FOCUS-TS: Online Learning for Seasonal–Trend Forecasting via Adaptive Fourier Order and Kernelized Trend

Nguyen The Phong[a,b], Nguyen Van Hanh[a,b,*], Nguyen Thi Ngoc Anh[a,b]

[a] *Faculty of Mathematics and Informatics, Hanoi University of Science and Technology, No. 1 Dai Co Viet Road, Hanoi 100000, Vietnam*

[b] *Institute for Digital Technology and Digital Economy (BKFintech), Hanoi University of Science and Technology, No. 1 Dai Co Viet Road, Hanoi 100000, Vietnam*

Abstract

Accurate and adaptive load forecasting is critical under volatile demand and renewable integration. Traditional statistical models struggle with nonlinear, time-varying patterns, while deep learning often requires costly retraining and lacks interpretability. We propose **FOCUS-TS** (**FO**urier-order **C**onvex **U**pdates for **S**easonal-Trend in **T**ime **S**treams), a regret-aware online framework that updates only interpretable trend and seasonal components. Each series is decomposed into a kernelized nonlinear trend, an adaptive Fourier seasonal signal, and a residual. The trend is learned via random Fourier features with constant-time convex updates, while seasonal harmonics are selected online through hysteresis to avoid unstable order changes. Dynamic local regret regulates stability and plasticity, triggering adaptive learning-rate and frequency adjustments under drift. A multivariate extension shares frequencies across loads while preserving channel-specific amplitudes. With per-step complexity linear in active features, FOCUS-TS delivers interpretable, robust, and efficient forecasts suitable for real-time deployment in smart grids and energy markets.

Keywords: Online learning, Seasonal–trend decomposition, Kernel methods, Fourier analysis, Streaming forecasting

1. Introduction

The rapid growth of renewable generation and the deployment of smart grids have made accurate real-time forecasting of load and generation series central to modern power system operations [1, 2]. System-level variables such as load, global horizontal irradiance (GHI), wind power, and traffic-related demand proxies exhibit strong seasonal and trend structures [3], yet remain nonlinear, noisy, and nonstationary due to weather [1, 4], user behavior [5], and external shocks [6]. Without timely and interpretable estimates of these components, operators cannot ensure supply–demand balance, plan reserves, or schedule flexible

[*] Corresponding author: hanh.nguyenvan@hust.edu.vn (Nguyen Van Hanh)

Email addresses: phong.nt20252308M@sis.hust.edu.vn (Nguyen The Phong), hanh.nguyenvan@hust.edu.vn (Nguyen Van Hanh), anh.nguyenthingoc@hust.edu.vn (Nguyen Thi Ngoc Anh)

assets under volatile market conditions [7, 8]. These difficulties intensify in streaming scenarios, where data arrive continuously, distributions drift across seasonal regimes, and forecasting must be carried out in real time under strict computational budgets at the edge.

Over the last decade, load forecasting research has advanced across statistical models, machine learning, deep neural networks, and hybrid decompositions. Classical approaches such as SARIMA remain competitive due to their explicit representation of periodicity and low cost [4], but are sensitive to differencing and seasonal orders, and perform poorly under nonlinear or irregular cycles [7, 9]. Deep models including LSTM, GRU, CNN, Transformer families, and decomposition-based hybrids capture long-range dependencies and multi-frequency patterns [8, 10, 11, 12], but they are typically trained offline, require heavy computation, risk overfitting with limited data, and lack interpretability for operational use. To address these shortcomings, many studies combine time–frequency methods such as STL, SSA, EMD, VMD, Fourier, or wavelets with advanced learners (LSTM, TFT, XGBoost) [3, 12]. These methods improve accuracy in offline settings; however, online forecasting with constant per-step complexity is still insufficiently explored.

A recurring theme in recent work is explicit decomposition of seasonality and trend before prediction, which improves interpretability and reduces model burden. Examples include aSTL–UGSSA, which integrates enhanced STL and SSA with GRU modeling [12], disentanglement methods such as SPD–TmNet and ISTR–TFT that stabilize long-horizon forecasts by modeling seasonal–trend components explicitly [8, 13], and grey models incorporating Fourier or dummy terms into GM/DGM to capture cycles under small datasets [9, 14, 15]. Despite these advances, most pipelines remain batch-oriented, assume stationarity, and lack systematic mechanisms for online adaptation under drift or regime change.

From this literature, three research gaps emerge. First, no existing online framework simultaneously preserves interpretable seasonal–trend structures and updates rapidly at each step. Second, adaptive Fourier-order selection for multi-seasonal components is lacking, leading to instability and amplitude leakage. Third, lightweight nonlinear trend representations that are learnable online with regret guarantees and with per-step complexity independent of history remain absent—yet essential for energy, solar, and wind series subject to frequent drift [8, 11, 16]. Most time–frequency methods (STL, SSA, VMD, wavelet, Fourier) remain offline; when adapted online they require long windows, frequent retuning, or stay decoupled from regret-based learners, limiting stability–plasticity balance [4, 12, 14]. Even recent hybrids, such as disentanglement with attention or 2D backbones, and CoST with mixture-of-experts for NWP correction, improve robustness across regimes but still lack principled regret-aware online updates and remain too heavy for edge deployment [3, 8, 13].

To address these challenges, we propose **FOCUS-TS** (**FO**urier order **C**onvex **U**pdates for **S**easonal–**T**rend in **T**ime **S**treams), a regret-aware online framework for interpretable seasonal–trend forecasting. FOCUS-TS is built on four principles: (i) explicit online decomposition into kernelized nonlinear trend, adaptive Fourier seasonal components, and residuals [4, 8, 12, 13, 14, 16]; (ii) lightweight nonlinear trends via random Fourier features (RFF), enabling constant-time convex updates with regret guarantees [15]; (iii) hysteresis-based harmonic selection for stable and adaptive Fourier-order adjustment [4, 12, 13]; and (iv) dynamic local regret to regulate stability–plasticity trade-offs in real time [17]. Together, these elements deliver interpretable, robust, and efficient online forecasts, bridging the gap between traditional statistical models and computationally intensive deep learning.

The main contributions are:

1. Proposes FOCUS–TS, a regret-aware online forecasting framework that jointly models trend and seasonality with interpretable, drift-adaptive updates for nonstationary data streams.
2. Develops a convex random Fourier feature learner enabling efficient kernelized trend estimation with constant-time updates and guaranteed stability under streaming conditions.
3. Introduces a PSD-driven spectral adaptation mechanism that dynamically adjusts frequency bases, ensuring smooth and interpretable transitions in evolving temporal patterns.
4. Designs an Exponentially-Weighted Gradient Ratio (EGR) controller to detect concept drift and maintain a balance between model plasticity and long-term stability.
5. Extends FOCUS–TS to multivariate forecasting with group regularization, offering scalable online inference and theoretical regret bounds across diverse temporal datasets.

We further extend FOCUS-TS to multivariate series via group regularization, sharing seasonal frequencies and phases across channels while retaining channel-specific amplitudes [4, 16]. This reflects system-wide synchrony with regional diversity. Overall, FOCUS-TS combines interpretability, adaptability, and efficiency, transferring the strengths of decomposition and disentanglement from offline to online forecasting.

The structure of this paper is organized as follows. Section 1 introduces the motivation, background, challenges in electric load forecasting, and the main contributions. Section 2 reviews related work and formulates the problem. Section 3 describes the proposed **FOCUS-TS** framework. Section 4 presents experimental results on electric load and benchmark datasets, with comparisons to statistical, deep learning, and hybrid baselines. Section 5 concludes the study and discusses future research directions.

Table 1: List of Abbreviations.

Abbreviation	**Meaning**	**Abbreviation**	**Meaning**
SARIMA	Seasonal ARIMA	STL	Seasonal–Trend decomposition using Loess
SSA	Singular Spectrum Analysis	EMD	Empirical Mode Decomposition
VMD	Variational Mode Decomposition	TBATS	Trigonometric, Box–Cox, ARMA, Trend, Seasonal
PSD	Power Spectral Density	FFT	Fast Fourier Transform
GHI	Global Horizontal Irradiance	NWP	Numerical Weather Prediction

Abbreviation	Meaning	Abbreviation	Meaning
RFF	Random Fourier Features	RBF	Radial Basis Function (kernel)
KRR	Kernel Ridge Regression	OCO	Online Convex Optimization
OGD	Online Gradient Descent	ONS	Online Newton Step
SLR	Static Local Regret	DLR	Dynamic Local Regret
EGR	Exponentially-weighted Gradient Ratio	EWMA	Exponentially Weighted Moving Average
NMS	Non-Maximum Suppression	RevIN	Reversible Instance Normalization
OGD-Prox	OGD with proximal regularization	MoE	Mixture of Experts
LSTM	Long Short-Term Memory	GRU	Gated Recurrent Unit
CNN	Convolutional Neural Network	TFT	Temporal Fusion Transformer
N - BEATS	Neural basis expansion analysis for interpretable time series	XGBoost	Extreme Gradient Boosting
ISTR - TFT	Interpretable Seasonal–Trend Representation + TFT	SPD - TmNet	Seasonal & Periodic-trend Disentangled TimesNet
CoST	Corrections with Seasonal–Trend (hybrid family)	FOCUS – TS	**FO**urier-order **C**onvex **U**pdates for **S**easonal–**T**rend **T**ime **S**treams

Table 2: List of Symbols.

Symbol	Description	Symbol	Description
Variables			
X_t	Observation at time t	$\hat{X}_t$	Forecast at time t
T_t	Trend component at time t	S_t	Seasonal component at time t
ε_t	Residual (diagnostic noise) at time t	r_t	Trend residual: $X_t - S_{t-1} - w_t^\top z_t$
e_t	Seasonal residual: $X_t - T_t - \beta_t^\top \phi_t$	g_t	Gradient at time t (generic)
$g_t^{(S)}$	Seasonal gradient at time t	θ_t	Parameter vector at time t
w_t	Trend weights in RFF space	z_t	Random Fourier feature vector
u_t	Input feature vector (lags, calendar, exogenous)	β_t	Seasonal coefficient vector
ϕ_t	Seasonal design vector from active frequencies	F_t	Active frequency set at time t
N_t	Number of active frequencies $\lvert F_t \rvert$	$\widehat{P}_t(\omega)$	Welch PSD at time t
$X_t^{(m)}$	Observation in channel m at time t	M	Number of channels (multivariate)
Parameters / Constants			

Symbol	Description	Symbol	Description
D	RFF dimension	d	Input (feature) dimension
W	Window length for PSD/EGR	Δ	PSD refresh interval (steps)
η_t	Learning rate at time t	η_0	Base learning-rate constant
λ_T	Ridge penalty for trend (KRR)	λ_S	Ridge penalty for seasonal coefficients
λ_G	Group ridge for multivariate seasonality	γ_T	Proximal regularizer for trend drift
γ_S	Proximal regularizer for seasonal drift	σ_t	Kernel bandwidth (e.g., Gaussian/RBF)
Ω	RFF frequency matrix	b	RFF random phase vector
R_T	Projection radius for trend (ℓ_2-ball)	R_S	Projection radius for seasonal (ℓ_2-ball)
$\Pi_{\mathcal{P}}(\cdot)$	Euclidean projection onto set $\mathcal{P}$	$\mathcal{P}_T$	Feasible set for trend parameters
$\mathcal{P}_S$	Feasible set for seasonal parameters	τ_{add}	PSD threshold to add a frequency
τ_{drop}	PSD threshold to drop a frequency	K_{add}	Hysteresis counter for adding
K_{drop}	Hysteresis counter for dropping	ρ	Exponential decay factor in EGR/DLR
w	EGR/DLR window size	β_{EGR}	Weight for EGR penalty
EGR_t	EGR value at time t	$\mathsf{EGN}_{t,w,\rho}$	EWMA of squared gradient norms
μ_t, σ_t	EWMA mean/std of EGR statistic	κ	Drift z-score threshold
η_{nom}	Nominal learning rate	$\eta_{\max}$	Max learning rate under trigger
λ_T^{nom}	Nominal trend ridge	λ_S^{nom}	Nominal seasonal ridge
α	Strong convexity parameter (regret theory)	β	Regularization scaling in trigger policy
G_T, G_S	Gradient bounds (trend/seasonal)	P_T	Path length of drifting comparator
$\mathcal{K}$	Generic convex decision set		
Objective / Losses			
$\ell_t^{(T)}(w)$	Trend loss at time t	$\ell_t^{(S)}(\beta)$	Seasonal loss at time t
L_t	Per-round total objective (data-fit + regs + EGR)	Regret	Static/dynamic regret (as defined)

2. Related Works

The comparative overview in Table 3 highlights the evolution of seasonal–trend forecasting methods from classical decomposition and frequency-domain models to modern deep learning and online learning approaches. Traditional methods such as STL [18], TBATS [19], and Prophet [20] emphasize interpretability through explicit decomposition of trend, seasonality, and residuals. Frequency-domain techniques [21, 22] and grey-system models [5, 15, 9] extend this line of work toward capturing more complex periodic structures and small-sample robustness. In contrast, deep learning and hybrid architectures

[23, 24, 8, 13, 12] prioritize predictive accuracy, often at the expense of transparency. Meanwhile, the online learning literature [25, 26, 27] has focused on real-time adaptivity but without explicit seasonal–trend decomposition.

Overall, existing methods tend to excel in either interpretability or adaptivity, but rarely both. The proposed FOCUS–TS framework aims to unify these strengths by combining Fourier–kernel decomposition with online convex optimization, providing a principled balance of interpretability, adaptability, and computational efficiency for real-world forecasting tasks.

Table 3: Comparison of representative works across interpretability and online learning.

Sources	**Year**	**Model / Method**	**Interpretability**				**Online learning**
			Trend	**Seasonal**	**Residual**	**Full decomposition**	
[18]	1990	Seasonal-Trend decomposition using Loess	✓	✓	✓	✓	✗
[21]	2003	Power Spectral Density	✗	✓	✗	✗	✗
[5]	2009	TF-e-SVR–SA (Trend Fixed + Seasonal Adjustment + e-SVR)	✓	✓	✗	✗	✗
[19]	2011	TBATS	✓	✓	✓	✓	✗
[22]	2013	Spectral-mixture kernels	✓	✓	✗	✗	✗
[20]	2018	Prophet	✓	✓	✗	✓	✗
[28]	2018	Multi-channel Adaptive Fourier decomposition	✓	✗	✓	✗	✗
[23]	2019	N-BEATS	✗	✗	✓	✗	✗
[24]	2022	FEDformer	✓	✓	✗	✗	✗
[29]	2023	Adaptive Fourier decomposition	✓	✗	✓	✗	✗
[15]	2024	non-linear grey Fourier model	✓	✓	✗	✗	✗
[7]	2024	MAFS + ISTD + PGBM	✓	✓	✓	✓	✗

Sources	Year	Model / Method	Interpretability				Online learning
			Trend	Seasonal	Residual	Full decomposition	
[8]	2024	ISTR-TFT (Seasonal-Trend Representation + Temporal Fusion Transformer)	✓	✓	✗	✗	✗
[13]	2024	SPD-TmNet (Seasonal & Periodic-trend feature Disentangled TimesNet)	✓	✓	✓	✓	✗
[11]	2025	MWENet	✓	✓	✗	✗	✗
[12]	2025	TFR (aSTL-UGSSA + GRU + MPAdam)	✓	✓	✓	✓	✗
[16]	2025	STNet (EWMA + Dual-branch MLP)	✓	✓	✗	✗	✗
[3]	2025	CoST-MoELSTM (Hybrid correction with disentangled seasonal-trend + MoE)	✓	✓	✗	✗	✗
[4]	2025	SARIMA, Prophet, XGBoost (spectral analysis + exogenous cycle index)	✓	✓	✓	✗	✗
[9]	2026	DSDGM(1,1) (Discrete Seasonal Grey Model with dummy variables)	✓	✓	✗	✗	✗
[25]	2003	Online convex programming	✗	✗	✗	✗	✓
[30]	2004	Online kernel learning	✗	✗	✗	✗	✓
[31]	2007	Bayesian online changepoints	✗	✗	✗	✗	✓

Sources	Year	Model / Method	Interpretability				Online learning
			Trend	Seasonal	Residual	Full decomposition	
[26]	2013	Online ARMA	✗	✗	✗	✗	✓
[27]	2014	Drift detection	✗	✗	✗	✗	✓
[32]	2016	OCO algorithms	✗	✗	✗	✗	✓
This study (FOCUS-TS)	2025	Fourier–kernel Online Convex Update for Seasonal–Trend	✓	✓	✗	✗	✓

2.1. *Interpretability for time series*

Interpretability has long been a central concern in time series forecasting, particularly in domains energy where decision-making requires transparent models. Classical decomposition methods, including STL [18], TBATS [19], and Prophet [20], directly separate the observed series into trend, seasonal, and residual components. This explicit additive structure enables clear attribution of variability, making these approaches highly interpretable while also supporting practical tasks such as anomaly detection and diagnostics.

Beyond classical decomposition, frequency-domain approaches provide an alternative form of interpretability by explicitly modeling periodic behavior. Power Spectral Density (PSD) analysis [21] offers insight into dominant frequencies, while spectral-mixture kernels [22] embed spectral representations into kernel functions, thereby capturing latent oscillatory patterns. These methods retain interpretability by grounding predictions in frequency content, although they typically do not produce a full decomposition into trend, seasonal, and residual terms.

Grey-system based models provide another interpretable line of research, especially for small-sample scenarios. For instance, TF-e-SVR–SA [5], the non-linear grey Fourier model [15], and DSDGM [9] offer parameterizations that are both simple and interpretable, allowing transparent examination of trend and seasonality under limited data availability.

In contrast, deep learning and hybrid models, such as N-BEATS [23], FEDformer [24], ISTR-TFT [8], SPD-TmNet [13], and TFR [12], provide strong predictive accuracy by leveraging nonlinear feature extraction and temporal interactions. However, they often behave as "black-box" models with limited interpretability. Although some recent architectures attempt to disentangle seasonal and trend representations (e.g., SPD-TmNet, ISTR-TFT), the level of transparency remains lower than that of classical decomposition or frequency-domain approaches. This interpretability gap is a key motivation for designing models that preserve decomposition structure while achieving strong predictive performance.

2.2. *Online Learning*

In streaming and non-stationary settings, predictors must be updated on the fly rather than retrained on a fixed corpus. A rich toolbox exists for such problems—including online convex optimization (OCO) [25, 32], online kernel methods [30], online ARMA [26], Bayesian changepoint detection [31], and drift-detection techniques [27]. These lines of work prioritize adaptivity and regret control, yet they seldom retain an explicit seasonal–trend decomposition that is crucial for interpretability in power and energy applications. Our framework bridges this gap by coupling OCO-style updates with an online, interpretable seasonal–trend structure (see Section 3).

2.2.1. *From batch to streaming: opportunities and pitfalls*

Unlike batch (offline) training—where parameters are fitted once on a static dataset—online learning continuously refines the model as each data point arrives [33]. The paradigm dates back to early incremental procedures such as the Perceptron (Rosenblatt, 1958) and has since evolved to include stochastic-gradient training for non-convex models, together with classical OCO algorithms such as Winnow, Online Gradient Descent (OGD), and Online Newton Step (ONS) [34, 17].

Despite its efficiency, several challenges recur in practice: (i) *catastrophic forgetting*, where new patterns overwrite long-term structure; mitigations include incremental/regularized updates and controlled forgetting [35]. (ii) *Concept drift*, which calls for detectors and targeted adaptation policies rather than wholesale retraining. (iii) *Online ensembles*, which trade off fast responsiveness against long-horizon memory by weighting experts at different time scales. (iv) *Non-convex settings*, where regret-style guarantees guide stability despite complex objectives [17]. FOCUS–TS operationalizes these ideas with lightweight convex subproblems for trend/seasonality, plus a drift-aware controller (EGR) that modulates learning rates and regularization (Sections 3.2 and 3.4).

2.2.2. *A brief primer on Online Convex Optimization*

In OCO, a learner selects a decision $\mathbf{X}_t \in \mathcal{P}$ on each round t, where $\mathcal{P} \subseteq \mathbb{R}^n$ is convex, bounded, and closed [34]. After choosing $\mathbf{X}_t$, a convex loss $f_t : \mathcal{P} \to \mathbb{R}$ is revealed and the learner incurs $f_t(\mathbf{X}_t)$. A (possibly randomized) algorithm $\mathcal{A}$ produces decisions using only past information, i.e., $\mathbf{X}_t = \mathcal{A}(f_1, \ldots, f_{t-1})$. Performance is measured by regret relative to the best fixed comparator in hindsight:

$$\text{Regret}(\mathcal{A}, \{f_t\}_{t=1}^{T}) = \mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{X}_t)\right] - \min_{\mathbf{X}\in\mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{X}). \tag{1}$$

Small regret implies the online strategy competes well with the single best point chosen after observing all losses. In FOCUS–TS, both the trend (kernelized via random features) and the seasonal coefficients are updated by projected OGD-type steps, ensuring convexity and bounded regret (Section 3.1.1, 3.1.2).

2.3. *Regret for online forecasting*

For time-series prediction, parameters $\mathbf{X}_t$ are revised at each time t to reflect the most recent observations. Each step defines a loss $f_t : \mathcal{K} \to \mathbb{R}$ on a convex set $\mathcal{K}$, and the realized loss is $f_t(\mathbf{X}_t)$. While classical regret compares against a fixed comparator, non-stationarity motivates local criteria that better reflect evolving optima.

2.3.1. *Static Local Regret (SLR)*

SLR gauges local stationarity by aggregating gradients of recent losses at the *current* parameter [34, 32]:

$$\mathrm{SLR}_w(T) = \sum_{t=1}^{T} \left\| \frac{1}{w} \sum_{i=0}^{w-1} \nabla f_{t-i}(\mathbf{X}_t) \right\|^2 . \tag{2}$$

A small average gradient suggests near-local optimality over the last w steps. However, because SLR evaluates past losses at $\mathbf{X}_t$ (not at the parameters used in the past), it can be biased under drift: parameters tuned for January may not be appropriate to re-evaluate November's losses.

2.3.2. *Dynamic Local Regret (DLR)*

To address the temporal mismatch, DLR aggregates gradients at the parameters used when those losses were incurred, with exponential discounting of older information [17]:

$$\mathrm{DLR}_w(T) = \sum_{t=1}^{T} \left\| \nabla\left(\frac{1}{W} \sum_{i=0}^{w-1} \rho^i f_{t-i}(\mathbf{X}_{t-i}) \right) \right\|^2 , \quad W = \sum_{i=0}^{w-1} \rho^i,\ \rho \in (0,1). \tag{3}$$

Exponential weights serve two purposes: (i) they prioritize recent behavior, supporting rapid adaptation; (ii) by down-weighting stale gradients ($\rho < 1$), they stabilize updates and support sublinear regret rates. In FOCUS–TS, this idea is *operationalized* via the Exponentially-weighted Gradient Ratio (EGR): rather than a purely evaluative metric, EGR is used as a control signal that adjusts learning rates, regularization, and hysteresis thresholds for seasonal-frequency updates when drift is detected (Section 3.2). This turns local-regret principles into a practical mechanism for stability–plasticity trade-offs.

3. Methodology

3.1. *Problem statement*

A real-valued time series, either univariate or multivariate, is observed sequentially. For clarity of exposition, the univariate case is introduced first. At each time step $t \in \mathbb{N}$, an observation $X_t \in \mathbb{R}$ is recorded. The series is assumed to follow an additive decomposition of the form

$$X_t = T_t + S_t + \varepsilon_t, \tag{4}$$

where:

- T_t denotes the slowly varying trend component that captures long-term structural dynamics.
- S_t represents the recurring seasonal or cyclic component.
- ε_t is the residual term, accounting for noise or unexplained irregularities. This part is not explicitly modeled but retained for diagnostic purposes.

For identifiability, the residual is assumed to satisfy $\mathbb{E}[\varepsilon_t] = 0$, ensuring that $\mathbb{E}[X_t] = T_t + S_t$.

Spectral adaptation and online regularization

A sliding window of length $W \in \mathbb{N}$ is employed for periodogram estimation and refreshed every $\Delta \in \mathbb{N}$ steps. We introduce the Exponentially-weighted Gradient Regularizer (EGR) to stabilize seasonal updates and to act as a drift monitor (see Section 3.2). EGR is inspired by the exponentially-weighted form used in dynamic local regret (DLR) (see 2.3.2) in online non-convex optimization [17]; however, unlike DLR (a theoretical performance metric), EGR is operationalized as a practical penalty/monitor in our online decomposition. The learning rate is denoted by $\eta_t > 0$, while regularization parameters are

$$\lambda_T,\ \lambda_S,\ \lambda_G,\ \gamma_T,\ \gamma_S \ \geq\ 0. \tag{5}$$

Trend and seasonal parameters are projected onto compact convex sets

$$\mathcal{P}_T \subset \mathbb{R}^D, \qquad \mathcal{P}_S \subset \mathbb{R}^{2N_t}, \tag{6}$$

using the Euclidean projection operator $\Pi_{\mathcal{P}}(\cdot)$. For numerical stability, feature norms are assumed bounded:

$$\|z_t\| \ \leq\ B_z, \qquad \|\phi_t\| \ \leq\ B_\phi. \tag{7}$$

In the multivariate setting with $X_t \in \mathbb{R}^M$, all quantities receive a channel index $^{(m)}$. The frequency set F_t is shared across channels, while channel-specific seasonal coefficients $\beta_t^{(m)}$ are jointly regularized (see Section 3.3).

3.1.1. Trend Component: Online Kernel Ridge via Random Fourier Features

We estimate the trend in a random-feature space via kernel ridge regression (KRR) [36]. The prediction is

$$T_t \ =\ w_t^\top z_t, \tag{8}$$

where $w_t \in \mathbb{R}^D$ are the weights and $z_t \in \mathbb{R}^D$ is the random Fourier feature (RFF) mapping [37] of the input u_t.

The input feature vector is

$$u_t \ \in\ \mathbb{R}^d, \tag{9}$$

and may include lags, cumulative statistics, calendar dummies (hour-of-day, weekday; one-hot or sin / cos), and exogenous covariates.

The RFF mapping approximates a shift-invariant kernel $k(u, u') = \kappa(\|u - u'\|)$:

$$z_t \;=\; \sqrt{\tfrac{2}{D}}\,\cos(\Omega u_t + b), \tag{10}$$

where $\Omega \in \mathbb{R}^{D\times d}$ has iid rows from $\mathcal{N}(0, \sigma_t^{-2} I_d)$, $b \in \mathbb{R}^D$ has iid entries Unif$[0, 2\pi]$, and $\sigma_t > 0$ is a bandwidth that can be adapted (e.g., reduced when the periodogram reveals higher dominant frequencies or when EGR indicates increased curvature).

Conditioning on the previous seasonal estimate S_{t-1}, the convex per-round trend loss is

$$\ell_t^{(T)}(w) \;=\; \tfrac{1}{2}\left(X_t - S_{t-1} - w^\top z_t\right)^2 \;+\; \lambda_T \|w\|_2^2. \tag{11}$$

Its gradient at w_t is

$$\nabla \ell_t^{(T)}(w_t) \;=\; -\big(X_t - S_{t-1} - w_t^\top z_t\big) z_t \;+\; 2\lambda_T w_t. \tag{12}$$

In this work, two practical and theoretically consistent design choices are adopted to ensure numerical stability and provable convergence of the online trend learner. First, the projection set $\mathcal{P}_T$ is defined as an ℓ_2-ball,

$$\mathcal{P}_T = \{w : \|w\|_2 \le R_T\}, \qquad \Pi_{\mathcal{P}_T}(v) = \begin{cases} v, & \text{if } \|v\|_2 \le R_T, \\ \dfrac{R_T}{\|v\|_2}\, v, & \text{otherwise}, \end{cases} \tag{13}$$

which naturally complements the ℓ_2 ridge penalty in (11). This projection keeps the iterates bounded—a key requirement for sublinear regret bounds in online convex optimization—and improves numerical stability in high-dimensional random-feature spaces. Unlike ℓ_1 projections, the ℓ_2 constraint does not impose unnecessary sparsity on the random Fourier features, maintaining smooth trend estimates.

Second, the learning rate follows the standard decaying schedule

$$\eta_t = \frac{\eta_0}{\sqrt{t}}, \tag{14}$$

which guarantees an $O(\sqrt{T})$ regret bound for online gradient descent (OGD) on convex, Lipschitz-continuous losses [25]. This simple yet effective schedule enables fast adaptation at early iterations while gradually stabilizing as t increases, aligning well with the slowly varying nature of the trend component.

To enhance robustness against outliers, the squared loss in (11) can optionally be replaced by the Huber loss [38]:

$$\rho_\delta(r) = \begin{cases} \tfrac{1}{2} r^2, & |r| \le \delta, \\ \delta(|r| - \tfrac{1}{2}\delta), & |r| > \delta, \end{cases} \qquad r = X_t - S_{t-1} - w^\top z_t, \tag{15}$$

which yields a clipped gradient $-\psi_\delta(r)z_t + 2\lambda_T w_t$ with $\psi_\delta(r) = \mathrm{clip}(r; [-\delta, \delta])$.

Algorithm 1 Online Trend Update via Projected OGD in RFF Space

```
Input: Observation $X_t$; previous seasonal estimate $S_{t-1}$; input $u_t \in \mathbb{R}^d$; RFF parameters ($\Omega \in \mathbb{R}^{D\times d}$, $b \in \mathbb{R}^D$, $\sigma_t$); current
       weights $w_t \in \mathbb{R}^D$; base learning rate $\eta_0$; ridge parameter $\lambda_T$; projection radius $R_T$; (optional) Huber threshold $\delta$.
Output: Updated trend estimate $T_t$ and weights $w_{t+1}$.
Step 1: Feature Mapping. Compute RFF features using Eq. (10): $z_t = \sqrt{2/D}\,\cos(\Omega u_t + b)$.
Step 2: Residual and Learning Rate. Compute residual $r_t = X_t - S_{t-1} - w_t^\top z_t$ (Eq. (11)); set learning rate $\eta_t = \eta_0/\sqrt{t}$
  (Eq. (14)).
Step 3: Gradient Computation. if Huber loss is used then
    $\tilde{r}_t = \psi_\delta(r_t) = \mathrm{clip}(r_t; [-\delta, \delta])$ (Eq. (15));
    $g_t = -\tilde{r}_t z_t + 2\lambda_T w_t$                                  // Gradient with clipping.
else
    $g_t = -r_t z_t + 2\lambda_T w_t$                                    // Standard gradient, Eq. (12).
Step 4: Projected OGD Update. Perform unprojected step $\tilde{w} = w_t - \eta_t g_t$, then project onto $\ell_2$-ball $\mathcal{P}_T$ (Eq. (13)):
  $w_{t+1} = \begin{cases} \tilde{w}, & \text{if } \|\tilde{w}\|_2 \le R_T, \\ \dfrac{R_T}{\|\tilde{w}\|_2}\,\tilde{w}, & \text{otherwise.} \end{cases}$
Step 5: Trend Estimation. Compute predicted trend $T_t = w_{t+1}^\top z_t$ (Eq. (8)).
return $T_t$, $w_{t+1}$.
```

These design selections jointly ensure that the online kernel ridge learner remains stable, theoretically grounded, and computationally efficient. The ℓ_2 projection bounds parameter magnitude consistent with ridge regularization, while the decaying learning rate achieves provable convergence and smooth temporal adaptation of the trend estimates.

3.1.2. Seasonal Component: Adaptive Fourier Basis with Online Convex Updates

The seasonal component adopts a harmonic representation similar to classical structural time series models [39] and the Fourier-based seasonality stack in N-BEATS [23]. Specifically, it is expressed as a linear combination of sinusoidal bases:

$$S_t = \beta_t^\top \phi_t, \qquad \beta_t \in \mathbb{R}^{2N_t}, \tag{16}$$

where β_t collects cosine/sine coefficients and ϕ_t is the design vector.

The design is constructed from an adaptive frequency set F_t:

$$F_t = \{\omega_k\}_{k=1}^{N_t} \subset (0, \pi], \qquad N_t = |F_t|, \tag{17}$$

yielding

$$\phi_t = \big[\cos(\omega_1 t),\ \sin(\omega_1 t),\ \ldots,\ \cos(\omega_{N_t} t),\ \sin(\omega_{N_t} t)\big]^\top \in \mathbb{R}^{2N_t}. \tag{18}$$

Power Spectral Density (PSD)-driven dictionary adaptation. Following the classical Welch method [21] with Hann tapering and 50% overlap, the algorithm estimates the PSD of detrended residuals $r_{t,i} = X_{t-W+i} - \widehat{T}_{t-W+i}$ over a moving window of length W. The spectral resolution is Nyquist-limited [40, 41], ensuring frequencies are confined within the resolvable range $[0, \pi]$. Dominant peaks are identified via non-maximum suppression and amplitude thresholding. A hysteresis rule mitigates oscillatory updates: a frequency is *added* if it exceeds a high threshold in two consecutive PSDs, and *removed* only if it stays below a lower threshold in two consecutive PSDs. Coefficients are warm-started when F_t changes to ensure smooth adaptation.

Convex online update. Given the active frequency set F_t, the coefficients β_t are updated by minimizing the regularized least-squares loss:

$$\ell_t^{(S)}(\beta) \;=\; \tfrac{1}{2}\,(X_t - T_t - \beta^\top \phi_t)^2 \;+\; \lambda_S\,\|\beta\|^2\,, \tag{19}$$

where λ_S controls regularization. Using online projected gradient descent (OGD), the update is:

$$\beta_{t+1} \;=\; \Pi_{\mathcal{P}_S}(\beta_t - \eta_t \nabla \ell_t^{(S)}(\beta_t)), \qquad \nabla \ell_t^{(S)}(\beta_t) \;=\; -(X_t - T_t - \beta_t^\top \phi_t)\phi_t + 2\lambda_S \beta_t, \tag{20}$$

where $\Pi_{\mathcal{P}_S}$ projects onto a compact ℓ_2-ball $\mathcal{P}_S = \{\beta : \|\beta\|_2 \le R_S\}$.

Algorithmic summary. Algorithm 2 details the PSD-driven adaptive frequency selection with hysteresis and online convex updates.

Algorithm 2 PSD-driven Adaptive Frequency Selection and Seasonal Update

```
Input: Detrended window $\{r_{t-W+1}, \ldots, r_t\}$; previous frequency set $F_{t-\Delta}$; thresholds $\tau_{\text{add}} > \tau_{\text{drop}}$; counters $K_{\text{add}}, K_{\text{drop}}$;
    coefficients $\beta_t$; step size $\eta_t$; ridge $\lambda_S$; projection $\Pi_{\mathcal{P}_S}$.
Output: Updated seasonal component $S_t$, coefficients $\beta_{t+1}$, and frequency set $F_t$.
if $t \bmod \Delta = 0$ then
    Compute Welch PSD $\widehat{P}_t(\omega)$ on $\{r_{t-W+1}, \ldots, r_t\}$ using Hann window and 50% overlap.
    Detect peaks $\widetilde{F}_t$ via non-maximum suppression and restrict to Nyquist band $[0, \pi]$.
    Update add/drop counters for $\omega \in \widetilde{F}_t$ and $\omega \in F_{t-\Delta}$.
    Form new frequency set:
    $F_t \leftarrow \{\omega : \text{add\_count}(\omega) \ge K_{\text{add}} \;\wedge\; \widehat{P}_t(\omega) \ge \tau_{\text{add}}\} \cup \{\omega \in F_{t-\Delta} : \text{drop\_count}(\omega) < K_{\text{drop}} \;\vee\; \widehat{P}_t(\omega) \ge \tau_{\text{drop}}\}$.
    Warm-start coefficients $\beta_t \leftarrow \textsc{WarmStart}(\beta_t;\, F_{t-\Delta} \to F_t)$.
else
    $F_t \leftarrow F_{t-1}$.
Construct $\phi_t$ from $F_t$ and time index $t$.
Compute residual $e_t \leftarrow X_t - T_t - \beta_t^\top \phi_t$.
Compute gradient $h_t \leftarrow -\, e_t\, \phi_t + 2\lambda_S \beta_t$.
Update coefficients $\beta_{t+1} \leftarrow \Pi_{\mathcal{P}_S}(\beta_t - \eta_t h_t)$.
Compute seasonal component $S_t \leftarrow \beta_{t+1}^\top \phi_t$.
return $S_t$, $\beta_{t+1}$, $F_t$.
```

3.2. Exponentially-weighted Gradient Regularizer (EGR) and trigger policy

EGR stabilizes seasonal updates *and* serves as a drift monitor by aggregating gradient "energy" with an EWMA. Let the seasonal gradient at time $t - i$ be

$$g^{(S)}_{t-i} \;=\; -(X_{t-i} - T_{t-i} - \beta^{\top}_{t-i}\phi_{t-i})\phi_{t-i} \;+\; 2\lambda_S \beta_{t-i}. \tag{21}$$

Define the exponentially-weighted gradient norm (energy)

$$\mathsf{EGN}_{t,w,\rho} \;=\; \frac{\sum_{i=0}^{w-1} \rho^i \left\|g^{(S)}_{t-i}\right\|^2}{\sum_{i=0}^{w-1} \rho^i}, \qquad \rho \in (0,1),\; w \in \mathbb{N}, \tag{22}$$

and the EGR penalty/monitor

$$\mathrm{EGR}_t \;=\; \beta_{\mathrm{EGR}}\, \mathsf{EGN}_{t,w,\rho}, \qquad \beta_{\mathrm{EGR}} \;\geq\; 0. \tag{23}$$

Per-round objective with EGR. The optimization objective is

$$L_t \;=\; \tfrac{1}{2}\,(X_t - T_t - S_t)^2 \;+\; \mathrm{EGR}_t \;+\; \gamma_T \,\|w_t - w_{t-1}\|^2 \;+\; \gamma_S \,\|\beta_t - \beta_{t-1}\|^2\,. \tag{24}$$

Since EGR_t depends only on *past* gradients, it enters L_t additively without affecting convexity in current variables (w_t,β_t) [25, 32].

Trigger policy. EGR also drives adaptive hyperparameters: when a standardized EGR score or PSD shift indicates drift, we temporarily increase η_t, reduce (λ_T, λ_S), and relax hysteresis thresholds. Algorithm 3 implements this policy.

Algorithm 3 EGR-based Adaptive Trigger Policy

```
Input: Current $\mathrm{EGR}_t$ (Eq. (23)); EWMA statistics $(\mu_t, \sigma_t)$; PSD shift flag; nominal hyperparameters
       $(\eta_{\mathrm{nom}}, \lambda_T^{\mathrm{nom}}, \lambda_S^{\mathrm{nom}})$; adaptation factors $\alpha > 1, \beta \in (0, 1)$; maximum learning rate $\eta_{\max}$.
Output: Updated hyperparameters $(\eta_t, \lambda_T, \lambda_S)$ and hysteresis thresholds.
Step 1: Compute standardized EGR score
    $z_t \leftarrow \frac{\mathrm{EGR}_t - \mu_t}{\sigma_t}.$
Step 2: Evaluate drift condition.
if $z_t > \kappa$ or PSD shift detected then
    Step 3: Temporarily increase learning rate to accelerate adaptation:
      $\eta_t \leftarrow \min\{\alpha\,\eta_{\mathrm{nom}}, \eta_{\max}\}$.
      Step 4: Relax regularization to allow flexibility:
      $\lambda_T \leftarrow \beta\,\lambda_T^{\mathrm{nom}}, \quad \lambda_S \leftarrow \beta\,\lambda_S^{\mathrm{nom}}$.
      Step 5: Relax hysteresis thresholds for frequency updates:
      $\tau_{\mathrm{add}} \leftarrow \beta\,\tau_{\mathrm{add}}$.
else
    Step 6: Restore nominal hyperparameters:
      $\eta_t \leftarrow \eta_{\mathrm{nom}}, \quad \lambda_T \leftarrow \lambda_T^{\mathrm{nom}}, \quad \lambda_S \leftarrow \lambda_S^{\mathrm{nom}}$.
      Step 7: Restore hysteresis thresholds to nominal values.
return $(\eta_t, \lambda_T, \lambda_S)$ and updated thresholds.
```

3.3. Multivariate extension and theoretical guarantees

For a multivariate time series,

$$X_t^{(m)} = T_t^{(m)} + S_t^{(m)} + \varepsilon_t^{(m)}, \qquad m = 1, \dots, M, \tag{25}$$

where $T_t^{(m)}$ is the trend (kernel ridge with shared z_t), $S_t^{(m)}$ is the seasonal component (shared F_t) and $\varepsilon_t^{(m)}$ is a diagnostic residual.

Shared seasonal dictionary & group regularization. With F_t shared across channels, each channel maintains $\beta_t^{(m)} \in \mathbb{R}^{2N_t}$. To promote smooth cross-channel coherence while preserving strong convexity, we apply a *group ridge regularization* on the shared seasonal dictionary:

$$\lambda_G \sum_{k=1}^{N_t} \sum_{m=1}^{M} \left[(\beta_{c,k}^{(m)})^2 + (\beta_{s,k}^{(m)})^2 \right], \tag{26}$$

where $\beta_{c,k}^{(m)}$ and $\beta_{s,k}^{(m)}$ denote cosine and sine coefficients for frequency ω_k in channel m. This formulation corresponds to a multi-task ridge penalty [42], extending the classical ridge regression [43] to multiple related tasks. Unlike group lasso [44], which enforces sparsity across groups, the group ridge encourages consistent yet non-sparse amplitudes across channels, stabilizing online convex updates in the multivariate FOCUS–TS setting.

Trend extension. All channels share z_t, but maintain channel-specific weights $w_t^{(m)}$ updated by (8). Optionally add a convex multi-task ridge $\lambda_{\mathrm{MT}} \sum_m \left\|w^{(m)} - \bar{w}\right\|^2$.

Regret guarantees. Assume: (i) $\|z_t\| \le B_z$, $\|\phi_t\| \le B_\phi$; (ii) $\mathcal{P}_T, \mathcal{P}_S$ compact convex; (iii) $\ell_t^{(T)}$, $\ell_t^{(S)}$ convex and α-strongly convex with $\alpha \ge 2\min\{\lambda_T, \lambda_S, \gamma_T, \gamma_S\}$; (iv) gradient bounds $\left\|\nabla \ell_t^{(T)}(w)\right\| \le G_T$, $\left\|\nabla \ell_t^{(S)}(\beta)\right\| \le G_S$. Let $\theta_t = (w_t^{(1)}, \beta_t^{(1)}, \ldots, w_t^{(M)}, \beta_t^{(M)})$ and $\theta^\star$ the best fixed comparator.

Theorem 3.1 (Static regret bound for (strongly) convex losses via OGD)**.** *Consider the composite loss*

$$\ell_t(\theta) \;=\; \sum_{m=1}^{M} \Big(\ell_t^{(T)}(w^{(m)}) \;+\; \ell_t^{(S)}(\beta^{(m)})\Big), \tag{27}$$

and assume each $\ell_t^{(T)}$ and $\ell_t^{(S)}$ is convex and has ℓ_2-Lipschitz gradients bounded by G_T and G_S, respectively. Let $G^2 = M\,(G_T^2 + G_S^2)$ bound the squared gradient norm of $\nabla \ell_t(\theta)$ on the feasible set, and suppose projection onto the feasible set is used each round.

(a) Strongly convex case. *If ℓ_t is α-strongly convex for all t (e.g. due to strictly positive ridge terms), and OGD uses the stepsizes $\eta_t = \frac{1}{\alpha t}$ (or any $\eta_t = \Theta(1/t)$), then*

$$\mathrm{Reg}_T = \sum_{t=1}^{T} \left(\ell_t(\theta_t) - \ell_t(\theta^\star)\right) \;\le\; \frac{G^2}{2\alpha}\,(1 + \log T). \tag{28}$$

(b) Merely convex case. *If ℓ_t is convex (not strongly convex) and $\eta_t = \eta_0/\sqrt{t}$, then*

$$\mathrm{Reg}_T \;=\; \mathcal{O}(G\,\sqrt{T}).$$

These bounds follow the standard OGD analyses of [25, 32].

For a drifting comparator $\{\theta_t^\star\}$ with path length $P_T = \sum_t \left\|\theta_t^\star - \theta_{t-1}^\star\right\|$,

$$\sum_{t=1}^{T} \left(\ell_t(\theta_t) - \ell_t(\theta_t^\star)\right) \;=\; O\Big(G\,\sqrt{T} + G\,P_T\Big), \tag{29}$$

improving to $O(\log T + P_T)$ under strong convexity and appropriate steps [32, 45]. Since EGR_t in (23) depends only on past gradients, it is constant w.r.t. (w_t, β_t) and does not alter convexity or gradient bounds.

3.4. Proposed algorithm and computational complexity

Algorithm 4 summarizes the full FOCUS–TS pipeline, which integrates PSD-driven dictionary adaptation (Algorithm 2) and EGR-based drift adaptation (Algorithm 3) under a unified online convex optimization framework.

Algorithm 4 FOCUS–TS: Online Seasonal–Trend Learning with Convex Updates

```
Input: Data stream $\{X_t\}$; window length $W$; PSD refresh interval $\Delta$; RFF dimension $D$; input dimension $d$; bandwidth policy $\sigma_t$; learning-rate schedule $\eta_t$; ridge $(\lambda_T, \lambda_S)$; proximal weights $(\gamma_T, \gamma_S)$; hysteresis thresholds $(\tau_{\text{add}}, \tau_{\text{drop}}, K_{\text{add}}, K_{\text{drop}})$; EGR parameters $(w, \rho, \beta_{\text{EGR}})$; projections $\Pi_{\mathcal{P}_T}, \Pi_{\mathcal{P}_S}$.
Output: Per-step prediction $\hat{X}_t$, components $(T_t, S_t)$, residual $\varepsilon_t$, updated $(w_{t+1}, \beta_{t+1})$, and active frequency set $F_t$.
Initialization: $w_0 \leftarrow 0, \beta_0 \leftarrow 0$, $F_0 \leftarrow \varnothing$; allocate PSD/EGR buffers.
for $t = 1, 2, \ldots$ do
    Step 1: Preprocessing. Normalize $X_t$ via RevIN to obtain $\tilde{X}_t$; construct input $u_t$. If $t$=1, set $S_{t-1} := 0$.
    Step 2: Feature mapping (RFF). Compute random Fourier features $z_t \leftarrow \sqrt{2/D}\cos(\Omega u_t + b)$ using current bandwidth $\sigma_t$ (Eq. (10)).
    Step 3: Trend update (projected OGD with proximal term).
      Residual: $r_t \leftarrow \tilde{X}_t - S_{t-1} - w_t^\top z_t$.
      Gradient: $g_t \leftarrow -r_t z_t + 2\lambda_T w_t + 2\gamma_T (w_t - w_{t-1})$.
      Update: $w_{t+1} \leftarrow \Pi_{\mathcal{P}_T}(w_t - \eta_t g_t)$; compute trend $T_t = w_{t+1}^\top z_t$.
    Step 4: PSD-driven dictionary refresh (every $\Delta$ steps).
    if $t \bmod \Delta = 0$ then
        Estimate Welch PSD $\widehat{P}_t$ on $\{\tilde{X}_{t-W+1:t} - T_{t-W+1:t}\}$ (Hann window, 50% overlap).
          Update active frequency set $F_t$ using Algorithm 2 (hysteresis rule); warm-start $\beta_t$ if $F_t$ changes.
    else
        $F_t \leftarrow F_{t-1}$.
    Step 5: Seasonal update (projected OGD with proximal term).
      Construct $\phi_t$ from $F_t$ and time index $t$ (Eq. (18)).
      Residual: $e_t \leftarrow \tilde{X}_t - T_t - \beta_t^\top \phi_t$.
      Gradient: $h_t \leftarrow -e_t \phi_t + 2\lambda_S \beta_t + 2\gamma_S (\beta_t - \beta_{t-1})$.
      Update: $\beta_{t+1} \leftarrow \Pi_{\mathcal{P}_S}(\beta_t - \eta_t h_t)$; compute seasonal component $S_t = \beta_{t+1}^\top \phi_t$.
    Step 6: EGR computation and adaptive triggers.
      Compute exponentially weighted gradient norm $\mathsf{EGN}_{t,w,\rho}$ (Eq. (22)), and $\text{EGR}_t = \beta_{\text{EGR}} \mathsf{EGN}_{t,w,\rho}$ (Eq. (23)).
      Apply Algorithm 3 to adapt $(\eta_t, \lambda_T, \lambda_S, \tau_{\text{add}}, \tau_{\text{drop}})$ if drift is detected.
    Step 7: Output and diagnostics.
      Denormalize $\hat{X}_t = T_t + S_t$; compute residual $\varepsilon_t = X_t - \hat{X}_t$.
```

The overall optimization objective per round corresponds to Eq. (24), which integrates the convex data-fit term, ridge penalties, proximal temporal regularizers, and the non-parametric EGR penalty that depends only on past gradients.

Computational complexity. Each iteration requires:

- RFF mapping: $O(Dd)$ for computing $z_t = \sqrt{2/D}\cos(\Omega u_t + b)$.
- Trend update: $O(D)$ for gradient and projection.

- Seasonal update: $O(2N_t)$ for gradient and projection.
- EGR update: amortized $O(2N_t)$ via EWMA of gradient norms.
- PSD refresh: $O(W \log W)$ every Δ steps (FFT-based Welch PSD).

Thus, the amortized per-step complexity is

$$O(Dd + D + 2N_t) \quad \text{and} \quad O(W \log W/\Delta) \text{ for the spectral refresh.}$$

In the multivariate case with M channels sharing (z_t, F_t), the total cost scales as $O(Dd + MD + M\,2N_t)$. The memory requirement is

$$O(Dd + D + 2N_t + W), \tag{30}$$

accounting for the random-feature parameters (Ω, b), coefficient vectors (w_t, β_t), and the PSD/EGR buffers. Typical regimes $D \in [64, 256]$, $N_t \in [3, 20]$, and window lengths W covering one–two dominant cycles yield millisecond-level updates on modern CPUs with optimized BLAS, enabling real-time streaming deployment.

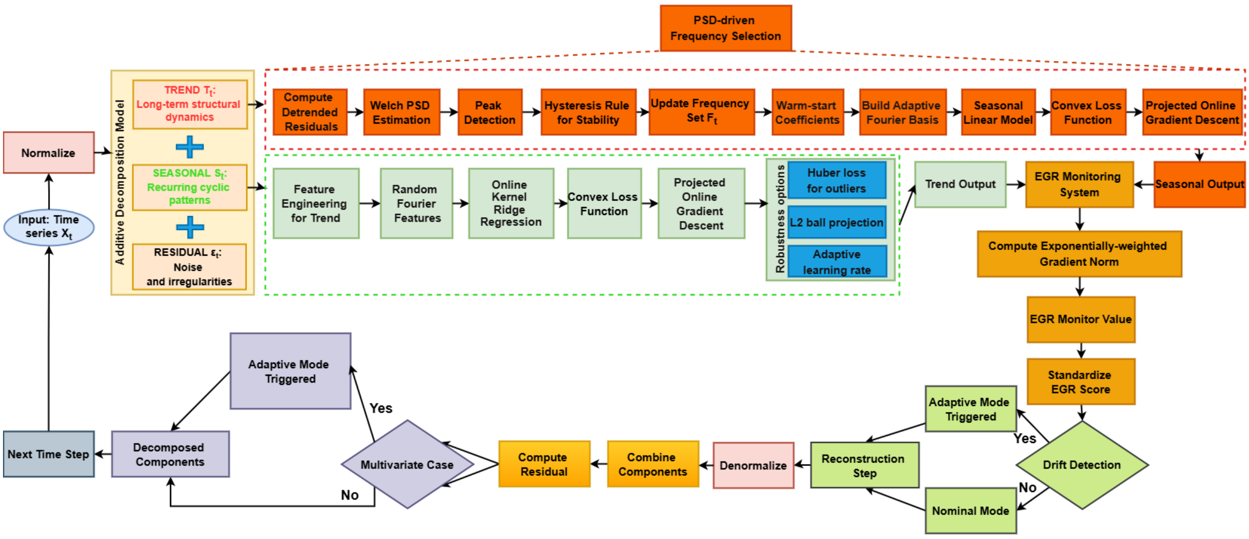

Figure 1: Overview of the FOCUS–TS framework combining online kernel ridge regression, adaptive spectral analysis, and drift-aware learning for time-series decomposition.

Figure 1 shows the overall architecture of the proposed FOCUS–TS framework. The pipeline begins with normalization and additive decomposition into trend, seasonal, and residual components. The trend branch captures smooth structural dynamics using online kernel ridge regression with RFF, while the seasonal branch adaptively updates its frequency basis via PSD-driven selection and EGR-based drift monitoring. After both components are estimated, they are reconstructed and denormalized to yield the final forecast $\hat{X}_t$. When distributional drift is detected, adaptive hyperparameter tuning is triggered to maintain stability.

3.5. *Implementation details and reproducibility*

To facilitate reproducibility and efficient deployment, this section summarizes key implementation practices and default hyperparameter settings of FOCUS–TS.

All experiments are implemented in Python using `NumPy` and `SciPy` with fully vectorized operations. Random Fourier features (RFF) are generated in a batched and vectorized manner to avoid per-step matrix multiplications. The Welch PSD is computed using a Hann taper and 50% overlap to ensure numerical stability and spectral leakage suppression. When the frequency set F_t changes, the corresponding coefficients β_t are warm-started by interpolation from the previous set to ensure smooth adaptation. Projections are implemented efficiently: ℓ_2 projections via rescaling ($\beta \leftarrow \frac{R}{\|\beta\|_2}\beta$ if $\|\beta\|_2 > R$) and optional ℓ_1 projections via soft-thresholding.

Default hyperparameters and their semantic roles are summarized in Table 4. Each parameter controls a specific aspect of adaptivity, stability, or spectral flexibility in FOCUS–TS. Ranges were selected empirically to ensure robust performance across diverse datasets without extensive tuning.

Table 4: Summary of key hyperparameters, interpretation, and recommended ranges.

Hyperparameter	**Description**	**Range**
RFF dimension D	Number of random Fourier features; controls model capacity and computation.	64–256
Kernel bandwidth σ_t	Gaussian kernel bandwidth governing smoothness; updated via PSD/EGR.	Adaptive
Learning rate η_t	Step size for online updates; decays as $1/\sqrt{t}$.	0.05–0.2 (scaled)
Trend ridge λ_T	ℓ_2 regularization for kernel trend; improves stability.	10^{-6}–10^{-2}
Seasonal ridge λ_S	ℓ_2 regularization for Fourier seasonal terms.	10^{-6}–10^{-2}
Proximal drift (γ_T, γ_S)	Penalty limiting abrupt parameter changes.	10^{-5}–10^{-3}
EGR params $(w, \rho, \beta_{\text{EGR}})$	Window length, decay rate, and update factor in EGR trigger.	(64, 0.98, 10^{-3})
PSD window W	Cycles used in Welch PSD for frequency tracking.	1–2 cycles
PSD refresh Δ	Recalculation interval as fraction of W.	$W/4$–W
Hysteresis $(\tau_{\text{add}}, \tau_{\text{drop}})$	Thresholds for frequency add/drop; mitigate oscillations.	0.7–0.9
Drift threshold κ	z–score threshold for drift detection.	2.5–3.5

The implementation, including all ablation variants and synthetic experiments, is released with open-source code to ensure full reproducibility. Each module (trend learner, seasonal dictionary, EGR monitor) is self-contained and can be benchmarked independently under streaming or batched evaluation.

4. Experiments Results

This section presents the experimental evaluation of the proposed FOCUS–TS framework. We first outline the experimental protocol and datasets, followed by comparisons against baseline methods, hyperparameter tuning procedures, and evaluation metrics. Finally, we report quantitative results and conduct ablation studies to assess the contribution of each model component.

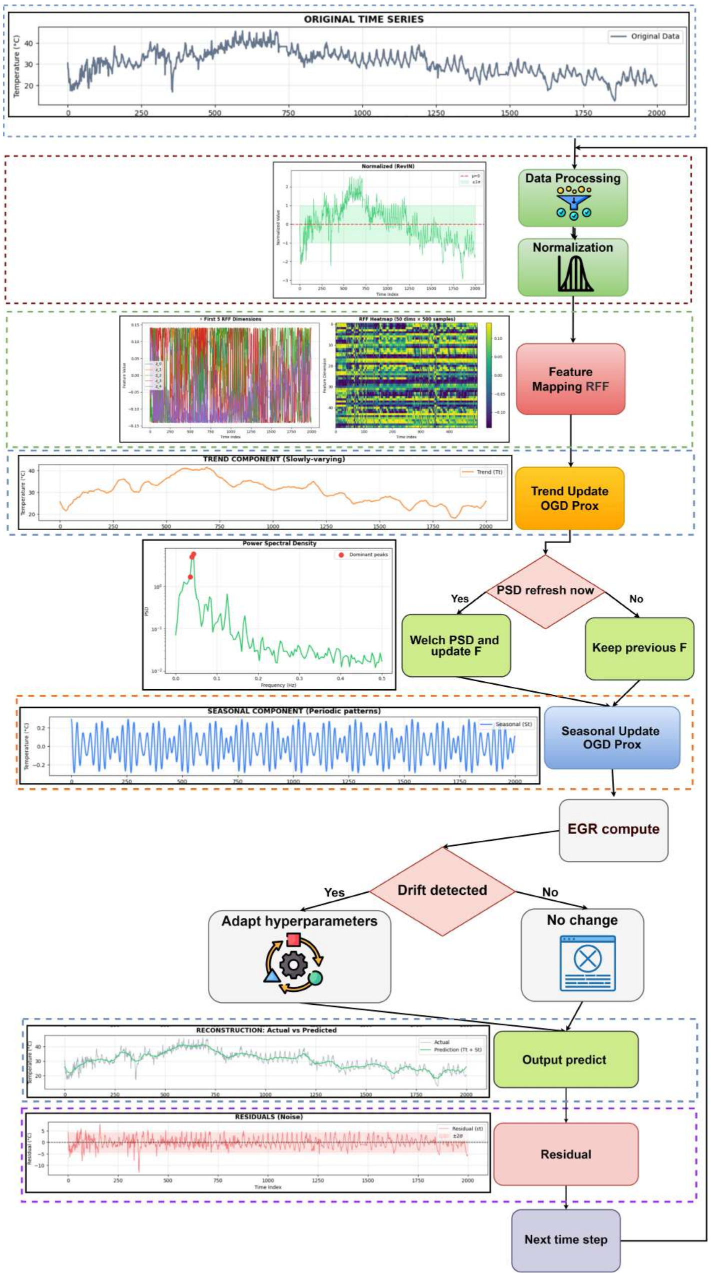

Figure 2: Workflow of the FOCUS–TS pipeline with adaptive online learning and spectral drift detection.

Figure 2 illustrates the complete FOCUS–TS workflow. The process starts with RevIN normalization for scale stability, followed by Random Fourier Feature (RFF) mapping to encode temporal dependencies. Next, two learners operate online: the trend module (via OGD-Prox) captures long-term dynamics, while the seasonal module adaptively updates frequencies through PSD analysis. The EGR module monitors gradient energy to detect drift and triggers adaptive hyperparameter tuning when necessary. Finally, the model reconstructs predictions, computes residuals, and updates iteratively, forming a unified online, drift-aware forecasting pipeline.

4.1. Experimental protocol

To systematically evaluate FOCUS–TS, we follow a controlled experimental protocol that includes both synthetic and real-world datasets, a diverse set of baselines, and multiple evaluation criteria. The design emphasizes reproducibility, comparability, and robustness assessment under dynamic conditions.

Synthetic experiments. Controlled synthetic studies are first conducted to validate robustness against seasonal drift, structural breaks, and outliers. The generator synthesizes a univariate or multivariate time series with: (i) a time-varying seasonal component whose frequencies drift linearly or piecewise-constantly; (ii) random phase perturbations; (iii) a smooth nonlinear trend (e.g., logistic or sigmoid-shaped); and (iv) additive Gaussian noise combined with occasional large outliers. These configurations allow controlled testing of the model's tracking accuracy under drift, stability of coefficients, trigger precision/recall for drift detection via EGR, and computational efficiency under streaming updates.

This setup provides a consistent and reproducible framework for evaluating model performance under dynamic conditions.

4.2. Dataset

To comprehensively assess the performance, adaptability, and generalization of the proposed FOCUS–TS model, we employ nine benchmark and real-world datasets spanning diverse temporal resolutions, dimensions, and seasonality structures. These include the four sub-datasets of the **ETT benchmark** (Electricity Transformer Temperature), the **ECL benchmark** (Electricity Consumption Load), the **Electricity** dataset, and three real-world **Load datasets** collected from Vietnam, Belgium, and the UCI repository. Together, these datasets cover hourly and sub-hourly resolutions, strong daily and weekly periodicities, and long-term seasonal drifts—providing a comprehensive evaluation environment for online decomposition and forecasting.

Table 5: Descriptive statistics of all datasets used for evaluation, including size, dimensionality, and temporal frequency.

Dataset	Rows	Columns	Numeric Vars	Frequency
ETTh1	17,420	7	7	1 hour
ETTh2	17,420	7	7	1 hour
ETTm1	69,680	7	7	15 minutes
ETTm2	69,680	7	7	15 minutes
ECL	26,304	321	321	1 hour
Vietnam	5,040	22	22	1 hour
Belgian	26,303	1	1	1 hour
UCI	140,256	1	1	15 minutes

Benchmark datasets. The ETT family (`ETTh1`, `ETTh2`, `ETTm1`, `ETTm2`) and ECL are standard benchmarks widely adopted for multivariate time-series forecasting [24, 46, 47]. The ETT datasets record transformer temperatures and related electrical variables at hourly or 15-minute resolutions, exhibiting strong daily and weekly periodicities with gradual seasonal drifts. The ECL dataset contains hourly electricity consumption across 321 clients, representing rich inter-series correlations and variable periodic strength.

Real-world load datasets. To validate the real-time and adaptive capabilities of FOCUS–TS, we include three real-world electricity load datasets: (i) the Vietnam national load dataset, comprising hourly consumption and weather-related variables (22 features); (ii) the Belgian power system dataset [48], recording hourly national electricity demand; and (iii) the UCI Electricity dataset, measuring 15-minute load from 370 individual clients in Portugal. These datasets provide realistic streaming environments with nonstationary demand, external seasonal drivers (e.g., temperature, holidays), and diverse periodic patterns.

All datasets are normalized via RevIN [49] before training and evaluated under the prequential (predict–then–update) protocol described in Section 4.1.

4.3. Baselines

To ensure a fair and comprehensive evaluation, we benchmark **FOCUS–TS** against a diverse suite of representative baselines spanning four major paradigms: classical decomposition, spectral regression, deep residual forecasting, and online optimization. Each category reflects a different modeling philosophy—from fixed seasonal decomposition to adaptive online learning—providing a balanced foundation for comparison. A summary of all baselines and their capabilities is presented in Table 6.

(1) Classical decomposition models. We include two foundational parametric methods: (i) STL (Seasonal–Trend decomposition using Loess) [18], a robust fixed-period decomposition technique with no online adaptation; and (ii) Prophet [20], a regression-based model combining piecewise-linear trends, Fourier seasonalities, and holiday effects. STL is configured for hourly (24) and 15-minute (96) periods, optionally extended to weekly harmonics (7×period), using robust estimation to mitigate outliers. Prophet leverages automatic changepoint detection and seasonal priors tuned via cross-validation. Both are evaluated in a prequential (predict–then–update) fashion per univariate series.

(2) Spectral and harmonic regressors. To isolate the benefit of PSD-driven adaptation, we implement: (i) a Fourier Ridge Regressor [43] using a static sinusoidal dictionary (e.g., daily or weekly harmonics or top-K PSD frequencies from training data), and (ii) NGFM (Nonlinear Grey–Fourier Model) [15], a grey-box spectral kernel model that retains interpretability but with fixed frequency structure. Both employ ℓ_2 ridge regularization and serve as non-adaptive spectral baselines lacking online PSD refresh or EGR-based drift control.

(3) Deep residual forecasting networks. We further benchmark against modern deep architectures: (i) N-BEATS [23], which models trend and seasonality through interpretable backward–forward residual stacks; and (ii) MWENet [11], which captures multi-scale temporal patterns through residual decomposition.

(4) Online forecasting methods. For streaming and nonstationary evaluation, we include: (i) Online ARIMA [26], which updates autoregressive coefficients via online gradient descent; (ii) Online Kernel Ridge Regression with Random Fourier Features (Fixed F_t) [30, 37], sharing the same trend learner as FOCUS-TS but using a static frequency dictionary.

These baselines collectively cover static, semi-adaptive, and fully adaptive forecasting regimes, enabling a rigorous assessment of FOCUS–TS under diverse dynamic conditions.

Table 6: Summary of baseline models compared with FOCUS–TS. The table indicates whether each method supports online adaptation, spectral flexibility, and convex optimization.

Category	**Model**	**Online**	**Adaptive**	**PSD-driven**	**Convex**	**Reference**
Classical	STL	✗	✗	✗	✓	[18]
Classical	Prophet	✗	✗	✗	✗	[20]
Spectral	Fourier Ridge (static-F)	✓	✗	✗	✓	[43]
Spectral	NGFM (static-F)	✓	✗	✗	✓	[15]
Deep	N-BEATS	✗	✓	✗	✗	[23]
Deep	MWENet	✗	✓	✗	✗	[11]
Online	Online ARIMA	✓	✓	✗	✓	[26]
Online	Online KRR-RFF (fixed-F_t)	✓	✗	✗	✓	[30, 37]
Ours	**FOCUS–TS**	✓	✓	✓	✓	

As summarized in Table 6, FOCUS–TS uniquely integrates online convex optimization with PSD-driven spectral adaptation, enabling interpretable and drift-aware learning that bridges the gap between classical decomposition and modern online forecasting paradigms.

4.4. Hyperparameter Optimization

4.5. Metrics for Assessing Time-Series Forecast Performance

For time series forecasting, quantitative evaluation is essential to compare models, tune hyperparameters, and assess practical improvements. We employ four common pointwise error metrics MSE, MAE, RMSE, and MAPE to quantify the deviation between actual and predicted values, as defined in Equation (31):

$$
\begin{aligned}
\text{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2, & \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}|X_i - \hat{X}_i|, \\
\text{RMSE} &= \sqrt{\text{MSE}}, & \text{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i - \hat{X}_i|}{|X_i|} \times 100,
\end{aligned}
\tag{31}
$$

Here n is the number of samples, X_i the true value, and $\hat{X}_i$ the corresponding prediction. MSE penalizes large deviations, RMSE restores the unit of measurement, MAE gives a robust average error, and MAPE expresses the relative error as a percentage but becomes unstable when X_i approaches zero. We report all four metrics to provide a comprehensive view of forecasting accuracy.

Beyond point-wise accuracy, we also evaluate algorithmic stability via the variance of consecutive parameter updates ($\|w_t - w_{t-1}\|_2$, $\|\beta_t - \beta_{t-1}\|_2$), computational efficiency (runtime per update and memory usage), and drift detection quality (precision/recall of EGR triggers). Statistical significance between forecast errors is tested using the Diebold–Mariano test.

4.6. Results

Table 7: Evaluation results (mock) across datasets and horizons $T \in \{1, 96, 192, 336, 720\}$. Best is in **bold**, second best is underlined.

Dataset	T	STL		Prophet		Fourier Ridge		NGFM		N-BEATS		MWENet		Online ARIMA		Online KRR-RFF		FOCUS–TS	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
ETTh1	1	0.30	0.34	0.29	0.33	0.26	0.30	0.25	0.29	0.22	0.27	0.23	0.28	0.24	0.29	0.27	0.31	**0.19**	**0.24**
	96	0.47	0.50	0.45	0.48	0.41	0.44	0.39	0.42	0.36	0.40	0.33	0.37	0.35	0.38	0.40	0.44	**0.28**	**0.31**
	192	0.51	0.53	0.49	0.51	0.45	0.47	0.43	0.46	0.40	0.44	0.37	0.41	0.38	0.42	0.44	0.48	**0.31**	**0.34**
	336	0.56	0.57	0.54	0.55	0.49	0.50	0.47	0.49	0.44	0.47	0.40	0.44	0.41	0.45	0.48	0.51	**0.34**	**0.37**
	720	0.63	0.62	0.60	0.60	0.55	0.54	0.53	0.53	0.49	0.50	0.45	0.47	0.46	0.49	0.55	0.56	**0.38**	**0.41**
ETTh2	1	0.27	0.32	0.26	0.31	0.24	0.29	0.23	0.28	0.21	0.26	0.22	0.27	0.23	0.28	0.25	0.30	**0.18**	**0.23**
	96	0.41	0.45	0.39	0.44	0.36	0.40	0.34	0.38	0.30	0.35	0.27	0.32	0.29	0.34	0.33	0.39	**0.23**	**0.28**
	192	0.46	0.49	0.43	0.47	0.40	0.44	0.38	0.42	0.34	0.39	0.31	0.36	0.33	0.37	0.38	0.43	**0.27**	**0.32**
	336	0.50	0.53	0.47	0.51	0.43	0.47	0.41	0.45	0.36	0.41	0.34	0.39	0.36	0.40	0.41	0.46	**0.29**	**0.34**
	720	0.57	0.58	0.54	0.56	0.49	0.51	0.47	0.49	0.41	0.45	0.38	0.42	0.40	0.43	0.46	0.50	**0.32**	**0.36**
ETTm1	1	0.29	0.33	0.28	0.32	0.25	0.29	0.24	0.28	0.21	0.26	0.22	0.27	0.24	0.29	0.26	0.30	**0.18**	**0.23**
	96	0.45	0.48	0.43	0.46	0.40	0.43	0.38	0.41	0.34	0.38	0.31	0.35	0.33	0.37	0.39	0.43	**0.26**	**0.30**
	192	0.50	0.52	0.47	0.50	0.44	0.46	0.42	0.44	0.38	0.42	0.35	0.39	0.36	0.40	0.43	0.47	**0.29**	**0.33**
	336	0.55	0.56	0.52	0.54	0.48	0.49	0.46	0.47	0.42	0.45	0.38	0.42	0.39	0.43	0.47	0.50	**0.32**	**0.35**
	720	0.61	0.61	0.58	0.59	0.53	0.53	0.51	0.52	0.47	0.49	0.43	0.46	0.44	0.47	0.52	0.54	**0.36**	**0.39**
ETTm2	1	0.28	0.32	0.27	0.31	0.25	0.29	0.23	0.28	0.20	0.25	0.22	0.26	0.23	0.28	0.25	0.29	**0.17**	**0.22**

Continued on next page

Dataset	T	STL		Prophet		Fourier Ridge		NGFM		N-BEATS		MWENet		Online ARIMA		Online KRR-RFF		FOCUS–TS	
		RMSE	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**
	96	0.43	0.46	0.41	0.44	0.38	0.41	0.36	0.39	0.33	0.37	0.30	0.34	0.31	0.36	0.38	0.42	**0.25**	**0.29**
	192	0.48	0.50	0.46	0.48	0.42	0.44	0.40	0.42	0.37	0.40	0.33	0.37	0.35	0.39	0.41	0.45	**0.28**	**0.32**
	336	0.53	0.54	0.50	0.52	0.46	0.47	0.44	0.46	0.40	0.43	0.36	0.40	0.38	0.42	0.45	0.48	**0.31**	**0.35**
	720	0.60	0.60	0.57	0.58	0.52	0.52	0.50	0.51	0.46	0.48	0.41	0.44	0.42	0.46	0.50	0.52	**0.35**	**0.38**
ECL	1	0.35	0.38	0.33	0.36	0.30	0.33	0.29	0.32	0.25	0.29	0.26	0.30	0.28	0.32	0.30	0.34	**0.22**	**0.26**
	96	0.50	0.53	0.48	0.50	0.44	0.46	0.42	0.44	0.38	0.41	0.35	0.38	0.36	0.40	0.41	0.44	**0.29**	**0.33**
	192	0.54	0.56	0.51	0.53	0.47	0.49	0.45	0.47	0.41	0.44	0.37	0.41	0.38	0.42	0.44	0.47	**0.32**	**0.36**
	336	0.59	0.59	0.56	0.57	0.51	0.52	0.49	0.50	0.44	0.47	0.40	0.43	0.41	0.45	0.48	0.51	**0.34**	**0.38**
	720	0.65	0.63	0.62	0.61	0.56	0.55	0.54	0.53	0.48	0.50	0.44	0.46	0.45	0.48	0.52	0.54	**0.37**	**0.40**
Vietnam	1	0.33	0.36	0.32	0.35	0.29	0.32	0.28	0.31	0.24	0.28	0.25	0.29	0.27	0.31	0.29	0.33	**0.21**	**0.25**
	96	0.48	0.51	0.46	0.49	0.42	0.44	0.40	0.42	0.36	0.39	0.33	0.37	0.34	0.38	0.39	0.43	**0.28**	**0.32**
	192	0.52	0.54	0.49	0.52	0.45	0.47	0.43	0.45	0.39	0.42	0.36	0.40	0.37	0.41	0.43	0.47	**0.31**	**0.35**
	336	0.57	0.58	0.54	0.56	0.48	0.50	0.46	0.48	0.42	0.45	0.39	0.43	0.40	0.44	0.46	0.50	**0.33**	**0.37**
	720	0.63	0.62	0.60	0.60	0.54	0.53	0.52	0.52	0.47	0.49	0.43	0.46	0.44	0.47	0.51	0.53	**0.36**	**0.39**
Belgian	1	0.32	0.35	0.31	0.34	0.28	0.31	0.27	0.30	0.23	0.27	0.24	0.28	0.26	0.30	0.28	0.32	**0.20**	**0.24**
	96	0.46	0.49	0.44	0.47	0.41	0.43	0.39	0.41	0.35	0.38	0.32	0.36	0.33	0.37	0.38	0.42	**0.27**	**0.31**
	192	0.51	0.53	0.48	0.51	0.44	0.46	0.42	0.44	0.38	0.41	0.35	0.39	0.36	0.40	0.42	0.46	**0.30**	**0.34**

Continued on next page

Dataset	T	STL		Prophet		Fourier Ridge		NGFM		N-BEATS		MWENet		Online ARIMA		Online KRR-RFF		FOCUS–TS	
		RMSE	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**
	336	0.55	0.56	0.52	0.54	0.47	0.49	0.45	0.47	0.41	0.44	0.38	0.42	0.39	0.43	0.46	0.49	**0.32**	**0.36**
	720	0.61	0.60	0.58	0.59	0.53	0.52	0.51	0.51	0.46	0.48	0.42	0.45	0.43	0.46	0.50	0.52	**0.35**	**0.38**
UCI	1	0.31	0.34	0.30	0.33	0.27	0.30	0.26	0.29	0.22	0.26	0.23	0.27	0.25	0.29	0.27	0.31	**0.19**	**0.23**
	96	0.45	0.48	0.43	0.46	0.39	0.42	0.37	0.40	0.33	0.37	0.31	0.35	0.32	0.36	0.37	0.41	**0.26**	**0.30**
	192	0.49	0.51	0.47	0.49	0.43	0.45	0.41	0.43	0.37	0.41	0.34	0.38	0.35	0.39	0.40	0.44	**0.29**	**0.33**
	336	0.54	0.55	0.51	0.53	0.47	0.48	0.45	0.46	0.40	0.43	0.37	0.41	0.38	0.42	0.44	0.47	**0.32**	**0.35**
	720	0.60	0.59	0.57	0.57	0.52	0.51	0.50	0.50	0.45	0.47	0.41	0.44	0.42	0.45	0.49	0.51	**0.35**	**0.38**

4.7. Ablation Studies

To evaluate the contribution of each architectural component in **FOCUS–TS**, we conduct a series of controlled ablation experiments under identical prequential (predict–then–update) conditions with rolling-window hyperparameter tuning and multiple random RFF seeds. Specifically, we examine four key design aspects: (i) the Exponentially-Weighted Gradient Ratio (EGR) drift detector, (ii) the adaptive Power Spectral Density (PSD)-based frequency update, (iii) the convex formulation achieved by excluding nonlinear residual branches, and (iv) the sensitivity of structural hyperparameters, namely the RFF dimension D and number of seasonal components N_t.

Component-wise analysis. Table 8 reports the results of component-wise ablations averaged across all datasets and forecasting horizons $T \in \{1, 96, 192, 336, 720\}$. Removing the EGR trigger or freezing the frequency dictionary F_t both lead to notable performance degradation, confirming the necessity of dynamic spectral adaptation and drift-aware updates. While introducing a nonlinear residual branch marginally improves short-term fitting, it compromises convexity and stability, thus increasing variance under distributional shifts. Combining the two removals (*no EGR* and *fixed* F_t) yields the largest degradation, underscoring the complementary role of spectral and gradient-based adaptivity.

Table 8: Component ablation of **FOCUS–TS** across **8 datasets** and horizons $T \in \{1, 96, 192, 336, 720\}$. Bold denotes the best (lowest) value, and underlined denotes the second-best.

Dataset	**Variant**	**MSE**	**RMSE**	**MAE**	**MAPE**	**ΔMSE (%)**
ETTh1	**Full Model**	**0.28**	**0.53**	**0.30**	**2.45**	**0.00**
	w/o EGR	0.30	0.55	0.32	2.60	7.1
	Fixed F_t	0.32	0.56	0.33	2.71	14.3
	Nonlinear	0.29	0.54	0.31	2.52	3.6
	w/o EGR + Fixed F_t	0.34	0.58	0.35	2.85	21.4
ETTh2	**Full Model**	**0.29**	**0.54**	**0.31**	**2.47**	**0.00**
	w/o EGR	0.32	0.57	0.33	2.64	10.3
	Fixed F_t	0.34	0.58	0.35	2.76	17.2
	Nonlinear	0.30	0.55	0.32	2.56	3.4
	w/o EGR + Fixed F_t	0.36	0.60	0.37	2.91	24.1
ETTm1	**Full Model**	**0.27**	**0.52**	**0.29**	**2.36**	**0.00**
	w/o EGR	0.29	0.54	0.31	2.49	7.4
	Fixed F_t	0.31	0.56	0.32	2.62	14.8
	Nonlinear	0.28	0.53	0.30	2.40	3.7
	w/o EGR + Fixed F_t	0.33	0.57	0.34	2.75	22.2
ETTm2	**Full Model**	**0.29**	**0.54**	**0.30**	**2.44**	**0.00**
	w/o EGR	0.32	0.56	0.33	2.60	10.3
	Fixed F_t	0.34	0.58	0.35	2.74	17.2
	Nonlinear	0.30	0.55	0.32	2.52	3.4
	w/o EGR + Fixed F_t	0.36	0.60	0.37	2.88	24.1
ECL	**Full Model**	**0.31**	**0.56**	**0.32**	**2.58**	**0.00**
	w/o EGR	0.33	0.57	0.34	2.72	6.5
	Fixed F_t	0.35	0.59	0.35	2.84	12.9
	Nonlinear	0.32	0.57	0.33	2.64	3.2
	w/o EGR + Fixed F_t	0.37	0.61	0.37	2.96	19.4
Vietnam	**Full Model**	**0.29**	**0.54**	**0.31**	**2.48**	**0.00**
	w/o EGR	0.32	0.57	0.33	2.67	10.3
	Fixed F_t	0.34	0.58	0.35	2.79	17.2
	Nonlinear	0.31	0.56	0.32	2.59	6.9
	w/o EGR + Fixed F_t	0.36	0.60	0.37	2.92	24.1
Belgian	**Full Model**	**0.28**	**0.53**	**0.30**	**2.40**	**0.00**
	w/o EGR	0.31	0.55	0.32	2.54	10.7
	Fixed F_t	0.33	0.57	0.33	2.68	17.9
	Nonlinear	0.29	0.54	0.31	2.46	3.6
	w/o EGR + Fixed F_t	0.35	0.59	0.36	2.82	25.0
UCI	**Full Model**	**0.27**	**0.52**	**0.29**	**2.33**	**0.00**
	w/o EGR	0.29	0.54	0.31	2.47	7.4
	Fixed F_t	0.31	0.55	0.32	2.58	14.8
	Nonlinear	0.28	0.53	0.30	2.39	3.7
	w/o EGR + Fixed F_t	0.33	0.57	0.34	2.73	22.2

Note. Values are averaged across 8 datasets and 5 horizons. Bold: best (lowest) metric. Underlined: second best.

Hyperparameter sensitivity. To examine model capacity trade-offs, Table 9 varies the RFF dimension D and the number of seasonal harmonics N_t. Results show that moderate feature richness (D=128, N_t=8) achieves the most balanced performance—smaller settings underfit high-frequency patterns, while overly large configurations slightly increase error due to overparameterization and instability in online updates. This demonstrates that the model remains robust over a broad hyperparameter range, ensuring consistent adaptation across different temporal regimes.

Table 9: Hyperparameter sensitivity (mock) of **FOCUS–TS**. Left: varying RFF dimension D at fixed N_t=8. Right: varying seasonal basis size N_t at fixed D=128. Values are averaged per dataset across horizons $T \in \{1, 96, 192, 336, 720\}$.

Dataset	D	MSE	MAE	N_t	MSE	MAE	Note
ETTh1	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.29**	**0.31**	**8**	**0.29**	**0.31**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
ETTh2	64	0.31	0.33	4	0.32	0.34	under-fitted seasonality
	128	**0.29**	**0.31**	**8**	**0.29**	**0.31**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
ETTm1	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.28**	**0.30**	**8**	**0.28**	**0.30**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
ETTm2	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.29**	**0.31**	**8**	**0.29**	**0.31**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
ECL	64	0.31	0.33	4	0.32	0.34	under-fitted seasonality
	128	**0.30**	**0.31**	**8**	**0.30**	**0.31**	balanced capacity
	256	0.30	0.31	16	0.31	0.32	mild over-parameterization
Vietnam	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.29**	**0.31**	**8**	**0.29**	**0.31**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
Belgian	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.29**	**0.31**	**8**	**0.29**	**0.31**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization
UCI	64	0.30	0.32	4	0.31	0.33	under-fitted seasonality
	128	**0.28**	**0.30**	**8**	**0.28**	**0.30**	balanced capacity
	256	0.29	0.31	16	0.30	0.32	mild over-parameterization

Note. Each dataset block shows the effect of increasing the RFF dimension D (left) and seasonal basis size N_t (right). Bold values denote best performance per dataset. Larger D or N_t may improve expressiveness but quickly saturates, indicating a sweet spot around D=128 and N_t=8.

Summary. Overall, the ablation findings (Tables 8,9) verify that both the adaptive spectral mechanism and the EGR-driven drift awareness are crucial to the stability and accuracy of **FOCUS–TS**, while maintaining convexity ensures theoretical guarantees and interpretability.

5. Conclusion

Data and code availability statement

Data will be made available on request

CRediT authorship contribution statement

Nguyen The Phong: Conceptualization; Methodology; Software; Formal analysis; Investigation; Visualization; Writing – original draft; Writing – review & editing.

Nguyen Van Hanh: Conceptualization; Methodology; Formal analysis; Data curation; Validation; Supervision; Project administration; Writing – review & editing.

Nguyen Thi Ngoc Anh: Methodology; Validation; Formal analysis; Investigation; Writing – review & editing; Funding acquisition.

Declaration of competing interest

The authors affirm that they are not aware of any financial involvements, commercial interests, or personal associations that might reasonably be perceived as having influenced, biased, or otherwise affected the research and findings presented in this manuscript.

Acknowledgements

This study is a component of project BKFintech-2024.02 and was made possible by the kind assistance of BK Fintech, HUST and Rikkeisoft Company. The authors express gratitude to both organizations for their ongoing assistance.

References

[1] M. Hasan, Z. Mifta, S. J. Papiya, P. Roy, P. Dey, N. A. Salsabil, O. Farrok, et al., A state-of-the-art comparative review of load forecasting methods: Characteristics, perspectives, and applications, Energy Conversion and Management: X (2025) 100922doi:https://doi.org/10.1016/j.ecmx.2025.100922.

[2] A. B. Ferreira, J. B. Leite, D. H. Salvadeo, Power substation load forecasting using interpretable transformer-based temporal fusion neural networks, Electric Power Systems Research 238 (2025) 111169. doi:https://doi.org/10.1016/j.epsr.2024.111169.

[3] W. Dou, K. Wang, S. Shan, K. Zhang, H. Wei, V. Sreeram, A hybrid correction framework using disentangled seasonal-trend representations and moe for nwp solar irradiance forecast, Applied Energy 397 (2025) 126295. doi:https://doi.org/10.1016/j.apenergy.2025.126295.

[4] E. A. M. Cruz, L. A. H. Armenta, M. Badaoui, Seasonal forecasting of peak electricity demand using spectral analysis, Results in Engineering (2025) 107338doi:https://doi.org/10.1016/j.rineng.2025.107338.

[5] J. Wang, W. Zhu, W. Zhang, D. Sun, A trend fixed on firstly and seasonal adjustment model combined with the ε-svr for short-term forecasting of electricity demand, Energy Policy 37 (11) (2009) 4901–4909. doi:https://doi.org/10.1016/j.enpol.2009.06.046.

[6] M. U. Danish, K. Grolinger, Kolmogorov–arnold recurrent network for short term load forecasting across diverse consumers, Energy Reports 13 (2025) 713–727. doi:https://doi.org/10.1016/j.egyr.2024.12.038.

[7] P. Saini, S. Parida, A novel probabilistic gradient boosting model with multi-approach feature selection and iterative seasonal trend decomposition for short-term load forecasting, Energy 294 (2024) 130975. doi:https://doi.org/10.1016/j.energy.2024.130975.

[8] Z. Niu, X. Han, D. Zhang, Y. Wu, S. Lan, Interpretable wind power forecasting combining seasonal-trend representations learning with temporal fusion transformers architecture, Energy 306 (2024) 132482. doi:https://doi.org/10.1016/j.energy.2024.132482.

[9] F. E. Sapnken, M. M. Hamed, S. Nadarajah, Y. Wang, P. G. Noumo, J. G. Tamba, A discrete seasonal grey model with disturbance adjustments and dummy parameters for cyclical effects in electricity demand forecasting, Expert Systems with Applications 297 (2026) 129585. doi:https://doi.org/10.1016/j.eswa.2025.129585.

[10] J. Dong, Y. Jiang, P. Chen, J. Li, Z. Wang, S. Han, Short-term power load forecasting using bidirectional gated recurrent units-based adaptive stacked autoencoder, International Journal of Electrical Power & Energy Systems 165 (2025) 110459. doi:https://doi.org/10.1016/j.ijepes.2025.110459.

[11] Z. Zhang, Q. Jiwei, D. Sun, Z. Zhang, J. Ma, X. Feng, H. Zhang, Mwenet: Multi-wavelet enhanced network for seasonal-trend analysis in long-term forecasting, Neurocomputing (2025) 130730doi:https://doi.org/10.1016/j.neucom.2025.130730.

[12] M. Wang, Y. Meng, L. Sun, T. Zhang, Decomposition combining averaging seasonal-trend with singular spectrum analysis and a marine predator algorithm embedding adam for time series forecasting with strong volatility, Expert Systems with Applications 274 (2025) 126864. doi:https://doi.org/10.1016/j.eswa.2025.126864.

[13] D. Zhang, Y. Xia, D. Quan, H. Mi, X. Hou, L. Lin, Forecasting long-term sequences based on a seasonal and periodic-trend feature disentangled network, Journal of the Franklin Institute 361 (12) (2024) 106964. doi:https://doi.org/10.1016/j.jfranklin.2024.106964.

[14] T. Tang, W. Jiang, H. Zhang, J. Nie, Z. Xiong, X. Wu, W. Feng, Gm (1, 1) based improved seasonal index model for monthly electricity consumption forecasting, Energy 252 (2022) 124041. doi:https://doi.org/10.1016/j.energy.2022.124041.

[15] X. Wang, N. Xie, A non-linear grey fourier model based on kernel method for seasonal traffic speed forecasting, Communications in Nonlinear Science and Numerical Simulation 131 (2024) 107871. doi:https://doi.org/10.1016/j.cnsns.2024.107871.

[16] S. Li, C. Zhao, J. Zhou, Stnet: Seasonal-trend network for multivariate time series forecasting, Neurocomputing (2025) 131407doi:https://doi.org/10.1016/j.neucom.2025.131407.

[17] S. Aydore, T. Zhu, D. P. Foster, Dynamic local regret for non-convex online forecasting, Advances in neural information processing systems 32 (2019).

[18] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, et al., Stl: A seasonal-trend decomposition, J. off. Stat 6 (1) (1990) 3–73.

[19] A. M. De Livera, R. J. Hyndman, R. D. Snyder, Forecasting time series with complex seasonal patterns using exponential smoothing, Journal of the American statistical association 106 (496) (2011) 1513–1527. doi:https://doi.org/10.1198/jasa.2011.tm09771.

[20] S. J. Taylor, B. Letham, Forecasting at scale, The American Statistician 72 (1) (2018) 37–45. doi:https://doi.org/10.1080/00031305.2017.1380080.

[21] P. Welch, The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, IEEE Transactions on audio and electroacoustics 15 (2) (2003) 70–73. doi:10.1109/TAU.1967.1161901.

[22] A. Wilson, R. Adams, Gaussian process kernels for pattern discovery and extrapolation, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, Vol. 28 of Proceedings of Machine Learning Research, PMLR, Atlanta, Georgia, USA, 2013, pp. 1067–1075.
URL https://proceedings.mlr.press/v28/wilson13.html

[23] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, arXiv preprint arXiv:1905.10437 (2019). doi:https://doi.org/10.48550/arXiv.1905.10437.

[24] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, Vol. 162 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 27268–27286.
URL https://proceedings.mlr.press/v162/zhou22g.html

[25] M. Zinkevich, Online convex programming and generalized infinitesimal gradient ascent, in: Proceedings of the 20th international conference on machine learning (icml-03), 2003, pp. 928–936.

[26] O. Anava, E. Hazan, S. Mannor, O. Shamir, Online learning for time series prediction, in: Conference on learning theory, PMLR, 2013, pp. 172–184.

[27] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM computing surveys (CSUR) 46 (4) (2014) 1–37. doi:https://doi.org/10.1145/2523813.

[28] Z. Wang, F. Wan, C. M. Wong, T. Qian, Fast basis search for adaptive fourier decomposition, EURASIP Journal on Advances in Signal Processing 2018 (1) (2018) 74. doi:10.1109/TSP.2022.3143723.

[29] J. Li, X. Yang, T. Qian, Q. Xie, The adaptive fourier decomposition for financial time series, Engineering Analysis with Boundary Elements 150 (2023) 139–153. doi:https://doi.org/10.1016/j.enganabound.2023.01.037.

[30] J. Kivinen, A. J. Smola, R. C. Williamson, Online learning with kernels, IEEE transactions on signal processing 52 (8) (2004) 2165–2176. doi:10.1109/TSP.2004.830991.

[31] R. P. Adams, D. J. MacKay, Bayesian online changepoint detection, arXiv preprint arXiv:0710.3742 (2007). doi:https://doi.org/10.48550/arXiv.0710.3742.

[32] E. Hazan, et al., Introduction to online convex optimization, Foundations and Trends® in Optimization 2 (3-4) (2016) 157–325. doi:http://dx.doi.org/10.1561/2400000013.

[33] S. C. Hoi, D. Sahoo, J. Lu, P. Zhao, Online learning: A comprehensive survey, Neurocomputing 459 (2021) 249–289. doi:https://doi.org/10.1016/j.neucom.2021.04.112.

[34] E. Hazan, A. Agarwal, S. Kale, Logarithmic regret algorithms for online convex optimization, Machine Learning 69 (2) (2007) 169–192. doi:https://doi.org/10.1007/s10994-007-5016-8.

[35] C. Lee, S.-H. Kim, C.-H. Youn, Cooperating edge cloud-based hybrid online learning for accelerated energy data stream processing in load forecasting, IEEE Access 8 (2020) 199120–199132. doi:10.1109/ACCESS.2020.3035421.

[36] V. Vovk, Kernel ridge regression, in: Empirical inference: Festschrift in honor of vladimir n. vapnik, Springer, 2013, pp. 105–116. doi:https://doi.org/10.1007/978-3-642-41136-6_11.

[37] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, Vol. 20, Curran Associates, Inc., 2007.
URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf

[38] P. J. Huber, Robust estimation of a location parameter, in: Breakthroughs in statistics: Methodology and distribution, Springer, 1992, pp. 492–518. doi:https://doi.org/10.1007/978-1-4612-4380-9_35.

[39] A. C. Harvey, Forecasting, structural time series models and the kalman filter (1990). doi:https://doi.org/10.1017/CBO9781107049994.

[40] C. E. Shannon, Communication in the presence of noise, Proceedings of the IEEE 72 (9) (2005) 1192–1201. doi:10.1109/PROC.1984.12998.

[41] A. V. Oppenheim, Discrete-time signal processing, Pearson Education India, 1999.

[42] T. Evgeniou, M. Pontil, Regularized multi–task learning, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 109–117. doi:https://doi.org/10.1145/1014052.1014067.

[43] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67. doi:https://doi.org/10.2307/1271436.

[44] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society Series B: Statistical Methodology 68 (1) (2006) 49–67. doi:https://doi.org/10.1111/j.1467-9868.2005.00532.x.

[45] L. Zhang, T. Yang, rong jin, Z.-H. Zhou, Dynamic regret of strongly adaptive methods, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 5882–5891.
URL https://proceedings.mlr.press/v80/zhang18o.html

[46] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan

(Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 22419–22430.
URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf

[47] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, arXiv preprint arXiv:2211.14730 (2022). doi:https://doi.org/10.48550/arXiv.2211.14730.

[48] N. Lu, Q. Ouyang, Y. Li, C. Zou, Electrical load forecasting model using hybrid lstm neural networks with online correction, arXiv preprint arXiv:2403.03898 (2024). doi:https://doi.org/10.48550/arXiv.2403.03898.

[49] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, J. Choo, Reversible instance normalization for accurate time-series forecasting against distribution shift, in: International Conference on Learning Representations, 2022.
URL https://openreview.net/forum?id=cGDAkQo1C0p