TIMERIFT: LEARNING BACKGROUND-RELATIVE DYNAMICS FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Similar local dynamics in multivariate time series can evolve into markedly different future trajectories under distinct recurrent temporal backgrounds. Although existing methods primarily incorporate temporal context or disentangle periodic components, they fail to explicitly model how recurrent temporal backgrounds shape the evolution of local dynamics. To address this critical issue, we propose **TimeRift**, a forecasting framework that explicitly models background-conditioned dynamics. Specifically, TimeRift firstly learns recurrent temporal background prototypes from recurrent temporal indices and represents each input as a deviation from its corresponding background, which serves as both a reference for local dynamics and a conditioning signal for subsequent evolution. Further, TimeRift calibrates background-conditioned deviations via context-dependent scaling and shifting, and then evolves them using adaptive spectral weights to fuse global patterns with background-specific adjustments. Finally, the evolved dynamics are recombined with the temporal background to obtain the forecasting. Comprehensive experiments on seven real-world benchmarks demonstrate that TimeRift achieves competitive forecasting accuracy with low computational overhead. Moreover, ablation and controlled intervention studies further confirm the contribution of background conditioning for accurate forecasting. Our code is available at https://anonymous.4open.science/r/TimeRift.

1 INTRODUCTION

Multivariate time series forecasting (MTSF) plays an important role in real-world systems such as energy management Ding et al. (2026); Lv et al. (2025), traffic control Liu et al. (2026); Tan et al. (2026), weather analysis Ye et al. (2026); Chen et al. (2026), and industrial monitoring Yang et al. (2026); Yu et al. (2026). An important challenge across these applications is to account for the temporal background when modeling local dynamics. Similar local variations observed under different temporal backgrounds may be followed by different future trajectories Chen et al. (2024); Ni et al. (2024); Wang et al. (2024a); Liu et al. (2023; 2025). For example, a short-term increase in electricity load may indicate the onset of a demand ramp during an active usage period. The same local variation during an off-peak period may instead be followed by stabilization or decline.

Existing forecasting methods have incorporated temporal information and temporal structure in various forms Wang et al. (2026); Guo et al. (2026); Jiang et al. (2025). Some methods encode time-based features as temporal embeddings to indicate the temporal position of each observation Zhou et al. (2021); Fraikin et al. (2024); Zhou et al. (2022); Woo et al. (2023). Others capture temporal regularities through decomposition, frequency-domain modeling, or multi-scale representations Zeng et al. (2023); Stitsyuk & Choi (2025); Sun et al. (2026); Wang et al. (2024b). Covariate-aware models further leverage known future covariates or exogenous variables for multi-step forecasting Han et al. (2026); Li et al. (2026); Wang et al. (2024c). Collectively, these studies highlight the importance of temporal information in time series forecasting.

Despite these advances, a more specific question remains. **How can forecasting models distinguish similar local dynamics occurring under different temporal backgrounds when they may be followed by different future trajectories?** In many forecasting scenarios, recurrent tempo-

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.

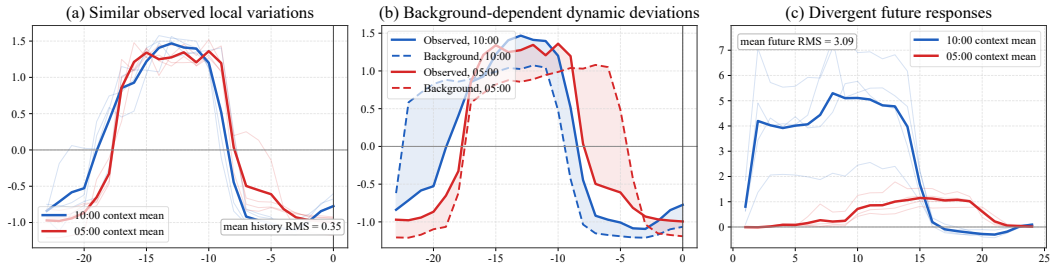


Figure 1: A motivating example from the Electricity dataset. (a) Historical windows observed at two recurring times of day, 10:00 and 05:00, exhibit similar local variations. (b) Relative to their corresponding empirical temporal backgrounds, their dynamic deviations are substantially different. (c) The two groups are subsequently followed by markedly different future trajectories. Solid and dashed lines in (b) denote the observed group means and empirical temporal backgrounds, respectively, with shaded regions indicating their deviations. Thin and thick lines in (a) and (c) denote individual windows and group means, respectively. RMS measures the root-mean-square distance between the two group-mean trajectories.

ral regularities provide a natural reference for the current dynamics Shao et al. (2022); Xun et al. (2026). The same local variation may align closely with the expected pattern under one temporal background, while representing a pronounced deviation under another. This observation suggests that future evolution cannot be inferred solely from recent local dynamics, but also depends on the temporal background in which they occur. Therefore, forecasting models should consider how temporal backgrounds influence the evolution of local dynamics. Figure 1 illustrates this phenomenon on the Electricity dataset. There, comparable local increases exhibit different deviations from their temporal backgrounds and are followed by distinct future trajectories. We further verify this phenomenon across all seven benchmarks in Appendix A.4.

Motivated by this observation, we propose **TimeRift**, a forecasting framework that explicitly models background-conditioned dynamics. Specifically, TimeRift firstly transforms the historical sequence into the spectral domain and constructs a learnable temporal background from recurrent temporal indices. Then, it obtains a background-relative dynamic deviation by removing the temporal background from the frequency representation. Next, TimeRift calibrates background-conditioned deviations via context-dependent scaling and shifting, and then evolves them using adaptive spectral weights to fuse global patterns with background-specific adjustments. This design enables the model to capture different evolution patterns for similar local dynamics under different temporal backgrounds. Finally, the evolved dynamic component is recombined with the temporal background and mapped back to the time domain for forecasting. The main contributions of this work are summarized as follows:

- We propose TimeRift, a forecasting framework that explicitly models background-conditioned dynamics. It constructs a learnable recurrent temporal background for each historical window and represents local dynamics as deviations relative to their corresponding temporal backgrounds.
- We develop a context-dependent dynamic evolution mechanism for background-relative deviations. The temporal background directly conditions their evolution, allowing TimeRift to capture different evolution patterns of local dynamics across recurrent temporal contexts.
- Extensive experiments on widely used MTSF benchmarks demonstrate that TimeRift achieves superior overall forecasting performance over strong state-of-the-art baselines. Efficiency analysis further shows that TimeRift maintains a favorable trade-off between forecasting accuracy and computational cost.

2 Related Work

2.1 Representation Learning for Time Series Forecasting

Existing time series forecasting methods have developed diverse representation mechanisms to capture complex temporal and cross-variable structures. TimeMixer Wang et al. (2024b) captures complex temporal variations by decomposing and mixing seasonal and trend patterns across multiple temporal scales. TimeXer Wang et al. (2024c) jointly models endogenous temporal patterns and exogenous information through patch-wise self-attention and variate-wise cross-attention. FilterTS Wang et al. (2025) employs learnable filters to extract and enhance informative frequency components in multivariate time series. TimeBridge Liu et al. (2025) models short-term temporal dependencies and long-term cross-variable relations by treating non-stationarity differently across temporal scales. TimePro Ma et al. (2025) models multi-delay effects using variable- and time-aware hyper-states, allowing different variables to influence future observations over different delay intervals. Collectively, these studies model complex multivariate time-series patterns through diverse representation and dependency modeling strategies.

2.2 Temporal and Periodic Modeling for Time Series Forecasting

Another important line of research exploits temporal and periodic information for time series forecasting. Informer Zhou et al. (2021) incorporates positional and timestamp embeddings into the input representation to encode temporal positions. GLAFF Wang et al. (2024a) separately models timestamps to capture global temporal dependencies and adaptively fuses them with local forecasts. EMAformer Zhang et al. (2026) introduces phase and joint channel-phase embeddings to incorporate phase-sensitive temporal information into variate representations. In recent years, increasing attention has been devoted to explicitly modeling recurrent temporal structures for time series forecasting. CycleNet Lin et al. (2024) learns recurrent cycles and performs forecasting on the residual components obtained after removing the corresponding cycle patterns. TQNet Lin et al. (2025) employs periodically shifted learnable vectors as queries to capture global inter-variable patterns. GTR Cao et al. (2026) maintains an adaptive global temporal embedding over the entire cycle and retrieves segments aligned with the current input to complement local observations. TimeEmb Xia et al. (2026) separates a learned time-invariant component from the time-varying component and processes the latter through frequency-domain filtering. PHAT Ma et al. (2026a) models periodic heterogeneity through periodic alignment and deviation, with periodic priors further incorporated into its attention mechanism. PTQNet Xun et al. (2026) separates periodic patterns from residual variations and uses seasonal phase to modulate temporal queries for residual dependency modeling. These studies demonstrate the importance of temporal and periodic information. However, similar local variations occurring under different recurrent temporal contexts can exhibit different future trajectories. How recurrent temporal backgrounds influence the evolution of local variations remains insufficiently explored.

3 Methodology

3.1 Problem Formulation

Let $X = \{x_t\}_{t=1}^{L} \in \mathbb{R}^{L \times C}$ denote a multivariate historical sequence, where L and C denote the look-back length and number of variables, respectively. Given X, the goal of MTSF is to predict the future sequence $Y = \{x_t\}_{t=L+1}^{L+T} \in \mathbb{R}^{T \times C}$, where T is the forecasting horizon. In addition to the historical observations, each input window is associated with the recurrent temporal indices of its forecasting origin, such as hour-of-day or day-of-week. We denote these indices by τ. TimeRift uses τ to construct a recurrent temporal background for modeling the dynamics of the historical sequence. The overall architecture of TimeRift is illustrated in Figure 2.

3.2 Overall Architecture

Given an input sequence $X \in \mathbb{R}^{L \times C}$, TimeRift applies reversible instance normalization. The normalized sequence is then transformed into the spectral domain using FFT. The recurrent temporal

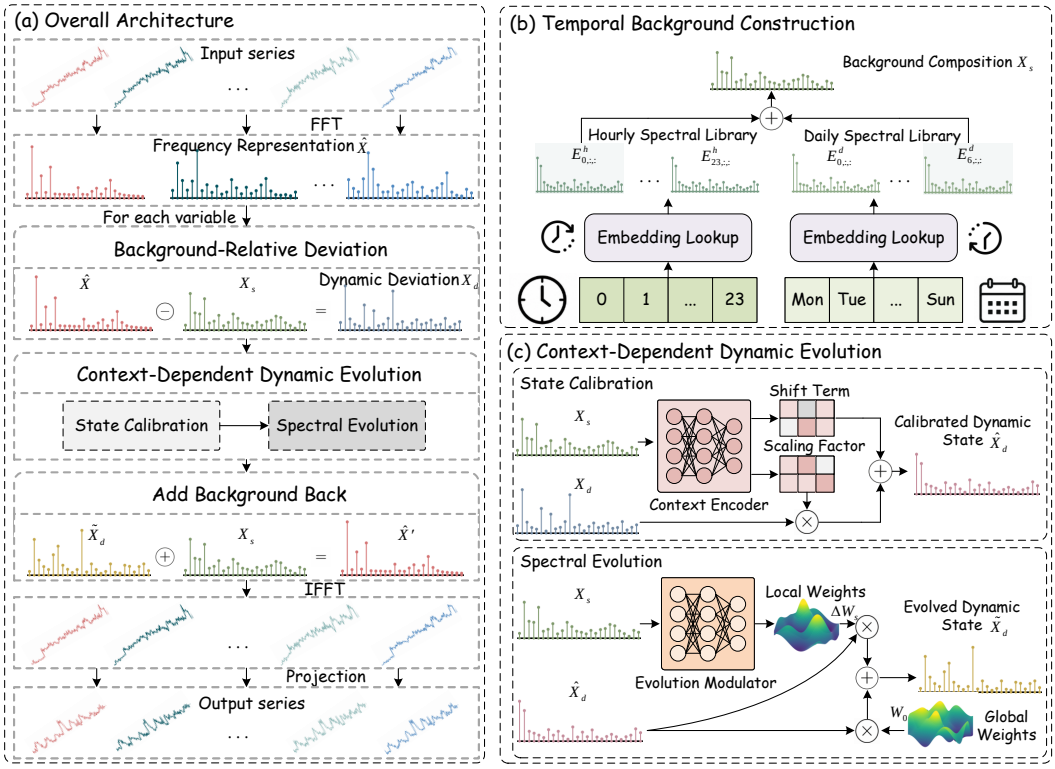


Figure 2: Overview of TimeRift. (a) **Overall Architecture:** TimeRift represents the input spectrum as a temporal background and a background-relative deviation. The temporal background guides the subsequent evolution of the deviation. The evolved deviation is recombined with the background for forecasting. (b) **Temporal Background Construction:** Hourly and daily spectral prototypes are retrieved according to the temporal indices and combined to construct the temporal background. (c) **Context-Dependent Dynamic Evolution:** State Calibration adjusts the background-relative deviation through context-dependent scaling and shifting. Spectral Evolution combines global weights with context-dependent local weights to evolve the calibrated dynamic state.

indices are used to retrieve spectral embeddings, which are combined to construct the temporal background X_s. Removing this background from the observed spectrum yields the background-relative dynamic deviation X_d. TimeRift then develop a context-dependent dynamic evolution mechanism. First, a Context Encoder conditioned on X_s generates scaling and shift terms to calibrate X_d, producing $\hat{X}_d$. Second, an Evolution Modulator generates a local spectral adjustment ΔW_s from X_s. It is combined with the global spectral weights W_0 to form background-conditioned spectral weights, which evolve $\hat{X}_d$ into $\tilde{X}_d$. Finally, X_s is added back to $\tilde{X}_d$ to reconstruct the spectrum. The reconstructed spectrum is transformed back to the time domain through IFFT and mapped to the forecasting horizon by a lightweight projection head. Inverse normalization produces the final prediction $Y \in \mathbb{R}^{T\times C}$. The following subsections describe each component in detail.

3.3 Background-Relative Deviation

TimeRift uses recurrent temporal indices to construct a real-valued spectral reference for the current temporal context. This reference represents the temporal background of the current window. The deviation of the observed spectrum from this background is then extracted.

Given an input sequence $X \in \mathbb{R}^{L\times C}$, we first apply reversible instance normalization. The normalized sequence is denoted by $\widetilde{X} \in \mathbb{R}^{L\times C}$. We then transform it into the spectral domain:

$$\hat{X} = \mathcal{F}(\widetilde{X}) \in \mathbb{C}^{C\times F}, \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the real-valued Fourier transform along the temporal dimension, and $F = \lfloor L/2 \rfloor + 1$ is the number of frequency components.

TimeRift maintains learnable spectral prototypes for recurrent temporal categories. For the hour-of-day and day-of-week factors, we define

$$E^h \in \mathbb{R}^{N_h \times C \times F}, \qquad E^d \in \mathbb{R}^{N_d \times C \times F}, \tag{2}$$

where N_h and N_d denote the numbers of hour and day categories, respectively. $E^h_{p,:,:}$ represents the spectral prototype of the p-th hour category. $E^d_{q,:,:}$ represents the prototype of the q-th day category.

For each input window, let τ_h and τ_d denote the hour-of-day and day-of-week indices of its forecasting origin, respectively. We map them to the corresponding recurrent positions:

$$p = \tau_h \bmod N_h, \qquad q = \tau_d \bmod N_d. \tag{3}$$

The temporal background is constructed from the corresponding spectral prototypes:

$$X_s = E^h_{p,:,:} + E^d_{q,:,:}. \tag{4}$$

where $X_s \in \mathbb{R}^{C \times F}$. The spectral prototypes are globally shared learnable parameters. They are jointly optimized with the forecasting model. Thus, X_s provides a spectral reference associated with the recurrent temporal context. We then obtain the background-relative deviation as

$$X_d = \hat{X} - X_s. \tag{5}$$

The deviation X_d measures the departure of the observed spectrum from its temporal background. Therefore, the same observed spectrum can produce different deviations under different temporal backgrounds. The resulting deviation is passed to the context-dependent dynamic evolution module.

3.4 Context-Dependent Dynamic Evolution

The background-relative deviation X_d characterizes how the observed spectrum departs from the recurrent temporal background X_s. However, the predictive implication of such a deviation may vary across temporal contexts. For example, a similar deviation may indicate an informative dynamic change under one temporal background, but merely correspond to a routine fluctuation under another. Therefore, its future evolution should be determined jointly by the deviation itself and the temporal background in which it occurs. To capture this context dependency, we formulate dynamic evolution as a temporal background conditioned operator:

$$\tilde{X}_d = \mathcal{E}(X_d; X_s), \tag{6}$$

where the operator consists of two stages, which are background-conditioned state calibration and background-conditioned spectral evolution.

Firstly, TimeRift calibrates the background-relative deviation under the current temporal background. The context encoder f_θ generates separate scaling and shift parameters for the real and imaginary components:

$$A^r_s, A^i_s, B^r_s, B^i_s = f_\theta(X_s). \tag{7}$$

The calibrated dynamic state is computed as

$$\begin{aligned} \mathrm{Re}(\hat{X}_d) &= (1 + \tanh(A^r_s)) \odot \mathrm{Re}(X_d) + B^r_s, \\ \mathrm{Im}(\hat{X}_d) &= \big(1 + \tanh(A^i_s)\big) \odot \mathrm{Im}(X_d) + B^i_s. \end{aligned} \tag{8}$$

Then, they are combined to form the calibrated complex-valued dynamic state:

$$\hat{X}_d = \mathrm{Re}(\hat{X}_d) + \mathrm{i}\,\mathrm{Im}(\hat{X}_d). \tag{9}$$

This operation applies background-dependent scaling and shifting to the background-relative deviation. It calibrates the current dynamic state according to the temporal background.

The calibrated dynamic state is further transformed by background-conditioned spectral weights. We first introduce a learnable global weight vector $w_0 \in \mathbb{R}^L$, which is shared across all input windows. Its frequency-domain representation is obtained as

$$W_0 = \mathcal{F}(w_0) \in \mathbb{C}^F, \tag{10}$$

where W_0 serves as the global spectral weights and captures evolution patterns shared across different temporal backgrounds.

Given the temporal background X_s, the evolution modulator generates a complex-valued local spectral weight adjustment:

$$\begin{aligned} \Delta W_s &= f_\psi(X_s), \\ W_s &= W_0 + \eta \Delta W_s, \end{aligned} \tag{11}$$

where, f_ψ is a two-layer feed-forward network with a ReLU activation. The local adjustment ΔW_s adapts the global spectral weights to the temporal background of the current input window. The coefficient η controls the magnitude of this adjustment. The resulting W_s represents the background-conditioned spectral weights.

The calibrated dynamic state is then evolved as

$$\tilde{X}_d = \hat{X}_d \odot W_s. \tag{12}$$

Finally, the evolved dynamic representation is recombined with the temporal background and transformed back into the time domain:

$$X' = \mathcal{F}^{-1}\left(\tilde{X}_d + X_s\right). \tag{13}$$

Theoretical analyses of background-relative deviation and background-conditioned dynamic evolution are provided in Appendix D.

3.5 Projection

After the evolved spectral representation is mapped back to the time domain, TimeRift obtains the learned representation X'. We then use a lightweight projection head to project this representation into the target forecasting horizon.

$$Y = \mathrm{Linear}\left(\mathrm{ReLU}\left(\mathrm{Linear}(X')\right)\right). \tag{14}$$

4 Experiments

4.1 Experimental Setup

We conduct experiments on seven real-world benchmark datasets widely used in long-term time series forecasting. These datasets include ETTh1, ETTh2, ETTm1, ETTm2 Zhou et al. (2021), Weather, Electricity (ECL), and Traffic Wu et al. (2021). Following previous studies Qiu et al. (2026); Ma et al. (2026b); Yue et al. (2026); Nie et al. (2023), the ETT datasets are chronologically divided into training, validation, and test sets with a ratio of 6:2:2, whereas the other datasets are divided with a ratio of 7:1:2. Detailed experimental settings and details are provided in Appendix A.

To comprehensively evaluate the forecasting performance, we compare TimeRift with competitive baselines covering several representative modeling paradigms: (1) **MLP-based models**: including DLinear Zeng et al. (2023), and CycleNet Lin et al. (2024); (2) **Frequency-based models**: FITS Xu et al. (2024), and FilterTS Wang et al. (2025); (3) **Transformer-based models**: iTransformer Liu et al. (2024), and TQNet Lin et al. (2025); and (4) **Temporal information-based models**: GTR Cao et al. (2026), and TimeEmb Xia et al. (2026).

4.2 Main Results

Table 1 reports the forecasting results on seven real-world datasets. TimeRift achieves the best Friedman average rank of 1.46 among all compared methods. Several observations can be made as follows:

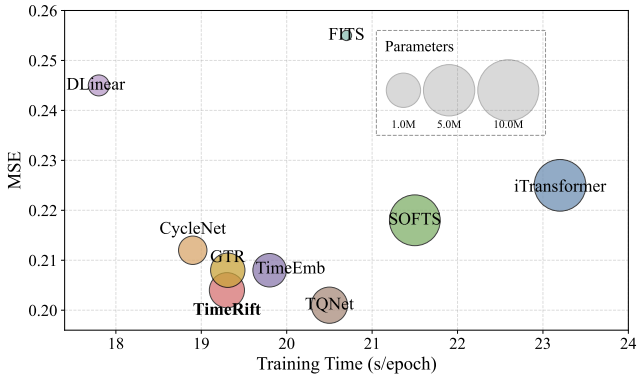


Figure 3: Comparison of prediction accuracy, parameter size, and training time on the Electricity dataset (look-back length $L = 96$, forecast horizon $T = 720$).

Table 1: Forecasting performance comparison on seven real-world datasets with prediction lengths $T \in \{96, 192, 336, 720\}$ and look-back window length $L = 96$. The Count row denotes the number of first-place results achieved by each method. The Rank row reports the results of the Friedman test. The best results are highlighted in **bold**, while the second-best results are underlined. Detailed results are provided in Appendix G.

Model	**TimeRift (ours)**		TimeEmb 2026		GTR 2026		TQNet 2025		FilterTS 2025		CycleNet 2024		iTransformer 2024		FITS 2024		DLinear 2023	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.422**	**0.423**	0.425	0.425	0.439	0.434	0.441	0.434	0.433	0.430	0.432	0.427	0.454	0.447	0.451	0.440	0.456	0.452
ETTh2	**0.358**	**0.387**	0.362	0.390	0.372	0.400	0.378	0.402	0.372	0.396	0.383	0.404	0.383	0.407	0.383	0.408	0.559	0.515
ETTm1	0.368	**0.384**	0.368	**0.384**	**0.367**	0.389	0.377	0.393	0.385	0.396	0.379	0.396	0.407	0.410	0.415	0.408	0.403	0.407
ETTm2	**0.263**	**0.307**	0.265	0.308	0.268	0.315	0.277	0.323	0.276	0.336	0.266	0.314	0.288	0.332	0.286	0.328	0.350	0.401
Weather	**0.236**	**0.261**	0.237	0.262	0.239	0.268	0.242	0.269	0.244	0.274	0.243	0.271	0.258	0.278	0.249	0.276	0.265	0.317
Electricity	**0.164**	**0.255**	0.167	0.260	0.166	0.260	**0.164**	0.259	0.180	0.271	0.168	0.259	0.178	0.270	0.217	0.295	0.212	0.300
Traffic	0.439	0.286	0.454	0.293	0.470	0.280	0.445	**0.276**	0.471	0.315	0.472	0.301	**0.428**	0.282	0.627	0.376	0.625	0.383
Count	**11**		1		1		2		0		0		1		0		0	
Rank	**1.46**		2.68		3.61		4.04		5.57		4.71		6.71		7.64		8.57	

(1) ***TimeRift achieves consistent improvements across diverse datasets.*** TimeRift achieves the best or tied-best results on the ETT series, Weather, and Electricity datasets. These datasets exhibit clear recurrent temporal regularities. Such regularities provide useful temporal references for interpreting local variations. On Traffic, iTransformer and TQNet achieve lower MSE and MAE, respectively. Traffic exhibits strong dependencies across variables. However, TimeRift mainly focuses on temporal background-dependent dynamics and does not explicitly model cross-variables interactions. This may limit its performance advantage on this dataset.

(2) ***TimeRift shows clear advantages over methods that explicitly exploit temporal regularities.*** TimeRift performs favorably against CycleNet, TimeEmb, and GTR. The key distinction lies in how recurrent temporal information is incorporated into the forecasting process. CycleNet captures recurring temporal patterns through learnable cycles and performs forecasting on the remaining residual component. TimeEmb models time-slot-specific spectral patterns and separates them from the dynamic component for forecasting. GTR retrieves aligned global periodic information and integrates it with local observations. In contrast, TimeRift uses the temporal background as an explicit reference to define the background-relative deviation and further conditions both its state calibration and spectral evolution. This enables similar local variations to be modeled differently under different temporal backgrounds.

(3) ***TimeRift achieves a favorable balance between forecasting accuracy and computational efficiency.*** To evaluate model efficiency, we compare the number of parameters, training time, and MSE on the Electricity dataset, as shown in Figure 3. TimeRift achieves an MSE of 0.204 with only 1.10M parameters and a training time of 19.3 s/epoch. Among the compared methods, TimeRift ranks second only to TQNet in terms of MSE. It also requires fewer parameters and less training time than TQNet. Compared with iTransformer, TimeRift uses approximately $4.7\times$ fewer param-

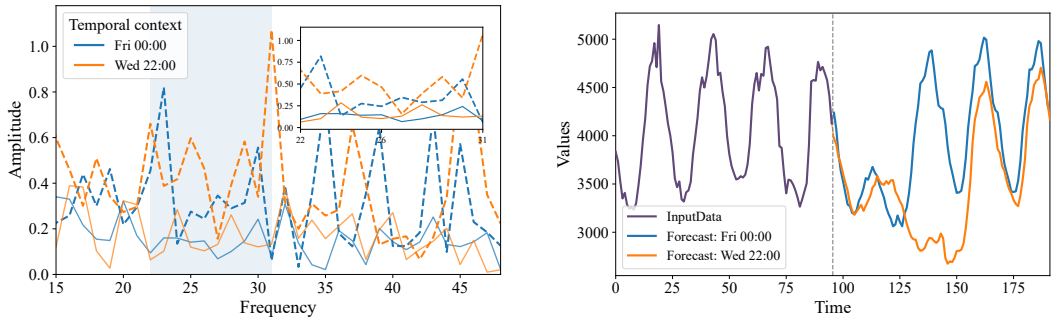
eters and reduces the training time by 16.8%. These results demonstrate that TimeRift achieves a competitive balance between forecasting accuracy and computational efficiency.

4.3 Context-Dependent Evolution Visualization

To examine how similar local observations behave under different temporal backgrounds, we conduct two complementary visual analyses on the Electricity dataset. We select two non-overlapping historical windows from different weekday-hour contexts. The two windows are highly similar in the normalized input space.

Background-relative deviation. Figure 4(a) compares the frequency-domain representations of two windows and their corresponding background-relative deviations. The solid lines denote the input spectra $|\hat{X}|$, while the dashed lines denote $|X_d|$, where $X_d = \hat{X} - X_s$. The two input spectra exhibit similar local structures. However, after subtracting their corresponding temporal backgrounds, the resulting deviations become more distinguishable. This shows that similar observations can yield different background-relative deviations.

Context-conditioned evolution. Figure 4(b) further examines how temporal background affects the forecasting behavior of TimeRift. We feed the same historical window into the trained model twice and change only its weekday-hour context. Although the historical input is identical, the resulting forecast trajectories gradually diverge after the forecasting origin. This demonstrates that the forecasting behavior of TimeRift varies with the temporal background rather than being fixed for the same local observation. Collectively, the two visualizations show how temporal background is used in TimeRift. It provides a reference for constructing the background-relative deviation and further influences its subsequent dynamic evolution. Whether the matched temporal background is important for forecasting is further examined through controlled intervention in Section 4.4. Additional representation-space analysis is provided in Appendix E.



(a) (b)

Figure 4: Visualization on the Electricity dataset with $L = 96$ and $T = 96$. (a) Frequency-domain representations and background-relative deviations of two similar historical windows from different weekday-hour contexts. (b) Forecasts for the same historical input under different weekday-hour contexts. The vertical dashed line marks the forecasting origin.

4.4 Temporal-Background Intervention on Dynamic Evolution

As shown in Table 2, using the temporally aligned conditioning background consistently yields the lowest MSE across all prediction lengths. Even a one-hour shift increases the MSE by up to 12.05%, while larger shifts lead to substantially greater degradation. Under a $\pm$12-hour shift, the MSE increases by 119.05%–223.81%. During the intervention, the historical input, background-relative deviation, and reconstruction background are kept fixed. Only the temporal background supplied to state calibration and spectral evolution is shifted. Therefore, the resulting performance degradation isolates the effect of temporal-background mismatch during dynamic evolution. These results show that temporal-background alignment is relevant to forecasting performance, rather than only affecting the learned internal representations. Stage-wise intervention results are provided in Appendix E.

Table 2: Temporal-background intervention results on the Traffic dataset. Results for $\pm k$ hours are averaged over both shift directions. Values in parentheses denote the relative MSE degradation from the correct temporal background.

Prediction Length	Temporal-Background Shift				
	Correct (0 h)	±1 h	±3 h	±6 h	±12 h
96	**0.411**	0.447 (+8.94%)	0.555 (+35.04%)	0.711 (+73.00%)	0.900 (+119.05%)
192	**0.427**	0.474 (+11.10%)	0.626 (+46.80%)	0.866 (+102.93%)	1.206 (+182.56%)
336	**0.442**	0.495 (+12.05%)	0.664 (+50.31%)	0.939 (+112.53%)	1.381 (+212.42%)
720	**0.475**	0.529 (+11.44%)	0.789 (+66.21%)	1.070 (+125.26%)	1.538 (+223.81%)

Table 3: Ablation study of TimeRift.

Method	ETTh1		ETTm1		Weather		Electricity		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeRift	**0.422**	**0.423**	**0.368**	**0.384**	**0.236**	**0.261**	**0.164**	**0.255**	**0.439**	**0.286**
w/o Background-Relative deviation	0.434	0.429	0.377	0.389	0.241	0.266	0.171	0.260	0.448	0.294
w/o Background-Conditioned State Calibration	0.427	0.427	0.376	0.388	0.238	0.263	0.167	0.258	0.443	0.289
w/o Background-Conditioned Spectral Evolution	0.447	0.436	0.372	0.386	0.245	0.269	0.166	0.256	0.444	0.287
w/o Context-Dependent Dynamic Evolution	0.453	0.439	0.383	0.391	0.263	0.282	0.195	0.276	0.494	0.312

4.5 Ablation Studies

To evaluate the contribution of each component in TimeRift, we conduct ablation studies on five representative datasets, as reported in Table 3. The full model consistently achieves the best performance across all datasets and metrics. Removing the background-relative deviation degrades forecasting accuracy on all five datasets, confirming the benefit of representing local dynamics relative to their recurrent temporal backgrounds. Removing either background-conditioned state calibration or spectral evolution also consistently leads to performance degradation, indicating that both stages contribute to context-dependent dynamic evolution. Their relative effects vary across datasets. Removing the entire context-dependent dynamic evolution causes the largest overall degradation. These results confirm the importance of both background-relative deviation and context-dependent dynamic evolution to forecasting performance. Additional ablations are provided in Appendix F.

5 Conclusion

Motivated by the observation that similar local variations may evolve differently under different temporal backgrounds, we propose TimeRift to explicitly model context-dependent dynamic evolution for MTSF. Specifically, TimeRift learns recurrent temporal backgrounds in the spectral domain and models local variations as background-relative deviations. It then models these deviations through a context-dependent dynamic evolution mechanism. State calibration first adjusts the deviation according to the temporal background. Spectral evolution then combines global spectral weights with background-conditioned local weights to evolve the calibrated dynamic state. Experiments on seven real-world benchmarks demonstrate strong forecasting performance. Efficiency analysis also shows a favorable trade-off between forecasting accuracy and computational cost. In the future, we will explore richer forms of temporal context and extend context-conditioned dynamic modeling to broader time series analysis tasks.

AI USE STATEMENT

In this work, we used generative AI tools only to aid and polish the writing. Specifically, these tools were used to improve the grammar, clarity, and readability of author-written text. We did not use generative AI tools for other research tasks requiring disclosure. All AI-assisted edits were reviewed by the authors to ensure technical accuracy and preservation of the intended meaning. We take full responsibility for the final content of this work.

REFERENCES

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

Jialin Chen, Ziyu Zhao, Gaukhar Nurbek, Aosong Feng, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Trace: Grounding time series in context for multimodal embedding and retrieval. *Advances in Neural Information Processing Systems*, 38:2440–2472, 2026.

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 341–352, 2024.

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.

Archibald Fraikin, Adrien Bennetot, and Stéphanie Allassonnière. T-rep: Representation learning for time series using time-embeddings. In *International Conference on Learning Representations*, volume 2024, pp. 31285–31305, 2024.

Qi Guo, Baolin Zhao, Mingchen Song, and Guoqiang Zhong. A survey of deep learning for time series forecasting: Taxonomy, analysis and future directions. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Lu Han, Yu Liu, Lan Li, Qiwen Deng, Jian Jiang, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, et al. Unica: Unified covariate adaptation for time series foundation model. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yushan Jiang, Kanghui Ning, Zijie Pan, Xuyang Shen, Jingchao Ni, Wenchao Yu, Anderson Schneider, Haifeng Chen, Yuriy Nevmyvaka, and Dongjin Song. Multi-modal time series analysis: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6043–6053, 2025.

Zhengyu Li, Xiangfei Qiu, Yuhan Zhu, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. Gcgnet: Graph-consistent generative network for time series forecasting with exogenous variables. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *International Conference on Machine Learning*, pp. 37797–37814. PMLR, 2025.

Bowen Liu, Haijian Lai, Chan-Tong Lam, Junhao Dong, Benjamin Ng, Wei Ke, and Sio-Kei Im. Tsnn: A non-parametric and interpretable framework for traffic time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

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

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

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*, volume 2024, pp. 11116–11140, 2024.

Zihao Lv, Yi Song, Yu Xue, Shijie Xu, Chunlin He, and Liming Xu. State estimation of lithium-ion batteries with state space model. *Engineering Applications of Artificial Intelligence*, 159:111463, 2025.

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. Phat: Modeling period heterogeneity for multivariate time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 135659–135685, 2026a.

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. *Advances in Neural Information Processing Systems*, 38:84871–84902, 2026b.

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 *Forty-second International Conference on Machine Learning*, 2025.

Ronghao Ni, Zinan Lin, Shuaiqi Wang, and Giulia Fanti. Mixture-of-linear-experts for long-term time series forecasting. In *International conference on artificial intelligence and statistics*, pp. 4672–4680. PMLR, 2024.

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

Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. Dbloss: Decomposition-based loss function for time series forecasting. *Advances in Neural Information Processing Systems*, 38:27741–27768, 2026.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 4454–4458, 2022.

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.

Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. *Advances in Neural Information Processing Systems*, 38:91810–91844, 2026.

Cheng Tan, Wenbin Zhang, and Mingjing Du. Dpia: Dynamic periodic interaction and adaptation for multivariate time series forecasting. *Pattern Recognition*, pp. 114310, 2026.

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. *Advances in Neural Information Processing Systems*, 37:22206–22232, 2024a.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024b.

Yulong Wang, Yushuo Liu, Xiaoyi Duan, and Kai Wang. Filterts: Comprehensive frequency filtering for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21375–21383, 2025.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024c.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Learning deep time-index models for time series forecasting. In *International Conference on Machine Learning*, pp. 37217–37237. PMLR, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Mingyuan Xia, Chunxu Zhang, Zijian Zhang, Hao Miao, Qidong Liu, Yuanshao Zhu, and Bo Yang. Timeemb: A lightweight static-dynamic disentanglement framework for time series forecasting. *Advances in Neural Information Processing Systems*, 38:76227–76258, 2026.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Yaling Xun, Jiahui Yan, Haifeng Yang, Jianghui Cai, and Xing Wang. Ptqnet: Periodic-temporal query network for long-term multivariate time series forecasting. *Information Processing & Management*, 63(7):104785, 2026.

Hechen Yang, Yu Yao, Licheng Yang, and Wei Yang. Rl-acid: Reinforcement learning-optimized adaptive causal discovery for robust anomaly detection in industrial systems. *IEEE Transactions on Industrial Informatics*, 2026.

Junjie Ye, Chengli Zhou, Xiaojun Zhou, Yaqun Huang, and Chunna Zhao. Cvacl-ma: Comprehensive variate analysis and collaborative learning with multi-adapter for multivariate time series forecasting. *Pattern Recognition*, 169:111936, 2026.

Junhao Yu, Fuyuan Xiao, Yi Zhang, Zehong Cao, and Chin-Teng Lin. A cdgfn-based quantum multisource information fusion with its application in time series classification. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *Advances in Neural Information Processing Systems*, 38:60573–60632, 2026.

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.

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.

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, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Implementation Details

A.1 Dataset Descriptions

We conduct experiments on seven widely used real-world datasets to evaluate the forecasting performance of TimeRift, including ETT, Electricity, Traffic, and Weather. Specifically, (1) **ETT**[1] consists of electricity transformer temperature records collected from July 2016 to July 2018 and contains seven variables. It includes four subsets with different sampling frequencies: **ETTh1** and **ETTh2** are recorded hourly, whereas **ETTm1** and **ETTm2** are sampled every 15 minutes. (2) **Electricity**[2] records the hourly electricity consumption of 321 clients, providing a large-scale multivariate benchmark for modeling electricity usage patterns. (3) **Traffic**[3] contains hourly road occupancy measurements collected by 862 sensors on freeways in the San Francisco Bay Area from January 2015 to December 2016. (4) **Weather**[4] contains 21 meteorological variables, such as air temperature, pressure, and humidity, recorded every 10 minutes at the Weather Station of the Max Planck Biogeochemistry Institute in 2020. These datasets cover diverse application domains, sampling frequencies, and numbers of variables, and are widely adopted as benchmarks for multivariate time series forecasting. Detailed statistics of all datasets are summarized in Table 4.

Table 4: The statistical details of the datasets.

Datasets	Variates	Timesteps	Granularity	Domain
ETTh1	7	14400	1 hour	Electricity
ETTh2	7	14400	1 hour	Electricity
ETTm1	7	57600	15 mins	Electricity
ETTm2	7	57600	15 mins	Electricity
Weather	21	52696	10 mins	Weather
Electricity	321	26304	1 hour	Electricity
Traffic	862	17544	1 hour	Transportation

We follow the standard data processing protocol widely adopted in multivariate time series forecasting. All datasets are chronologically divided into training, validation, and test sets to avoid data leakage. Specifically, the four ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2) are split with a ratio of 6:2:2, while Electricity, Traffic, and Weather are split with a ratio of 7:1:2. For all seven datasets, we fix the look-back length to 96 and evaluate the forecasting performance under four prediction horizons, i.e., $\{96, 192, 336, 720\}$.

A.2 Baselines

We compare TimeRift with 9 representative and state-of-the-art models to comprehensively evaluate its forecasting performance, including MLP-, Frequency-, Transformer- and Temporal information-based models. The details of these baselines are as follows:

TimeEmb introduces a lightweight static-dynamic disentanglement framework that separates time series into time-invariant and time-varying components. It captures persistent patterns through a global embedding module and models dynamic variations using an efficient frequency-domain filtering mechanism. The detailed implementation is available in https://github.com/showmeon/TimeEmb.

GTR learns a global temporal representation over the entire cycle. For each input sequence, it retrieves the relevant global segment according to its temporal position. The retrieved global context is then jointly modeled with the local historical observations, allowing the model to capture periodic patterns beyond the limited look-back window. The detailed implementation is available in https://github.com/macovaseas/GTR.

[1] https://github.com/zhouhaoyi/ETDataset

[2] https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014

[3] https://pems.dot.ca.gov

[4] https://www.bgc-jena.mpg.de/wetter

TQNet introduces a Temporal Query mechanism for modeling multivariate correlations. It uses periodically shifted learnable vectors as queries, while keys and values are derived from the input series. A lightweight attention architecture then combines global inter-variable patterns with sample-specific information. The detailed implementation is available in https://github.com/ACAT-SCUT/TQNet.

FilterTS develops a frequency-domain forecasting model. It uses dynamic cross-variable filtering and static global filtering. The former exploits shared frequency information across variables, while the latter captures stable frequency components from the training data. This design enables efficient modeling of multivariate temporal patterns. The detailed implementation is available in https://github.com/wyl010607/FilterTS.

CycleNet introduces Residual Cycle Forecasting to model periodic patterns. It represents these patterns using learnable recurrent cycles. The modeled periodic patterns are removed from the input series. Forecasting is then performed on the residual components. The detailed implementation is available in https://github.com/ACAT-SCUT/CycleNet.

iTransformer adopts an inverted Transformer architecture that treats individual variables as tokens rather than temporal points. Self-attention is applied across variate tokens to capture multivariate correlations, while feed-forward networks learn nonlinear representations for individual variables. The detailed implementation is available in https://github.com/thuml/iTransformer.

SOFTS proposes an efficient MLP-based forecasting architecture with a STar Aggregate-Redistribute module. It first aggregates information from multiple variables into a centralized core representation. The global information is then redistributed to individual series. This design enables efficient cross-variable interaction with linear complexity. The detailed implementation is available in https://github.com/Secilia-Cxy/SOFTS.

FITS models time series through interpolation in the complex frequency domain. It retains informative low-frequency components and filters less influential high-frequency components. This results in a lightweight forecasting architecture with approximately 10K parameters. The detailed implementation is available in https://github.com/VEWOXIC/FITS.

DLinear adopts a simple linear architecture based on series decomposition. It separates the input sequence into trend and seasonal components. Linear mappings are then applied to the decomposed components for forecasting. Despite its simplicity, DLinear provides a strong and computationally efficient baseline. The detailed implementation is available in https://github.com/cure-lab/LTSF-Linear.

A.3 Implementation Details

To ensure a fair comparison, we adopt a unified forecasting setting for all models. The look-back length is fixed as $L = 96$, and the prediction length $T \in \{96, 192, 336, 720\}$ for all datasets. For all baselines, we use their official implementations whenever available and determine the model-specific hyperparameters according to the original papers or validation performance. Detailed hyperparameter settings are provided in the accompanying code. Forecasting performance is evaluated using mean squared error (MSE) and mean absolute error (MAE). TimeRift is trained for at most 100 epochs with early stopping, where the patience is set to 5 according to the validation loss. The learning rate is selected from $\{2 \times 10^{-3}, 3 \times 10^{-3}, 4 \times 10^{-3}, 5 \times 10^{-3}, 6 \times 10^{-3}\}$ according to the validation performance. The hidden dimension d_{model} is fixed to 512 for all datasets. The adjustment coefficient η is fixed to 0.02 for all datasets. All experiments are implemented in PyTorch 2.4.1 and conducted on a single NVIDIA RTX 4090 24GB GPU. The pseudo-code of TimeRift is provided in Algorithm 1. We further report its forecasting performance over five independent runs with different random seeds in Table 10. The consistently small standard deviations indicate that TimeRift exhibits stable performance across different random seeds.

A.4 Analysis of Future Trajectory Differences across Temporal Backgrounds

The motivating example in Figure 1 shows that similar local variations can be followed by different future trajectories under different temporal backgrounds. We further investigate whether this phenomenon is observed more broadly across datasets.

Algorithm 1 TimeRift-Overall Architecture.

Require: Historical sequence X; temporal indices (τ_h, τ_d)
Ensure: Forecast Y

$\widetilde{X} \leftarrow \mathrm{RevIN}(X)$
$\hat{X} \leftarrow \mathcal{F}(\widetilde{X})$
$p \leftarrow \tau_h \bmod N_h$, $q \leftarrow \tau_d \bmod N_d$
$X_s \leftarrow E^h_{p,:,:} + E^d_{q,:,:}$
$X_d \leftarrow \hat{X} - X_s$
$(A^r_s, A^i_s, B^r_s, B^i_s) \leftarrow f_\theta(X_s)$
$\mathrm{Re}(\hat{X}_d) \leftarrow (1 + \tanh(A^r_s)) \odot \mathrm{Re}(X_d) + B^r_s$
$\mathrm{Im}(\hat{X}_d) \leftarrow (1 + \tanh(A^i_s)) \odot \mathrm{Im}(X_d) + B^i_s$
$\hat{X}_d \leftarrow \mathrm{Re}(\hat{X}_d) + \mathrm{i}\,\mathrm{Im}(\hat{X}_d)$
$W_0 \leftarrow \mathcal{F}(w_0)$
$\Delta W_s \leftarrow f_\psi(X_s)$
$W_s \leftarrow W_0 + \eta \Delta W_s$
$\widetilde{X}_d \leftarrow \hat{X}_d \odot W_s$
$X' \leftarrow \mathcal{F}^{-1}(\widetilde{X}_d + X_s)$
$\widetilde{Y} \leftarrow \mathrm{Linear}(\mathrm{ReLU}(\mathrm{Linear}(X')))$
$Y \leftarrow \mathrm{RevIN}^{-1}(\widetilde{Y})$
return Y

Table 5: Analysis of historical and future differences across temporal backgrounds with $L = T = 96$. Same context and different context pairs are matched according to their historical RMS differences. Historical and future differences are averaged over three independent pair samplings. Relative Increase denotes the increase in future difference from same context pairs to different context pairs, and is reported as mean±standard deviation.

Dataset	Historical Difference		Future Difference		Relative Increase
	Same Context	Different Context	Same Context	Different Context	
ETTh1	0.642	0.642	0.853	0.899	$+5.4\% \pm 1.1\%$
ETTh2	0.339	0.339	0.513	0.519	$+1.2\% \pm 1.3\%$
ETTm1	0.457	0.456	0.739	0.913	$+23.5\% \pm 3.2\%$
ETTm2	0.253	0.253	0.443	0.498	$+12.5\% \pm 2.1\%$
Weather	0.191	0.191	0.444	0.484	$+9.0\% \pm 2.5\%$
Electricity	0.492	0.491	0.590	0.647	$+9.8\% \pm 2.5\%$
Traffic	0.812	0.811	0.929	1.013	$+9.1\% \pm 0.7\%$

For two historical windows X_i and X_j and their corresponding future sequences Y_i and Y_j, we measure their differences using the root-mean-square (RMS) distance:

$$d_H(i,j) = \sqrt{\frac{1}{LC} \|X_i - X_j\|_F^2}, \tag{15}$$

$$d_F(i,j) = \sqrt{\frac{1}{TC} \|Y_i - Y_j\|_F^2}. \tag{16}$$

We divide window pairs according to whether they share the same temporal background. The recurrent temporal factors are defined independently of the sampling interval and characterize recurring temporal contexts rather than individual sampling positions. For ETTh1, ETTh2, ETTm1, ETTm2, and Weather, the temporal background is determined by the hour of the forecasting origin. For Electricity and Traffic, both the hour and weekday are used. To control for historical similarity, each different-context pair is matched with a same-context pair having the closest historical RMS difference. Overlapping window pairs are excluded. All experiments use $L = T = 96$ and are conducted on the test set.

As shown in Table 5, the historical differences between same context and different context pairs are closely matched on all datasets. However, different context pairs exhibit larger future differences on most datasets. The effect is particularly clear on ETTm1 and ETTm2, where the future difference increases by 23.5% and 12.5%, respectively. ETTh2 shows only a marginal difference. Among historically similar window pairs, those from different recurrent temporal backgrounds exhibit larger future trajectory differences on most datasets. This provides dataset-level evidence that similar local variations can correspond to different future trajectories under different temporal backgrounds.

B Hyperparameter Sensitivity

Sensitivity to look-back length. We investigate the effect of the historical input length L on the ETTm1 and Electricity datasets. As shown in Figure 5, most baselines benefit noticeably as the look-back length increases from short to moderate ranges. In contrast, TimeRift already achieves highly competitive performance with $L = 96$. As L increases, TimeRift maintains relatively stable performance across different historical input lengths. Overall, these results demonstrate that TimeRift remains robust across different look-back lengths. It also maintains strong forecasting performance even with limited historical context.

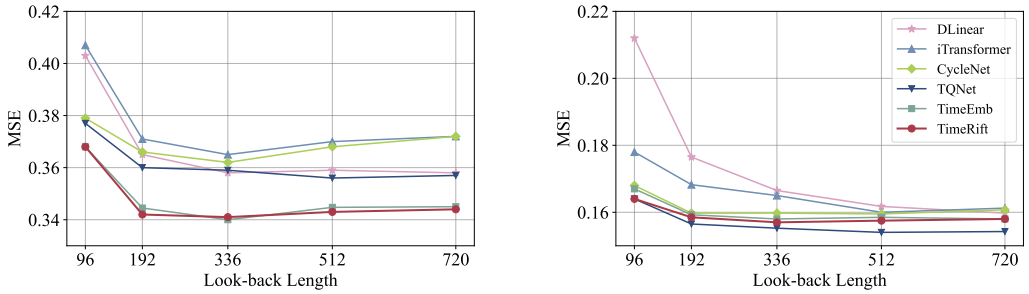


Figure 5: Sensitivity analysis of the look-back length on the (a) ETTm1 and (b) Electricity datasets.

Effect of hourly context size. We further investigate the sensitivity of TimeRift to the hourly context size N_h, i.e., the number of recurrent categories for the hour-of-day factor, by varying N_h over $\{4, 6, 8, 12, 24\}$. As shown in Figure 6, the forecasting performance remains relatively stable across different settings, while larger N_h generally yields lower MSE. In particular, $N_h = 24$ achieves the best or tied-best performance on all four datasets. With a smaller N_h, multiple hour indices are mapped to the same recurrent category and share the corresponding spectral prototype. A larger N_h reduces the sharing of spectral prototypes across hour-of-day indices, enabling a finer-grained characterization of hourly temporal backgrounds.

C Computational Complexity Analysis

We analyze the computational complexity of the core components of TimeRift, including temporal background construction, and context-dependent dynamic evolution.

Temporal Background Construction. The temporal background is constructed through lightweight temporal embedding lookup and element-wise addition. Given the recurrent temporal indices, the corresponding embeddings are retrieved from the temporal embedding banks and combined to obtain the background representation $X_s \in \mathbb{R}^{C \times F}$, where $F = \lfloor L/2 \rfloor + 1$ denotes the frequency dimension. The lookup and addition operations require $\mathcal{O}(C \times F)$ complexity.

Context-Dependent Dynamic Evolution. The two stages are implemented with a lightweight context encoder and evolution modulator, respectively. The context encoder maps X_s to the scaling factors and shift terms used to calibrate X_d. The evolution modulator maps X_s to local spectral weights ΔW_s, which are combined with the global weights W_0 for spectral evolution. Each con-

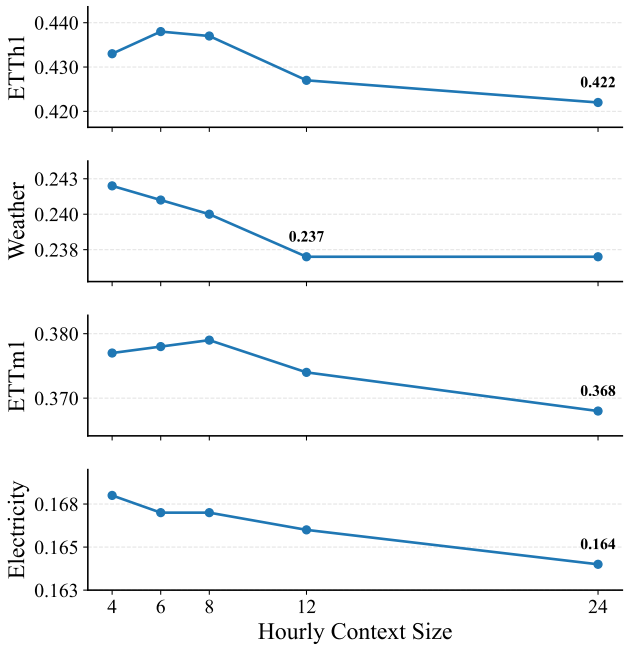


Figure 6: Sensitivity analysis with respect to the hourly context size.

text mapping requires $\mathcal{O}(C \times F \times D)$ operations, while the subsequent element-wise calibration and spectral evolution require only $\mathcal{O}(C \times F)$. Therefore, the overall complexity of this module is $\mathcal{O}(C \times F \times D)$.

Combining the Fourier transform, temporal background construction, and context-dependent dynamic evolution. The computational complexity of the core modeling components of TimeRift is

$$\mathcal{O}(CL \log L + CFD).$$

Since $F = \lfloor L/2 \rfloor + 1$, the complexity can be further expressed as

$$\mathcal{O}(CL(\log L + D)).$$

The computational cost of TimeRift scales linearly with C and near-linearly with L. This suggests good scalability to long-sequence and high-dimensional forecasting settings.

To complement the theoretical analysis, we further evaluate the practical efficiency of different models in Table 6. TimeRift achieves the best MSE of 0.433 and the lowest peak memory usage of 39.26 MB, while requiring 3.61 s/epoch and 0.63M parameters. These results demonstrate a favorable balance between forecasting accuracy and computational efficiency.

D Theoretical Analysis

D.1 Spectral Interpretation of Background-Relative Deviation

TimeRift characterizes each historical window relative to its recurrent temporal background in the frequency domain. Here, we provide a simple interpretation of the resulting background-relative deviation.

Proposition 1. *Let $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively. Given a historical window X with $\hat{X} = \mathcal{F}(X)$. Let $X_s(\tau)$ denote the spectral temporal background associated with temporal context τ. Define*

$$X_d(\tau) = \hat{X} - X_s(\tau). \tag{17}$$

Let the corresponding time-domain background be

$$B_\tau = \mathcal{F}^{-1}(X_s(\tau)). \tag{18}$$

Table 6: Comparison of forecasting accuracy and computational efficiency on ETTm1 with $L = 96$ and $T = 720$.

Model	Training Time(s/epoch)	MSE	Max Memory(MB)	Parameters(M)
CycleNet	**2.21**	0.447	70.67	**0.05**
iTransformer	18.53	0.491	203.01	3.58
SOFTS	42.84	0.466	160.68	1.13
TQNet	<u>2.64</u>	0.452	75.81	0.98
GTR	3.55	<u>0.435</u>	94.13	0.95
TimeEmb	2.80	<u>0.435</u>	<u>55.64</u>	<u>0.43</u>
TimeRift	3.61	**0.433**	**39.26**	0.63

Then

$$X_d(\tau) = \mathcal{F}(X - B_\tau)\,. \tag{19}$$

Proof. By the linearity of the Fourier transform,

$$\begin{aligned} \mathcal{F}(X - B_\tau) &= \mathcal{F}(X) - \mathcal{F}(B_\tau) \\ &= \hat{X} - X_s(\tau) \\ &= X_d(\tau). \end{aligned} \tag{20}$$

This completes the proof. □

D.2 Expressiveness of Background-Conditioned Dynamic Evolution

We analyze the expressive effect introduced by background-conditioned dynamic evolution. Let $\hat{X}$ denote the spectral representation of a local historical window and X_s denote its temporal background. TimeRift first constructs the background-relative deviation

$$X_d = \hat{X} - X_s. \tag{21}$$

The spectral representation produced by the core evolution module before the inverse Fourier transform can be abstracted as

$$\Psi_{\mathrm{tr}}(\hat{X}, X_s) = X_s + W_s \left[A_s(\hat{X} - X_s) + B_s \right], \tag{22}$$

where A_s and B_s denote the background-conditioned scaling and shift terms used for dynamic state calibration, and

$$W_s = W_0 + \eta \Delta W(X_s) \tag{23}$$

denotes the background-conditioned spectral evolution weights. For a fixed X_s, the calibration mapping is affine with respect to the real-valued representation of the complex spectrum, while the spectral evolution mapping is linear with respect to the calibrated dynamic state.

Theorem 1. *For two local spectral representations $\hat{X}_a$ and $\hat{X}_b$, define*

$$\Delta X = \hat{X}_a - \hat{X}_b. \tag{24}$$

The contrast between their evolved representations under temporal background X_s is

$$\mathcal{C}_{\mathrm{tr}}(\hat{X}_a, \hat{X}_b; X_s) = \mathcal{T}(X_s)\Delta X, \tag{25}$$

where

$$\mathcal{T}(X_s) = W_s A_s \tag{26}$$

is the effective background-conditioned evolution operator.

Consider the corresponding background-independent evolution subclass, in which A_s and W_s are constrained to be independent of X_s. For any model in this subclass, the contrast transformation

is invariant to the temporal background. In contrast, if there exist two temporal backgrounds $X_s^{(1)}$ and $X_s^{(2)}$ and a nonzero local contrast ΔX such that

$$\left[\mathcal{T}(X_s^{(1)}) - \mathcal{T}(X_s^{(2)})\right] \Delta X \neq 0, \tag{27}$$

then the corresponding TimeRift mapping cannot be represented by any background-independent evolution operator.

Therefore, background-independent evolution is a special case of TimeRift, while TimeRift further allows the transformation of local dynamic variations to depend on the temporal background.

Proof. According to Eq. 22, the evolved representations of $\hat{X}_a$ and $\hat{X}_b$ under the same temporal background X_s are

$$\Psi_{\mathrm{tr}}(\hat{X}_a, X_s) = X_s + W_s \left[A_s(\hat{X}_a - X_s) + B_s\right] \tag{28}$$

and

$$\Psi_{\mathrm{tr}}(\hat{X}_b, X_s) = X_s + W_s \left[A_s(\hat{X}_b - X_s) + B_s\right]. \tag{29}$$

Therefore, their evolution contrast is

$$\begin{aligned} \mathcal{C}_{\mathrm{tr}}(\hat{X}_a, \hat{X}_b; X_s) &= \Psi_{\mathrm{tr}}(\hat{X}_a, X_s) - \Psi_{\mathrm{tr}}(\hat{X}_b, X_s) \\ &= W_s A_s(\hat{X}_a - \hat{X}_b) \\ &= \mathcal{T}(X_s)\Delta X. \end{aligned} \tag{30}$$

Notice that both the additive temporal background X_s and the background-dependent shift B_s cancel in the contrast. Hence, the transformation of a local variation is determined by the effective operator $\mathcal{T}(X_s)$.

Now consider a background-independent counterpart for which

$$A_s = A, \qquad W_s = W, \quad \forall X_s. \tag{31}$$

Therefore, its effective evolution operator is

$$\mathcal{T}_0 = WA, \tag{32}$$

and its contrast satisfies

$$\mathcal{C}_{\mathrm{ind}}(\hat{X}_a, \hat{X}_b; X_s) = \mathcal{T}_0 \Delta X, \tag{33}$$

which is independent of X_s. Consequently, the same local contrast must be transformed identically under different temporal backgrounds.

The background-independent case can be recovered by TimeRift by constraining its context-conditioned calibration and evolution operators to remain constant across temporal backgrounds. Thus, the background-independent model is a special case of TimeRift.

On the other hand, suppose that Eq. 27 holds for some $X_s^{(1)}$, $X_s^{(2)}$, and $\Delta X \neq 0$. From Eq. 30,

$$\begin{aligned} &\mathcal{C}_{\mathrm{tr}}(\hat{X}_a, \hat{X}_b; X_s^{(1)}) - \mathcal{C}_{\mathrm{tr}}(\hat{X}_a, \hat{X}_b; X_s^{(2)}) \\ &\quad = \left[\mathcal{T}(X_s^{(1)}) - \mathcal{T}(X_s^{(2)})\right] \Delta X \neq 0. \end{aligned} \tag{34}$$

Therefore, the same local contrast is transformed differently under the two temporal backgrounds. Such a mapping cannot be represented by a background-independent operator, whose contrast transformation is invariant to X_s.

Hence, TimeRift contains the background-independent evolution mechanism as a special case while admitting an additional class of background-dependent dynamic transformations. This completes the proof. □

E Analysis of Context-Conditioned Dynamic Evolution

To further examine how temporal background is learned and used in TimeRift, we analyze the learned background prototypes and the intermediate dynamic representations produced during context-conditioned evolution.

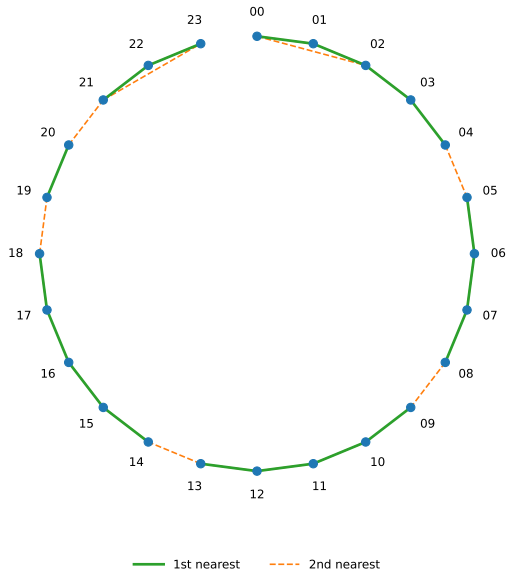


Figure 7: Temporal organization of the learned hourly backgrounds on the Electricity dataset with $L = 96$ and $T = 96$. The 24 hourly prototypes are arranged according to their hour-of-day positions. Solid and dashed connections denote the first and second nearest neighbors in the learned background space, respectively. Connections represent learned neighborhood relations rather than chronological adjacency.

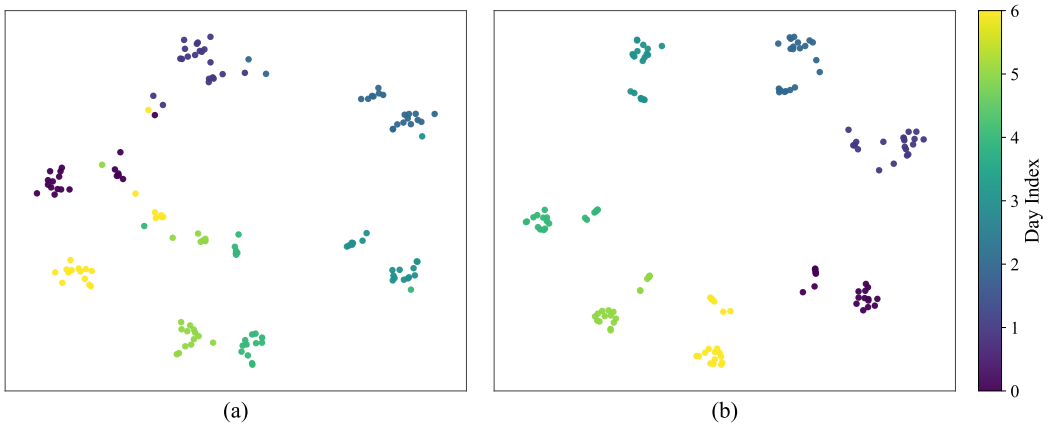


Figure 8: T-SNE visualization of the dynamic representations learned by TimeRift on the Traffic test set. The forecasting-origin hour is fixed at 18:00, and samples are color-coded according to the day of the week, ranging from 0 (Monday) to 6 (Sunday). (a) Background-relative deviation X_d. (b) Context-conditioned evolved representation X'.

E.1 Temporal Organization of Learned Backgrounds

We first examine the organization of the learned hourly background prototypes on the Electricity dataset. For two hourly prototypes $E^h_{i,:,:}$ and $E^h_{j,:,:}$, we measure their distance as

$$d(i,j) = \sqrt{\frac{1}{CF}\left\|E^h_{i,:,:} - E^h_{j,:,:}\right\|^2_F}. \tag{35}$$

For each hour, we then identify its first and second nearest prototypes in the learned background space. Figure 7 arranges the 24 prototypes according to their hour-of-day positions. All 24 proto-

Table 7: Quantitative representation-space analysis on the Traffic dataset. The forecasting origin is fixed at 18:00. Higher values indicate greater within-weekday compactness and between-weekday separation.

T	Representation	Silhouette ↑	Fisher Ratio ↑	Separation Ratio ↑
96	X_d	0.276	1.057	2.841
	Evolved Representation	**0.493**	**1.650**	**4.565**
192	X_d	0.267	1.008	2.720
	Evolved Representation	**0.448**	**1.365**	**3.878**
336	X_d	0.254	0.972	2.621
	Evolved Representation	**0.447**	**1.476**	**4.099**
720	X_d	0.262	1.025	2.755
	Evolved Representation	**0.599**	**2.211**	**5.772**
Avg.	X_d	0.265	1.016	2.734
	Evolved Representation	**0.497**	**1.675**	**4.579**

types have an adjacent hour as their nearest neighbor. Among the first two nearest neighbors, 46 of 48 correspond to adjacent hours, and all 48 remain within ± 2 hours. These results show that the learned hourly backgrounds exhibit a clear local temporal organization.

E.2 Representation-Space Analysis

To complement the case based analysis in Figure 4, we further examine the organization of the intermediate dynamic representations over the Traffic test set. We select test windows whose forecasting origins are fixed at 18:00 to exclude variations associated with different hours of the day. Each sample is color-coded according to its weekday index, ranging from 0 (Monday) to 6 (Sunday).

Background-relative deviations. Figure 8(a) visualizes the background-relative deviation X_d. The samples form several local neighborhoods, while partial intermixing remains among different weekday contexts. Although X_d represents each sample relative to its corresponding temporal background, it still reflects the dynamic characteristics of individual samples. Consequently, samples from the same weekday remain relatively dispersed, with partial overlap across different weekday contexts.

Context-conditioned evolved representations. Figure 8(b) visualizes the dynamic representation after background-conditioned state calibration and spectral evolution. Compared with X_d, samples associated with the same weekday exhibit more coherent local organization. Different weekday contexts also show less intermixing. The forecasting origin is fixed at 18:00. Therefore, samples from the same weekday share the same temporal context. They are conditioned by the same temporal background during state calibration and spectral evolution. The resulting context-related organization is consistent with the role of temporal background in conditioning the evolution of background-relative deviations.

Quantitative Analysis. To complement the representation-space visualization, we further quantify the organization of the learned dynamic representations on the Traffic dataset. For a controlled comparison across prediction horizons, we use the same set of test windows for all horizons. For evaluation, we concatenate the real and imaginary parts of each representation and apply ℓ_2 normalization. We evaluate the resulting representation space using the cosine Silhouette score, Fisher ratio, and separation ratio. The Fisher ratio measures between-context scatter relative to within-context scatter. The separation ratio measures the mean inter-context centroid distance relative to the mean within-context distance. Higher values indicate clearer context-related organization. As shown in Table 7, the evolved representation consistently achieves higher scores than X_d across all prediction horizons and metrics. These results quantitatively corroborate the representation-space visualization. This shows that context-dependent dynamic evolution produces a more clearly organized representation with respect to recurrent temporal contexts.

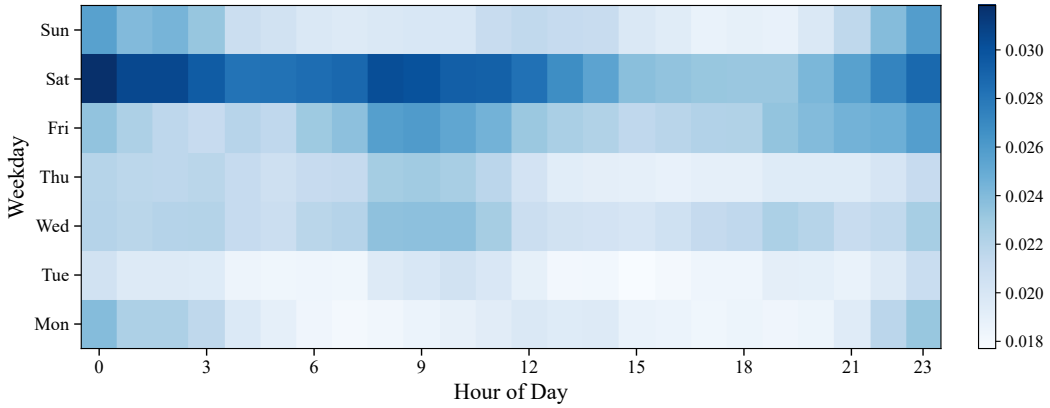


Figure 9: Visualization of context-dependent evolution-rule modulation on the Electricity dataset with $L = 96$ and $T = 96$.

Table 8: Stage-wise temporal-background intervention on the Traffic dataset. Relative MSE degradation is averaged over prediction horizons $T \in \{96, 192, 336, 720\}$.

Intervention Stage	Temporal-Background Shift			
	± 1 h	± 3 h	± 6 h	± 12 h
State Calibration	9.13%	44.35%	96.32%	179.73%
Spectral Evolution	0.69%	1.51%	2.20%	3.11%
Both Stages	10.88%	49.59%	103.43%	184.46%

E.3 Background-Conditioned Spectral Evolution

To further examine how recurrent temporal backgrounds affect spectral evolution, we visualize the magnitude of the local spectral weight adjustment on Electricity. For each weekday-hour context, we construct the corresponding temporal background X_s and obtain the local adjustment ΔW_s from the Evolution Modulator. We then average $|\Delta W_s|$ over variables and frequency bins, producing a 7×24 context map. As shown in Figure 9, the adjustment magnitude exhibits structured variations across weekday-hour contexts. This indicates that different temporal backgrounds induce different strengths of local adjustment to the global spectral weights. Together with the context-intervention results, this observation provides empirical evidence that TimeRift adapts its spectral evolution according to the temporal background.

E.4 Stage-Wise Temporal-Background Intervention

We further perturb the temporal background supplied to the two evolution stages separately on the Traffic dataset. The historical input, background-relative deviation, and reconstruction background are kept unchanged. Results are averaged over positive and negative shifts and over prediction horizons $T \in \{96, 192, 336, 720\}$. As shown in Table 8, state calibration is highly sensitive to temporal-background misalignment. Its average MSE degradation increases from 9.13% at ± 1 h to 179.73% at ± 12 h. Spectral evolution shows a smaller but consistently increasing degradation, from 0.69% to 3.11%. The results show that state calibration is more sensitive to temporal-background misalignment. Spectral evolution exhibits a smaller but consistently increasing sensitivity as the context shift grows. This difference is expected because the background-conditioned local weights serve as an adjustment to the shared global spectral weights.

Table 9: Analysis of temporal-context utilization in TimeRift. Each entry reports MSE/MAE. Avg. denotes the average over prediction horizons $T \in \{96, 192, 336, 720\}$. Factor-ablation and auxiliary-embedding variants are retrained, while Shuffled Context is applied only at inference.

Dataset	Variant	96	192	336	720	Avg.
		MSE/MAE	MSE/MAE	MSE/MAE	MSE/MAE	MSE/MAE
ETTm1	Full	**0.307/0.345**	**0.355/0.373**	**0.378/0.393**	**0.433/0.427**	**0.368/0.384**
	Auxiliary Temporal Embedding	0.311/0.346	0.359/0.378	0.385/0.395	0.441/0.433	0.374/0.388
	w/o Temporal Factor	0.314/0.348	0.362/0.377	0.394/0.400	0.461/0.439	0.383/0.391
	Shuffled Context	0.420/0.421	0.621/0.508	0.653/0.527	0.757/0.575	0.613/0.508
Electricity	Full	**0.133/0.226**	**0.151/0.242**	**0.168/0.260**	**0.204/0.292**	**0.164/0.255**
	Auxiliary Temporal Embedding	0.137/0.231	0.153/0.246	0.171/0.264	0.210/0.298	0.168/0.260
	w/o Hour	0.139/0.232	0.156/0.247	0.173/0.265	0.212/0.297	0.170/0.260
	w/o Day	0.152/0.245	0.164/0.255	0.181/0.273	0.219/0.305	0.179/0.270
	w/o Temporal Factors	0.177/0.256	0.182/0.264	0.199/0.281	0.240/0.314	0.200/0.279
	Shuffled Context	0.308/0.365	0.313/0.381	0.390/0.443	0.578/0.563	0.397/0.438
Traffic	Full	**0.411/0.268**	**0.427/0.281**	**0.443/0.289**	**0.475/0.307**	**0.439/0.286**
	Auxiliary Temporal Embedding	0.425/0.271	0.438/0.284	0.456/0.291	0.486/0.315	0.451/0.290
	w/o Hour	0.425/0.275	0.438/0.285	0.455/0.294	0.485/0.312	0.451/0.292
	w/o Day	0.471/0.299	0.466/0.301	0.480/0.306	0.514/0.323	0.483/0.307
	w/o Temporal Factors	0.480/0.306	0.482/0.306	0.495/0.310	0.527/0.329	0.496/0.313
	Shuffled Context	0.942/0.548	0.890/0.525	1.011/0.588	1.213/0.666	1.014/0.582

F Ablation Studies

We further study the role of temporal factors in TimeRift. We select ETTm1, Electricity, and Traffic for this analysis. These datasets cover two temporal-context settings. ETTm1 uses only the hour factor, while Electricity and Traffic use both hour and day factors.

Temporal-factor ablation. We first remove the temporal factors used on each dataset. For the ETTm1 dataset, we remove the hour factor. For the Electricity and Traffic datasets, we separately remove the hour factor, the day factor, or both. Other components remain unchanged. Each ablated model is retrained from scratch. The results are reported in Table 9. Removing temporal factors consistently reduces forecasting accuracy. On the ETTm1 dataset, removing the hour factor increases the average MSE from 0.368 to 0.383. On Electricity and Traffic, removing the day factor causes a larger degradation than removing the hour factor. Removing both factors leads to the largest degradation among the retrained variants. These results show that the temporal factors provide useful information for modeling the temporal background.

Temporal-context correspondence analysis. We further examine whether the temporal context needs to match the current input window. During inference, we randomly reassign temporal contexts across test samples. The historical inputs, prediction targets, and model parameters remain unchanged. For the Electricity and Traffic datasets, the hour and day factors are reassigned together. This preserves the temporal-context values while breaking their correspondence with the input windows. The shuffled setting causes a substantial performance drop on three datasets. The average MSE increases from 0.368 to 0.613 on ETTm1, from 0.164 to 0.397 on Electricity, and from 0.439 to 1.014 on Traffic. The degradation is much larger than that caused by removing the temporal factors. This shows that the correspondence between the input window and its temporal context plays an important role in TimeRift.

Auxiliary temporal embedding. We further examine whether the gains of TimeRift mainly arise from the availability of temporal information. We construct the same temporal representation from the hour and day indices, but use it only as an additive auxiliary representation. It does not define the background-relative deviation or condition the subsequent dynamic evolution. As shown in Table 9, this variant consistently improves over removing the temporal factors. This confirms that temporal information itself is useful for forecasting. However, the full TimeRift achieves lower average MSE on three datasets. These results show that temporal information is useful even when used only as

Table 10: Full results of different models with the look-back length $L = 96$. The reported results with standard deviation of TimeRift are averaged from 5 runs with different random seeds $\{2021, 2022, 2023, 2024, 2025\}$. The best results are highlighted in **bold**, while the second-best results are underlined.

Model		TimeRift (ours)		TimeEmb 2026		GTR 2026		TQNet 2025		FilterTS 2025		CycleNet 2024		iTransformer 2024		FITS 2024		DLinear 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.362±0.001**	**0.387±0.000**	0.366	**0.387**	0.367	0.391	0.371	0.393	0.374	0.391	0.378	0.391	0.386	0.405	0.386	0.396	0.386	0.400
	192	**0.416±0.000**	**0.416±0.001**	0.417	**0.416**	0.420	0.421	0.428	0.426	0.424	0.421	0.426	0.419	0.441	0.436	0.436	0.423	0.437	0.432
	336	**0.457±0.001**	0.437±0.001	**0.457**	**0.436**	0.476	0.447	0.476	0.446	0.464	0.441	0.464	0.439	0.487	0.458	0.478	0.444	0.481	0.459
	720	**0.456±0.002**	**0.455±0.001**	0.459	0.460	0.493	0.476	0.487	0.470	0.470	0.466	0.461	0.460	0.503	0.491	0.502	0.495	0.519	0.516
	avg	**0.422±0.001**	**0.423±0.001**	0.425	0.425	0.439	0.434	0.441	0.434	0.433	0.430	0.432	0.427	0.454	0.447	0.451	0.440	0.456	0.452
ETTh2	96	**0.276±0.000**	**0.325±0.001**	0.277	0.328	0.290	0.342	0.295	0.343	0.290	0.338	0.285	0.335	0.297	0.349	0.295	0.350	0.333	0.387
	192	**0.349±0.002**	**0.375±0.001**	0.356	0.379	0.362	0.389	0.367	0.393	0.374	0.390	0.373	0.391	0.380	0.400	0.381	0.396	0.477	0.476
	336	**0.397±0.001**	**0.413±0.001**	0.400	0.417	0.414	0.429	0.417	0.427	0.406	0.420	0.421	0.433	0.428	0.432	0.426	0.438	0.594	0.541
	720	**0.410±0.002**	**0.435±0.001**	0.416	0.437	0.423	0.442	0.433	0.446	0.418	0.437	0.453	0.458	0.427	0.445	0.431	0.446	0.831	0.657
	avg	**0.358±0.001**	**0.387±0.001**	0.362	0.390	0.372	0.400	0.378	0.402	0.372	0.396	0.383	0.404	0.383	0.407	0.383	0.408	0.559	0.515
ETTm1	96	0.307±0.001	0.345±0.002	**0.304**	**0.343**	0.305	0.349	0.311	0.353	0.321	0.360	0.319	0.360	0.334	0.368	0.355	0.375	0.345	0.372
	192	0.355±0.001	**0.373±0.001**	0.354	**0.373**	**0.349**	0.375	0.356	0.378	0.363	0.382	0.360	0.381	0.377	0.391	0.392	0.393	0.380	0.389
	336	**0.378±0.001**	**0.393±0.001**	0.379	**0.393**	0.380	0.398	0.390	0.401	0.395	0.403	0.389	0.403	0.426	0.420	0.424	0.414	0.413	0.413
	720	**0.433±0.001**	**0.427±0.000**	0.435	0.428	0.435	0.436	0.452	0.440	0.462	0.438	0.447	0.441	0.491	0.459	0.487	0.449	0.474	0.453
	avg	0.368±0.001	**0.384±0.001**	0.368	**0.384**	**0.367**	0.389	0.377	0.393	0.385	0.396	0.379	0.396	0.407	0.410	0.415	0.408	0.403	0.407
ETTm2	96	**0.161±0.001**	**0.240±0.001**	0.163	0.242	0.168	0.249	0.173	0.256	0.172	0.255	0.163	0.246	0.180	0.264	0.183	0.266	0.193	0.292
	192	**0.224±0.001**	**0.282±0.001**	0.226	0.285	0.232	0.292	0.238	0.298	0.237	0.299	0.229	0.290	0.250	0.309	0.247	0.305	0.284	0.362
	336	0.287±0.000	**0.324±0.001**	0.286	**0.324**	0.287	0.330	0.301	0.340	0.299	0.398	**0.284**	0.327	0.311	0.348	0.307	0.342	0.369	0.427
	720	**0.380**±0.001	**0.381±0.001**	0.383	**0.381**	0.386	0.390	0.397	0.396	0.397	0.394	0.389	0.391	0.412	0.407	0.407	0.399	0.554	0.522
	avg	**0.263±0.001**	**0.307±0.001**	0.265	0.308	0.268	0.315	0.277	0.323	0.276	0.336	0.266	0.314	0.288	0.332	0.286	0.328	0.350	0.401
Weather	96	**0.150±0.000**	**0.190±0.001**	**0.150**	**0.190**	0.154	0.200	0.157	0.200	0.162	0.207	0.158	0.203	0.174	0.214	0.166	0.213	0.196	0.255
	192	**0.200±0.001**	**0.238±0.000**	**0.200**	**0.238**	0.202	0.244	0.206	0.245	0.209	0.252	0.207	0.247	0.221	0.254	0.213	0.254	0.237	0.296
	336	**0.257±0.000**	**0.281±0.000**	0.259	0.282	0.259	0.287	0.262	0.287	0.263	0.292	0.262	0.289	0.278	0.296	0.269	0.294	0.283	0.335
	720	**0.339±0.001**	**0.336±0.001**	**0.339**	**0.336**	0.341	0.342	0.344	0.342	0.344	0.344	0.344	0.344	0.358	0.347	0.346	0.343	0.345	0.381
	avg	**0.236±0.001**	**0.261±0.001**	0.237	0.262	0.239	0.268	0.242	0.269	0.244	0.274	0.243	0.271	0.258	0.278	0.249	0.276	0.265	0.317
Electricity	96	**0.133±0.001**	**0.226±0.000**	0.136	0.231	0.134	0.229	0.134	0.229	0.151	0.245	0.136	0.229	0.148	0.240	0.200	0.278	0.197	0.282
	192	**0.151±0.001**	**0.242±0.001**	0.153	0.246	0.152	0.245	0.154	0.247	0.163	0.256	0.152	0.244	0.162	0.253	0.200	0.280	0.196	0.285
	336	**0.168±0.001**	**0.260±0.001**	0.170	0.264	0.171	0.264	0.169	0.264	0.180	0.274	0.170	0.264	0.178	0.269	0.214	0.295	0.209	0.301
	720	0.204±0.001	**0.292±0.001**	0.208	0.297	0.208	0.300	**0.201**	0.294	0.224	0.311	0.212	0.299	0.225	0.317	0.255	0.327	0.245	0.333
	avg	**0.164±0.001**	**0.255±0.001**	0.167	0.260	0.166	0.260	**0.164**	0.259	0.180	0.271	0.168	0.259	0.178	0.270	0.217	0.295	0.212	0.300
Traffic	96	0.411±0.002	0.268±0.001	0.432	0.279	0.440	0.263	0.413	**0.261**	0.448	0.309	0.458	0.296	**0.395**	0.268	0.651	0.391	0.650	0.396
	192	0.427±0.001	0.281±0.001	0.442	0.289	0.454	0.274	0.432	**0.271**	0.455	0.307	0.457	0.294	**0.417**	0.276	0.602	0.363	0.598	0.370
	336	0.443±0.001	0.289±0.001	0.456	0.295	0.472	0.282	0.450	**0.277**	0.472	0.313	0.470	0.299	**0.433**	0.283	0.609	0.366	0.605	0.373
	720	0.475±0.001	0.307±0.001	0.487	0.311	0.514	0.301	0.486	**0.295**	0.508	0.332	0.502	0.314	**0.467**	0.302	0.647	0.385	0.645	0.394
	avg	0.439±0.001	0.286±0.001	0.454	0.293	0.470	0.280	0.445	**0.276**	0.471	0.315	0.472	0.301	**0.428**	0.282	0.627	0.376	0.625	0.383

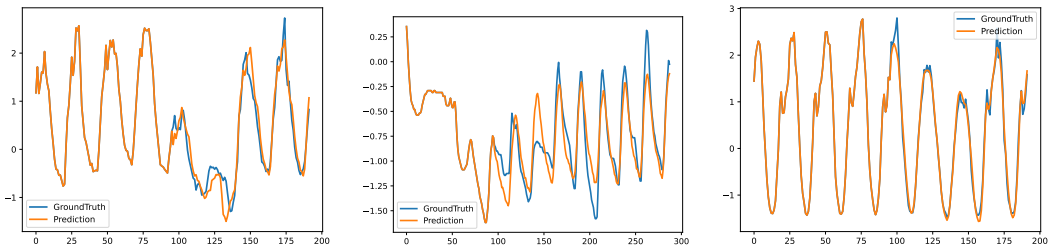
an auxiliary embedding. TimeRift achieves further improvements by using it to define background-relative deviations and condition their subsequent evolution.

G Full Comparison Results

Table 10 presents the full comparison results of TimeRift on seven multivariate time series forecasting datasets. TimeRift achieves superior overall forecasting performance under most experimental settings. These results demonstrate the effectiveness of the proposed method.

H Visualization of Prediction

Figure 10 shows representative forecasting examples of TimeRift on Electricity, ETTh2, and Traffic. The forecasts generally follow the ground-truth trajectories and capture the main temporal patterns across the three datasets.



(a) Electricity (b) ETTh2 (c) Traffic

Figure 10: Qualitative comparison between TimeRift forecasts and the groundtruth sequences on Electricity, ETTh2, and Traffic.

I LIMITATION

While TimeRift demonstrates strong performance in modeling background-conditioned dynamics, several limitations remain. First, the effectiveness of TimeRift depends on the presence of meaningful recurrent temporal regularities. When such regularities are weak, or when future dynamics are mainly driven by irregular events, the benefit of temporal-background conditioning may become less pronounced. Second, the current design constructs temporal backgrounds from a predefined set of recurrent temporal factors, such as hour-of-day and day-of-week. This may limit its ability to represent more complex temporal contexts beyond regular calendar patterns. A more flexible background construction mechanism that incorporates richer contextual information could further improve its capability to characterize context-dependent dynamics. In addition, TimeRift mainly focuses on temporal background-dependent evolution and does not explicitly model cross-variable interactions. Introducing lightweight cross-variable modeling could further improve its ability to handle multivariate series with strong inter-variable dependencies.