BEYOND STATIC POSITIONAL ENCODING: BAYESIAN TIME-AWARE ATTENTION FOR NON-STATIONARY LONG-TERM TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers have demonstrated strong performance in long-term time series forecasting (LTSF), yet modeling non-stationary temporal dependencies remains challenging. In this paper, we interpret self-attention as a Bayesian posterior over attended positions, combining a content-based prior induced by scaled dot-product similarity with a temporal-evidence likelihood induced by positional encoding. This formulation provides a probabilistic interpretation of how positional encoding shapes the temporal dependencies captured by self-attention. Building on this perspective, we derive a population-level evidence lower bound (ELBO) that characterizes the role of positional encoding in forecasting weakly stationary time series. Our analysis further highlights a limitation of static positional encoding: when temporal dependence structures vary across input windows, a fixed temporal bias must accommodate heterogeneous dynamics and can therefore be suboptimal for non-stationary forecasting. Motivated by this analysis, we propose Bayesian Time-aware Attention (BTA), which employs a mixture-of-experts (MoE) router to adaptively combine temporal kernels and generate input-conditioned, head-specific positional biases. Extensive experiments on standard LTSF benchmarks demonstrate that BTA consistently improves forecasting performance across multiple Transformer backbones and achieves competitive overall performance with modest computational overhead and stable training.

1 INTRODUCTION

Long-term time series forecasting (LTSF) aims to predict future trajectories over extended horizons from historical observations (Chen et al., 2023). Transformers have emerged as a prominent architecture for LTSF largely because self-attention (Vaswani et al., 2017) enables the modeling of long-range temporal dependencies (Wen et al., 2022). However, real-world time series often exhibit non-stationarity (Kim et al., 2021), including evolving trends (Liu et al., 2025a), time-varying seasonality (Sun et al., 2026), and distribution shifts (Li et al., 2026). Such changes can alter which historical observations are informative for future prediction, making it challenging for Transformer-based models to capture temporal dependencies that vary across historical input windows.

Existing approaches to non-stationary forecasting broadly follow two strategies: normalization and architectural adaptation. First, normalization methods, such as RevIN (Kim et al., 2021), standardize historical input windows to mitigate shifts in marginal statistics and restore the original scale and location to the predictions. However, local statistical normalization may be insufficient to address distribution shifts (Li et al., 2026); even after normalization, windows with similar marginal statistics can exhibit different temporal dependence structures (Liu et al., 2026). Second, architectural approaches introduce specialized mechanisms to capture evolving temporal patterns, as exemplified by Non-stationary Transformer (Liu et al., 2022), TimeBridge (Liu et al., 2025a), and DTAF (Lu et al., 2026). While these approaches improve temporal modeling, architecture-specific designs can introduce additional complexity and limit their applicability across different Transformer backbones. These limitations motivate a broadly applicable mechanism that enables Transformers to adapt to changing temporal dependencies without extensive architectural changes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Beyond these methodological advances, a fundamental question remains: how does a Transformer forecaster model changing temporal dependencies? Without positional information, canonical self-attention is permutation-equivariant (Zeng et al., 2023) and therefore cannot distinguish temporal positions from content alone. Positional encoding (PE) addresses this limitation by introducing information about token positions or relative distances (Shaw et al., 2018), enabling self-attention to account for temporal order. This motivates a more specific question: *how does PE interact with self-attention to support forecasting when temporal dependencies vary across historical input windows?* Answering this question can clarify the temporal modeling behavior of Transformers under non-stationarity and provide a principled foundation for designing adaptive positional mechanisms.

In this paper, we address this question from a Bayesian perspective. We first interpret position-aware self-attention as a Bayesian posterior over attended positions: content-only scaled dot-product attention defines a prior over historical keys, while PE provides temporal-evidence likelihood that updates this prior according to temporal relationships. Building on this interpretation, we derive a population-level evidence lower bound (ELBO) (Bishop, 2006) that characterizes the effect of PE on forecasting weakly-stationary time series. To extend the analysis to non-stationarity, we introduce a regime-conditioned weakly-stationary formulation and show that a static PE, which induces shared temporal evidence across input windows, can be strictly suboptimal when heterogeneous regimes induce incompatible regime-specific optima. This theoretical result motivates positional mechanisms that adapt their temporal evidence to each historical input window.

Motivated by our theoretical analysis, we propose *Bayesian Time-aware Attention* (BTA), a plug-and-play module that learns input-conditioned, head-specific temporal biases over relative lags. BTA aggregates attention scores by lag, augments them with Fourier features, and uses a mixture-of-experts (MoE) (Cai et al., 2025) router to combine periodic and non-periodic temporal kernels. This enables different heads and input windows to capture distinct local, periodic, and long-range dependencies. Extensive experiments show that BTA generalizes across conventional Transformers (Zhou et al., 2021; Nie et al., 2023) and time-series foundation models (Kottapalli et al., 2025), achieving competitive performance on non-stationary LTSF with high computational efficiency.

Our main contributions are summarized as follows:

- **Theory.** We interpret position-aware self-attention from a Bayesian perspective and derive a population-level ELBO that characterizes the role of PE in time-series forecasting, proving that static PE can be suboptimal under changing temporal dependencies.
- **Methodology.** We propose *Bayesian Time-aware Attention* (BTA), a plug-and-play module that learns input-conditioned, head-specific temporal biases through an MoE router combining complementary temporal kernels.
- **Experiments.** Extensive experiments across conventional Transformers and time-series foundation models demonstrate that BTA consistently improves non-stationary LTSF while maintaining computational efficiency.

2 Related Work

Non-stationary TSF methods can be broadly organized into two directions: representation-level stationarization and model-level adaptation. Representation-level methods reduce distributional variation through statistical normalization (Kim et al., 2021; Fan et al., 2023; Liu et al., 2023), frequency-domain processing (Ye et al., 2024; Piao et al., 2024), or learned transformations (Fan et al., 2025; Liu et al., 2026; Li et al., 2026). Model-level methods instead modify the forecasting mechanism to capture evolving distributions, changing dynamics, and heterogeneous temporal patterns (Du et al., 2021; Duan et al., 2023; Liu & Lehrmann, 2024; Wang et al., 2023; Pham et al., 2023; Zhang et al., 2023; Qiu et al., 2025; Sun et al., 2025). In Transformer-based forecasting, these methods adapt attention to distribution shifts while preserving non-stationary information that is predictive of future behavior. The Non-stationary Transformer framework injects learned de-stationary factors into attention (Liu et al., 2022), whereas GBT decomposes forecasting into two Transformer stages to reduce discrepancies between historical and future distributions (Shen et al., 2023). Distribution-routing methods such as DUET and TFPS assign heterogeneous temporal patterns to specialized extractors or experts (Qiu et al., 2025; Sun et al., 2025). Attention-oriented architectures such as TimeBridge and TimeStacker separately model stable, cointegrated, or multi-

level temporal dependencies (Liu et al., 2025a;b), while DTAF combines temporal expert filtering with frequency differencing (Lu et al., 2026). BTA follows this model-level direction but serves as a plug-and-play attention module with an explicit probabilistic interpretation, allowing it to enhance existing Transformers without redesigning their forecasting backbones.

Positional encoding is commonly introduced to represent the order and temporal distance between tokens because self-attention is permutation-equivariant (Vaswani et al., 2017). Representative relative positional encoding methods in NLP include RPE (Shaw et al., 2018), Transformer-XL (Dai et al., 2019), T5 relative bias (Raffel et al., 2020), ALiBi (Press et al., 2021), KERPLE (Chi et al., 2022), and RoPE (Su et al., 2024). In time-series modeling, eRPE and STTRE introduce relative embeddings tailored to temporal and spatio-temporal dependencies (Foumani et al., 2024; Deihim et al., 2023), whereas Powerformer and Dozerformer impose power-law locality and structured local-periodic biases, respectively (Hegazy et al., 2025; Zhang et al., 2024). These methods generally use predefined or globally learned positional structures shared across samples. Although PAREformer constructs adaptive positional representations by integrating positional encodings with seasonal–trend features (Tong & Yuan, 2025), BTA differs by directly constructing an input-conditioned relative attention bias as a mixture of probabilistic temporal kernels. Specifically, a mixture-of-experts router dynamically combines these kernels for each input series, enabling the positional bias to adapt to heterogeneous periodicities and temporal scales.

3 THEORY

3.1 PRELIMINARIES AND NOTATIONS

Weakly stationary time series (Gaussian process). Let $(\Omega, \mathcal{F}, P)$ be a probability space and $\{X_t\}_{t\in\mathbb{Z}}$ be a zero-mean weakly stationary Gaussian process (GP), $X(t) \sim \mathcal{GP}(0, k)$, with $\mathbb{E}[X_t] = 0$ and stationary covariance kernel $k(t, t') = \Gamma(t - t')$, where $\Gamma(\tau)$ denotes the autocovariance at lag $\tau = t - t'$. For forecasting at time n, define the historical and future random vectors as $\boldsymbol{X}^- = [X_{n-T+1}, \ldots, X_n]^\top \in \mathbb{R}^T$ and $\boldsymbol{Y} = [X_{n+1}, \ldots, X_{n+L}]^\top \in \mathbb{R}^L$. Let $\boldsymbol{\Sigma}_- = \mathrm{Cov}(\boldsymbol{X}^-)$, $\boldsymbol{\Sigma}_{+-} = \mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{X}^-)$, and $\boldsymbol{\Sigma}_+ = \mathrm{Cov}(\boldsymbol{Y})$, where $\boldsymbol{\Sigma}_-$ is assumed to be positive definite. By stationarity, these covariance matrices are fully determined by $\Gamma(\tau)$.

Self-attention. Based on the kernel interpretation (Tsai et al., 2019) of scaled dot-product self-attention, the attention weight assigned by the ith query $\boldsymbol{q}_i \in \mathbb{R}^{d_k}$ to the jth key position is

$$\alpha_{ij} = \mathrm{Softmax}_{j\in\mathcal{C}(i)} \left(\boldsymbol{q}_i^\top \boldsymbol{k}_j / \sqrt{d_k} + b_{ij} \right), \tag{1}$$

where $i \in \{1, \ldots, T\}$, $\boldsymbol{k}_j \in \mathbb{R}^{d_k}$ denotes the jth key, and b_{ij} denotes the PE-induced positional score associated with the query–key pair (i, j). Depending on the PE mechanism, b_{ij} may encode temporal order, relative lag, or other predefined positional structure. For a fixed query position i, let $\boldsymbol{b}_i = (b_{ij})_{j\in\mathcal{C}(i)}$, where $\mathcal{C}(i) = \{1, \ldots, i-1\}$ denotes the set of accessible key positions under strict causal attention; otherwise, $\mathcal{C}(i) = \{1, \ldots, T\}$. Without PE, i.e., $b_{ij} = 0$, self-attention is permutation-equivariant with respect to token order and therefore cannot distinguish temporal positions from content alone (Tsai et al., 2019; Kim et al., 2024; Irani & Metsis, 2025).

Simplified Transformer. Consider a point-wise Transformer consisting of a single self-attention encoder followed by average pooling over the T historical positions and a linear forecasting decoder. Its output can be written as

$$f_{\theta,\phi}(\boldsymbol{x}^-) = \boldsymbol{w}\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-) \cdot \boldsymbol{x}^- = \boldsymbol{M}_{\theta,\phi}(\boldsymbol{x}^-) \cdot \boldsymbol{x}^- \in \mathbb{R}^L, \tag{2}$$

where $\boldsymbol{x}^- \in \mathbb{R}^T$ denotes the historical input, $\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-) \in \mathbb{R}^{1\times T}$ denotes the averaged attention weights induced by PE method ϕ, and $\boldsymbol{M}_{\theta,\phi}(\boldsymbol{x}^-) := \boldsymbol{w}\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-) \in \mathbb{R}^{L\times T}$ denotes the effective forecasting operator, with $\boldsymbol{w} \in \mathbb{R}^{L\times 1}$ denoting the effective linear decoder that absorbs the embedding and value projections. Although simplified, this architecture retains the core components of a Transformer—content-based self-attention, positional encoding, representation aggregation, and linear prediction—while remaining sufficiently tractable for theoretical analysis. A detailed specification of the simplified Transformer is provided in Appendix A.

3.2 BAYESIAN EXPLANATIONS FOR TRANSFORMER IN NON-STATIONARY LTSF

Proposition 1 (PE provides latent temporal evidence)**.** *For a fixed query position i, let $S_i \in \mathcal{C}(i)$ be a categorical random variable representing the selected key position. For each candidate position $j \in \mathcal{C}(i)$, let Z_{ij} denote a latent random variable representing the temporal evidence associated with the query–key pair (i, j), and let z_{ij} denote a realization of Z_{ij}. Let $\boldsymbol{Z}_i = \{Z_{ij}\}_{j\in\mathcal{C}(i)}$ and $\boldsymbol{z}_i = \{z_{ij}\}_{j\in\mathcal{C}(i)}$. Assuming that the candidate-specific contribution to $p(\boldsymbol{Z}_i = \boldsymbol{z}_i \mid S_i = j)$ is characterized by $p(Z_{ij} = z_{ij} \mid S_i = j)$ up to a multiplicative factor independent of j, the attention distribution in Equation 1 admits the Bayesian interpretation $\alpha_{ij} = p(S_i = j \mid \boldsymbol{Z}_i = \boldsymbol{z}_i)$,*

$$p(S_i = j \mid \boldsymbol{Z}_i = \boldsymbol{z}_i) \propto \underbrace{p(S_i = j)}_{\textit{content-based prior}} \cdot \underbrace{p(Z_{ij} = z_{ij} \mid S_i = j)}_{\textit{temporal-evidence likelihood}}, \tag{3}$$

where the content-based prior $p(S_i = j) = \mathrm{Softmax}_{j\in\mathcal{C}(i)}\left(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{d_k}\right)$ is induced by scaled dot-product self-attention without PE, and PE can be interpreted as specifying a candidate-specific temporal-evidence likelihood such that $b_{ij} \overset{c}{=} \log p(Z_{ij} = z_{ij} \mid S_i = j)$, where $\overset{c}{=}$ denotes equality up to an additive term independent of j. The proof is in Appendix B.

Proposition 1 provides a probabilistic interpretation of representative PE mechanisms. In particular, different PE mechanisms can be explained as different probabilistic assumptions of the temporal-evidence likelihood. Detailed probabilistic interpretations are provided in Appendix C. Throughout the following analysis, we call a PE *static* if its induced positional score b_{ij}^{ϕ} is independent of the historical realization $\boldsymbol{x}^-$; this definition refers to the induced attention bias rather than whether the PE parameters themselves are fixed. Consequently, $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-) = q_\phi(\boldsymbol{Z})$.

Theorem 2 (Population-level ELBO of PE on weakly-stationary time series)**.** *Let $\mathcal{D} = p(\boldsymbol{X}^-, \boldsymbol{Y})$ be the weakly-stationary Gaussian distribution in Section 3.1. For fixed model parameters θ, under the frozen-attention Gaussian forecasting model $p_{\theta,\phi}(\boldsymbol{Y} \mid \boldsymbol{X}^-, \boldsymbol{Z} = \boldsymbol{z}) = \mathcal{N}(\boldsymbol{Y}; \boldsymbol{M}_{\phi,\boldsymbol{z}}\boldsymbol{X}^-, \sigma_\epsilon^2 \boldsymbol{I}_L)$, let $p(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ and $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ denote the GP-derived temporal-evidence prior and PE-induced variational posterior, respectively. Then $\mathbb{E}_{\mathcal{D}}[\log p_{\theta,\phi}(\boldsymbol{Y} \mid \boldsymbol{X}^-)] \geq \mathcal{L}_{\mathrm{pop}}(\phi) := \mathbb{E}_{\mathcal{D}}\mathbb{E}_{q_\phi(\boldsymbol{Z}\mid\boldsymbol{X}^-)}[\log p_{\theta,\phi}(\boldsymbol{Y} \mid \boldsymbol{X}^-, \boldsymbol{Z})] - \mathbb{E}_{\boldsymbol{X}^-} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{X}^-)\right)$. For a static PE, $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-) = q_\phi(\boldsymbol{Z})$, and the population-level ELBO reduces to*

$$\begin{aligned}\mathcal{L}_{\mathrm{pop}}(\phi) = C_0 + \frac{1}{\sigma_\epsilon^2}\mathbb{E}_{q_\phi}\,\mathrm{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{Z}}\boldsymbol{\Sigma}_{+-}^\top\right) - \frac{1}{2\sigma_\epsilon^2}\mathbb{E}_{q_\phi}\,\mathrm{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{Z}}\boldsymbol{\Sigma}_-\boldsymbol{M}_{\phi,\boldsymbol{Z}}^\top\right) \\ - \mathbb{E}_{\boldsymbol{X}^-} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{Z}) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{X}^-)\right),\end{aligned} \tag{4}$$

where $C_0 = -\,\mathrm{tr}(\boldsymbol{\Sigma}_+)/(2\sigma_\epsilon^2) - L\log(2\pi\sigma_\epsilon^2)/2$ is independent of ϕ. The general input-dependent covariance form and detailed proof are provided in Appendix D.

General non-stationary time series are difficult to characterize theoretically because their temporal dependence structures may vary over time. To enable tractable analysis, we model non-stationarity through a regime-conditioned weakly-stationary process, where each forecasting window is weakly stationary conditional on its regime. This allows Theorem 2 to be applied within each regime, providing a tractable characterization of the effect of PE under non-stationarity.

Corollary 3 (Potential suboptimality of static PE under non-stationarity)**.** *Let $R \in \{1, \ldots, J\}$ denote the regime of a forecasting window, with $\pi_r = p(R = r) > 0$ and $\sum_{r=1}^J \pi_r = 1$. Conditional on $R = r$, assume $\mathcal{D}_r = p(\boldsymbol{X}^-, \boldsymbol{Y} \mid R = r)$ is a weakly-stationary Gaussian distribution with regime-specific autocovariance function $\Gamma_r(\tau)$ and corresponding temporal-evidence prior $p_r(\boldsymbol{Z} \mid \boldsymbol{X}^-)$. By Theorem 2, the population-level ELBO of PE method ϕ under the regime-conditioned process is $\mathcal{L}_{\mathrm{reg}}(\phi) = \sum_{r=1}^J \pi_r \mathcal{L}_r(\phi)$, where $\mathcal{L}_r(\phi)$ denotes the population ELBO under regime r. For a static PE, the PE-induced posterior $q_\phi(\boldsymbol{Z})$ is shared across regimes, whereas the regime-specific prior $p_r(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ and covariance structure may vary with $\Gamma_r(\tau)$. Let Φ denote the admissible PE class and $\Phi_{\mathrm{static}} \subseteq \Phi$ its static subclass. Therefore,*

$$\sup_{\phi\in\Phi_{\mathrm{static}}} \mathcal{L}_{\mathrm{reg}}(\phi) \leq \sum_{r=1}^J \pi_r \sup_{\phi_r\in\Phi} \mathcal{L}_r(\phi_r). \tag{5}$$

The detailed derivation is provided in Appendix E.

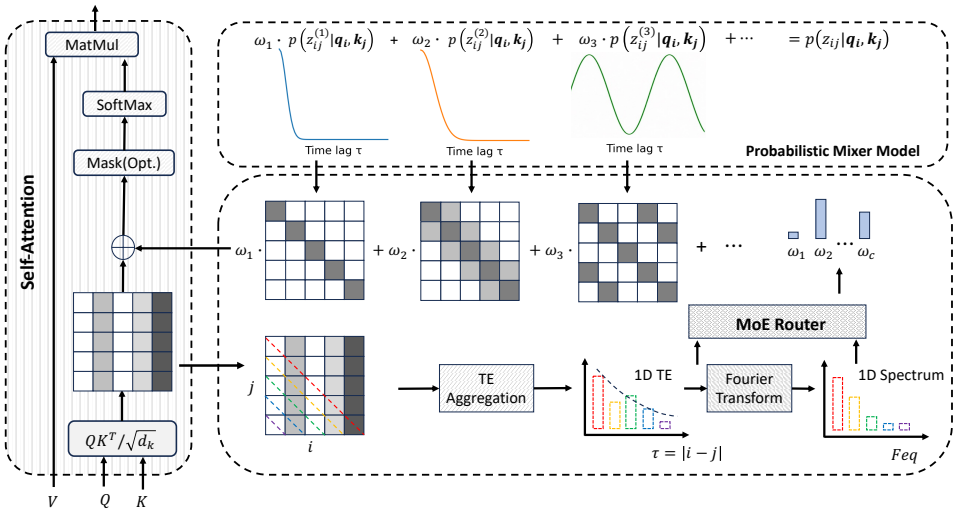


Figure 1: BTA consists of two components: a probabilistic mixer and a MoE router. The probabilistic mixer defines an over-complete set of candidate temporal patterns using time-aware kernels $\psi^{(c)}\left(\tau_{ij}^{(c)};\boldsymbol{\theta}^{(c)}\right)$. The MoE router learns the kernel weights $\{\omega_c\}_{c=1}^{C}$ with an MLP that processes the 1D temporal-evidence (TE) profile, obtained by aggregating the original 2D attention scores over relative lags, together with its Fourier spectrum. This plug-and-play design can be seamlessly integrated into standard multi-head self-attention.

The right-hand side of Equation 5 represents a regime-specific oracle that may select a different PE for each regime. Thus, Corollary 3 identifies a potential limitation of static PE: when heterogeneous regimes induce incompatible regime-specific optima, a single shared positional mechanism can be strictly suboptimal relative to regime-specific solutions. We illustrate Corollary 3 with a binary-regime forecasting example in Section F, where windows alternate between smooth and periodic GP dependence and adaptive PE achieves a higher estimated ELBO than static PEs. It motivates input-adaptive PE mechanisms that can vary their temporal evidence across forecasting windows.

4 Methodology

Motivated by Corollary 3, we seek an adaptive PE mechanism that adjusts its temporal evidence to the dependence structure of each input window. Since different windows may exhibit distinct temporal patterns, a mixture-of-experts (MoE) formulation provides a natural mechanism for representing multiple candidate structures and adaptively weighting them without requiring explicit regime labels. Based on this idea, we propose *Bayesian Time-aware Attention* (BTA), consisting of two components: (A) a probabilistic mixture of time-aware kernels that provides an over-complete set of candidate temporal patterns, and (B) an MoE router that produces input- and head-specific weights for combining these kernels. The resulting temporal-evidence likelihood is incorporated into the attention logits as an adaptive positional bias, as illustrated in Fig. 1.

4.1 Probabilistic Mixture of Temporal Kernels

For notational simplicity, we omit the input and attention-head indices (n, h) throughout this subsection. Following Proposition 1, we model the temporal-evidence likelihood as a mixture of M component likelihoods:

$$p_\phi(Z_{ij} = z_{ij} \mid S_i = j) = \sum_{c=1}^{M} \omega_c \, p_c(Z_{ij} = z_{ij} \mid S_i = j), \tag{6}$$

where $\omega_c \geq 0$, $\sum_{c=1}^{M} \omega_c = 1$, and each component represents a distinct temporal-dependence pattern, such as locality, periodicity, or long-range dependence. The mixture weights are subsequently generated by an MoE router for each input and attention head.

We parameterize the cth component through a time-aware log-kernel $\psi^{(c)}(\tau_{ij};\boldsymbol{\theta}^{(c)})$ such that $\log p_c(Z_{ij} = z_{ij} \mid S_i = j) \overset{c}{=} \psi^{(c)}\left(\tau_{ij};\boldsymbol{\theta}^{(c)}\right)$, where $\tau_{ij} = |i-j|$ and $\overset{c}{=}$ denotes equality up to an additive normalization term independent of j. Equivalently, $p_c(Z_{ij} = z_{ij} \mid S_i = j) \propto \exp[\psi^{(c)}(\tau_{ij};\boldsymbol{\theta}^{(c)})]$. Any component-specific normalization constant can be absorbed into the mixture weights without affecting the attention softmax. Therefore, the PE-induced temporal bias is $b_{ij} \overset{c}{=} \log\left[\sum_{c=1}^{M} \omega_c \exp\big(\psi^{(c)}\left(\tau_{ij};\boldsymbol{\theta}^{(c)}\right)\big)\right]$. Different choices of $\psi^{(c)}$ encode complementary assumptions about how temporal relevance varies with relative lag. The resulting kernel bank provides an over-complete set of candidate temporal patterns rather than imposing a single positional structure. The MoE router then determines their input- and head-specific contributions, allowing BTA to exploit useful temporal priors while adapting the positional evidence across forecasting windows.

4.2 MoE Router

To adapt the temporal-kernel mixture to each input, we design an MoE router that extracts lag-dependent structure from the content-based attention scores, as illustrated in Fig. 1. For the hth attention head of the nth input, the content-based attention score is $s_{ij}^{(n,h)} = \left[(\boldsymbol{q}_i^{(n,h)})^\top \boldsymbol{k}_j^{(n,h)}\right]/\sqrt{d_k}$. We aggregate scores associated with the same relative lag to construct a one-dimensional lag-wise routing profile: $e_\tau^{(n,h)} = \frac{1}{|\mathcal{I}_\tau|}\sum_{(i,j)\in\mathcal{I}_\tau} s_{ij}^{(n,h)}, \mathcal{I}_\tau = \{(i,j) : |i-j| = \tau\}$. This aggregation summarizes how strongly the content representations associate query–key pairs at different temporal lags. To capture both lag-domain structure and periodicity, we concatenate the routing profile with the magnitude of its Fourier spectrum and feed them into an MLP, namely $\boldsymbol{\omega}^{(n,h)} = \mathrm{Softmax}\left[\mathrm{MLP}\left(\left[\boldsymbol{e}^{(n,h)}; \left|\mathcal{F}(\boldsymbol{e}^{(n,h)})\right|\right]\right)\right]$, where $\mathcal{F}(\cdot)$ denotes the discrete Fourier transform and $\boldsymbol{\omega}^{(n,h)} = [\omega_1^{(n,h)}, \ldots, \omega_M^{(n,h)}]$ denote the input- and head-specific mixture weights. Finally, BTA constructs the adaptive positional bias from the temporal-kernel mixture in Equation 6:

$$b_{ij}^{(n,h)} \overset{c}{=} \log\left[\sum_{c=1}^{M} \omega_c^{(n,h)} \exp\left(\psi^{(c)}\left(\tau_{ij};\boldsymbol{\theta}^{(c)}\right)\right)\right]. \tag{7}$$

The resulting bias is added to the content-based attention logits before the softmax, allowing different inputs and attention heads to emphasize distinct local, periodic, or long-range temporal dependencies. Because the candidate kernels and their hyperparameters are fixed, only the router learns how to combine the candidate temporal patterns. This yields a lightweight, plug-and-play mechanism that converts static PE kernels into input- and head-adaptive positional biases and can be incorporated into standard multi-head self-attention without modifying the forecasting backbone.

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate on eight widely used multivariate time-series forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, and Solar. They span diverse domains and temporal resolutions, with 7–862 variables and Augmented Dickey–Fuller (ADF) statistics ranging from -37.23 to -4.13 (Elliott et al., 1992), providing a broad testbed with varying stationarity and temporal patterns. Detailed descriptions are provided in Appendix G.

Transformer backbones and Non-stationary LTSF references. We select four representative Transformer architectures – Informer (Zhou et al., 2021), PatchTST (Nie et al., 2023), TimeBridge (Liu et al., 2025a), and Timer-XL (Liu et al., 2025c) – as backbones and integrate BTA into each model. These backbones are chosen to cover complementary stages and design paradigms of Transformer-based LTSF. This diversity allows us to evaluate whether BTA generalizes across substantially different Transformer architectures rather than benefiting a particular backbone. In addition, we compare our method to ten LTSF baselines: PULSE (Liu et al., 2026), DTAF (Lu et al., 2026), DUET (Qiu et al., 2025), TimeBridge (Liu et al., 2025a), iTransformer (Liu et al., 2024), TimeMixer (Wang et al., 2024), PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023), DLinear (Zeng et al., 2023), and Nonstationary Transformer (Liu et al., 2022). Detailed descriptions of backbone and reference methods are provided in Appendix H and Appendix I, respectively.

Implementation Details. We use the official implementations of all the aforementioned models. The implementation details of BTA are provided in Appendix J. We evaluate LTSF performance using mean squared error (MSE) and mean absolute error (MAE), and report FLOPs and the number of learnable parameters to quantify model complexity. All experiments are implemented in PyTorch and executed on an NVIDIA A100 GPU with 80 GB of memory.

5.2 Main Results

We integrate BTA into three backbones. Table 1 shows that BTA consistently improves all three backbones across the four datasets with different ADFs. Averaged across datasets, BTA reduces MSE by 9.70%, 10.35%, and 5.70% for Informer, PatchTST, and Timer-XL, respectively, with corresponding MAE reductions of 5.99%, 6.84%, and 2.68%. The improvements across heterogeneous architectures and datasets demonstrate that BTA provides a broadly effective plug-and-play enhancement to Transformer-based models for non-stationary LTSF. Additional three-seed experiments in Table 11 show consistently lower mean MSE with limited variation across random initializations.

Table 1: **BTA generalizes across three Transformer backbones**. Sequence lengths and backbone-specific settings in Appendix H. Each dataset entry is averaged over $H \in \{96, 192, 336, 720\}$. Gain(%) denotes the MSE and MAE decrease from Vanilla attention to BTA. Full results in Table 10.

Model		**Informer** (Zhou et al., 2021)						**PatchTST** (Nie et al., 2023)						**Timer-XL** (Liu et al., 2025c)					
Self-Attention		Vanilla		BTA		Gain(%)		Vanilla		BTA		Gain(%)		Vanilla		BTA		Gain(%)	
Data	**ADF**	**MSE**	**MAE**	**MSE**	**MAE**	**ΔMSE**	**ΔMAE**	**MSE**	**MAE**	**MSE**	**MAE**	**ΔMSE**	**ΔMAE**	**MSE**	**MAE**	**MSE**	**MAE**	**ΔMSE**	**ΔMAE**
ETTh1	-5.91	1.096	0.835	**0.981**	**0.769**	10.50	7.88	0.516	0.477	**0.425**	**0.429**	17.51	10.02	0.409	0.430	**0.379**	**0.407**	7.28	5.35
ETTm1	-14.98	0.826	0.690	**0.756**	**0.652**	8.47	5.47	0.406	0.408	**0.371**	**0.395**	8.80	3.07	0.360	0.383	**0.353**	**0.380**	1.74	0.65
Weather	-26.68	0.463	0.459	**0.398**	**0.424**	14.05	7.73	0.265	0.286	**0.254**	**0.278**	4.34	2.80	0.240	0.273	**0.235**	**0.266**	1.98	2.48
Solar	-37.23	0.233	0.268	**0.220**	**0.261**	5.79	2.89	0.270	0.307	**0.241**	**0.272**	10.74	11.48	0.218	0.247	**0.193**	**0.242**	11.80	2.22

We integrate BTA into TimeBridge and compare the resulting model with ten reference algorithms for non-stationary LTSF. As shown in Table 2, BTA consistently improves the TimeBridge backbone across all eight datasets, reducing the average MSE and MAE by 2.30% and 1.89%, respectively. Moreover, TimeBridge+BTA achieves the best MSE on 7 of 8 datasets and the best MAE on 5, outperforming both general LTSF baselines and methods specifically designed for non-stationarity. These results demonstrate that adaptive temporal evidence can further strengthen a competitive non-stationary forecasting backbone and generalize effectively across diverse temporal dynamics.

Table 2: BTA outperforms other methods for non-stationary LTSF. Sequence length $T = 720$ for all models. Each dataset entry is averaged over $H \in \{96, 192, 336, 720\}$. Best and second-best results are shown in bold and underlined, respectively. Full results are available in Table 12.

Model	**Ours**		**PULSE** Liu et al. (2026)		**DTAF** Lu et al. (2026)		**DUET** Qiu et al. (2025)		**TimeBridge** Liu et al. (2025a)		**iTransformer** Liu et al. (2024)		**TimeMixer** Wang et al. (2024)		**PatchTST** Nie et al. (2023)		**TimesNet** Wu et al. (2023)		**DLinear** Zeng et al. (2023)		**Stationary** Liu et al. (2022)	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	**0.394**	0.423	0.412	**0.422**	0.416	0.435	0.404	0.427	0.399	0.426	0.439	0.448	0.427	0.441	0.516	0.477	0.488	0.452	0.423	0.437	0.644	0.554
ETTh2	**0.338**	**0.382**	0.361	0.391	0.363	0.396	0.344	0.390	0.344	0.384	0.374	0.406	0.349	0.397	0.351	0.395	0.414	0.427	0.431	0.447	0.521	0.487
ETTm1	0.337	0.371	0.363	0.388	0.338	**0.368**	**0.336**	0.370	0.348	0.379	0.362	0.391	0.356	0.380	0.406	0.408	0.399	0.406	0.357	0.379	0.527	0.472
ETTm2	**0.246**	**0.305**	0.262	0.314	0.249	0.307	0.255	0.316	0.250	0.307	0.269	0.329	0.257	0.318	0.256	0.317	0.314	0.333	0.267	0.332	0.567	0.462
Weather	**0.218**	**0.248**	0.239	0.271	0.222	0.251	0.230	0.264	0.220	0.250	0.233	0.271	0.265	0.286	0.226	0.264	0.259	0.287	0.240	0.300	0.295	0.315
ELC	**0.152**	**0.245**	0.160	0.255	0.160	0.248	0.162	0.253	0.154	0.249	0.164	0.261	0.185	0.284	0.159	0.253	0.193	0.295	0.166	0.264	0.194	0.296
Traffic	**0.346**	**0.242**	0.460	0.286	0.403	0.249	0.386	0.254	0.360	0.253	0.397	0.282	0.409	0.279	0.391	0.264	0.620	0.336	0.434	0.295	0.624	0.340
Solar	**0.176**	0.218	0.194	0.243	0.195	0.235	0.194	**0.213**	0.184	0.233	0.200	0.260	0.193	0.252	0.194	0.245	0.244	0.334	0.247	0.309	0.391	0.389
1st Count	7	5	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3 Case Study

To illustrate how BTA adapts positional information to heterogeneous temporal patterns, we select two ETTm1 windows with 720 historical time steps and use TimeBridge to forecast the subsequent 96 steps. Figure 2 visualizes the temporal kernels selected by BTA and their corresponding mixture weights. The left window exhibits a strong weekly periodic pattern, for which the router assigns 0.75 of the total weight to periodic kernels, including a maximum weight of 0.24 on a 7-day periodic

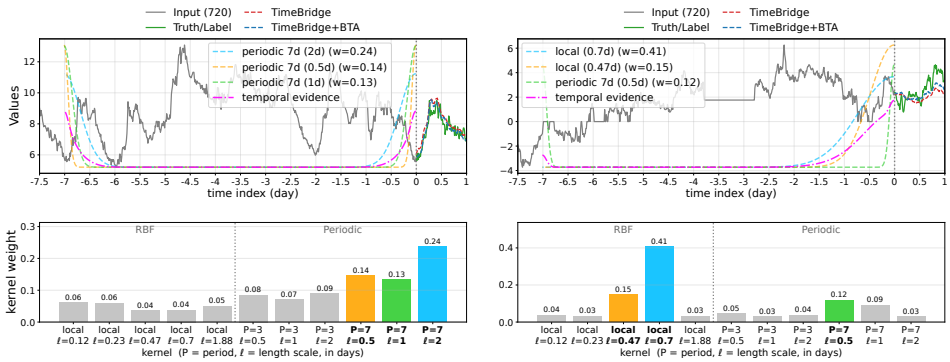


Figure 2: **Input-adaptive temporal kernel selection in BTA.** Two ETTm1 windows with a 720-step input and 96-step forecasting horizon. *Top:* TimeBridge and TimeBridge+BTA forecasts, together with the three highest-weight temporal kernels and their weighted combination. *Bottom:* MoE router weights over candidate kernels; for periodic kernels, ℓ denotes the peak width $\ell'_{r,e}p_e$. The left window emphasizes weekly periodicity, assigning 0.75 of the total weight to periodic kernels, whereas the right window emphasizes local dependence, assigning 0.66 to RBF kernels. BTA reduces the OT-channel MSE by approximately 37% in both cases ($0.219 \rightarrow 0.139$ and $0.970 \rightarrow 0.608$).

kernel. In contrast, the right window exhibits stronger local dependence, and BTA assigns 0.66 of the total weight to local RBF kernels, with the largest weight (0.41) placed on the 0.7-day local kernel. Consequently, BTA induces different positional biases for the two input windows and enables TimeBridge to better capture their distinct temporal dependencies, reducing the forecasting MSE by approximately 37% in both cases. This case study demonstrates the advantage of input-adaptive temporal evidence over a fixed positional structure uniformly across forecasting windows. More case studies in Appendix L further confirm this advantage in different datasets and backbones.

5.4 Representation Analysis

Figure 3 further shows that BTA improves not only non-stationary LTSF but also the interpretability of Transformer temporal behavior. The learned kernel weights vary systematically with the forecasting horizon: shorter horizons emphasize relatively local dependencies, whereas longer horizons assign greater importance to broader local and periodic patterns. Thus, BTA exposes how the temporal inductive bias of self-attention changes with the prediction horizon, providing an interpretable characterization of the temporal dependencies used for LTSF. Extra analysis on Solar in Appendix M.

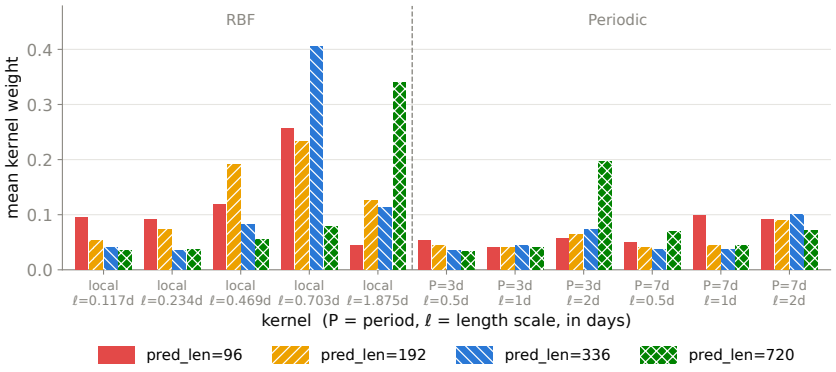


Figure 3: **BTA reveals horizon-dependent temporal dependencies.** Mean kernel weights across ETTm1 test samples and attention heads for $H \in \{96, 192, 336, 720\}$. The dashed line separates local RBF and periodic kernels. As the forecasting horizon changes, BTA shifts weight across temporal scales, with longer horizons emphasizing broader local and periodic patterns.

5.5 Hyperparameter Analysis

We analyze two MoE-router hyperparameters: Top-k and hidden dimension d_r. Figure 4 shows the ETTm1 results, with ETTh1 and Solar reported in Appendix N. Across datasets, BTA generally favors sparse expert selection: $k = 3$ performs best on ETTm1 and Solar, while ETTh1 prefers $k = 1$. Router capacity is also dataset-dependent and non-monotonic, with the best d_r equal to 192, 160, and 96 for ETTh1, ETTm1, and Solar, respectively. Overall, a small number of experts and moderate router capacity are sufficient, with d_r selected by validation performance. In addition, the complexity Analysis moved to Appendix O due to space limitation.

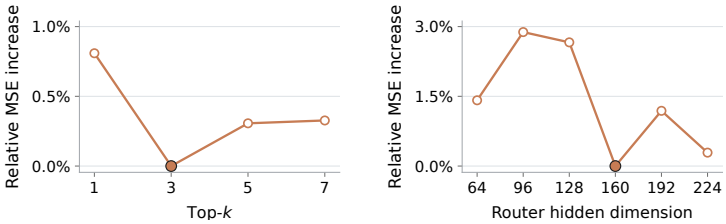


Figure 4: Hyperparameter sensitivity on ETTm1: Top-k (left) and router hidden dimension (right). Each panel shows the percentage MSE increase relative to its lowest MSE.

5.6 Ablation Study

Table 3 shows the contribution of each component in BTA. Adding local kernels improves vanilla self-attention, while combining local and periodic kernels provides further gains, demonstrating the benefit of a richer temporal kernel bank. Replacing static kernel weights with an MoE router using TE profile further reduces MSE and MAE, highlighting the value of input-adaptive kernel selection. Incorporating the Fourier spectrum of TE achieves the largest gains 5.65% and 6.55%, respectively, indicating that frequency-domain information provides complementary guidance for temporal-kernel routing. The ablation study of BTA on Weather in Appendix P.

Table 3: Ablation study of BTA on Solar (TimeBridge as the backbone).

Attention		Self-attention		Bayesian Time-aware Attention							
Optimization		Static		Static				MoE (TE only)		MoE (TE+Fourier)	
Kernels		None		Local		Local+Periodic		Local+Periodic		Local+Periodic	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Solar	96	0.161	0.224	0.158	0.219	0.157	0.213	0.152	0.201	**0.150**	**0.198**
	192	0.177	0.237	0.176	0.225	0.176	0.219	0.173	0.214	**0.171**	**0.211**
	336	0.196	0.232	0.192	0.230	0.191	0.227	0.187	0.227	**0.184**	**0.226**
	720	0.210	0.239	0.207	0.238	0.205	0.236	0.201	**0.235**	**0.197**	0.236
	Avg	0.186	0.233	0.183	0.228	0.182	0.224	0.178	0.219	**0.176**	**0.218**
	Gain(%)	–	–	1.48	2.15	2.02	3.97	4.17	5.90	**5.65**	**6.55**

6 Conclusions

We studied how positional encoding shapes Transformer forecasting under non-stationary temporal dependencies. By interpreting position-aware self-attention as Bayesian inference over attended positions, we derived a population-level ELBO and showed that static PE can be suboptimal when temporal dependence varies across input windows. Motivated by this analysis, we proposed BTA, a plug-and-play mechanism that adaptively combines complementary temporal kernels through an input- and head-specific MoE router. Experiments across multiple Transformer backbones and eight LTSF benchmarks demonstrate that BTA consistently improves forecasting accuracy and remains competitive with recent methods designed for non-stationarity. Overall, our results suggest that adaptive temporal evidence offers a principled, effective, and interpretable alternative to static positional structures for non-stationary time-series forecasting.

DISCLOSURE OF AI USE

During this work, we used ChatGPT-5.6 Sol solely to polish the language and improve the readability of the manuscript. We did not use generative AI to develop the theoretical framework, formulate mathematical claims, derive the ELBO or write proofs, design or implement BTA, design experiments, generate or process datasets, interpret results, or translate the manuscript. The authors reviewed all AI-assisted edits to ensure that they preserved the intended technical meaning and introduced no unsupported claims. The authors take full responsibility for the final manuscript, including its mathematical claims, experimental results, code, and conclusions.

REFERENCES

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819, 2023.

Ta-Chung Chi, Ting-Han Fan, Peter J. Ramadge, and Alexander I. Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Azad Deihim, Eduardo Alonso, and Dimitra Apostolopoulou. STTRE: A spatio-temporal transformer with relative embeddings for multivariate time series forecasting. *Neural Networks*, 168: 549–559, 2023. doi: 10.1016/j.neunet.2023.09.039. URL https://doi.org/10.1016/j.neunet.2023.09.039.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. AdaRNN: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 402–411. Association for Computing Machinery, 2021. doi: 10.1145/3459637.3482315. URL https://doi.org/10.1145/3459637.3482315.

Wenying Duan, Xiaoxi He, Lu Zhou, Lothar Thiele, and Hong Rao. Combating distribution shift for accurate time series forecasting via hypernetworks. In *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 900–907. IEEE, 2023. doi: 10.1109/ICPADS56603.2022.00121. URL https://doi.org/10.1109/ICPADS56603.2022.00121.

Graham Elliott, Thomas J Rothenberg, and James H Stock. Efficient tests for an autoregressive unit root, 1992.

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, pp. 7522–7529, 2023.

Wei Fan, Shun Zheng, Pengyang Wang, Rui Xie, Kun Yi, Qi Zhang, Jiang Bian, and Yanjie Fu. IN-Flow: Instance normalization flow for non-stationary time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 295–306. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709260. URL https://doi.org/10.1145/3690624.3709260.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data mining and knowledge discovery*, 38(1):22–48, 2024.

Maosheng Guo, Yu Zhang, and Ting Liu. Gaussian transformer: a lightweight approach for natural language inference. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 6489–6496, 2019.

Kareem Hegazy, Michael W Mahoney, and N Benjamin Erichson. Powerformer: A transformer with weighted causal attention for time-series forecasting. *arXiv preprint arXiv:2502.06151*, 2025.

Habib Irani and Vangelis Metsis. Positional encoding in transformer-based time series models: a survey. *arXiv preprint arXiv:2502.12370*, 2025.

Dongbin Kim, Jinseong Park, Jaewook Lee, and Hoki Kim. Are self-attentions effective for time series forecasting? *Advances in Neural Information Processing Systems*, 37:114180–114209, 2024.

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.

Siva Rama Krishna Kottapalli, Karthik Hubli, Sandeep Chandrashekhara, Garima Jain, Sunayana Hubli, Gayathri Botla, and Ramesh Doddaiah. Foundation models for time series: A survey. *arXiv preprint arXiv:2504.04011*, 2025.

Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Tao Sun, Yongjun Xu, and Fei Wang. Apt: Affine prototype-timestamp for time series forecasting under distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15180–15188, 2026.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. In *International Conference on Machine Learning*, pp. 39815–39840. PMLR, 2025a.

Qinglong Liu, Cong Xu, Wenhao Jiang, Kaixuan Wang, Lin Ma, and Haifeng Li. Timestacker: A novel framework with multilevel observation for capturing nonstationary patterns in time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025b.

Siqi Liu and Andreas Lehrmann. DynaConF: Dynamic forecasting of non-stationary time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=48pHFcg0YO.

Yangyou Liu, Zezhi Shao, Xinyu Chen, Hu Chen, Fei Wang, and Yuankai Wu. Pulse: Generative phase evolution for non-stationary time series forecasting. In *Forty-third International Conference on Machine Learning*, 2026.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 83982–84006, 2025c.

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.

Antoine Liutkus, Ondřej Cıfka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pp. 7067–7079. PMLR, 2021.

Junkai Lu, Peng Chen, Chenjuan Guo, Yang Shu, Meng Wang, and Bin Yang. Towards non-stationary time series forecasting with temporal stabilization and frequency differencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24070–24078, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Tenth International Conference on Learning Representations*, 2023.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven C. H. Hoi. Learning fast and slow for online time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=q-PbpHD3EOk.

Xihao Piao, Zheng Chen, Yushun Dong, Yasuko Matsubara, and Yasushi Sakurai. FredNormer: Frequency domain normalization for non-stationary time series forecasting, 2024. URL https://arxiv.org/abs/2410.01860.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1185–1196. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709325. URL https://doi.org/10.1145/3690624.3709325.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Li Shen, Yuning Wei, and Yangzhu Wang. Gbt: Two-stage transformer framework for non-stationary time series forecasting. *Neural Networks*, 165:953–970, 2023. doi: 10.1016/j.neunet.2023.06.044.

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

Tian Sun, Yuqi Chen, and Weiwei Sun. Penguin: Enhancing transformer with periodic-nested group attention for long-term time series forecasting. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=CtoIG9Iwas.

Suxin Tong and Jingling Yuan. PAREformer: Positional adaptive and recurrent enhanced transformer for time series forecasting. *Information Fusion*, pp. 103967, 2025. doi: 10.1016/j.inffus.2025.103967.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Rui Wang, Yihe Dong, Sercan O. Arik, and Rose Yu. Koopman neural forecaster for time series with temporal distribution shifts. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kUmdmHxK5N.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Haixu Wu, Tengge Hu, Yong Liu, and et al. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

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.

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

Yi-Fan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. OneNet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69949–69980, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dd6a47bc0aad6f34aa5e77706d90cdc4-Abstract-Conference.html.

Yifan Zhang, Rui Wu, and Jr. Harris, Frederick C. Sparse transformer with local and seasonal adaptation for multivariate time series forecasting. *Scientific Reports*, 14:15909, 2024. doi: 10.1038/s41598-024-66886-1. URL https://doi.org/10.1038/s41598-024-66886-1.

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, pp. 11106–11115, 2021.

A THE SIMPLIFIED TRANSFORMER ARCHITECTURE

Here we provide the detailed configuration of the simplified Transformer used in the theoretical analysis. Consider a point-wise Transformer consisting of a single self-attention encoder, average pooling over the T historical positions, and a linear forecasting decoder.

Given a historical input $\boldsymbol{x}^- \in \mathbb{R}^T$, each scalar observation is first mapped to a d_x-dimensional representation through a linear embedding,

$$\boldsymbol{H} = \boldsymbol{x}^- \boldsymbol{w}_e \in \mathbb{R}^{T \times d_x}, \tag{8}$$

where $\boldsymbol{w}_e \in \mathbb{R}^{1 \times d_x}$ denotes the embedding vector.

The query, key, and value representations are

$$\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{W}_q, \qquad \boldsymbol{K} = \boldsymbol{H}\boldsymbol{W}_k, \qquad \boldsymbol{V} = \boldsymbol{H}\boldsymbol{W}_v, \tag{9}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k \in \mathbb{R}^{d_x \times d_k}$ and $\boldsymbol{W}_v \in \mathbb{R}^{d_x \times d_v}$ are learnable projection matrices. Thus, the ith query and jth key are

$$\boldsymbol{q}_i = x_i^- \boldsymbol{w}_e \boldsymbol{W}_q \in \mathbb{R}^{1 \times d_k}, \qquad \boldsymbol{k}_j = x_j^- \boldsymbol{w}_e \boldsymbol{W}_k \in \mathbb{R}^{1 \times d_k}.$$

For a positional encoding method ϕ, let $\boldsymbol{\alpha}_\phi(\boldsymbol{x}^-) \in \mathbb{R}^{T \times T}$ denote the corresponding attention matrix, whose (i, j)th entry is

$$\alpha_{\phi,ij}(\boldsymbol{x}^-) = \frac{\exp\left(\boldsymbol{q}_i \boldsymbol{k}_j^\top / \sqrt{d_k} + b_{ij}^\phi(\boldsymbol{x}^-)\right)}{\sum_{\ell \in \mathcal{C}(i)} \exp\left(\boldsymbol{q}_i \boldsymbol{k}_\ell^\top / \sqrt{d_k} + b_{i\ell}^\phi(\boldsymbol{x}^-)\right)}, \qquad j \in \mathcal{C}(i), \tag{10}$$

and $\alpha_{\phi,ij}(\boldsymbol{x}^-) = 0$ for $j \notin \mathcal{C}(i)$. Here, $b_{ij}^\phi(\boldsymbol{x}^-)$ denotes the PE-induced positional score (this dependence on $\boldsymbol{x}^-$ is absent for static PE) and $\mathcal{C}(i)$ is the set of key positions accessible to query i, as defined in Section 3.1.

The output of the self-attention encoder is

$$\boldsymbol{H}_{\text{attn}} = \boldsymbol{\alpha}_\phi(\boldsymbol{x}^-)\boldsymbol{V} \in \mathbb{R}^{T \times d_v}. \tag{11}$$

Following the convention (Nie et al., 2023), we aggregate the T output representations using average pooling,

$$\overline{\boldsymbol{h}}_\phi^\top(\boldsymbol{x}^-) = \frac{1}{T}\mathbf{1}_T^\top \boldsymbol{\alpha}_\phi(\boldsymbol{x}^-)\boldsymbol{V} \in \mathbb{R}^{1 \times d_v}, \tag{12}$$

and apply a linear decoder $\boldsymbol{W}_d \in \mathbb{R}^{L \times d_v}$ to obtain the L-step forecast,

$$f_{\theta,\phi}(\boldsymbol{x}^-) = \boldsymbol{W}_d \overline{\boldsymbol{h}}_\phi(\boldsymbol{x}^-) \in \mathbb{R}^L, \tag{13}$$

where $\theta = \{\boldsymbol{w}_e, \boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v, \boldsymbol{W}_d\}$ collects the learnable model parameters.

Since

$$\boldsymbol{V} = \boldsymbol{x}^- \boldsymbol{w}_e \boldsymbol{W}_v,$$

Equation 13 can be rewritten as

$$f_{\theta,\phi}(\boldsymbol{x}^-) = \boldsymbol{W}_d(\boldsymbol{w}_e \boldsymbol{W}_v)^\top \left[\frac{1}{T}\mathbf{1}_T^\top \boldsymbol{\alpha}_\phi(\boldsymbol{x}^-)\right] \boldsymbol{x}^-. \tag{14}$$

Define

$$\boldsymbol{w} := \boldsymbol{W}_d(\boldsymbol{w}_e \boldsymbol{W}_v)^\top \in \mathbb{R}^{L \times 1} \tag{15}$$

and

$$\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-) := \frac{1}{T}\mathbf{1}_T^\top \boldsymbol{\alpha}_\phi(\boldsymbol{x}^-) \in \mathbb{R}^{1 \times T}. \tag{16}$$

The simplified Transformer therefore admits the compact representation

$$f_{\theta,\phi}(\boldsymbol{x}^-) = \boldsymbol{w}\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-)\boldsymbol{x}^- = \boldsymbol{M}_{\theta,\phi}(\boldsymbol{x}^-)\boldsymbol{x}^-, \tag{17}$$

where

$$\boldsymbol{M}_{\theta,\phi}(\boldsymbol{x}^-) := \boldsymbol{w}\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-) \in \mathbb{R}^{L\times T}. \tag{18}$$

This recovers the effective forecasting operator introduced in Equation 2.

Because each row of $\boldsymbol{\alpha}_\phi(\boldsymbol{x}^-)$ is an attention distribution over its accessible key positions, $\boldsymbol{a}_\phi^\top(\boldsymbol{x}^-)$ represents the attention coefficients averaged across the T query positions. Thus, the effect of PE method ϕ on the simplified Transformer enters the forecasting operator through its influence on the attention distribution.

For analytical clarity, we omit layer normalization, feed-forward sublayers, and residual connections. Including a residual connection in the attention block replaces $\boldsymbol{\alpha}_\phi(\boldsymbol{x}^-)$ by $\boldsymbol{\alpha}_\phi(\boldsymbol{x}^-) + \boldsymbol{I}_T$ in the derivation above and does not alter the resulting effective linear-operator representation.

B Proof of Proposition 1

Proof. For a fixed query position i, let $S_i \in \mathcal{C}(i)$ denote the selected key position. We treat the query $\boldsymbol{q}_i$ and candidate keys $\{\boldsymbol{k}_j : j \in \mathcal{C}(i)\}$ as fixed representations. Without positional information, scaled dot-product self-attention induces the content-based prior

$$p(S_i = j) = \frac{\exp\big(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{d_k}\big)}{\sum_{\ell\in\mathcal{C}(i)} \exp\big(\boldsymbol{q}_i^\top \boldsymbol{k}_\ell/\sqrt{d_k}\big)}. \tag{19}$$

Hence,

$$\log p(S_i = j) \overset{c}{=} \frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d_k}}, \tag{20}$$

where $\overset{c}{=}$ denotes equality up to an additive term independent of j.

Let $\boldsymbol{Z}_i = \{Z_{ij}\}_{j\in\mathcal{C}(i)}$ denote the full temporal evidence associated with query i, with observed realization $\boldsymbol{z}_i = \{z_{ij}\}_{j\in\mathcal{C}(i)}$. By the candidate-specific assumption in Proposition 1, there exists a factor $C_i(\boldsymbol{z}_i) > 0$, independent of the candidate position j, such that

$$p(\boldsymbol{Z}_i = \boldsymbol{z}_i \mid S_i = j) = C_i(\boldsymbol{z}_i)\, p(Z_{ij} = z_{ij} \mid S_i = j). \tag{21}$$

Here and below, p denotes a probability mass or density as appropriate.

Applying Bayes' rule and using Equation 21,

$$p(S_i = j \mid \boldsymbol{Z}_i = \boldsymbol{z}_i) = \frac{p(S_i = j)\, p(Z_{ij} = z_{ij} \mid S_i = j)}{\sum_{\ell\in\mathcal{C}(i)} p(S_i = \ell)\, p(Z_{i\ell} = z_{i\ell} \mid S_i = \ell)}, \tag{22}$$

because the common factor $C_i(\boldsymbol{z}_i)$ cancels between the numerator and denominator. Therefore,

$$\log p(S_i = j \mid \boldsymbol{Z}_i = \boldsymbol{z}_i) \overset{c}{=} \frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d_k}} + \log p(Z_{ij} = z_{ij} \mid S_i = j). \tag{23}$$

On the other hand, self-attention with PE assigns

$$\alpha_{ij} = \frac{\exp\big(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{d_k} + b_{ij}\big)}{\sum_{\ell\in\mathcal{C}(i)} \exp\big(\boldsymbol{q}_i^\top \boldsymbol{k}_\ell/\sqrt{d_k} + b_{i\ell}\big)}. \tag{24}$$

Thus, if PE specifies the temporal-evidence likelihood such that

$$b_{ij} \overset{c}{=} \log p(Z_{ij} = z_{ij} \mid S_i = j), \tag{25}$$

then Equations 23 and 24 have identical normalized logits. Consequently,

$$\alpha_{ij} = p(S_i = j \mid \boldsymbol{Z}_i = \boldsymbol{z}_i). \tag{26}$$

Therefore, position-aware self-attention can be interpreted as Bayesian reweighting over candidate key positions: scaled dot-product similarity defines the content-based prior, while PE contributes the candidate-specific temporal-evidence likelihood.

C Probabilistic Interpretations of Positional Encoding

Proposition 1 provides a unified probabilistic interpretation of representative positional encoding (PE) mechanisms, including Gaussian/RBF positional bias (Guo et al., 2019), ALiBi (Press et al., 2021), KERPLE (Chi et al., 2022), the power-law decay bias of Powerformer (Hegazy et al., 2025), and RoPE (Su et al., 2024). Specifically, an additive positional term in the attention logits can be interpreted as the log-likelihood contribution of temporal evidence, up to an additive normalization term that is invariant to the key position j and therefore cancels under the softmax. Table 4 summarizes the resulting probabilistic interpretations and illustrates how different PE mechanisms encode distinct assumptions about the dependence of temporal evidence on the relative lag $\Delta_{ij} = i - j$.

Table 4: Probabilistic interpretations of representative PE methods, where $\Delta_{ij} = i-j$. Here, $\sigma^2 > 0$ denotes the variance of the Gaussian/RBF kernel; $m_h > 0$ is the head-specific ALiBi slope; $r_1, r_2 > 0$ are KERPLE parameters controlling the magnitude and scale of its logarithmic decay, respectively; and $\alpha > 0$ is the power-law decay exponent in Powerformer. Log-likelihood terms are shown up to additive normalization constants that do not affect the softmax over j.

PE method	Domain	$p(Z_{ij} = z_{ij} \mid S_i = j)$	$\log p(Z_{ij} = z_{ij} \mid S_i = j)$
Gaussian/RBF	NLP	$\propto \exp\left(-\frac{\Delta_{ij}^2}{2\sigma^2}\right)$	$-\frac{\Delta_{ij}^2}{2\sigma^2}$
ALiBi	NLP	$\propto \exp(-m_h\lvert\Delta_{ij}\rvert)$	$-m_h\lvert\Delta_{ij}\rvert$
KERPLE	NLP	$\propto (1 + r_2\lvert\Delta_{ij}\rvert)^{-r_1}$	$-r_1 \log(1 + r_2\lvert\Delta_{ij}\rvert)$
Powerformer	TSF	$\propto (1 + \lvert\Delta_{ij}\rvert)^{-\alpha}$	$-\alpha \log(1 + \lvert\Delta_{ij}\rvert)$
RoPE	NLP & TSF	—	$\{\boldsymbol{q}_i^\top \left(\boldsymbol{R}_{\Delta_{ij}} - \boldsymbol{I}\right) \boldsymbol{k}_j\}/\sqrt{d_k}$

The first four methods can therefore be understood as imposing different assumptions on how temporal evidence decays with relative distance. Gaussian/RBF bias yields a quadratic log-likelihood, $-\frac{\Delta_{ij}^2}{2\sigma^2}$, which induces rapid decay and favors temporally local interactions. The variance σ^2 controls the effective temporal range: a larger σ^2 produces a broader likelihood and assigns relatively more weight to distant positions. ALiBi instead yields the linear decay

$$-m_h|\Delta_{ij}|,$$

which corresponds to a Laplace-type temporal likelihood. Its head-specific slope m_h determines how strongly each attention head penalizes distant positions.

KERPLE introduces the heavier-tailed logarithmic decay

$$-r_1 \log(1 + r_2|\Delta_{ij}|)\,,$$

where r_1 controls the overall strength of the decay and r_2 determines its distance scale. Similarly, the Powerformer bias

$$-\alpha \log(1 + |\Delta_{ij}|)$$

corresponds to a power-law temporal likelihood, with α controlling the decay rate. Compared with Gaussian or exponential decay, these logarithmic forms assign relatively greater likelihood to distant positions and therefore permit heavier-tailed temporal dependencies.

More generally, σ^2, m_h, r_1, r_2, and α determine the temporal scale and tail behavior of their corresponding evidence likelihoods. Thus, different PE mechanisms can be viewed not merely as alternative deterministic position transformations, but as encoding different probabilistic assumptions about the range and strength of temporal dependencies.

RoPE admits a related but qualitatively different interpretation. Let $\boldsymbol{R}_{\Delta_{ij}}$ denote the relative rotation matrix associated with lag Δ_{ij}. The RoPE attention score can be decomposed as

$$(\boldsymbol{q}_i^\top \boldsymbol{R}_{\Delta_{ij}} \boldsymbol{k}_j)/\sqrt{d_k} = \{\boldsymbol{q}_i^\top \boldsymbol{k}_j + \boldsymbol{q}_i^\top \left(\boldsymbol{R}_{\Delta_{ij}} - \boldsymbol{I}\right) \boldsymbol{k}_j\}/\sqrt{d_k},$$

where $\boldsymbol{I}$ denotes the identity matrix. Under Proposition 1, with $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ treated as fixed representations, the second term can therefore be interpreted as the candidate-specific log temporal-evidence

contribution,

$$\log p(Z_{ij} = z_{ij} \mid S_i = j) \overset{c}{=} \frac{\boldsymbol{q}_i^\top \left(\boldsymbol{R}_{\Delta_{ij}} - \boldsymbol{I}\right) \boldsymbol{k}_j}{\sqrt{d_k}},$$

where $\overset{c}{=}$ denotes equality up to an additive normalization term independent of j and therefore irrelevant to the attention softmax. Unlike Gaussian/RBF, ALiBi, KERPLE, and Powerformer, whose positional terms are determined primarily by relative distance, the temporal-evidence contribution induced by RoPE depends jointly on the relative lag Δ_{ij} and the content representations $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$. Consequently, the same temporal lag can contribute different amounts of evidence for different query–key pairs. This distinction highlights the generality of Proposition 1: both distance-dependent positional biases and content-dependent relative rotations can be interpreted within a common probabilistic framework as mechanisms that modify the temporal evidence entering self-attention.

D Proof of Theorem 2

We prove Theorem 2 in five steps:

1. **Sample-level ELBO.** We derive the standard variational lower bound for a forecasting sample $(\boldsymbol{x}^-, \boldsymbol{y})$ with latent temporal evidence $\boldsymbol{Z}$.
2. **PE-induced future-free posterior.** We specialize the variational family to $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$, reflecting that PE depends only on the historical input.
3. **Population-level ELBO.** We average the sample-level bound over $\mathcal{D} = p(\boldsymbol{X}^-, \boldsymbol{Y})$.
4. **Covariance-based population form.** Under the Gaussian forecasting likelihood, we express the forecasting term through q_ϕ-weighted second-order moments. For static PE, these reduce to the population covariance matrices $\boldsymbol{\Sigma}_-$ and $\boldsymbol{\Sigma}_{+-}$.
5. **Prior and posterior of temporal evidence.** We derive the GP-based prior $p(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ and parameterize the PE-induced variational posterior $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ using the first- and second-order statistics of lag-wise PE profiles.

D.1 Step 1: Sample-level ELBO

Consider a forecasting sample $(\boldsymbol{x}^-, \boldsymbol{y})$ and latent temporal evidence $\boldsymbol{Z}$. The conditional marginal likelihood is

$$p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-) = \int p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-, \boldsymbol{z}) p(\boldsymbol{z} \mid \boldsymbol{x}^-)\, d\boldsymbol{z}. \tag{27}$$

For any variational distribution $q(\boldsymbol{Z} \mid \boldsymbol{x}^-, \boldsymbol{y})$ whose support is contained in that of $p(\boldsymbol{Z} \mid \boldsymbol{x}^-)$, Jensen's inequality gives

$$\begin{aligned} \log p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-) \geq\ & \mathbb{E}_{q(\boldsymbol{Z} \mid \boldsymbol{x}^-, \boldsymbol{y})} \left[\log p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-, \boldsymbol{Z})\right] \\ & - D_{\mathrm{KL}} \left(q(\boldsymbol{Z} \mid \boldsymbol{x}^-, \boldsymbol{y}) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{x}^-)\right). \end{aligned} \tag{28}$$

Thus, Equation 28 is the standard sample-level variational lower bound and requires neither the Gaussian forecasting assumption nor weak stationarity.

D.2 Step 2: PE-induced Future-free Variational Posterior

PE is determined from the historical input and cannot access the future target at inference time. We therefore choose the future-free PE-induced variational family $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$. Substituting this family into Equation 28 gives

$$\begin{aligned} \log p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-) \geq\ & \mathbb{E}_{q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)} \left[\log p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-, \boldsymbol{Z})\right] \\ & - D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{x}^-)\right). \end{aligned} \tag{29}$$

Accordingly, define the future-free sample-level ELBO as

$$\begin{aligned} \widetilde{\mathcal{L}}_{\mathrm{ELBO}}(\theta, \phi; \boldsymbol{x}^-, \boldsymbol{y}) :=\ & \mathbb{E}_{q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)} \left[\log p_{\theta,\phi}(\boldsymbol{y} \mid \boldsymbol{x}^-, \boldsymbol{Z})\right] \\ & - D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{x}^-)\right). \end{aligned} \tag{30}$$

Although ϕ may be learned from forecasting errors involving future targets during training, evaluating $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$ depends only on the historical input.

D.3 STEP 3: POPULATION-LEVEL ELBO

Let $\mathcal{D} = p(\boldsymbol{X}^-, \boldsymbol{Y})$ denote the population distribution of historical and future observations. Taking expectation of Equation 30 with respect to $\mathcal{D}$ defines

$$\mathcal{L}_{\text{pop}}(\theta, \phi) := \mathbb{E}_{\mathcal{D}} \left[\widetilde{\mathcal{L}}_{\text{ELBO}}(\theta, \phi; \boldsymbol{X}^-, \boldsymbol{Y}) \right]. \tag{31}$$

Substituting Equation 30 gives

$$\begin{aligned} \mathcal{L}_{\text{pop}}(\theta, \phi) = \mathbb{E}_{\mathcal{D}} \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X}^-)} \left[\log p_{\theta,\phi}(\boldsymbol{Y} \mid \boldsymbol{X}^-, \boldsymbol{Z}) \right] \\ - \mathbb{E}_{\boldsymbol{X}^-} \left[D_{\text{KL}} \left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{X}^-) \right) \right]. \end{aligned} \tag{32}$$

Since Equation 29 holds for every forecasting sample,

$$\mathbb{E}_{\mathcal{D}} \left[\log p_{\theta,\phi}(\boldsymbol{Y} \mid \boldsymbol{X}^-) \right] \geq \mathcal{L}_{\text{pop}}(\theta, \phi). \tag{33}$$

Therefore, $\mathcal{L}_{\text{pop}}(\theta, \phi)$ is a population-level evidence lower bound.

D.4 STEP 4: COVARIANCE-BASED POPULATION FORM

We next characterize the forecasting term in Equation 32. Throughout this step, the model parameters θ are fixed, and their dependence is suppressed for notational simplicity. Conditioned on $\boldsymbol{Z} = \boldsymbol{z}$, assume the Gaussian forecasting likelihood

$$p_{\theta,\phi} \left(\boldsymbol{Y} \mid \boldsymbol{X}^-, \boldsymbol{Z} = \boldsymbol{z} \right) = \mathcal{N} \left(\boldsymbol{Y}; \boldsymbol{M}_{\phi,\boldsymbol{z}} \boldsymbol{X}^-, \sigma_\epsilon^2 \boldsymbol{I}_L \right), \tag{34}$$

where $\boldsymbol{M}_{\phi,\boldsymbol{z}} \in \mathbb{R}^{L \times T}$ denotes the effective forecasting operator associated with temporal evidence $\boldsymbol{z}$. Relative to the original input-dependent operator $\boldsymbol{M}_{\theta,\phi}(\boldsymbol{x}^-)$, Equation 34 uses a frozen-attention representation in which the forecasting operator is fixed conditional on $\boldsymbol{Z}$. Importantly, $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ remains input-dependent and is not replaced by a population marginal.

From Equation 34,

$$\log p_{\theta,\phi} \left(\boldsymbol{Y} \mid \boldsymbol{X}^-, \boldsymbol{Z} \right) = -\frac{L}{2} \log(2\pi\sigma_\epsilon^2) - \frac{1}{2\sigma_\epsilon^2} \left\| \boldsymbol{Y} - \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^- \right\|_2^2. \tag{35}$$

Expanding the squared error yields

$$\begin{aligned} \left\| \boldsymbol{Y} - \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^- \right\|_2^2 = \boldsymbol{Y}^\top \boldsymbol{Y} - 2\boldsymbol{Y}^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^- \\ + (\boldsymbol{X}^-)^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}}^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^-. \end{aligned} \tag{36}$$

To preserve the dependence between $\boldsymbol{Z}$ and $\boldsymbol{X}^-$ induced by $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$, define the q_ϕ-weighted second-order moments

$$\boldsymbol{\Omega}_{-,\phi}(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{X}^-} \left[q_\phi(\boldsymbol{z} \mid \boldsymbol{X}^-) \boldsymbol{X}^- (\boldsymbol{X}^-)^\top \right] \tag{37}$$

and

$$\boldsymbol{\Omega}_{+-,\phi}(\boldsymbol{z}) := \mathbb{E}_{\mathcal{D}} \left[q_\phi(\boldsymbol{z} \mid \boldsymbol{X}^-) \boldsymbol{Y} (\boldsymbol{X}^-)^\top \right]. \tag{38}$$

These quantities retain the dependence between the PE-induced temporal evidence and the historical input.

Because $\int q_\phi(\boldsymbol{z} \mid \boldsymbol{X}^-) \, d\boldsymbol{z} = 1$, the first term in Equation 36 satisfies

$$\mathbb{E}_{\mathcal{D}} \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X}^-)} \left[\boldsymbol{Y}^\top \boldsymbol{Y} \right] = \text{tr}(\boldsymbol{\Sigma}_+). \tag{39}$$

The cross term satisfies

$$\mathbb{E}_{\mathcal{D}} \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X}^-)} \left[\boldsymbol{Y}^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^- \right] = \int \text{tr} \left(\boldsymbol{M}_{\phi,\boldsymbol{z}} \boldsymbol{\Omega}_{+-,\phi}^\top(\boldsymbol{z}) \right) d\boldsymbol{z}, \tag{40}$$

and the quadratic term satisfies

$$\mathbb{E}_{\mathcal{D}} \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X}^-)} \left[(\boldsymbol{X}^-)^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}}^\top \boldsymbol{M}_{\phi,\boldsymbol{Z}} \boldsymbol{X}^- \right] = \int \text{tr} \left(\boldsymbol{M}_{\phi,\boldsymbol{z}} \boldsymbol{\Omega}_{-,\phi}(\boldsymbol{z}) \boldsymbol{M}_{\phi,\boldsymbol{z}}^\top \right) d\boldsymbol{z}. \tag{41}$$

Substituting Equations 39–41 into Equation 35 gives

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}\mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X}^-)}\left[\log p_{\theta,\phi}(\boldsymbol{Y}\mid\boldsymbol{X}^-,\boldsymbol{Z})\right]\\&=C_0+\frac{1}{\sigma_\epsilon^2}\int\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{z}}\boldsymbol{\Omega}_{+-,\phi}^\top(\boldsymbol{z})\right)d\boldsymbol{z}\\&\quad-\frac{1}{2\sigma_\epsilon^2}\int\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{z}}\boldsymbol{\Omega}_{-,\phi}(\boldsymbol{z})\boldsymbol{M}_{\phi,\boldsymbol{z}}^\top\right)d\boldsymbol{z},\end{aligned}\tag{42}$$

where

$$C_0=-\frac{1}{2\sigma_\epsilon^2}\operatorname{tr}(\boldsymbol{\Sigma}_+)-\frac{L}{2}\log(2\pi\sigma_\epsilon^2).\tag{43}$$

Combining Equation 42 with Equation 32 yields

$$\begin{aligned}\mathcal{L}_{\text{pop}}(\phi)=C_0&+\frac{1}{\sigma_\epsilon^2}\int\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{z}}\boldsymbol{\Omega}_{+-,\phi}^\top(\boldsymbol{z})\right)d\boldsymbol{z}\\&-\frac{1}{2\sigma_\epsilon^2}\int\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{z}}\boldsymbol{\Omega}_{-,\phi}(\boldsymbol{z})\boldsymbol{M}_{\phi,\boldsymbol{z}}^\top\right)d\boldsymbol{z}\\&-\mathbb{E}_{\boldsymbol{X}^-}\left[D_{\text{KL}}\left(q_\phi(\boldsymbol{Z}\mid\boldsymbol{X}^-)\,\|\,p(\boldsymbol{Z}\mid\boldsymbol{X}^-)\right)\right].\end{aligned}\tag{44}$$

Equation 44 preserves the full dependence of $q_\phi(\boldsymbol{Z}\mid\boldsymbol{X}^-)$ on the historical input and therefore requires no population-marginal approximation.

For a static PE,

$$q_\phi(\boldsymbol{z}\mid\boldsymbol{X}^-)=q_\phi(\boldsymbol{z}),\tag{45}$$

so the weighted moments factor as

$$\boldsymbol{\Omega}_{-,\phi}(\boldsymbol{z})=q_\phi(\boldsymbol{z})\boldsymbol{\Sigma}_-,\qquad\boldsymbol{\Omega}_{+-,\phi}(\boldsymbol{z})=q_\phi(\boldsymbol{z})\boldsymbol{\Sigma}_{+-}.\tag{46}$$

Equation 44 therefore reduces to

$$\begin{aligned}\mathcal{L}_{\text{pop}}(\phi)=C_0&+\frac{1}{\sigma_\epsilon^2}\mathbb{E}_{q_\phi}\left[\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{Z}}\boldsymbol{\Sigma}_{+-}^\top\right)\right]\\&-\frac{1}{2\sigma_\epsilon^2}\mathbb{E}_{q_\phi}\left[\operatorname{tr}\left(\boldsymbol{M}_{\phi,\boldsymbol{Z}}\boldsymbol{\Sigma}_-\boldsymbol{M}_{\phi,\boldsymbol{Z}}^\top\right)\right]\\&-\mathbb{E}_{\boldsymbol{X}^-}\left[D_{\text{KL}}\left(q_\phi(\boldsymbol{Z})\,\|\,p(\boldsymbol{Z}\mid\boldsymbol{X}^-)\right)\right].\end{aligned}\tag{47}$$

Thus, for static PE, the population ELBO admits a direct covariance-based representation in terms of the historical covariance $\boldsymbol{\Sigma}_-$ and the future–history cross-covariance $\boldsymbol{\Sigma}_{+-}$.

D.5 Step 5: Prior and Posterior of Temporal Evidence

We finally specify the prior and PE-induced variational posterior of the latent temporal evidence. For the lag-wise population analysis, we aggregate the pairwise temporal evidence in Proposition 1 according to relative lag and represent the resulting temporal structure through a latent process $Z(\tau)$, where $\tau=|i-j|$. The prior is induced by the Gaussian process governing the historical input, whereas the variational posterior is parameterized directly by the lag-wise positional evidence supplied by PE.

GP-derived prior over temporal evidence. Let $\mathcal{T}=\{1,\dots,K\}$, with $K\le T-1$, denote the set of relative lags considered in the theoretical analysis. For each $\tau\in\mathcal{T}$, consider the empirical autocovariance statistic

$$U_\tau=\frac{1}{T-\tau}\sum_{t=1}^{T-\tau}X_tX_{t+\tau}.\tag{48}$$

Since $\{X_t\}$ is a zero-mean weakly stationary Gaussian process,

$$\mathbb{E}[U_\tau]=\Gamma(\tau).\tag{49}$$

For $h, k \in \mathcal{T}$, Isserlis' theorem further gives

$$\begin{aligned} k_Z(h,k) &:= \operatorname{Cov}(U_h, U_k) \\ &= \frac{1}{(T-h)(T-k)} \sum_{t=1}^{T-h} \sum_{s=1}^{T-k} \Big[\Gamma(t-s)\Gamma(t+h-s-k) \\ &\qquad + \Gamma(t-s-k)\Gamma(t+h-s) \Big]. \end{aligned} \tag{50}$$

Thus, the first- and second-order statistics of the lag-wise temporal evidence are determined by the autocovariance function $\Gamma(\tau)$ of the underlying Gaussian process.

Motivated by these exact first- and second-order moments, we adopt a moment-matched Gaussian-process prior for the latent temporal evidence on the finite lag domain $\mathcal{T}$,

$$Z(\tau) \sim \mathcal{GP}\left(\Gamma(\tau), k_Z(\tau, \tau')\right), \qquad \tau, \tau' \in \mathcal{T}. \tag{51}$$

This Gaussian construction is a moment-matched prior: the mean and covariance are inherited from the empirical autocovariance statistics U_τ, without requiring the statistics themselves to be jointly Gaussian. For the finite lag set $\mathcal{T}$, define

$$\boldsymbol{Z} = [Z(1), \ldots, Z(K)]^\top, \qquad \boldsymbol{\gamma} = [\Gamma(1), \ldots, \Gamma(K)]^\top, \tag{52}$$

and let $\boldsymbol{K}_Z \in \mathbb{R}^{K \times K}$ satisfy $[\boldsymbol{K}_Z]_{hk} = k_Z(h,k)$. Motivated by these exact moments, we specify the conditional temporal-evidence prior used in the ELBO as

$$p(\boldsymbol{Z} \mid \boldsymbol{X}^-) = \mathcal{N}\left(\boldsymbol{\gamma}, \boldsymbol{K}_0\right), \qquad \boldsymbol{K}_0 = \boldsymbol{K}_Z + \epsilon_0 \boldsymbol{I}_K, \tag{53}$$

where $\epsilon_0 > 0$ ensures positive definiteness. Because $\boldsymbol{K}_Z$ is a covariance matrix, it is positive semidefinite by construction.

We retain the conditional notation $p(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ to emphasize that the temporal-evidence prior is associated with the Gaussian process governing the historical input. Under a fixed weakly stationary process, $\Gamma(\tau)$, and hence $\boldsymbol{\gamma}$ and $\boldsymbol{K}_0$, are invariant across historical realizations. Under regime-conditioned non-stationarity, however, these quantities may vary across regimes through the regime-specific autocovariance function.

Lag-wise PE profiles. We next characterize the temporal evidence represented by PE. Let

$$\mathcal{Q}_K = \{i : i - \tau \in \mathcal{C}(i),\ \forall \tau \in \mathcal{T}\}$$

denote the query positions for which all K relative lags are accessible. For each $i \in \mathcal{Q}_K$, define the lag-wise PE profile

$$\boldsymbol{b}_{\phi,i}(\boldsymbol{x}^-) = \left[b^\phi_{i,i-1}(\boldsymbol{x}^-), \ldots, b^\phi_{i,i-K}(\boldsymbol{x}^-) \right]^\top \in \mathbb{R}^K.$$

Since Proposition 1 interprets PE scores as log-likelihood contributions rather than realizations of $\boldsymbol{Z}$ themselves, we do not directly identify b^ϕ_{ij} with Z_{ij}. Instead, we use the calibrated PE profile as a variational parameterization of the latent temporal-evidence posterior:

$$\boldsymbol{e}_{\phi,i}(\boldsymbol{x}^-) = \frac{\boldsymbol{b}_{\phi,i}(\boldsymbol{x}^-) - c\mathbf{1}_K}{a}, \qquad a > 0, \tag{54}$$

where $a > 0$ and $c \in \mathbb{R}$ are calibration constants aligning the PE-score representation with the latent temporal-evidence coordinates.

PE-induced variational posterior. We approximate the temporal evidence represented by PE with the multivariate Gaussian variational posterior

$$q_\phi\left(\boldsymbol{Z} \mid \boldsymbol{x}^-\right) = \mathcal{N}\left(\boldsymbol{\mu}_\phi(\boldsymbol{x}^-), \boldsymbol{K}_\phi(\boldsymbol{x}^-)\right). \tag{55}$$

Let $N_K = |\mathcal{Q}_K|$. We parameterize the posterior mean by the average lag-wise PE profile,

$$\boldsymbol{\mu}_\phi(\boldsymbol{x}^-) = \frac{1}{N_K} \sum_{i \in \mathcal{Q}_K} \boldsymbol{e}_{\phi,i}(\boldsymbol{x}^-), \tag{56}$$

so its τth component represents the average PE-induced temporal evidence associated with lag τ.

The posterior covariance is parameterized by the variation of these lag-wise PE profiles,

$$\boldsymbol{K}_\phi(\boldsymbol{x}^-) = \frac{1}{N_K} \sum_{i \in \mathcal{Q}_K} \left(\boldsymbol{e}_{\phi,i}(\boldsymbol{x}^-) - \boldsymbol{\mu}_\phi(\boldsymbol{x}^-)\right) \left(\boldsymbol{e}_{\phi,i}(\boldsymbol{x}^-) - \boldsymbol{\mu}_\phi(\boldsymbol{x}^-)\right)^\top + \epsilon_q \boldsymbol{I}_K, \tag{57}$$

where $\epsilon_q > 0$ ensures positive definiteness. Therefore, $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$ summarizes the average lag-wise temporal evidence encoded by PE together with its variation across query positions.

For a static PE, the positional scores b_{ij}^ϕ do not depend on the historical realization $\boldsymbol{x}^-$. Consequently, $\boldsymbol{\mu}_\phi(\boldsymbol{x}^-)$ and $\boldsymbol{K}_\phi(\boldsymbol{x}^-)$ are shared across input windows, yielding

$$q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-) = q_\phi(\boldsymbol{Z}). \tag{58}$$

In contrast, an adaptive PE produces input-dependent positional scores and therefore induces a window-specific variational posterior $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$.

Posterior–prior discrepancy. The GP-derived prior $p(\boldsymbol{Z} \mid \boldsymbol{x}^-)$ represents the temporal dependence supported by the underlying process, whereas $q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-)$ represents the temporal dependence encoded by PE. Since both distributions are Gaussian,

$$\begin{aligned} &D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{x}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{x}^-)\right) \\ &= \frac{1}{2}\Bigg[\operatorname{tr}\left(\boldsymbol{K}_0^{-1}\boldsymbol{K}_\phi(\boldsymbol{x}^-)\right) + \left(\boldsymbol{\mu}_\phi(\boldsymbol{x}^-) - \boldsymbol{\gamma}\right)^\top \boldsymbol{K}_0^{-1}\left(\boldsymbol{\mu}_\phi(\boldsymbol{x}^-) - \boldsymbol{\gamma}\right) \\ &\qquad - K + \log \frac{\det(\boldsymbol{K}_0)}{\det(\boldsymbol{K}_\phi(\boldsymbol{x}^-))} \Bigg]. \end{aligned} \tag{59}$$

Accordingly, the posterior–prior term in the population ELBO is

$$\mathbb{E}_{\boldsymbol{X}^-}\left[D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-) \,\|\, p(\boldsymbol{Z} \mid \boldsymbol{X}^-)\right)\right]. \tag{60}$$

This term quantifies the population-level discrepancy between the temporal structure represented by PE and that supported by the Gaussian process governing the historical input.

Conclusion of the proof. Combining Steps 1–5, the population-level variational formulation provides a genuine lower bound on the expected forecasting log-likelihood while preserving the input dependence of $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$. The GP governing the historical input determines the temporal-evidence prior $p(\boldsymbol{Z} \mid \boldsymbol{X}^-)$, whereas PE determines the variational posterior $q_\phi(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ through the first- and second-order statistics of its lag-wise positional profiles.

The forecasting component is characterized by the q_ϕ-weighted historical and future–history second-order moments in Equation 44. For static PE, the posterior is shared across input windows, so these weighted moments factor into the population covariance matrices $\boldsymbol{\Sigma}_-$ and $\boldsymbol{\Sigma}_{+-}$, yielding Equation 47. The posterior–prior term in Equation 60 further measures the discrepancy between the temporal structure encoded by PE and that supported by the historical process. Together, these results establish Theorem 2.

E Proof of Corollary 3

Proof. Conditional on regime $R = r$, the forecasting distribution

$$\mathcal{D}_r = p(\boldsymbol{X}^-, \boldsymbol{Y} \mid R = r)$$

is weakly stationary Gaussian by assumption. Therefore, Theorem 2 applies within each regime and induces the regime-specific population ELBO $\mathcal{L}_r(\phi)$, with temporal-evidence prior $p_r(\boldsymbol{Z}|\boldsymbol{X}^-)$ determined by the regime-specific autocovariance function $\Gamma_r(\tau)$.

Taking expectation over the regime variable R and using the law of total expectation gives

$$\mathcal{L}_{\mathrm{reg}}(\phi) = \sum_{r=1}^{J} \pi_r \mathcal{L}_r(\phi). \tag{61}$$

Thus, the population objective under regime-conditioned non-stationarity is a weighted combination of the regime-specific population ELBOs.

For any static PE $\phi \in \Phi_{\text{static}}$, we have, for every regime r,

$$\mathcal{L}_r(\phi) \leq \sup_{\phi_r \in \Phi} \mathcal{L}_r(\phi_r).$$

Multiplying by $\pi_r > 0$, summing over regimes, and then taking the supremum over $\phi \in \Phi_{\text{static}}$ yields

$$\sup_{\phi \in \Phi_{\text{static}}} \mathcal{L}_{\text{reg}}(\phi) \leq \sum_{r=1}^{J} \pi_r \sup_{\phi_r \in \Phi} \mathcal{L}_r(\phi_r). \tag{62}$$

To characterize when the inequality is strict, define

$$\mathcal{L}_r^\star := \sup_{\phi_r \in \Phi} \mathcal{L}_r(\phi_r)$$

and, for a static PE ϕ, its aggregate regime-wise optimality gap

$$\Delta(\phi) := \sum_{r=1}^{J} \pi_r \left[\mathcal{L}_r^\star - \mathcal{L}_r(\phi)\right] \geq 0.$$

Equation 62 is strict whenever

$$\inf_{\phi \in \Phi_{\text{static}}} \Delta(\phi) > 0.$$

Equivalently, because J is finite and $\pi_r > 0$, equality can hold only if there exists a sequence of static PE methods $\{\phi_n\} \subset \Phi_{\text{static}}$ satisfying

$$\mathcal{L}_r(\phi_n) \longrightarrow \mathcal{L}_r^\star, \qquad \forall r \in \{1, \ldots, J\}.$$

Hence, if no single sequence of static PE methods can simultaneously approach the regime-specific optimal ELBOs, the inequality in Equation 62 is strict.

For static PE, the induced temporal-evidence posterior $q_\phi(\boldsymbol{Z})$ is shared across regimes, whereas the GP-derived prior $p_r(\boldsymbol{Z} \mid \boldsymbol{X}^-)$ and the covariance structure entering $\mathcal{L}_r(\phi)$ may vary with the regime. Consequently, heterogeneous regimes can induce incompatible regime-specific ELBO objectives, forcing a static PE to compromise across them. An input-adaptive PE can instead vary its temporal evidence across forecasting windows, motivating the adaptive positional mechanism developed in this work. This establishes Corollary 3.

F The demonstration of Theorem 2

F.1 Binary-Regime Synthetic Dataset

We generate independent trajectories from two zero-mean Gaussian process (GP) regimes. For each trajectory, a regime $r \sim \text{Bernoulli}(1/2)$ is sampled once and held fixed throughout the trajectory. The regime-specific covariance functions are

$$k_{\text{smooth}}(t, t') = \exp\left(-\frac{(t-t')^2}{2(28)^2}\right), \tag{63}$$

$$k_{\text{periodic}}(t, t') = \exp\left(-\frac{2\sin^2\big(\pi(t-t')/24\big)}{(0.55)^2}\right). \tag{64}$$

Given the selected regime, we draw $f_{0:67} \sim \mathcal{GP}(0, k_r)$ and observe $x_t = f_t + \epsilon_t$, where $\epsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1^2)$. The forecasting input is $x_{0:63}$, and the four-step target is $x_{64:67}$.

We generate 256 training, 64 validation, and 64 test trajectories *per regime*, with independent trajectories across splits. Inputs and targets are standardized using the mean and standard deviation of the training inputs. Regime labels are retained for evaluation but are not provided to the forecasting models.

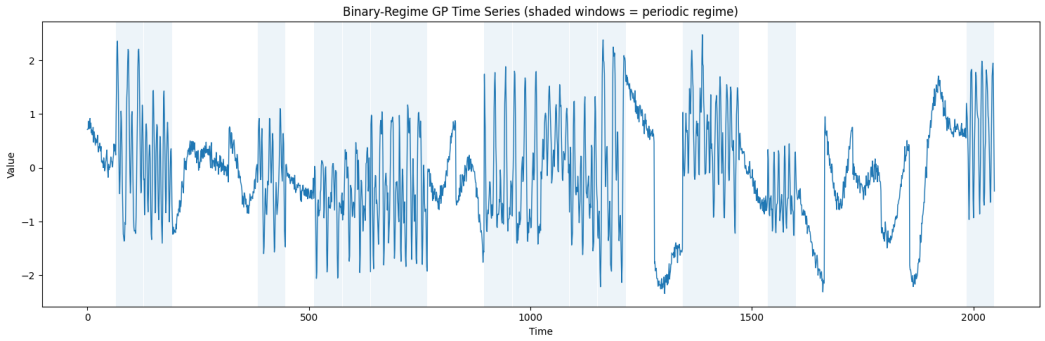


Figure 5: Synthetic binary-regime GP time series of 2,048 observations. Each 64-step window is generated from either a smooth RBF GP (unshaded) or a periodic GP (shaded). The periodic regime has period 24.

F.2 Simplified Transformer for Forecasting

We use a single-layer, single-head Transformer with model dimension $d_{\text{model}} = 32$. For a historical input $\boldsymbol{x}^- \in \mathbb{R}^{64}$, a point-wise linear layer embeds each observation, and sinusoidal absolute positional encoding (APE) is added in every PE configuration considered below. The encoder computes

$$\boldsymbol{A}_\phi(\boldsymbol{x}^-) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{32}} + \boldsymbol{B}_\phi(\boldsymbol{x}^-)\right), \qquad \boldsymbol{H}' = \boldsymbol{H} + \boldsymbol{W}_O(\boldsymbol{A}_\phi \boldsymbol{V}),$$

where the softmax is applied across keys and $\boldsymbol{B}_\phi$ denotes the contribution of RPE, ALiBi, RoPE, or BTA to the attention scores. For RoPE, this notation represents the difference between rotated and unrotated query–key scores; that contribution depends on the input representations. BTA generates an input-conditioned bias by combining RBF experts initialized with length scales $7, 14, 28, 42$ and periodic experts initialized with period 24 and length scales $0.35, 0.55, 0.9$.

The 64 encoder outputs are averaged and passed through a linear decoder to produce $\widehat{\boldsymbol{Y}} \in \mathbb{R}^4$. We train each model to minimize mean squared forecasting error using AdamW, a batch size of 64, a learning rate of 10^{-3}, weight decay of 10^{-4}, and at most 30 epochs. The best model is selected by validation MSE, with early stopping after seven epochs without improvement. For BTA, we additionally select the router width from $\{16, 32, 64, 128\}$ and the number of selected experts from $\{1, 3, 5, 7\}$, using validation data only. All methods use the same data splits and random seed.

We tain the simplified Transformer with four different PE methods including RPE (Liutkus et al., 2021), ALiBi (Press et al., 2021), RoPE (Su et al., 2024), and BTA on the binary-regime synthetic dataset. The performance of the simplified Transformer with the four different PE methods is summarized in Table 5 and the forecasting result given an input time series is shown in Figure 6.

Table 5: The forecasting performance on the binary-regime GP dataset. MSE is reported on standardized observations; lower is better. Smooth and periodic denote the two subsets of the test set.

PE Method	Best epoch	Val. MSE	Test MSE	Smooth MSE	Periodic MSE
RPE	30	0.2578	0.2337	0.1792	0.2882
RoPE	30	0.2364	0.2124	0.1354	0.2893
ALiBi	30	0.2573	0.2336	0.1801	0.2871
BTA	30	**0.2049**	**0.1788**	**0.0993**	**0.2583**

In addition, Figure 7 visualizes the positional bias induced by the four different PE methods.

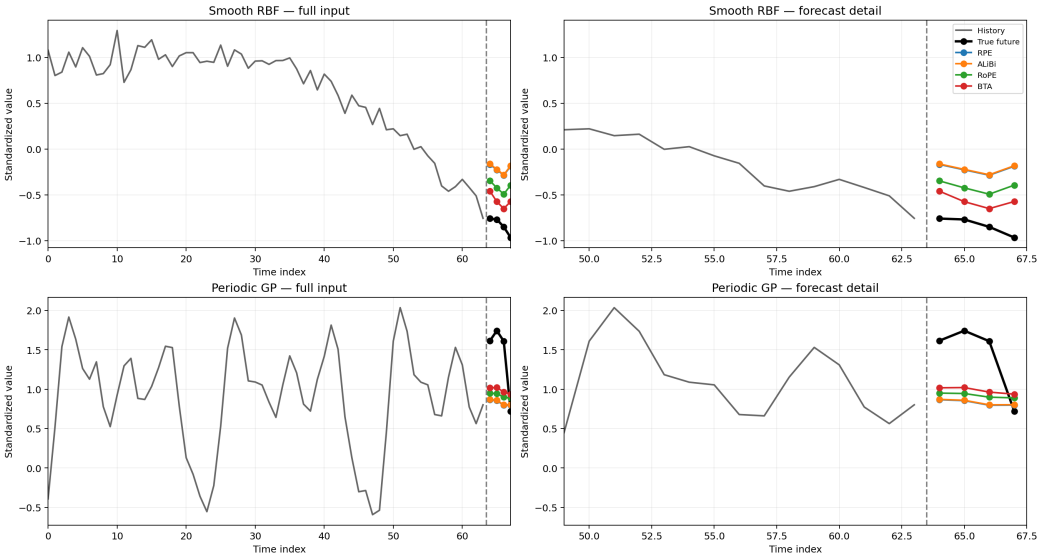


Figure 6: Forecasting results from the smooth RBF GP (top row) and periodic GP (bottom row). The left panels show all 64 historical observations and the four future values; the right panels magnify the forecast boundary. Gray lines denote the observed history, black markers the true future, and colored markers the predictions from the simplified Transformer with the four PE methods: RPE, ALiBi, RoPE, and BTA. The dashed line marks the end of the input window.

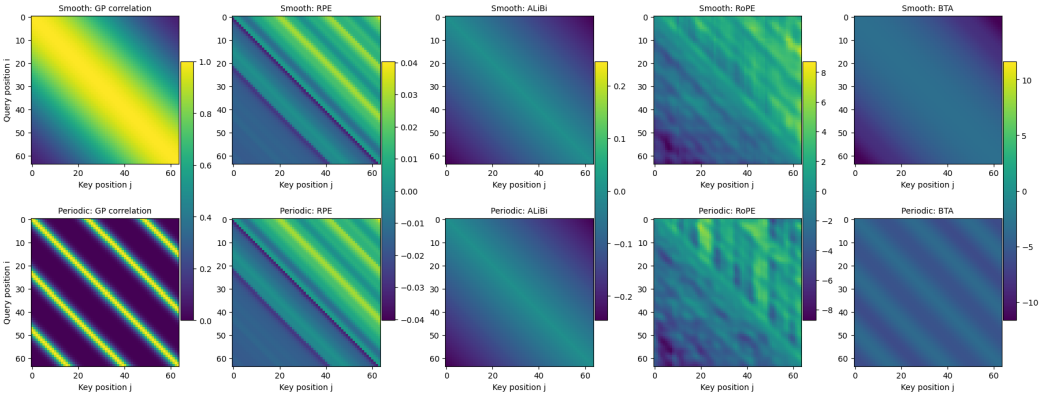


Figure 7: The figure compares positional biases under smooth and periodic Gaussian process regimes. RPE and ALiBi retain the same bias pattern across both regimes, while BTA produces different biases for the two inputs, illustrating its ability to adapt temporal evidence to the regime.

F.3 Synthetic Validation of the Population-Level ELBO

We design a binary-regime synthetic dataset to illustrate the population-level ELBO in Theorem 2 and examine the value of adaptive positional encoding (PE) when temporal dependencies change across forecasting windows. Each regime is generated by a zero-mean, weakly stationary Gaussian process (GP): the smooth regime uses an RBF covariance, while the periodic regime uses a periodic covariance with a 24-step cycle. Each model receives a 64-step history and predicts the next four observations. We train the same single-head Transformer with RPE, ALiBi, RoPE, or BTA; BTA uses its validation-selected router configuration.

For each trained model, we extract the positional score matrix $b_\phi(\boldsymbol{x}^-)$ and construct one-sided score profiles over lags $1, \ldots, K$. We set $K = 32$ for the 64-step input windows. Here K specifies the maximum lag used to estimate temporal evidence, not the input length: it leaves $T - K = 32$ query positions with complete one-sided lag profiles and includes the periodic regime's 24-step cycle. The

Table 6: Forecasting performance and estimated ELBO on the binary-regime synthetic test set with $\epsilon_q = 10^{-4}$. Lower MSE and higher ELBO are better.

PE	Test MSE	Expected log likelihood	Mean KL	ELBO
RPE	0.2337	-4.1431	348.0968	-352.2399
ALiBi	0.2336	-4.1429	271.2189	-275.3619
RoPE	0.2124	-4.1105	51.4227	-55.5333
BTA	**0.1869**	$\mathbf{-4.0492}$	**46.6581**	$\mathbf{-50.7073}$

forecaster continues to use the full 64-step history. A method-specific affine calibration is estimated from training windows and then held fixed on test windows. As in Appendix D, the mean and query-wise covariance of the calibrated profiles define

$$q_\phi(Z \mid \boldsymbol{x}^-) = \mathcal{N}\big(\boldsymbol{\mu}_\phi(\boldsymbol{x}^-), \boldsymbol{K}_\phi(\boldsymbol{x}^-)\big)\,.$$

We add $\epsilon_q I$ to the posterior covariance, with $\epsilon_q = 10^{-4}$. For each regime r, the GP autocovariance determines the mean and covariance of the lag-wise empirical autocovariance statistics, yielding the moment-matched temporal-evidence prior $p_r(Z) = \mathcal{N}(\boldsymbol{\gamma}_r, \boldsymbol{K}_{0,r})$.

We estimate the forecasting term by sampling $S = 16$ values from $q_\phi(Z \mid \boldsymbol{x}_n^-)$ for each test window. A sampled value perturbs the trained model's observed positional scores at the corresponding lags; the trained parameters and observed content scores remain fixed. At $Z = \boldsymbol{\mu}_\phi(\boldsymbol{x}_n^-)$, the perturbation vanishes and the original forecast is recovered. Using the same Gaussian observation variance $\sigma^2 = 1$ for every method, we compute

$$\widehat{\mathcal{L}}(\phi) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{S}\sum_{s=1}^{S}\log p_{\theta,\phi}\big(\boldsymbol{y}_n \mid \boldsymbol{x}_n^-, Z_{ns}\big) - D_{\mathrm{KL}}\big(q_\phi(Z \mid \boldsymbol{x}_n^-)\,\|\,p_{r_n}(Z)\big)\right], \quad Z_{ns} \sim q_\phi(Z \mid \boldsymbol{x}_n^-), \tag{65}$$

where r_n denotes the regime of window n. The first term measures forecasting fit under sampled temporal evidence; the second measures how closely the PE-induced evidence agrees with the window's regime-specific GP prior. Their average is a Monte Carlo estimate of the population-level variational objective for this trained, input-dependent perturbation model.

As shown in Table 6, BTA obtains both the lowest forecasting MSE and the highest estimated ELBO. Its expected log likelihood is higher and its posterior–prior KL is lower than those of the static PE methods. The result is consistent with the hypothesis underlying Theorem 2 and Corollary 3: when different windows have smooth or periodic temporal dependence, an adaptive PE can supply more appropriate temporal evidence than a shared static positional structure. Thus, the synthetic dataset provides an empirical illustration of the population-level ELBO framework and its motivation for adaptive PE in non-stationary forecasting. The reported values pertain to the explicitly defined input-dependent perturbation model in Eq. equation 65; they are not a direct numerical evaluation of the static covariance-reduced formula in Theorem 2.

G Data Description

We evaluate non-stationary LTSF models on eight widely used multivariate time-series benchmarks spanning diverse sampling frequencies, dimensionalities, and temporal dynamics. Table 7 summarizes their characteristics and forecasting protocols. Following established settings (Liu et al., 2025a), we use chronological train/validation/test splits and prediction horizons $H \in \{96, 192, 336, 720\}$. All data-dependent preprocessing parameters are estimated exclusively from the training set and applied unchanged to the validation and test sets. We additionally report the Augmented Dickey–Fuller (ADF) statistic as an indicator of stationarity: under a fixed test specification, a less negative ADF statistic indicates weaker evidence against the unit-root null and thus comparatively stronger non-stationary behavior.

- **ETT** (Zhou et al., 2021) contains transformer temperature and power-load measurements from two electricity-transformer stations in China between 2016 and 2018. It comprises two hourly datasets (ETTh1 and ETTh2) and two 15-minute datasets (ETTm1 and ETTm2), each with seven variables.
- **Weather** (Wu et al., 2023) contains 21 meteorological variables recorded in Germany at 10-minute intervals during 2020.
- **Electricity** (Wu et al., 2023) contains hourly electricity-consumption measurements from 321 customers between 2012 and 2014.
- **Traffic** (Wu et al., 2023) contains hourly freeway-occupancy measurements from 862 sensors in the San Francisco Bay Area during 2015–2016.
- **Solar-Energy** (Liu et al., 2024) contains 10-minute power-generation measurements from 137 photovoltaic plants in Alabama during 2006.

Table 7: Dataset characteristics and forecasting protocols. C denotes the number of variables. Training, Validation, and Testing columns denote the numbers of samples in the corresponding chronological partitions. Interval means the time elapsed between two consecutive time-series samples.

Dataset	C	Training	Validation	Testing	Interval	ADF stat.
ETTh1	7	8,545	2,881	2,881	1 hour	-5.91
ETTh2	7	8,545	2,881	2,881	1 hour	-4.13
ETTm1	7	34,465	11,521	11,521	15 min	-14.98
ETTm2	7	34,465	11,521	11,521	15 min	-5.66
Electricity	321	18,317	2,633	5,261	1 hour	-8.44
Traffic	862	12,185	1,757	3,509	1 hour	-15.02
Weather	21	36,792	5,271	10,540	10 min	-26.68
Solar-Energy	137	36,601	5,161	10,417	10 min	-37.23

H Transformer-based backbone architectures

To evaluate whether BTA generalizes across substantially different Transformer architectures rather than benefiting a particular backbone, we select the four Transformer-based backbone architectures because of the following reasons:

- Informer represents an early efficient Transformer for long-sequence forecasting;
- PatchTST represents the widely adopted patch-based architecture;
- TimeBridge is specifically designed to model non-stationary temporal dependencies;
- Timer-XL represents recent generative time-series foundation models.

Moreover, Table 8 summarizes the key difference between the four backbone architectures.

Table 8: Comparison of the Transformer backbones used in our experiments.

Backbone	Informer (Zhou et al., 2021)	PatchTST (Nie et al., 2023)	TimeBridge (Liu et al., 2025a)	Timer-XL (Liu et al., 2025c)
Representative role	Early LTSF Transformer	Patch-based Transformer	Non-stationary Transformer	Pretrained TSFM
Architecture	Encoder–decoder	Encoder-only	Encoder-only	Decoder-only
Attention design	ProbSparse	Self-attention	Integrated + PatchDownsampling + Cointegrated	Causal TimeAttention
Tokenization	Point-wise	Patch-wise	Patch-wise	Patch-wise
Positional mechanism	Sinusoidal absolute PE	Learnable absolute PE	None	Rotary PE + variable bias
Training paradigm	Task-specific	Task-specific	Task-specific	Pretraining / task-specific

I LTSF REFERENCE METHODS

To compare BTA with representative methods for non-stationary LTSF, we select ten models as baselines. They are divided into two groups. First, we include five methods specifically designed to address non-stationarity, distribution shifts, or heterogeneous temporal dynamics. Second, we include five widely used LTSF models spanning Transformer-, MLP-, convolution-, and linear-based architectures to evaluate BTA against representative general-purpose forecasting methods.

1. **Methods for non-stationary forecasting:**
   - **DTAF** (Lu et al., 2026) addresses non-stationarity in both temporal and frequency domains through temporal stabilization and frequency differencing.
   - **PULSE** (Liu et al., 2026) models evolving non-stationary dynamics through phase-anchored disentanglement, generative phase evolution, and statistic-aware augmentation.
   - **TimeBridge** (Liu et al., 2025a) employs Integrated and Cointegrated Attention to separately model stable short-term dependencies and non-stationary long-term cointegration.
   - **DUET** (Qiu et al., 2025) uses dual clustering over temporal distributions and channels to capture heterogeneous temporal patterns and dynamic inter-variable dependencies.
   - **Non-stationary Transformer** (Liu et al., 2022) combines Series Stationarization with De-stationary Attention to improve predictability while preserving informative non-stationary dependencies.
2. **General LTSF methods:**
   - **iTransformer** (Liu et al., 2024) treats individual variates as tokens and applies self-attention across variables to capture multivariate dependencies.
   - **TimeMixer** (Wang et al., 2024) employs multiscale decomposition and mixing to jointly model fine-scale seasonal patterns and coarse-scale trends.
   - **PatchTST** (Nie et al., 2023) represents time series as patch-level tokens and uses channel-independent Transformers to capture long-range temporal dependencies efficiently.
   - **TimesNet** (Wu et al., 2023) transforms one-dimensional time series into two-dimensional representations based on multiple periods to model intra- and inter-period temporal variations.
   - **DLinear** (Zeng et al., 2023) decomposes a time series into trend and seasonal components and forecasts them using simple linear mappings.

J More Details of BTA Implementation

This section provides additional implementation details of BTA, including backbone integration, temporal-kernel construction, MoE routing, hyperparameter selection, and training.

J.1 Backbone Integration

BTA is designed to augment rather than replace the native attention mechanism of each backbone. For the hth attention head of the nth input, the MoE router is constructed from the content-based attention scores

$$s_{ij}^{(n,h),\text{content}} = \frac{(\boldsymbol{q}_i^{(n,h)})^\top \boldsymbol{k}_j^{(n,h)}}{\sqrt{d_k}}.$$

The adaptive positional bias generated by BTA is then added to the backbone-specific native attention logits:

$$s_{ij}^{(n,h),\text{BTA}} = s_{ij}^{(n,h),\text{native}} + b_{ij}^{(n,h),\text{BTA}},$$

where $s_{ij}^{(n,h),\text{native}}$ includes the native positional mechanism of the backbone when applicable. The resulting logits are normalized by the original attention softmax.

We integrate BTA into Informer, PatchTST, TimeBridge, and Timer-XL through the same additive modification. These backbones differ substantially in tokenization, attention design, and positional mechanisms, but no other backbone component is modified. Thus, BTA preserves the original architecture while introducing an additional input- and head-adaptive temporal bias.

J.2 Temporal Kernel Bank

We construct an over-complete temporal kernel bank using radial basis function (RBF) and periodic (PER) kernels to represent complementary local, long-range, and seasonal dependencies. Let $\tau_{ij} = |i - j|$ denote the relative token lag. For point-wise models, τ_{ij} is measured in time steps, whereas for patch-based models it is measured in patches.

RBF kernels. The RBF log-kernel is

$$\psi_r^{\text{RBF}}(\tau_{ij}; \ell_r) = -\frac{\tau_{ij}^2}{2\ell_r^2}, \qquad r = 1, \ldots, 5,$$

where ℓ_r controls the temporal scale. We parameterize each RBF scale relative to the available input context,

$$\ell_r = \rho_r L, \qquad \rho_r \in \mathcal{R}_{\text{RBF}},$$

where L denotes the number of input tokens or patches and $\mathcal{R}_{\text{RBF}}$ contains five fixed scale coefficients. This normalization allows the same kernel construction to be used across backbones with different input lengths and tokenization schemes.

Periodic kernels. To model recurring temporal structure, we first identify the two dominant periods

$$p_e \in \{p_1, p_2\}, \qquad e \in \{1, 2\},$$

using Fourier analysis of the training data only. For each dominant period, we instantiate three periodic kernels with different peak widths:

$$\psi_{r,e}^{\text{PER}}(\tau_{ij}; \eta_r, p_e) = -\frac{2\sin^2(\pi\tau_{ij}/p_e)}{\eta_r^2}, \qquad r = 1, 2, 3, \quad e = 1, 2,$$

where η_r is a dimensionless width parameter. The corresponding temporal peak width can equivalently be expressed as

$$\ell'_{r,e} = \eta_r p_e.$$

Thus, the periodic kernel bank contains $2 \times 3 = 6$ candidate kernels, spanning two dominant periods and three temporal widths for each period.

The complete kernel bank therefore contains

$$M = 5 \text{ RBF kernels} + 6 \text{ periodic kernels} = 11$$

candidate temporal patterns. All kernel parameters are fixed before model training; BTA learns only how to combine these candidate kernels for each input and attention head.

J.3 MoE Router Configuration

For each input n and attention head h, we aggregate the content-based attention scores associated with the same relative lag to construct the one-dimensional temporal-evidence (TE) routing profile

$$e_\tau^{(n,h)} = \frac{1}{|\mathcal{I}_\tau|} \sum_{(i,j)\in\mathcal{I}_\tau} s_{ij}^{(n,h),\text{content}}, \qquad \mathcal{I}_\tau = \{(i,j) : |i-j| = \tau\}.$$

This profile summarizes how strongly the learned content representations associate query–key pairs at different temporal lags.

To capture complementary lag-domain and frequency-domain information, the router receives

$$\left[\boldsymbol{e}^{(n,h)}; \left|\mathcal{F}\left(\boldsymbol{e}^{(n,h)}\right)\right|\right],$$

where $\mathcal{F}(\cdot)$ denotes the discrete Fourier transform and $|\mathcal{F}(\cdot)|$ denotes its magnitude spectrum.

The router is a one-hidden-layer MLP with $M = 11$ outputs, corresponding to the five RBF and six periodic candidate kernels. Let $\boldsymbol{g}^{(n,h)} \in \mathbb{R}^M$ denote its output logits. We retain the Top-k logits and apply softmax over the selected experts, setting the remaining mixture weights to zero. This yields

$$\boldsymbol{\omega}^{(n,h)} = [\omega_1^{(n,h)}, \ldots, \omega_M^{(n,h)}], \qquad \sum_{c=1}^{M} \omega_c^{(n,h)} = 1.$$

The resulting input- and head-specific mixture weights are used in Equation 7 to construct the adaptive positional bias.

J.4 Hyperparameter Configuration

The principal BTA hyperparameters are the router hidden dimension d_r and the number of retained experts k. We search

$$k \in \{1, 3, 5, 7\}$$

and use backbone-specific search grids for d_r, as summarized in Table 9.

Each router-dimension grid is represented by $(d_{\text{start}}, d_{\text{ub}}, \Delta d)$ and contains candidate values

$$d_r = d_{\text{start}} + m\Delta d, \qquad m \in \mathbb{Z}_{\geq 0}, \qquad d_r \leq d_{\text{ub}},$$

where d_{ub} denotes an upper bound rather than necessarily an included endpoint. Hyperparameters are selected according to validation-set forecasting performance and fixed before test-set evaluation.

J.5 Training and Optimization

For each backbone, BTA follows the corresponding vanilla training protocol. The temporal kernels remain fixed throughout training, whereas the MoE router and trainable backbone parameters are optimized jointly. Informer, PatchTST, and TimeBridge are trained task-specifically. Timer-XL is initialized from its pretrained checkpoint and adapted to each forecasting dataset through full-shot fine-tuning.

For each experimental run, the random seed is fixed for reproducibility. The robustness analysis in Table 11 additionally repeats each configuration across three predefined random seeds and reports the mean and sample standard deviation.

Table 9: BTA configuration across the four Transformer backbones. Router hidden-dimension search ranges are reported as $(d_{\text{start}}, d_{\text{ub}}, \Delta d)$, where d_{ub} is the search upper bound and Δd is the step size.

Configuration	Informer	PatchTST	TimeBridge	Timer-XL
Token level	Point-wise	Patch-wise	Patch-wise	Patch-wise
Sequence length	96	96	720	672
Patch length	1	16	24 or 48	96
Patch numbers	96	4	30 or 15	7
Lag unit	Time step	Patch	Patch	Patch
Kernel families	RBF + Periodic	RBF + Periodic	RBF + Periodic	RBF + Periodic
RBF kernels	5	5	5	5
Dominant periods	2	2	2	2
Periodic widths per period	3	3	3	3
Periodic kernels	6	6	6	6
Total kernels M	11	11	11	11
Router architecture	1-layer MLP	1-layer MLP	1-layer MLP	1-layer MLP
Top-k			$\{1, 3, 5, 7\}$	
Router hidden-dimension grid $(d_{\text{start}}, d_{\text{ub}}, \Delta d)$	$(32, 1024, 64)$	$(32, 512, 32)$	$(32, 512, 32)$	$(32, 2048, 64)$
Optimization	Task-specific training	Task-specific training	Task-specific training	Pretrained + full-shot fine-tuning

K Main Experiment Appendix

BTA generalizes across different Transformer backbones. Table 10 reports the per-horizon results underlying Table 1. Across three backbones with different tokenization and attention designs, BTA consistently improves the vanilla attention mechanism while leaving the remaining backbone architecture unchanged. Averaged over the four forecasting horizons, BTA reduces MSE by 5.79–14.05% for Informer, 4.34–17.51% for PatchTST, and 1.74–11.80% for Timer-XL. Across all 96 backbone–dataset–horizon–metric settings, BTA improves 95, with the only exception being the MAE of Informer on ETTm1 at $H = 192$ (-0.49%). These consistent gains across heterogeneous architectures demonstrate the broad applicability of BTA as a plug-and-play attention enhancement.

Table 10: **Per-horizon forecasting results of BTA across three Transformer backbones.** Results are reported for $H \in \{96, 192, 336, 720\}$. Within each backbone and row, the lower MSE and MAE between Vanilla and BTA are shown in **bold**. *Avg.* denotes the mean over the four horizons, and Gain (%) is the relative error reduction $100 \times (\text{Vanilla} - \text{BTA})/\text{Vanilla}$, computed from unrounded values for the *Avg.* rows. Backbone configurations are summarized in Table 9.

Model		**Informer** (Zhou et al., 2021)					**PatchTST** (Nie et al., 2023)					**Timer-XL** (Liu et al., 2025c)				
Self-Attention		Vanilla		BTA		Gain (%)		Vanilla		BTA		Gain (%)		Vanilla		BTA		Gain (%)	
Dataset		MSE	MAE	MSE	MAE	ΔMSE	ΔMAE	MSE	MAE	MSE	MAE	ΔMSE	ΔMAE	MSE	MAE	MSE	MAE	ΔMSE	ΔMAE
ETTh1	96	0.992	0.785	**0.878**	**0.719**	11.49	8.41	0.460	0.447	**0.367**	**0.394**	20.22	11.86	0.364	0.397	**0.343**	**0.379**	5.77	4.53
	192	1.008	0.801	**0.895**	**0.726**	11.21	9.36	0.512	0.447	**0.423**	**0.425**	17.38	4.92	0.405	0.424	**0.376**	**0.402**	7.16	5.19
	336	1.179	0.870	**1.038**	**0.793**	11.96	8.85	0.546	0.496	**0.441**	**0.436**	19.23	12.10	0.427	0.439	**0.391**	**0.413**	8.43	5.92
	720	1.204	0.882	**1.112**	**0.837**	7.64	5.10	0.544	0.517	**0.470**	**0.461**	13.60	10.83	0.439	0.459	**0.406**	**0.433**	7.52	5.66
	Avg.	1.096	0.835	**0.981**	**0.769**	10.50	7.88	0.516	0.477	**0.425**	**0.429**	17.51	10.02	0.409	0.430	**0.379**	**0.407**	7.28	5.35
ETTm1	96	0.672	0.614	**0.586**	**0.557**	12.80	9.28	0.352	0.374	**0.315**	**0.355**	10.51	5.08	0.290	0.341	**0.284**	**0.337**	2.07	1.17
	192	0.692	**0.610**	**0.671**	0.613	3.03	-0.49	0.390	0.393	**0.358**	**0.381**	8.21	3.05	0.337	0.369	**0.332**	**0.367**	1.48	0.54
	336	0.942	0.749	**0.831**	**0.685**	11.78	8.54	0.421	0.414	**0.373**	**0.401**	11.40	3.14	0.374	0.392	**0.367**	**0.390**	1.87	0.51
	720	0.998	0.787	**0.936**	**0.754**	6.21	4.19	0.462	0.449	**0.436**	**0.443**	5.63	1.34	0.437	0.428	**0.430**	**0.426**	1.60	0.47
	Avg.	0.826	0.690	**0.756**	**0.652**	8.47	5.47	0.406	0.408	**0.371**	**0.395**	8.80	3.07	0.360	0.383	**0.353**	**0.380**	1.74	0.65
Weather	96	0.313	0.375	**0.253**	**0.335**	19.17	10.67	0.186	0.227	**0.170**	**0.212**	8.60	6.61	0.157	0.205	**0.153**	**0.197**	2.55	3.90
	192	0.377	0.408	**0.299**	**0.369**	20.69	9.56	0.234	0.265	**0.219**	**0.256**	6.41	3.40	0.206	0.250	**0.200**	**0.243**	2.91	2.80
	336	0.448	0.449	**0.389**	**0.413**	13.17	8.02	0.284	0.301	**0.274**	**0.295**	3.52	1.99	0.259	0.291	**0.255**	**0.285**	1.54	2.06
	720	0.712	0.604	**0.649**	**0.577**	8.85	4.47	0.356	0.349	**0.351**	**0.347**	1.40	0.57	0.337	0.344	**0.332**	**0.338**	1.48	1.74
	Avg.	0.463	0.459	**0.398**	**0.424**	14.05	7.73	0.265	0.286	**0.254**	**0.278**	4.34	2.80	0.240	0.273	**0.235**	**0.266**	1.98	2.48
Solar	96	0.200	0.253	**0.191**	**0.249**	4.50	1.58	0.234	0.286	**0.208**	**0.249**	11.11	12.94	0.174	0.215	**0.166**	**0.214**	4.60	0.47
	192	0.240	0.273	**0.224**	**0.260**	6.67	4.76	0.267	0.310	**0.241**	**0.272**	9.74	12.26	0.199	0.234	**0.187**	**0.232**	6.03	0.85
	336	0.244	0.273	**0.234**	**0.262**	4.10	4.03	0.290	0.315	**0.258**	**0.284**	11.03	9.84	0.225	0.255	**0.200**	**0.248**	11.11	2.75
	720	0.248	0.274	**0.229**	**0.271**	7.66	1.09	0.289	0.317	**0.257**	**0.282**	11.07	11.04	0.275	0.285	**0.217**	**0.273**	21.09	4.21
	Avg.	0.233	0.268	**0.220**	**0.261**	5.79	2.89	0.270	0.307	**0.241**	**0.272**	10.74	11.48	0.218	0.247	**0.193**	**0.242**	11.80	2.22

To examine the robustness of BTA, we train the three backbones integrated with BTA across the four datasets and four different forecasting horizons. Table 11 shows that BTA generally achieves lower mean MSE than vanilla self-attention. The multi-seed results also show limited variation across random initializations, supporting the robustness of BTA's forecasting performance.

Table 11: **Forecasting MSE across three Transformer backbones and three random seeds.** Results are reported as mean ± sample standard deviation for $H \in \{96, 192, 336, 720\}$. Within each backbone and row, the method with the lower mean MSE has both its mean and standard deviation shown in bold.

Model		**Informer** (Zhou et al., 2021)		**PatchTST** (Nie et al., 2023)		**Timer-XL** (Liu et al., 2025c)	
Dataset	**Horizon**	**Vanilla MSE**	**BTA MSE**	**Vanilla MSE**	**BTA MSE**	**Vanilla MSE**	**BTA MSE**
ETTh1	96	0.994 ± 0.003	$\mathbf{0.876 \pm 0.004}$	0.462 ± 0.005	$\mathbf{0.366 \pm 0.004}$	0.366 ± 0.005	$\mathbf{0.345 \pm 0.003}$
	192	1.012 ± 0.012	$\mathbf{0.895 \pm 0.003}$	0.511 ± 0.004	$\mathbf{0.422 \pm 0.006}$	0.404 ± 0.011	$\mathbf{0.379 \pm 0.006}$
	336	1.174 ± 0.009	$\mathbf{1.036 \pm 0.020}$	0.548 ± 0.016	$\mathbf{0.447 \pm 0.012}$	0.422 ± 0.018	$\mathbf{0.397 \pm 0.010}$
	720	1.204 ± 0.001	$\mathbf{1.111 \pm 0.006}$	0.543 ± 0.005	$\mathbf{0.470 \pm 0.002}$	0.437 ± 0.006	$\mathbf{0.406 \pm 0.003}$
ETTm1	96	0.672 ± 0.001	$\mathbf{0.586 \pm 0.001}$	0.352 ± 0.001	$\mathbf{0.315 \pm 0.001}$	0.290 ± 0.001	$\mathbf{0.283 \pm 0.001}$
	192	0.693 ± 0.003	$\mathbf{0.669 \pm 0.005}$	0.391 ± 0.005	$\mathbf{0.359 \pm 0.002}$	0.337 ± 0.004	$\mathbf{0.332 \pm 0.002}$
	336	0.940 ± 0.008	$\mathbf{0.831 \pm 0.001}$	0.421 ± 0.002	$\mathbf{0.373 \pm 0.000}$	0.372 ± 0.004	$\mathbf{0.371 \pm 0.003}$
	720	0.998 ± 0.007	$\mathbf{0.937 \pm 0.007}$	0.460 ± 0.013	$\mathbf{0.436 \pm 0.001}$	0.440 ± 0.010	$\mathbf{0.435 \pm 0.004}$
Weather	96	0.315 ± 0.002	$\mathbf{0.254 \pm 0.002}$	0.184 ± 0.002	$\mathbf{0.172 \pm 0.002}$	0.157 ± 0.002	$\mathbf{0.150 \pm 0.003}$
	192	0.378 ± 0.001	$\mathbf{0.294 \pm 0.005}$	0.233 ± 0.001	$\mathbf{0.222 \pm 0.004}$	0.206 ± 0.006	$\mathbf{0.197 \pm 0.003}$
	336	0.449 ± 0.001	$\mathbf{0.389 \pm 0.001}$	0.282 ± 0.003	$\mathbf{0.275 \pm 0.003}$	0.261 ± 0.002	$\mathbf{0.251 \pm 0.004}$
	720	0.712 ± 0.004	$\mathbf{0.651 \pm 0.002}$	0.355 ± 0.002	$\mathbf{0.352 \pm 0.001}$	0.336 ± 0.004	$\mathbf{0.327 \pm 0.006}$
Solar	96	0.200 ± 0.000	$\mathbf{0.191 \pm 0.000}$	0.234 ± 0.000	$\mathbf{0.208 \pm 0.000}$	0.174 ± 0.000	$\mathbf{0.166 \pm 0.000}$
	192	0.240 ± 0.000	$\mathbf{0.225 \pm 0.001}$	0.268 ± 0.001	$\mathbf{0.242 \pm 0.001}$	0.200 ± 0.001	$\mathbf{0.186 \pm 0.001}$
	336	0.244 ± 0.001	$\mathbf{0.234 \pm 0.000}$	0.290 ± 0.001	$\mathbf{0.258 \pm 0.000}$	0.225 ± 0.000	$\mathbf{0.201 \pm 0.001}$
	720	0.248 ± 0.001	$\mathbf{0.228 \pm 0.001}$	0.289 ± 0.000	$\mathbf{0.256 \pm 0.001}$	0.275 ± 0.001	$\mathbf{0.218 \pm 0.001}$

BTA Improves TimeBridge for Non-stationary LTSF. Table 12 compares TimeBridge+BTA (*Ours*) with the original TimeBridge and other representative LTSF methods under the same input length $T = 720$. Integrating BTA consistently improves TimeBridge across all eight datasets, reducing the overall average MSE and MAE by 2.30% and 1.89%, respectively. Moreover, TimeBridge+BTA achieves the lowest MSE in 29 of 32 horizon-level settings and the lowest MAE in 23, substantially exceeding all competing methods. These results demonstrate that BTA provides a consistent additional benefit to an already competitive non-stationary forecasting backbone.

Table 12: **Per-horizon comparison for non-stationary LTSF.** *Ours* denotes TimeBridge+BTA, enabling direct comparison with the original TimeBridge to isolate the contribution of BTA. All methods use input length $T = 720$ and prediction horizons $H \in \{96, 192, 336, 720\}$. *Avg.* denotes the mean over the four horizons. The best and second-best results in each row are shown in **bold** and underlined, respectively. *1st Count* reports the number of horizon-level settings in which each method achieves the best result, excluding *Avg.* rows and counting ties for all tied methods.

Model		Ours		PULSE (Liu et al., 2026)		DTAF (Lu et al., 2026)		DUET (Qiu et al., 2025)		TimeBridge (Liu et al., 2025a)		iTransformer (Liu et al., 2024)		TimeMixer (Wang et al., 2024)		PatchTST (Nie et al., 2023)		TimesNet (Wu et al., 2023)		DLinear (Zeng et al., 2023)		Stationary (Liu et al., 2022)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.275**	0.330	0.303	0.347	0.277	**0.329**	0.279	0.336	0.284	0.337	0.300	0.353	0.293	0.345	0.293	0.346	0.338	0.375	0.299	0.343	0.393	0.404
	192	**0.315**	**0.359**	0.349	0.379	0.329	0.361	0.318	**0.359**	0.323	0.365	0.345	0.382	0.335	0.372	0.333	0.370	0.371	0.387	0.335	0.365	0.481	0.447
	336	0.351	0.385	0.375	0.398	**0.345**	**0.374**	**0.345**	0.378	0.370	0.394	0.374	0.398	0.368	0.386	0.369	0.390	0.410	0.411	0.369	0.386	0.555	0.489
	720	0.408	0.411	0.424	0.426	**0.399**	**0.407**	0.400	0.408	0.415	0.418	0.429	0.430	0.426	0.417	0.416	0.420	0.478	0.450	0.425	0.421	0.677	0.548
	Avg.	0.337	0.371	0.363	0.388	0.338	**0.368**	**0.336**	0.370	0.348	0.379	0.362	0.391	0.356	0.380	0.353	0.381	0.399	0.406	0.357	0.379	0.527	0.472
ETTm2	96	**0.155**	**0.242**	0.163	0.248	0.160	0.245	0.164	0.253	0.157	0.243	0.175	0.266	0.165	0.256	0.166	0.256	0.187	0.267	0.167	0.260	0.266	0.324
	192	**0.215**	**0.283**	0.227	0.291	**0.215**	0.285	0.220	0.293	0.224	0.286	0.242	0.312	0.225	0.298	0.223	0.296	0.249	0.309	0.224	0.303	0.434	0.405
	336	**0.266**	**0.320**	0.283	0.330	0.269	0.321	0.277	0.331	0.269	0.321	0.282	0.340	0.277	0.332	0.274	0.329	0.321	0.351	0.281	0.342	0.808	0.563
	720	**0.346**	**0.375**	0.374	0.387	0.351	0.378	0.358	0.386	0.348	0.378	0.378	0.398	0.360	0.387	0.362	0.385	0.497	0.403	0.397	0.421	0.758	0.555
	Avg.	**0.246**	**0.305**	0.262	0.314	0.249	0.307	0.255	0.316	0.250	0.307	0.269	0.329	0.257	0.318	0.256	0.317	0.314	0.333	0.267	0.332	0.567	0.462
ETTh1	96	**0.347**	**0.385**	0.362	0.390	0.369	0.396	0.358	0.391	0.354	0.390	0.386	0.405	0.372	0.401	0.370	0.400	0.384	0.402	0.375	0.399	0.547	0.501
	192	**0.386**	**0.414**	0.408	0.418	0.404	0.419	0.399	0.416	0.389	**0.414**	0.424	0.440	0.413	0.430	0.413	0.429	0.557	0.436	0.405	0.416	0.603	0.529
	336	**0.404**	**0.426**	0.443	0.433	0.449	0.453	0.418	0.431	0.408	0.430	0.449	0.460	0.438	0.450	0.422	0.440	0.491	0.469	0.439	0.443	0.711	0.585
	720	0.437	0.465	**0.433**	**0.448**	0.441	0.471	0.441	0.471	0.443	0.468	0.495	0.487	0.486	0.484	0.447	0.468	0.521	0.500	0.472	0.490	0.714	0.601
	Avg.	**0.394**	0.423	0.412	**0.422**	0.416	0.435	0.404	0.427	0.399	0.426	0.439	0.448	0.427	0.441	0.413	0.434	0.488	0.452	0.423	0.437	0.644	0.554
ETTh2	96	**0.267**	**0.329**	0.277	0.330	0.271	**0.329**	0.273	0.339	0.271	0.331	0.297	0.348	0.281	0.351	0.274	0.337	0.340	0.374	0.289	0.353	0.392	0.420
	192	**0.332**	**0.371**	0.356	0.381	0.333	**0.371**	0.333	0.376	0.335	0.372	0.371	0.403	0.349	0.387	0.348	0.384	0.402	0.414	0.383	0.418	0.519	0.480
	336	**0.360**	**0.398**	0.399	0.417	0.368	0.402	0.372	0.411	0.371	0.402	0.404	0.428	0.366	0.413	0.377	0.416	0.452	0.452	0.448	0.465	0.572	0.511
	720	**0.392**	**0.429**	0.413	0.436	0.480	0.482	0.398	0.435	0.399	0.432	0.424	0.444	0.401	0.436	0.406	0.441	0.462	0.468	0.605	0.551	0.600	0.536
	Avg.	**0.338**	**0.382**	0.361	0.391	0.363	0.396	0.344	0.390	0.344	0.384	0.374	0.406	0.349	0.397	0.351	0.395	0.414	0.427	0.431	0.447	0.521	0.487
Weather	96	**0.142**	**0.182**	0.150	0.196	0.145	0.185	0.149	0.195	0.144	0.184	0.159	0.208	0.147	0.198	0.149	0.198	0.172	0.220	0.152	0.237	0.183	0.230
	192	**0.185**	**0.225**	0.200	0.245	0.190	0.228	0.194	0.239	0.187	**0.225**	0.200	0.248	0.192	0.243	0.194	0.241	0.219	0.261	0.220	0.282	0.253	0.291
	336	**0.237**	**0.267**	0.260	0.291	0.241	0.269	0.254	0.285	0.239	0.268	0.253	0.289	0.247	0.284	0.245	0.282	0.280	0.306	0.265	0.319	0.326	0.341
	720	**0.309**	**0.319**	0.347	0.353	0.313	0.321	0.324	0.336	0.311	0.321	0.321	0.338	0.318	0.330	0.314	0.334	0.365	0.359	0.323	0.362	0.417	0.397
	Avg.	**0.218**	**0.248**	0.239	0.271	0.222	0.251	0.230	0.264	0.220	0.250	0.233	0.271	0.226	0.264	0.226	0.264	0.259	0.287	0.240	0.300	0.295	0.315
ELC	96	**0.124**	0.217	0.134	0.228	0.129	0.221	0.131	0.224	**0.124**	**0.216**	0.138	0.237	0.153	0.256	0.129	0.222	0.168	0.272	0.140	0.237	0.172	0.276
	192	**0.144**	0.237	0.150	0.243	0.146	**0.235**	0.156	0.247	0.145	0.241	0.157	0.256	0.168	0.269	0.147	0.240	0.184	0.289	0.152	0.249	0.185	0.287
	336	**0.158**	**0.252**	0.166	0.261	0.165	0.254	0.164	0.257	0.166	0.263	0.167	0.264	0.189	0.291	0.163	0.259	0.198	0.300	0.169	0.267	0.196	0.300
	720	**0.180**	**0.272**	0.188	0.289	0.200	0.283	0.195	0.285	0.182	0.277	0.194	0.286	0.228	0.320	0.197	0.290	0.220	0.320	0.203	0.301	0.221	0.319
	Avg.	**0.152**	**0.245**	0.160	0.255	0.160	0.248	0.162	0.253	0.154	0.249	0.164	0.261	0.185	0.284	0.159	0.253	0.193	0.295	0.166	0.264	0.194	0.296
Traffic	96	**0.318**	**0.225**	0.414	0.268	0.370	0.232	0.351	0.237	0.340	0.240	0.363	0.265	0.369	0.257	0.360	0.249	0.593	0.321	0.410	0.282	0.612	0.338
	192	**0.336**	**0.237**	0.444	0.278	0.392	0.243	0.368	0.246	0.343	0.250	0.385	0.273	0.400	0.272	0.379	0.256	0.617	0.336	0.423	0.287	0.613	0.340
	336	**0.348**	**0.245**	0.458	0.287	0.408	0.252	0.393	0.260	0.363	0.257	0.396	0.277	0.407	0.272	0.392	0.264	0.629	0.336	0.436	0.296	0.618	0.328
	720	**0.382**	**0.260**	0.525	0.312	0.441	0.269	0.430	0.272	0.395	0.263	0.445	0.312	0.461	0.316	0.432	0.286	0.640	0.350	0.466	0.315	0.653	0.355
	Avg.	**0.346**	**0.242**	0.460	0.286	0.403	0.249	0.386	0.254	0.360	0.253	0.397	0.282	0.409	0.279	0.391	0.264	0.620	0.336	0.434	0.295	0.624	0.340
Solar	96	**0.150**	0.198	0.176	0.229	0.182	0.218	0.175	**0.197**	0.161	0.224	0.188	0.242	0.179	0.232	0.178	0.229	0.219	0.314	0.216	0.287	0.381	0.398
	192	**0.171**	0.211	0.192	0.245	0.182	0.235	0.193	**0.210**	0.177	0.237	0.193	0.258	0.201	0.259	0.189	0.246	0.231	0.322	0.244	0.305	0.395	0.386
	336	**0.184**	**0.226**	0.200	0.247	0.189	0.229	0.197	0.230	0.196	0.232	0.195	0.259	0.190	0.256	0.198	0.249	0.246	0.337	0.263	0.319	0.410	0.394
	720	**0.197**	0.235	0.208	0.249	0.228	0.256	0.210	**0.215**	0.202	0.239	0.223	0.281	0.203	0.261	0.209	0.256	0.280	0.363	0.264	0.324	0.377	0.376
	Avg.	**0.176**	0.218	0.194	0.243	0.195	0.235	0.194	**0.213**	0.184	0.233	0.200	0.260	0.193	0.252	0.194	0.245	0.244	0.334	0.247	0.309	0.391	0.389
1st Count		29	23	1	1	3	6	1	4	1	3	0	0	0	0	0	0	0	0	0	0	0	0

L MORE CASE STUDIES

Figures 8, 9, and 10 further illustrate how BTA adapts positional biases to heterogeneous temporal dynamics. On Solar, both windows exhibit strong periodic structure, but BTA selects different temporal scales: irregular peak amplitudes lead to a combination of broader periodic and long-range local kernels, whereas more regularly recurring peaks favor narrower periodic kernels. On Weather, BTA adapts primarily across kernel families, emphasizing a long-range local kernel for a window with a recent level shift and a 3-day periodic kernel for a window with recurring temporal structure. A similar behavior is observed with Timer-XL: an abrupt recent change favors short-range local kernels, while a more stable window assigns greater weight to broader local and periodic dependencies. These input-specific kernel mixtures induce distinct temporal-evidence profiles and hence adaptive positional biases, demonstrating how BTA enables Transformer forecasters to adjust their temporal dependencies to non-stationary input dynamics.

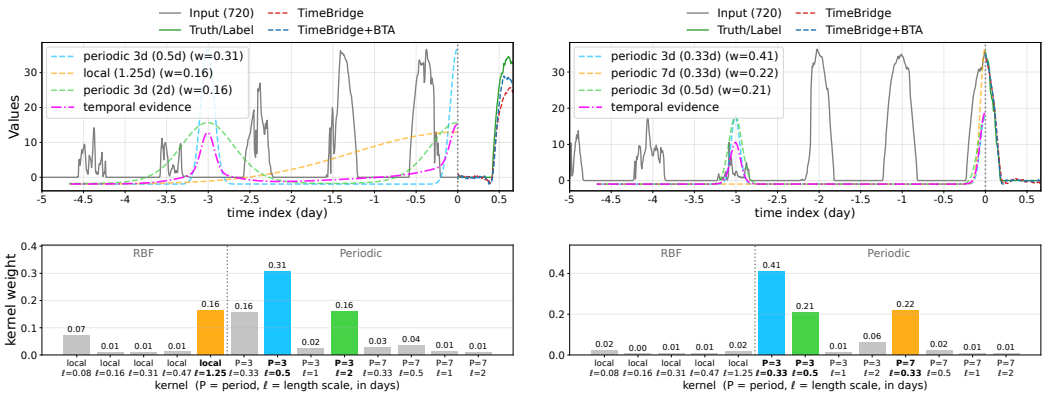


Figure 8: **Input-adaptive positional bias on Solar.** Two Solar windows with a 720-step input and 96-step forecasting horizon; panels follow Figure 2. Both windows are dominated by periodic kernels, but BTA adapts their temporal scales to the input dynamics. The left window, with more irregular peak amplitudes, combines a 3-day periodic kernel with $\ell = 0.5$ days ($w = 0.31$), a long-range local kernel with $\ell = 1.25$ days ($w = 0.16$), and a broader 3-day periodic kernel with $\ell = 2$ days ($w = 0.16$). In contrast, the right window concentrates its weight on narrower periodic kernels, led by 3-day kernels with $\ell = 0.33$ and 0.5 days ($w = 0.41$ and 0.21) and a 7-day kernel with $\ell = 0.33$ days ($w = 0.22$). The resulting mixtures induce distinct temporal-evidence profiles and positional biases. BTA reduces MSE by 58% and 34%, respectively ($22.48 \rightarrow 9.41$ and $1.46 \rightarrow 0.96$).

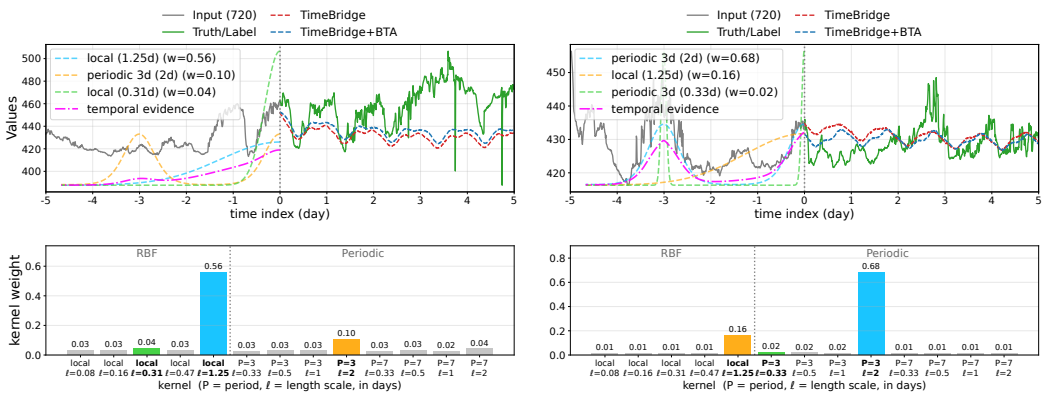


Figure 9: **Input-adaptive positional bias on Weather.** Two Weather windows with a 720-step input and 720-step forecasting horizon; panels follow Figure 2. The left window exhibits a recent level shift without a clear recurring pattern, for which BTA assigns 0.69 of the total weight to local RBF kernels, dominated by the long-range $\ell = 1.25$-day kernel ($w = 0.56$). In contrast, the right window exhibits stronger recurring structure and is dominated by a 3-day periodic kernel with $\ell = 2$ days ($w = 0.68$), while the long-range local kernel receives $w = 0.16$. The contrasting kernel mixtures produce different temporal-evidence profiles and positional biases, illustrating how BTA adapts between local and periodic dependencies across input windows.

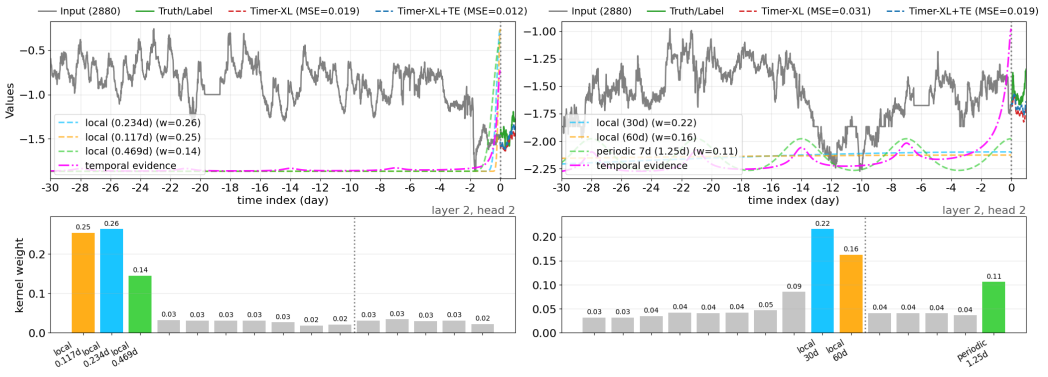


Figure 10: **Input-adaptive positional bias in Timer-XL on ETTm1.** Two 30-day input windows are used to forecast the following day. The top panels compare Timer-XL and Timer-XL+BTA predictions and visualize the dominant temporal kernels selected by BTA; the bottom panels show the corresponding router weights for layer 2, head 2. For the left window, which exhibits an abrupt recent change, BTA emphasizes short-range local kernels, with its three largest weights assigned to local scales of 0.234, 0.117, and 0.469 days. For the more stable right window, BTA shifts toward substantially broader local dependencies together with a periodic component. These input-dependent mixtures induce different positional biases and reduce the forecasting MSE from 0.019 to 0.012 and from 0.031 to 0.019, respectively, demonstrating that BTA can adapt the temporal dependencies of a pretrained Transformer to non-stationary input dynamics.

M MORE REPRESENTATION ANALYSIS ON SOLAR

Figure 11 further confirms that BTA adapts its temporal inductive bias to the forecasting horizon on Solar. The learned kernel weights vary systematically across prediction lengths: shorter horizons place greater emphasis on relatively local dependencies, whereas longer horizons increasingly favor broader local structure, with the $\ell = 1.25$-day RBF kernel becoming dominant at $H = 720$. The allocation among periodic kernels also changes with the horizon, with longer forecasts placing greater weight on broader 3-day periodic patterns. These results further demonstrate that BTA does not rely on a fixed positional structure, but instead adjusts its temporal dependencies to the forecasting horizon, providing an interpretable view of the adaptive positional bias for non-stationary LTSF.

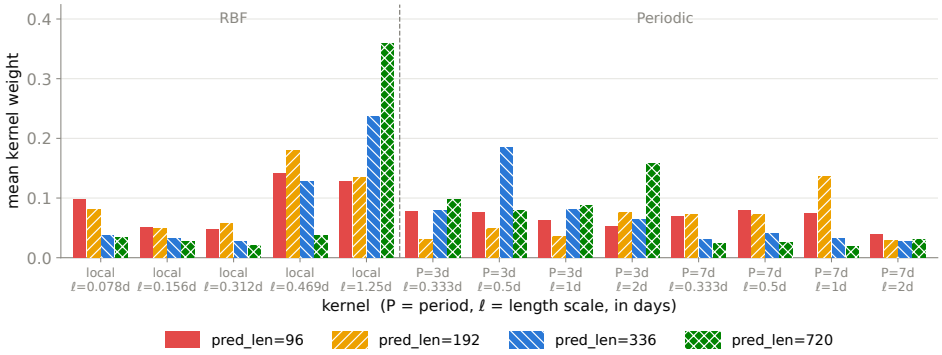


Figure 11: **BTA reveals horizon-dependent temporal dependencies on Solar.** Mean kernel weights across Solar test samples and attention heads for $H \in \{96, 192, 336, 720\}$. The dashed line separates local RBF and periodic kernels. As the forecasting horizon increases, BTA shifts weight toward broader local dependencies and redistributes weight among periodic kernels, with the broadest local RBF kernel becoming dominant at $H = 720$.

N Additional Hyperparameter Analysis

Figure 12 complements the ETTm1 hyperparameter analysis in Figure 4 with results on ETTh1 and Solar.

Top-k. ETTh1 is nearly insensitive to the tested Top-k configurations, with a maximum relative MSE difference below 0.04%. Solar exhibits greater variation, with $k = 3$ attaining the lowest displayed MSE and the remaining settings increasing MSE by up to 8.20%. However, the ETTh1 $k = 1$ run uses a different batch size from the $k \in \{3, 5, 7\}$ runs, while the Solar $k = 3$ run uses a different number of patch-downsampling layers. These results should therefore be interpreted as descriptive sensitivity analyses rather than controlled ablations of k alone.

Router hidden dimension. Sensitivity to router capacity is also dataset-dependent. ETTh1 attains its lowest MSE at $d_r = 192$, with at most a 1.23% increase across the tested dimensions, whereas Solar attains its minimum at $d_r = 96$ and increases by up to 9.30% at $d_r = 224$. Together with the ETTm1 results, these findings suggest that moderate router capacity is generally sufficient, while the optimal hidden dimension depends on the dataset.

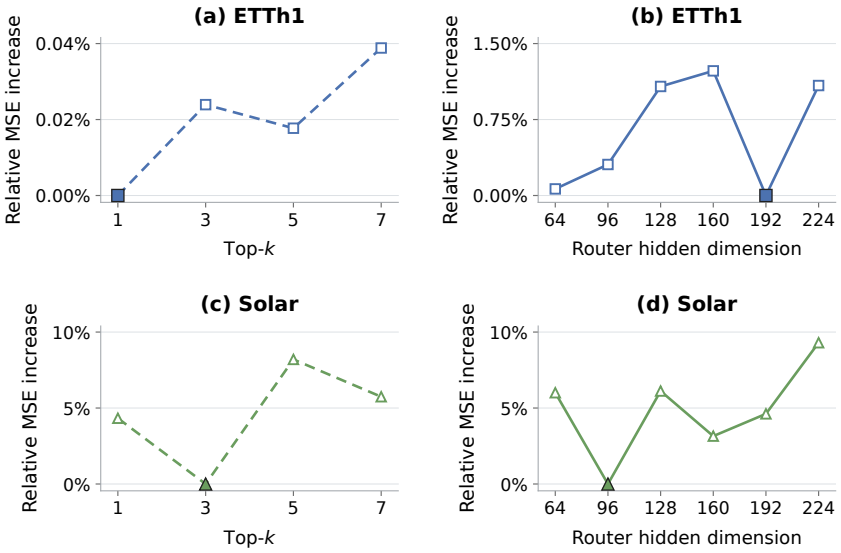


Figure 12: **Additional hyperparameter sensitivity analysis on ETTh1 and Solar.** The top and bottom rows report results on ETTh1 and Solar, respectively, with Top-k varied in the left column and router hidden dimension d_r varied in the right column. Each panel reports the relative MSE increase with respect to the lowest MSE within that panel and uses an independent vertical scale; filled markers indicate the minimum. Dashed curves denote the Top-k sweeps, for which auxiliary configuration differences prevent a strictly controlled comparison of k alone.

O EFFICIENCY ANALYSIS

Table 13: **Computational efficiency on ETTh1, ETTm1, and Weather.** *Ours* denotes TimeBridge+BTA. FLOPs are measured per input window, and Size denotes the number of trainable parameters. Results are reported for forecasting horizons $H \in \{96, 192, 336, 720\}$.

Model		Ours		TimeBridge Liu et al. (2025a)		DUET Qiu et al. (2025)		DTAF Lu et al. (2026)	
Metric		FLOPs	Size	FLOPs	Size	FLOPs	Size	FLOPs	Size
ETTh1	96	274.9M	0.786M	274.6M	0.772M	35.7M	3.605M	19.0M	0.582M
	192	280.0M	1.155M	279.7M	1.141M	46.7M	5.132M	26.8M	1.121M
	336	287.8M	1.708M	287.5M	1.693M	65.4M	5.730M	38.4M	1.964M
	720	308.4M	3.182M	308.1M	3.168M	72.6M	6.981M	69.4M	4.176M
ETTm1	96	47.4M	0.252M	46.9M	0.230M	46.1M	5.083M	19.0M	0.587M
	192	48.7M	0.344M	48.2M	0.322M	19.2M	2.423M	26.8M	1.165M
	336	50.6M	0.482M	50.1M	0.460M	47.8M	5.206M	38.6M	1.980M
	720	55.8M	0.851M	55.3M	0.829M	10.5M	1.432M	69.9M	4.253M
Weather	96	288.9M	0.575M	288.0M	0.556M	73.1M	2.794M	57.7M	0.581M
	192	296.6M	0.759M	295.7M	0.740M	206.5M	6.710M	80.9M	1.145M
	336	308.2M	1.036M	307.3M	1.017M	209.6M	6.784M	115.8M	1.974M
	720	339.2M	1.773M	338.3M	1.754M	217.8M	6.981M	208.7M	4.187M

Controlled comparison with the backbone. Because BTA is integrated into TimeBridge while retaining the remaining backbone components and training protocol, the comparison between TimeBridge+BTA and TimeBridge directly quantifies the additional computational cost of BTA. Table 13 shows that this overhead is small relative to the forecasting gains. On ETTh1, BTA adds approximately 0.31 M FLOPs per input window (0.11% on average) and 14.4 K trainable parameters (1.1%), while reducing the average MSE from 0.399 to 0.394 (1.3%). On ETTm1, it adds approximately 0.48 M FLOPs (0.96%) and 22.5 K parameters (6.1%), while reducing the average MSE from 0.348 to 0.337 (3.2%). On Weather, the additional computational cost is likewise small, while BTA further improves the TimeBridge backbone. These results show that BTA achieves forecasting gains with modest additional computation and capacity.

Horizon-independent overhead. Because BTA operates on the input-side attention representation and introduces no additional forecasting-head component, its absolute FLOP and parameter overheads remain approximately constant across prediction horizons. In contrast, the backbone cost increases with the forecasting horizon, causing the relative overhead of BTA to decrease. For example, on ETTh1 at $H = 720$, BTA increases FLOPs by only 0.10% and trainable parameters by 0.46%.

Performance gains are not explained by model capacity. The comparison with other methods further indicates that larger model capacity does not directly account for the forecasting improvements. On ETTh1, DUET uses 3.6–7.0 M trainable parameters, approximately 2.2–$4.6\times$ those of TimeBridge+BTA across the four horizons, yet has a higher average MSE (0.404 versus 0.394). Together, these results suggest that the gains of BTA primarily arise from its adaptive modeling of temporal dependencies rather than from substantially increased computation or parameter count.

P More Ablation Study

Table 14 shows that each component of BTA contributes to forecasting performance on Weather. Adding local and periodic kernels improves vanilla self-attention, while replacing static kernel weights with input-adaptive MoE routing yields further gains. Incorporating the Fourier spectrum of the TE profile achieves the best overall performance, reducing average MSE and MAE by 14.05% and 7.73%, respectively. These results demonstrate the complementary benefits of a richer temporal kernel bank, input-adaptive routing, and frequency-domain information.

Table 14: **Ablation study of BTA on Weather with Informer as the backbone.** Starting from vanilla self-attention, we progressively introduce local kernels, periodic kernels, input-adaptive MoE routing based on the temporal-evidence (TE) profile, and its Fourier spectrum. Gain (%) denotes the relative reduction in MSE and MAE with respect to vanilla self-attention.

Attention		Self-attention		Bayesian Time-aware Attention							
Optimization		Static		Static				MoE (TE only)		MoE (TE+Fourier)	
Kernels		None		Local		Local+Periodic		Local+Periodic		Local+Periodic	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.313	0.375	0.274	0.353	0.270	0.348	0.266	0.339	**0.253**	**0.335**
	192	0.377	0.408	0.333	0.394	0.320	0.384	0.313	0.377	**0.299**	**0.369**
	336	0.448	0.449	0.467	0.458	0.460	0.457	0.412	0.428	**0.389**	**0.413**
	720	0.712	0.604	0.696	0.597	0.689	0.590	0.662	0.583	**0.649**	**0.577**
	Avg	0.463	0.459	0.443	0.451	0.435	0.445	0.413	0.432	**0.398**	**0.424**
	Gain(%)	–	–	4.32	1.85	6.00	3.10	10.65	5.94	**14.05**	**7.73**