NOT ALL FREQUENCIES ARE EQUAL: ASYMMETRIC SPECTRAL MODELING FOR CLOUD WORKLOAD FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Workload forecasting is essential for cloud resource management, yet real-world workloads often exhibit complicated periodic patterns across multiple temporal scales. Although frequency-domain forecasting provides an effective representation for such patterns, existing methods commonly apply homogeneous transformations across the spectrum, overlooking the distinct structures of low- and high-frequency components. To address this issue, we propose DuoFreq, a lightweight frequency-domain forecasting framework based on Dual-Band Spectral Modulation (DBSM). DuoFreq separates the spectrum into low- and high-frequency bands and models them with specialized mechanisms. Smooth Low-Frequency Modulation (SLFM) preserves and adapts low-frequency structures through smooth residual modulation, while Local-Global High-Frequency Modulation (LG-HFM) captures neighboring and band-wide dependencies, together with Complex Spectral Transformation (CST) for complex-valued high-frequency modeling. Extensive experiments on five real-world cloud workload datasets show that DuoFreq achieves the best MSE in 17 of 20 forecasting settings and the best MAE in 18, with average reductions of 9.48% and 7.09%, respectively. DuoFreq also generalizes well to unseen workload instances while maintaining a favorable accuracy–efficiency trade-off with approximately 0.61M parameters.

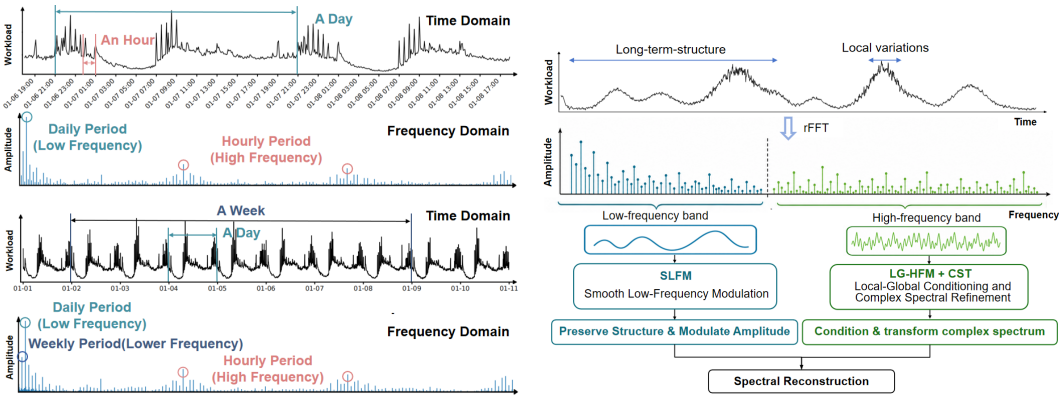
1 INTRODUCTION

Modern cloud platforms host large numbers of applications and services with dynamically changing resource demands Ye et al. (2025). Accurate workload forecasting is critical for proactive auto-scaling, resource provisioning, and scheduling, where prediction quality directly affects service quality and resource utilization Golshani & Ashtiani (2021); Beikzadeh Abbasi et al. (2023); Calheiros et al. (2015). Large-scale cloud environments also place stringent requirements on computational efficiency, since workload forecasts may need to be generated frequently for many service instances Ye et al. (2025). Achieving both accurate and efficient workload forecasting is therefore important for practical cloud resource management.

Multi-scale periodicity is common in real-world time series Wu et al. (2021); Zhou et al. (2022b); Wu et al. (2023); Wang et al. (2024); Dai et al. (2024); Challu et al. (2023). In cloud systems, this challenge takes a particularly complex form because heterogeneous services generate fine-grained workload traces in which short-term fluctuations coexist with hourly, daily, and longer-term recurring patterns Ye et al. (2025). These periodic components can be difficult to disentangle directly in the time domain. As illustrated in Figure 1a, spectral transformation makes workload patterns at different temporal scales more distinguishable. The frequency domain therefore provides a natural representation for modeling the complicated periodic structures of cloud workloads Zhou et al. (2022b;a); Xu et al. (2024); Yi et al. (2023); Ye et al. (2025).

Frequency-domain modeling has attracted increasing attention in time series forecasting. FEDformer exploits the sparse structure of time series in the frequency domain, FiLM enhances long-term dependency modeling through frequency representations, while FITS and FilterNet demonstrate the effectiveness of lightweight spectral operations Zhou et al. (2022b;a); Xu et al. (2024); Yi et al. (2024). Fremer further demonstrates the effectiveness of frequency-domain modeling for

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



(a) Multi-scale workload periodicity.

(b) Distinct low- and high-frequency structures.

Figure 1: Motivation of DuoFreq: cloud workloads exhibit multi-scale periodicity and heterogeneous spectral structures, motivating asymmetric modeling of low- and high-frequency components.

fine-grained cloud workloads with complicated multi-period patterns Ye et al. (2025). These studies establish the frequency domain as an effective representation for periodic forecasting.

However, a less explored question remains: should different regions of the spectrum be modeled in the same way? As shown in Figure 1b, low- and high-frequency components of cloud workloads exhibit markedly different structural characteristics. Low-frequency components mainly represent slowly varying structures and longer-period patterns, whereas high-frequency components contain finer periodic variations and rapid workload dynamics Ye et al. (2025); Ma et al. (2026). Applying a homogeneous transformation to the entire spectrum requires the same modeling mechanism to accommodate these different behaviors. Such a design may unnecessarily disturb stable low-frequency structures while providing insufficient flexibility for complex high-frequency dynamics. Related work has also shown that heterogeneous temporal structures can benefit from specialized rather than monolithic representations Ma et al. (2026). The structural heterogeneity within the workload spectrum therefore calls for frequency-band-specific modeling.

To address this problem, we propose DuoFreq, a lightweight workload forecasting framework based on Dual-Band Spectral Modulation (DBSM). DuoFreq models the spectrum of an extended sequence constructed from the observed history and an initial future estimate. Targeting the distinct characteristics of different frequency regions, DBSM separates the spectrum into low- and high-frequency bands and applies dedicated modeling mechanisms to each band. Smooth Low-Frequency Modulation (SLFM) performs smooth residual modulation to adapt low-frequency components while preserving their underlying structure. Local-Global High-Frequency Modulation (LG-HFM) captures both neighboring frequency interactions and band-wide dependencies in the high-frequency region, while Complex Spectral Transformation (CST) further models its complex-valued representation. The modeled frequency components are combined to produce the final forecast.

We evaluate DuoFreq on five real-world cloud workload datasets against eleven competitive forecasting models across four prediction horizons. Averaged over all baseline comparisons and forecasting settings, DuoFreq reduces MSE by 9.48% and MAE by 7.09%. The component and sensitivity analyses verify the effectiveness of the proposed dual-band design. DuoFreq also generalizes well to unseen workload instances and achieves strong forecasting accuracy with approximately 0.61M parameters, demonstrating a favorable balance between forecasting effectiveness and computational efficiency.

Our contributions are summarized as follows:

- We propose DuoFreq, a lightweight workload forecasting framework built upon Dual-Band Spectral Modulation, which explicitly separates low- and high-frequency components and assigns dedicated modeling mechanisms to their distinct spectral structures.
- We design Smooth Low-Frequency Modulation to preserve and adapt low-frequency structures, and Local-Global High-Frequency Modulation together with Complex Spectral Transformation to capture local and band-wide dependencies in high-frequency dynamics.

- Extensive experiments on five real-world cloud workload datasets demonstrate the effectiveness of DuoFreq. DuoFreq achieves strong forecasting accuracy across multiple horizons, generalizes well to unseen workload instances, and maintains a favorable balance between forecasting accuracy and computational efficiency.

2 Related Work

Workload Forecasting in Cloud Systems. Deep learning has been widely applied to cloud workload forecasting to model nonlinear and highly variable workload dynamics. L-PAW and esDNN employ recurrent architectures, while SG-CBA and VTGAN explore attention-based and generative forecasting Chen et al. (2020); Xu et al. (2022); Chen et al. (2022); Maiyza et al. (2023). MSCNet further adopts multi-scale modeling to capture periodic patterns and long-term dependencies Zhao et al. (2025). These studies highlight the importance of modeling complex temporal structures in cloud workloads.

Frequency-Domain Time Series Forecasting. Frequency-domain forecasting has been widely explored for modeling periodic and long-range temporal structures. FEDformer and FiLM leverage spectral representations, while FITS and FilterNet adopt lightweight frequency-domain transformations Zhou et al. (2022b;a); Xu et al. (2024); Yi et al. (2024). Fremer extends spectral modeling to cloud workloads Ye et al. (2025), and FP-MLP reduces frequency bias through band-wise modeling Lu et al. (2025). However, existing methods still largely lack explicit asymmetric treatment of low- and high-frequency regions. DuoFreq addresses this gap through band-specific spectral modeling.

Heterogeneous Temporal Modeling. Several forecasting methods model heterogeneous temporal structures with specialized mechanisms. Autoformer and DLinear separate trend and seasonal components, while TimeMixer captures patterns across multiple scales Wu et al. (2021); Zeng et al. (2023); Wang et al. (2024). N-HiTS, MICN, and SCINet further model heterogeneous temporal dynamics through hierarchical multi-rate interpolation, multi-scale local-global modeling, and multi-resolution interaction, respectively Challu et al. (2023); Wang et al. (2023); Liu et al. (2022). MixLinear models local patterns in the time domain and global patterns in the frequency domain Ma et al. (2026). In contrast, DuoFreq distinguishes and asymmetrically models low- and high-frequency components within the frequency domain.

3 Methodology

In this section, we present the architecture of DuoFreq. DuoFreq is built upon DBSM, which explicitly separates low- and high-frequency components and models them according to their different spectral characteristics. Its low-frequency branch employs SLFM to preserve and adapt low-frequency structures, while its high-frequency branch combines LG-HFM and CST to model fine-grained spectral variations and complex-valued high-frequency representations.

As illustrated in Figure 2, given an input workload sequence $\mathbf{X} \in \mathbb{R}^{L \times C}$, where L denotes the look-back window and C denotes the number of workload variables, we center the input and use Temporal Extension (TE) to generate an initial future segment of length T. The historical sequence and the estimated future segment are concatenated and transformed into the frequency domain using the real-valued Fast Fourier Transform (rFFT)Cooley & Tukey (1965). After spectral normalization, DBSM divides the spectrum into low- and high-frequency bands. SLFM models the low-frequency components, while LG-HFM and CST process the high-frequency components. The two frequency bands are recombined and transformed back to the time domain using the inverse real-valued Fast Fourier Transform (irFFT), from which the last T points are taken as the forecasting result. The following sections describe the main components of DuoFreq.

3.1 Temporal Extension

DuoFreq models the spectrum over both the observed history and the forecasting horizon, while the future segment is unavailable at prediction time. TE therefore generates an input-dependent future initialization before spectral modeling.

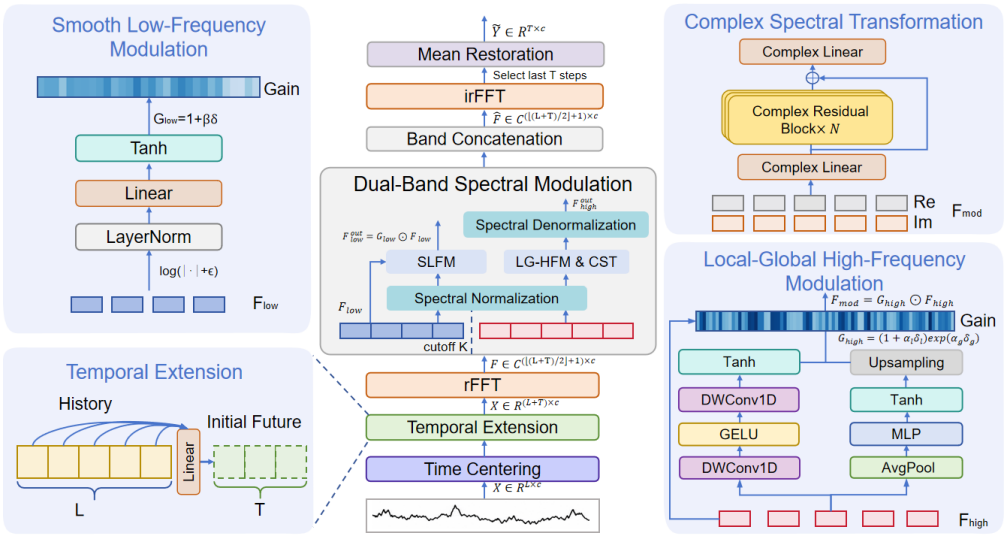


Figure 2: Model architecture of DuoFreq.

Given $\mathbf{X}$, we remove its temporal mean $\boldsymbol{\mu}_x$ and denote the centered history by $\widetilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}_x$. TE applies a shared linear projection along the temporal dimension and constructs the extended sequence as

$$\mathbf{X}_{\text{ext}} = \text{Concat}\left(\widetilde{\mathbf{X}}, \mathbf{W}_{\text{TE}}\widetilde{\mathbf{X}} + \mathbf{b}_{\text{TE}}\right), \tag{1}$$

where $\mathbf{W}_{\text{TE}} \in \mathbb{R}^{T\times L}$ and $\mathbf{b}_{\text{TE}} \in \mathbb{R}^{T}$ are shared across workload variables. The resulting $\mathbf{X}_{\text{ext}} \in \mathbb{R}^{(L+T)\times C}$ is used for subsequent spectral modeling.

3.2 Dual-Band Spectral Modulation

After temporal extension, DuoFreq models the frequency representation of the extended sequence. Low- and high-frequency components exhibit different structural characteristics: low-frequency components mainly describe slowly varying and longer-period workload structures, whereas high-frequency components contain finer periodic variations and rapid workload dynamics. Applying the same transformation to both regions requires a single mechanism to accommodate these different behaviors. DBSM instead assigns dedicated modeling mechanisms to the two frequency bands.

Spectral Normalization and Band Separation. We transform $\mathbf{X}_{\text{ext}}$ into the frequency domain using rFFT and obtain $\mathbf{F} \in \mathbb{C}^{K\times C}$, where $K = \lfloor (L+T)/2 \rfloor + 1$. To reduce scale variation among workload spectra, we normalize the spectrum as

$$\overline{\mathbf{F}} = \frac{\mathbf{F} - \boldsymbol{\mu}_f}{\boldsymbol{\sigma}_f}, \tag{2}$$

where $\boldsymbol{\mu}_f$ is the mean of $\mathbf{F}$ over frequency bins and $\boldsymbol{\sigma}_f = \sqrt{\text{Var}(|\mathbf{F}|) + \epsilon}$ is computed from the spectral magnitude.

Given a cutoff K_l, we use the first K_l frequency bins as the low-frequency band and the remaining $K_h = K - K_l$ bins as the high-frequency band. We denote the original low-frequency coefficients by $\mathbf{F}_l$ and the corresponding normalized representations by $\overline{\mathbf{F}}_l$ and $\overline{\mathbf{F}}_h$. The original coefficients $\mathbf{F}_l$ are retained for low-frequency modulation, whereas the high-frequency branch operates on $\overline{\mathbf{F}}_h$.

Smooth Low-Frequency Modulation. Low-frequency components mainly encode smooth and slowly varying structures. Directly applying a highly flexible transformation to this region can unnecessarily alter these patterns. SLFM therefore adopts an identity-centered residual modulation that preserves the original spectrum while learning the required adjustments.

Let $\mathbf{U}_l = \log(|\overline{\mathbf{F}}_l| + \epsilon)$ denote the low-frequency log-amplitude representation. SLFM is defined as

$$\widehat{\mathbf{F}}_l = [1 + \rho\,\mathbf{M} \odot \tanh\left(\mathcal{P}_l(\mathbf{U}_l)\right)] \odot \mathbf{F}_l, \tag{3}$$

where $\mathcal{P}_l$ consists of layer normalization Ba et al. (2016) followed by a shared linear projection, ρ controls the residual scale, and $\mathbf{M}$ masks the DC component so that it remains unchanged. The linear predictor is zero-initialized, making SLFM an identity mapping at the beginning of training.

Denoting the multiplicative term in Equation 3 by $\mathbf{G}_l$, we further encourage smooth variation of the modulation across neighboring frequencies:

$$\mathcal{L}_{\text{smooth}} = \lambda \, \text{Mean}_{k,c} \left[\left(G_{l,k+1,c} - G_{l,k,c} \right)^2 \right], \tag{4}$$

where λ controls the regularization strength. This constraint allows SLFM to adapt low-frequency components without introducing abrupt changes across adjacent frequency bins.

Local-Global High-Frequency Modulation. High-frequency components exhibit more fine-grained spectral variations. Their structure involves interactions between neighboring frequency components as well as dependencies spanning a wider portion of the frequency band. LG-HFM models these two types of information through complementary local and global modulation paths.

The local path operates at the original high-frequency resolution. A depthwise convolutional block extracts neighboring spectral variations from the magnitude representation:

$$\mathbf{G}_{\text{loc}} = 1 + \alpha_{\text{loc}} \tanh\left(\mathcal{C}_{\text{loc}} \left(|\overline{\mathbf{F}}_h| \right) \right), \tag{5}$$

where $\mathcal{C}_{\text{loc}}$ contains two depthwise convolutions with a GELU activation Hendrycks & Gimpel (2016), and α_{loc} is a learnable scaling parameter.

The global path operates on a compressed spectral representation. Let $\mathbf{P}_h = \text{Pool}\big(\log(|\overline{\mathbf{F}}_h| + \epsilon) \big)$ denote the pooled log-amplitude representation. A shared MLP models dependencies over this representation, and its output is upsampled to the original high-frequency resolution:

$$\mathbf{G}_{\text{glo}} = \exp\left(\alpha_{\text{glo}} \, \text{Up}\left[\tanh\left(\mathcal{M}_{\text{glo}}(\mathbf{P}_h) \right) \right] \right), \tag{6}$$

where $\mathcal{M}_{\text{glo}}$ denotes the shared MLP and α_{glo} is learnable.

The local and global gains are combined to modulate the normalized high-frequency spectrum:

$$\mathbf{F}_h^m = \mathbf{G}_{\text{loc}} \odot \mathbf{G}_{\text{glo}} \odot \overline{\mathbf{F}}_h. \tag{7}$$

The local path preserves sensitivity to neighboring frequency variations, while the global path introduces a wider spectral receptive field. Neither path explicitly exchanges information across workload variables.

3.3 Complex Spectral Transformation

LG-HFM conditions each high-frequency coefficient using local and band-wide spectral information, but its output is obtained through real-valued modulation of the complex spectrum. To further transform the joint complex-valued high-frequency representation, we introduce Complex Spectral Transformation (CST).

CST is built upon complex-valued linear projection. For $\mathbf{Z} = \mathbf{Z}_r + j\mathbf{Z}_i$, the transformation is written as

$$\text{CLinear}(\mathbf{Z}) = (\mathbf{W}_r \mathbf{Z}_r - \mathbf{W}_i \mathbf{Z}_i) + j \left(\mathbf{W}_r \mathbf{Z}_i + \mathbf{W}_i \mathbf{Z}_r \right), \tag{8}$$

where bias terms are omitted for clarity.

For each workload variable, CST projects the complete high-frequency band into a compact complex latent space. The latent representation passes through E complex residual blocks, each consisting of complex layer normalization, two complex linear transformations, Cartesian activation, dropout, and a residual connection. A complex output projection maps the latent representation back to the original high-frequency dimension. The same transformation parameters are applied to individual workload variables, without explicit cross-channel mixing.

The transformed high-frequency representation is restored to the original spectral scale using $\boldsymbol{\mu}_f$ and $\boldsymbol{\sigma}_f$ and concatenated with $\widehat{\mathbf{F}}_l$ to recover the complete spectrum. Applying irFFT reconstructs the extended sequence. We take its last T points and add back the temporal mean $\boldsymbol{\mu}_x$ to obtain the final prediction $\widehat{\mathbf{Y}}$.

DuoFreq is optimized using the forecasting error together with the smoothness regularization introduced in SLFM:

$$\mathcal{L} = \mathrm{MSE}\left(\widehat{\mathbf{Y}}, \mathbf{Y}\right) + \mathcal{L}_{\mathrm{smooth}}. \tag{9}$$

4 EXPERIMENTS

In this section, we conduct extensive experiments to address the following research questions: **RQ1** How does DuoFreq compare with state-of-the-art time series forecasting models in workload forecasting? **RQ2** Can the key components of DuoFreq be identified as significant contributors to its forecasting performance? **RQ3** How well does DuoFreq generalize to unseen workload instances? **RQ4** Can DuoFreq achieve a favorable balance between forecasting accuracy and computational efficiency?

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate DuoFreq on five real-world cloud workload datasets, including Huawei CPU, Huawei Memory, FaaS, IaaS, and PaaS. Huawei CPU and Huawei Memory are constructed from minute-level monitoring traces, while FaaS, IaaS, and PaaS are obtained from the ByteDance CloudTimeSeriesData repository ByteDance (2025). Dataset statistics are summarized in Table 1, with preprocessing details provided in Appendix A.1.

Table 1: Statistics of the workload datasets.

Dataset	# Var.	Length	Freq.	Start	End
Huawei CPU	82	9504	5-min	2023-01-01	2023-02-02
Huawei Memory	67	9504	5-min	2023-01-01	2023-02-02
FaaS	226	2305	10-min	2022-04-02	2022-04-18
IaaS	93	3456	10-min	2023-06-30	2023-07-24
PaaS	426	7776	10-min	2024-09-01	2024-10-24

Baselines. We compare DuoFreq with eleven forecasting models, including EMAFormer Zhang et al. (2026), DUET Qiu et al. (2025), FilterNet Yi et al. (2024), FreEformer Yue et al. (2025), xPatch Stitsyuk & Choi (2025), S-Mamba Wang et al. (2025), TimeKAN Huang et al. (2025), TimePro Ma et al. (2025), DLinear Zeng et al. (2023), MSCNet Zhao et al. (2025), and TFDNet Luo et al. (2025). These methods represent competitive recent approaches for time series forecasting and provide a broad comparison across different modeling paradigms. Additional descriptions of the baseline methods are provided in Appendix A.2.

Evaluation. We use mean squared error (MSE) and mean absolute error (MAE) as the evaluation metrics. Following a unified evaluation protocol, all datasets are chronologically split into training, validation, and test sets with a ratio of $70\% : 15\% : 15\%$ and standardized using training-set statistics. We set the input length to $L = 720$ for all models and prediction horizons to $P \in \{48, 96, 144, 288\}$.

Implementation Details. DuoFreq is implemented in PyTorch Paszke et al. (2019). All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory. We use the Adam optimizer Kingma & Ba (2015) with an initial learning rate of 1×10^{-4} and a batch size of 32. MSE is used as the forecasting loss. Additional evaluation and implementation details are provided in Appendix A.3.

4.2 RQ1: WORKLOAD FORECASTING PERFORMANCE

In this section, we evaluate the forecasting performance of DuoFreq against eleven baseline models on the five workload datasets. The input length is set to 720, and the prediction horizons are configured as $P \in \{48, 96, 144, 288\}$.

As shown in Table 2, averaged over all eleven baseline models and all forecasting settings, DuoFreq reduces MSE by 9.48% and MAE by 7.09%. These results show that DuoFreq provides consistently strong forecasting accuracy across diverse cloud workloads. Detailed results for each dataset–horizon setting are provided in Appendix B.1.

Table 2: Main forecasting results averaged over four prediction horizons $P \in \{48, 96, 144, 288\}$ on five workload datasets. Lower values are better. Rankings are determined by the displayed values rounded to four decimal places; the best result is shown in bold and the second-best result is underlined.

Data	DuoFreq		EMAFormer		DUET		FilterNet		FreEformer		xPatch	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Huawei CPU	**0.6269**	**0.3942**	0.7422	0.4302	0.7251	0.4477	0.7085	0.4175	0.6782	0.4211	0.6640	0.4121
Huawei Memory	1.1057	**0.5481**	1.1397	0.5663	1.1370	0.5805	1.2051	0.5887	**1.0942**	0.5611	1.1272	0.5648
FaaS	**0.2120**	**0.2910**	0.2575	0.3268	0.2340	0.3292	0.2834	0.3431	0.2505	0.3237	0.2333	0.3086
IaaS	**0.6925**	**0.5508**	0.7107	0.5664	0.7144	0.5680	0.7292	0.5804	0.7174	0.5699	0.7270	0.5781
PaaS	**0.0589**	**0.1348**	0.0662	0.1476	0.1004	0.2281	0.0648	0.1411	0.0641	0.1446	0.0637	0.1408
Data	**S-Mamba**		**TimeKAN**		**TimePro**		**DLinear**		**MSCNet**		**TFDNet**	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Huawei CPU	0.7219	0.4249	0.6886	0.4188	0.6618	0.4116	0.7214	0.4234	0.6673	0.4042	0.6642	0.4103
Huawei Memory	1.2061	0.5922	1.1822	0.5865	1.1399	0.5618	1.2101	0.5933	1.1498	0.5657	1.1089	0.5493
FaaS	0.2696	0.3364	0.2512	0.3216	0.2702	0.3234	0.2961	0.3660	0.2479	0.3155	0.2203	0.3010
IaaS	0.7385	0.5807	0.7345	0.5719	0.7223	0.5714	0.7520	0.5878	0.7220	0.5688	0.7138	0.5653
PaaS	0.0672	0.1461	0.0653	0.1514	0.0622	0.1372	0.0682	0.1502	0.0625	0.1413	0.0634	0.1484

Table 3: Ablation study of DuoFreq averaged over four prediction horizons $P \in \{48, 96, 144, 288\}$. Lower values are better. The best result is shown in bold and the second-best result is underlined.

Model	Huawei CPU		Huawei Memory		FaaS		IaaS		PaaS	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o LG-HFM	0.6539	0.3981	1.1539	0.5600	0.2139	0.2925	0.7011	0.5568	0.0591	**0.1342**
w/o CST	0.6653	0.4114	1.1509	0.5686	0.2153	0.2944	0.7020	0.5556	0.0610	0.1415
w/o SLFM	0.6336	0.3957	1.1283	0.5521	0.2130	0.2924	0.6950	0.5521	0.0596	0.1358
w/o Spectral Norm.	0.7018	0.4146	1.2812	0.5837	0.2149	0.2936	0.6928	0.5515	0.0624	0.1401
DuoFreq	**0.6269**	**0.3942**	**1.1057**	**0.5481**	**0.2120**	**0.2910**	**0.6925**	**0.5508**	**0.0589**	0.1348

It can also be observed that DuoFreq achieves the best MSE across all four prediction horizons on Huawei CPU, FaaS, IaaS, and PaaS. The consistent performance across datasets with different temporal granularities and numbers of workload variables indicates the effectiveness of the proposed dual-band spectral modeling for heterogeneous workload patterns.

4.3 RQ2: Component Analysis

4.3.1 Ablation Study

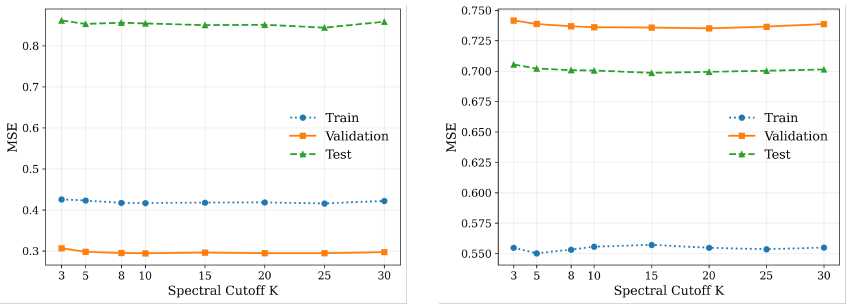
We evaluate the contributions of LG-HFM, CST, SLFM, and spectral normalization through ablation studies. The variants "w/o LG-HFM", "w/o CST", "w/o SLFM", and "w/o Spectral Norm." denote DuoFreq with the corresponding component removed. Table 3 reports the results averaged over the four prediction horizons.

As shown in Table 3, removing each component generally degrades the forecasting performance, demonstrating that all components contribute to DuoFreq. The removal of spectral normalization causes the most pronounced degradation on Huawei CPU and Huawei Memory, showing its importance for stabilizing spectral representations across heterogeneous workloads. Removing CST also consistently increases both MSE and MAE, confirming the benefit of complex-valued spectral transformation. LG-HFM and SLFM provide complementary improvements by modeling high- and low-frequency components with dedicated mechanisms.

4.3.2 Sensitivity to the Spectral Cutoff

We investigate the impact of the spectral cutoff on DuoFreq using Huawei CPU and IaaS. The input length and prediction horizon are fixed to $L = 720$ and $P = 288$, respectively, while the cutoff is varied over $\{3, 5, 8, 10, 15, 20, 25, 30\}$.

Figure 3 shows that DuoFreq remains relatively stable across a broad range of spectral cutoffs. Huawei CPU is more sensitive at small cutoff values, while IaaS exhibits smaller variations, indicating workload-dependent spectral partitions.

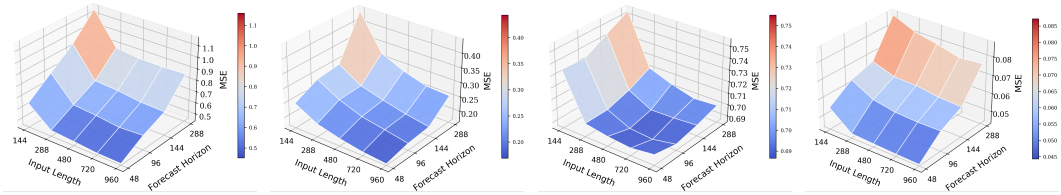


(a) Huawei CPU (b) IaaS

Figure 3: Sensitivity of DuoFreq to the spectral cutoff parameter on Huawei CPU and IaaS.

4.3.3 Input Length L and Forecast Horizon T

We evaluate DuoFreq on Huawei CPU, FaaS, IaaS, and PaaS across varying input lengths $L \in \{144, 288, 480, 720, 960\}$ and forecast horizons $T \in \{48, 96, 144, 288\}$, with the results shown in Figure 4.



(a) Huawei CPU (b) FaaS (c) IaaS (d) PaaS

Figure 4: Sensitivity of DuoFreq to the input length and forecast horizon across different workload datasets.

Increasing the input length generally reduces forecasting error, with the largest improvement from $L = 144$ to 288 and smaller gains beyond $L = 720$. Longer historical contexts also make performance more stable across forecasting horizons. At $L = 720$, increasing T from 48 to 288 raises MSE by only 1.0% on IaaS, although larger variations remain on the other datasets.

4.4 RQ3: Generalization to Unseen Workload Instances

To evaluate the generalizability of DuoFreq to unseen workload instances, we train the models on one half of the workload instances within each dataset and evaluate them on the remaining unseen instances. The detailed transfer protocol is provided in Appendix A.4.

Table 4: Results of intra-dataset unseen-instance transfer forecasting averaged over four prediction horizons $P \in \{48, 96, 144, 288\}$. Lower values are better. The best result is shown in bold and the second-best result is underlined.

Models		DuoFreq	EMAFormer	DUET	FilterNet	FreEformer	xPatch	TimePro	DLinear	MSCNet
Huawei CPU	MSE	**0.4964**	0.5319	0.5826	0.5618	0.5301	0.5324	0.5104	<u>0.5092</u>	0.5110
	MAE	**0.3817**	0.4009	0.4382	0.4182	0.4048	0.4118	0.4004	<u>0.3862</u>	0.3905
Huawei Memory	MSE	**0.7068**	0.7300	0.7314	0.7692	0.7317	<u>0.7099</u>	0.7304	0.7373	0.7310
	MAE	**0.4915**	0.5129	0.5217	0.5398	0.5145	<u>0.5017</u>	0.5055	0.5140	0.5074
FaaS	MSE	**0.1968**	0.2409	<u>0.2210</u>	0.2566	0.2394	0.2231	0.2871	0.3239	0.2379
	MAE	**0.2789**	0.3166	0.3193	0.3261	0.3197	<u>0.3002</u>	0.3275	0.3751	0.3107
IaaS	MSE	**0.7198**	0.7798	0.8321	0.8578	0.7902	0.8895	0.7669	0.7500	<u>0.7488</u>
	MAE	**0.6106**	0.6433	0.6742	0.6850	0.6500	0.6907	0.6384	0.6299	<u>0.6298</u>
PaaS	MSE	**0.0458**	0.0603	0.0887	0.0572	0.0563	0.0529	0.0517	<u>0.0468</u>	0.0508
	MAE	**0.1216**	0.1507	0.2144	0.1403	0.1452	0.1340	0.1326	<u>0.1270</u>	0.1324

As shown in Table 4, DuoFreq achieves the best average MSE and MAE on all five workload datasets. These results demonstrate that DuoFreq can effectively generalize the learned workload patterns to unseen instances across different forecasting horizons. Complete results for individual forecasting horizons are reported in Appendix B.2.

4.5 RQ4: Computational Efficiency and Scalability

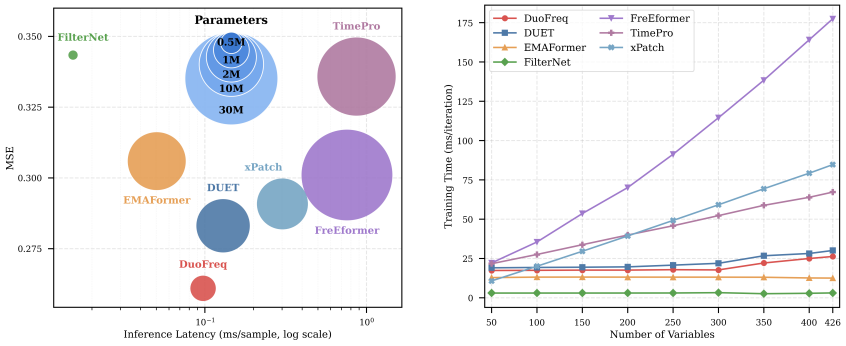
In this section, we evaluate the computational efficiency of DuoFreq and compare it with competing forecasting models. We report the forecasting accuracy, parameter size, training time, inference latency, and throughput on FaaS and Huawei CPU. The efficiency evaluation results are summarized in Table 5. The detailed efficiency measurement protocol is provided in Appendix A.5.

Table 5: Forecasting accuracy and computational efficiency on FaaS and Huawei CPU at $P = 288$. Lower is better for MSE, parameter count, training time, and inference latency; higher is better for throughput. Inference latency and throughput are measured with batch size 32.

	Models	DuoFreq	EMAFormer	DUET	FilterNet	FreEformer	xPatch	TimePro	DLinear	MSCNet
FaaS	MSE	0.2610	0.3059	0.2831	0.3434	0.3011	0.2908	0.3358	0.4323	0.3079
	Params (M)	0.6065	4.8828	2.9215	0.2598	27.6652	2.2215	17.0952	0.4153	16.1223
	Train (ms/iter)	11.9627	8.0044	16.9789	2.2600	80.7710	44.4585	42.1789	1.7267	102.6437
	Infer (ms/sample)	0.0974	0.0503	0.1295	0.0153	0.7563	0.3016	0.8668	0.0164	0.8370
	Throughput	10267	19865	7719	65469	1322	3315	1154	61032	1195
Huawei CPU	MSE	0.8502	0.9375	0.9761	0.9160	0.8765	0.8740	0.8783	0.9292	0.8808
	Params (M)	0.6081	2.2102	2.9212	0.2595	27.3988	2.2212	17.0952	0.4153	16.1223
	Train (ms/iter)	17.3383	8.0762	14.2120	2.1470	30.1971	17.2762	27.5688	1.2956	37.0113
	Infer (ms/sample)	0.0757	0.0476	0.1115	0.0138	0.2582	0.1142	0.3585	0.0088	0.2725
	Throughput	13204	21026	8970	72574	3873	8758	2789	113959	3670

As shown in Table 5, DuoFreq achieves the lowest forecasting error on both datasets with only about 0.61M parameters. Compared with DUET, the strongest baseline on FaaS, DuoFreq reduces MSE by 7.8%, training time by 29.5%, inference latency by 24.8%, and parameter size by 79.2%, while improving throughput by 33.0%. On Huawei CPU, compared with the strongest baseline xPatch, DuoFreq reduces MSE by 2.7%, inference latency by 33.7%, and parameter size by 72.6%, while increasing throughput by 50.8%, with comparable training time.

Figure 5a further illustrates the favorable balance between forecasting accuracy, inference latency, and model size achieved by DuoFreq. Figure 5b evaluates the training cost as the number of workload variables increases on PaaS. DuoFreq maintains practical training costs as the workload dimension grows, demonstrating its scalability to high-dimensional cloud workloads.



(a) Accuracy–efficiency trade-off on FaaS. (b) Training-time scaling on PaaS.

Figure 5: Computational efficiency and scalability analysis of DuoFreq and competing methods.

5 Conclusion

In this work, we address the heterogeneous spectral structures of cloud workloads, where low- and high-frequency components exhibit different characteristics and are not naturally suited to homogeneous modeling. We propose DuoFreq, a lightweight frequency-domain forecasting framework based on asymmetric dual-band spectral modeling. DuoFreq preserves and adapts low-frequency structures through SLFM, while LG-HFM and CST model fine-grained and band-wide high-frequency dynamics. Extensive experiments on five real-world cloud workload datasets demonstrate that DuoFreq achieves strong forecasting accuracy across multiple horizons, generalizes well to unseen workload instances, and maintains a favorable accuracy–efficiency trade-off. These results demonstrate the effectiveness of explicitly modeling heterogeneous frequency regions for practical cloud workload forecasting.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. doi: 10.48550/arXiv.1607.06450.

Felor Beikzadeh Abbasi, Ali Rezaee, Sahar Adabi, and Ali Movaghar. Fault-tolerant scheduling of graph-based loads on fog/cloud environments with multi-level queues and LSTM-based workload prediction. *Computer Networks*, 235:109964, 2023. doi: 10.1016/j.comnet.2023.109964.

ByteDance. CloudTimeSeriesData. Hugging Face Dataset Repository, 2025. URL https://huggingface.co/datasets/ByteDance/CloudTimeSeriesData. Accessed: 2026-09-26.

Rodrigo N. Calheiros, Enayat Masoumi, Rajiv Ranjan, and Rajkumar Buyya. Workload prediction using ARIMA model and its impact on cloud applications' QoS. *IEEE Transactions on Cloud Computing*, 3(4):449–458, 2015. doi: 10.1109/TCC.2014.2350475.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854.

Lei Chen, Weiwen Zhang, and Haiming Ye. Accurate workload prediction for edge data centers: Savitzky–golay filter, CNN and BiLSTM with attention mechanism. *Applied Intelligence*, 52(11): 13027–13042, 2022. doi: 10.1007/s10489-021-03110-x.

Zheyi Chen, Jia Hu, Geyong Min, Albert Y. Zomaya, and Tarek El-Ghazawi. Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning. *IEEE Transactions on Parallel and Distributed Systems*, 31(4):923–934, 2020. doi: 10.1109/TPDS.2019.2953745.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *International Conference on Learning Representations*, 2024.

Ehsan Golshani and Mehrdad Ashtiani. Proactive auto-scaling for cloud environments using temporal convolutional neural networks. *Journal of Parallel and Distributed Computing*, 154:119–141, 2021. doi: 10.1016/j.jpdc.2021.04.006.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. doi: 10.48550/arXiv.1606.08415.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *International Conference on Learning Representations*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yao Lu, Yongjing Shang, Jie Cui, Hong Zhong, Lu Liu, and Geyong Min. FP-MLP: A frequency domain patch-based MLP model for GPU-dominated cloud workload prediction. *IEEE Transactions on Cloud Computing*, 13(4):1235–1246, 2025. doi: 10.1109/TCC.2025.3620437.

Yuxiao Luo, Songming Zhang, Ziyu Lyu, and Yuhan Hu. TFDNet: Time–frequency enhanced decomposed network for long-term time series forecasting. *Pattern Recognition*, 162:111412, 2025. doi: 10.1016/j.patcog.2025.111412.

Aitian Ma, Dongsheng Luo, and Mo Sha. MixLinear: Extreme low resource multivariate time series forecasting with 0.1K parameters. In *International Conference on Learning Representations*, 2026.

Xiaowen Ma, Zhen-Liang Ni, Shuai Xiao, and Xinghao Chen. TimePro: Efficient multivariate long-term time series forecasting with variable- and time-aware hyper-state. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42096–42111. PMLR, 2025.

Aya I. Maiyza, Noha O. Korany, Karim Banawan, Hanan A. Hassan, and Walaa M. Sheta. VT-GAN: Hybrid generative adversarial networks for cloud workload prediction. *Journal of Cloud Computing*, 12, 2023. doi: 10.1186/s13677-023-00473-z. Article 97.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8024–8035, 2019.

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

Artyom Stitsyuk and Jaesik Choi. xPatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025. doi: 10.1609/aaai.v39i19.34270.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025. doi: 10.1016/j.neucom.2024.129178.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Minxian Xu, Chenghao Song, Huaming Wu, Sukhpal Singh Gill, Kejiang Ye, and Chengzhong Xu. esDNN: Deep neural network based multivariate workload prediction in cloud computing environments. *ACM Transactions on Internet Technology*, 22(3):75:1–75:24, 2022. doi: 10.1145/3524114.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *International Conference on Learning Representations*, 2024.

Hengyu Ye, Jiadong Chen, Fuxin Jiang, Xiao He, Tieying Zhang, Jianjun Chen, and Xiaofeng Gao. Fremer: Lightweight and effective frequency transformer for workload forecasting in cloud services. *Proceedings of the VLDB Endowment*, 18(11):3812–3825, 2025. doi: 10.14778/3749646.3749656.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, An Ning, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76656–76679, 2023.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. FilterNet: Harnessing frequency filters for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 55115–55140, 2024. doi: 10.52202/079017-1749.

Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. FreEformer: Frequency enhanced transformer for multivariate time series forecasting. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3606–3614, 2025. doi: 10.24963/ijcai.2025/401.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Zhiwei Zhang, Xinyi Du, Xuanchi Guo, Weihao Wang, and Wenjuan Han. EMAformer: Enhancing transformer through embedding armor for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28636–28644, 2026. doi: 10.1609/aaai.v40i34.40095.

Feiyu Zhao, Weiwei Lin, Shengsheng Lin, Shaomin Tang, and Keqin Li. MSCNet: Multi-scale network with convolutions for long-term cloud workload prediction. *IEEE Transactions on Services Computing*, 18(2):969–982, 2025. doi: 10.1109/TSC.2025.3536313.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12677–12690, 2022a. doi: 10.52202/068431-0921.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022b.

A Additional Experimental Details

This appendix provides additional details on dataset preprocessing, baseline methods, the evaluation and implementation protocol, the unseen-instance transfer setting, and computational efficiency measurement.

A.1 Dataset Preprocessing

We evaluate DuoFreq on five real-world cloud workload datasets: Huawei CPU, Huawei Memory, FaaS, IaaS, and PaaS. For Huawei CPU and Huawei Memory, we select the longest continuous monitoring interval and reconstruct the minute-level workload timeline. The resulting workload series are aggregated at a 5-minute granularity. Workload variables with more than 5% missing observations are removed, while the remaining missing values are filled by interpolation. After preprocessing, Huawei CPU and Huawei Memory contain 82 and 67 workload variables, respectively, with 9,504 time steps in each dataset.

FaaS, IaaS, and PaaS are obtained from the ByteDance CloudTimeSeriesData repository ByteDance (2025). The workload sequences used in our experiments have a 10-minute granularity. FaaS, IaaS, and PaaS contain 226, 93, and 426 workload variables, respectively, with 2,305, 3,456, and 7,776 time steps.

For all datasets, temporal order is preserved throughout preprocessing and evaluation. Each dataset is chronologically divided into training, validation, and test sets with a ratio of $70\% : 15\% : 15\%$. Standardization statistics are computed only from the training split and subsequently applied to the validation and test splits.

A.2 Baseline Details

We compare DuoFreq with eleven forecasting methods covering Transformer-based, state-space, linear, frequency-domain, multi-scale, and cloud-workload forecasting architectures. The compared methods are summarized below.

EMAFormer. EMAFormer Zhang et al. (2026) enhances Transformer-based forecasting through specialized embedding representations designed to improve the modeling of temporal characteristics in multivariate time series.

DUET. DUET Qiu et al. (2025) employs dual clustering over temporal and channel dimensions to model heterogeneous temporal patterns and inter-variable dependencies in multivariate time series.

FilterNet. FilterNet Yi et al. (2024) performs forecasting using learnable frequency filters that selectively preserve informative spectral components and suppress less useful frequency information.

FreEformer. FreEformer Yue et al. (2025) performs multivariate forecasting in the frequency domain and enhances Transformer-based dependency modeling over spectral representations.

xPatch. xPatch Stitsyuk & Choi (2025) adopts exponential seasonal–trend decomposition and a dual-stream architecture to combine linear and nonlinear temporal modeling.

S-Mamba. S-Mamba Wang et al. (2025) introduces state-space modeling into multivariate time series forecasting and uses Mamba-based representations to capture dependencies among time-series variables.

TimeKAN. TimeKAN Huang et al. (2025) combines frequency decomposition with KAN-based representation learning to model temporal patterns at different frequency scales.

TimePro. TimePro Ma et al. (2025) constructs variable- and time-aware representations to jointly model variable relationships and temporal information for multivariate forecasting.

DLinear. DLinear Zeng et al. (2023) decomposes a time series into trend and seasonal components and forecasts the decomposed components using lightweight linear mappings.

MSCNet. MSCNet Zhao et al. (2025) is designed for long-term cloud workload forecasting and employs multi-scale modeling to capture periodic patterns and long-range workload dependencies.

TFDNet. TFDNet Luo et al. (2025) combines time- and frequency-domain representations with decomposition-based modeling to capture long-term variations and periodic structures.

A.3 Evaluation and Implementation Protocol

For the main forecasting experiments, all methods are evaluated using the same preprocessed datasets, chronological data splits, historical input length, forecasting horizons, and evaluation metrics. The historical input length is fixed to $L = 720$, and the forecasting horizons are $P \in \{48, 96, 144, 288\}$. We use mean squared error (MSE) and mean absolute error (MAE) to evaluate forecasting accuracy. Lower values indicate better forecasting performance.

DuoFreq is implemented in PyTorch Paszke et al. (2019). All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory. We optimize DuoFreq using Adam Kingma & Ba (2015) with an initial learning rate of 1×10^{-4} and a batch size of 32. MSE is used as the forecasting loss together with the smoothness regularization introduced in SLFM. Training is performed for at most 150 epochs, and early stopping with a patience of 8 epochs is applied according to validation performance.

A.4 Unseen-Instance Transfer Protocol

We further evaluate whether forecasting models can transfer learned workload patterns to workload instances that are not observed during model training. For each dataset, workload variables retain their original order in the corresponding data file. Let C denote the total number of workload

variables and $H = \lfloor C/2 \rfloor$. We use the first H variables as the seen source set and the last H variables as the unseen target set. When C is odd, the single variable between the two subsets is excluded so that the source and target sets have equal cardinalities.

The temporal dimension continues to follow the $70\% : 15\% : 15\%$ chronological split. Model training and validation are conducted exclusively on the seen source variables, whereas final evaluation is performed exclusively on the unseen target variables. Consequently, target variables do not participate in gradient updates, validation-based early stopping, or checkpoint selection.

To account for scale differences between workload instances, the source and target subsets are standardized separately on a per-variable basis. For each source variable, the normalization statistics are estimated using only its temporal training portion. The target variables are normalized in the same manner using only their own first 70% temporal portion. The target-side statistics are used solely for preprocessing: no validation observations or future test labels are used to estimate them, and target variables remain excluded from model optimization. MSE and MAE are evaluated in the target variables' train-standardized space.

The same fixed source–target partition is retained across forecasting horizons. We evaluate $P \in \{48, 96, 144, 288\}$ separately and report their average in the main paper. The transfer experiments use the same MSE objective, Adam optimizer with a learning rate of 1×10^{-4}, maximum training length of 150 epochs, and early-stopping patience of 8. The transfer benchmark uses a micro-batch size of 4 with eight gradient-accumulation steps, corresponding to an effective batch size of approximately 32. Gradient clipping is applied with a maximum norm of 1, and mixed-precision training is not used.

Because several forecasting architectures contain parameters that depend on the number or positions of input variables, equal source and target cardinalities are maintained throughout this experiment. The resulting evaluation should therefore be interpreted as fixed-cardinality unseen-instance transfer under an original-order half split, rather than arbitrary-channel or channel-count-agnostic generalization.

A.5 Efficiency Measurement Protocol

We evaluate computational efficiency on FaaS and Huawei CPU using the same hardware environment as the main experiments. All efficiency measurements use an input length of $L = 720$ and a forecast horizon of $P = 288$. Existing best checkpoints are loaded for each model, and the reported forecasting error is obtained from the corresponding untouched checkpoint. All timing experiments are conducted in FP32 without automatic mixed precision or TF32.

The parameter count is computed directly from the instantiated model corresponding to each checkpoint configuration. We report the total number of model parameters in millions.

For training efficiency, we measure the computation time of one optimizer update rather than the total time to convergence. One training iteration consists of gradient initialization, forward propagation, computation of the model's native training objective, backpropagation, gradient clipping when configured, and an Adam optimizer update. Data loading and host-to-device transfer are excluded from the timing measurement so that the reported value reflects model computation. All models are evaluated with an effective training batch size of 32.

Before timing, 10 warm-up optimizer updates are performed. Each measurement then contains 50 consecutive optimizer updates. To reduce measurement variance, the training benchmark is independently repeated five times, with the model and Adam optimizer reinitialized from the same checkpoint before each trial. The value reported as training time is the mean wall-clock time per optimizer update over the five trials. We additionally check measurement stability using the relative range between the maximum and minimum trial means and accept the average when this range does not exceed 5% of the five-trial mean.

For inference efficiency, a real test batch is first transferred to the GPU so that data loading and host-to-device transfer are excluded from the timing measurement. We perform 30 warm-up forward passes followed by 100 measured forward passes. CUDA events are used to measure GPU execution time after device synchronization. Inference latency in the main paper is measured with a batch size of 32 and is reported as the mean batch execution time divided by the batch size. Throughput is

Table 6: Full forecasting results over four prediction horizons $P \in \{48, 96, 144, 288\}$ on five workload datasets. Lower values are better. Rankings are determined by the displayed values rounded to four decimal places; the best result is shown in bold and the second-best result is underlined.

Data	P	DuoFreq		EMAFormer		DUET		FilterNet		FreEformer		xPatch	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Huawei CPU	48	**0.4549**	**0.3468**	0.5409	0.3760	0.5357	0.4017	0.5005	0.3657	0.4949	0.3711	0.4897	0.3641
	96	**0.5533**	**0.3789**	0.6858	0.4252	0.6537	0.4313	0.6580	0.4064	0.6217	0.4125	0.6047	0.4021
	144	**0.6491**	**0.4066**	0.8048	0.4492	0.7350	0.4558	0.7592	0.4298	0.7198	0.4353	0.6876	0.4220
	288	**0.8502**	**0.4447**	0.9375	0.4703	0.9760	0.5021	0.9160	0.4681	0.8765	0.4655	0.8740	0.4602
Huawei Memory	48	**0.7067**	**0.4274**	0.7407	0.4561	0.7466	0.4670	0.7915	0.4630	0.7077	0.4425	0.7068	0.4362
	96	0.9711	**0.5160**	**0.9633**	0.5277	1.0369	0.5568	1.0717	0.5550	0.9870	0.5337	0.9858	0.5336
	144	1.1663	0.5739	1.1796	0.5802	1.1668	0.5948	1.2774	0.6189	**1.1485**	0.5837	1.1865	0.5921
	288	1.5786	0.6751	1.6750	0.7012	1.5976	0.7032	1.6798	0.7177	**1.5335**	0.6845	1.6296	0.6974
FaaS	48	**0.1759**	**0.2617**	0.2126	0.2937	0.1965	0.3033	0.2317	0.3045	0.2024	0.2891	0.1925	0.2791
	96	**0.2023**	**0.2852**	0.2541	0.3232	0.2238	0.3245	0.2745	0.3382	0.2425	0.3174	0.2177	0.2979
	144	**0.2088**	**0.2889**	0.2573	0.3274	0.2324	0.3289	0.2838	0.3443	0.2559	0.3282	0.2320	0.3086
	288	**0.2610**	**0.3282**	0.3059	0.3628	0.2831	0.3602	0.3434	0.3854	0.3010	0.3602	0.2908	0.3489
IaaS	48	**0.6917**	**0.5495**	0.6963	0.5526	0.7084	0.5621	0.7113	0.5666	0.7122	0.5628	0.7231	0.5712
	96	**0.6866**	**0.5482**	0.6989	0.5593	0.7077	0.5639	0.7268	0.5783	0.7128	0.5672	0.7223	0.5763
	144	**0.6931**	**0.5514**	0.7126	0.5704	0.7173	0.5704	0.7367	0.5852	0.7197	0.5725	0.7278	0.5805
	288	**0.6986**	**0.5541**	0.7350	0.5832	0.7242	0.5755	0.7421	0.5916	0.7249	0.5770	0.7348	0.5843
PaaS	48	**0.0447**	**0.1110**	0.0518	0.1248	0.0860	0.2072	0.0517	0.1214	0.0493	0.1157	0.0503	0.1192
	96	**0.0544**	**0.1269**	0.0626	0.1423	0.0959	0.2234	0.0613	0.1342	0.0609	0.1418	0.0594	0.1339
	144	**0.0608**	**0.1387**	0.0689	0.1515	0.1054	0.2365	0.0667	0.1439	0.0670	0.1511	0.0657	0.1442
	288	**0.0757**	**0.1624**	0.0814	0.1718	0.1144	0.2452	0.0794	0.1648	0.0790	0.1699	0.0794	0.1660

Data	P	S-Mamba		TimeKAN		TimePro		DLinear		MSCNet		TFDNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Huawei CPU	48	0.5270	0.3743	0.5173	0.3723	0.4845	0.3630	0.5462	0.3808	0.4921	0.3608	0.4994	0.3697
	96	0.6612	0.4169	0.6365	0.4112	0.5942	0.3988	0.6611	0.4135	0.6078	0.3962	0.6007	0.4000
	144	0.7706	0.4376	0.7205	0.4316	0.6904	0.4225	0.7490	0.4329	0.6886	0.4096	0.6890	0.4193
	288	0.9288	0.4708	0.8803	0.4599	0.8783	0.4619	0.9292	0.4665	0.8808	0.4499	0.8676	0.4521
Huawei Memory	48	0.7826	0.4635	0.7718	0.4630	0.7465	0.4400	0.7834	0.4718	0.7482	0.4465	0.7093	0.4338
	96	1.0817	0.5608	1.0476	0.5544	1.0201	0.5337	1.0871	0.5661	1.0377	0.5395	0.9820	0.5208
	144	1.3006	0.6215	1.2419	0.6133	1.2144	0.5887	1.2888	0.6210	1.2149	0.5913	1.1688	**0.5700**
	288	1.6596	0.7231	1.6675	0.7152	1.5786	0.6850	1.6812	0.7141	1.5985	0.6855	1.5754	**0.6727**
FaaS	48	0.2119	0.2963	0.2013	0.2821	0.2287	0.2875	0.2006	0.2919	0.2015	0.2796	0.1797	0.2698
	96	0.2682	0.3318	0.2396	0.3154	0.2516	0.3134	0.2580	0.3424	0.2357	0.3053	0.2065	0.2921
	144	0.2605	0.3324	0.2504	0.3239	0.2649	0.3230	0.2935	0.3686	0.2467	0.3146	0.2168	0.2998
	288	0.3380	0.3850	0.3135	0.3648	0.3358	0.3696	0.4323	0.4610	0.3079	0.3624	0.2781	0.3421
IaaS	48	0.7257	0.5707	0.7156	0.5580	0.7174	0.5649	0.7343	0.5743	0.7164	0.5628	0.7137	0.5615
	96	0.7318	0.5779	0.7243	0.5672	0.7158	0.5691	0.7373	0.5805	0.7184	0.5690	0.7065	0.5622
	144	0.7436	0.5854	0.7369	0.5756	0.7249	0.5736	0.7518	0.5889	0.7262	0.5720	0.7129	0.5655
	288	0.7529	0.5889	0.7610	0.5868	0.7313	0.5779	0.7845	0.6075	0.7268	0.5714	0.7223	0.5718
PaaS	48	0.0552	0.1276	0.0531	0.1332	0.0504	0.1160	0.0510	0.1267	0.0508	0.1227	0.0493	0.1252
	96	0.0625	0.1392	0.0614	0.1470	0.0568	0.1278	0.0618	0.1425	0.0584	0.1342	0.0591	0.1428
	144	0.0699	0.1486	0.0654	0.1497	0.0634	0.1406	0.0690	0.1517	0.0635	0.1434	0.0647	0.1512
	288	0.0814	0.1688	0.0813	0.1757	0.0781	0.1646	0.0911	0.1801	0.0772	0.1651	0.0804	0.1743

computed as

$$\text{Throughput} = \frac{32 \times 1000}{t_{\text{batch}}},$$

where t_{batch} denotes the mean inference time of a batch of 32 samples in milliseconds.

B Additional Forecasting Results

B.1 Full Forecasting Results

The main paper reports forecasting results averaged over the four prediction horizons. Here, we provide the complete MSE and MAE results for every dataset–horizon setting at $P \in \{48, 96, 144, 288\}$. Across the 20 settings, DuoFreq achieves the best MSE in 17 settings and the best MAE in 18 settings. In particular, DuoFreq achieves the best MSE across all four forecasting horizons on Huawei CPU, FaaS, IaaS, and PaaS, demonstrating consistent performance across different prediction lengths.

B.2 Full Unseen-Instance Transfer Results

The main paper reports unseen-instance transfer results averaged over the four prediction horizons. Here, we provide the complete results for each prediction horizon under the fixed-cardinality unseen-instance transfer protocol described in Appendix A.4. For each dataset, the models are trained and validated exclusively on the seen workload instances and evaluated on the disjoint unseen instances. We report MSE and MAE for $P \in \{48, 96, 144, 288\}$.

Across the 20 dataset–horizon settings, DuoFreq achieves the best MSE in 16 settings and the best MAE in 19 settings. It obtains the best MSE and MAE at all four horizons on FaaS, IaaS, and PaaS. On Huawei CPU, DuoFreq ranks first for both metrics at $P \in \{48, 96, 144\}$ and second at $P = 288$. On Huawei Memory, DuoFreq achieves the best MAE at all four horizons and remains among the top two methods in MSE for every horizon.

Table 7: Full results of intra-dataset unseen-instance transfer forecasting over four prediction horizons $P \in \{48, 96, 144, 288\}$. Lower values are better. Rankings are determined by the displayed values rounded to four decimal places; the best result is shown in bold and the second-best result is underlined.

Data	P	Metric	DuoFreq	EMAFormer	DUET	FilterNet	FreEformer	xPatch	TimePro	DLinear	MSCNet
Huawei CPU	48	MSE	**0.4235**	0.4351	0.4961	0.4605	0.4415	0.4452	0.4346	0.4407	0.4341
		MAE	**0.3511**	0.3633	0.4019	0.3799	0.3683	0.3747	0.3694	0.3595	0.3641
	96	MSE	**0.4671**	0.5004	0.5615	0.5303	0.5069	0.5047	0.4881	0.4905	0.4899
		MAE	**0.3759**	0.3898	0.4332	0.4038	0.4022	0.4020	0.3958	0.3825	0.3902
	144	MSE	**0.5143**	0.5541	0.6088	0.5823	0.5451	0.5515	0.5288	0.5264	0.5261
		MAE	**0.3899**	0.4117	0.4511	0.4266	0.4103	0.4231	0.4149	0.3940	0.3921
	288	MSE	0.5807	0.6382	0.6641	0.6742	0.6269	0.6283	0.5901	**0.5792**	0.5937
		MAE	0.4100	0.4389	0.4664	0.4626	0.4384	0.4472	0.4213	**0.4088**	0.4156
Huawei Memory	48	MSE	0.4910	0.5085	0.5273	0.5382	0.4974	**0.4840**	0.5032	0.4986	0.5074
		MAE	**0.4050**	0.4262	0.4333	0.4476	0.4172	0.4105	0.4120	0.4194	0.4149
	96	MSE	0.6469	0.6713	0.6600	0.6882	0.6685	**0.6463**	0.6767	0.6681	0.6798
		MAE	**0.4689**	0.4931	0.4978	0.5098	0.4910	0.4811	0.4888	0.4908	0.4906
	144	MSE	0.7438	0.7664	0.7584	0.8084	0.7801	**0.7406**	0.7648	0.7732	0.7708
		MAE	**0.5108**	0.5317	0.5422	0.5569	0.5391	0.5205	0.5254	0.5343	0.5349
	288	MSE	**0.9457**	0.9740	0.9800	1.0420	0.9808	0.9687	0.9769	1.0093	0.9662
		MAE	**0.5812**	0.6007	0.6136	0.6449	0.6110	0.5949	0.5959	0.6115	0.5892
FaaS	48	MSE	**0.1627**	0.1946	0.1881	0.2106	0.1910	0.1786	0.2292	0.1991	0.1948
		MAE	**0.2498**	0.2815	0.2934	0.2857	0.2797	0.2645	0.2905	0.2891	0.2772
	96	MSE	**0.1809**	0.2211	0.2044	0.2357	0.2170	0.2042	0.2315	0.2766	0.2203
		MAE	**0.2699**	0.3061	0.3114	0.3165	0.3030	0.2897	0.3002	0.3507	0.2973
	144	MSE	**0.1879**	0.2339	0.2165	0.2481	0.2446	0.2164	0.2472	0.3361	0.2253
		MAE	**0.2732**	0.3136	0.3169	0.3245	0.3256	0.2978	0.3149	0.3881	0.3047
	288	MSE	**0.2555**	0.3140	0.2748	0.3319	0.3048	0.2934	0.4403	0.4838	0.3114
		MAE	**0.3225**	0.3651	0.3556	0.3776	0.3705	0.3487	0.4042	0.4726	0.3637
IaaS	48	MSE	**0.7292**	0.7567	0.8161	0.7865	0.7575	0.8079	0.7682	0.7416	0.7504
		MAE	**0.6110**	0.6312	0.6656	0.6511	0.6302	0.6558	0.6335	0.6180	0.6259
	96	MSE	**0.7120**	0.7681	0.7990	0.8425	0.7512	0.8357	0.7541	0.7325	0.7344
		MAE	**0.6082**	0.6399	0.6620	0.6883	0.6346	0.6739	0.6333	0.6219	0.6249
	144	MSE	**0.7146**	0.7952	0.8276	0.8316	0.7871	0.8946	0.7601	0.7471	0.7440
		MAE	**0.6095**	0.6528	0.6727	0.6777	0.6497	0.6968	0.6373	0.6312	0.6302
	288	MSE	**0.7234**	0.7990	0.8859	0.9706	0.8648	1.0198	0.7853	0.7789	0.7663
		MAE	**0.6138**	0.6492	0.6964	0.7230	0.6855	0.7365	0.6497	0.6487	0.6384
PaaS	48	MSE	**0.0321**	0.0510	0.0757	0.0441	0.0370	0.0401	0.0368	0.0328	0.0346
		MAE	**0.1002**	0.1389	0.1947	0.1219	0.1142	0.1148	0.1075	0.1075	0.1074
	96	MSE	**0.0414**	0.0606	0.0840	0.0537	0.0542	0.0506	0.0473	0.0432	0.0493
		MAE	**0.1143**	0.1519	0.2079	0.1344	0.1425	0.1295	0.1234	0.1213	0.1289
	144	MSE	**0.0483**	0.0608	0.0938	0.0591	0.0616	0.0551	0.0580	0.0492	0.0551
		MAE	**0.1245**	0.1482	0.2220	0.1417	0.1534	0.1351	0.1435	0.1282	0.1393
	288	MSE	**0.0612**	0.0689	0.1011	0.0720	0.0727	0.0659	0.0646	0.0620	0.0642
		MAE	**0.1475**	0.1639	0.2329	0.1630	0.1706	0.1566	0.1559	0.1510	0.1539

The detailed results show that DuoFreq maintains strong unseen-instance transfer performance across different prediction lengths. DuoFreq ranks first in MSE for 16 of the 20 dataset–horizon settings and first in MAE for 19 settings. The exceptions in MSE occur on Huawei CPU at $P = 288$ and Huawei Memory at $P \in \{48, 96, 144\}$, where DuoFreq remains the second-best method. For MAE, the only setting in which DuoFreq does not rank first is Huawei CPU at $P = 288$, where it ranks second. These results are consistent with the horizon-averaged transfer results reported in

the main paper and indicate that the transfer advantage of DuoFreq is not driven by a particular forecasting horizon.