PERICAL: ADAPTIVELY CALIBRATING PERIODIC PATTERNS FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Periodic patterns provide important temporal information in multivariate time-series forecasting by capturing regular variations in real-world systems. However, periodic patterns do not recur identically across contexts: their timing and magnitude may vary with evolving system states and external conditions. Existing methods typically model periodic patterns with fixed representations, which may become inconsistent with current observations when such variations occur. To address this limitation, we propose PeriCal, a lightweight forecasting framework that adapts periodic patterns through Adaptive Periodic Calibration. PeriCal learns input-conditioned calibration factors to adjust periodic patterns along both temporal and value dimensions, enabling the periodic representation to better match the current temporal context before being integrated with local observations through two-dimensional convolution. Experiments on 11 real-world benchmarks demonstrate that PeriCal achieves strong forecasting performance across datasets from multiple application domains. Further analysis verifies the effectiveness of the proposed adaptive periodic modeling approach.

1 INTRODUCTION

Long-term multivariate time-series forecasting (MTSF) is important in financial markets Sezer et al. (2020), weather prediction Karevan & Suykens (2020), and traffic management Miao et al. (2024), where recurring patterns provide valuable predictive information. Existing approaches capture such patterns through seasonal decomposition Wu et al. (2021); Zhou et al. (2022b); Yu et al. (2024), frequency analysis Cao et al. (2023); Fei et al. (2025), period-aware architectures Wu et al. (2023); Cai et al. (2024); Dai et al. (2024a); Nematirad et al. (2025); Ma et al. (2025), or learnable periodic priors Fan et al. (2022); Ni et al. (2023); Lin et al. (2024a); Cao et al. (2026). Although these methods have advanced periodic modeling, they primarily identify or represent recurrence and often provide limited adaptation of the resulting representation to the current temporal context. In particular, learning a reusable pattern does not determine how that pattern should be expressed when the next cycle differs from previous ones.

Recurring patterns need not appear identically in every context. Traffic demand, for example, retains daily structure even as peak times and intensities change; weather and electricity consumption exhibit analogous variations. Within-cycle regularity and cycle-to-cycle change are not contradictory: the former makes a periodic template reusable, whereas the latter can make an unmodified template unreliable. Identifying a dominant cycle length alone does not resolve this discrepancy, because peak positions and values may change even when the nominal period remains stable. The same recurring structure can therefore shift locally in time and change in value without ceasing to be periodic (Figure 1(a)). Reusing a fixed representation can then place events at incorrect times or misestimate their magnitude. Temporal and value mismatches require distinct adjustments, and a single global shift cannot capture nonuniform timing changes; related distinctions between temporal and shape discrepancies also arise in distortion-aware forecasting Le Guen & Thome (2019). A useful periodic prior should therefore adapt to current observations while remaining complementary to their transient fluctuations and short-term dynamics (Figure 1(b)).

To address these challenges, we propose **PeriCal**, a lightweight forecasting framework built on Adaptive Periodic Calibration. PeriCal adaptively adjusts periodic patterns according to current observations along both temporal and value dimensions. Specifically, it learns input-conditioned

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.

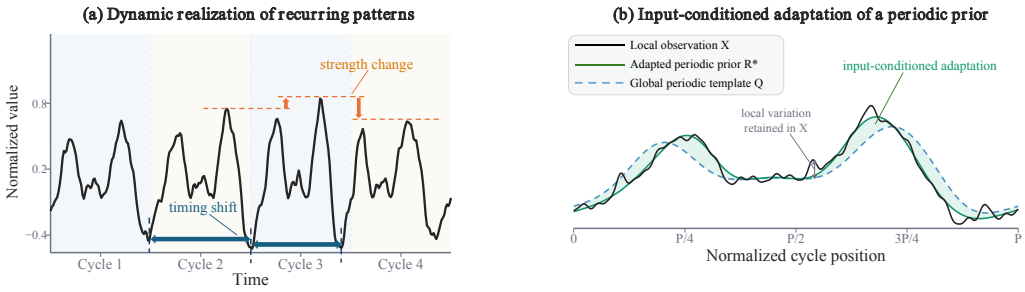


Figure 1: Adaptive periodic calibration in PeriCal. (a) Periodic patterns may preserve similar overall structures while their temporal positions and values vary under different contexts. (b) PeriCal calibrates periodic patterns along temporal and value dimensions using current observations before integrating them with local dynamics.

factors to calibrate the temporal positions and values of periodic information. The calibrated periodic component is then integrated with local observations through two-dimensional convolution, enabling the forecasting model to combine periodic information with transient and fine-grained dynamics. By focusing on how periodic patterns should be adapted rather than simply represented, PeriCal addresses the mismatch between fixed periodic representations and context-dependent temporal variations. Calibration is applied to the shared periodic prior, rather than using that prior to reconstruct every detail of the input. This distinction retains a separate path for local, irregular events that a recurring template need not explain. The primary contributions are:

- We propose PeriCal, a lightweight forecasting framework that adaptively calibrates periodic patterns according to the current temporal context, enabling periodic information to better match changing observations.
- We design an Adaptive Periodic Calibration module that performs input-conditioned temporal and value calibration over periodic patterns, allowing periodic information to adapt to the current temporal context.
- We evaluate PeriCal on 11 real-world MTSF datasets, demonstrating competitive accuracy across diverse prediction horizons, while ablation and efficiency analyses assess the benefits and computational cost of adaptive calibration.

2 Related Work

2.1 DL-based Time Series Forecasting

Deep learning methods for multivariate time-series forecasting aim to capture temporal dependencies, inter-variable relationships, and evolving temporal patterns. Transformer-based approaches model long-range dependencies through sparse attention Zhou et al. (2021), patching Nie et al. (2023), learnable bases Ni et al. (2023), or adaptive attention Luo & Wang (2024). Meanwhile, iTransformer Liu et al. (2024), MSGNet Cai et al. (2024), and TQNet Lin et al. (2025) emphasize variable interactions, while DLinear Zeng et al. (2023) demonstrates the competitiveness of lightweight architectures. Recent methods further address changing temporal conditions through adaptive normalization Kim et al. (2022); Liu et al. (2023b); Ye et al. (2024); Dai et al. (2024b), non-stationary dependency learning Liu et al. (2022), or adaptive representations Liu et al. (2023a); Hu et al. (2024); Liu et al. (2025); Lu et al. (2026); Ding et al. (2026). These methods improve general forecasting backbones by adapting data statistics, latent dynamics, or dependency structures, and have shown that robust temporal representations are essential under distribution shifts. However, these methods mainly focus on adapting general temporal representations or dependency structures, rather than explicitly modeling how periodic components should be adjusted according to current observations. PeriCal instead focuses on the context-dependent realization of recurring components while retaining local observations as complementary information.

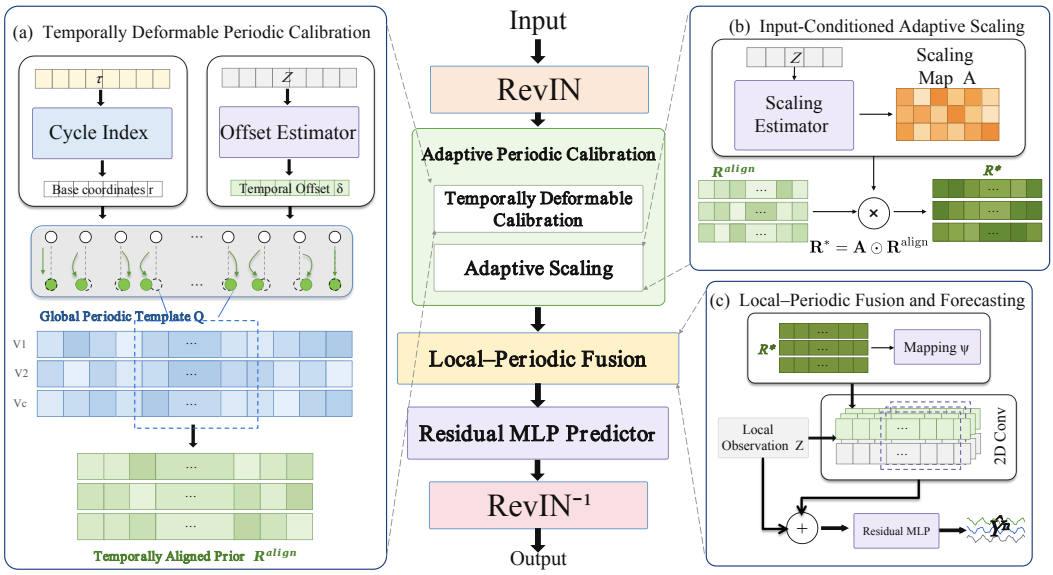


Figure 2: Overall architecture of PeriCal. Historical observations are optionally normalized by RevIN and processed by Adaptive Periodic Calibration. (a) Temporal calibration predicts smooth offsets from the normalized input and performs cyclic interpolation over the learnable global template $\mathbf{Q}$. (b) Value calibration adjusts the temporally aligned prior $\mathbf{R}^{\text{temp}}$ to obtain the adaptive periodic prior $\mathbf{R}^*$. (c) The adapted prior is mapped to periodic cue $\mathbf{G}$, stacked with local observations, fused by two-dimensional convolution, and passed to a residual MLP for forecasting.

2.2 Time Series Forecasting Decomposition

Decomposition methods separate observed series into components with different temporal characteristics, typically trend, seasonality, and residual variation. Classical approaches employ statistical decomposition, while neural models integrate component separation into end-to-end forecasting. N-BEATS Oreshkin et al. (2020) learns interpretable trend and seasonality bases; Autoformer Wu et al. (2021), FEDformer Zhou et al. (2022b), and ETSformer Woo et al. (2022) explicitly model trend–seasonal components. DLinear Zeng et al. (2023) combines moving-average decomposition with linear prediction, Leddam Yu et al. (2024) models inter-series dependencies and intra-series variations, and TimeMixer Wang et al. (2024b) mixes decomposed components across temporal resolutions. These studies establish component-wise modeling as an effective forecasting paradigm. However, the extracted seasonal component is commonly modeled with fixed representations, which may become inconsistent with current observations when temporal or value variations occur. PeriCal follows this component-wise perspective but further adapts the temporal alignment and strength of the periodic component according to the current observations.

2.3 Periodicity Modeling in Time Series Forecasting

Beyond general decomposition, periodicity has been modeled through frequency analysis, period-aware architectures, and explicit periodic representations. Frequency-domain methods exploit spectral projections, frequency-domain MLPs or graphs, and learnable filters Zhou et al. (2022a); Yi et al. (2023b;a; 2024); Fei et al. (2025). Period-aware architectures reorganize sequences to capture intra- and inter-period dependencies: TimesNet Wu et al. (2023) constructs two-dimensional temporal variations, PDF Dai et al. (2024a) decouples multi-period patterns, SparseTSF Lin et al. (2024b) performs cross-period sparse forecasting, and MoFo Ma et al. (2025) forms period-structured patches. Explicit approaches learn periodic states, bases, cycles, or templates: DEPTS Fan et al. (2022), BasisFormer Ni et al. (2023), CycleNet Lin et al. (2024a), and timestamp-guided methods Wang et al. (2024a); Cao et al. (2026) represent recurring structures as reusable priors, while Implicit Forecaster Li et al. (2025) synthesizes future patterns through spectral parameterization. These approaches differ in how periodicity is discovered or represented, but they largely focus on identifying the recur-

ring structure itself. Once learned, the resulting basis, cycle, or template is generally reused without adapting its temporal alignment and strength to the current input. This can produce systematic mismatch when recurring events occur earlier, later, or with different intensity. PeriCal addresses this limitation through input-conditioned temporal calibration and variable-specific value calibration.

3 METHODOLOGY

3.1 OVERVIEW OF PERICAL

Given a historical sequence $\mathbf{X} \in \mathbb{R}^{L\times C}$ of C variables and look-back length L, we predict the next H observations $\mathbf{Y} \in \mathbb{R}^{H\times C}$ with $\widehat{\mathbf{Y}} = \mathcal{F}(\mathbf{X}, \tau)$, where τ is the cycle index supplied with the input window. As shown in Figure 2, PeriCal retrieves recurring structure from a learnable global template, calibrates its temporal coordinates and values conditioned on the input, and fuses the resulting prior with local observations via two-dimensional convolution. A residual MLP then produces the forecast, and all components are trained jointly.

The global template captures shared recurrence, while the calibration branches specify its realization for the current sequence. Their bounded adjustments modify the timing and strength of the prior without requiring it to reconstruct transient or irregular events, which remain available through the local observation path. Thus, the periodic prior complements rather than replaces the input.

Following reversible instance normalization Kim et al. (2022), we optionally normalize each variable along the temporal dimension:

$$\widetilde{X}_{t,c} = \frac{X_{t,c} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \quad \mu_c = \frac{1}{L}\sum_{t=1}^{L} X_{t,c}, \tag{1}$$

where σ_c^2 is the corresponding temporal variance. We use $\mathbf{Z} = \widetilde{\mathbf{X}}^\top \in \mathbb{R}^{C\times L}$ below and omit the batch dimension for clarity.

3.2 ADAPTIVE PERIODIC CALIBRATION

Time-indexed template initialization. We maintain a learnable global periodic template $\mathbf{Q} \in \mathbb{R}^{P\times C}$, where the cycle length P is prespecified from the sampling interval and an expected daily or weekly cycle (Appendix Table 4). For an input window beginning at zero-based index n_0 in the original, unsplit series, the data loader assigns $\tau = (n_0 + L) \bmod P$, the cycle index of the first forecast step. This preserves a common time origin across the training, validation, and test splits. The base retrieval coordinate at slot t is then:

$$r_t = (\tau + t) \bmod P, \qquad t = 0, \ldots, L-1. \tag{2}$$

The cycle index is deterministic rather than learned from the input. Directly using $\mathbf{Q}_{r_t,:}$ assumes that every cycle is aligned with the same discrete coordinates Cao et al. (2026). PeriCal relaxes this assumption by learning input-conditioned continuous adjustments to $\{r_t\}_{t=0}^{L-1}$.

Input-conditioned temporal calibration. The offset estimator summarizes the multivariate context at each time step using the cross-variable mean and standard deviation:

$$m_t = \frac{1}{C}\sum_{c=1}^{C} Z_{c,t}, \qquad s_t = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(Z_{c,t} - m_t)^2}. \tag{3}$$

The resulting context $\mathbf{U} = [\mathbf{m}; \mathbf{s}] \in \mathbb{R}^{2\times L}$ is processed by a lightweight two-layer one-dimensional convolutional network f_ϕ with a GELU activation. We bound the raw offset with

$$\boldsymbol{\delta}^{\,\mathrm{raw}} = \Delta_{\max} \tanh(f_\phi(\mathbf{U})), \tag{4}$$

where $\Delta_{\max}$ limits the maximum offset from the base coordinates. To suppress implausible pointwise jitter while retaining local temporal calibration, we smooth the offsets with a length-K moving average:

$$\delta_t = \frac{1}{K}\sum_{j=-\lfloor K/2\rfloor}^{\lfloor K/2\rfloor} \delta_{t+j}^{\mathrm{raw}}, \tag{5}$$

Here, K is an odd positive integer. For indices outside the observation window, we replicate the nearest boundary offset. The trajectory $\boldsymbol{\delta} \in \mathbb{R}^L$ is shared across variables because the variables share a common temporal coordinate, whereas allowing an independent coordinate system for every variable can disrupt cross-variable synchronization. Importantly, δ_t remains position dependent, allowing different portions of a cycle to advance or lag by different amounts.

The calibrated coordinate is $u_t = r_t + \delta_t$. Since u_t is continuous, we obtain the periodic prior through cyclic linear interpolation. Let

$$l_t = \lfloor u_t \rfloor \bmod P, \qquad \rho_t = u_t - \lfloor u_t \rfloor. \tag{6}$$

The temporally aligned periodic cue for variable c is:

$$R^{\text{align}}_{c,t} = (1 - \rho_t) Q_{l_t,c} + \rho_t Q_{(l_t+1) \bmod P,c}. \tag{7}$$

Interpolation preserves periodic boundary continuity and allows the forecasting loss to propagate gradients to both the template and the temporal calibration network. We zero-initialize the final convolution of f_ϕ, so PeriCal starts from the base coordinates with $\delta_t = 0$ and learns nonzero offsets only when they improve forecasting.

Input-conditioned value calibration. Temporal calibration determines *where* to sample from the periodic template, but the values of a recurring pattern can also vary across cycles. For each variable, PeriCal predicts a position-dependent calibration trajectory from its normalized observation:

$$\mathbf{a}_c = 1 + \tanh(\mathbf{W}_a \mathbf{z}_c + \mathbf{b}_a), \qquad \mathbf{a}_c \in (0, 2)^L, \tag{8}$$

where $\mathbf{z}_c \in \mathbb{R}^L$ is the c-th row of $\mathbf{Z}$. The parameters are shared across variables, while the resulting trajectory remains variable and input dependent. The final adaptive periodic prior is:

$$R^*_{c,t} = a_{c,t} \left[(1 - \rho_t) Q_{l_t,c} + \rho_t Q_{(l_t+1) \bmod P,c} \right]. \tag{9}$$

Equations 7–9 jointly calibrate the temporal positions and values of the shared periodic template without forcing the prior to reconstruct transient fluctuations in the local observations.

Design rationale and stable optimization. The two calibration operations deliberately use different conditioning structures. Temporal calibration is shared across variables because variables observed at the same timestamps follow a common temporal coordinate; independent coordinate adjustments could disrupt their synchronization. Nevertheless, the offset remains position dependent, so different portions of a cycle can shift by different amounts. In contrast, value calibration is variable specific because the strength of a recurring event need not change equally across channels. Its positive, bounded factors prevent sign reversal and limit amplification, while allowing the strength of the retrieved periodic cue to vary across variables and time positions. Together, these choices allow PeriCal to correct temporal and value mismatches without transforming the periodic branch into another copy of the local input.

We further adopt a conservative initialization. The global template is initialized without a predefined pattern, and the final layer of the offset estimator is zero initialized. Consequently, training begins from the original time-indexed coordinates and introduces nonzero offsets only through the forecasting objective. The template and both calibration networks then develop jointly, avoiding arbitrary coordinate adjustments before a meaningful recurring structure has been learned. The offset bound and smoothing window act as regularizers that control the admissible temporal calibration; they are treated as validation hyperparameters rather than assumptions about a universally optimal amount of variation.

3.3 Local-Periodic Fusion and Forecasting

The adaptive prior is first processed by a temporal linear map ψ, shared across variables, to obtain $\mathbf{G} = \psi(\mathbf{R}^*) \in \mathbb{R}^{C \times L}$. This map converts the calibrated template values into periodic features before they interact with the observations. We then stack the local and periodic representations as:

$$\mathbf{S} = \text{Stack}(\mathbf{Z}, \mathbf{G}) \in \mathbb{R}^{C \times 2 \times L}. \tag{10}$$

A two-dimensional convolution whose first kernel dimension spans both information sources produces the fused representation:

$$\mathbf{F} = \text{Conv2D}(\mathbf{S}) \in \mathbb{R}^{C \times L}. \tag{11}$$

Table 1: Multivariate forecasting results averaged over all prediction horizons. Each model reports MSE and MAE; lower values are better. P03–P08 denote the four PEMS datasets. The best results are in **bold**, and the second-best results are underlined. Values are rounded to three decimals, and ties after rounding share the same rank.

Dataset	**PeriCal (Ours)**		TimeMosaic (2026)		TimeBridge (2025)		TQNet (2025)		CycleNet (2024)		RAFT (2025)		SOFTS (2024)		S-Mamba (2025)		iTransformer (2024)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.438	0.432	**0.425**	**0.424**	0.439	0.434	0.441	0.434	0.457	0.441	0.428	0.433	0.449	0.443	0.455	0.450	0.454	0.448
ETTh2	0.373	0.400	**0.363**	**0.388**	0.375	0.398	0.378	0.402	0.388	0.409	0.382	0.410	0.373	0.400	0.381	0.405	0.383	0.407
ETTm1	**0.366**	0.390	0.381	**0.381**	0.384	0.391	0.377	0.393	0.379	0.396	0.381	0.400	0.393	0.402	0.398	0.405	0.407	0.410
ETTm2	0.271	**0.313**	0.273	0.314	0.279	0.319	0.277	0.323	**0.266**	0.314	0.281	0.330	0.287	0.330	0.288	0.332	0.288	0.332
ECL	0.167	0.263	0.187	0.279	0.175	0.265	**0.164**	**0.259**	0.168	**0.259**	0.175	0.272	0.174	0.264	0.170	0.265	0.178	0.270
Solar	**0.195**	0.246	0.240	0.270	0.233	**0.244**	0.198	0.256	0.210	0.261	0.301	0.303	0.229	0.256	0.240	0.273	0.233	0.262
Weather	**0.240**	**0.265**	0.251	0.267	0.254	0.272	0.242	0.269	0.243	0.271	0.270	0.309	0.255	0.278	0.251	0.276	0.258	0.278
PEMS03	**0.089**	**0.192**	0.175	0.275	0.119	0.226	0.097	0.203	0.118	0.226	0.144	0.230	0.104	0.210	0.122	0.228	0.113	0.222
PEMS04	**0.090**	**0.195**	0.215	0.304	0.207	0.298	0.091	0.197	0.119	0.232	0.104	0.210	0.102	0.208	0.103	0.211	0.111	0.221
PEMS07	0.079	0.174	0.169	0.266	0.112	0.206	**0.075**	**0.171**	0.113	0.214	0.094	0.193	0.086	0.184	0.089	0.188	0.101	0.204
PEMS08	**0.106**	**0.199**	0.220	0.297	0.199	0.292	0.142	0.230	0.150	0.246	0.151	0.234	0.138	0.220	0.148	0.224	0.150	0.226
1st Count	**6**	**5**	2	3	0	1	2	2	1	1	0	0	0	0	0	0	0	0

The remaining kernel dimension covers a periodicity-aware temporal neighborhood, with cyclic structure handled during retrieval and length-preserving padding used during fusion. Unlike direct addition, the convolution can respond jointly to local evidence and the adapted prior at neighboring time steps. It therefore learns when the periodic cue is compatible with the observation and how much of that cue should be retained. The resulting feature $\mathbf{F}$ preserves local and fine-grained dynamics that are absent from the global template while selectively injecting recurring structure.

Finally, PeriCal adds the fused periodic information to the normalized input and projects the temporal dimension into a latent space:

$$\mathbf{E} = \mathbf{W}_{\text{in}}(\mathbf{Z} + \mathbf{F}) + \mathbf{b}_{\text{in}}. \tag{12}$$

A two-layer MLP $\mathcal{M}$ and a residual connection produce the normalized forecast:

$$\widehat{\mathbf{Y}}^{\,n} = \mathbf{W}_{\text{out}}\left(\mathcal{M}(\mathbf{E}) + \mathbf{E}\right) + \mathbf{b}_{\text{out}}. \tag{13}$$

When normalization is enabled, the prediction is transformed back to the original value range using the statistics in Equation equation 1. The training objective is specified in the appendix.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate PeriCal on 11 widely used multivariate benchmarks covering diverse temporal patterns and periodic variations across different domains and temporal regimes. The ETT datasets, Electricity, and Solar-Energy represent energy systems with pronounced daily or longer-term regularities; Weather contains environmental variables whose periodic variations are affected by changing atmospheric conditions; and PEMS03/04/07/08 provide traffic measurements with strong daily cycles and substantial variation in peak timing and intensity. This diversity allows us to assess whether adaptive periodic calibration generalizes across energy, environment, and transportation domains rather than relying on a specific type of temporal variation. For the first seven datasets, the prediction horizons are $H \in \{96, 192, 336, 720\}$; for the PEMS datasets, $H \in \{12, 24, 48, 96\}$. The look-back length is fixed to $L = 96$ in the full-data experiments. Detailed dataset statistics and variable descriptions are provided in the appendix.

Metrics. We evaluate forecasting accuracy using Mean Squared Error (MSE) and Mean Absolute Error (MAE):

$$\text{MSE} = \frac{1}{HC}\sum_{t=1}^{H}\sum_{c=1}^{C}\left(Y_{t,c} - \widehat{Y}_{t,c}\right)^2, \tag{14}$$

$$\text{MAE} = \frac{1}{HC}\sum_{t=1}^{H}\sum_{c=1}^{C}\left|Y_{t,c} - \widehat{Y}_{t,c}\right|. \tag{15}$$

Lower values indicate better performance. The comparative table reports both metrics averaged over all prediction horizons for each dataset. The horizon-wise results are in Table 7 of the appendix.

Baselines. We compare PeriCal with representative state-of-the-art forecasters covering adaptive representation learning, periodic modeling, retrieval augmentation, lightweight multivariate modeling, and structured sequence architectures: TimeMosaic Ding et al. (2026), TimeBridge Liu et al. (2025), TQNet Lin et al. (2025), CycleNet Lin et al. (2024a), RAFT Han et al. (2025), SOFTS Han et al. (2024), S-Mamba Wang et al. (2025), and iTransformer Liu et al. (2024). All methods use the same train/validation/test splits, and we follow the experimental configurations reported in their original papers for a consistent comparison.

4.2 Comparative Experiments

Table 1 reports the average forecasting performance over all prediction horizons on 11 multivariate benchmarks, including the ETT, Electricity, Solar-Energy, Weather, and PEMS datasets. PeriCal achieves leading performance on ETTm1, Solar-Energy, Weather, PEMS03, PEMS04, and PEMS08, while remaining competitive on the other datasets. These benchmarks cover diverse application domains, sampling frequencies, variable dimensions, and forecasting horizons, indicating that PeriCal is not tied to a specific temporal resolution or data regime.

The gains are particularly evident on traffic forecasting. On PEMS08, PeriCal reduces the average MSE from 0.138 for the strongest baseline to 0.106, a relative improvement of 23.2%. Traffic series exhibit recurring patterns associated with commuting behavior, but the timing and intensity of peaks can vary across days. These results suggest that adapting the alignment and strength of periodic cues to the current input is more effective than reusing a fixed periodic representation.

The gains are not uniform: TimeMosaic remains stronger on the hourly ETT datasets, and CycleNet attains a lower MSE on ETTm2. Thus, the results support adaptive calibration as a useful forecasting mechanism, not as a universal substitute for other model biases. Horizon-wise results in the appendix show the variation hidden by dataset averages.

4.3 Computational Efficiency

We measure computational cost on Electricity with $L = 96$ and $H = 720$ using PyTorch 2.6.0 and CUDA 12.4 on one NVIDIA RTX A6000 GPU. Parameters count trainable weights, MACs are multiply–accumulate operations per forward pass, and training time is measured per epoch under the same hardware and data pipeline.

Table 2: Computational efficiency on Electricity ($L = 96$, $H = 720$); lower is better.

Model	Parameters	MACs	Time (s/epoch)
Informer (2021)	12.53M	3.97G	70.1
Autoformer (2021)	12.22M	4.41G	107.7
FEDformer (2022)	17.98M	4.41G	238.7
PatchTST (2023)	10.74M	25.87G	129.5
iTransformer (2024)	5.15M	1.65G	35.1
CycleNet (2024)	472.9K	134.84M	30.8
PeriCal (Ours)	**1.02M**	**309.89M**	**34.2**

Table 2 shows that PeriCal is substantially lighter than the Transformer-based baselines despite its input-conditioned calibration. Compared with iTransformer, it uses approximately 80.2% fewer parameters and 81.2% fewer MACs at a comparable per-epoch training time. CycleNet is more compact, but PeriCal's training time remains similar despite its two calibration branches. Parameters and MACs describe model size and forward-pass arithmetic, whereas epoch time also depends on backward propagation and data loading. The smaller MAC count therefore should not be read as a proportional speedup: PeriCal's measured epoch time is close to iTransformer's.

Table 3: Ablation study. VC and TC denote value calibration and temporal calibration, respectively; "w/o T&V" removes both operations. Best results are shown in bold.

Dataset	H	**PeriCal**		w/o VC		w/o TC		w/o T&V	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.309**	**0.355**	0.315	0.362	0.319	0.366	0.322	0.371
	192	**0.346**	**0.374**	0.353	0.377	0.355	0.379	0.361	0.382
	336	**0.376**	**0.397**	0.380	0.400	0.386	0.401	0.390	0.404
	720	**0.433**	**0.433**	0.439	0.436	0.442	0.438	0.448	0.442
Weather	96	**0.155**	**0.199**	0.163	0.206	0.168	0.213	0.174	0.225
	192	**0.203**	**0.244**	0.209	0.250	0.212	0.255	0.216	0.260
	336	**0.261**	**0.276**	0.269	0.292	0.271	0.295	0.279	0.298
	720	**0.342**	**0.342**	0.352	0.349	0.355	0.354	0.365	0.362
PEMS08	12	**0.067**	**0.165**	0.070	0.168	0.071	0.169	0.074	0.172
	24	**0.087**	**0.187**	0.090	0.190	0.094	0.193	0.098	0.196
	48	**0.111**	**0.208**	0.123	0.219	0.137	0.223	0.149	0.236
	96	**0.157**	**0.236**	0.201	0.259	0.214	0.273	0.258	0.295

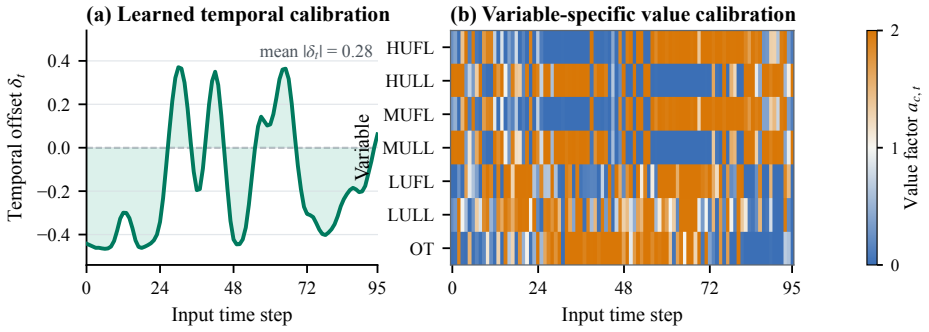


Figure 3: Learned (a) temporal offsets and (b) variable-specific value factors on ETTm1.

4.4 Ablation Study

We isolate the two calibration operations in PeriCal through four controlled variants. Removing temporal calibration fixes the periodic coordinates ($\boldsymbol{\delta} = \mathbf{0}$), while removing value calibration fixes the calibration map ($\mathbf{A} = \mathbf{1}$). Removing both operations recovers a fixed time-indexed periodic prior. All variants retain the same forecasting backbone, optimization settings, and cycle lengths.

Table 3 shows that PeriCal performs best across all reported datasets, horizons, and metrics. Removing either operation degrades accuracy; temporal calibration generally contributes more, while value calibration provides a complementary gain. These controlled comparisons support the benefit of calibrating both when a fixed periodic prior is inconsistent with the current input. The gap from fixed retrieval widens at longer PEMS08 horizons, where periodic mismatch may be more consequential: at $H = 96$, removing both operations increases MSE from 0.157 to 0.258. Each single-removal variant probes a different requirement: adapting the retrieval coordinates or adapting the cue values. Their complementary forecasting gains do not, on their own, establish that either learned factor recovers a physical timing shift or magnitude change.

4.5 Qualitative Analysis

Figure 4 shows that PeriCal better aligns recurring patterns than a fixed prior while leaving transient fluctuations to the local path. Figure 3 reveals smooth temporal offsets and variable-specific value factors. The offsets vary across input positions rather than imposing a single shift, while the value factors differ across variables. Together with Table 3, these observations support the intended roles of the two calibration operations.

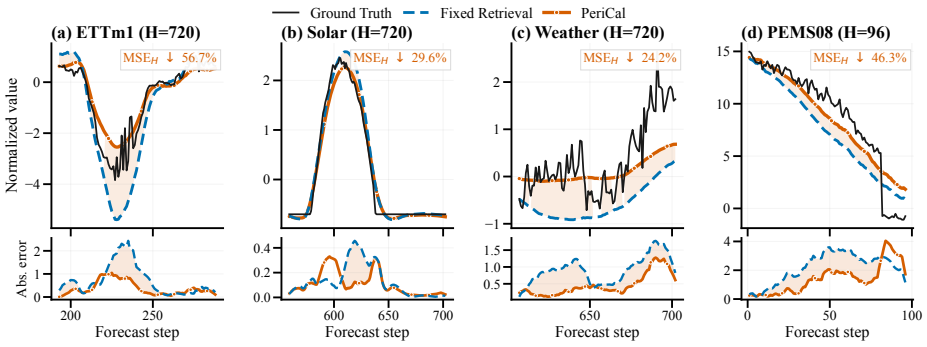


Figure 4: Representative forecasts and moving-averaged absolute errors. Fixed Retrieval disables both calibration operations.

The figures offer complementary evidence: Figure 4 shows forecast behavior, while Figure 3 displays the internal calibration factors. Smooth, position-dependent offsets and unequal variable-wise factors are consistent with the intended operations, but are not ground-truth recovery measurements. Moreover, Fixed Retrieval in Figure 4 disables both factors at test time in the trained PeriCal model; it is not a separately retrained baseline. The matched training-time variants in Table 3 provide the controlled accuracy comparison.

5 Limitations

PeriCal uses a prespecified nominal cycle length and a temporal offset shared across variables to keep retrieval compact and synchronized. This setting matches the daily and weekly regularities evaluated here; more complex temporal structures, including overlapping or changing periods, may benefit from multi-period extensions. The ablations establish forecasting gains from calibration, while the visualized factors serve as diagnostic signals rather than direct estimates of physical timing or magnitude changes. Future work can test their fidelity on controlled data.

6 Conclusion

We presented PeriCal, a lightweight forecaster that calibrates the timing and values of periodic cues before fusing them with local observations. Results on 11 benchmarks, together with component and efficiency analyses, show that input-conditioned calibration improves the utility of periodic information across domains and horizons. Together, these findings support adapting a shared periodic prior while retaining local observations as a separate source of forecasting evidence.

AI use statement

Generative AI tools were used during manuscript preparation for research brainstorming, language editing, LaTeX formatting, figure refinement, and assistance with experimental code and scripts. The authors reviewed and validated all AI-assisted text, citations, implementation changes, experimental results, figures, and conclusions, and take full responsibility for the final content of this work.

Ethics statement

This work uses only established public time-series benchmarks and does not involve human participants, private data, or deployment in safety-critical settings. We report the experimental protocol and limitations as accurately as possible and do not identify additional ethical concerns specific to this study.

Reproducibility statement

The model architecture is specified in Section 3, while datasets, evaluation metrics, baselines, and experimental protocols are described in Section 4. The appendix provides dataset details, the training objective, gradient sensitivity analysis, and full few-shot results. Source code, configuration files, and execution scripts will be released upon publication.

References

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11141–11149, 2024.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *International Conference on Learning Representations*, 2026.

Haizhou Cao, Zhenhao Huang, Tiechui Yao, Jue Wang, Hui He, and Yangang Wang. Inparformer: Evolutionary decomposition transformers with interactive parallel attention for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6906–6915, 2023.

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*, 2024a.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Yuerong Xue, Shu-Tao Xia, and Zexuan Zhu. DDN: Dual-domain dynamic normalization for non-stationary time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026.

Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. Depts: Deep expansion learning for periodic time series forecasting. In *International Conference on Learning Representations*, 2022.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11645–11653, 2025.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O. Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21774–21797, 2025.

Jiaxi Hu, Yuehong Hu, Wei Chen, Ming Jin, Shirui Pan, Qingsong Wen, and Yuxuan Liang. Attractor memory for long-term time series forecasting: A chaos perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Zahra Karevan and Johan A. K. Suykens. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4191–4203, 2019.

Xinyu Li, Yuchen Luo, Hao Wang, Haoxuan Li, Liuhua Peng, Feng Liu, Yandong Guo, Kun Zhang, and Mingming Gong. Towards accurate time series forecasting via implicit decoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30211–30226, 2024b.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814, 2025.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. TimeBridge: Non-stationarity matters for long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840, 2025.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12271–12290, 2023a.

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

Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

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

Donghao Luo and Xue Wang. DeformableTST: Transformer for time series forecasting without over-reliance on patching. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. MoFo: Empowering long-term time series forecasting with periodic pattern modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Hao Miao, Senzhang Wang, Meiyue Zhang, Diansheng Guo, Funing Sun, and Fan Yang. Deep multi-view channel-wise spatio-temporal network for traffic flow prediction. *arXiv preprint arXiv:2404.15034*, 2024.

Reza Nematirad, Anil Pahwa, and Balasubramaniam Natarajan. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19651–19658, 2025.

Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. Basisformer: Attention-based time series forecasting with learnable and interpretable basis. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, and Jianxin Liao. Rethinking the power of timestamps for robust time series forecasting: A global-local fusion perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

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*, 2024b.

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.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. ETSformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

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.

Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, 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, 2023b.

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, 2024.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I. Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57818–57841, 2024.

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

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78478–78497, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

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, 2022a.

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, 2022b.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Dataset Details

We evaluate PeriCal on 11 public multivariate time-series forecasting benchmarks spanning energy, weather, and transportation. Table 4 summarizes their domains, sampling frequencies, sequence lengths, and numbers of variables. The four ETT subsets record electricity-transformer measurements at hourly or 15-minute resolution. Electricity contains hourly consumption measurements from 321 clients, Solar-Energy records solar-power production every 10 minutes, and Weather contains 21 meteorological variables sampled every 10 minutes. PEMS03, PEMS04, PEMS07, and PEMS08 contain traffic measurements from California sensor networks at five-minute resolution.

Table 4: Dataset statistics and prespecified cycle lengths. Length denotes the number of time points before constructing forecasting windows; P is the daily or weekly cycle length in sampling steps.

Dataset	Frequency	Length	Vars.	P	Horizons
ETTh1	1 h	17,420	7	24	96/192/336/720
ETTh2	1 h	17,420	7	24	96/192/336/720
ETTm1	15 min	69,680	7	96	96/192/336/720
ETTm2	15 min	69,680	7	96	96/192/336/720
Electricity	1 h	26,304	321	168	96/192/336/720
Solar-Energy	10 min	52,560	137	144	96/192/336/720
Weather	10 min	52,696	21	144	96/192/336/720
PEMS03	5 min	26,208	358	288	12/24/48/96
PEMS04	5 min	16,992	307	288	12/24/48/96
PEMS07	5 min	28,224	883	288	12/24/48/96
PEMS08	5 min	17,856	170	288	12/24/48/96

B Implementation and Evaluation Details

The full-data experiments use a look-back length of $L = 96$. The ETT, Electricity, Solar-Energy, and Weather datasets are evaluated at $H \in \{96, 192, 336, 720\}$, whereas the PEMS datasets use $H \in \{12, 24, 48, 96\}$. Models are selected using validation loss and evaluated with MSE and MAE on the standard test splits. The few-shot experiments use a look-back length of 512 and retain 5%

of the training windows. ETTh1 and ETTh2 are evaluated at horizons 96, 192, and 336 because the retained training subset is insufficient to construct the $H = 720$ setting under this protocol.

C Training Objective

The complete model is optimized end to end with mean squared forecasting error, computed after reversing instance normalization when it is enabled:

$$\mathcal{L} = \frac{1}{HC} \sum_{t=1}^{H} \sum_{c=1}^{C} \left(Y_{t,c} - \widehat{Y}_{t,c} \right)^2 . \tag{16}$$

D Few-Shot Forecasting Results

Pretrained language and vision-language models can perform well with limited time-series supervision Zhong et al. (2025); Jin et al. (2024); Zhou et al. (2023). PeriCal instead uses only time-series data. Following Time-VLM's 5% training-data protocol, we retain 5% of training windows and use a look-back length of 512. We evaluate ETTh1/ETTh2 at horizons $\{96, 192, 336\}$ and ETTm1/ETTm2 additionally at $H = 720$.

Table 5: Average forecasting results under the 5% few-shot setting. Each dataset entry averages the evaluated horizons. Best results are in bold, and second-best results are underlined; ties after rounding share the same rank.

Dataset	**PeriCal (Ours)**		Time-VLM (2025)		Time-LLM (2024)		GPT4TS (2023)	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.430**	**0.441**	<u>0.442</u>	<u>0.453</u>	0.627	0.543	0.682	0.560
ETTh2	**0.344**	**0.390**	<u>0.354</u>	<u>0.402</u>	0.382	0.418	0.401	0.434
ETTm1	<u>0.370</u>	<u>0.392</u>	**0.364**	**0.385**	0.425	0.434	0.472	0.450
ETTm2	<u>0.271</u>	<u>0.330</u>	**0.263**	**0.323**	0.274	**0.323**	0.308	0.346

Table 5 shows that PeriCal leads on ETTh1 and ETTh2 and remains competitive on ETTm1 and ETTm2. Unlike the pretrained baselines, it does so without language corpora, visual encoders, or cross-modal pretraining. This is not zero-shot transfer; rather, it indicates that adaptive periodic modeling provides a time-series-specific bias when supervision is limited.

Table 6 expands the dataset averages in Table 5 into individual prediction horizons. The look-back length is $L = 512$, and PeriCal uses 5% of the original training windows. The hourly ETT datasets report $H \in \{96, 192, 336\}$; under this few-shot protocol, the retained subset is insufficient to form a valid $H = 720$ training set. Their averages therefore cover three horizons, whereas the ETTm1 and ETTm2 averages cover all four.

PeriCal leads on all reported ETTh1 horizons and most ETTh2 horizons, whereas Time-VLM leads on several minute-level horizons. These results support competitive limited-data forecasting without language or vision-language pretraining, rather than universal superiority.

E Gradient Sensitivity Analysis

We compare local gradient sensitivity for GTR and PeriCal on an ETTm1 test window ($L = H = 96$), targeting the OT variable. Let $R_{\mathrm{OT},t}$ denote its periodic cue before the temporal linear mapping and $\widetilde{y}_{h,\mathrm{OT}}$ the forecast before RevIN denormalization. Figure 5 plots $|\partial \widetilde{y}_{h,\mathrm{OT}} / \partial R_{\mathrm{OT},t}|$ at 12 evenly spaced future steps. The independently trained checkpoints share the same dataset, configuration, and random seed.

The window (index 2496) represents the median positive OT-MSE difference among 120 regularly spaced candidates, not the largest gain. GTR shows broader contiguous sensitivity, whereas PeriCal varies more sharply across input steps. These absolute derivatives neither reveal effect direction nor

Table 6: Full 5% few-shot forecasting results on the ETT datasets with look-back length $L = 512$. Best results are in bold and second-best results are underlined; a dash marks an unavailable horizon.

Dataset	H	**PeriCal (Ours)**		Time-VLM (2025)		Time-LLM (2024)		GPT4TS (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.393**	**0.416**	0.417	0.435	0.483	0.464	0.543	0.506
	192	**0.438**	**0.446**	0.450	0.458	0.629	0.540	0.748	0.580
	336	**0.458**	**0.462**	0.460	0.465	0.768	0.626	0.754	0.595
	720	–	–	–	–	–	–	–	–
	Avg	**0.430**	**0.441**	0.442	0.453	0.627	0.543	0.682	0.560
ETTh2	96	0.312	**0.365**	**0.302**	**0.365**	0.336	0.397	0.376	0.421
	192	**0.350**	**0.392**	0.361	0.406	0.406	0.425	0.418	0.441
	336	**0.370**	**0.412**	0.398	0.434	0.405	0.432	0.408	0.439
	720	–	–	–	–	–	–	–	–
	Avg	**0.344**	**0.390**	0.354	0.402	0.382	0.418	0.401	0.434
ETTm1	96	0.324	0.364	**0.314**	**0.357**	0.316	0.377	0.386	0.405
	192	0.350	0.379	**0.343**	**0.373**	0.450	0.464	0.440	0.438
	336	0.380	0.399	**0.373**	**0.391**	0.450	0.424	0.485	0.459
	720	0.427	0.427	**0.425**	**0.420**	0.483	0.471	0.577	0.499
	Avg	0.370	0.392	**0.364**	**0.385**	0.425	0.434	0.472	0.450
ETTm2	96	0.178	0.269	**0.169**	**0.260**	0.174	0.261	0.199	0.280
	192	0.232	0.304	0.224	0.298	**0.215**	**0.287**	0.256	0.316
	336	0.290	0.344	0.282	0.338	**0.273**	**0.330**	0.318	0.353
	720	0.382	0.401	**0.375**	**0.397**	0.433	0.412	0.460	0.436
	Avg	0.271	0.330	**0.263**	**0.323**	0.274	**0.323**	0.308	0.346

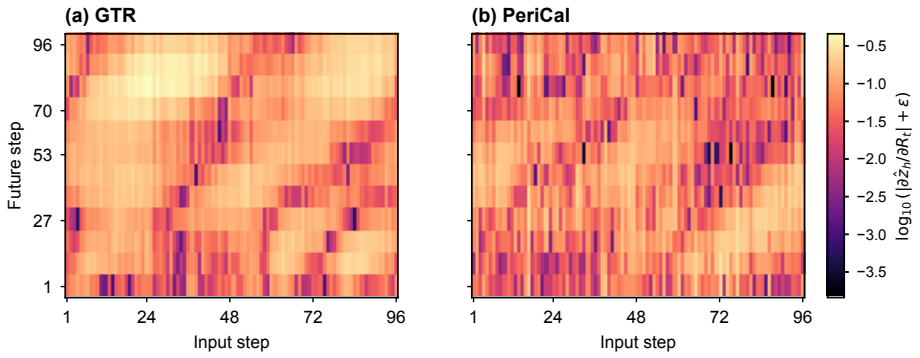


Figure 5: OT-specific gradient sensitivity on ETTm1: (a) GTR and (b) PeriCal. Rows are future steps and columns are input steps. The shared color scale is $\log_{10}(|\partial \widetilde{y}_{h,\mathrm{OT}}/\partial R_{\mathrm{OT},t}| + \varepsilon)$; brighter means greater local sensitivity, not higher forecasting accuracy.

isolate a calibration branch's causal contribution; the main-paper ablations and forecasting metrics provide the performance evidence.

Table 7: Complete multivariate forecasting results over all prediction horizons with look-back length $L = 96$. Each model reports MSE and MAE. Avg denotes the mean across four horizons; lower is better. Best values are in **bold**, and second-best values are underlined.

Dataset	H	**PeriCal (Ours)**		TimeMosaic (2026)		TimeBridge (2025)		TQNet (2025)		CycleNet (2024)		RAFT (2025)		SOFTS (2024)		S-Mamba (2025)		iTransformer (2024)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.371	0.392	**0.369**	**0.389**	0.375	0.390	0.371	0.393	0.375	0.395	0.376	0.401	0.381	0.399	0.386	0.405	0.386	0.405
	192	0.424	0.422	**0.416**	**0.415**	0.428	0.421	0.428	0.426	0.436	0.428	0.417	0.424	0.435	0.431	0.443	0.437	0.441	0.436
	336	0.469	0.443	**0.454**	**0.435**	0.470	0.443	0.476	0.446	0.496	0.455	**0.454**	0.441	0.480	0.452	0.489	0.468	0.487	0.458
	720	0.487	0.471	**0.461**	**0.458**	0.481	0.481	0.487	0.470	0.520	0.484	0.465	0.465	0.499	0.488	0.502	0.489	0.503	0.491
	Avg	0.438	0.432	**0.425**	**0.424**	0.439	0.434	0.441	0.434	0.457	0.441	0.428	0.433	0.449	0.443	0.455	0.450	0.454	0.448
ETTh2	96	0.289	0.339	**0.286**	**0.332**	**0.286**	0.333	0.295	0.343	0.298	0.344	0.289	0.342	0.297	0.347	0.296	0.348	0.297	0.349
	192	0.368	0.391	**0.362**	**0.380**	0.364	0.385	0.367	0.393	0.372	0.396	0.382	0.400	0.373	0.394	0.376	0.396	0.380	0.400
	336	0.414	0.427	**0.409**	**0.416**	0.416	0.425	0.417	0.427	0.431	0.439	0.424	0.439	0.410	0.426	0.424	0.431	0.428	0.432
	720	0.422	0.441	**0.396**	**0.422**	0.435	0.448	0.433	0.446	0.450	0.458	0.433	0.457	0.411	0.433	0.426	0.444	0.427	0.445
	Avg	0.373	0.400	**0.363**	**0.388**	0.375	0.398	0.378	0.402	0.388	0.409	0.382	0.410	0.373	0.400	0.381	0.405	0.383	0.407
ETTm1	96	0.309	0.355	**0.308**	**0.338**	0.312	0.344	0.311	0.353	0.319	0.360	0.325	0.369	0.325	0.361	0.333	0.368	0.334	0.368
	192	**0.346**	0.374	0.364	**0.368**	0.366	0.377	0.356	0.378	0.360	0.381	0.364	0.388	0.375	0.389	0.376	0.390	0.377	0.391
	336	**0.376**	0.397	0.394	**0.390**	0.403	0.404	0.390	0.401	0.389	0.403	0.391	0.407	0.405	0.412	0.408	0.413	0.426	0.420
	720	**0.433**	0.433	0.460	**0.428**	0.454	0.438	0.452	0.440	0.447	0.441	0.444	0.438	0.466	0.447	0.475	0.448	0.491	0.459
	Avg	**0.366**	0.390	0.381	**0.381**	0.384	0.391	0.377	0.393	0.379	0.396	0.381	0.400	0.393	0.402	0.398	0.405	0.407	0.410
ETTm2	96	0.168	0.249	0.170	0.248	0.173	0.250	0.173	0.256	**0.163**	**0.246**	0.176	0.264	0.180	0.261	0.179	0.263	0.180	0.264
	192	0.232	**0.287**	0.236	0.291	0.243	0.297	0.238	0.298	**0.229**	0.290	0.241	0.306	0.246	0.306	0.250	0.309	0.250	0.309
	336	0.293	**0.325**	0.295	0.331	0.302	0.336	0.301	0.340	**0.284**	0.327	0.302	0.345	0.319	0.352	0.312	0.349	0.311	0.348
	720	0.392	0.391	0.392	**0.386**	0.399	0.394	0.397	0.396	**0.389**	0.391	0.404	0.405	0.405	0.401	0.411	0.406	0.412	0.407
	Avg	0.271	**0.313**	0.273	0.314	0.279	0.319	0.277	0.323	**0.266**	0.314	0.281	0.330	0.287	0.330	0.288	0.332	0.288	0.332
Electricity	96	0.136	0.232	0.160	0.258	0.138	0.231	**0.134**	**0.229**	0.136	**0.229**	0.152	0.257	0.143	0.233	0.139	0.235	0.148	0.240
	192	0.154	0.250	0.174	0.269	0.158	0.250	0.154	0.247	**0.152**	**0.244**	0.156	0.257	0.158	0.248	0.159	0.255	0.162	0.253
	336	0.172	0.268	0.189	0.283	0.180	0.271	**0.169**	**0.264**	0.170	**0.264**	0.171	0.272	0.178	0.269	0.176	0.272	0.178	0.269
	720	0.207	0.301	0.225	0.304	0.223	0.308	**0.201**	**0.294**	0.212	0.299	0.221	0.300	0.218	0.305	0.204	0.298	0.225	0.317
	Avg	0.167	0.263	0.187	0.279	0.175	0.265	**0.164**	**0.259**	0.168	**0.259**	0.175	0.272	0.174	0.264	0.170	0.265	0.178	0.270
Solar-Energy	96	0.177	0.235	0.209	0.248	0.196	**0.214**	**0.173**	0.233	0.190	0.247	0.278	0.278	0.200	0.230	0.205	0.244	0.203	0.237
	192	**0.193**	0.246	0.238	0.272	0.230	**0.239**	0.199	0.257	0.210	0.266	0.297	0.297	0.229	0.253	0.237	0.270	0.233	0.261
	336	**0.202**	**0.250**	0.255	0.277	0.250	0.258	0.211	0.263	0.217	0.266	0.311	0.311	0.243	0.269	0.258	0.288	0.248	0.273
	720	**0.206**	**0.252**	0.257	0.284	0.256	0.264	0.209	0.270	0.223	0.266	0.318	0.325	0.245	0.272	0.260	0.288	0.249	0.275
	Avg	**0.195**	0.246	0.240	0.270	0.233	**0.244**	0.198	0.256	0.210	0.261	0.301	0.303	0.229	0.256	0.240	0.273	0.233	0.262
Weather	96	**0.155**	0.199	0.165	**0.198**	0.169	0.206	0.157	0.200	0.158	0.203	0.185	0.241	0.166	0.208	0.165	0.210	0.174	0.214
	192	**0.203**	0.244	0.215	**0.243**	0.219	0.249	0.206	0.245	0.207	0.247	0.237	0.287	0.217	0.253	0.214	0.252	0.221	0.254
	336	**0.261**	**0.276**	0.272	0.286	0.274	0.290	0.262	0.287	0.262	0.289	0.290	0.327	0.282	0.300	0.274	0.297	0.278	0.296
	720	**0.342**	0.342	0.353	**0.340**	0.352	0.341	0.344	0.342	0.344	0.344	0.367	0.380	0.356	0.351	0.350	0.345	0.358	0.349
	Avg	**0.240**	**0.265**	0.251	0.267	0.254	0.272	0.242	0.269	0.243	0.271	0.270	0.309	0.255	0.278	0.251	0.276	0.258	0.278
PEMS03	12	**0.057**	**0.157**	0.078	0.186	0.062	0.164	0.060	0.161	0.066	0.172	0.071	0.175	0.064	0.165	0.065	0.169	0.071	0.174
	24	**0.070**	**0.174**	0.116	0.229	0.085	0.192	0.077	0.182	0.089	0.201	0.099	0.201	0.083	0.188	0.087	0.196	0.093	0.201
	48	**0.096**	**0.202**	0.190	0.296	0.132	0.242	0.104	0.215	0.136	0.247	0.153	0.243	0.114	0.223	0.133	0.243	0.125	0.236
	96	**0.133**	**0.234**	0.315	0.388	0.197	0.304	0.148	0.253	0.182	0.282	0.253	0.301	0.156	0.264	0.201	0.305	0.164	0.275
	Avg	**0.089**	**0.192**	0.175	0.275	0.119	0.226	0.097	0.203	0.118	0.226	0.144	0.230	0.104	0.210	0.122	0.228	0.113	0.222
PEMS04	12	**0.067**	0.167	0.100	0.211	0.094	0.199	**0.067**	**0.166**	0.078	0.186	0.077	0.179	0.074	0.176	0.076	0.180	0.078	0.183
	24	**0.077**	**0.181**	0.140	0.248	0.139	0.246	**0.077**	**0.181**	0.099	0.212	0.091	0.195	0.088	0.194	0.084	0.193	0.095	0.205
	48	**0.096**	**0.203**	0.231	0.322	0.233	0.326	0.097	0.206	0.133	0.248	0.111	0.218	0.110	0.219	0.115	0.224	0.120	0.233
	96	**0.121**	**0.230**	0.388	0.434	0.362	0.419	0.123	0.233	0.167	0.281	0.137	0.247	0.135	0.244	0.137	0.248	0.150	0.262
	Avg	**0.090**	**0.195**	0.215	0.304	0.207	0.297	0.091	0.197	0.119	0.232	0.104	0.210	0.102	0.208	0.103	0.211	0.111	0.221
PEMS07	12	0.052	0.144	0.074	0.181	0.060	0.152	**0.051**	**0.143**	0.062	0.162	0.062	0.159	0.057	0.152	0.063	0.159	0.067	0.165
	24	0.064	**0.159**	0.113	0.221	0.083	0.179	**0.063**	**0.159**	0.086	0.192	0.078	0.178	0.073	0.173	0.081	0.183	0.088	0.190
	48	0.084	0.182	0.197	0.292	0.123	0.220	**0.081**	**0.179**	0.128	0.234	0.102	0.203	0.096	0.195	0.093	0.192	0.110	0.215
	96	0.117	0.210	0.292	0.369	0.180	0.274	**0.103**	**0.203**	0.176	0.268	0.135	0.231	0.120	0.218	0.117	0.217	0.139	0.245
	Avg	0.079	0.174	0.169	0.266	0.112	0.206	**0.075**	**0.171**	0.113	0.214	0.094	0.193	0.086	0.184	0.089	0.188	0.101	0.204
PEMS08	12	**0.067**	**0.165**	0.107	0.221	0.086	0.191	0.071	0.170	0.082	0.185	0.078	0.181	0.074	0.171	0.076	0.178	0.079	0.182
	24	**0.087**	**0.187**	0.124	0.224	0.128	0.236	0.096	0.196	0.117	0.226	0.102	0.205	0.104	0.201	0.104	0.209	0.115	0.219
	48	**0.111**	**0.208**	0.304	0.371	0.217	0.317	0.149	0.244	0.169	0.268	0.148	0.244	0.164	0.253	0.167	0.228	0.186	0.235
	96	**0.157**	**0.236**	0.345	0.371	0.364	0.425	0.253	0.309	0.233	0.306	0.277	0.307	0.211	0.253	0.245	0.280	0.221	0.267
	Avg	**0.106**	**0.199**	0.220	0.297	0.199	0.292	0.142	0.230	0.150	0.246	0.151	0.234	0.138	0.220	0.148	0.224	0.150	0.226