-of-experts head with stochastic context that can explain diverse futures.

By learning to align injected vectors with distributional variations in the data, LEV allows the model to represent latent external influences. This reduces the tendency toward mode averaging and equips the forecasting process with a flexible parametric handle for uncertainty, while keeping inference cost negligible.

4.3 Segment Causal Residual Noise

To maintain causal dependencies across adjacent segments while preserving the efficiency of non-autoregressive decoding, we propose *Segment Causal Residual Noise* (SCRN). Unlike FiLM-based modulation that explicitly parameterizes γ and β via convolution, SCRN introduces lightweight learnable embeddings to inject causal residuals in the form of structured noise.

Let $\mathbf{y}^{(s)} \in \mathbb{R}^{B \times C \times L}$ denote the prediction of the s -th segment, where B is the batch size, C the channel dimension, and L the segment length. For each segment index $s \in \{1, \dots, S\}$, we associate

a learnable embedding vector $\mathbf{e}^{(s)} \in \mathbb{R}^L$ initialized with Kaiming uniform and scaled by a small factor. When $s > 1$, the embedding is broadcast to all batches and channels, and combined with the previous segment prediction $\mathbf{y}^{(s-1)}$ to generate residual noise:

$$\mathbf{n}^{(s)} = \mathbf{y}^{(s-1)} \odot \mathbf{e}^{(s)}, \quad (8)$$

where \odot denotes element-wise multiplication. The current segment is then refined as

$$\tilde{\mathbf{y}}^{(s)} = \mathbf{y}^{(s)} + \alpha \mathbf{n}^{(s)}, \quad (9)$$

where α is a small learnable scalar controlling the noise strength. For the first segment $s = 1$, no adjustment is applied.

To prevent the embeddings from drifting toward large magnitudes, we regularize them with an ℓ_2 penalty:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathbf{E}|} \sum_{s=1}^S \|\mathbf{e}^{(s)}\|_2^2, \quad (10)$$

where \mathbf{E} is the set of all embeddings. This encourages embeddings to stay close to zero, thereby functioning as small perturbations rather than dominant signals.

SCRN can be interpreted as a form of *causal residual noise injection* [17]: each segment prediction is gently perturbed using information from its immediate predecessor, which stabilizes long-horizon forecasts and reduces abrupt inconsistencies across segments while incurring negligible computational cost.

4.4 Model Implementation

KAIROS integrates three major modules: (i) Scenario-Aware Generative Experts for segment-wise parallel generation, (ii) Learnable Exogenous Vectors for capturing latent external variability, and (iii) Segment Causal Residual Noise (SCRN) for causal refinement across segments. In addition, the encoder employs *adaptive patch embedding* to represent histories at variable granularities. All components are trained jointly in an end-to-end fashion.

Forecasting loss. Given ground truth segments $\{\mathbf{y}^{(k)}\}_{k=1}^K$ and predictions $\{\hat{\mathbf{y}}^{(k)}\}_{k=1}^K$ after SCRN refinement, the primary forecasting loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{K} \sum_{k=1}^K \ell(\mathbf{y}^{(k)}, \hat{\mathbf{y}}^{(k)}), \quad (11)$$

where $\ell(\cdot, \cdot)$ is a pointwise criterion that are mean squared error (MSE) or mean absolute error (MAE).

Expert balancing. SAGE introduces an auxiliary loss to avoid expert collapse and encourage balanced routing:

$$\mathcal{L}_{\text{aux}} = \lambda_{\text{aux}} E \sum_{e=1}^E r_e f_e, \quad (12)$$

where r_e and f_e denote the average routing probability and selection frequency of expert e , E is the number of experts, and λ_{aux} is a balancing weight.

Patch budget loss. To prevent adaptive patching from degenerating into a single granularity, we regularize the frequency of patch selections. Let p_m denote the empirical probability of choosing patch length m across training samples and $u_m = 1/M$ the uniform

target distribution over M candidate patch sizes. The budget loss is defined as

$$\mathcal{L}_{\text{budget}} = \lambda_{\text{budget}} \sum_{m=1}^M (p_m - u_m)^2, \quad (13)$$

where λ_{budget} controls the strength of regularization. This encourages the model to explore all granularities during training.

objective. The total loss is the weighted sum

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{budget}}. \quad (14)$$

LEV and SCRn are trained implicitly through \mathcal{L} without additional terms.

5 Experiments

This section investigates the following key research questions through extensive empirical studies:

- **RQ1:** How well does KAIROS generalize in zero-shot forecasting scenarios compared with state-of-the-art autoregressive and non-autoregressive baselines?
- **RQ2:** Does the non-autoregressive formulation of KAIROS lead to measurable inference speedup while preserving forecasting performance?
- **RQ3:** What is the impact of varying the design parameters of KAIROS, such as the number of experts in SAGE, the dimensionality of learnable exogenous vectors, and the capacity of the SCRn module?

5.1 Dataset and Experimental Settings

Datasets. Following TimeMoE [51], we evaluate our model on six widely used benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, and GlobalTemp. For pre-training, we leverage BLAST [50], a large-scale dataset designed for universal time series modeling. BLAST contains 321 billion observations drawn from diverse sources including CMP6 [13] (32.5%), ERA5 [20] (30.0%), WeatherBench [47] (25.7%), Buildings_900K [12] (6.9%), and over 300 additional public datasets [50] (4.9%), covering domains such as climate, meteorology, and energy. To avoid data leakage, benchmark datasets used for downstream evaluation (e.g., ETTh/ETTM, Weather, GlobalTemp) are explicitly excluded from BLAST during pre-training. Consequently, other datasets such as ECL, Traffic, and PEMS are not included in our evaluation since they appear in the pre-training corpora of several existing TSFMs, making fair comparison difficult. Moreover, although GIFT-Eval [1] provides a comprehensive evaluation protocol, we do not adopt it here because many of its benchmark datasets overlap with our pre-training corpus, which could compromise evaluation independence.

Baseline. We compare against pre-trained foundation time series models, including TimeMoE [51], MOIRAI [58], Chronos [3], TimesFM [8], and Moment [15], using the parameter settings as specified in their respective publications. For fairness, we omit TimesFM on the Weather dataset that was included in its pretraining corpus. Sundial and Timer-XL did not report benchmark results on the GlobalTemp dataset.

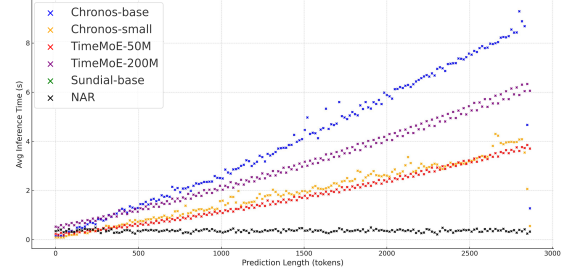


Figure 5: Comparison of inference times across different time-series foundation models. The x-axis denotes the prediction length (tokens), and the y-axis shows the average inference time (seconds). Results include Chronos-base, Chronos-small, TimeMoE-50M, TimeMoE-200M, and our proposed NAR model.

Evaluation Metrics. Following previous works, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics to assess the performance.

Implementation Details. For the ablation studies, we set the number of encoder layers to 1, the input length to 512, and the prediction length to 720. The adaptive patch granularity list is fixed as [8, 16, 32, 64]. We train for 1 epoch with 8 attention heads and a batch size of 32 on a single NVIDIA RTX 3090 GPU. For the experiments reported in Table 2, we set the batch size to 1024. The training is conducted on 8 NVIDIA V100-48G GPUs and requires approximately 24 hours to complete. All other hyperparameters follow the settings described in previous work [9, 33, 59]. The results of other baselines in Table 2 come from [36, 51].

5.2 Zero-shot across Benchmarks (RQ1)

We evaluate zero-shot generalization on six benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, Weather, GlobalTemp) and four horizons {96, 192, 336, 720}, using MSE and MAE without task-specific fine-tuning. The model is pre-trained on BLAST with evaluation datasets excluded to avoid leakage. Table 2 reports comparisons against representative autoregressive and non-autoregressive TSFMs.

Results show that KAIROS achieves performance broadly comparable to strong baselines across datasets and horizons. With 130M parameters, it is on par with SUNDIAL_{base} (128M) and TIMEMoE-50M (113M), while smaller than CHRONOS_{base} (205M) and TIMEMoE (453M). Our Kairos model demonstrates remarkable efficiency: it can be trained with only a **single NVIDIA RTX 3090 GPU**, or completed within one day on 8 NVIDIA V100-48G GPUs for the full training. In contrast, Sundial requires **32 A100 GPUs** and TimeMoE requires **128 A100 GPUs with 10 days** of training. This parity is consistent with our design goal: to demonstrate that a non-autoregressive framework can match the accuracy of established foundation models under strict zero-shot evaluation, while preparing the ground for efficiency gains discussed in RQ2. The competitiveness of KAIROS is therefore both expected and desirable, confirming that explicit modeling of segment-level multi-peak distributions can yield accuracy comparable to larger autoregressive models, without trading off inference efficiency.

Table 2: Zero-shot Results. Lower MAE and MSE values indicate superior performance. The symbols s , b , and l represent the small, base, and large versions, respectively. Models with top-3 performance are highlighted in bold.

Method	KAIROS (Ours)		Sundial _b [2025c]		TimeMoE _b [2025]		TimeMoE _l [2025]		Timer-XL [2025b]		MOIRAI _s [2024]		MOIRAI _b [2024]		MOIRAI _l [2024]		Chronos _s [2024]		Chronos _b [2024]		TimesFM [2024]		Moment [2024]	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.365 0.397	0.348 0.385	0.357 0.381	0.350 0.382	0.369 0.391	0.401 0.402	0.376 0.392	0.381 0.388	0.466 0.409	0.440 0.393	0.414 0.404	0.688 0.557											
	192	0.393 0.414	0.393 0.418	0.384 0.404	0.388 0.412	0.405 0.413	0.435 0.421	0.412 0.413	0.434 0.415	0.530 0.450	0.492 0.426	0.465 0.434	0.688 0.560											
	336	0.410 0.423	0.422 0.440	0.411 0.434	0.411 0.430	0.418 0.423	0.438 0.434	0.433 0.428	0.485 0.445	0.570 0.486	0.500 0.462	0.503 0.456	0.675 0.563											
	720	0.422 0.443	0.481 0.493	0.449 0.477	0.427 0.455	0.423 0.441	0.439 0.454	0.447 0.444	0.611 0.510	0.615 0.543	0.882 0.591	0.511 0.481	0.683 0.585											
	AVG	0.395 0.419	0.411 0.434	0.400 0.424	0.394 0.419	0.404 0.417	0.428 0.427	0.417 0.419	0.480 0.439	0.545 0.472	0.591 0.468	0.473 0.443	0.683 0.566											
ETTh2	96	0.284 0.342	0.271 0.333	0.305 0.359	0.302 0.354	0.283 0.342	0.297 0.336	0.294 0.330	0.296 0.330	0.307 0.356	0.308 0.343	0.315 0.349	0.342 0.396											
	192	0.351 0.384	0.327 0.376	0.351 0.386	0.364 0.385	0.340 0.379	0.368 0.381	0.365 0.375	0.361 0.371	0.376 0.401	0.384 0.392	0.388 0.395	0.354 0.402											
	336	0.368 0.405	0.354 0.402	0.391 0.418	0.417 0.425	0.366 0.400	0.370 0.393	0.376 0.390	0.390 0.390	0.408 0.431	0.429 0.430	0.422 0.427	0.356 0.407											
	720	0.382 0.423	0.381 0.435	0.419 0.454	0.537 0.496	0.397 0.431	0.411 0.426	0.416 0.433	0.423 0.418	0.604 0.533	0.501 0.477	0.443 0.454	0.395 0.434											
	AVG	0.346 0.389	0.333 0.387	0.366 0.404	0.405 0.415	0.347 0.388	0.361 0.384	0.362 0.382	0.367 0.377	0.424 0.430	0.405 0.410	0.392 0.406	0.361 0.409											
ETTm1	96	0.369 0.380	0.280 0.334	0.338 0.368	0.309 0.357	0.317 0.356	0.418 0.392	0.363 0.356	0.380 0.361	0.511 0.423	0.454 0.408	0.361 0.370	0.654 0.527											
	192	0.406 0.401	0.321 0.366	0.353 0.388	0.346 0.381	0.358 0.381	0.431 0.405	0.388 0.375	0.412 0.383	0.618 0.485	0.567 0.477	0.414 0.405	0.662 0.532											
	336	0.436 0.419	0.350 0.389	0.381 0.413	0.373 0.408	0.386 0.401	0.433 0.412	0.416 0.392	0.416 0.392	0.683 0.524	0.662 0.525	0.445 0.429	0.672 0.537											
	720	0.482 0.441	0.394 0.418	0.504 0.493	0.475 0.477	0.430 0.431	0.462 0.432	0.460 0.418	0.462 0.420	0.748 0.566	0.900 0.591	0.512 0.471	0.692 0.551											
	AVG	0.423 0.410	0.336 0.377	0.394 0.415	0.376 0.405	0.373 0.392	0.436 0.410	0.406 0.385	0.422 0.391	0.640 0.499	0.645 0.500	0.433 0.418	0.670 0.536											
ETTm2	96	0.186 0.272	0.170 0.256	0.201 0.291	0.197 0.286	0.198 0.288	0.214 0.288	0.205 0.273	0.211 0.274	0.209 0.291	0.199 0.274	0.197 0.271	0.260 0.335											
	192	0.247 0.310	0.229 0.300	0.258 0.334	0.250 0.322	0.241 0.315	0.284 0.332	0.275 0.316	0.281 0.318	0.280 0.341	0.261 0.322	0.289 0.321	0.289 0.350											
	336	0.305 0.346	0.281 0.337	0.324 0.373	0.337 0.375	0.286 0.348	0.331 0.362	0.329 0.350	0.341 0.355	0.354 0.390	0.326 0.366	0.360 0.366	0.324 0.369											
	720	0.396 0.400	0.351 0.387	0.488 0.464	0.480 0.461	0.375 0.402	0.402 0.408	0.437 0.411	0.485 0.428	0.553 0.499	0.455 0.439	0.462 0.430	0.394 0.409											
	AVG	0.284 0.298	0.258 0.320	0.317 0.365	0.316 0.361	0.273 0.336	0.307 0.347	0.311 0.337	0.329 0.343	0.349 0.380	0.310 0.350	0.328 0.346	0.316 0.365											
Weather	96	0.178 0.236	0.157 0.205	0.160 0.214	0.159 0.213	0.171 0.225	0.198 0.222	0.220 0.217	0.199 0.211	0.211 0.243	0.203 0.238	-	-	0.243 0.255										
	192	0.227 0.279	0.205 0.251	0.210 0.260	0.215 0.266	0.221 0.271	0.247 0.265	0.271 0.259	0.246 0.251	0.263 0.294	0.256 0.290	-	-	0.278 0.329										
	336	0.279 0.316	0.253 0.289	0.274 0.309	0.291 0.322	0.274 0.311	0.283 0.303	0.286 0.297	0.274 0.291	0.321 0.339	0.314 0.336	-	-	0.306 0.346										
	720	0.341 0.360	0.320 0.336	0.418 0.405	0.415 0.400	0.356 0.370	0.373 0.354	0.373 0.354	0.337 0.340	0.404 0.397	0.397 0.396	-	-	0.350 0.374										
	AVG	0.256 0.298	0.234 0.270	0.265 0.297	0.270 0.300	0.256 0.294	0.275 0.286	0.287 0.281	0.264 0.273	0.300 0.318	0.292 0.315	-	-	0.294 0.326										
Global	96	0.229 0.348	-	-	0.211 0.343	0.210 0.342	-	-	0.227 0.354	0.224 0.351	0.224 0.351	0.234 0.361	0.230 0.355	0.255 0.375	0.363 0.472									
	192	0.268 0.389	-	-	0.257 0.386	0.254 0.385	-	-	0.269 0.396	0.266 0.394	0.267 0.395	0.276 0.400	0.273 0.395	0.313 0.423	0.387 0.489									
	336	0.320 0.427	-	-	0.281 0.405	0.267 0.395	-	-	0.292 0.419	0.296 0.420	0.291 0.417	0.314 0.431	0.324 0.434	0.362 0.460	0.430 0.517									
	720	0.403 0.481	-	-	0.354 0.465	0.289 0.420	-	-	0.351 0.437	0.403 0.498	0.387 0.488	0.418 0.504	0.505 0.542	0.486 0.545	0.582 0.617									
	AVG	0.305 0.411	-	-	0.275 0.400	0.255 0.385	-	-	0.285 0.409	0.297 0.416	0.292 0.413	0.311 0.424	0.333 0.431	0.354 0.451	0.440 0.524									
Counts	36		38		23		28		40		9		19		20		0		1		1		4	

5.3 Inference Efficiency (RQ2)

A central motivation for adopting the NAR formulation is to remove the strict left-to-right dependency of AR decoding. As shown in Fig. 5, the inference time of KAIROS remains nearly constant across different prediction lengths, demonstrating its scalability advantage. In contrast, AR-based TSFMs such as Chronos and TimeMoE exhibit linearly increasing latency as the horizon grows, since each step requires conditioning on previously generated outputs. This stepwise generation becomes a critical bottleneck in long-horizon forecasting, limiting their applicability in real-time scenarios.

When compared with Sundial [36], a different behavior is observed. Sundial exhibits staircase-like transitions in inference time, with abrupt jumps at horizons that are multiples of 720 tokens. This arises from its multi-patch prediction design, where each block outputs a fixed length of 720, leading to block-wise parallel prediction for horizons shorter than 720. As a result, its runtime often resembles that of non-autoregressive models. Moreover, Sundial incorporates system-level optimizations such as FlashAttention, KV caching, and shared condition, which further improve efficiency. Consequently, Sundial is substantially faster than conventional

AR baselines and can achieve near NAR-like runtime for horizons within 720 tokens. However, due to its reliance on multi-patch generation rather than strict stepwise decoding, Sundial no longer fits the definition of a purely autoregressive model. For this reason, we do not include Sundial in the direct AR-vs-NAR runtime comparison in Figure 5.

In summary, the non-autoregressive formulation of KAIROS yields consistent and efficient inference across horizons, while conventional AR TSFMs suffer from increasing latency. Even with advanced optimizations, AR models such as Sundial only partially alleviate this limitation under specific settings, confirming the inherent advantage of NAR decoding for time series foundation models.

5.4 Effects of Key Design Parameters (RQ3)

Due to computational constraints, all ablation experiments in this subsection are conducted on a 1% subset of the BLAST corpus. This reduced setting still preserves sufficient diversity to examine the effect of different architectural choices, while allowing controlled comparisons of model variants.

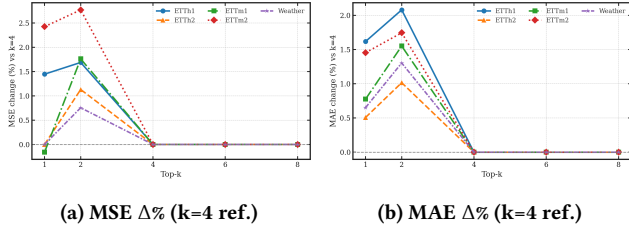


Figure 6: Relative performance change across datasets when varying Top- k (experts $E=8$). Values are percentage changes w.r.t. $k=4$; gray dashed line marks 0%.

Segment Cross-Attention Decoder. To further investigate causal refinement strategies, we implement *Segment Cross-Attention Decoder* (SCAD) as a comparison to SCRNN. Instead of injecting lightweight residual noise, SCAD employs a cross-attention mechanism to explicitly condition each forecast segment on its immediate predecessor. Concretely, the first segment is predicted directly from the hidden representation, while subsequent segments are generated by querying their predecessors through a multi-head attention module. Although this design captures richer dependencies across adjacent segments, it introduces additional computational cost, as each new segment requires a full attention pass. We therefore use SCAD primarily as a controlled baseline to assess whether the added complexity of attention-based refinement yields tangible benefits over the lightweight noise-based approach.

Ablation study of core modules. We conduct an ablation study to quantify the contribution of each proposed component, as summarized in Table 3. The *Baseline* corresponds to a simplified encoder-decoder with adaptive patch embeddings but without any of the proposed modules. Adding LEV does not consistently improve performance; in fact, in most cases it slightly degrades accuracy. A possible reason is that the injected exogenous vectors may introduce more stochastic noise than useful conditioning signals, thereby increasing the risk of overfitting rather than capturing genuine latent factors. This suggests that while exogenous variability is an important consideration, learning it implicitly is non-trivial and may require more sophisticated regularization. In contrast, SAGE proves to be the most effective component, yielding clear improvements across datasets (e.g., substantial gains on ETTh2). This validates our claim that segment-wise mixture-of-experts is crucial for handling multi-peak distributions and avoiding mode collapse, making it the main driver of KAIROS’s forecasting performance. Adding SCRNN on top of LEV and SAGE provides only marginal benefits and in some cases even offsets the gains. This may be due to the fact that SAGE already provides strong segment-level predictions, and SCRNN’s residual correction sometimes introduces additional noise rather than meaningful refinement. Although SCRNN is designed to reduce discontinuities, prior work has pointed out that temporal coherence can often emerge implicitly from self-attention representations without explicit local corrections [55]. In this sense, SCRNN plays more of a stabilizing role than a decisive factor for accuracy. Finally, SCAD, the cross-attention alternative, achieves competitive results on Weather but generally underperforms SCRNN, reflecting its higher computational cost and tendency to overfit

local dependencies. Overall, the ablation confirms that SAGE is the dominant contributor, while LEV and SCRNN provide auxiliary but less reliable gains. The full KAIROS model (LEV + SAGE + SCRNN) is adopted as the default configuration in subsequent experiments, though the analysis highlights the importance of segment-level expert modeling as the key source of improvement.

Hyperparameter Sensitivity. In Fig. 6, we ablate the gating sparsity by varying the Top- k (with $E=8$ total experts) on ETTh1, ETTh2, ETTh1, ETTh2 and Weather. Both MSE and MAE curves generally improve from $k=1$ to $k=4$, and then plateau or slightly degrade for $k \geq 6$. Averaging over horizons and datasets, Top- $k=4$ achieves the lowest error, confirming it as the optimal setting for our MoE configuration, see details in Appendix Table 8.

Limitations and outlook. The ablation results make clear that the strongest improvements of KAIROS originate from the segment-wise expert design, while the auxiliary modules show mixed effectiveness. The learnable exogenous vectors often degrade performance, which we attribute to the difficulty of distinguishing informative latent factors from stochastic perturbations. In practice, the vectors may introduce additional noise or encourage the model to overfit, especially in zero-shot scenarios where the alignment between pre-training and downstream distributions is fragile. The Segment Causal Residual Noise module also demonstrates limited benefit. Although its purpose is to reduce discontinuities across segments, its corrective signals can sometimes conflict with the already specialized predictions generated by the experts. In those cases, the adjustments may act more like disturbances than refinements. This phenomenon is consistent with prior observations in sequence modeling, where implicit correlations between consecutive predictions already provide a degree of temporal coherence without explicit correction.

We acknowledge that our current design is therefore not a fully effective solution. Nonetheless, these results do not undermine the central message of our work. On the contrary, they emphasize two essential lessons. First, non-autoregressive forecasting is crucial for building foundation models that scale to web-scale time series data while meeting the demands of real-time inference. Second, introducing localized causal refinement remains a promising idea, even though our present implementation yields only modest improvements. This direction is inspired by related progress in machine translation, where similar mechanisms have been shown to mitigate error accumulation. Our findings highlight that the challenge is not in the validity of the idea but in its execution: future work must focus on designing refinement strategies that reinforce rather than conflict with expert predictions. By recognizing these limitations explicitly, we aim to make clear where the most pressing challenges lie and to underline the importance of pursuing efficient non-autoregressive approaches as the foundation for the next generation of time series forecasting models.

6 Conclusion

In this work, We introduce KAIROS, a non-autoregressive time series foundation model that explicitly tackles the segment-level multi-peak nature of future distributions. By combining adaptive patch embeddings with segment-wise generative experts, KAIROS

Table 3: Ablation results averaged over horizons (96, 192, 336, 720). Each cell shows Avg MSE / Avg MAE. Best (lower) per dataset and overall average is bold. See details in Appendix Table 7

Method	ETTh1		ETTTh2		ETTh1		ETTm2		Weather		Average	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Baseline	0.415	0.435	0.352	0.392	0.632	0.514	0.293	0.347	0.266	0.310	0.392	0.400
Baseline + LEV	0.452	0.457	0.358	0.399	0.611	0.507	0.303	0.352	0.263	0.311	0.397	0.405
Baseline + LEV + SAGE	0.420	0.439	0.356	0.396	0.612	0.507	0.291	0.345	0.263	0.308	0.388	0.399
Baseline + LEV + SAGE + SCRN	0.426	0.441	0.361	0.399	0.634	0.514	0.297	0.351	0.267	0.314	0.397	0.404
Baseline + LEV + SAGE + SCAD	0.448	0.459	0.369	0.403	0.612	0.511	0.309	0.352	0.266	0.307	0.401	0.406

achieves diverse predictions and efficient parallel decoding. Pre-trained on BLAST and evaluated in zero-shot settings, KAIROS delivers performance comparable to leading autoregressive and non-autoregressive models, while offering significant inference efficiency gains. Although some auxiliary modules show mixed effectiveness, our study underscores the necessity of non-autoregressive design for scalable time series forecasting and highlights promising directions for improving causal refinement and exogenous variability modeling.

Acknowledgments

This work was supported by the Innovation Funding of Institute of Computing Technology, Chinese Academy of Sciences under Grant No. E461070.

References

- [1] Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. 2025. GIFT-Eval: A Benchmark for General Time Series Forecasting Model Evaluation.
- [2] Musleh Alharthi and Ausif Mahmood. 2024. xLSTMTIME: Long-term Time Series Forecasting With xLSTM. arXiv:2407.10240 [cs.LG]
- [3] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research (TMLR)* (2024).
- [4] Peter J Brockwell and Richard A Davis. 2009. *Time series: theory and methods*. Springer science & business media.
- [5] Hui Chen, Viet Luong, Lopamudra Mukherjee, and Vikas Singh. 2025. SimpleTM: A Simple Baseline for Multivariate Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.
- [6] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.
- [7] Ernesto Colacrai, Federico Cinus, Gianmarco De Francisci Morales, and Michele Starnini. 2024. Navigating Multidimensional Ideologies with Reddit’s Political Compass: Economic Conflict and Social Affinity. In *Proceedings of the ACM Web Conference*. 2582–2593.
- [8] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.
- [9] Kuiye Ding, Fanda Fan, Chunyi Hou, Zheyu Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. 2025. TimeMosaic: Temporal Heterogeneity Guided Time Series Forecasting via Adaptive Granularity Patch and Segment-wise Decoding.
- [10] Kuiye Ding, Fanda Fan, Yao Wang, Ruijie jian, Xiaorui Wang, Luqi Gong, Yishan Jiang, and Chunjie Luo an Jianfeng Zhan. 2025. DualSG: A Dual-Stream Explicit Semantic-Guided Multivariate Time Series Forecasting Framework. In *ACM International Conference on Multimedia (MM)*.
- [11] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. 2021. AdARNN: Adaptive Learning and Forecasting of Time Series. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 402–411.
- [12] Patrick Emami, Abhijeet Sahu, and Peter Graf. 2023. BuildingsBench: A Large-Scale Dataset of 900K Buildings and Benchmark for Short-Term Load Forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [13] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. 2016. Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development* 9, 5 (2016), 1937–1958.
- [14] Georg Goerg. 2013. Forecastable component analysis. *ICML* (2013).
- [15] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning (ICML)*.
- [16] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations*.
- [17] Jiatao Gu and Xu Tan. 2022. Non-Autoregressive Sequence Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. 21–27.
- [18] Shangdong Gui, Chenze Shao, Zhengrui Ma, Xishan Zhang, Yunji Chen, and Yang Feng. 2023. Non-autoregressive Machine Translation with Probabilistic Context-free Grammar. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- [19] Yunda Guo, Jiake Ge, Panfeng Guo, Yunpeng Chai, Tao Li, Mengnan Shi, Yang Tu, and Jian Ouyang. 2024. PASS: Predictive Auto-Scaling System for Large-scale Enterprise Web Applications. In *Proceedings of the ACM Web Conference*. Association for Computing Machinery, New York, NY, USA, 2747–2758.
- [20] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Corneli Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elias Hölm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. 2020. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society* 146, 730 (2020), 1999–2049.
- [21] Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. 2025. TimeFilter: Patch-Specific Spatial-Temporal Graph Filtration for Time Series Forecasting. In *International Conference on Machine Learning*.
- [22] Qihe Huang, Zhengyang Zhou, Kuo Yang, and Yang Wang. 2025. Exploiting Language Power for Time Series Forecasting with Exogenous Variables. In *THE WEB CONFERENCE 2025*.
- [23] Sheo Yon Jhin, Jaehoon Lee, Minju Jo, Seungji Kook, Jinsung Jeon, Jiyeon Hyeon, Jayoung Kim, and Noseong Park. 2022. EXIT: Extrapolation and Interpolation-based Neural Controlled Differential Equations for Time-series Classification and Forecasting. In *Proceedings of the ACM Web Conference*. 3102–3112.
- [24] Renhe Jiang, Zhaonan Wang, Yudong Tao, Chuang Yang, Xuan Song, Ryosuke Shibasaki, Shu-Ching Chen, and Mei-Ling Shyu. 2023. Learning Social Meta-knowledge for Nowcasting Human Mobility in Disaster. In *Proceedings of the ACM Web Conference*. 2655–2665.
- [25] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations (ICLR)*.
- [26] Harshavardhan Kamarthi, Linghai Kong, Alexander Rodriguez, Chao Zhang, and B Aditya Prakash. 2022. CAMul: Calibrated and Accurate Multi-view Time-Series Forecasting. In *Proceedings of the ACM Web Conference*. 3174–3185.

- [27] Rohaifa Khaldi, Abdellatif El Afia, Raddouane Chiheb, and Siham Tabik. 2023. What is the best RNN-cell structure to forecast each time series behavior? *Expert Systems with Applications* 215 (2023), 119140.
- [28] Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of Autoregressive Models and Their Alternatives. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 5147–5173.
- [29] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. 2023. Segrrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200* (2023).
- [30] Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2025. TimeCMA: Towards LLM-Empowered Multivariate Time Series Forecasting via Cross-Modality Alignment. In *AAAI* 18780–18788.
- [31] Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025. CALF: Aligning LLMs for Time Series Forecasting via Cross-modal Fine-Tuning. In *AAAI*. 18915–18923. <https://doi.org/10.1609/aaai.v39i18.34082>
- [32] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. 2024. UniTime: A Language-Empowered Unified Model for Cross-Domain Time Series Forecasting. In *Proceedings of the ACM Web Conference*. 4095–4106.
- [33] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.
- [34] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [35] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2025. Timer-XL: Long-Context Transformers for Unified Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.
- [36] Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2025. Sundial: A Family of Highly Capable Time Series Foundation Models. In *International Conference on Machine Learning*.
- [37] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. 2024. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *Forty-first International Conference on Machine Learning*.
- [38] Kentaro Miyake, Hiroyoshi Ito, Christos Faloutsos, Hirotomo Matsumoto, and Atsuyuki Morishima. 2024. NETEVOLVE: Social Network Forecasting using Multi-Agent Reinforcement Learning with Interpretable Features. In *Proceedings of the ACM Web Conference 2024*. 2542–2551.
- [39] Shikai Qiu, Nate Gruver, Marc Finzi, and Andrew Gordon Wilson. 2023. Large Language Models Are Zero Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*.
- [40] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*.
- [41] Wentao Ning, Reynold Cheng, Xiao Yan, Ben Kao, Nan Huo, Nur Al Hasan Haldar, and Bo Tang. 2024. Debiasing Recommendation with Personal Popularity. In *The Web Conference 2024*.
- [42] Kohei Obata, Koki Kawabata, Yasuko Matsubara, and Yasushi Sakurai. 2024. Dynamic Multi-Network Mining of Tensor Time Series. In *Proceedings of the ACM Web Conference 2024*. 4117–4127.
- [43] QingqingLong, Zheng Fang, Chen Fang, Chong Chen, pengfei wang, and Yuanchun Zhou. 2024. Unveiling Delay Effects in Traffic Forecasting: A Perspective from Spatial-Temporal Delay Differential Equations. In *The Web Conference 2024*.
- [44] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proc. VLDB Endow.* 17, 9 (2024), 2363–2377.
- [45] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. 2025. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*. 1185–1196.
- [46] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence Level Training with Recurrent Neural Networks.
- [47] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. 2020. WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems* 12, 11 (2020), e2020MS002203.
- [48] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Bilos, Hena Ghonia, Nadhir Hassen, Anderson Schneider, Sahil Garg, Alexandre Drouin, Nicolas Chapados, Yuriy Nevmyvaka, and Irina Rish. 2023. Lag-Llama: Towards Foundation Models for Time Series Forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.
- [49] Yuxin Ren, Qiya Yang, Yichun Wu, Wei Xu, Yalong Wang, and Zhiqiang Zhang. 2024. Non-autoregressive Generative Models for Reranking Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5625–5634.
- [50] Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. 2025. BLAST: Balanced Sampling Time Series Corpus for Universal Forecasting Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*.
- [51] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2025. Time-MoE: Billion-Scale Time Series Foundation Models with Mixture of Experts. In *International Conference on Learning Representations (ICLR)*.
- [52] Kamile Stankeviciute, Ahmed M Alaa, and Mihaela Van der Schaar. 2021. Conformal time-series forecasting. *Advances in neural information processing systems* 34 (2021), 6216–6228.
- [53] Artyom Stitsyuk and Jaesik Choi. 2025. xPatch: Dual-Stream Time Series Forecasting with Exponential Seasonal-Trend Decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [54] Xue Wang, Liang Sun, Rong Jin, Tian Zhou, Peisong Niu. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *NeurIPS*.
- [55] Hao Wang, Licheng Pan, Zhichao Chen, Degui Yang, Sen Zhang, Yifei Yang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. 2025. FreDF: Learning to Forecast in the Frequency Domain. In *ICLR*.
- [56] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Sheng-tong Ju, Zhixuan Chu, and Ming Jin. 2025. TimeMixer++: A General Time Series Pattern Machine for Universal Predictive Analysis. In *International Conference on Learning Representations (ICLR)*.
- [57] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 790–800.
- [58] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *Forty-first International Conference on Machine Learning (ICML)*.
- [59] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.
- [60] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems* 34 (2021), 22419–22430.
- [61] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. 2023. Interpretable Weather Forecasting for Worldwide Stations with a Unified Deep Model. *Nature Machine Intelligence* (2023).
- [62] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. REST: Relational Event-driven Stock Trend Forecasting. In *Proceedings of the Web Conference 2021*. 1–10.
- [63] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting?. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [64] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, Vol. 35*. AAAI Press, 11106–11115.

A Dataset

Benchmark Details. We evaluate the performance of various models for long-term forecasting across eight well-established datasets, including the Weather [60], Global Temp [61], and ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) [64]. A detailed description of each dataset is provided in Table 6. Forecastability is calculated by one minus the entropy of Fourier decomposition of time series [14]. A larger value indicates better predictability.

B Sensitivity Analysis

Exogenous vector size. Table 5 reports the effect of varying the dimensionality d_{exo} of the learnable exogenous vectors. Across all datasets, smaller dimensions ($d_{\text{exo}}=1$ or 2) lead to the most stable performance, while further enlarging the vector size generally results in degraded accuracy. This trend suggests that the injected vectors introduce more noise than informative conditioning when their capacity is too large, thereby weakening generalization. In practice, a compact design is sufficient to capture latent variability, whereas higher-dimensional exogenous vectors risk amplifying stochastic fluctuations and overfitting to training data.

Segment length. Table 4 reports the sensitivity of KAIROS to different segment lengths across five benchmark datasets. Overall, the results indicate that performance remains stable when varying the segment size from 8 to 48, with only marginal fluctuations in MSE and MAE. On ETTh1 and ETTh2, shorter segments (8 or 16) achieve slightly better accuracy, suggesting that finer-grained decomposition can better capture local dynamics. In contrast, on high-frequency datasets such as ETTm1 and ETTm2, longer segments exhibit comparable or even improved performance, likely due to their ability to aggregate short-term noise. For the Weather dataset, the choice of segment length shows minimal impact, highlighting the robustness of the proposed framework to this hyperparameter. These findings confirm that KAIROS is not overly sensitive to segment size, maintaining stable forecasting accuracy across diverse temporal resolutions.

Table 4: Sensitivity analysis on segment length. Results are reported in MSE/MAE.

Dataset	Segment=8	Segment=16	Segment=48
ETTh1	0.3874/0.4158	0.3893/0.4131	0.3899/0.4169
	0.4101/0.4298	0.4135/0.4280	0.4099/0.4298
	0.4189/0.4388	0.4205/0.4361	0.4178/0.4379
	0.4370/0.4633	0.4378/0.4612	0.4398/0.4638
ETTh2	0.2795/0.3446	0.2889/0.3460	0.2855/0.3461
	0.3412/0.3820	0.3513/0.3854	0.3496/0.3859
	0.3757/0.4075	0.3783/0.4088	0.3803/0.4109
	0.3887/0.4286	0.3909/0.4292	0.3934/0.4309
ETTh1	0.6043/0.4951	0.5998/0.4910	0.6123/0.5048
	0.6238/0.5068	0.6217/0.5074	0.6293/0.5163
	0.6455/0.5182	0.6407/0.5203	0.6457/0.5253
	0.6720/0.5335	0.6642/0.5356	0.6680/0.5400
ETTh2	0.2045/0.2966	0.2067/0.2949	0.2047/0.2949
	0.2587/0.3284	0.2590/0.3265	0.2591/0.3273
	0.3154/0.3620	0.3094/0.3571	0.3126/0.3591
	0.4061/0.4168	0.3974/0.4106	0.4022/0.4137
Weather	0.1907/0.2573	0.1932/0.2573	0.1865/0.2494
	0.2350/0.2927	0.2395/0.2923	0.2343/0.2871
	0.2827/0.3261	0.2849/0.3231	0.2820/0.3205
	0.3530/0.3730	0.3468/0.3655	0.3464/0.3636

Table 5: Ablation study on the exogenous noise vector size (d_{exo}). Results are reported in MSE/MAE.

Dataset	exo_noise=1	exo_noise=2	exo_noise=4	exo_noise=6	exo_noise=8
ETTh1	0.4087/0.4258	0.4100/0.4261	0.4286/0.4343	0.4747/0.4654	0.5022/0.4806
	0.4275/0.4380	0.4279/0.4368	0.4492/0.4489	0.4928/0.4790	0.5182/0.4916
	0.4348/0.4477	0.4323/0.4442	0.4579/0.4604	0.5000/0.4897	0.5241/0.5010
	0.4494/0.4738	0.4464/0.4705	0.4704/0.4843	0.5219/0.5210	0.5355/0.5267
ETTh2	0.2890/0.3507	0.2997/0.3567	0.2946/0.3542	0.3055/0.3618	0.3052/0.3661
	0.3498/0.3875	0.3603/0.3925	0.3545/0.3903	0.3656/0.3983	0.3560/0.3965
	0.3795/0.4124	0.3865/0.4140	0.3878/0.4170	0.3918/0.4207	0.3789/0.4157
	0.3897/0.4300	0.3926/0.4306	0.3940/0.4329	0.3984/0.4369	0.3996/0.4380
ETTh1	0.5835/0.4966	0.5760/0.4889	0.5750/0.4877	0.5437/0.4843	0.5951/0.5076
	0.6090/0.5091	0.6016/0.5009	0.5986/0.5001	0.5675/0.4969	0.6171/0.5173
	0.6266/0.5194	0.6240/0.5123	0.6214/0.5131	0.5864/0.5064	0.6354/0.5263
	0.6523/0.5373	0.6505/0.5296	0.6500/0.5290	0.6126/0.5227	0.6604/0.5390
ETTh2	0.2057/0.2946	0.2068/0.2946	0.2087/0.2960	0.2102/0.2955	0.2126/0.3009
	0.2631/0.3283	0.2604/0.3264	0.2650/0.3286	0.2673/0.3279	0.2634/0.3309
	0.3205/0.3621	0.3138/0.3591	0.3240/0.3639	0.3223/0.3606	0.3140/0.3614
	0.4104/0.4163	0.4032/0.4145	0.4126/0.4183	0.4092/0.4144	0.3997/0.4137
Weather	0.1915/0.2547	0.1971/0.2615	0.1961/0.2644	0.1971/0.2603	0.2013/0.2644
	0.2385/0.2915	0.2417/0.2958	0.2364/0.2941	0.2390/0.2910	0.2435/0.2973
	0.2859/0.3234	0.2854/0.3259	0.2801/0.3228	0.2824/0.3204	0.2851/0.3259
	0.3439/0.3611	0.3489/0.3676	0.3414/0.3636	0.3428/0.3599	0.3480/0.3688

C Complexity Analysis

Let K be the prediction horizon, S the number of forecast segments, L the segment length, and d the hidden dimension.

Autoregressive decoding. AR models generate one step at a time, yielding a total complexity of

$$O_{\text{AR}} = O(K \cdot d^2), \quad (15)$$

which scales linearly with K and limits parallelism.

Non-autoregressive decoding. KAIROS predicts $S=K/L$ segments in parallel, each via a mixture-of-experts head:

$$O_{\text{NAR}} = O(S \cdot d^2). \quad (16)$$

Thus the cost depends on S rather than K , and all segments can be generated in a single forward pass.

Parallelism. This design makes inference nearly independent of horizon length, enabling substantial speedup over AR decoding while retaining segment-level modeling flexibility.

D Formalization of Multi-Peak Distribution

In forecasting, the conditional distribution of future Y given history X is often *multi-peak*:

$$p(Y | X) = \sum_{m=1}^M \pi_m(X) p_m(Y | X), \quad (17)$$

where different modes p_m correspond to distinct but plausible futures.

Conventional NAR models trained with pointwise losses approximate the expectation

$$\hat{Y} \approx \mathbb{E}_{p(Y|X)}[Y], \quad (18)$$

leading to over-smoothed predictions. This motivates segment-wise expert modeling to preserve diverse futures.

Table 6: Detailed dataset descriptions. Dataset sizes are listed as (Train, Validation, Test).

Tasks	Dataset	Dim	Series Length	Dataset Size	Frequency	Forecastability*	Information
Long-term Forecasting	ETTh1	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15min	0.46	Temperature
	ETTh2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15min	0.55	Temperature
	ETTh1	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Hourly	0.38	Temperature
	ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Hourly	0.45	Temperature
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10min	0.75	Weather
	Global Temp	1000	{96, 192, 336, 720}	(12280, 1755, 3509)	Hourly	0.78	Temperature

Table 7: Ablation study across five datasets. Each cell shows MSE / MAE. Best (lower) per dataset and horizon is bold.

Dataset	Method	96	192	336	720
ETTh1	Baseline	0.389 / 0.413	0.414 / 0.428	0.420 / 0.436	0.438 / 0.461
	Baseline + LEV	0.429 / 0.434	0.449 / 0.449	0.458 / 0.460	0.470 / 0.484
	Baseline + LEV + SAGE	0.395 / 0.416	0.419 / 0.432	0.427 / 0.442	0.440 / 0.466
	Baseline + LEV + SAGE + SCRNN	0.403 / 0.419	0.423 / 0.432	0.432 / 0.443	0.445 / 0.471
	Baseline + LEV + SAGE + SCAD	0.422 / 0.438	0.445 / 0.452	0.452 / 0.461	0.472 / 0.487
ETTh2	Baseline	0.289 / 0.346	0.351 / 0.385	0.378 / 0.409	0.391 / 0.429
	Baseline + LEV	0.295 / 0.354	0.355 / 0.390	0.388 / 0.417	0.394 / 0.433
	Baseline + LEV + SAGE	0.292 / 0.349	0.354 / 0.388	0.385 / 0.414	0.395 / 0.432
	Baseline + LEV + SAGE + SCRNN	0.297 / 0.355	0.360 / 0.391	0.390 / 0.416	0.397 / 0.434
	Baseline + LEV + SAGE + SCAD	0.302 / 0.355	0.372 / 0.397	0.404 / 0.424	0.399 / 0.435
ETTh1	Baseline	0.600 / 0.491	0.622 / 0.507	0.641 / 0.520	0.664 / 0.536
	Baseline + LEV	0.575 / 0.488	0.599 / 0.500	0.621 / 0.513	0.650 / 0.529
	Baseline + LEV + SAGE	0.578 / 0.485	0.602 / 0.501	0.621 / 0.513	0.648 / 0.530
	Baseline + LEV + SAGE + SCRNN	0.597 / 0.491	0.621 / 0.506	0.647 / 0.520	0.671 / 0.537
	Baseline + LEV + SAGE + SCAD	0.576 / 0.492	0.603 / 0.506	0.624 / 0.516	0.645 / 0.529
ETTh2	Baseline	0.207 / 0.295	0.259 / 0.327	0.309 / 0.357	0.397 / 0.411
	Baseline + LEV	0.209 / 0.296	0.265 / 0.329	0.324 / 0.364	0.413 / 0.418
	Baseline + LEV + SAGE	0.203 / 0.291	0.256 / 0.324	0.309 / 0.356	0.397 / 0.410
	Baseline + LEV + SAGE + SCRNN	0.206 / 0.297	0.260 / 0.328	0.315 / 0.362	0.407 / 0.417
	Baseline + LEV + SAGE + SCAD	0.210 / 0.294	0.272 / 0.329	0.331 / 0.364	0.422 / 0.420
Weather	Baseline	0.193 / 0.257	0.239 / 0.292	0.285 / 0.323	0.347 / 0.366
	Baseline + LEV	0.196 / 0.264	0.236 / 0.294	0.280 / 0.323	0.341 / 0.364
	Baseline + LEV + SAGE	0.189 / 0.255	0.234 / 0.290	0.280 / 0.322	0.346 / 0.366
	Baseline + LEV + SAGE + SCRNN	0.196 / 0.264	0.239 / 0.297	0.284 / 0.328	0.347 / 0.369
	Baseline + LEV + SAGE + SCAD	0.192 / 0.252	0.238 / 0.288	0.284 / 0.320	0.351 / 0.366

Table 8: MoE Top- k (total experts $E=8$). Numbers are averaged over horizons (96/192/336/720). Each cell shows Avg MSE / Avg MAE. Best (lower) per dataset and overall is bold.

Top- k	ETTh1		ETTh2		ETTh1		ETTh2		Weather		Overall	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Top- $k=1$	0.421	0.440	0.354	0.396	0.624	0.520	0.296	0.349	0.265	0.309	0.392	0.403
Top- $k=2$	0.422	0.442	0.358	0.398	0.636	0.524	0.297	0.350	0.267	0.311	0.396	0.405
Top- $k=4$	0.415	0.433	0.354	0.394	0.625	0.516	0.289	0.344	0.265	0.307	0.390	0.399
Top- $k=6$	0.415	0.433	0.354	0.394	0.625	0.516	0.289	0.344	0.265	0.307	0.390	0.399
Top- $k=8$	0.415	0.433	0.354	0.394	0.625	0.516	0.289	0.344	0.265	0.307	0.390	0.399