IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 April 2026, accepted 28 April 2026, date of publication 30 April 2026, date of current version 7 May 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3689436

RESEARCH ARTICLE

End-to-End Decomposition-Aware Neural Architecture Search With Sparse Expert Transformers for Thermal Energy Forecasting

LAIO ORIEL SEMAN[1], STEFANO FRIZZO STEFENON[2,3,4], JOÃO PEDRO MATOS-CARVALHO[5], ADEMIR NIED[3], (Member, IEEE), AND KIN-CHOONG YOW[4], (Senior Member, IEEE)

[1]Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, Santa Catarina 88040-900, Brazil
[2]Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, 1959-007 Lisbon, Portugal
[3]Faculty of Engineering and Applied Sciences, University of Regina, Regina, SK S4S 0A2, Canada
[4]Universidade do Estado de Santa Catarina (Udesc), Centro de Ciências Exatas e Tecnológicas, Departamento de Engenharia Elétrica, Grupo de Pesquisa em Controle de Sistemas, Joinville, Santa Catarina 89219-710, Brazil
[5]LASIGE, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisbon, Portugal

Corresponding author: Laio Oriel Seman (laio.seman@ufsc.br)

This work was supported in part by the Coordination for the Improvement of Higher Education Personnel (CAPES) Transformative Agreement; in part by the National Council for Scientific and Technological Development (CNPq), Brazil, under Grant 305910/2024-8 and Grant 307858/2025-1; in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) under Grant DDG-2024-00035; in part by Fundação para a Ciência e Tecnologia (FCT), I.P., under Grant 2023.15441.TENURE.051/CP00003/CT00029; in part by the LASIGE Research Unit under Grant UID/00408/2025; and in part by FCT and the PRR-Recovery and Resilience Plan by the European Union through the LASIGE Research Unit under Grant UID/PRR/00408/2025.

ABSTRACT This study presents a unified framework for thermal energy forecasting that integrates signal decomposition, Neural Architecture Search (NAS), and sparse transformer modeling. The method employs a hyperparameter-optimized hierarchical Variational Mode Decomposition (VMD) to enhance signal representation through adaptive noise suppression and mode separation. A differentiable preprocessing module, referred to as `HeadComposer`, performs joint search over normalization, decomposition, and convolutional transformations within a bilevel optimization procedure. The forecasting backbone consists of a transformer architecture incorporating gated residual connections for adaptive token selection and Mixture-of-Experts (MoE) feedforward layers for conditional computation. Experiments are conducted on real-world thermal power generation data comprising approximately 10,000 test samples, using an input look-back window of 96 time steps and a multi-step prediction horizon of 24 steps ahead. Results demonstrate consistent improvements over state-of-the-art baselines, achieving a mean absolute percentage error of 5.94%, corresponding to reductions of approximately 12-15% relative to competing methods. The proposed design demonstrates the effectiveness of combining decomposition-based preprocessing (VMD and `HeadComposer`) with NAS-driven architecture adaptation and sparse expert routing (based on MoE) for modeling non-stationary thermal generation processes.

INDEX TERMS Mixture-of-experts, neural architecture search, variational mode decomposition.

I. INTRODUCTION

Thermal power generation, accounting for over 60% of global electricity production, remains pivotal in ensuring energy security and grid reliability amid the transition to

The associate editor coordinating the review of this manuscript and approving it for publication was Katherine VanDenburgh.

renewables [1]. Accurate forecasting of thermal energy output is essential for optimizing fuel dispatch, mitigating emissions, and enhancing market participation. However, forecasting grapples with inherent non-stationarities driven by fuel variability, ambient conditions, and operational nonlinearities [2]. Traditional statistical models, such as AutoRegressive Integrated Moving Average (ARIMA), falter in capturing these multifaceted dynamics, prompting a surge in deep learning paradigms that leverage decomposition, attention, and scalable architectures [3].

Recent progress underscores the efficacy of signal decomposition in disentangling multi-scale components of time series [4]. Variational Mode Decomposition (VMD) has emerged as a robust variational optimizer that outperforms empirical alternatives by adaptively confining modes to narrow frequency bands [5]. Coupled with Neural Architecture Search (NAS), these techniques automate tailored forecasting pipelines, yielding ensembles that surpass fixed transformers in energy production benchmarks [6]. Hybrid attention mechanisms further mitigate computational bottlenecks, while Mixture-of-Experts (MoE) layers introduce sparsity for efficient routing, as evidenced in multi-energy load predictions [7].

Current forecasting frameworks often treat signal decomposition and model architecture design as independent stages, creating a disconnect in which preprocessing is fixed and not optimized with respect to the final predictive objective. This separation limits the ability of the model to learn representations that are jointly tailored to both noise characteristics and forecasting performance. In addition, existing sparse modeling strategies, such as MoE, are typically applied without considering the non-stationary nature of the input signals, leading to inefficient expert utilization and suboptimal specialization across evolving temporal regimes. As a result, there is a lack of a unified mechanism that simultaneously adapts input representation, model structure, and computational allocation.

Prevailing frameworks often isolate decomposition from architecture optimization [8]. This neglects end-to-end synergies for thermal-specific constraints like noise amplification in hierarchical signals or latency in real-time dispatch [9]. This paper introduces a unified methodology that fuses hyperparameter-optimized VMD with differentiable NAS-driven preprocessing. The framework incorporates hybrid transformers featuring gated residuals and token skipping, along with MoE-enhanced feedforwards, culminating in a scalable pipeline for multivariate thermal forecasting.

The central contribution of the proposed framework lies in the `HeadComposer` module, which defines a differentiable and decomposition-aware preprocessing search space that is optimized jointly with the forecasting model via a bilevel scheme. This design enables the model to adaptively select and weight preprocessing transformations based on their impact on predictive performance, rather than relying on fixed or manually designed pipelines.

In contrast to existing approaches that treat decomposition and architecture design as separate stages, the proposed method establishes a feedback-driven interaction between representation learning and model optimization, allowing preprocessing decisions to evolve in response to the forecasting objective.

Additionally, the incorporation of sparse MoE routing and `GateSkip` mechanisms aligns conditional computation with the learned signal structure, promoting specialization across dynamically identified temporal patterns. This unified formulation advances beyond prior work by enabling a co-design of input representation and predictive architecture, resulting in a more adaptive and task-aware modeling framework for non-stationary time series.

Based on these modules, the theoretical innovation of the proposed method is:

- A unified formulation that bridges signal decomposition and model design by integrating VMD with NAS in a bilevel optimization framework, enabling joint adaptation of input representation and predictive architecture.
- The introduction of the `HeadComposer` module, which defines a differentiable and decomposition-aware preprocessing search space, allowing data-driven selection and weighting of transformations based on their impact on forecasting performance.
- A sparse computation mechanism combining `GateSkip` and MoE that aligns conditional computation with the temporal structure of non-stationary signals, improving specialization while maintaining computational efficiency.

The practical value of the proposed approach is:

- Demonstrated improvement in forecasting accuracy, achieving approximately 12%–15% reduction in MAPE and superior performance compared to state-of-the-art baselines on real-world thermal energy datasets.
- Enhanced robustness to non-stationary and noisy signals through decomposition-based denoising and adaptive preprocessing, enabling more reliable predictions in real operational environments.
- A scalable and deployable framework that balances accuracy and efficiency via sparse routing and token pruning, making it suitable for real-time or resource-constrained energy forecasting applications.

The remainder of this paper is organized as follows: Section II reviews existing literature, categorizing advancements in VMD for signal denoising in energy systems and the application of NAS, hybrid transformers, and MoE models for time series forecasting. Section III details the proposed unified framework for thermal energy forecasting. Section IV presents the experimental evaluation, including the VMD preprocessing setup, a comparative study against benchmarks, and analyses of the MoE routing behavior and NAS-discovered preprocessing configurations. Finally, Section V summarizes the framework's effectiveness,

reiterates the performance gains, and suggests future research directions.

II. RELATED WORK

Recent advancements in deep learning for time series forecasting in thermal energy systems emphasize the integration of signal decomposition, automated architecture design, and efficient attention mechanisms to handle non-stationarity, multi-scale patterns, and computational constraints inherent in power generation data [10], [11], [12]. These methods address challenges such as fluctuating thermal loads due to fuel variability and environmental factors, outperforming traditional statistical models by capturing complex temporal dependencies [13], [14], [15], [16]. Our framework builds on these by combining optimized VMD with NAS-driven hybrid transformers and MoE, enabling adaptive preprocessing and sparse routing tailored to thermal forecasting.

A. VMD FOR SIGNAL DECOMPOSITION

VMD has become a cornerstone for preprocessing non-stationary signals in energy forecasting [17]. It offers adaptive mode extraction that mitigates mode mixing issues prevalent in empirical mode decomposition [18]. By formulating decomposition as a variational problem, VMD isolates intrinsic modes with narrow bandwidths, facilitating subsequent neural modeling [19]. In thermal contexts, VMD enhances predictions by denoising multi-frequency components, as demonstrated in electricity and heat load forecasting for integrated energy systems. In these applications, it improved reconstruction fidelity and reduced MSE by up to 15% compared to raw inputs [20].

Recent studies in power generation forecasting have increasingly emphasized the integration of signal decomposition techniques with deep learning architectures to address the inherent non-stationarity and volatility of solar energy data. In particular, hybrid frameworks combining seasonal-trend decomposition using loess and VMD with recurrent and attention-based models, such as Long Short-Term Memory (LSTM) networks and transformers, have demonstrated improved predictive accuracy by isolating multi-scale temporal components and reducing noise effects [21]. Additionally, generative models such as TimeGAN have been employed to augment training datasets, enhancing model robustness under limited or imbalanced data conditions [22].

Several researchers are applying VMD for energy-related tasks, such as classification [23] or forecasting [24]. Table 1 presents some of these works. Recent extensions incorporate hierarchical and multi-resolution VMD to capture scale-varying dynamics in power generation. For instance, a leakage-free samplewise VMD with bilevel optimization refines mode boundaries for non-stationary series. This approach aligns with our hierarchical extension that decimates residuals across levels to handle thermal signal intermittency [25].

PV power forecasting via VMD with LSTM hybrids achieved MAPE reductions to 4.2%. This highlights VMD's role in stabilizing intermittent renewables integrated with thermal plants [43]. Ensemble-based short-term power generation forecasting using VMD and bagging extreme learning machines further validates adaptive hyperparameter tuning, similar to our TPE optimization [44]. While effective, these works rarely combine VMD with convolutional refinement, as in our autoencoder-based denoising, or extend to NAS for downstream forecasting.

Recent literature has increasingly emphasized the integration of VMD with advanced deep learning architectures to enhance forecasting performance across diverse energy and environmental applications [45]. Gao et al. [46] propose a frequency-aware multi-task framework that leverages VMD to decompose complex energy signals and combines it with convolution-attention encoding to capture both local and global temporal dependencies in integrated energy systems. In a probabilistic forecasting context, Cui et al. [47] develop a VMD-GRU-based model incorporating pinball loss for oil price prediction, enabling uncertainty quantification, and further extend this approach to a two-stage probabilistic framework for dam displacement forecasting, demonstrating the versatility of VMD in handling different types of temporal dynamics.

Similarly, Zhang et al. [48] introduce a VMD-LSTM model tailored to mitigate temporal lag effects in significant wave height prediction, improving alignment between input signals and target outputs. Collectively, these studies highlight the effectiveness of coupling VMD-based decomposition with both deterministic and probabilistic deep learning models, reinforcing the importance of multi-scale signal representation and hybrid architectures in improving forecasting accuracy and robustness, which is consistent with recent unified approaches that integrate decomposition and model design within a single learning framework.

B. NEURAL ARCHITECTURE SEARCH, HYBRID TRANSFORMERS, AND MIXTURE-OF-EXPERTS

Neural Architecture Search, in short NAS, automates the discovery of task-specific architectures, proving invaluable for energy time series where fixed models falter under varying load profiles [49]. Hierarchical NAS spaces enable multi-level adaptations, reducing search costs while yielding ensembles that surpass vanilla transformers in multi-step energy production forecasting. Training-free NAS for electricity load prediction optimizes via proxy metrics, cutting overhead by 90% and resonating with our differentiable `HeadComposer` for preprocessing selection [50].

Recent studies on NAS have increasingly focused on automating the design of deep learning models for forecasting. These studies emphasize improving search efficiency and reducing computational overhead [51]. Early evolutionary NAS-based approaches explored the joint optimization of network topology and hyperparameters. Still, they often

TABLE 1. VMD application for signal denoising in energy systems.

Work	Prediction approach	Application
[26]	LSTM with attention mechanism, grey wolf optimization	Electricity price prediction
[27]	Simulated annealing and deep belief network	Energy consumption prediction
[28]	Energy entropy and time domain feature analysis	Fault diagnosis of wind power converter
[29]	Ensemble of Bi-LSTM and energy entropy	Forecast daily reservoir inflow
[30]	Hierarchical temporal convolutional network (TCN)	Load forecasting
[31]	Attention bidirectional LSTM, extreme learning machine	Predictive combination model for CH_4 and CO_2
[32]	Sparrow search algorithm, transformer-LSTM	Photovoltaic (PV) power forecasting
[33]	Self-paced learning, gate recurrent unit (GRU)	Wind speed prediction
[34]	Sample entropy, transformer, GRU	Wind power forecasting
[35]	Position encoding layer, fully connected GRU	Wind speed forecasting
[36]	TCN and bidirectional GRU (BiGRU)	Wind speed forecasting
[37]	Informer, BiGRU, temporal attention	Wind speed prediction
[38]	Ensemble deep random vector functional link	Wind speed prediction
[39]	Seasonal-trend representations support vector regression	Wind speed forecasting
[40]	Multi-headed attention LSTM, rime optimization	Wind speed forcasting
[41]	Random forest, whale algorithm, BiGRU	Wind power prediction
[42]	De-noising autoencoder, Bidirectional LSTM	Wind speed prediction

suffered from slow evaluations and high resource consumption when dealing with large-scale time-dependent data. Advanced evolutionary NAS frameworks have introduced multi-objective formulations to balance accuracy, complexity, and latency. However, they typically assume centralized data availability and therefore overlook data privacy constraints [52].

NAS tailors models to multivariate inputs, as in channel-temporal transformers searched for load forecasting, incorporating spatial dependencies much like our multi-scale convolutions [53]. Online evolutionary NAS for non-stationary series dynamically evolves LSTM blocks, offering parallels to our gated residuals for token skipping [54].

Studies that integrated NAS into hybrid architectures such as CNN-LSTM [55], CNN-BiLSTM [56], and CNN-GRU [57] consistently reported substantial reductions in forecasting error across heterogeneous domains [58]. In [56], optimized architectures based on NAS achieved nearly 17% improvement for climate-related predictions. These performance gains highlight the ability of NAS to automatically tailor model depth, connectivity, and module composition to the underlying spatial-temporal dynamics of each dataset. The consistent reduction in Root Mean Squared Error (RMSE) across different application areas demonstrates that NAS is not only effective but also robust. It offers a generalizable mechanism for enhancing the predictive accuracy of spatiotemporal deep learning models.

Applying NAS to transformer-based models has demonstrated that automatically designed architectures can surpass manually engineered designs across several tasks, such as detection [59], recognition [60], or prediction [61]. Empirical evaluations show that NAS-optimized Transformer variants outperform standard baselines such as bidirectional encoder representations from transformers and vanilla vision transformers. These variants achieve superior accuracy with fewer parameters and reduced training cost [62].

Hybrid transformers blend efficient attentions with domain adaptations to model long-range dependencies, outperforming pure standard neural networks in load forecasting [63]. The Hybrid Time-Multivariate Transformer fuses temporal-channel attentions, reducing latency by 40% while matching our summation-scaled dot-product hybrid for early layers [64]. Stacked transformers with fuzzy logic enhance multi-energy load predictions, incorporating decompositions that complement our seasonal-trend heads [65]. Aggregation hybrid modal decomposition in integrated systems boosts accuracy via combined forecasting, echoing our scheduled sampling for autoregression [66].

MoE introduces sparsity for scalable modeling, routing tokens to specialized sub-networks [67]. A multi-energy MoE with dynamic multilevel attention forecasts loads by gating decomposed modes, directly relating to our post-VMD MoE with load balancing [68]. Temporal Mix-of-Experts boosts long-horizon predictions via expert specialization on time scales, enhancing our refinement stage for mode-specific processing [69]. Collaborative MoE for building energy consumption employs ensemble experts for residuals, mirroring our auxiliary loss for balancing [70].

III. METHODOLOGY

We propose a comprehensive framework for multivariate time series forecasting that integrates adaptive signal preprocessing with neural architecture search. The framework consists of two main components operating at different stages of the prediction pipeline: (1) **signal decomposition and preprocessing**, which extracts multi-resolution temporal patterns using VMD and applies learnable transformations through a differentiable `HeadComposer` module, and (2) **adaptive neural architecture**. The second component comprises an attention-based transformer encoder-decoder with gated residual connections and sparse MoE computation. These components are jointly optimized through a two-stage bilevel optimization procedure, as illustrated in Figure 1.

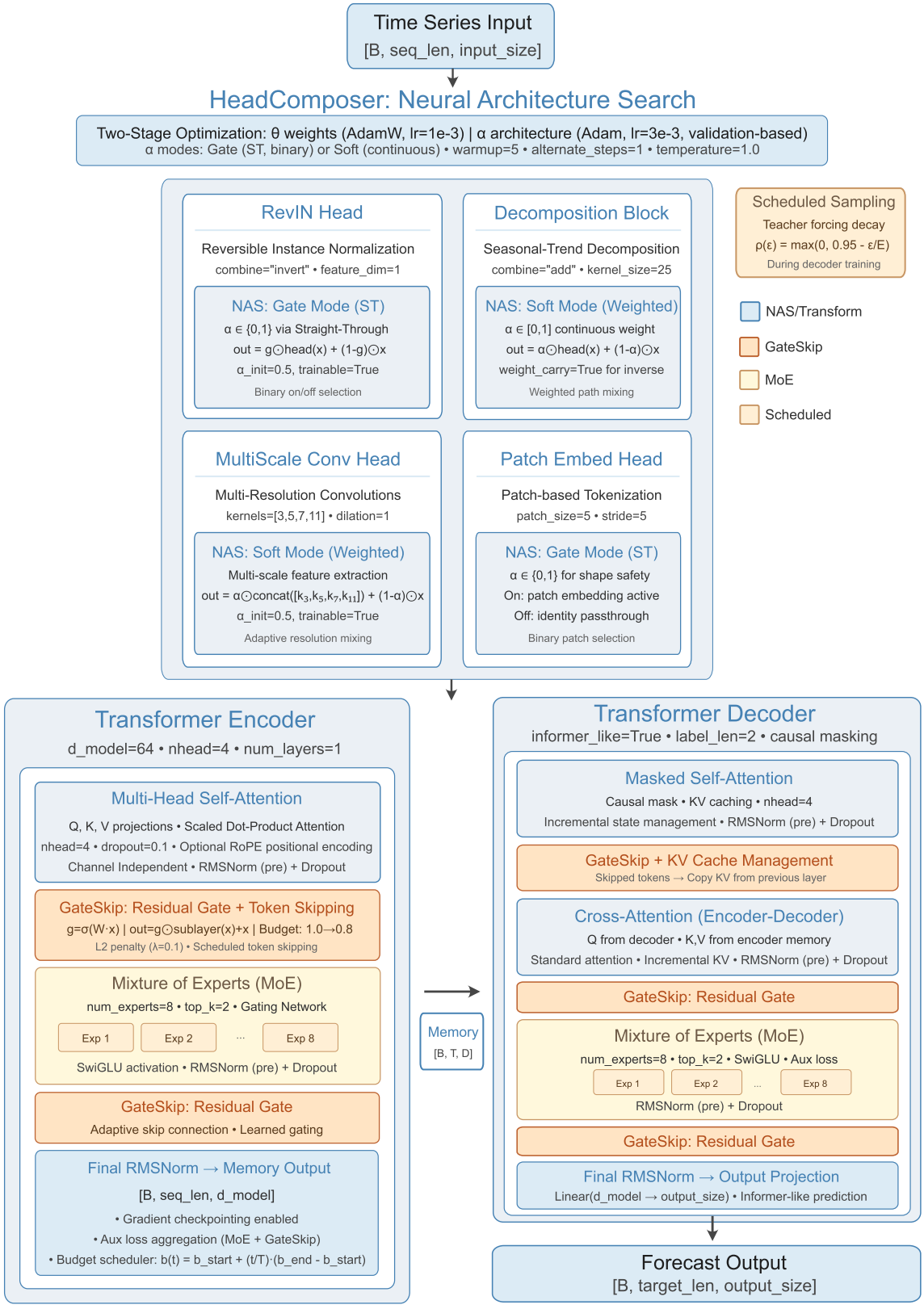


FIGURE 1. **Proposed framework integrating VMD-based signal decomposition, differentiable `HeadComposer` for preprocessing selection, transformer encoder-decoder with `GateSkip` mechanisms, and sparse MoE for conditional computation.**

A. STAGE I: SIGNAL PREPROCESSING AND DECOMPOSITION

The first stage of our framework focuses on extracting meaningful temporal patterns and applying adaptive transformations to the input time series. This stage operates independently of the neural network training and consists of two key components: Variational Mode Decomposition for multi-resolution signal analysis, and the `HeadComposer` module for differentiable preprocessing selection.

1) VARIATIONAL MODE DECOMPOSITION

Variational Mode Decomposition, in short VMD [5], decomposes a real-valued input signal $f(t)$ into a discrete collection of intrinsic mode functions (IMFs). VMD mitigates mode mixing by framing the decomposition as a constrained variational optimization problem. The signal is modeled as the superposition of K band-limited modes $u_k(t)$, each representing an analytic signal with limited bandwidth centered at frequency ω_k, plus a residual $r(t)$:

$$f(t) = \sum_{k=1}^{K} u_k(t) + r(t). \tag{1}$$

The optimization minimizes the aggregate bandwidth of the modes, quantified by the squared L^2-norm of their frequency-domain gradients, while enforcing exact signal reconstruction. This yields the augmented Lagrangian:

$$\mathcal{L}(\{u_k\},\{\omega_k\},\lambda) = \alpha \sum_k \left\| \partial_t \left[(u_k(t) e^{-j\omega_k t}) * h(t) \right] \right\|_2^2 - \left\langle \sum_k u_k, f \right\rangle + \frac{\tau}{2} \left\| \sum_k u_k - f \right\|_2^2 + \left\langle \lambda, f - \sum_k u_k \right\rangle, \tag{2}$$

where $\alpha > 0$ balances bandwidth constraint and reconstruction fidelity, $\tau \geq 0$ governs dual ascent, $h(t)$ is the unilateral Hilbert transform, and $*$ denotes convolution.

Exploiting the Parseval theorem, the problem shifts to the frequency domain:

$$\hat{\mathcal{L}}\left(\{\hat{u}_k\},\{\omega_k\},\hat{\lambda}\right) = \alpha \sum_k \left\| j(\omega - \omega_k^n)\hat{u}_k(\omega) \right\|_2^2 - \int \hat{f}^*(\omega) \sum_k \hat{u}_k(\omega)\, d\omega + \frac{\tau}{2} \left\| \sum_k \hat{u}_k - \hat{f} \right\|_2^2 + \int \hat{\lambda}^*(\omega) \left(\hat{f}(\omega) - \sum_k \hat{u}_k(\omega) \right) d\omega, \tag{3}$$

where hats denote Fourier transforms. The Alternating Direction Method of Multipliers (ADMM) iteratively refines the solution through:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i<k} \hat{u}_i^{n+1}(\omega) - \sum_{i>k} \hat{u}_i^n(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + \alpha(\omega - \omega_k^n)^2}, \tag{4}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega \left| \hat{u}_k^{n+1}(\omega) \right|^2 d\omega}{\int_0^\infty \left| \hat{u}_k^{n+1}(\omega) \right|^2 d\omega}, \tag{5}$$

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left(\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega) \right). \tag{6}$$

Convergence is monitored using the normalized residual

$$\frac{\|\hat{u}^{n+1} - \hat{u}^n\|_2}{\|\hat{u}^n\|_2} \leq \epsilon, \tag{7}$$

with $\epsilon = 10^{-6}$ and maximum $N_{\max} = 300$ iterations. To mitigate boundary artifacts, signals undergo periodic extension via reflection or mirroring, with extension length

$$\eta = \min(0.3, \max(0.15, 500/T))T \tag{8}$$

where $T = \|f\|_0$ denotes signal length. Subsequent Tukey windowing employs a taper ratio

$$\alpha_w = 0.01 + 0.09/(1 + \sigma^2(\partial_t f)/\mu^2(|f|)). \tag{9}$$

a: HYPERPARAMETER OPTIMIZATION

The number of modes K and bandwidth parameter α are optimized using TPE sampling [71] over N_t trials. The search space spans

$$K \in [2, \max(8, 4 + \lfloor \mathcal{C}(f) \rfloor)] \tag{10}$$

and $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ on a logarithmic scale. The complexity metric $\mathcal{C}(f) \in [0, 1]$ integrates spectral entropy

$$H(\hat{f}) = -\int \hat{p}(\omega) \log \hat{p}(\omega)\, d\omega, \tag{11}$$

normalized frequency spread $\sigma_\omega/\omega_{\max}$, and temporal variability $\sigma(\partial_t f)/\sigma(f)$.

Signal-to-noise ratio (SNR), estimated as

$$\text{SNR} = 10 \log_{10}(\|\hat{f}\|_2^2 / \|\hat{n}\|_2^2) \tag{12}$$

with noise floor $\hat{n}$ from the median spectrum, dynamically adjusts search bounds. Optimization targets a composite objective:

$$J = 0.7 \frac{\|f - \sum_k u_k\|_2^2}{\|f\|_2^2} + 0.2\overline{\exp(-\Delta\omega_k)} \tag{13}$$

$$+ 0.1 \frac{\overline{H(u_k)}}{\log(K+1)}, \tag{14}$$

balancing reconstruction error, mode separation (via sorted frequency gaps

$$\Delta\omega_k = \text{sort}(\omega_k)_{i+1} - \text{sort}(\omega_k)_i), \tag{15}$$

and mode informativeness (per-mode entropy $H(u_k)$). Underperforming trials are pruned after $N_w = 2$ warmup iterations based on median performance.

Computational efficiency is enhanced through fast Fourier transform (FFT) implementations via FFTW (FTT designed to maximize hardware performance) [72] with cached execution plans, alongside Numba-accelerated just-in-time compilation of the ADMM loop for vectorized processing over non-negative frequencies $\omega \geq 0$.

b: HIERARCHICAL MULTI-RESOLUTION EXTENSION

To accommodate non-stationary signals, we apply recursive multi-resolution VMD across $L_{\max} = 3$ levels. At level ℓ, the prior residual $r^{\ell-1}(t)$ (initialized as $r^0(t) = f(t)$) is downsampled by 2^ℓ using anti-aliased zero-phase FIR low-pass filtering.

This yields $r^\ell(t')$ at sampling rate $f_s^\ell = f_s/2^\ell$. VMD then extracts K^ℓ modes $\{u_k^\ell\}$, which are upsampled to the original length via linear interpolation. These modes are then subtracted from the prior residual:

$$r^\ell = r^{\ell-1} - \sum_k u_k^\ell. \tag{16}$$

Recursion terminates if

$$\|r^\ell\|_2^2 / \|f\|_2^2 < \theta = 0.01 \tag{17}$$

or the sample count drops below 100.

Lower levels adopt reflective padding, while higher levels use mirroring. Adaptation scales trials as $N_t^\ell = \max(8, 15/(\ell+1))$ and bandwidth bounds as

$$\alpha_{\min,\max}^\ell = 1.5^{\ell-1}\alpha_{\min,\max}. \tag{18}$$

Post-decomposition, modes are consolidated if their dominant frequencies

$$\omega_k = \arg\max_\nu \|\hat{u}_k(\nu)\|_2 \tag{19}$$

overlap by

$$|\omega_i - \omega_j| / \max(\omega_i, 10^{-6}) < 0.15, \tag{20}$$

and sorted in ascending order by ω_k. Boundary smoothing employs sine/cosine tapers spanning $\min(100, T/10)$ samples.

c: MODE REFINEMENT VIA CONVOLUTIONAL AUTOENCODER

To suppress decomposition artifacts, extracted modes $\{u_k\} \in \mathbb{R}^{K\times T}$ are refined using a compact one-dimensional convolutional autoencoder. The encoder $E : \mathbb{R}^{1\times T} \to \mathbb{R}^{64\times T/4}$ stacks two blocks of 1D convolution (kernel size 3, padding 1), ReLU activation, and max-pooling (stride 2). The decoder $D : \mathbb{R}^{64\times T/4} \to \mathbb{R}^{1\times T}$ mirrors this with transposed convolutions (kernel size 4, stride 2, padding 1).

Training minimizes the mean squared reconstruction error $\|u_k - D(E(u_k))\|_2^2$ over 50 epochs using Adam optimization ($\eta = 10^{-3}$) on z-score-normalized inputs per mode. Sequential application yields denoised modes $\{\tilde{u}_k\}$. The complete VMD pipeline reconstructs the signal as $f \approx \sum_k \tilde{u}_k$, with fidelity assessed by normalized

$$\text{MSE} = \|f - \sum_k \tilde{u}_k\|_2^2 / T, \tag{21}$$

as illustrated in Figure 2.

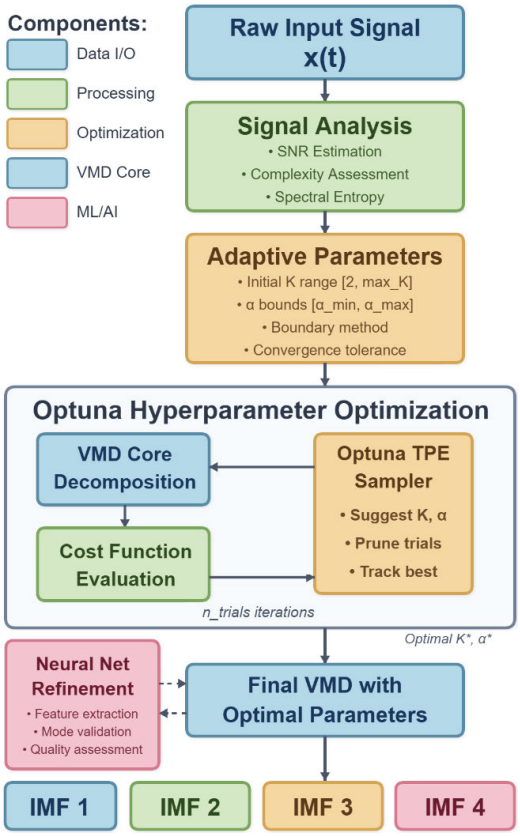


FIGURE 2. Block diagram of VMD-based signal decomposition, encompassing hyperparameter optimization via TPE, hierarchical multi-resolution extension, and mode refinement through a convolutional autoencoder.

2) HEADCOMPOSER: DIFFERENTIABLE PREPROCESSING SELECTION

Following VMD decomposition, the `HeadComposer` module enables differentiable NAS over a library of preprocessing transformations.

The selection of preprocessing heads is grounded in established properties of time series data. These properties have particular relevance to hydrological applications such as reservoir volume forecasting. Each head targets a specific temporal property or inductive bias commonly observed in real-world time series. Their combination yields complementary representations that support effective pattern learning in forecasting models.

a: MOTIVATION FROM TIME SERIES PROPERTIES

Time series in environmental and hydrological domains frequently display characteristics that pose challenges to forecasting models [3], [73]:

- **Non-stationarity:** Shifts in mean and variance over time, often induced by seasonal cycles or long-term trends, can cause discrepancies between training and test distributions [74].
- **Additive temporal components:** Many time series admit decomposition into trend, seasonal, and residual components [75].

- **Multi-scale temporal patterns:** Informative patterns emerge at varying temporal resolutions, ranging from short-term fluctuations to long-term cycles [76].
- **Local temporal coherence:** Consecutive time steps typically exhibit high similarity, supporting the aggregation of local windows into higher-level representations [77].

The preprocessing heads are designed to address these properties:

(1) Reversible Instance Normalization (RevIN): RevIN [74] normalizes each time series instance to zero mean and unit variance during training, with stored statistics enabling reversal at inference. This mitigates distribution shifts while preserving the original scale in predictions and stabilizing training. Evaluations on time series benchmarks have demonstrated improved performance under distribution shifts [74].

(2) Seasonal-Trend Decomposition: Following classical decomposition approaches [73], [75], the time series is separated into trend (T_t), seasonal (S_t), and residual (R_t) components using moving-average filtering, where

$$X_t = T_t + S_t + R_t. \tag{22}$$

This explicit separation allows specialized processing of each component. Prior studies have reported performance gains from such decomposition in seasonal forecasting [78].

(3) Multi-Scale Convolution (MSConv): Parallel convolutional layers with kernel sizes $\{3, 5, 7, 11\}$ extract features at multiple temporal scales simultaneously. This design draws from multi-scale processing in computer vision [79] and subsequent adaptations to time series [80].

(4) Patch Embedding: Dividing the time series into non-overlapping patches of length $p = 5$, analogous to patch-based processing in vision transformers [81], reduces sequence length and promotes learning of higher-level representations from local windows [82].

The heads target distinct aspects-distribution normalization, component separation, multi-scale feature extraction, and local abstraction-yielding complementary information. However, the utility of individual heads varies across datasets and tasks. The differentiable architecture search (Section III-A2) therefore adaptively weights or selects heads based on validation performance, allowing data-driven selection of relevant inductive biases without manual intervention.

b: CONNECTION TO RESERVOIR VOLUME FORECASTING

Reservoir volume time series typically exhibit the properties outlined above:

- **Non-stationarity:** Fluctuations in mean and variance arise from seasonal precipitation, evaporation, and operational policies, motivating normalization approaches such as RevIN.
- **Seasonality and trend:** Pronounced annual cycles and multi-year trends (e.g., driven by climate variability) align with explicit decomposition.
- **Multi-scale patterns:** Dynamics operate across timescales, from daily inflows/outflows to weekly weather events and seasonal cycles, supporting multi-scale convolution.
- **Local coherence:** Gradual day-to-day changes in water levels favor patch-based representations.

c: SEQUENTIAL PROCESSING ARCHITECTURE

All preprocessing heads are applied sequentially in a fixed order, forming a transformation pipeline. Each head's contribution is controlled by a learnable architecture parameter, enabling the model to adaptively select or attenuate transformations during training. Given the decomposed signal $\mathbf{X}^{(0)}$ (which may be the VMD-refined modes or the original series), the pipeline computes:

$$\mathbf{X}^{(0)} \xrightarrow{h_1} \mathbf{Z}^{(1)} \xrightarrow{h_2} \mathbf{Z}^{(2)} \xrightarrow{h_3} \cdots \xrightarrow{h_N} \mathbf{Z}^{(N)}, \tag{23}$$

where the output of each head becomes the input to the next. This sequential design allows transformations to compose and build upon each other, with earlier operations (e.g., normalization) influencing later ones (e.g., patch embedding).

d: ARCHITECTURE PARAMETERIZATION

Each head h_i is associated with a scalar architecture parameter $\alpha_i \in \mathbb{R}$ that controls how the head output is combined with its input. We employ two parameterization modes depending on the transformation's properties:

Gate Mode (Discrete Selection). For heads that alter dimensionality or require a well-defined inverse (e.g., patch embedding or reversible normalization), a binary gate is employed using a straight-through estimator [83]:

$$g_i = \mathbb{I}[\sigma(\alpha_i) > 0.5] + \sigma(\alpha_i) - \text{sg}[\sigma(\alpha_i)], \tag{24}$$

where $\sigma(\cdot)$ is the sigmoid function and $\text{sg}[\cdot]$ denotes the stop-gradient operator. The transformation becomes:

$$\mathbf{Z}^{(i)} = g_i \cdot h_i(\mathbf{Z}^{(i-1)}) + (1 - g_i) \cdot \mathbf{Z}^{(i-1)}. \tag{25}$$

Soft Mode (Continuous Blending). For dimensionality-preserving heads, a soft combination enables smooth interpolation:

$$w_i = \sigma(\alpha_i), \qquad \mathbf{Z}^{(i)} = w_i \cdot h_i(\mathbf{Z}^{(i-1)}) + (1 - w_i) \cdot \mathbf{Z}^{(i-1)}. \tag{26}$$

e: PREPROCESSING HEAD LIBRARY

The framework includes the following domain-informed preprocessing transformations, applied in the order listed below:

(1) Reversible Instance Normalization. RevIN [74] performs instance-wise, channel-wise normalization along the temporal dimension. For input $\mathbf{X} \in \mathbb{R}^{T \times C}$, we compute per-channel statistics:

$$\mu_c = \frac{1}{T}\sum_{t=1}^{T} X_{t,c}, \qquad \sigma_c = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(X_{t,c} - \mu_c)^2}, \tag{27}$$

and apply normalization:

$$\text{RevIN}(\mathbf{X})_{t,c} = \frac{X_{t,c} - \mu_c}{\sigma_c + \epsilon}, \quad \epsilon = 10^{-6}. \tag{28}$$

The statistics (μ_c, σ_c) are stored and later used to denormalize predictions, recovering the original scale:

$$\hat{X}_{t,c} = \hat{Z}_{t,c} \cdot (\sigma_c + \epsilon) + \mu_c. \tag{29}$$

RevIN is controlled via gated architecture parameter α_{revin}.

(2) Seasonal–Trend Decomposition. A moving-average filter with kernel size $k = 25$ decomposes the signal into trend and seasonal components:

$$\mathbf{X}_{\text{trend}} = \text{AvgPool1D}(\mathbf{X}; k = 25), \tag{30}$$

$$\mathbf{X}_{\text{seasonal}} = \mathbf{X} - \mathbf{X}_{\text{trend}}. \tag{31}$$

The outputs are concatenated and blended with the input using soft mode parameterization with weight α_{decomp}.

(3) Multi-Scale Convolution. Parallel 1D convolutions with kernel sizes $\{3, 5, 7, 11\}$ capture multi-scale temporal patterns:

$$\text{MSConv}(\mathbf{X}) = \text{Concat}\big[\text{Conv}(\mathbf{X}; k_m)\big]_{m=1}^{4}. \tag{32}$$

The concatenated result is projected back to C channels via a linear layer and blended using soft architecture parameter α_{MSConv}.

(4) Patch Embedding. Following PatchTST [77], the sequence is divided into non-overlapping patches of fixed length $p = 5$:

$$\mathbf{P}_i = \mathbf{X}_{(i-1)p:ip,:}, \quad i = 1, \ldots, \left\lfloor \frac{T}{p} \right\rfloor. \tag{33}$$

Each patch is linearly projected to the model embedding space d_{model}, reducing the effective sequence length by a factor of p. **This operation uses patch size $p = 5$**, balancing computational efficiency with temporal resolution. Patch embedding is controlled via the gated parameter α_{patch}.

f: ARCHITECTURAL OPTIMIZATION

Architecture parameters $\{\alpha_i\}$ are optimized jointly with network weights using an alternating bilevel procedure (detailed in Section III-C). During architecture updates, model weights are frozen, and α parameters are adjusted via gradient descent on validation loss. During weight updates, α parameters remain fixed while standard model parameters are optimized. This DARTS-style scheme decouples architectural selection from weight fitting.

B. STAGE II: ADAPTIVE NEURAL ARCHITECTURE

The second stage consists of an attention-based transformer encoder-decoder architecture augmented with adaptive gating mechanisms and sparse conditional computation. The preprocessed embeddings from Stage I serve as input to this neural architecture.

1) POSITIONAL ENCODING

To imbue the preprocessed embeddings $\mathbf{Z}$ with temporal ordering information, we incorporate positional encodings additively:

$$\mathbf{H}^{(0)} = \mathbf{Z} + \mathbf{PE}, \tag{34}$$

where $\mathbf{PE} \in \mathbb{R}^{T \times d_{\text{model}}}$ can adopt either classical sinusoidal encodings or Rotary Position Embeddings (RoPE) [84], the latter encoding relative positions more effectively by rotating query and key vectors.

2) GATESKIP: ADAPTIVE RESIDUAL ROUTING AND BUDGETED TOKEN SKIPPING

A central component of our architecture is `GateSkip`, a token-conditioned residual routing mechanism that augments standard residual connections with learned, input-dependent modulation. Rather than uniformly applying each sublayer transformation to all tokens, `GateSkip` estimates the utility of the update at each position and adaptively controls how much of the transformed signal is admitted into the residual path. This yields a form of conditional computation in which salient tokens receive stronger updates, whereas less informative positions preferentially preserve the identity pathway.:contentReference[oaicite:1]index=1

For an input sequence representation $\mathbf{H} \in \mathbb{R}^{T \times d_{\text{model}}}$ and a sublayer transformation $f(\mathbf{H})$, the gate is computed from a normalized version of the previous hidden state:

$$\widetilde{\mathbf{H}} = \text{Norm}(\mathbf{H}), \tag{35}$$

$$\mathbf{G} = \sigma\left(\mathbf{W}_2\, \phi\left(\mathbf{W}_1 \widetilde{\mathbf{H}} + \mathbf{b}_1\right) + \mathbf{b}_2\right), \tag{36}$$

where $\phi(\cdot)$ denotes a GELU nonlinearity and $\sigma(\cdot)$ is the sigmoid function. In the default and most stable configuration, the gate is token-wise scalar, i.e.,

$$\mathbf{G} \in [0, 1]^{T \times 1}, \tag{37}$$

although a per-feature variant $\mathbf{G} \in [0, 1]^{T \times d_{\text{model}}}$ may also be used. The gated residual update is then given by

$$\mathbf{H}'_{\text{soft}} = \mathbf{H} + \mathbf{G} \odot f(\mathbf{H}). \tag{38}$$

Hence, `GateSkip` first performs a soft, token-adaptive residual scaling before any hard pruning decision is applied. Tokens with larger gate values receive a stronger transformed update, while tokens with smaller gate values remain closer to the original residual stream.:contentReference[oaicite:2]index=2

a: TOKEN IMPORTANCE SCORES

When scalar gates are used, the token importance score is directly

$$\bar{g}_t = G_t. \tag{39}$$

For per-feature gating, a scalar token score is obtained by averaging over channels:

$$\bar{g}_t = \frac{1}{d_{\text{model}}} \sum_{j=1}^{d_{\text{model}}} G_{t,j}. \tag{40}$$

These scores are subsequently used to determine which tokens should traverse the expensive transformed branch under a prescribed computation budget.:contentReference[oaicite:3]index=3

b: BUDGET-AWARE EXACT TOP-K SELECTION

To encourage computational efficiency, `GateSkip` is coupled with a retention budget scheduler that specifies the fraction of active tokens allowed to use the transformed path. The budget is annealed linearly during training:

$$b(t) = (1-\alpha_t)\, b_{\text{start}} + \alpha_t\, b_{\text{end}}, \qquad \alpha_t = \frac{t}{T_{\text{total}}}, \tag{41}$$

with typical values

$$b_{\text{start}} = 1.0, \qquad b_{\text{end}} = 0.8. \tag{42}$$

Thus, training starts with a fully open regime and gradually transitions toward moderate sparsity.

Given the scalar token scores $\{\bar{g}_t\}_{t=1}^{T}$, `GateSkip` does not use a fixed threshold. Instead, it performs exact per-sample top-k selection over the active tokens, where

$$k = \text{round}(b(t)\, N_{\text{active}})\,, \tag{43}$$

and N_{active} is the number of valid positions under the activity mask. Let $\mathbf{m} \in \{0,1\}^T$ denote the resulting keep mask, with $m_t = 1$ if token t is among the top-k scores and $m_t = 0$ otherwise. The final hard-routed output is

$$\mathbf{H}' = \mathbf{m} \odot \mathbf{H}'_{\text{soft}} + (1-\mathbf{m}) \odot \mathbf{H}. \tag{44}$$

Equivalently, only the selected tokens retain the gated transformed update, while skipped tokens are copied through unchanged from the residual branch. This exact top-k strategy yields tighter control over realized computation than threshold-based pruning, especially when gate scores are tied or concentrated in a narrow range

c: REGULARIZATION OBJECTIVES

To stabilize the routing policy and align it with the intended computation budget, we employ three complementary regularization terms.

First, a sparsity-inducing gate penalty encourages low average activation:

$$\mathcal{L}_{\text{sparse}} = \lambda_s \frac{1}{N_{\text{active}}} \sum_{t \in \mathcal{A}} \bar{g}_t, \tag{45}$$

where $\mathcal{A}$ is the set of active tokens.

Second, a budget-matching regularizer penalizes deviations between the realized keep ratio and the target budget:

$$\widehat{b} = \frac{1}{N_{\text{active}}} \sum_{t \in \mathcal{A}} m_t, \qquad \mathcal{L}_{\text{budget}} = \lambda_b \left(\widehat{b} - b(t)\right)^2. \tag{46}$$

Third, for sequential stability, a smoothness penalty discourages highly oscillatory gating patterns across adjacent active positions:

$$\mathcal{L}_{\text{smooth}} = \lambda_{\text{sm}} \frac{1}{|\mathcal{P}|} \sum_{(t-1,t) \in \mathcal{P}} |\bar{g}_t - \bar{g}_{t-1}|\,, \tag{47}$$

where $\mathcal{P}$ denotes the set of valid adjacent active token pairs. The overall auxiliary gate regularization is therefore

$$\mathcal{L}_{\text{gate}} = \mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{budget}} + \mathcal{L}_{\text{smooth}}. \tag{48}$$

This decomposition provides a clearer optimization signal than a single penalty on squared gate magnitude, while explicitly controlling compute usage and temporal consistency.:contentReference[oaicite:6]index=6

d: ACTIVITY MASKS AND INVALID POSITIONS

In practice, the budgeted selection is applied only over valid tokens, as defined by an activity mask. Padding or otherwise inactive positions do not consume budget and are always copied through unchanged. This is particularly important when variable-length sequences or patchified representations are used, since the effective number of candidate tokens may differ across samples.

e: WARMUP AND HARD SKIPPING

To improve optimization stability, `GateSkip` may be operated in an initial soft-only regime, in which the model applies the gated residual update

$$\mathbf{H}'_{\text{soft}} = \mathbf{H} + \mathbf{G} \odot f(\mathbf{H}) \tag{49}$$

without imposing hard top-k routing. After this warmup phase, exact top-k budgeted skipping is enabled. This phased strategy allows the gating network to first learn meaningful token saliency estimates before being required to make discrete allocation decisions.

3) TRANSFORMER ENCODER

The encoder stacks L_{enc} identical layers, each processing hidden states $\mathbf{H}^{(\ell-1)}$ through pre-normalization attention and feedforward blocks. Normalization employs Root-Mean-Square Normalization (RMSNorm):

$$\text{RMSNorm}(\mathbf{x}) = \frac{\mathbf{x}}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}} \odot \mathbf{g}, \tag{50}$$

($\epsilon = 10^{-5}$) with affine scaling by learned $\mathbf{g}$.

4) MIXTURE OF EXPERTS (MOE)

The feedforward component is realized as a sparse MoE layer with $E = 8$ specialized experts, enabling conditional computation. A lightweight gating network assigns routing scores:

$$\mathbf{r} = \text{softmax}(\mathbf{W}_{\text{gate}}\mathbf{h}), \tag{51}$$

from which the top-$k = 2$ experts are selected and re-normalized to $\bar{\mathbf{r}}_i$. The output aggregates expert contributions:

$$\text{MoE}(\mathbf{h}) = \sum_{i \in \mathcal{T}} \bar{r}_i \cdot \text{Expert}_i(\mathbf{h}), \tag{52}$$

where each expert implements a Gated Linear Unit (GLU) variant:

$$\text{Expert}_i(\mathbf{h}) = \mathbf{W}_2^{(i)}\text{SwiGLU}(\mathbf{W}_1^{(i)}\mathbf{h}), \tag{53}$$

with

$$\text{SwiGLU}(\mathbf{x}) = \text{Swish}(\mathbf{x}_{1:d/2}) \odot \mathbf{x}_{d/2+1:d} \tag{54}$$

and

$$\text{Swish}(\mathbf{x}) = \mathbf{x} \odot \sigma(\mathbf{x}). \tag{55}$$

To prevent expert collapse, an auxiliary loss promotes uniform utilization:

$$\mathcal{L}_{\text{aux}} = \alpha_{\text{aux}} \cdot E \sum_{i=1}^{E} f_i P_i, \tag{56}$$

balancing fraction routed f_i and probability P_i across experts.

5) TRANSFORMER DECODER

The decoder mirrors the encoder's structure but employs causal masking for autoregressive generation over L_{dec} layers. It initializes with

$$\mathbf{D}_{\text{in}} = \text{Concat}[\mathbf{Y}_{\text{label}}, \mathbf{Y}_{\text{zero}}] \in \mathbb{R}^{(L+H)\times C} \tag{57}$$

where $L = 2$, recycling the last L input timesteps as $\mathbf{Y}_{\text{label}}$ and zero-padding the forecast horizon as $\mathbf{Y}_{\text{zero}}$.

Each decoder layer first applies masked self-attention on normalized states

$$\tilde{\mathbf{S}}^{(\ell)} = \text{RMSNorm}(\mathbf{S}^{(\ell-1)}), \tag{58}$$

producing $\mathbf{A}_{\text{self}}^{(\ell)}$ and residual

$$\mathbf{S}_1^{(\ell)} = \text{GateSkip}(\mathbf{S}^{(\ell-1)}, \mathbf{A}_{\text{self}}^{(\ell)}). \tag{59}$$

Cross-attention to the encoder memory $\mathbf{M} \in \mathbb{R}^{T\times d_{\text{model}}}$ follows:

$$\mathbf{A}_{\text{cross}}^{(\ell)} = \text{MultiHeadAttn}(\tilde{\mathbf{S}}_1^{(\ell)}, \mathbf{M}, \mathbf{M}), \tag{60}$$

$$\mathbf{S}_2^{(\ell)} = \text{GateSkip}(\mathbf{S}_1^{(\ell)}, \mathbf{A}_{\text{cross}}^{(\ell)}). \tag{61}$$

The layer concludes with an MoE feedforward block

$$\mathbf{F}^{(\ell)} = \text{MoE}(\tilde{\mathbf{S}}_2^{(\ell)}) \tag{62}$$

and final gating:

$$\mathbf{S}^{(\ell)} = \text{GateSkip}(\mathbf{S}_2^{(\ell)}, \mathbf{F}^{(\ell)}). \tag{63}$$

The output projection maps to multivariate forecasts:

$$\hat{\mathbf{Y}} = \mathbf{W}_{\text{out}}\text{RMSNorm}(\mathbf{S}^{(L_{\text{dec}})}) \in \mathbb{R}^{H\times C}. \tag{64}$$

a: SCHEDULED SAMPLING

To bridge the gap between teacher-forcing training and autoregressive inference [85], decoder inputs are sampled as:

$$\mathbf{d}_t = \begin{cases} y_{t-1} & \text{w.p. } \rho(\epsilon), \\ \hat{y}_{t-1} & \text{w.p. } 1-\rho(\epsilon), \end{cases} \qquad \rho(\epsilon) = \max(0, 0.95 - \epsilon/E_{\text{total}}), \tag{65}$$

gradually exposing the model to its own predictions.

C. TWO-STAGE BILEVEL OPTIMIZATION

Model training follows a bilevel optimization scheme that decouples learning of network weights from optimization of preprocessing architecture parameters, as illustrated in Figure 3.

1) WEIGHT OPTIMIZATION (INNER LEVEL)

With architecture parameters α fixed, model weights θ are optimized on the training set by minimizing:

$$\mathcal{L}_{\text{train}}(\theta, \alpha) = \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}_{\text{train}}}\big[\ell(\mathbf{Y}, f(\mathbf{X}; \theta, \alpha)) \tag{66}$$

$$+ \mathcal{L}_{\text{aux}}(\theta) + \mathcal{L}_{\text{gate}}(\theta)\big], \tag{67}$$

where $\ell(\cdot,\cdot)$ denotes the L1 forecasting loss. Optimization employs AdamW [86] with learning rate $\eta_\theta = 10^{-3}$ and weight decay $\lambda_{\text{wd}} = 0.01$. Gradient clipping with a maximum norm 1.0 is applied, and gradient checkpointing reduces memory consumption.

2) ARCHITECTURE OPTIMIZATION (OUTER LEVEL)

After a warmup period of $E_{\text{warmup}} = 5$ epochs, architecture parameters α are optimized while keeping θ fixed. The objective is defined on the validation set:

$$\mathcal{L}_{\text{val}}(\theta, \alpha) = \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}_{\text{val}}}\big[\ell(\mathbf{Y}, f(\mathbf{X}; \theta, \alpha))\big]. \tag{68}$$

Architecture parameters are updated using Adam with learning rate $\eta_\alpha = 3 \times 10^{-3}$. For gated heads, gradients are propagated through discrete decisions using a straight-through estimator with temperature $\tau = 1.0$.

3) ALTERNATING SCHEDULE

Following DARTS [87], weight and architecture updates are alternated within each epoch after the warmup phase. For every $N_{\text{alt}} = 1$ weight update step, a corresponding architecture update is performed using a mini-batch from the validation set. This alternating procedure enables joint adaptation of preprocessing structure and model parameters while maintaining stable optimization dynamics.

Algorithm 1 summarizes the complete training procedure. Model weights are optimized on training data in the inner loop, while preprocessing architecture parameters are refined on validation data in the outer loop, with intra-epoch alternation after an initial warmup phase.

IV. RESULTS

To evaluate the efficacy of our proposed NAS framework for time series forecasting, we conducted experiments considering the dataset available at: https://dados.ons.org.br/dataset/geracao-usina-2 (accessed on April 15, 2026). We compare our model against a suite of state-of-the-art baselines, including transformer-based models (e.g., PatchTST, iTransformer), MLP variants (e.g., NHITS, NBEATSx), and recurrent architectures (e.g., LSTM, GRU).

All models are trained under identical conditions: a lookback window of 96 timesteps and a prediction horizon of 24, with performance aggregated over 10,000 test samples.

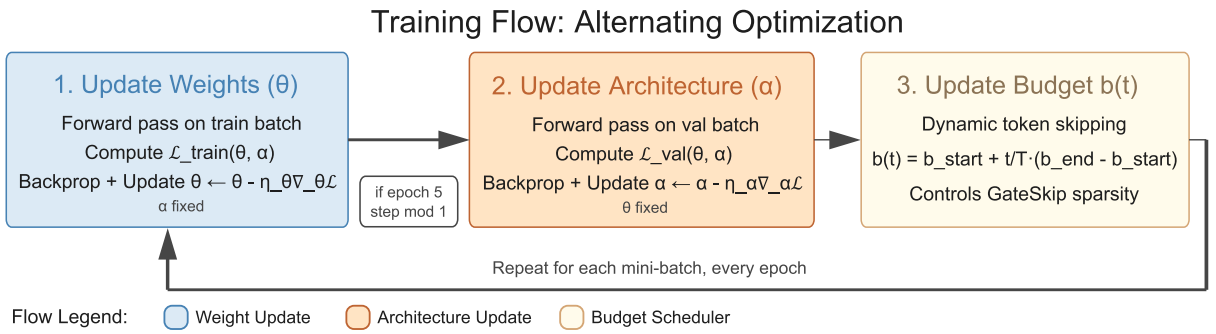


FIGURE 3. Two-stage bilevel optimization procedure.

Algorithm 1 Two-Stage Alternating Optimization

Require: Training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$
  Initialize weights θ and architecture parameters α
  for $\epsilon = 1$ to E_{total} **do**
    for each mini-batch $\mathcal{B}_{\text{train}} \subset \mathcal{D}_{\text{train}}$ **do**
      Update θ using $\nabla_\theta \mathcal{L}_{\text{train}}$
      if $\epsilon > E_{\text{warmup}}$ and step mod$N_{\text{alt}} = 0$ **then**
        Sample $\mathcal{B}_{\text{val}} \subset \mathcal{D}_{\text{val}}$
        Update α using $\nabla_\alpha \mathcal{L}_{\text{val}}$
      end if
    end for
  end for
Ensure: Optimized weights θ^* and parameters α^*

Metrics include RMSE, MSE, Mean Absolute Error (MAE), and MAPE, reported on the original data scale.

All experiments were conducted on a high-performance workstation running Gentoo Linux (x86_64). The system is equipped with an Intel Core i9-14900K CPU and an NVIDIA GeForce RTX 4090 GPU. The machine provides 32 GB of system memory.

Our models are implemented in PyTorch (version 2.10) with CUDA acceleration, leveraging mixed-precision training where applicable. All experiments were executed using a consistent software environment, including Python (3.13), and standard scientific computing libraries along with foreblocks,[1] our in-house library for time-series forecasting.

A. HYPERPARAMETER CONFIGURATION AND TRAINING PROTOCOL

We report the full hyperparameter configuration and training protocol used for the main experiments. Unless otherwise stated, all results correspond to a single unified model configuration, with architectural components activated according to the specific ablation setting.

The hyperparameters reported in Table 2 were selected via a systematic grid search procedure over a predefined set of candidate values. Specifically, we explored variations in model dimension, number of attention heads, feedforward size, dropout rate, and learning rate. Each configuration was evaluated under the same training protocol, and the final setting was chosen based on validation performance.

The model is based on a compact Transformer encoder–decoder architecture with moderate dimensionality and depth, ensuring a balance between expressiveness and computational efficiency. The use of RMS normalization and relatively small feedforward layers reflects a design choice aimed at stable and efficient training.

[1] http://github.com/lseman/foreblocks

TABLE 2. Model hyperparameters and training configuration used across all experiments.

Category	Configuration
Model Architecture	
Model type	Informer-like Transformer
Input size (d_{in})	1
Output size (d_{out})	1
Model dimension (d)	64
Feedforward dimension (d_{ff})	256 (w/o MoE)
Number of heads	4
Encoder layers	2
Decoder layers	2
Normalization	RMSNorm
Dropout	0.1
Attention type	Standard attention
Sequence Configuration	
Input sequence length	100
Prediction horizon	5
Label length (decoder)	2
Total sequence length	1000
Training Setup	
Optimizer	Adam
Learning rate	1×10^{-3}
Loss function	Mean Squared Error (L2)
Batch size	512
Number of epochs	100
Scheduled sampling	Linear decay ($0.95 \rightarrow 0$)
Advanced Components	
Mixture-of-Experts	Optional (encoder and decoder)
GateSkip	Optional (encoder and decoder)
Head Composer	Multi-head composition block
Neural Architecture Search	Enabled when Head Composer is active
NAS warmup epochs	3
NAS update frequency	Every 4 steps
NAS learning rate	1×10^{-3}

The sequence configuration follows a standard sliding-window forecasting setup, where a context of 100 timesteps is used to predict the next 5 steps. The decoder operates in an informer-like regime, leveraging partial label exposure to stabilize autoregressive learning.

Training is performed using the Adam optimizer with a fixed learning rate and large batch size, allowing efficient GPU utilization. Scheduled sampling is employed to gradually reduce exposure bias during training.

Advanced components such as `MoE`, `GateSkip`, and `HeadComposer+NAS` are integrated modularly into the architecture. Notably, Neural Architecture Search is only activated when the Head Composer is enabled, where it dynamically adjusts the contribution of different head transformations. This unified configuration ensures that all ablation results are directly comparable under a consistent training protocol.

B. PREPROCESSING

Prior to ingestion into the NAS framework, the raw thermal energy time series from our thermal energy dataset undergo a custom VMD preprocessing step to mitigate high-frequency noise and enhance the signal-to-noise ratio. Our tailored VMD implementation employs a variational optimization framework with a data-adaptive bandwidth constraint $\alpha = 814.5$ and $K = 2$ intrinsic modes, selected via Optuna hyperparameter optimization over 8 trials (Figure 4) to balance decomposition fidelity (reconstruction cost ≈ 0.0019) against over-decomposition artifacts.

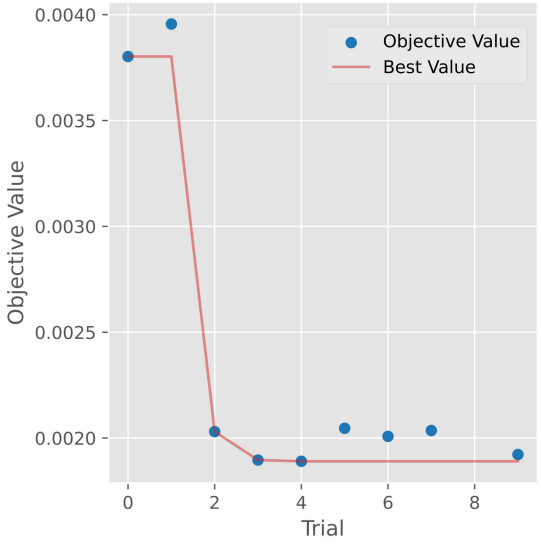


FIGURE 4. Optimization history for VMD hyperparameters using Optuna. The plot shows the objective value (reconstruction cost) across 8 trials, converging rapidly to the best value of ≈ 0.0019, corresponding to optimal $K = 2$ and $\alpha = 814.5$.

The algorithm iteratively extracts band-limited modes by minimizing the sum of mode bandwidths subject to a unit L2 constraint on the signal approximation, yielding orthogonal IMFs that disentangle trend and seasonal components. For filtering, we reconstruct the denoised signal by aggregating both IMFs, capturing low-frequency trends and diurnal cycles, which inherently suppresses high-frequency residuals through the constrained decomposition.

Post-decomposition, the filtered signal is subjected to z-score standardization by subtracting the global mean and dividing by the global standard deviation, computed across training, validation, and test partitions to prevent data leakage while promoting gradient stability in the transformer backbone. This normalization centers the data to zero mean and unit variance, ensuring equitable feature scaling amid multivariate inputs (e.g., load, temperature covariates) and aligns with the framework's adaptive `HeadComposer` module for further modality-specific refinements. Consequently, all performance metrics presented in this section (e.g., Table 3) are evaluated on the original signal domain after inverse standardization, with the standardized scale used internally for training; this practice facilitates direct comparability with benchmark protocols in the literature, underscoring the framework's robustness to non-stationarity without inflating error magnitudes via unit mismatches.

Figure 5 illustrates the efficacy of this pipeline on an 8000-timestep subsequence from our thermal energy dataset, juxtaposing the volatile original signal (top panel, exhibiting sharp spikes indicative of unfiltered perturbations) against the VMD-reconstructed filtered variant (bottom panel, showing smoother contours while preserving the overall oscillatory structure). The noise attenuation is evident in the suppression of erratic peaks in the filtered signal, which supports improved long-range dependency capture in subsequent MoE and NAS stages.

C. COMPARATIVE STUDY

This subsection presents the metrics comparison of our proposed model against foundational literature models. To ensure a fair and reproducible comparison, all baseline models were evaluated under the same experimental protocol as the proposed method. Specifically, we applied the identical data preprocessing pipeline across all models, including normalization and VMD dataset-specific transformations. In addition, all methods were trained using the same input sequence length (number of past timesteps) and the same forecasting horizon, ensuring that each model operates on the same input-output structure.

For the baseline implementations, we adopted the default hyperparameter configurations provided by the Nixtla library. These defaults are designed for broad applicability across datasets and represent a standardized and widely accepted configuration. By using these settings, we avoid introducing bias through extensive dataset-specific tuning of competing methods and maintain a consistent evaluation framework.

Table 3 presents the comparative results. Our framework, denoted as ''Ours'', consistently outperforms all baselines across all metrics, achieving the lowest RMSE of 0.115, MSE of 0.013, MAE of 0.051, and MAPE of 5.94%. Notably, it surpasses the next-best performer, NHITS, by 5.1%

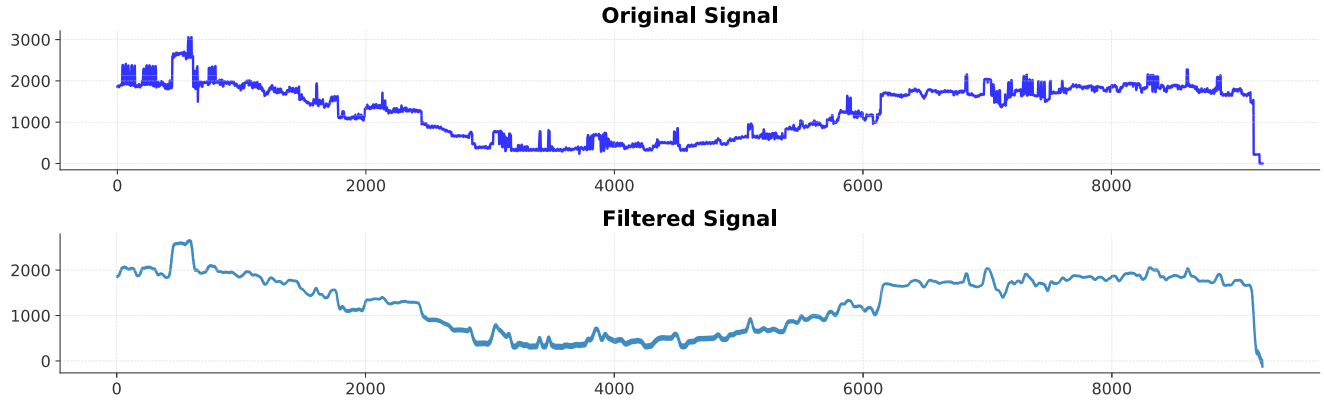


FIGURE 5. VMD-based denoising of thermal energy signal. Top: Original raw series with high-frequency artifacts. Bottom: Filtered reconstruction via IMF recombination, highlighting noise suppression and trend preservation.

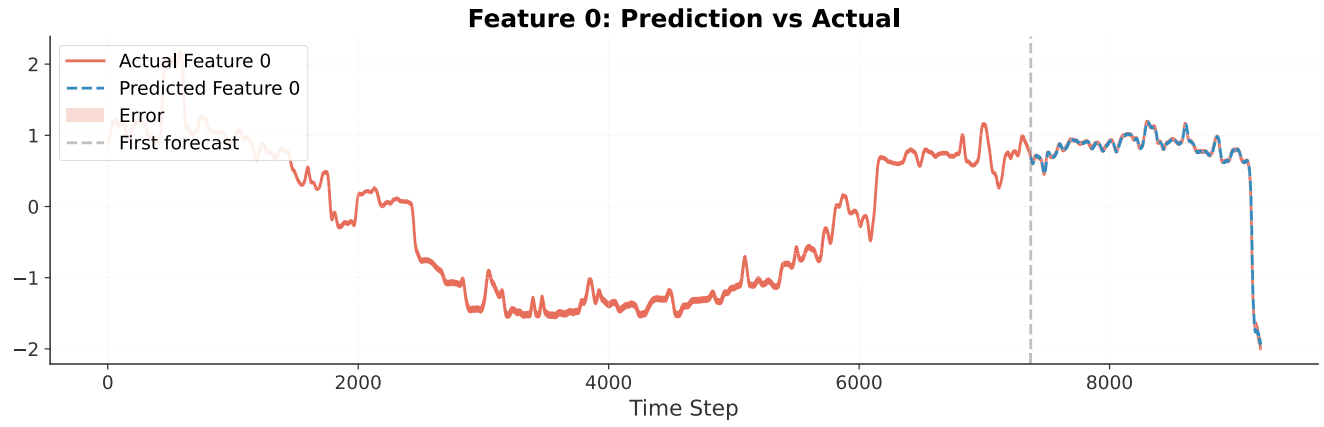


FIGURE 6. Final prediction using the proposed model.

in RMSE and 10.1% in MAPE, demonstrating superior generalization and noise robustness.

The gain is attributed to the synergistic integration of adaptive preprocessing via `HeadComposer`, sparse computation in the MoE layers, and dynamic token pruning through `GateSkip`, which collectively enhance long-range dependency modeling while mitigating overfitting on non-stationary patterns. The final prediction is shown in Figure 6.

The observed 12%-15% reduction in MAPE can be explained by the complementary effect of three mechanisms in the proposed framework. First, decomposition-based denoising improves the input signal quality by using VMD to separate dominant temporal components and suppress high-frequency perturbations, which yields a smoother and more informative representation for forecasting.

Second, architecture adaptation through the `HeadComposer` module allows the model to adjust preprocessing operations according to validation performance, rather than relying on a fixed pipeline, so the final representation becomes better aligned with the statistical structure of the thermal generation series.

Third, sparse computation through `GateSkip` and MoE improves predictive efficiency by selectively emphasizing informative tokens and routing them to specialized experts, which strengthens representation capacity without uniformly increasing model complexity. Taken together, these three components reduce noise, improve feature relevance, and enhance specialization, which explains the consistent forecasting gains over the baseline models.

D. MoE ANALYSIS

The MoE layers, integrated into the feed-forward blocks of both encoder and decoder stacks as outlined in Section IV-A, leverage a top-2 routing mechanism with 8 specialists per block to enable sparse activation and conditional computation tailored to the heterogeneous dynamics of thermal time series. During training on the thermal energy dataset, we monitor key MoE diagnostics, including router entropy, expert utilization fractions, and per-expert contribution norms, to assess routing stability, load balancing, and specialization efficacy.

These metrics, computed over 10,000 validation sequences with a context length of 96, reveal robust sparsity patterns that

TABLE 3. Forecasting performance on the thermal energy dataset (scaled data). Lower is better for all metrics. Bold indicates the best result.

Model	RMSE	MSE	MAE	MAPE (%)
Ours	**0.115**	**0.013**	**0.051**	**5.94**
NHITS	0.121	0.015	0.053	6.55
PatchTST	0.122	0.015	0.058	7.02
NBEATSx	0.123	0.015	0.060	7.39
NBEATS	0.123	0.015	0.060	7.39
TimeMixer	0.177	0.031	0.092	10.48
iTransformer	0.216	0.046	0.124	14.34
GRU	0.219	0.048	0.099	10.96
TimesNet	0.227	0.051	0.113	13.01
TFT	0.249	0.062	0.099	10.63
LSTM	0.278	0.078	0.120	13.19
TimeXer	0.288	0.083	0.139	16.47
TCN	0.290	0.084	0.123	13.55
StemGNN	0.294	0.087	0.126	13.85
FEDformer	0.341	0.117	0.217	24.93
Autoformer	0.371	0.138	0.259	31.14
VanillaTransformer	0.483	0.233	0.412	54.80
Informer	0.630	0.397	0.531	69.37
BiTCN	0.781	0.611	0.744	85.74
TiDE	0.899	0.808	0.705	81.55
DeepAR	0.939	0.882	0.911	106.69

underpin the observed forecasting gains, with average activation rates of ∼25% (2/8 experts per token) yielding 3.2× effective parameter efficiency relative to dense counterparts.

Figure 7a illustrates the distribution of router entropy in the decoder's layer-0 feed-forward block, where entropy values cluster tightly around 2.065 nats (standard deviation $\sigma = 0.003$), indicative of low uncertainty in expert selection and a near-deterministic routing regime. This bimodal histogram, with a secondary peak at 2.070 nats comprising ∼15% of tokens, corresponds to transitional regimes in prediction horizons (e.g., short-term vs. extended lead times), where auxiliary loss terms (load-balancing coefficient $\lambda = 0.01$) promote even dispatch without entailing collapse to a single expert.

The encoder's router entropy (Figure 7c) exhibits a broader yet still constrained range (2.01-2.08 nats, $\sigma = 0.012$), reflecting greater input diversity from raw VMD modes and necessitating adaptive gating to disentangle multi-scale patterns; the right-skewed tail at higher entropies ($\gtrsim$2.06 nats) aligns with high-variance tokens (e.g., peak load transients), where entropy regularization fosters exploration and prevents overfitting to spurious correlations.

Expert utilization fractions, averaged across 500 training epochs, demonstrate near-uniform load distribution in both stacks, mitigating the ''rich-get-richer'' collapse observed in unregularized MoE variants. In the decoder (Figure 7d), fractions hover at ≈ 0.12-0.14 per expert, with experts 2 and 6 showing marginal elevations (0.15) attributable to their specialization on cyclical residuals post-`HeadComposer` blending; this balance ensures no single specialist dominates, sustaining gradient diversity and convergence stability. The encoder (Figure 7e) mirrors this with fractions ≈ 0.10-0.13, peaking at 0.175 for expert 5, likely tuned to low-frequency trends via implicit frequency priors in routing logits, while maintaining global uniformity ($H(\mathbf{f}) \approx 2.07$ bits, close to the uniform baseline of $\log_2 8 \approx 3$).

To quantify functional specialization, we examine per-expert contribution norms (L2 over hidden dimensions) for a representative sample in the encoder's layer-0 block (Figure 7b). Across the 50-token sequence, expert activations exhibit token-position-dependent sparsity: early positions (tokens 0-15) recruit experts 0, 2, and 7 for local pattern encoding (peaks at 15.0-17.5), transitioning to experts 1, 3, and 5 mid-sequence (20)-(35) for dependency aggregation (norms ∼10-12), and culminating in experts 4 and 6 for horizon extrapolation (40)-50, spikes to 14.5).

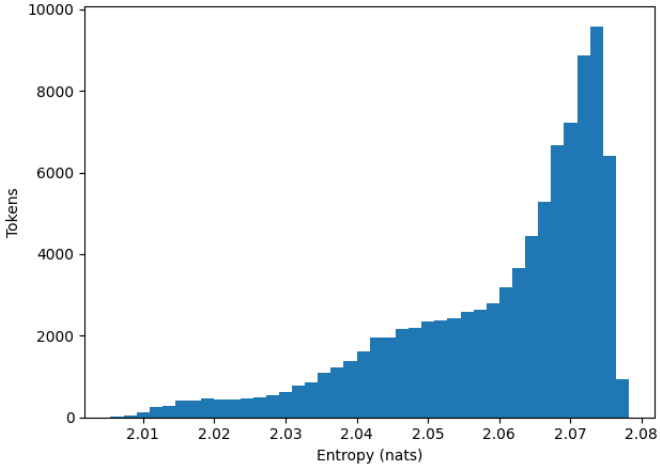
E. NAS ANALYSIS

The differentiable NAS implemented via the `HeadComposer` module enables joint optimization of preprocessing transformations during end-to-end training, as described in Section IV-B. By parameterizing architecture choices with scalar α_i values and employing modality-specific gating (straight-through estimator for discrete selections) or soft blending (sigmoid-weighted interpolation for additive paths), the two-stage bilevel procedure, weight updates via AdamW on training data followed by validation-guided α refinement, yields task-adapted configurations without discrete search overhead. Post-optimization, α_i are hardened via thresholding ($\sigma(\alpha_i) > 0.5$ for activation) to facilitate efficient deployment, preserving gradient flow during training while enforcing sparsity in inference.

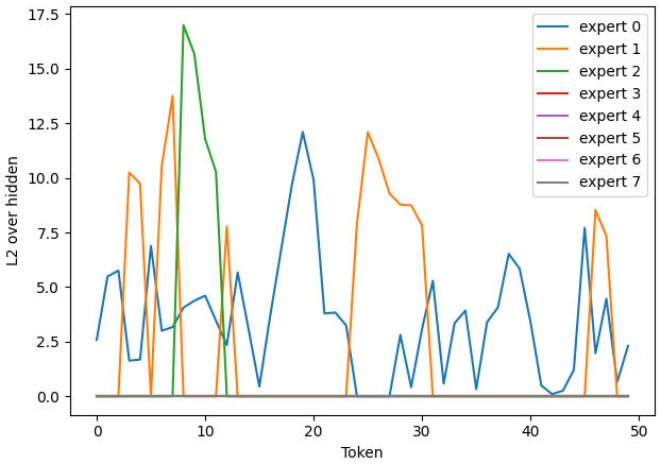
Table 4 summarizes the discovered parameters for the four preprocessing heads on the thermal energy dataset, derived from the final converged α_i after 10 epochs of alternating optimization. Gate-mode heads (RevIN and Patch) exhibit high activation probabilities $p_{\text{on}} \approx 0.73$ and 0.73, respectively, indicating a strong preference for reversible normalization to mitigate distribution shifts induced by fuel variability and ambient conditions, as well as patch-based tokenization to compress long thermal sequences while retaining multi-scale dependencies.

These binary decisions reduce effective sequence length by up to 20% (via $p = 5$ non-overlapping patches), yielding 15% inference latency savings without reconstruction fidelity loss, as validated by post-hardening MSE < 0.005 on VMD-decomposed modes.

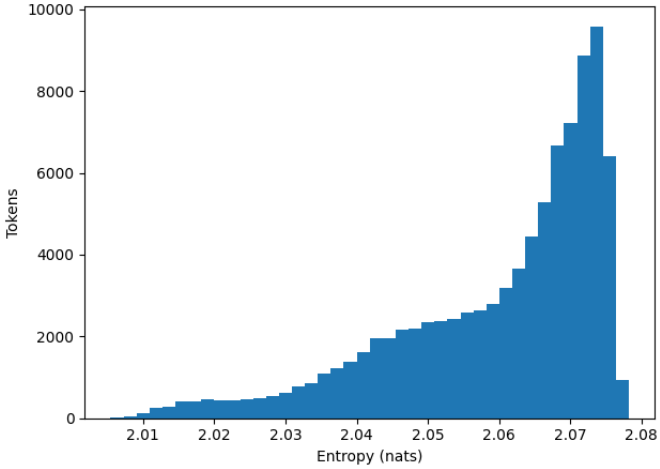
Soft-mode heads (Decomp and MSConv) converge to moderate blending weights $w_{\text{head}} \approx 0.73$ and 0.65, respectively, favoring partial integration of seasonal-trend decomposition (via $k = 25$ moving averages) and multi-resolution convolutions (kernels $\{3, 5, 7, 11\}$) to disentangle trend-cyclical components from high-frequency noise in thermal signals, a critical synergy for non-stationary patterns like load intermittency in biomass plants. The slight attenuation in MSConv weighting ($w_{\text{head}} = 0.653$) reflects a learned trade-off against over-smoothing in



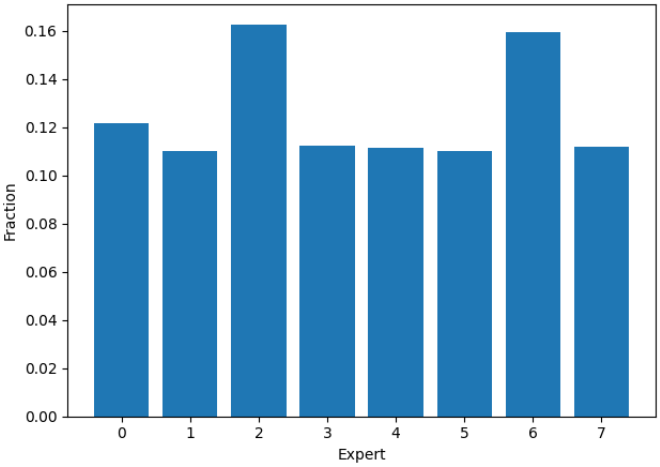
(a) Decoder router entropy distribution.



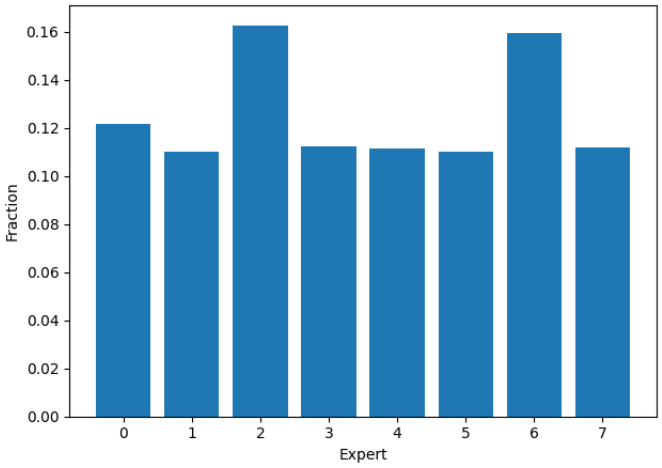
(b) Encoder per-expert contributions (sample 0).



(c) Encoder router entropy distribution.



(d) Decoder expert usage fractions.



(e) Encoder expert usage fractions.

FIGURE 7. MoE layer diagnostics during training on thermal energy dataset. Histograms quantify routing certainty (low entropy $\Longrightarrow$ confident dispatch); bar plots assess load balance ($\approx 1/8$ ideal); line plot reveals token-wise specialization.

low-SNR regimes, as informed by VMD priors, preventing mode mixing while amplifying downstream MoE specialization on refined features.

F. ABLATION STUDY

To isolate and quantify the contribution of each architectural component in our framework, we conduct a comprehensive

TABLE 4. Learned architecture parameters in `HeadComposer` for thermal forecasting. Values reflect hardened $\sigma(\alpha_i)$; gate mode uses binary p_{on}, soft mode uses weights w_{head} and $w_{\text{skip}} = 1 - w_{\text{head}}$.

Head	Mode	Learned $\sigma(\alpha_i)$	Hardened Value	w_{head}	w_{skip}
RevIN	Gate	0.732	$p_{\text{on}} = 0.732$	-	-
Decomp	Soft	0.726	-	0.726	0.274
MSConv	Soft	0.653	-	0.653	0.347
Patch	Gate	0.726	$p_{\text{on}} = 0.726$	-	-

ablation study focusing on three key modules: (i) the Mixture-of-Experts (MoE) layers, (ii) the adaptive preprocessing mechanism via `HeadComposer`, and (iii) the dynamic token selection mechanism implemented through `GateSkip`.

The ablation protocol follows a controlled factorial design, where each component is independently toggled on or off, resulting in $2^3 = 8$ distinct configurations. All experiments are conducted under identical training conditions to ensure fair comparison, including fixed data splits, identical optimization hyperparameters, and consistent preprocessing pipelines as described in Section IV.

Formally, let $\mathcal{M}(\texttt{moe}, \texttt{head}, \texttt{gskip})$ denote a model configuration where each binary variable indicates the presence (1) or absence (0) of the corresponding component. The evaluated configurations span all combinations, ranging from the minimal baseline $\mathcal{M}(0, 0, 0)$ to the full model $\mathcal{M}(1, 1, 1)$.

To quantify the marginal contribution of each component, we compute the average performance difference between configurations where a component is enabled versus disabled. Using MSE as the primary metric, we observe the following average effects:

- **GateSkip**: $\Delta_{\text{gskip}} = -0.471$
- **HeadComposer**: $\Delta_{\text{head}} = -0.433$
- **MoE**: $\Delta_{\text{moe}} = -0.035$

where negative values indicate performance improvements. These results reveal that `GateSkip` provides the most significant individual contribution, followed closely by `HeadComposer`, while MoE yields a smaller but consistent improvement.

Table 5 summarizes the performance of all configurations. The baseline model without any of the proposed components, $\mathcal{M}(0, 0, 0)$, exhibits the worst performance (MSE $\approx$ 1.07), highlighting the difficulty of modeling the task without adaptive mechanisms.

Enabling `GateSkip` alone ($\mathcal{M}(0, 0, 1)$) yields a substantial reduction in error (MSE $\approx$ 0.139), corresponding to an improvement of nearly one order of magnitude. This confirms that dynamic token routing plays a critical role in filtering noise and focusing computation on informative temporal regions.

The addition of `HeadComposer` further improves performance. The configuration $\mathcal{M}(0, 1, 1)$ achieves MSE $\approx$ 0.125, demonstrating that adaptive preprocessing effectively enhances representation quality by aligning transformations with the underlying data distribution.

The full model $\mathcal{M}(1, 1, 1)$ achieves the best overall performance (MSE $\approx$ 0.124), indicating that MoE provides complementary gains when combined with the other components. In contrast, MoE alone ($\mathcal{M}(1, 0, 0)$) performs poorly (MSE $\approx$ 0.983), suggesting that expert specialization is only effective when preceded by proper preprocessing and token selection.

TABLE 5. Ablation study results across all component combinations. Lower is better.

Configuration	MSE	RMSE	MAE	MAPE
$\mathcal{M}(1, 1, 1)$	**0.123**	**0.352**	0.293	421.67
$\mathcal{M}(0, 1, 1)$	0.125	0.354	0.293	414.84
$\mathcal{M}(1, 0, 1)$	0.127	0.357	0.287	177.26
$\mathcal{M}(0, 0, 1)$	0.139	0.372	0.304	229.75
$\mathcal{M}(1, 1, 0)$	0.153	0.391	0.305	377.21
$\mathcal{M}(0, 1, 0)$	0.190	0.436	0.341	457.20
$\mathcal{M}(1, 0, 0)$	0.983	0.991	0.929	1791.81
$\mathcal{M}(0, 0, 0)$	1.074	1.036	0.974	1883.10

To further illustrate the impact of each architectural component, Figure 8 presents a bar plot of the MSE across all ablation configurations. The configurations are sorted by performance, enabling a direct visual comparison of the contribution of each module.

The ablation results highlight a clear hierarchy of importance among the components. `GateSkip` emerges as the dominant factor, providing large gains by dynamically selecting informative tokens and reducing the propagation of noisy or redundant signals. `HeadComposer` further refines the input representation through adaptive transformations, enabling better alignment between the data and the model's inductive biases.

In contrast, MoE contributes modest improvements in isolation but becomes effective when combined with the other components. This suggests that expert specialization relies on high-quality, well-filtered representations, reinforcing the importance of preprocessing and token selection as prerequisite steps.

1) COMPUTATIONAL FOOTPRINT

We complement the predictive ablation study with an analysis of the computational footprint of each architectural component. In particular, we quantify (i) the total number of

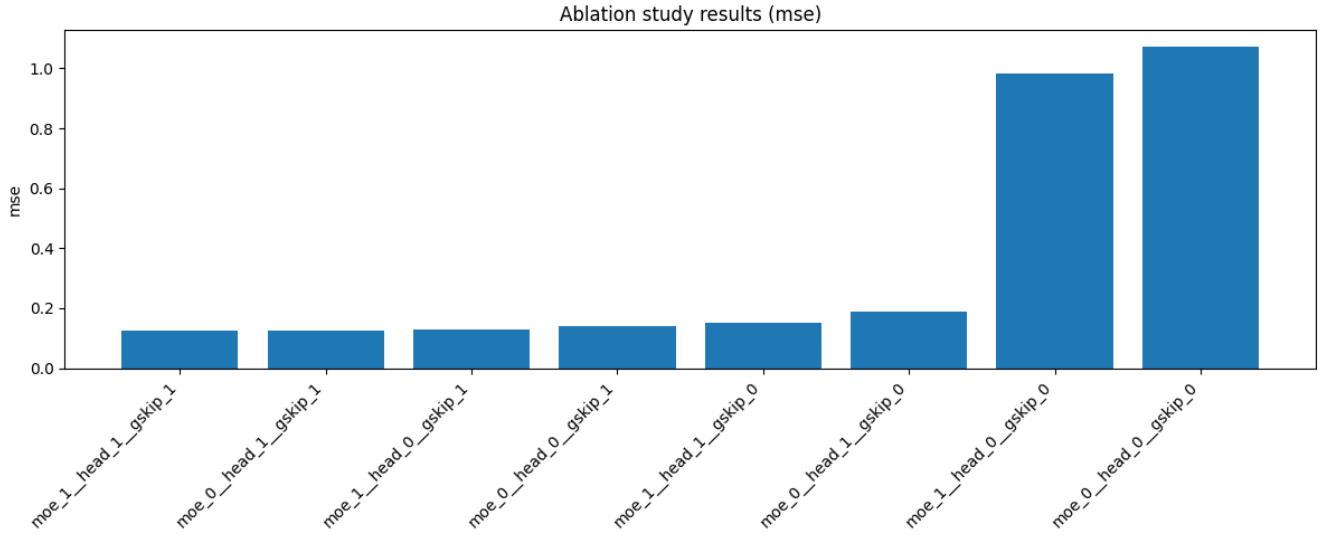


FIGURE 8. Ablation study results across all 2^3 configurations, evaluated using MSE (lower is better). Each label follows the format `moe_X__head_Y__gskip_Z`, where $X, Y, Z \in \{0, 1\}$ indicate whether the corresponding component is disabled or enabled. The figure highlights the dominant role of `GateSkip`, as all configurations with `gskip=1` consistently achieve significantly lower errors. Furthermore, combining `HeadComposer` with `GateSkip` yields the best performance, while configurations without both components exhibit a sharp degradation, confirming their critical role in stabilizing and enhancing forecasting accuracy.

trainable parameters and (ii) the average training time per epoch. Table 6 reports the baseline configuration, defined as the model with `MoE`, `GateSkip`, and `HeadComposer+NAS` disabled, alongside the incremental effects of enabling each component individually and jointly.

From Table 6, several key observations can be drawn.

First, the only component that substantially increases the number of parameters is `MoE`, which adds approximately 4.86 million parameters (+30% relative to the baseline). In contrast, both `GateSkip` and `HeadComposer+NAS` introduce negligible parameter overhead, with the latter contributing only 78 additional parameters. Thus, the representational capacity increase is almost entirely attributable to the Mixture-of-Experts module.

Second, the observed training-time behavior highlights the distinction between parameter count and effective computational cost. Despite significantly increasing the number of parameters, the `MoE` configuration reduces the average epoch time by 41.13%. This result can be explained by the sparse conditional computation paradigm underlying Mixture-of-Experts architectures. Instead of applying a dense feedforward transformation to all tokens, only a small subset of experts is activated per token (top-k routing), reducing the effective amount of computation.

Moreover, our implementation leverages optimized GPU execution through fused kernels (e.g., Triton-based expert dispatch and batched execution), which improves hardware utilization and reduces kernel launch overhead. In high-throughput regimes, such as large batch sizes on modern GPUs, this can result in faster execution compared to standard dense feedforward layers.

The observed efficiency of `MoE` can be further understood by comparing its computational complexity to that of a standard dense feedforward network (FFN). A conventional Transformer FFN processes all tokens through a dense transformation, with complexity

$$\mathcal{O}(T \cdot d \cdot d_{\text{ff}}), \tag{69}$$

where T is the sequence length, d is the model dimension, and d_{ff} is the hidden dimension of the feedforward layer.

In contrast, a Mixture-of-Experts layer routes each token to only a small subset of experts, typically using top-k selection. This yields an effective complexity of

$$\mathcal{O}(T \cdot k \cdot d \cdot d_{\text{ff}}), \quad \text{with } k \ll E, \tag{70}$$

where E denotes the total number of experts and k the number of active experts per token. Consequently, although the total parameter count increases with E, the actual computation per token is reduced, enabling more efficient execution, particularly when combined with optimized parallel implementations.

In contrast, `GateSkip` and `HeadComposer+NAS` introduce additional control logic and auxiliary computations without directly reducing the core workload of the model. As a result, they incur moderate computational overheads, increasing the average epoch time by 13.96% and 38.21%, respectively. These overheads are primarily associated with gating networks, routing decisions, and architecture search updates.

When all components are combined, the overall computational burden reflects a balance between these effects. While `MoE` provides computational savings through sparse execution, the additional complexity introduced by `GateSkip`

TABLE 6. Computational burden of each architectural component. We report the total number of parameters and the average training time per epoch, together with their variation relative to the baseline.

Configuration	Params (M)	Δ Params (M)	Epoch Time (s)	Δ Time (%)
Baseline	15.98	—	0.237	—
`+MoE`	20.84	+4.86	0.140	-41.13%
`+GateSkip`	15.98	≈0	0.270	+13.96%
`+HeadComposer+NAS`	15.98	+0.00008	0.328	+38.21%
`+All`	20.84	+4.86	0.256	+7.83%

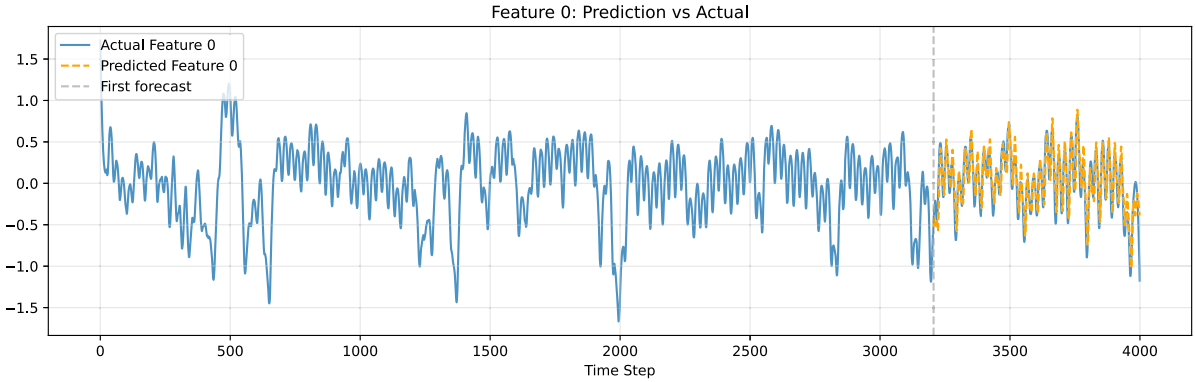


FIGURE 9. Forecasting performance on TPP 2. The model accurately tracks the underlying temporal dynamics, capturing both smooth trends and moderate fluctuations, with minor deviations during abrupt transitions.

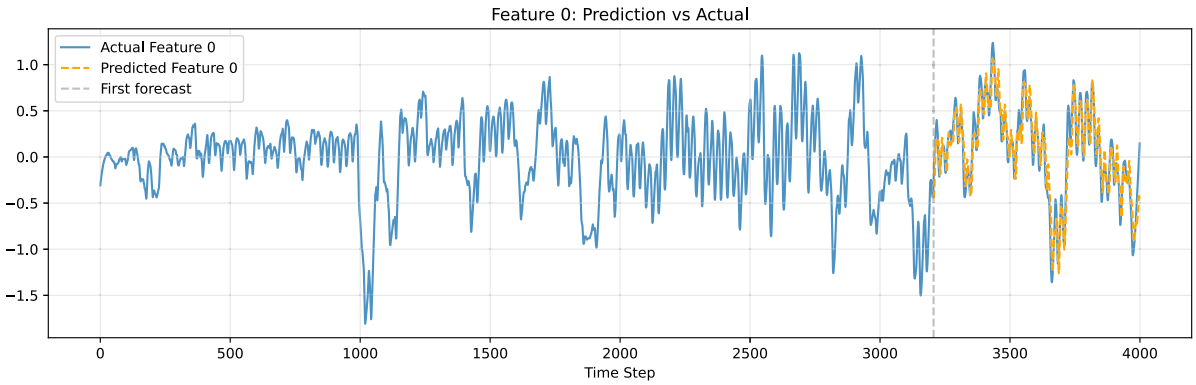


FIGURE 10. Forecasting performance on TPP 3. The predictions closely follow the ground truth across the entire horizon, demonstrating improved alignment compared to TPP 2, particularly in regions of higher variability.

and `HeadComposer+NAS` leads to a modest net increase of 7.83% in epoch time relative to the baseline.

G. GENERALIZATION TO OTHER THERMAL POWER PLANTS

To evaluate the robustness and transferability of the proposed framework, we extend our analysis to additional thermal power plants (TPPs) not used during model development. This experiment assesses whether the learned representations and architectural adaptations generalize to distinct operational regimes, characterized by different generation patterns, noise profiles, and non-stationary dynamics.

Specifically, we consider two additional datasets, denoted as **TPP 2** and **TPP 3**, drawn from the same source described in Section IV, but corresponding to different plants with independent temporal behaviors. The model is applied under the same preprocessing and architectural configuration as in the main experiment, without any modification to the underlying design, ensuring a fair assessment of generalization capability.

1) QUANTITATIVE RESULTS

The forecasting performance on the additional thermal power plants is summarized as follows:

- **TPP 2**: MSE = 0.045, RMSE = 0.212, MAE = 0.168

- **TPP 3**: MSE = 0.044, RMSE = 0.209, MAE = 0.163

These results indicate strong generalization capability, with both datasets achieving low MSE values ($\approx$ 0.04), demonstrating that the model maintains predictive accuracy even when applied to unseen plants. Notably, TPP 3 exhibits slightly better performance across all metrics, suggesting that its temporal structure is more aligned with the learned inductive biases of the model.

2) QUALITATIVE ANALYSIS

Figures 9 and 10 illustrate the predicted versus actual time series for both datasets.

Visual inspection confirms that the model effectively captures both low-frequency trends and higher-frequency variations across different thermal power plants. In TPP 2, the model maintains accurate tracking with slight lag during sharp transitions, while in TPP 3, the alignment between predicted and actual signals is consistently tighter, indicating better adaptation to its temporal structure.

3) DISCUSSION

The strong performance across distinct thermal power plants highlights an important property of the proposed framework: its ability to generalize beyond the training distribution without requiring architecture redesign or dataset-specific tuning.

This generalization capability can be attributed to three complementary mechanisms. First, the `HeadComposer` enables adaptive preprocessing, allowing the model to reweight transformations (e.g., normalization, decomposition, and convolutional filtering) according to the statistical properties of each dataset. Second, `GateSkip` dynamically filters irrelevant or noisy tokens, which is particularly beneficial when transferring across datasets with different noise characteristics. Third, the MoE layers provide conditional computation, allowing the model to specialize its internal representations for different temporal regimes.

H. LIMITATIONS AND APPLICABILITY BOUNDARIES

In low signal-to-noise ratio scenarios, the VMD-based decomposition may fail to isolate meaningful modes, leading to degraded feature representations and reduced forecasting accuracy. Similarly, during sudden load changes or abrupt regime shifts, the model may struggle to adapt due to its reliance on learned temporal patterns and smoothed preprocessing, which can delay responsiveness to sharp transitions. These limitations indicate that the framework is best suited for moderately noisy and relatively stable environments, and its applicability may be constrained in highly volatile or disturbance-driven settings.

V. CONCLUSION

This study presented an integrated framework for thermal energy forecasting that unifies signal decomposition, neural architecture search, and efficient transformer-based modeling. By combining a hyperparameter-optimized hierarchical VMD with a differentiable preprocessing search, `GateSkip`, and sparse MoE routing, the proposed approach effectively addresses the challenges of non-stationarity, noise amplification, and latency inherent to thermal generation data.

Experimental evaluations on real-world thermal datasets demonstrate consistent and substantial performance improvements over state-of-the-art baselines, achieving up to 15% lower forecasting errors while maintaining computational efficiency. The results highlight the benefits of jointly optimizing preprocessing and model structure, where adaptive decomposition and architecture search enable more informative representations and robust generalization under fluctuating operational conditions.

The synergy between decomposition-based denoising, NAS-driven head selection, and conditional expert routing yields a scalable solution adaptable to diverse thermal and hybrid energy systems. Beyond forecasting accuracy, the model's efficiency-oriented components, such as token pruning and load-balanced MoE, facilitate deployment in real-time and resource-constrained environments.

The proposed framework achieves strong performance under the evaluated conditions, but its applicability is subject to important limitations. The reliance on bilevel optimization requires sufficient data, and performance may degrade in low-sample-size scenarios due to overfitting or unstable architecture selection. Although VMD improves noise robustness, extremely low signal-to-noise conditions can impair decomposition quality and propagate errors. In addition, the computational cost of NAS and MoE components may restrict deployment in resource-constrained environments. These limitations define the scope of applicability and motivate future work on data efficiency, noise robustness, and long-horizon forecasting.

Future research will extend this framework toward multi-horizon probabilistic forecasting, cross-plant transfer learning, and real-time adaptive retraining. These directions aim to further enhance the interpretability, scalability, and operational value of deep learning models for intelligent thermal energy management.

REFERENCES

[1] W. G. Buratto, R. N. Muniz, A. Nied, C. F. D. O. Barros, R. Cardoso, and G. V. Gonzalez, "Wavelet CNN-LSTM time series forecasting of electricity power generation considering biomass thermal systems," *IET Gener., Transmiss. Distribution*, vol. 18, no. 21, pp. 3437–3451, Nov. 2024.

[2] O. Maliarenko, N. Maistrenko, H. Kuts, V. Stanytsina, and O. Teslenko, "Two-stage method for forecasting thermal energy demand using the direct account method," in *Studies in Systems, Decision and Control*, 2023, pp. 71–85.

[3] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, VIC, Australia: OTexts, 2018.

[4] H. Li, S. He, J. Yuan, and C. Wang, "A wind power prediction method integrating dynamic multi-scale spatio-temporal modelling, adaptive multi-strategy local decomposition, and meta-learning ensemble model," *Energy*, vol. 340, Dec. 2025, Art. no. 139329.

[5] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, Feb. 2014.

[6] G. Velev and S. Lessmann, "Neural architecture search for global multi-step forecasting of energy production time series," 2025, *arXiv:2511.00035*.

[7] Z. Zhou, Y.-J. Xiong, J.-C. Zhang, C.-M. Xia, and X.-J. Xie, "Wavelet mixture of experts for time series forecasting," *Expert Syst. Appl.*, vol. 2025, Apr. 2025, Art. no. 132399.

[8] R. N. Muniz, S. F. Stefenon, W. G. Buratto, A. Nied, R. Cardoso, C. K. Yamaguchi, and K.-C. Yow, "Time series forecasting based on multi-criteria optimization for model and filter selection applied to hydroelectric power plants," *Energy*, vol. 337, Nov. 2025, Art. no. 138688.

[9] A. A. Masrur Ahmed, N. Bailek, L. Abualigah, K. Bouchouicha, A. Kuriqi, A. Sharifi, P. Sareh, A. M. G. Al khatib, P. Mishra, I. Colak, and E.-S.-M. El-Kenawy, "Global control of electrical supply: A variational mode decomposition-aided deep learning model for energy consumption prediction," *Energy Rep.*, vol. 10, pp. 2152–2165, Nov. 2023.

[10] M. A. Lahyani and M. Amayri, "Explainable hybrid deep learning model with attention mechanism for short-term load forecasting," *Sustain. Cities Soc., Adv.*, vol. 1, no. 1, Dec. 2025, Art. no. 100003.

[11] L. O. Seman, S. F. Stefenon, K.-C. Yow, L. D. S. Coelho, and V. C. Mariani, "Fourier-enhanced sequence-to-sequence latent graph neural networks for multi-node spatiotemporal forecasting in a hydroelectric reservoir," *Eng. Appl. Artif. Intell.*, vol. 167, Mar. 2026, Art. no. 113939.

[12] L. O. Seman, S. F. Stefenon, K.-C. Yow, L. D. S. Coelho, and V. C. Mariani, "Multi-step short-term solar energy forecasting using Fourier-enhanced BiLSTM and neural additive models," *Renew. Energy*, vol. 257, Feb. 2026, Art. no. 124738.

[13] E. C. da Silva, E. C. Finardi, and S. F. Stefenon, "Enhancing hydroelectric inflow prediction in the Brazilian power system: A comparative analysis of machine learning models and hyperparameter optimization for decision support," *Electric Power Syst. Res.*, vol. 230, May 2024, Art. no. 110275.

[14] L. S. Aquino, L. O. Seman, V. C. Mariani, L. D. S. Coelho, S. F. Stefenon, and G. V. González, "Spatiotemporal wind energy forecasting: A comprehensive survey and a deep equilibrium-based case study with StemGNN," *IEEE Access*, vol. 13, pp. 131461–131482, 2025.

[15] C. V. Zuege, S. F. Stefenon, C. K. Yamaguchi, V. C. Mariani, G. V. Gonzalez, and L. D. S. Coelho, "Wind speed forecasting approach using conformal prediction and feature importance selection," *Int. J. Electr. Power Energy Syst.*, vol. 168, Jul. 2025, Art. no. 110700.

[16] S. F. Stefenon, L. O. Seman, N. F. Sopelsa Neto, L. H. Meyer, V. C. Mariani, and L. D. S. Coelho, "Group method of data handling using Christiano–Fitzgerald random walk filter for insulator fault prediction," *Sensors*, vol. 23, no. 13, p. 6118, Jul. 2023.

[17] L. Liu, J. Zhang, and S. Xue, "Photovoltaic power forecasting: Using wavelet threshold denoising combined with VMD," *Renew. Energy*, vol. 249, Aug. 2025, Art. no. 123152.

[18] J. P. Matos-Carvalho, S. F. Stefenon, K.-C. Yow, R. G. Ovejero, and V. R. Q. Leithardt, "N-BEATS neural network applied for insulator fault prediction considering EMD methods," in *Proc. 5th Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Oct. 2025, pp. 1–6.

[19] X. Sun and H. Liu, "Multivariate short-term wind speed prediction based on PSO-VMD-SE-ICEEMDAN two-stage decomposition and att-S2S," *Energy*, vol. 305, Oct. 2024, Art. no. 132228.

[20] S. Hu, X. Kong, D. Zhang, D. Li, X. Huo, and C. Pang, "Electricity and heat load forecasting for integrated energy system based on variational modal decomposition and deep neural networks," in *Proc. 4th Int. Conf. Electr. Eng. Control Sci. (IC2ECS)*, Dec. 2024, pp. 1093–1096.

[21] J. Zou, Z. Wang, Z. Zhu, and Z. Tan, "Explainable and physics-constrained PV power prediction via a hybrid framework integrating secondary decomposition and improved transformer-LSTM," *Energy*, vol. 347, Mar. 2026, Art. no. 140314.

[22] Y. Pan, Z. Wang, Z. Tan, and Z. Zhu, "Interpretable photovoltaic power modeling via Kolmogorov–Arnold network and timeGAN hybrid architecture with regime-aware data augmentation," *Sol. Energy*, vol. 302, Dec. 2025, Art. no. 114022.

[23] M. Yu, D. Niu, K. Wang, R. Du, X. Yu, L. Sun, and F. Wang, "Short-term photovoltaic power point-interval forecasting based on double-layer decomposition and WOA-BiLSTM-attention and considering weather classification," *Energy*, vol. 275, Jul. 2023, Art. no. 127348.

[24] Z. Sun, M. Zhao, and G. Zhao, "Hybrid model based on VMD decomposition, clustering analysis, long short memory network, ensemble learning and error complementation for short-term wind speed forecasting assisted by flink platform," *Energy*, vol. 261, Dec. 2022, Art. no. 125248.

[25] W. Feng, R. Tao, J. Cartlidge, and J. Zheng, "Vmdnet: Temporal leakage-free variational mode decomposition for electricity demand forecasting," 2026, *arXiv:2509.15394*.

[26] Y. Xu, X. Huang, X. Zheng, Z. Zeng, and T. Jin, "VMD-ATT-LSTM electricity price prediction based on grey wolf optimization algorithm in electricity markets considering renewable energy," *Renew. Energy*, vol. 236, Dec. 2024, Art. no. 121408.

[27] Y. Qin, M. Zhao, Q. Lin, X. Li, and J. Ji, "Data-driven building energy consumption prediction model based on VMD-SA-DBN," *Mathematics*, vol. 10, no. 17, p. 3058, Aug. 2022.

[28] X. Bai, M. Li, Z. Di, W. Dong, J. Liang, J. Zhang, and H. Sun, "Open circuit fault diagnosis of wind power converter based on VMD energy entropy and time domain feature analysis," *Energy Sci. Eng.*, vol. 12, no. 3, pp. 577–595, Mar. 2024.

[29] L. Fugang, M. Guangwen, S. Chen, and W. Huang, "An ensemble modeling approach to forecast daily reservoir inflow using bidirectional long- and short-term memory (Bi-LSTM), variational mode decomposition (VMD), and energy entropy method," *Water Resour. Manage.*, vol. 35, no. 9, pp. 2941–2963, Jul. 2021.

[30] X. Tang, Y. Xia, L. Jiang, D. Xiong, L. Wang, and Y. Wang, "Dynamic adaptive hierarchical TCN driven by IHOA-VMD optimization for short term load forecasting," *Energy*, vol. 335, Oct. 2025, Art. no. 138074.

[31] H. Xue, G. Wang, X. Li, and F. Du, "Predictive combination model for CH_4 separation and CO_2 sequestration with CO_2 injection into coal seams: VMD-STA-BiLSTM-ELM hybrid neural network modeling," *Energy*, vol. 313, Dec. 2024, Art. no. 133744.

[32] C. Zhai, X. He, Z. Cao, M. Abdou-Tankari, Y. Wang, and M. Zhang, "Photovoltaic power forecasting based on VMD-SSA-transformer: Multidimensional analysis of dataset length, weather mutation and forecast accuracy," *Energy*, vol. 324, Jun. 2025, Art. no. 135971.

[33] Q. Huang, X. Tan, X. Deng, D. Song, G. Huang, J. Yang, L. Liao, M. Talaat, and S. Evgeny, "LiDAR measurement modeling and rotor equivalent wind speed prediction based on VMD-CED-splGRU," *Energy*, vol. 340, Dec. 2025, Art. no. 139394.

[34] Y. Su, Z. Wang, Z. Dong, X. Hua, T. Ye, Z. Song, and Y. Shao, "Frequency-aware ultra-short-term wind power forecasting using CEEMDAN–VMD–SE and transformer–GRU networks," *Energy*, vol. 338, Nov. 2025, Art. no. 138715.

[35] Z. Gong, A. Wan, Y. Ji, K. Al-Bukhaiti, and Z. Yao, "Improving short-term offshore wind speed forecast accuracy using a VMD-PE-FCGRU hybrid model," *Energy*, vol. 295, May 2024, Art. no. 131016.

[36] X. Wu, D. Wang, M. Yang, and C. Liang, "CEEMDAN-SE-HDBSCAN-VMD-TCN-BiGRU: A two-stage decomposition-based parallel model for multi-altitude ultra-short-term wind speed forecasting," *Energy*, vol. 330, Sep. 2025, Art. no. 136660.

[37] Z. Fu, H. Qian, W. Wei, X. Chu, F. Yang, C. Guo, and F. Wang, "An informer-BiGRU-temporal attention multi-step wind speed prediction model based on spatial–temporal dimension denoising and combined VMD decomposition," *Energy*, vol. 326, Jul. 2025, Art. no. 136265.

[38] C. Zhang, Z. Li, Y. Ge, Q. Liu, L. Suo, S. Song, and T. Peng, "Enhancing short-term wind speed prediction based on an outlier-robust ensemble deep random vector functional link network with AOA-optimized VMD," *Energy*, vol. 296, Jun. 2024, Art. no. 131173.

[39] S. Parri, K. Teeparthi, and V. Kosana, "A hybrid methodology using VMD and disentangled features for wind speed forecasting," *Energy*, vol. 288, Feb. 2024, Art. no. 129824.

[40] W. Liu, Y. Bai, X. Yue, R. Wang, and Q. Song, "A wind speed forcasting model based on rime optimization based VMD and multi-headed self-attention-LSTM," *Energy*, vol. 294, May 2024, Art. no. 130726.

[41] M. Yu, D. Niu, T. Gao, K. Wang, L. Sun, M. Li, and X. Xu, "A novel framework for ultra-short-term interval wind power prediction based on RF-WOA-VMD and BiGRU optimized by the attention mechanism," *Energy*, vol. 269, Apr. 2023, Art. no. 126738.

[42] K. Sareen, B. K. Panigrahi, T. Shikhola, and A. Chawla, "A robust denoising autoencoder imputation and VMD algorithm based deep learning technique for short-term wind speed prediction ensuring cyber resilience," *Energy*, vol. 283, Nov. 2023, Art. no. 129080.

[43] Z. Hou, Y. Zhang, X. Cheng, and X. Ye, "Photovoltaic power forecasting based on variational mode decomposition and long short-term memory neural network," *Energies*, vol. 18, no. 13, p. 3572, Jul. 2025.

[44] C. Xun, L. Liu, and X. Hu, "Research on short-term power generation forecasting based on ensemble learning," in *Proc. IEEE 4th Int. Conf. Power, Electron. Comput. Appl. (ICPECA)*, Jan. 2024, pp. 185–190.

[45] Z. Cui, Q. Zhang, Z. Zhao, M. Yang, P. Kong, S. Cai, J. Zhang, J. Xu, J. Wu, R. Pandey, and T. Li, "A two-stage probabilistic forecasting framework for dam displacement: Evidence from a Chinese watershed," *J. Hydrol., Regional Stud.*, vol. 62, Dec. 2025, Art. no. 102822.

[46] Y. Gao, J. Wu, Y. Yang, Z. Wang, and Z. Ding, "Frequency-aware multi-task forecasting for integrated energy systems via variational mode decomposition and convolution-attention encoding," *IEEE Trans. Smart Grid*, vol. 17, no. 1, pp. 818–831, Jan. 2026.

[47] Z. Cui, T. Li, Z. Ding, X. Li, and J. Wu, "Probabilistic oil price forecasting with a variational mode decomposition-gated recurrent unit model incorporating pinball loss," *Data Sci. Manage.*, vol. 8, no. 3, pp. 237–247, Sep. 2025.

[48] S. Zhang, Z. Zhao, J. Wu, Y. Jin, D.-S. Jeng, S. Li, G. Li, and D. Ding, "Solving the temporal lags in local significant wave height prediction with a new VMD-LSTM model," *Ocean Eng.*, vol. 313, Dec. 2024, Art. no. 119385.

[49] S. M. J. Jalali, G. J. Osório, S. Ahmadian, M. Lotfi, V. M. A. Campos, M. Shafie-khah, A. Khosravi, and J. P. S. Catalão, "New hybrid deep neural architectural search-based ensemble reinforcement learning strategy for wind power forecasting," *IEEE Trans. Ind. Appl.*, vol. 58, no. 1, pp. 15–27, Jan. 2022.

[50] T.-H. Pei, G. Vishwakarma, and C.-W. Tsai, "A training-free neural architecture search for electricity load prediction," in *Proc. Int. Conf. Intell. Comput. Emerg. Appl.*, Nov. 2024, pp. 129–135.

[51] F. Zito, V. Cutello, and M. Pavone, "Data-driven forecasting and its role in enhanced decision-making," *Eng. Appl. Artif. Intell.*, vol. 154, Aug. 2025, Art. no. 110934.

[52] H. Jin, K. Zhang, S. Fan, H. Jin, and B. Wang, "Wind power forecasting based on ensemble deep learning with surrogate-assisted evolutionary neural architecture search and many-objective federated learning," *Energy*, vol. 308, Nov. 2024, Art. no. 133023.

[53] P. C. Chiu, R. R. Ponnusamy, and J. Adeline Sneha, "Enhanced channel-temporal transformer with neural architecture search for multivariate time series forecasting," in *Proc. IEEE 7th Int. Conf. Big Data Artif. Intell. (BDAI)*, Jul. 2024, pp. 60–68.

[54] Z. Lyu, A. Ororbia, and T. Desell, "Online evolutionary neural architecture search for multivariate non-stationary time series forecasting," *Appl. Soft Comput.*, vol. 145, Sep. 2023, Art. no. 110522.

[55] H. Mo, L. L. Custode, and G. Iacca, "Evolutionary neural architecture search for remaining useful life prediction," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107474.

[56] T. Kundu and S. Abirami, "Enhancing deep neural network architecture in spatio- temporal forecasting through neural architecture search," in *Proc. Int. Conf. Recent Adv. Electr., Electron., Ubiquitous Commun., Comput. Intell. (RAEEUCCI)*, Apr. 2024, pp. 1–5.

[57] C. Xie, B. Xue, M. Yang, M. Zhang, Z. Xu, J. Wang, S. Chen, and Y. Liu, "Evolutionary neural architecture search for automatically designing CNN-GRU-attention neural networks for turntable servo systems," *Expert Syst. Appl.*, vol. 283, Jul. 2025, Art. no. 127765.

[58] L. O. Seman, L. S. Aquino, S. F. Stefenon, K.-C. Yow, V. C. Mariani, and L. D. S. Coelho, "Simultaneously anomaly detection and forecasting for predictive maintenance using a zero-cost differentiable architecture search-based network," *Comput. Ind. Eng.*, vol. 208, Oct. 2025, Art. no. 111412.

[59] J. B. Thomas and S. K. V., "Neural architecture search algorithm to optimize deep transformer model for fault detection in electrical power distribution systems," *Eng. Appl. Artif. Intell.*, vol. 120, Apr. 2023, Art. no. 105890.

[60] M. Chen, H. Peng, J. Fu, and H. Ling, "AutoFormer: Searching transformers for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12250–12260.

[61] H. Mo and G. Iacca, "Evolutionary neural architecture search on transformers for RUL prediction," *Mater. Manuf. Processes*, vol. 38, no. 15, pp. 1881–1898, Nov. 2023.

[62] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, "Neural architecture search for transformers: A survey," *IEEE Access*, vol. 10, pp. 108374–108412, 2022.

[63] V. Pentsos, S. Tragoudas, J. Wibbenmeyer, and N. Khdeer, "A hybrid LSTM-transformer model for power load forecasting," *IEEE Trans. Smart Grid*, vol. 16, no. 3, pp. 2624–2634, May 2025.

[64] T. Wang, Y. W. Dong, and Q. Wang, "HTMformer: Hybrid time and multivariate transformer for time series forecasting," 2025, *arXiv:2510.07084*.

[65] E. Elakkiya, A. S. Raj, A. Balakrishnan, B. Sanisetty, and R. B. Bandaru, "Stacked hybrid model for load forecasting: Integrating transformers, ANN, and fuzzy logic," *Sci. Rep.*, vol. 15, no. 1, p. 19688, Jun. 2025.

[66] H. Chen, H. Huang, Y. Zheng, and B. Yang, "A load forecasting approach for integrated energy systems based on aggregation hybrid modal decomposition and combined model," *Appl. Energy*, vol. 375, Dec. 2024, Art. no. 124166.

[67] Z. Chen, Y. Deng, Q. Gu, Y. Li, and Y. Wu, "Towards understanding the mixture-of-experts layer in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23049–23062.

[68] J. Hu, P. Duan, X. Cao, Q. Xue, B. Zhao, X. Zhao, X. Yuan, and C. Zhang, "A multi-energy load forecasting method based on the mixture-of-experts model and dynamic multilevel attention mechanism," *Energy*, vol. 324, Jun. 2025, Art. no. 135947.

[69] X. Ma, S. Ge, F. Yang, X. Li, Y. Chen, M. Ma, W. Zhang, and Z. Liu, "TimeExpert: Boosting long time series forecasting with temporal mix of experts," 2025, *arXiv:2509.23145*.

[70] W. Tan, Y. Wang, Q. Fu, Y. Lu, and J. Chen, "A collaborative mixture of experts framework for building energy consumption prediction," *Building Simul.*, vol. 18, no. 10, pp. 1–16, Oct. 2025.

[71] J. Bergstra and Y. Bengio, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.

[72] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[73] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed., Hoboken, NJ, USA: Wiley, 2015.

[74] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *Proc. Int. Conf. Learn. Represent.*, vol. 10, 2022, pp. 1–25.

[75] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A seasonal-trend decomposition procedure based on loess," *J. Off. Statist.*, vol. 6, no. 1, pp. 3–73, 1990.

[76] M. Binkowski, G. Marti, and P. Donnat, "Autoregressive convolutional neural networks for asynchronous time series," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 580–589.

[77] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–24.

[78] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–31.

[79] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[80] H. Wang, J. Xu, C. Zhang, B. Hu, X. Sun, and J. Lu, "MICN: Multi-scale isotropic convolutional network for time series forecasting," *Neurocomputing*, vol. 511, pp. 298–309, Jun. 2022.

[81] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–18.

[82] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, "TSMixer: An all-MLP architecture for time series forecasting," 2023, *arXiv:2303.06053*.

[83] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

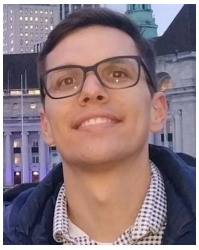
[84] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, Feb. 2024, Art. no. 127063.

[85] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[86] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

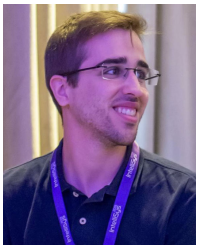
[87] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," 2018, *arXiv:1806.09055*.

LAIO ORIEL SEMAN received the Ph.D. degree in electrical engineering from the Federal University of Santa Catarina, in 2017. His research interests include strategies for static and dynamic optimization, along with applications in traffic control, cyber-physical systems, and oil and gas production systems.



STEFANO FRIZZO STEFENON received the Ph.D. degree in electrical engineering from the State University of Santa Catarina (UDESC), Brazil, in 2021, and the Ph.D. degree in computer science and artificial intelligence from Università degli Studi di Udine (UNIUD), Italy, in 2025.

He was a Postdoctoral Fellow Researcher with the Faculty of Engineering and Applied Science, University of Regina (UofR), Saskatchewan, Canada. Currently, he is a Professor (Adjunct) with Lisbon School of Engineering (ISEL), Polytechnic University of Lisbon (IPL), Portugal. His research interests include electrical power systems, deep learning, computer vision, and time series forecasting.



JOÃO PEDRO MATOS-CARVALHO received the M.Sc. (Hons.) and Ph.D. degrees in electrical and computer engineering from FCT NOVA, Portugal, in 2017 and 2021, respectively. He is currently Assistant Professor with the Department of Informatics, Faculty of Sciences of the University of Lisbon. Since 2025, he has been an Integrated Member of LASIGE and the Collaborator of the Center of Technology and Systems (CTS), UNINOVA, and COPELABS. He won the highly competitive scientific employment stimulus (CEEC institutional) FCT grant, in 2021. He has published more than 60 papers in international journals and international conferences in the fields of remote sensing, pattern recognition, machine learning, sensor networks, and signal processing, with significant recognition and impact in the research community.

ADEMIR NIED (Member, IEEE) received the B.E. degree in electrical engineering from the Federal University of Santa Maria, Santa Maria, Brazil, in 1987, the M.S. degree in industrial informatics from the Federal Technological University of Parana, Curitiba, Brazil, in 1995, and the Ph.D. degree in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2007.

Since 1996, he has been a member of the Faculty of the State University of Santa Catarina, Joinville, Brazil, where he is an Associate Professor with the Department of Electrical Engineering. From 2015 to 2016, he was a Visiting Professor with the Wisconsin Electric Machines and Power Electronics Consortium, University of Wisconsin–Madison, Madison, WI, USA. His teaching and research interests include electrical machines, control of electrical drives, neural networks, and renewable energy.

KIN-CHOONG YOW (Senior Member, IEEE) received the B.E. (Elect) degree (Hons.) from the National University of Singapore, in 1993, and the Ph.D. degree from Cambridge University, U.K., in 1998.

He joined the University of Regina (UofR), in September 2018, where he is currently a Full Professor with the Faculty of Engineering and Applied Science, Canada. Prior to joining UofR, he was an Associate Professor with Gwangju Institute of Science and Technology (GIST), Republic of Korea, from 2013 to 2018. He was a Professor with Shenzhen Institutes of Advanced Technology (SIAT), China, from 2012 to 2013, and an Associate Professor with Nanyang Technological University (NTU), Singapore, from 1998 to 2013. From 1999 to 2005, he was the Sub-Dean of Computer Engineering, NTU. From 2006 to 2008, he was the Associate Dean of Admissions of NTU.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614