Rethinking Time Series Tokenization from a Frequency Perspective

Zelong Tian[1,2] **Ming Jin**[1*] **Bo Du**[2] **Shirui Pan**[1*]
[1] School of Information and Communication Technology,
[2] Griffith Business School
Griffith University, Brisbane, Australia
zelongtian006@gmail.com, {ming.jin, bo.du, s.pan}@griffith.edu.au

Abstract

Tokenization that partitions time series into subsequences has become a foundational paradigm in modern time series modeling. However, we find that current tokenization strategies inherently introduce spectral distortions, forcing the learned representations to diverge from the true underlying patterns such as trend, periodicity, and seasonality, thereby impairing model performance. Through theoretical analysis from a frequency perspective, we derive the boundary conditions under which such distortions occur. To overcome this distortion, we propose **TimeFT**, a nearly distortion-free frequency-based tokenizer for time series data. TimeFT obtains tokens through frequency-domain partitioning, frequency shifting, and Nyquist sampling. The method does not have additional learnable parameters, incurs negligible computational complexity, and can serve as a drop-in replacement for existing tokenizers. Our extensive experiments on diverse tasks such as forecasting, classification, and anomaly detection demonstrate that TimeFT consistently and significantly improves performance. The code has been made available at `https://github.com/ME1682/TimeFT`.

1 Introduction

Models [1] such as PatchTST [2], PatchMLP [3] and PhaseFormer [4] have established a tokenization paradigm [5, 6] for time series data by partitioning the raw sequence into a set of non-overlapping (or overlapping) segments:

$$\mathbf{Z}_{\text{patch } i} := X\left[\sum_{j=1}^{i-1} S_j \;:\; \sum_{j=1}^{i-1} S_j + L_i\right] \quad i \in \{1, 2, \ldots, I\} \tag{1}$$

$$\mathbf{Z}_{\text{phase } i} := [\, X[\, j \,] \text{ for } j = i, i+S, \ldots] \quad j \le N-1,\ i \in \{0, 1, 2, \ldots, I\} \tag{2}$$

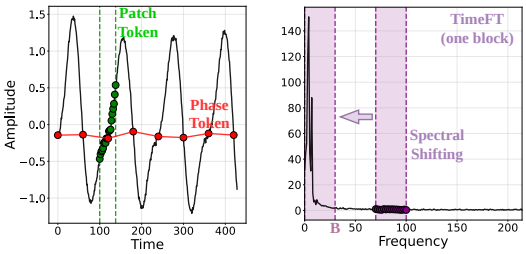


Figure 1: **The key distinction between existing tokenization and TimeFT.** *Left:* The processes of patch and phase tokenization in the time domain. *Right:* Spectral shifting moves the frequency spectrum, preventing aliasing by ensuring all components satisfy the Nyquist criterion.

where $\mathbf{Z}_{\text{patch } i}$ is the patch token [2] and $\mathbf{Z}_{\text{phase } i}$ is the phase token [4], which represent two main classes of tokenization methods. X denotes the original input sequence of length N, L_i is the length

[*] Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

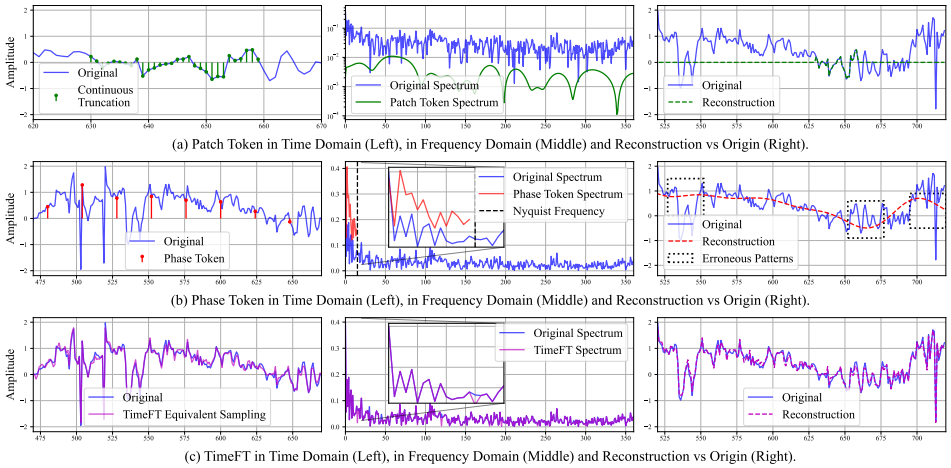


Figure 2: **The comparison of tokenization methods across the time domain, frequency domain and reconstruction versus origin.** (a) Patch tokens lead to a completely different frequency spectrum and cause the loss of all information outside the rectangular window. (b) Phase tokens cause frequency components above the Nyquist frequency to completely alias into the low frequencies, thereby exhibiting erroneous patterns. (c) Our TimeFT effectively solves the above issues and achieves almost distortion-free tokenization. The further explanation of Figure 2 and the details of reconstruction methods are provided in the Appendix C.

of each patch, S_i or S is the stride, and I is the number of tokens. As shown in Figure 1, the patch tokenizer can be seen as segmenting the sequence in the time domain, and the phase tokenizer can be regarded as downsampling in the time domain. Extensive existing work shows that the tokenization paradigm has become a fundamental component of modern time series modeling (§5). However, the frequency domain effects of tokenization remain largely unexplored.

For patch tokens, rectangular window truncation in the time domain leads to issues such as information loss and spectral leakage in the frequency domain (§B.1). For phase tokens, downsampling in the time domain leads to frequency aliasing in the frequency domain (§B.2). These distortions in tokenization cause the patterns of tokens, such as trend, periodicity, and seasonality, to diverge significantly from those of the original time series, thereby leading the model to learn erroneous patterns and degrading model performance. The fundamental reason for this phenomenon lies in distortions caused by tokenization methods and the constraints that the sequence length imposes on the frequency. Towards this, we propose **TimeFT**, a nearly distortion-free frequency-based tokenizer for time series data.

$$\mathbf{Z}_{\mathrm{TimeFT}} := \mathrm{Reshape}\left(\mathrm{Nyquist\ Sampling}_S\left(\mathcal{F}_{\mathbb{R}}^{-1}\left(\mathrm{Shift}\left(\mathrm{Block}_K\left(\mathcal{F}_{\mathbb{R}}(X)\right)\right)\right)\right)\right) \tag{3}$$

where $\mathbf{Z}_{\mathrm{TimeFT}}$ is the set of TimeFT tokens, S is the sampling stride, and K is the number of blocks. $\mathrm{Block}_K(\cdot)$, $\mathrm{Shift}(\cdot)$, $\mathrm{Nyquist\ Sampling}_S(\cdot)$ and $\mathrm{Reshape}(\cdot)$ denote the operations 7, 9, 12 and 13 of TimeFT, which split the time series spectrum into multiple sub-bands that satisfy the alias-free condition, shift all of them to baseband, and then apply Nyquist sampling. $\mathcal{F}_{\mathbb{R}}$ and $\mathcal{F}_{\mathbb{R}}^{-1}$ denote the real-valued fast Fourier transform (RFFT) and its inverse transform. The comparison of three tokenization methods is illustrated in Figure 2.

As mentioned, TimeFT is a method without additional learnable parameters that can serve as a fundamental time series tokenization module for a wide range of models, and its time complexity is negligible (§4.3). We prove that our method solves the above data distortion issue (§B.4). As a result, TimeFT can consistently and effectively improve model performance in various tasks (§4.2). More motivation explanations and theoretical insights are provided in §A. The main contributions of our work are as follows.

- We find that the existing time series tokenization paradigm causes distortions in the frequency domain, which leads the data to exhibit erroneous temporal patterns. We then derive the transformation relationships in the time domain and the frequency domain during tokenization and further establish the boundary conditions of such distortions.

- We propose TimeFT based on the above boundary conditions, which generates tokens while preserving frequency information by employing frequency-domain partitioning, spectrum shifting, and Nyquist sampling. Our method effectively solves the frequency distortion issue during tokenization.
- Extensive experiments demonstrate that TimeFT achieves state-of-the-art performance compared to existing tokenization methods in forecasting, classification, and anomaly detection tasks. As a plug-in module, TimeFT results in an improvement of nearly 10% in many cases, and the time cost is consistently less than 1.4% of the training time per epoch.

2 Preliminaries

In this section, we first define the tokenization problem. Then, from a frequency-domain perspective, we theoretically analyze the causes of the distortions shown in Figure 2 for patch and phase tokens. Detailed theoretical analysis and illustrations are provided in the Appendix B.

Problem Definition. Consider a multivariate time series input denoted as $\mathbf{X} \in \mathbb{R}^{V \times N}$, where V is the number of variates (channels) and N is the length of the temporal sequence. The fundamental objective of our tokenization strategy is to discretize time series $\mathbf{X} \in \mathbb{R}^{V \times N}$ into a finite collection of tokens $\mathbf{Z} \in \mathbb{R}^{V \times S \times *}$ that can be learned by models, where S and $*$ are the shapes of the tokens.

Patch Tokenization in Frequency Domain. Let $x[n]$, $n \in \{0, 1, \ldots, N-1\}$, be a discrete sequence of finite length N. The patch tokenization in Equation 1 can be equivalent to the operation of rectangular windowing. The subsequence of length L is obtained by extracting the segment in the rectangular window and re-indexing it from zero, i.e., $y[m] = x[n_0 + m]$ for $m = 0, \ldots, L-1$. This is equivalent to convolving with a shifted Dirichlet kernel in the frequency domain.

Intuitively, it is easy to assume that patch tokens distort high frequency information [7–9] due to reduced data length, but the actual situation is much more complex. Applying a rectangular window of length L to a sequence of length N induces three main distortions. First, the kernel's slowly decaying sidelobes produce **spectral leakage**, spreading energy from each frequency component across the entire spectrum and potentially obscuring weak signals. Second, the width of the mainlobe establishes an absolute **resolution limit**: components spaced closer than half this interval merge into a single broadened peak, irrespective of the original sequence length N. Moreover, the windowed sequence is further truncated to length L, so it **discards information** beyond length L while inheriting identical leakage and resolution penalties.

Phase Tokenization in Frequency Domain. The phase tokenization in Equation 2 is equivalent to the operation of downsampling, i.e., $y[k] = x[kS]$ for $k = 0, 1, \ldots, L-1$, where $L = \lceil N/S \rceil$. If the original frequency spectrum contains nonzero energy for frequencies higher than the bound B, the adjacent replicas will overlap when downsampling, causing the higher frequency components to fold into lower frequencies. This phenomenon is known as **spectral aliasing** and constitutes an irreversible distortion. To avoid aliasing, downsampling must satisfy the **Nyquist sampling theorem [10]**.

$$2\pi \frac{B}{f_s} \leq \frac{\pi}{S} \quad \Longrightarrow \quad B \leq \frac{f_s}{2S} \tag{4}$$

where f_s represents the sampling frequency of the original discrete time sequence. The downsampled signal will be free from aliasing if and only if the original continuous-time signal is bandlimited to less than or equal to the new **Nyquist frequency** $f_s/2S$. Sampling that satisfies this condition is referred to as **Nyquist sampling**.

3 Methodology

As shown in Figure 3, our TimeFT consists of three principal stages: (1) frequency-domain decomposition based on boundary conditions that extracts different sub-bands, (2) frequency shifting translates high-frequency components into the low-frequency band, and (3) information-preserving token extraction that transforms these representations into final tokens.

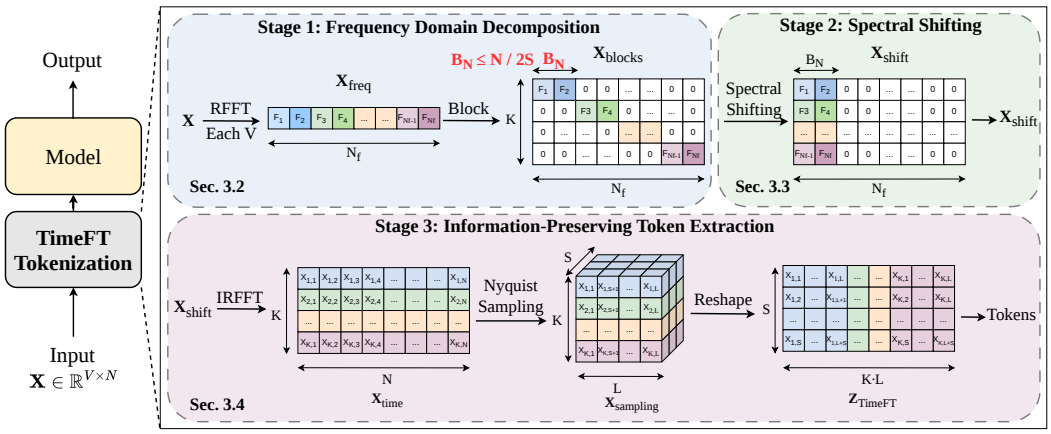


Figure 3: **The pipeline of TimeFT** is implemented by (1) frequency domain decomposition, (2) spectral shifting and (3) information-preserving token extraction. Different colors and markers are used to denote the processes of operations. Based on the frequency threshold B_N derived from downsampling without information loss, TimeFT shifts all spectra of bandwidth B_N to the baseband, and then obtains almost distortion-free tokens by Nyquist sampling.

3.1 Boundary Conditions for Distortion-Free Tokenization

Section 2 shows that for any finite discrete sequence $\mathbf{X}$, there exists a simple method (such as Digital-to-Analog Conversion) that can reconstruct it into a continuous process $\mathcal{X}$ (an infinite sequence) over a period of time. $\mathbf{X}$ can be regarded as being obtained from $\mathcal{X}$. When the boundary conditions are satisfied, both the sampling and reconstruction processes are invertible and lossless, and in this case $\mathbf{X}$ is equivalent to $\mathcal{X}$. However, due to the finite length constraint on the information contained in $\mathbf{X}$, an incorrect $\mathcal{X}$ is often reconstructed, leading to tokens of erroneous patterns. We next present the boundary conditions for this conversion.

Boundary Condition for Patch Tokenization. Patch tokenization multiplies $x[n]$ by a rectangular window of length L and extracts the non-zero segment. The necessary and sufficient condition for the token to retain all the information required for an exact reconstruction of the original signal is that the entire valid sequence of $x[n]$ is contained within the windowed interval. For a finite-duration sequence of length N, this condition is satisfied if and only if the window covers all non-zero values. If $x[n]$ is periodic with period T, the condition becomes $L = mT$ ($m \in \mathbb{N}^+$), which means that any contiguous L sample block then suffices for perfect periodic extension, completely eliminating information loss. However, these situations rarely occur for general time series.

Boundary Condition for Phase Tokenization. Compared to patch tokenization, the boundary condition of phase tokenization is much simpler. Due to the Nyquist sampling theorem, the alias-free condition requires $x[n]$ to satisfy Equation 4. Therefore, for any discrete $x[n]$, we define:

$$B_N = \lfloor \frac{N}{2S} \rfloor \tag{5}$$

As long as the maximum frequency component of $x[n]$ is less than or equal to B_N, it satisfies the boundary condition for phase tokenization. In this case, the original sequence can be perfectly recovered from the tokens, making the tokenization free from information loss.

3.2 Frequency Domain Decomposition Based on Boundary Conditions

In order to achieve lossless tokenization in the frequency domain, we first project the input time series into the frequency domain via the real-valued fast Fourier transform applied along the temporal axis, as shown in Stage 1 in Figure 3. Let $\mathcal{F}_{\mathbb{R}}$ denote the RFFT operator. The resulting complex-valued tensor is as follows:

$$\mathbf{X}_{\text{freq}} = \mathcal{F}_{\mathbb{R}}(\mathbf{X}) \in \mathbb{C}^{V \times N_f} \tag{6}$$

where $N_f = \lfloor N/2 \rfloor + 1$ is the number of frequency bins corresponding to the non-negative frequencies of a real signal of length N. We adopt a channel-independent approach for different variables to avoid frequency domain mixing. For a univariate real sequence $\mathbf{x} = (x_0, \ldots, x_{N-1})$, its k-th RFFT coefficient is given by $[\mathcal{F}_{\mathbb{R}}(\mathbf{x})]_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N},\ k = 0, 1, \ldots, N_f - 1$. To retain multi-frequency information, we partition the frequency representation along the spectral dimension into K contiguous blocks, each of size B_N. This makes the spectral bandwidth satisfy Equation 5 and the boundary condition in §3.1. For the k-th band ($k = 0, \ldots, K-1$), we define a full extraction that retains only the frequency components within the interval $[kB_N, kB_N + B_N - 1]$ and zeros out the rest:

$$\left[\mathbf{X}_{\text{block}}^{(k)}\right]_{v,f} = \begin{cases} [\mathbf{X}_{\text{freq}}]_{v,f}, & \text{if } kB_N \le f < (k+1)B_N, \\ 0, & \text{otherwise,} \end{cases} \qquad f = 0, \ldots, N_f - 1 \tag{7}$$

$$\mathbf{X}_{\text{blocks}} = \left[\mathbf{X}_{\text{block}}^{(0)}; \mathbf{X}_{\text{block}}^{(1)}; \ldots; \mathbf{X}_{\text{block}}^{(K-1)}\right] \in \mathbb{C}^{V \times K \times N_f} \tag{8}$$

where $K = \lceil N_f / B_N \rceil$ denotes the total number of spectral blocks, and the last portion of $\mathbf{X}_{\text{freq}}$ is filled by zero-padding. In this step, we separate the frequency components that correspond to characteristics such as trend, periodicity, and seasonality. Through this partitioning, we obtain a series of sequences in the frequency-domain that satisfy the required frequency band length under the boundary conditions, which is the first step in avoiding aliasing.

3.3 Spectral Shifting

However, existing effective frequency bands are located at high frequencies, causing high frequency components to irreversibly alias into low frequencies during downsampling. Therefore, we need to shift the high frequency information down to the low frequency range. As shown in Stage 2 in Figure 3, we align all bands to the baseband and multiply by a factor that compensates for the band starting frequency:

$$\left[\mathbf{X}_{\text{shift}}\right]_{v,k,b} = \left[\mathbf{X}_{\text{blocks}}\right]_{v,k,\,kB_N+b} \cdot e^{-i2\pi(kB_N)b/N}, \quad b = 0, 1, \ldots, B_N - 1 \tag{9}$$

The frequency portion of $\mathbf{X}_{\text{shift}}$ from B_N to N_f is zero-padded. Equivalently, one may view this as applying the shifting operator $\mathcal{S}$ to each band, which is $\mathbf{X}_{\text{shift}} = \mathcal{S}\left(\mathbf{X}_{\text{blocks}}\right) \in \mathbb{C}^{V \times K \times N_f}$. The resulting sequences $\mathbf{X}_{\text{shift}}$ satisfy the Nyquist sampling condition when sampled with a step size of S, ensuring that no aliasing occurs. In Equation 9, the factor is applied to align the phase reference with the time origin after shifting the frequency band. It compensates for the phase shift introduced by the translation without altering the magnitude of the complex signal in the frequency domain. Since TimeFT does not compress the temporal dimension, the phase factor in Equation 9 is ignored for models (§B.4.2). Therefore, Equations 7-9 can be simplified to the following form to avoid computing the above complex steps:

$$\mathbf{X}_{\text{shift}} = \text{ZeroPad}_{B_N \to N_f}\Big(\text{Reshape}\big(\mathbf{X}_{\text{freq}}, (V, K, B_N)\big)\Big) \in \mathbb{C}^{V \times K \times N_f} \tag{10}$$

where the $\text{ZeroPad}_{B_N \to N_f}$ operation denotes appending zeros to the end of each frequency-domain sequence of length B_N to extend its dimension to N_f. The $\text{Reshape}(\cdot)$ operation zero-pads at the end if the dimension is insufficient. Here, the spectrum will slightly lose the imaginary part at zero frequency. In this way, we obtain a frequency-domain sequence that satisfies the Nyquist sampling theorem.

3.4 Information-Preserving Token Extraction

However, the tokenization extraction operation should be performed in the time domain. As shown in Stage 3 in Figure 3, to recover the corresponding time domain sequence, we apply the inverse real fast Fourier transform (IRFFT) $\mathcal{F}_{\mathbb{R}}^{-1}$ to every frequency block independently.

$$\mathbf{X}_{\text{time}} = \mathcal{F}_{\mathbb{R}}^{-1}\left(\mathbf{X}_{\text{shift}}\right) \in \mathbb{R}^{V \times K \times N} \tag{11}$$

where $\mathbf{X}_{\text{time}}$ denotes the time-domain representation of the sequences of K different frequency bands of width B_N for each variable V. Then we perform *dense sampling*: for every possible offset $\lambda \in \{0, 1, \ldots, S-1\}$, we sample the signal starting at the index λ with step S. Formally, for the k-th band of the variable v, let $\mathbf{x}_{v,k} = \mathbf{X}_{\text{time}}[v, k, \cdot] \in \mathbb{R}^N$. The dense sampling operator $\mathcal{D}_S$ produces a matrix of size $S \times L$, where $L = \lceil N/S \rceil$:

$$\big[\mathcal{D}_S(\mathbf{x}_{v,k})\big]_{\lambda,\ell} = \mathbf{x}_{v,k}[\lambda + \ell \cdot S], \quad \lambda = 0, \ldots, S-1, \;\; \ell = 0, \ldots, L-1 \tag{12}$$

Applying this operator independently to every sub-band yields the final token tensor, which is $\mathbf{X}_{\text{sampling}} = \mathcal{D}_S\big(\mathbf{X}_{\text{time}}\big) \in \mathbb{R}^{V \times K \times S \times L}$. Thus, each variate is represented by $K \times S$ vectors of length L, where K indexes the frequency bands and S captures different temporal phases. In our framework, before sampling, the k-th band has been shifted to the baseband and truncated to the B frequency bins. Due to Equation 5 the sampling stride S satisfies $S \leq N/(2B_N)$. Hence, the operator $\mathcal{D}_S(\cdot)$ corresponds to the **Nyquist sampling** operation. Consequently, the dense sampling procedure preserves information and is lossless with respect to the sub-band representation. Finally, to obtain a unified token, we first permute the dimension to group the frequency and sampling dimensions, then flatten the frequency bands into the feature dimension.

$$\mathbf{Z}_{\text{TimeFT}} = \text{reshape}\Big(\text{permute}\big(\mathbf{X}_{\text{sampling}}\big),\, (V, S, K \cdot L)\Big) \in \mathbb{R}^{V \times S \times (K \cdot L)} \tag{13}$$

where $\text{permute}(\cdot)$ reorders the dimension from (V, K, S, L) to (V, S, K, L), and the subsequent reshape collapses the last two dimensions into a single feature dimension of size $K \cdot L$. Since models typically apply embedding to tokens, the $K \cdot L$ dimension here does not lead to the curse of dimensionality. Each step of TimeFT is designed to be approximately invertible. The simplified spectral shifting only introduces a negligible loss at the zero-frequency component, and we can almost perfectly reconstruct the original spectrum from each token (§C.4).

4 Experiments

In this section, we conduct a comprehensive evaluation of TimeFT, including three major time series tasks (§4.2), the computational efficiency (§4.3), the plug-in experiment (§4.4), the ablation study (§4.5) and the hyperparameter sensitivity (§4.7). More details, experiments under noisy conditions and results on different random seeds are provided in § D, and visualizations of TimeFT are provided in § F.

4.1 Experimental Setup

Datasets. We evaluate TimeFT on a comprehensive set of benchmarks spanning three core time-series tasks, and we select commonly used datasets for each task. **Long-term forecasting** is performed on seven real-world datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity and Traffic [11–13]. **Classification** uses 10 UEA datasets [14–16]: EthanolConcentration, FaceDetection, Handwriting, Heartbeat, JapaneseVowels, PEMS-SF, SelfRegulationSCP1, SelfRegulationSCP2, SpokenArabicDigits, and UWaveGestureLibrary. **Anomaly detection** is conducted on five widely used benchmarks [14, 17, 18]: SMD, MSL, SMAP, SWaT, and PSM.

Baselines. We compare TimeFT with state-of-the-art tokenization methods and classic time series models. We consider EntroPE [14], TimeMosaic [19], PhaseFormer [4], PatchMLP [3] and PatchTST [2] as standard baselines. In addition, we also use TimeFilter [20], xPatch [21], iTransformer [13], TimeMixer [22], TimesNet [12], LightTS [23], DLinear [24], GPT4TS [25], ETSformer [26], FEDformer [27] as baselines. The specific baselines differ slightly between tasks, due to the choice of baselines in previous papers. Since TimeFT is a general tokenization method rather than a model and due to expensive pre-train costs, we do not use time series foundation models (TSFMs) as baselines or plug-in models. **We use PatchTST as the backbone model for TimeFT in all experiments except the plug-in experiment.**

Table 1: **For long-term forecasting task**, we evaluate average MSE and MAE across prediction horizons $H \in \{96, 192, 336, 720\}$. All baselines use a fixed input sequence length of 320. Best results are in **bold** and second best are underlined. The detailed performance is provided in Table 8.

Models	**TimeFT (Ours)**		EntroPE (2026)		TimeMosaic (2026)		PhaseFormer (2026)		TimeFilter (2025)		xPatch (2025)		PatchMLP (2025)		iTransformer (2024)		TimeMixer (2024)		PatchTST (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.348**	**0.373**	0.369	0.383	0.360	0.377	0.364	0.392	0.399	0.408	0.354	0.386	0.374	0.396	0.386	0.406	0.356	0.385	0.372	0.399
ETTm2	**0.255**	**0.307**	0.264	0.320	0.256	0.310	0.268	0.324	0.278	0.330	0.262	0.322	0.269	0.324	0.293	0.338	0.265	0.323	0.278	0.332
ETTh1	0.417	**0.423**	0.424	0.433	**0.412**	**0.423**	0.489	0.475	0.511	0.482	0.446	0.445	0.462	0.460	0.487	0.475	0.430	0.438	0.503	0.485
ETTh2	0.358	**0.390**	0.374	0.401	0.363	0.393	0.384	0.417	0.402	0.421	**0.354**	0.397	0.380	0.416	0.411	0.425	0.358	0.396	0.398	0.427
Weather	0.232	0.261	0.237	0.264	0.228	**0.256**	0.233	0.268	0.231	0.270	**0.224**	0.262	0.230	0.267	0.237	0.272	0.227	0.266	0.236	0.271
Electricity	**0.165**	**0.258**	0.177	0.271	0.170	0.263	0.177	0.273	0.167	0.260	0.170	0.264	0.200	0.302	0.180	0.277	0.173	0.267	0.173	0.272
Traffic	**0.420**	0.290	0.444	0.300	0.434	0.287	0.441	0.305	**0.420**	**0.284**	0.430	0.300	0.525	0.382	0.463	0.335	0.444	0.316	0.428	0.297

Table 2: **For classification task**, we evaluate accuracy (%) across all UEA datasets. Best results are in **bold** and second best are underlined. The detailed performance is provided in Table 9.

Models	**TimeFT (Ours)**	EntroPE (2026)	TimeMosaic (2026)	PhaseFormer (2026)	PatchMLP (2025)	GPT4TS (2024)	PatchTST (2023)	TimesNet (2023)	MLP -
Average	**71.31**	70.07	68.82	67.19	69.42	58.51	69.13	66.55	65.60

Table 3: **For anomaly detection task**, we evaluate F1-score (%) on five benchmark datasets. Best results are in **bold** and second best are underlined. The detailed performance is provided in Table 11.

Models	**TimeFT (Ours)**	EntroPE (2026)	TimeMosaic (2026)	PhaseFormer (2026)	LightTS (2024)	TimesNet (2023)	PatchTST (2023)	DLinear (2023)	ETSformer (2023)	FEDformer (2022)
Average	**86.12**	85.38	82.50	83.01	84.23	85.24	82.79	82.46	82.87	84.97

4.2 Main Results on Various Tasks

For the long-term forecasting task, we evaluate the performance of TimeFT against state-of-the-art tokenization baselines across diverse datasets. The experiments adopt the settings in TimeMosaic [19]. As demonstrated in Table 1, TimeFT achieves the best performance in the majority of datasets and metrics. By outperforming both classic and recent baselines, TimeFT establishes an excellent paradigm for obtaining tokens of time series. The detailed performance is provided in Table 8.

For the classification task, we evaluate the performance on 10 commonly used UEA datasets. As shown in Tables 2 and 9, TimeFT achieves the highest average accuracy and achieves the best or second-best result in a majority of datasets. This strong performance improvement is due to TimeFT's almost distortion-free tokenization ability, which faithfully preserves discriminative temporal patterns that are often corrupted by tokenizers.

For the anomaly detection task, we evaluate the capability on five standard datasets. TimeFT preserves the integrity of anomalous patterns and shows excellent detection performance. As shown in Tables 3 and 11, experiments on extensive datasets demonstrate that TimeFT exhibits superior performance and achieves the highest average F1-score of 86.12%.

4.3 Time Complexity Analysis

The computational complexity of the proposed TimeFT tokenization is $\mathcal{O}(VN \log N + VKN \log N + VKN)$. Each of the $V \times K$ frequency blocks is transformed back to the time domain via an IRFFT (Equation 11), contributing the dominant term of $\mathcal{O}(VKN \log N)$. As shown in Table 4, the one-time overhead introduced by TimeFT is negligible relative to the aggregate cost of model training and evaluation. Moreover, tokenization can be efficiently batched and parallelized, further amortizing its practical runtime. For multiple epochs, TimeFT produces identical tokens as long as the input sequence is the same, thus eliminating the need for repeated execution.

Table 4: **Comparison of TimeFT tokenization time and training time** per epoch across different datasets. The percentage represents the proportion of TimeFT tokenization time relative to the training time per epoch.

Datasets	ETTm1	ETTm2	ETTh1	ETTh2	Weather	Electricity	Traffic
TimeFT (s)	**1.492**	**1.376**	**0.352**	**0.362**	**3.799**	**28.50**	**50.27**
Training (s)	299.5	177.5	80.55	127.8	833.7	2157	3629
Percentage	0.50%	0.78%	0.44%	0.28%	0.46%	1.32%	1.38%

Table 5: **Plug-in performance** on long-term forecasting task. Evaluation of TimeFT as a plug-in tokenization module on prediction horizon $H = 96$ and input length of 320. Experiments on more architectures are provided in Table 14.

Datasets	ETTm1		ETTm2		ETTh1		ETTh2		Weather		Electricity		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST	0.321	0.361	0.181	0.270	0.403	0.423	0.324	0.374	0.155	0.204	0.142	0.245	0.403	0.285
+TimeFT	**0.290**	**0.339**	**0.162**	**0.244**	**0.376**	**0.392**	**0.297**	**0.339**	**0.147**	**0.194**	**0.139**	**0.234**	**0.395**	**0.279**
$\Delta\%$ Improvement	9.7%	6.1%	10.5%	9.6%	6.7%	7.3%	8.3%	9.4%	5.2%	4.9%	2.1%	4.5%	2.0%	2.1%
PhaseFormer	0.312	0.352	0.174	0.261	0.389	0.411	0.309	0.361	0.152	0.201	0.145	0.248	0.418	0.289
+TimeFT	**0.293**	**0.341**	**0.161**	**0.249**	**0.379**	**0.397**	**0.299**	**0.343**	**0.149**	**0.193**	**0.143**	**0.242**	**0.406**	**0.278**
$\Delta\%$ Improvement	6.1%	3.1%	7.5%	4.6%	2.6%	3.4%	3.2%	5.0%	1.9%	3.9%	1.4%	2.4%	2.9%	3.8%

Table 6: **Ablation study** on long-term forecasting task on prediction horizon $H = 96$ and input length of 320. The "w/o" indicates that the operations of the corresponding stage in Figure 3 are skipped in the pipeline, while the rest remains unchanged. Stage 1 is frequency domain decomposition, Stage 2 is spectral shifting and Stage 3 is information-preserving token extraction.

Datasets	ETTm1		ETTm2		ETTh1		ETTh2		Weather		Electricity		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeFT	**0.290**	**0.339**	**0.162**	**0.244**	**0.376**	**0.392**	**0.297**	**0.339**	**0.147**	**0.194**	**0.139**	**0.234**	**0.395**	**0.279**
w/o Stage 1	0.328	0.364	0.175	0.261	0.412	0.421	0.315	0.358	0.162	0.211	0.151	0.255	0.408	0.288
w/o Stage 2	0.314	0.352	0.194	0.282	0.398	0.415	0.342	0.384	0.158	0.207	0.146	0.249	0.423	0.301
w/o Stage 3	0.375	0.412	0.228	0.315	0.482	0.485	0.396	0.432	0.189	0.242	0.178	0.294	0.475	0.342

4.4 Plug-in Experiment

We evaluate the universality of TimeFT using it as a plug-in tokenization module for PatchTST and PhaseFormer. As shown in Table 5, integrating TimeFT consistently improves performance in all seven datasets. For PatchTST, performance in MSE and MAE improves from nearly 10% in many cases. For PhaseFormer, the corresponding improvements are more than 5% in many cases. These results demonstrate that TimeFT's near-lossless tokenization is readily transferable to different architectures, delivering immediate gains without modifying the base model design.

4.5 Ablation Study

We conduct the ablation study to demonstrate the contribution of each part in TimeFT. Removing Stage 1 means that only a single block exists, and the subsequent operations artificially introduce aliasing. Removing Stage 2 means that the high-frequency components of each block will individually alias into the low frequencies, and removing Stage 3 is equivalent to using different frequency components directly as tokens. As shown in Table 6, removing any individual stage significantly degrades the performance, showing that every component is essential to the model.

4.6 Distortion Comparison of Different Tokenizations

To intuitively illustrate the degree of distortion for different types of tokens, we compare the mean absolute error between the reconstructed sequences and the original sequences across different

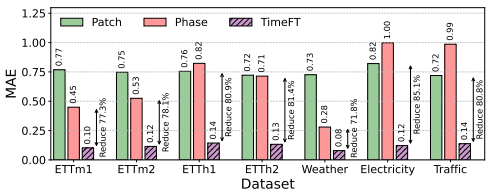


Figure 4: **Reconstruction performance of different tokens types** across six commonly used datasets. The details are provided in Table 17.

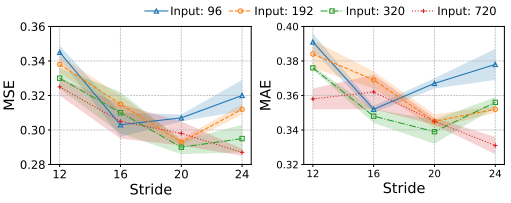


Figure 5: **Hyperparameter sensitivity analysis** of tokenization stride S for different input lengths N. Metric: *Left:* MSE, *Right:* MAE.

datasets in the time domain. This experiment can be viewed as a statistical summary of Figure 2 and the visualization section F. Due to Appendices B and C, the reconstructions from tokens to the reconstructed signals here are theoretically an invertible and lossless transformation. We ensure that all hyperparameters remain identical in this experiment. As shown in Figure 4, TimeFT reduces more than 80% reconstruction error in many datasets, which is shown by the percentage of "Reduce" on the right side of each bar.. This phenomenon arises because patch and phase tokenization introduce data distortion, causing the reconstructed sequence to differ substantially from the original sequence. In contrast, TimeFT achieves nearly distortion-free tokenization. The details are provided in Table 17 and the reconstruction methods are provided in Appendix C.

4.7 Sensitivity Analysis

We further examine the impact of the tokenization stride S on model performance across various input lengths in the ETTm1 dataset. As shown in Figure 5, the results reveal a significant synergy between the stride and the input scale, where the optimal configuration changes according to the input length. For shorter input sequences such as $N = 96$, smaller strides ensure sufficient coverage of temporal details. However, as the input length increases to $N = 720$, the model demonstrates a clear preference for larger strides. Since the stride determines the Nyquist frequency, a smaller stride results in a higher Nyquist frequency and longer tokens, while also leading to fewer tokens (downsampling). This trade-off affects the model's performance, making it necessary to choose an appropriate stride for the input. Moreover, longer input sequences generally result in better performance.

5 Related Work

Existing time series tokenizers can be classified into two fundamental categories: patch tokens and phase tokens. We consider only the partitioning operation here without the embedding step. Quantization methods such as "bins" [28, 29] and some frequency methods [30–33] are not discussed because they are irrelevant to our topic (§B.5). According to the modeling paradigm, token-based methods fall into the following three categories.

For deep learning models, PatchTST[2] introduces a patch-based approach to time series analysis, establishing patching as a key component of time series modeling. Subsequently, PITS [34], PatchMLP [3] and xPatch [21] demonstrate that patch tokens can improve performance. Beyond static patching [35–37], some dynamic patching methods [20, 38–41] have also been proposed for time series tokenization. SRSNet [42] leverages information from contextual time series to improve the forecasting performance of patch-based models. TimeMosaic [19] employs a lightweight classifier to determine the length of patches. Additionally, PhaseFormer [4] introduces time series phase tokens, which can be viewed as an approximate transpose of patch tokenization.

For pre-trained tokenizers, TOTEM [43] divides the time series into patches and maps them to a discrete embedding using a pre-trained model. EntroPE [14] employs a pre-trained entropy model to determine optimal patch boundaries. These methods still belong to the patch token class.

For pre-trained foundation models, recent state-of-the-art TSFMs [44], including Chronos-2 [45], Sundial [46], TiRex [47], Moirai [48], and TimeFM [49], have adopted patch tokenization as a preprocessing step. Tokenization has become a fundamental component of TSFMs.

6 Conclusion

We identify the inherent spectral distortion in the dominant time series tokenization paradigm and derive its boundary conditions for distortion-free tokenization. We propose TimeFT, an approach obtains time series tokens in a nearly distortion-free manner through frequency-domain partitioning, shifting, and Nyquist sampling. TimeFT is parameter-free, lightweight, and plug-and-play tokenizer for time series data. Extensive experiments on various tasks showcase that TimeFT outperforms existing practices and yields nearly 10% improvement as a plug-in module in many cases.

7 Acknowledge

This work was supported by the NVIDIA Academic Grant in Higher Education and Developer program.

References

[1] X. Kong, Z. Chen, W. Liu, K. Ning, L. Zhang, S. Muhammad Marier, Y. Liu, Y. Chen, and F. Xia, "Deep learning for time series forecasting: a survey," *International Journal of Machine Learning and Cybernetics*, pp. 1–34, 2025.

[2] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[3] P. Tang and W. Zhang, "Unlocking the power of patch: Patch-based mlp for long-term time series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 39, no. 12, 2025, pp. 12 640–12 648.

[4] Y. Niu, J. Deng, and Y. Tong, "Phaseformer: From patches to phases for efficient and effective time series forecasting," in *International Conference on Learning Representations (ICLR)*, 2026.

[5] Y. Cheng, "TimeCAT: Hierarchical context-aware transformer with dynamic grouping for time series forecasting," 2024. [Online]. Available: https://openreview.net/forum?id=0ziGSo4uWp

[6] X. Han, X. Zhang, Y. Wu, Z. zhang, Z. Deng, and Z. Wu, "Patchcat: Rethinking temporal tokenization in time series forecasting," 2026. [Online]. Available: https://openreview.net/forum?id=vG9vqUwjbu

[7] K. Yamamoto and K. Okusa, "Aape: Aliasing-aware patch embedding for self-supervised audio representation learning," *arXiv preprint arXiv:2512.03637*, 2025.

[8] H. Goksu, "Hi-wavetst: A hybrid high-frequency wavelet-transformer for time-series classification," *arXiv preprint arXiv:2511.01254*, 2025.

[9] G. Zhang, Y. Zhang, and Z. Tan, "Protohpe: Prototype-guided high-frequency patch enhancement for visible-infrared person re-identification," in *Proceedings of the 31st ACM international conference on multimedia*, 2023, pp. 944–954.

[10] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ: Pearson, 2010.

[11] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, 2021.

[12] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[13] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[14] S. Abeywickrama, E. Eldele, M. Wu, X. Li, and C. Yuen, "Entrope: Entropy-guided dynamic patch encoder for time series forecasting," *arXiv preprint arXiv:2509.26157*, 2025.

[15] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[16] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[17] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–42, 2024.

[18] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "ADBench: Anomaly detection benchmark," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. [Online]. Available: https://openreview.net/forum?id=foA_SFQ9zo0

[19] K. Ding, F. Fan, C. Hou, Z. Wang, L. Wang, Z. Yang, and J. 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*, vol. 40, no. 25, 2026, pp. 20 790–20 798.

[20] Y. Hu, G. Zhang, P. Liu, D. Lan, N. Li, D. Cheng, T. Dai, S.-T. Xia, and S. Pan, "Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=490VcNtjh7

[21] A. Stitsyuk and J. Choi, "xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 19, 2025, pp. 20 601–20 609.

[22] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations (ICLR)*, 2024.

[23] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "Lightts: Lightweight time series classification with adaptive ensemble distillation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, 2023.

[24] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *arXiv preprint arXiv:2205.13504*, 2022.

[25] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.

[26] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. C. H. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," 2022. [Online]. Available: https://arxiv.org/abs/2202.01381

[27] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

[28] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor *et al.*, "Chronos: Learning the language of time series," *Transactions on Machine Learning Research*, vol. 2024, 2024.

[29] T. Yin, J. Wang, Y. Ma, H. Wang, C. Wang, Y. Zhao, M. Liu, and W. Shen, "Apollo-forecast: Overcoming aliasing and inference speed challenges in language models for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 21, 2025, pp. 22 173–22 181.

[30] Z. An, J. You, J. Li, Y. Tang, W. Li, H. Du, and S. Du, "Fredn: Spectral disentanglement for time series forecasting via learnable frequency decomposition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 24, 2026, pp. 19 623–19 631.

[31] J. Li, W. Ma, and Y. Liu, "Modeling time series as text sequence a frequency-vectorization transformer for time series forecasting," 2024. [Online]. Available: https://openreview.net/forum?id=N1cjy5iznY

[32] L. Qianyang, Z. Xingjun, W. Shaoxun, and W. Jia, "Dpwmixer: Dual-path wavelet mixer for long-term time series forecasting," *arXiv preprint arXiv:2512.02070*, 2025.

[33] J. Bai and Y. Kawahara, "Dualformer: Time-frequency dual domain learning for long-term time series forecasting," *arXiv preprint arXiv:2601.15669*, 2026.

[34] S. Lee, T. Park, and K. Lee, "Learning to embed time series patches independently," in *12th International Conference on Learning Representations, ICLR 2024*, 2024.

[35] J. Kim, K. Park, S. Yun, and S. Lee, "Ppt: Patch order do matters in time series pretext task," in *The Thirteenth International Conference on Learning Representations*, 2025.

[36] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2023, pp. 459–469.

[37] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, "Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting," in *ICLR 2024: The Twelfth International Conference on Learning Representations.*, 2024.

[38] S. K. Ankireddy, N. Seleznev, N. H. Nguyen, Y. Wu, S. Kumar, F. Huang, and C. B. Bruss, "Timesqueeze: Dynamic patching for efficient time series forecasting," *arXiv preprint arXiv:2603.11352*, 2026.

[39] X. Liu, X. Qiu, X. Wu, Z. Li, C. Guo, J. Hu, and B. Yang, "Rethinking irregular time series forecasting: A simple yet effective baseline," in *AAAI*, 2026.

[40] J. Xiao, Y. Chen, G. Cong, W. Nejdl, and S. Gottschalk, "Flextsf: A flexible forecasting model for time series with variable regularities," *arXiv preprint arXiv:2410.23160*, 2024.

[41] Y. Luo, B. Zhang, Z. Liu, and Q. Ma, "Hi-patch: Hierarchical patch GNN for irregular multivariate time series," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=nBgQ66iEUu

[42] X. Wu, X. Qiu, H. Cheng, Z. Li, J. Hu, C. Guo, and B. Yang, "Enhancing time series forecasting through selective representation spaces: A patch perspective," in *NeurIPS*, 2025.

[43] S. J. Talukder, Y. Yue, and G. Gkioxari, "TOTEM: TOkenized time series EMbeddings for general time series analysis," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=QlTLkH6xRC

[44] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6555–6565.

[45] A. F. Ansari, O. Shchur, J. Küken, A. Auer, B. Han, P. Mercado, S. S. Rangapuram, H. Shen, L. Stella, X. Zhang *et al.*, "Chronos-2: From univariate to universal forecasting," *arXiv preprint arXiv:2510.15821*, 2025.

[46] Y. Liu, G. Qin, Z. Shi, Z. Chen, C. Yang, X. Huang, J. Wang, and M. Long, "Sundial: A family of highly capable time series foundation models," *arXiv preprint arXiv:2502.00816*, 2025.

[47] A. Auer, P. Podest, D. Klotz, S. Böck, G. Klambauer, and S. Hochreiter, "TiRex: Zero-Shot Forecasting Across Long and Short Horizons with Enhanced In-Context Learning," in *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://arxiv.org/abs/2505.23719

[48] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers," *arXiv preprint arXiv:2402.02592*, 2024.

[49] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 10 148–10 167.

[50] I. Zexer and O. Azencot, "XCTFormer: Leveraging cross-channel and cross-time dependencies for enhanced time-series analysis," *Transactions on Machine Learning Research*, 2026. [Online]. Available: https://openreview.net/forum?id=TEfyR4t0Tw

[51] L. Götz, M. Kollovieh, S. Günnemann, and L. Schwinn, "Efficient time series processing for transformers and state-space models through token merging," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=Y2FyXF5hRe

[52] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024.

[53] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International conference on learning representations*, 2021.

A Motivation Explanations and Theoretical Insights

A.1 Theoretical Contribution and Insights

In this work, our theoretical contribution is the first systematic introduction of Digital Signal Processing (DSP) tools to time series tokenization analysis, to address the inherent distortion problems in previous human intuition approaches such as patch and phase tokenization. The tokenization distortion phenomenon, the distortion boundary conditions, and TimeFT are all proposed for the first time. Specifically:

- For time-series tokenization, we establish the first frequency-based theoretical framework that precisely derives the frequency-domain effects of tokenization and reveals the root causes of distortions in existing tokenization methods.
- Based on the above analysis, for the first time we provide the exact distortion-free boundary conditions for patch and phase tokenization, and accordingly design TimeFT to eliminate these distortions.

A.2 Why Distortion in Individual Token Degrades Model Performance

The distortion occurs during the tokenization step, before the tokens are fed into self-attention. Consequently, every individual token representation is already corrupted at the input level. While self-attention can mix information across tokens, it cannot recover the discarded information because that information was permanently lost prior to the attention layers. It is a matter of irreversible information loss at the representation.

This distortion is an inherent characteristic of tokenization and holds for any token. This leads to an irreversible alteration of the distribution of the input data, as shown in Appendix F, which causes deep learning models to receive corrupted information. Therefore, distortion in individual tokens degrades model performance.

A.3 Why Distortion should be Addressed in Tokenization Stage

While existing deep learning models for time series can capture complex dependencies thanks to the Universal Approximation Theorem, this does not imply that they are immune to input corruption. Distortions such as spectral leakage or aliasing occur during the tokenization process, i.e., before the data is input into the model. This discrepancy directly harms the model's generalization ability, as the learned patterns are tied to the distorted input rather than the underlying data relationship.

General deep learning models are not designed to correct systematic input distortions, and they are function approximators that fit the distribution they are trained on. If the training distribution is already biased, the model will simply converge to a biased solution. As shown in our experiments (Table 5), even when the backbone architecture remains unchanged, replacing the tokenizer from patch/phase to TimeFT leads to significant performance improvements.

B Frequency Domain Theoretical Analysis and Boundary Conditions

B.1 Patch Tokenization

Let $x[n], n \in \{0, 1, \ldots, N-1\}$, be a discrete time sequence of finite length N. The patch tokenization in Equation 1 can be equivalent to the operation of rectangular windowing. A rectangular window $w_L[n; n_0]$ of length L starting at the index n_0 is defined as $w_L[n; n_0] = 1$ for $n_0 \leq n \leq n_0 + L - 1$ and 0 otherwise. The windowed sequence is $x_w[n] = x[n] \cdot w_L[n; n_0]$, and the sliced sequence of length L is obtained by extracting the segment in the window and re-indexing it from zero, i.e., $y[m] = x_w[n_0 + m]$ for $m = 0, \ldots, L-1$. We analyze the spectral consequences of these operations within the continuous frequency domain provided by the discrete time Fourier transform (DTFT). The DTFT of $x[n]$ is

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n} \tag{14}$$

where $\omega \in [-\pi, \pi)$ denotes the normalized angular frequency in radians. Note that DTFT of a sequence $x[n]$ of length N is well-defined, since by treating the values at undefined n as 0, the sequence naturally extends to an infinite-length sequence, and the infinite summation in DTFT reduces to a finite sum. The DTFT of the rectangular window $w_L[n; n_0]$ is derived as follows:

$$\begin{aligned} W_L(e^{j\omega}; n_0) &= \sum_{n=n_0}^{n_0+L-1} e^{-j\omega n} \\ &= e^{-j\omega n_0} \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} \\ &= e^{-j\omega n_0} \cdot \frac{2je^{-j\omega L/2} \sin(\omega L/2)}{2je^{-j\omega/2} \sin(\omega/2)} \\ &= e^{-j\omega n_0}\, e^{-j\omega(L-1)/2}\, \frac{\sin(\omega L/2)}{\sin(\omega/2)} \\ &= e^{-j\omega\left(n_0 + \frac{L-1}{2}\right)} \cdot \frac{\sin(\omega L/2)}{\sin(\omega/2)} \end{aligned} \tag{15}$$

where the first step expresses the windowed sum as a finite geometric series. Factor out the term corresponding to the starting index by letting $n = n_0 + k,\ k = 0, 1, \ldots, L-1$:

$$\sum_{k=0}^{L-1} e^{-j\omega(n_0+k)} = e^{-j\omega n_0} \sum_{k=0}^{L-1} \left(e^{-j\omega}\right)^k \tag{16}$$

The sum on the right is a finite geometric series with the first term 1, the common ratio $r = e^{-j\omega}$, and L terms. Using the geometric sum formula $\sum_{k=0}^{L-1} r^k = \frac{1-r^L}{1-r}$ (for $r \neq 1$), we obtain

$$\sum_{k=0}^{L-1} \left(e^{-j\omega}\right)^k = \frac{1 - \left(e^{-j\omega}\right)^L}{1 - e^{-j\omega}} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} \tag{17}$$

Substituting this back gives the closed form

$$W_L(e^{j\omega}; n_0) = e^{-j\omega n_0} \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} \tag{18}$$

In Equation 15, the factor $D_L(\omega) \triangleq \frac{\sin(\omega L/2)}{\sin(\omega/2)}$ is the Dirichlet kernel, a periodic sinc-like function whose main lobe has a null-to-null width of $4\pi/L$ and whose highest side lobe is approximately -13 dB below the main peak. The complex exponential factor $e^{-j\omega(n_0+(L-1)/2)}$ imparts a linear phase shift, but since we are concerned with magnitude spectral characteristics, this phase term does not affect the analysis of problems. The windowed sequence $x_w[n] = x[n] \cdot w_L[n; n_0]$ is the pointwise product of the original sequence and the rectangular window. By the DTFT multiplication property, multiplication in the time domain corresponds to *circular convolution* of their DTFTs over a period of length 2π:

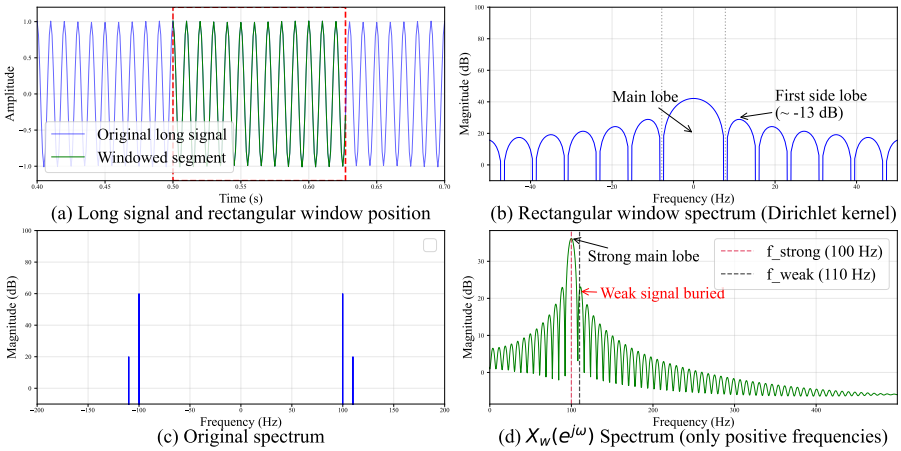
$$X_w(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})\, W_L(e^{j(\omega-\theta)}; n_0)\, d\theta \tag{19}$$

where $X_w(e^{j\omega})$ is the DTFT of sequence $x_w[n]$. We then yield the following:

$$\begin{aligned} X_w(e^{j\omega}) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})\, e^{-j(\omega-\theta)\left(n_0 + \frac{L-1}{2}\right)} \frac{\sin\left((\omega-\theta)L/2\right)}{\sin\left((\omega-\theta)/2\right)}\, d\theta \\ &= e^{-j\omega\left(n_0 + \frac{L-1}{2}\right)} \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[X(e^{j\theta})\, e^{j\theta\left(n_0 + \frac{L-1}{2}\right)}\right] D_L(\omega - \theta)\, d\theta \end{aligned} \tag{20}$$

where $D_L(\omega-\theta) = \frac{\sin((\omega-\theta)L/2)}{\sin((\omega-\theta)/2)}$ is the Dirichlet kernel shifted to θ. Equation 20 reveals that, up to a global linear phase shift, $X_w(e^{j\omega})$ is the convolution of a phase-rotated version of $X(e^{j\omega})$ with the Dirichlet kernel $D_L(\cdot)$. The distortion mainly takes the following three forms: (1) **spectral leakage:** Slowly decaying sidelobes of $D_L(\cdot)$ spread the energy from each frequency component across the entire spectrum, potentially masking weaker components. (2) **resolution limit:** The main lobe of $D_L(\cdot)$ has a width of $4\pi/L$, which limits the ability to distinguish adjacent spectral peaks. Components closer than half this interval merge into a single broadened lobe, irrespective of the original sequence length N. Moreover, the windowed sequence is further truncated to length L, so it **discards information** beyond length L while inheriting the identical leakage and resolution penalties. Since discarding information outside the window is intuitive in the time domain, we next analyze the underlying causes of spectral leakage and resolution limit.

Spectral Leakage. For $X_w(e^{j\omega})$ in Equation 20, the linear phase factor $e^{-j\omega(n_0+(L-1)/2)}$ does not affect the magnitude spectrum, so the magnitude distortion is entirely determined by the convolution with the Dirichlet kernel. This convolution is the origin of *spectral leakage*: each frequency component of the original signal is not confined to its ideal location but spreads to neighboring frequencies due to the side lobes of $D_L(\omega)$, as shown in Figure 6:



(a) Long signal and rectangular window position
(b) Rectangular window spectrum (Dirichlet kernel)
(c) Original spectrum
(d) $X_w(e^{j\omega})$ Spectrum (only positive frequencies)

Figure 6: Illustration of spectral leakage and weak-signal masking caused by rectangular window truncation. (a) Rectangular window truncation of a time-domain signal. (b) Spectral characteristics of the Dirichlet kernel corresponding to the rectangular window. (c) Original spectrum containing adjacent strong and weak signals. (d) Masking of weak signals caused by sidelobe leakage after windowing. The decibel (dB) is a logarithmic unit, so a small difference can represent a large change.

To illustrate the effect concretely with a typical signal from practical time series, consider a real sinusoid infinite-duration sequence $x[n] = \cos(\omega_0 n)$. Using Euler's formula, $x[n] = \frac{1}{2}e^{j\omega_0 n} + \frac{1}{2}e^{-j\omega_0 n}$, whose DTFT consists of two impulses at $\pm\omega_0$. Since the DTFT of the window sequence $x_w[n]$ is the convolution of $X(e^{j\omega})$ with $W_L(e^{j\omega}; n_0)$, linearity allows us to treat each complex exponential. Substituting the rectangular window DTFT from Equation 19 yields the following:

$$\begin{aligned} X_w(e^{j\omega}) &= \frac{1}{2}W_L(e^{j(\omega-\omega_0)}; n_0) + \frac{1}{2}W_L(e^{j(\omega+\omega_0)}; n_0) \\ &= \frac{1}{2}e^{-j(\omega-\omega_0)\left(n_0+\frac{L-1}{2}\right)}D_L(\omega-\omega_0) + \frac{1}{2}e^{-j(\omega+\omega_0)\left(n_0+\frac{L-1}{2}\right)}D_L(\omega+\omega_0) \end{aligned} \tag{21}$$

When ω_0 is sufficiently far from 0 and π, the main lobes centered at $\pm\omega_0$ are well separated, and the cross terms between them become negligible. In this case, the magnitude spectrum is approximately:

$$|X_w(e^{j\omega})| \approx \frac{1}{2}\left|D_L(\omega-\omega_0)\right| + \frac{1}{2}\left|D_L(\omega+\omega_0)\right| \tag{22}$$

which explicitly shows that each ideal spectral line is replaced by a broadened main lobe accompanied by oscillatory side lobes. The factor $1/2$ accounts for the splitting of the sinusoid's energy between positive and negative frequencies. The side lobes of $|D_L(\cdot)|$ slowly decay, with the highest side lobe only about 13 dB below the main peak. Consequently, a weak sinusoidal component located near a strong one may be completely masked by the side-lobe energy of the stronger component.

Resolution Limit. To understand the resolution limit imposed by the rectangular window, consider two complex exponentials $e^{j\omega_1 n}$ and $e^{j\omega_2 n}$ with $\omega_1 < \omega_2$. After windowing with $w_L[n; n_0]$, the DTFT of each component becomes a shifted Dirichlet kernel $D_L(\omega - \omega_i)$ (up to a phase factor). The magnitude spectrum therefore contains two main lobes centered at ω_1 and ω_2. According to the **Rayleigh criterion** commonly adopted in spectral analysis, two peaks are resolved when their separation is at least half the null-to-null width of the main lobe. Since the main lobe of $D_L(\omega)$ has a null-to-null width of $4\pi/L$, the critical separation is $2\pi/L$. Hence, two sinusoidal components can be distinguished in the DTFT magnitude spectrum only if

$$|\omega_1 - \omega_2| \geq \frac{2\pi}{L} \tag{23}$$

If the separation is smaller, the two main lobes merge into a single broadened peak, and the components become indistinguishable regardless of the continuous nature of the DTFT. This is a trade-off relationship, reducing the window length L improves temporal localization but inevitably degrades frequency resolution and exacerbates spectral leakage.

First, we analyze whether the transformation from $x_w[n]$ to $y[n]$ introduces the resolution limit problem. The patch tokenization extracts the windowed segment and re-indexes it to start at zero, yielding $y[m] = x_w[n_0 + m]$ for $m = 0, 1, \ldots, L-1$. We now derive its DTFT:

$$\begin{aligned} Y(e^{j\omega}) &= \sum_{m=0}^{L-1} y[m]\, e^{-j\omega m} \\ &= \sum_{n=n_0}^{n_0+L-1} x_w[n]\, e^{-j\omega(n-n_0)} \qquad (24) \\ &= e^{j\omega n_0} \sum_{n=n_0}^{n_0+L-1} x_w[n]\, e^{-j\omega n} \\ &= e^{j\omega n_0}\, X_w(e^{j\omega}) \qquad (25) \end{aligned}$$

Equation 25 shows that slicing multiplies the DTFT of the windowed sequence by $e^{j\omega n_0}$, which corresponds to a time advance of n_0 samples. This shows that cutting merely applies a pure time advance, leaving the magnitude spectrum unchanged. Substituting the expression for $X_w(e^{j\omega})$ from Equation 20:

$$X_w(e^{j\omega}) = e^{-j\omega\left(n_0 + \frac{L-1}{2}\right)} \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})\, e^{j\theta\left(n_0 + \frac{L-1}{2}\right)} D_L(\omega - \theta)\, d\theta \tag{26}$$

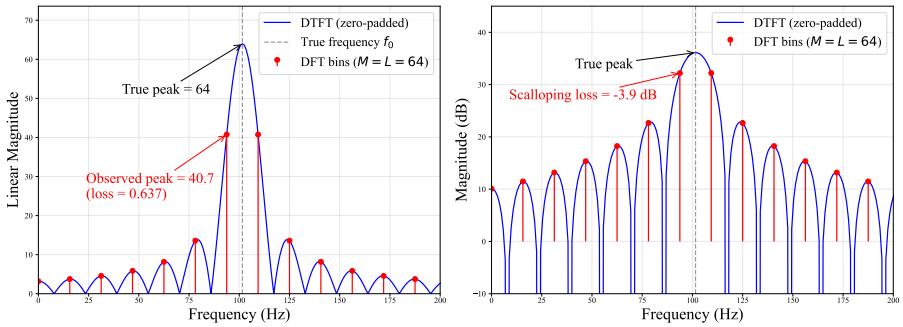
We obtain

$$\begin{aligned} Y(e^{j\omega}) &= e^{j\omega n_0}\, e^{-j\omega\left(n_0 + \frac{L-1}{2}\right)} \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})\, e^{j\theta\left(n_0 + \frac{L-1}{2}\right)} D_L(\omega - \theta)\, d\theta \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta})\, e^{j\theta n_0}\, e^{-j(\omega-\theta)\frac{L-1}{2}} D_L(\omega - \theta)\, d\theta \qquad (27) \end{aligned}$$

From Equation 25 it immediately follows that the magnitude spectra are identical:

$$|Y(e^{j\omega})| = |X_w(e^{j\omega})| \tag{28}$$

Thus, the step $y[m] = x_w[n_0 + m]$ for $m = 0, \ldots, L-1$ does not alter the leak pattern or the frequency resolution; it simply realigns the temporal origin. All subsequent discussion of spectral leakage and resolution applies equally to the sliced sequence $y[m]$ that constitutes the patch token.



(a) Picket-fence effect (worst case, $\delta = 0.5$) (b) Picket-fence effect (dB scale)

Figure 7: Illustration of the picket-fence effect. (a) Picket-fence effect at linear magnitude. (b) Picket-fence effect at dB magnitude.

Although DTFT $Y(e^{j\omega})$ is a continuous function of ω, any practical spectral analysis must evaluate it at a finite set of discrete frequencies. When we sample the DTFT at M uniformly spaced frequencies $\omega_k = 2\pi k/M$, $k = 0, \ldots, M-1$, we obtain a discrete representation that is subject to the **picket-fence effect**. When viewing a landscape through a picket fence, one can see only the portions of the scene that align with the gaps between the slats, while the rest remains hidden. In the spectral context, the continuous DTFT is the "landscape," and the M frequency samples ω_k are the "gaps." If a signal component's true frequency lies exactly on one of these sample points, its full amplitude is observed; if it falls between two points, the observed value at the nearest sample is only an attenuated version of the actual spectral peak. The severity of this attenuation depends on how far the true frequency deviates from the nearest sampling point.

To quantify this effect, consider a single complex sinusoid of frequency ω_0 after rectangular windowing. Its continuous DTFT is a shifted Dirichlet kernel, i.e., $Y(e^{j\omega}) = e^{j\phi(\omega)} D_L(\omega - \omega_0)$, where $\phi(\omega)$ is a linear phase term. The true magnitude peak occurs exactly at $\omega = \omega_0$ and is equal to $|D_L(0)| = L$. Suppose that the frequency ω_0 falls between two discrete frequency grid points, written as $\omega_0 = 2\pi(k_0 + \delta)/M$, where k_0 is the index of the nearest grid point and $0 < |\delta| < 0.5$ measures the fractional bin offset. The discrete spectrum is evaluated only on the fixed grid $\omega_k = 2\pi k/M$. Hence, the maximum amplitude that we can observe among the discrete bins is

$$|Y(e^{j\omega_{k_0}})| = |D_L(2\pi\delta/M)| = \left| \frac{\sin(\pi\delta L/M)}{\sin(\pi\delta/M)} \right| \tag{29}$$

which is strictly smaller than L whenever $\delta \neq 0$. The ratio of the observed peak to the true peak,

$$\frac{|Y(e^{j\omega_{k_0}})|}{|D_L(0)|} = \frac{1}{L} \left| \frac{\sin(\pi\delta L/M)}{\sin(\pi\delta/M)} \right| \tag{30}$$

This is known as the scalloping loss or picket-fence loss. For example, as shown in Figure 7, in the worst-case offset $\delta = 0.5$ with $M = L$ (no zero-padding), this ratio is about $|\sin(\pi/2)/(\pi/2)| \approx 0.637$, implying an amplitude reduction of more than 36% and nearly 4 dB compared to the true peak. The effect can be mitigated by increasing M, the number of frequency samples over the same interval, which is usually achieved by zero padding the time-domain sequence $y[m]$ before computing the DFT. Zero-padding appends zeros to $y[m]$, resulting in a denser set of DTFT evaluation points ω_k without altering the underlying continuous spectrum $Y(e^{j\omega})$.

B.2 Phase Tokenization

Let $x[n]$ be a finite-length discrete-time sequence of length N, and let $S \geq 2$ be an integer downsampling factor. The downsampled sequence is $y[m] = x[mS],\ m = 0, 1, \ldots, \lceil N/S \rceil - 1$. We

decompose this operation into two stages: (i) zero insertion, and (ii) time compression. To analyze the spectral consequences, it is convenient to introduce an auxiliary zero-filled sequence:

$$w[n] = \begin{cases} x[n], & n = kS,\ k \in \mathbb{Z}, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

which is simply $x[n]$ sampled on every S-th index and padded with zeros elsewhere. The relation between $w[n]$ and the final downsampled signal $y[m]$ is

$$y[m] = w[mS],\ m \in \mathbb{Z} \tag{32}$$

The sequence $w[n]$ can be expressed as the product of $x[n]$ with a periodic impulse sequence

$$p[n] = \sum_{k=-\infty}^{\infty} \delta[n - kS] \tag{33}$$

where $\delta[n]$ denotes the unit sample ($\delta[0] = 1$ and $\delta[n] = 0$ for $n \neq 0$). That is,

$$w[n] = x[n]\, p[n] \tag{34}$$

The DTFT of the impulse sequence $p[n]$ is obtained via the Poisson summation formula. The sequence $p[n] = \sum_{k=-\infty}^{\infty} \delta[n - kS]$ is periodic with period S. Within one period $n = 0, 1, \ldots, S-1$, we have $p[0] = 1$ and $p[n] = 0$ for $n \neq 0$. Its discrete Fourier series coefficients are therefore

$$P_c[k] = \frac{1}{S} \sum_{n=0}^{S-1} p[n]\, e^{-j2\pi kn/S} = \frac{1}{S},\ k = 0, 1, \ldots, S-1 \tag{35}$$

Therefore,

$$p[n] = \sum_{k=0}^{S-1} P_c[k]\, e^{j2\pi kn/S} = \frac{1}{S} \sum_{k=0}^{S-1} e^{j2\pi kn/S} \tag{36}$$

Taking the DTFT of both sides,

$$\begin{aligned} P(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} p[n]\, e^{-j\omega n} = \frac{1}{S} \sum_{k=0}^{S-1} \sum_{n=-\infty}^{\infty} e^{-j(\omega - 2\pi k/S)n} \\ &= \frac{2\pi}{S} \sum_{k=0}^{S-1} \sum_{r=-\infty}^{\infty} \delta\!\left(\omega - \frac{2\pi(k + rS)}{S}\right) \end{aligned} \tag{37}$$

where we have used $\sum_{n=-\infty}^{\infty} e^{-j\alpha n} = 2\pi \sum_{r=-\infty}^{\infty} \delta(\alpha - 2\pi r)$. Letting $m = k + rS$, the double sum collapses to a single sum over all integers m:

$$P(e^{j\omega}) = \frac{2\pi}{S} \sum_{m=-\infty}^{\infty} \delta\!\left(\omega - \frac{2\pi m}{S}\right) \tag{38}$$

Due to the 2π-periodicity of the DTFT, only S distinct impulses fall within any interval of length 2π. For the principal interval $[-\pi, \pi)$ one may restrict the index set to $m = 0, 1, \ldots, S-1$ when operating inside the convolution integral. Multiplication in the time domain corresponds to circular convolution in the frequency domain:

$$W(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})\, P\big(e^{j(\omega-\theta)}\big)\, d\theta \tag{39}$$

Equations 38 and 39 give

$$\begin{aligned} W(e^{j\omega}) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})\, \frac{2\pi}{S} \sum_{m=-\infty}^{\infty} \delta\left(\omega - \theta - \frac{2\pi m}{S}\right) d\theta \\ &= \frac{1}{S}\sum_{m=-\infty}^{\infty} \int_{-\pi}^{\pi} X(e^{j\theta})\, \delta\left(\omega - \theta - \frac{2\pi m}{S}\right) d\theta \end{aligned} \tag{40}$$

For a fixed $\omega \in (-\pi, \pi]$, the argument of the delta function is zero when $\theta = \omega - 2\pi m/S$. The integral is not zero only if this θ is within the integration limits $[-\pi, \pi)$. Because $X(e^{j\theta})$ is 2π-periodic, exactly S distinct values of m contribute (conventionally $m = 0, 1, \ldots, S-1$ after reindexing). Exploiting the sifting property of the Dirac delta, we obtain:

$$W(e^{j\omega}) = \frac{1}{S}\sum_{k=0}^{S-1} X\left(e^{j(\omega - 2\pi k/S)}\right) \tag{41}$$

Equation 41 shows that the zero-filled sequence possesses a spectrum consisting of S shifted replicas of the original spectrum $X(e^{j\omega})$ (each scaled by $1/S$), packed into the interval $[-\pi, \pi)$. From Equation 32 we have $y[m] = w[mS]$. Its DTFT is

$$Y(e^{j\omega}) = \sum_{m=-\infty}^{\infty} y[m]\, e^{-j\omega m} = \sum_{m=-\infty}^{\infty} w[mS]\, e^{-j\omega m} \tag{42}$$

Setting $n = mS$, we see that the sum runs over all integer multiples of S. Because $w[n] = 0$ for n not a multiple of S, we may sum over all integers n:

$$Y(e^{j\omega}) = \sum_{n=-\infty}^{\infty} w[n]\, e^{-j(\omega/S)n} = W\left(e^{j\omega/S}\right) \tag{43}$$

Thus, the time-compression S in the time domain expands the frequency axis by a factor S; i.e., the DTFT of $y[m]$ is the same as that of $w[n]$ but evaluated at ω/S. Substituting Equation 41 into Equation 43 yields the DTFT of the downsampled sequence:

$$Y(e^{j\omega}) = \frac{1}{S}\sum_{k=0}^{S-1} X\left(e^{j(\omega/S - 2\pi k/S)}\right) \tag{44}$$

This is a fundamental description of downsampling in the frequency domain. The original spectrum $X(e^{j\omega})$ is first compressed by the factor S (argument ω/S) and then shifted by multiples of $2\pi/S$. If $X(e^{j\omega})$ contains non-zero energy for frequencies $|\omega| \geq \pi/S$, the shifted replicas will overlap, causing high frequency components to irreversibly fold into lower frequencies, which is **spectral aliasing**. To avoid aliasing, the replicas must remain distinct.

Equation 44 shows both spectral stretching and the potential for aliasing. Consider first the term corresponding to $k = 0$, namely $\frac{1}{S}X(e^{j\omega/S})$. Here, the original spectrum $X(e^{j\omega})$ undergoes a scaling of the frequency axis by the factor S. A frequency component originally located at ω_0 in $x[n]$ will appear at $S\omega_0$ in $Y(e^{j\omega})$. This expansion of the frequency axis is called spectral stretching. Consequently, a narrowband signal in $x[n]$ spreads over a proportionally wider bandwidth in the downsampled sequence, and its amplitude is reduced by the factor S.

The remaining $S-1$ terms of $k = 1, \ldots, S-1$, have the form $\frac{1}{S}X\big(e^{j(\omega/S - 2\pi k/S)}\big)$. These arise from the shifted replicas created during the intermediate zero-insertion step. In the spectrum of

the zero-filled sequence $W(e^{j\omega})$ in Equation 41, these replicas are centred at $\omega = 2\pi k/S$. The time-compression step $y[m] = w[mS]$ corresponds to the frequency mapping $Y(e^{j\omega}) = W(e^{j\omega/S})$, which expands the frequency axis by S. Hence, a replica originally centered at $2\pi k/S$ is mapped to the new center:

$$S \cdot \frac{2\pi k}{S} = 2\pi k \tag{45}$$

Because the DTFT is 2π-periodic, all these centers fall at integer multiples of 2π, which are exactly the boundaries of the principal interval $[-\pi, \pi)$. Thus, after the downsampling operation, every one of the S replicas is assigned to the same baseband interval.

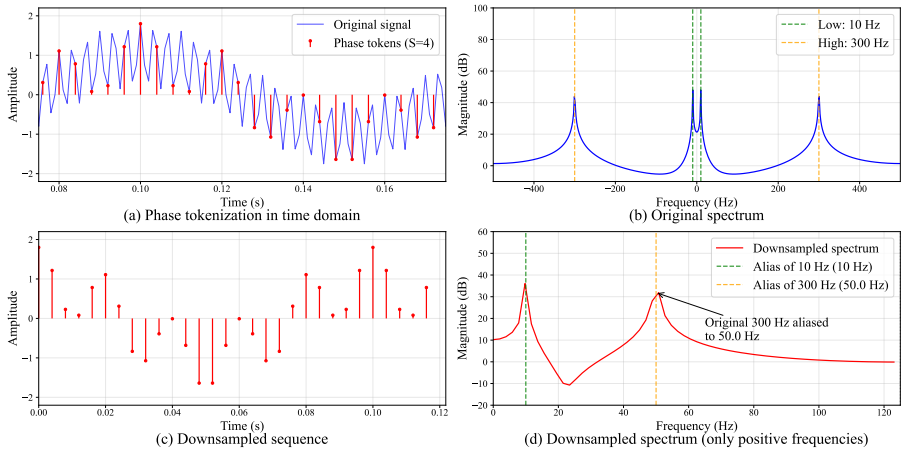


Figure 8: Illustration of spectral aliasing in the frequency domain. When the original spectrum is not confined to $(-\pi/S, \pi/S)$, high frequency components appear as spurious low frequency components.

Whether these replicas overlap destructively depends on the bandwidth of $X(e^{j\omega})$. If $X(e^{j\omega})$ contains non-zero energy for $|\omega| \geq \pi/S$, then the $k = 0$ baseband replica already extends beyond $|\omega| \leq \pi$ after stretching, and the $k \neq 0$ terms bring additional copies of the spectrum that intrude into the same region. More precisely, a frequency ω in $Y(e^{j\omega})$ receives contributions from all k for which $\omega/S - 2\pi k/S$ falls within the support of $X(e^{j\omega})$. As shown in Figure 8, when the original spectrum is not confined to $(-\pi/S, \pi/S)$, there exist frequencies at which two or more replicas contribute, causing high frequency components to appear as spurious low frequency components. This irreversible superposition is spectral aliasing. Because the overlapping terms all share the same frequency in $Y(e^{j\omega})$, the aliased components cannot be separated after downsampling, leading to permanent distortion.

To avoid aliasing entirely, the support of $X(e^{j\omega})$ must be a subset of $(-\pi/S, \pi/S)$. In other words, the maximum normalized frequency $\omega_{\max}$ present in $x[n]$ must satisfy

$$\omega_{\max} \leq \frac{\pi}{S} \tag{46}$$

When this condition is satisfied, each frequency ω in the principal interval receives a contribution from exactly one value of k (the one for which $\omega/S - 2\pi k/S$ lies within the support of X). The replicas are then placed side-by-side without overlapping, and the original signal can in principle be recovered from the downsampled version.

The condition can be expressed in terms of physical frequencies by relating ω to the underlying continuous-time signal from which $x[n]$ was obtained. Assume $x[n] = x_c(nT)$, with the sampling period $T = 1/f_s$ and f_s the original sampling rate. The normalized frequency is $\omega = 2\pi f/f_s$. If the analog signal $x_c(t)$ is bandlimited, say $|f| \leq B$, then $X(e^{j\omega})$ is non-zero only for $|\omega| \leq 2\pi B/f_s$. Imposing the no-aliasing requirement Equation 46 gives

$$\frac{2\pi B}{f_s} \leq \frac{\pi}{S} \quad \Longrightarrow \quad B \leq \frac{f_s}{2S} \tag{47}$$

After downsampling, the effective sampling rate becomes $f_s' = f_s/S$, and its associated Nyquist frequency is $f_s'/2 = f_s/(2S)$. Therefore, alias-free downsampling is possible if and only if the original continuous-time signal is bandlimited to less than or equal to the new Nyquist frequency.

B.3 Boundary Conditions for Distortion-Free Tokenization

The preceding analyses establish that patch tokenization and phase tokenization are governed by distinct mathematical constraints. We now formalize the precise boundary conditions to avoid irreversible distortion.

B.3.1 Boundary Condition for Patch Tokenization

Patch tokenization, as defined in Equation 1, is mathematically equivalent to multiplying the original sequence $x[n]$ by a rectangular window of length L starting at index n_0, and then re-indexing the non-zero segment from zero. The window sequence is $x_w[n] = x[n]\, w_L[n; n_0]$, and the final patch token is $y[m] = x_w[n_0 + m]$ for $m = 0, \ldots, L-1$.

For an exact reconstruction of the original finite length sequence $x[n]$ of length N from its patch token, it is necessary and sufficient that the window covers all non-zero samples of $x[n]$. This requires $n_0 \leq 0$ and $n_0 + L - 1 \geq N - 1$. When the window starts at $n_0 = 0$, the condition reduces to $L \geq N$. Under this condition, the token $y[m] = x[m]$ for $m = 0, \ldots, N-1$ (and zero for $m \geq N$), so the original $x[n]$ can be recovered losslessly by discarding the zero-padded tail.

If $x[n]$ is periodic with a fundamental period $T \in \mathbb{N}^+$, i.e., $x[n+T] = x[n]$ for all integers n, then the entire signal is fully determined by any contiguous block of T samples. Hence, choosing the window length as an integer multiple of the period, $L = mT$ ($m \in \mathbb{N}^+$), guarantees that the token contains exactly the information needed to reconstruct the whole periodic sequence by periodic extension, irrespective of the starting index n_0.

Although time domain preservation is straightforward, the windowing operation inherently alters the frequency-domain representation. As derived in Section B.1, multiplication by the rectangular window convolves the true spectrum with a Dirichlet kernel, whose main-lobe width ($4\pi/L$) limits frequency resolution and whose sidelobes cause spectral leakage.

For periodic signals with $L = mT$, an important special case arises: the discrete Fourier transform (or the DTFT evaluated at the harmonic frequencies) samples the Dirichlet kernel exactly at its zero crossings between harmonics. This coherent sampling completely eliminates spectral leakage at those discrete frequency bins because the contribution from any harmonic to a different harmonic's bin is identically zero. However, this cancellation occurs only at the sampled frequencies; the continuous DTFT still exhibits sidelobes between the bins, and the fundamental resolution limit imposed by the main-lobe width persists. For non-periodic signals or when $L \neq mT$, the Dirichlet kernel's nulls no longer align with the analysis frequencies, and spectral leakage becomes inevitable.

The resolution limit states that two sinusoids can be separated in the magnitude spectrum only if their frequency separation exceeds half the main-lobe width, $|\omega_1 - \omega_2| \geq \frac{2\pi}{L}$. This limit is entirely independent of whether the lossless condition in the time domain is satisfied. This follows only from the finite observation length L.

In summary, the time-domain lossless condition for patch tokenization is $L \geq N$ for finite signals, or $L = mT$ for periodic signals. In the frequency domain, spectral leakage is an unavoidable consequence of the finite rectangular window, except that for coherently sampled periodic signals it is null at the DFT bins.

B.3.2 Boundary Condition for Phase Tokenization

Phase tokenization, equivalent to downsampling the original sequence by an integer factor $S \geq 2$, imposes a fundamentally different and more restrictive boundary condition. Compared to patch tokenization, the boundary condition of phase tokenization is much simpler. As derived from Equation 44, the alias-free condition for the discrete time signal is given by Equation 47. This constitutes the necessary and sufficient condition for distortion-free phase tokenization. If the original signal violates this bandlimit, high-frequency energy residing outside the interval $|\omega| \leq \pi/S$ will be folded into the baseband, causing irreversible distortion.

B.4 Analysis of TimeFT Methodology

B.4.1 Analysis of Distortion

Consider a real sequence $\mathbf{X} \in \mathbb{R}^N$. Its RFFT $\mathbf{X}_{\text{freq}} \in \mathbb{C}^{N_f}$ collects the non-negative frequency components, with the f-th bin corresponding to the angular frequency $\omega_f = 2\pi f/N$, $f = 0, \ldots, N_f - 1$. The block size is chosen as $B_N = \lfloor N/(2S) \rfloor$, satisfying the boundary condition 5.

The $k-th$ spectral block $\mathbf{X}_{\text{block}}^{(k)}$ retains only the frequency bins $f \in [kB_N,\, kB_N + B_N - 1]$ and zeros out the remaining ones. After the shifting step, i.e., reshaping into blocks of size B_N and zero-padding back to length N_f, each block becomes a spectrum $\widetilde{\mathbf{X}}^{(k)}$ concentrated in the low-frequency interval $f = 0, \ldots, B_N - 1$:

$$\left[\widetilde{\mathbf{X}}^{(k)}\right]_b = \begin{cases} [\mathbf{X}_{\text{freq}}]_{kB_N+b}, & 0 \le b \le B_N - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{48}$$

The inverse RFFT of $\widetilde{\mathbf{X}}^{(k)}$ yields a real-valued time-domain sequence $\mathbf{x}_{\text{time}}^{(k)} \in \mathbb{R}^N$. By the properties of the DFT, the spectral support of $\mathbf{x}_{\text{time}}^{(k)}$ is confined to (normalized) frequencies

$$|\omega| \le \frac{2\pi(B_N - 1)}{N} < \frac{2\pi B_N}{N} \tag{49}$$

We then obtain the strict band limit:

$$|\omega| < \frac{2\pi}{N} \cdot \frac{N}{2S} = \frac{\pi}{S} \tag{50}$$

Equation 50 is exactly the alias-free condition for downsampling by a factor S (§B.2). Hence, every sub-band signal $\mathbf{x}_{\text{time}}^{(k)}$ obeys the Nyquist criterion for the subsequent decimation step. The operator $\mathcal{D}_S$ defined in Equation 12 extracts the polyphase components of $\mathbf{x}_{\text{time}}^{(k)}$ with offsets $\lambda = 0, \ldots, S-1$. Because the signal is bandlimited to $|\omega| < \pi/S$, each decimated polyphase stream is free from spectral aliasing. Therefore, the entire information contained in the k-th frequency sub-band is almost perfectly transferred to the tokens.

The simplified spectral shifting of Equation 10 moves a complex Fourier coefficient originally located at frequency bin $kB_N + b$ directly to position b. When $b = 0$, this coefficient becomes the DC term of the shifted block. For a real signal, the DC component of the RFFT must be purely real. The IRFFT therefore discards any imaginary part at that point. Consequently, a single complex value per block loses its imaginary part, a negligible fraction of the total coefficients. As shown in the visualization in Appendix F, TimeFT tokens can be well reconstructed into the original sequence, and thus this discarding is negligible. All other frequencies ($b \neq 0$) are almost perfectly preserved. Hence, TimeFT is almost distortion-free.

B.4.2 Linear Phase Induced by Frequency Shifting

Let $x[n]$, $n = 0, \ldots, N-1$, be a real sequence and $X_r[f]$, $f = 0, \ldots, N_f - 1$, its RFFT coefficients, where $N_f = \lfloor N/2 \rfloor + 1$. We isolate the k-th sub-band of width B_N, starting at index kB_N ($B_N = \lfloor N/(2S) \rfloor$, S is the downsampling factor) $\mathcal{I}_k = \{\, kB_N,\, kB_N + 1, \ldots,\, kB_N + B_N - 1 \,\}$. The spectral shifting stage moves the content of $\mathcal{I}_k$ to the baseband $0, \ldots, B_N - 1$. A natural implementation copies the coefficients without modification:

$$\widetilde{X}_r[b] = X_r[kB_N + b], \quad b = 0, \ldots, B_N - 1 \tag{51}$$

and zeros the remaining bins. We then perform a N-point inverse RFFT to obtain a real sequence $\tilde{x}^{(k)}[n]$ of length N. We will show that this simple copy operation already produces the correct baseband signal, no extra phase rotation is required when the inverse transform retains the original length N.

Why Direct Copy Preserves the Phase Relations. The N-point DFT of the original signal satisfies $X[\ell] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi \ell n}{N}}$ and for real $x[n]$ we have $X_r[f] = X[f]$ for $0 \le f \le N/2$. Consider the contribution of the k-th sub-band to $x[n]$:

$$x_{\text{band},k}[n] = \frac{2}{N} \sum_{b=0}^{B_N-1} \text{Re}\Big(X[kB_N + b]\, e^{j\frac{2\pi(kB_N+b)n}{N}} \Big) \tag{52}$$

Now, take the DFT of the sequence $\tilde{x}^{(k)}$ obtained from the copied spectrum 51. By construction, its DFT is non-zero only at indices b and $N - b$ with coefficients $X[kB_N + b]$ and $X^*[kB_N + b]$, respectively. Therefore,

$$\begin{aligned} \tilde{x}^{(k)}[n] &= \frac{1}{N} \sum_{b=0}^{B_N-1} \Big(X[kB_N + b]\, e^{j\frac{2\pi bn}{N}} + X^*[kB_N + b]\, e^{-j\frac{2\pi bn}{N}} \Big) \\ &= \frac{2}{N} \sum_{b=0}^{B_N-1} \text{Re}\Big(X[kB_N + b]\, e^{j\frac{2\pi bn}{N}} \Big) \end{aligned} \tag{53}$$

Comparing Equation 53 with the demodulated version of 52, i.e., multiplying $x_{\text{band},k}[n]$ by $e^{-j\frac{2\pi(kB_N)}{N}n}$ and taking the real part, reveals that the two expressions are identical after canceling the complex exponential factor. Hence $\tilde{x}^{(k)}[n]$ is exactly the real baseband signal obtained by demodulating the k-th band to zero frequency and low-pass filtering. All sinusoids maintain the same relative phases as they had in the original band.

When Does a Phase Factor Become Necessary? If we were to directly obtain a critically downsampled baseband signal of length $L = N/S$ without going through the N-point grid, we would need to perform an inverse transform of length L. In that case, the time-axis compression changes the time origin: the new sample index $m = 0$ corresponds to the original time $n = 0$, but the frequency grid changes from $2\pi b/N$ to $2\pi b/L$. To correctly align the phases of the complex exponentials, each coefficient must be multiplied by $\exp(-j\frac{2\pi(kB_N)}{N}b)$, compensating for the shift of the frequency center kB_N relative to the new sampling grid. Explicitly, if we form a length-L DFT as

$$Z[b] = \frac{1}{S}\, X[kB_N + b]\, e^{-j\frac{2\pi(kB_N)}{N}b}, \quad b = 0, \ldots, B_N - 1 \tag{54}$$

and apply a length-M inverse DFT, the resulting sequence is exactly the downsampled version of $\tilde{x}^{(k)}[n]$ without any aliasing. This is the context in which the phase rotation $\exp(-j2\pi(kB_N)b/N)$ appears; it accounts for the different time origin of the shortened sequence.

TimeFT Avoids the Explicit Phase Factor. In the TimeFT pipeline we deliberately stay on the original N-point grid. After copying the coefficients, we apply an N-point inverse RFFT and then employ the dense sampling operator $\mathcal{D}_S$ to extract the tokens. Because the downsampling is performed after recovering the full-length baseband signal, the phase relations are automatically correct because the phase factor is implicitly absorbed by the subsequent decimation step. Had we shortened the inverse transform to B_N or L points, the phase compensation would have been indispensable.

B.4.3 Analysis of Multiple Tokens

A natural question that arises is whether the distortions introduced by individual tokens can be mitigated when multiple tokens are processed jointly. One might argue that since a collection of tokens together covers the entire original sequence, no information is lost in principle. However, in the case of patch tokenization, this argument does not hold when tokens are processed independently, which is the standard step in models such as PatchTST [2]. Consequently, the patterns that the model learns are inherently distorted.

Several works have proposed cross-token learning mechanisms that allow information to flow between tokens [50, 51]. These methods can partially alleviate the distortion by enabling tokens to share

contextual information, but their cross-token capacity is inherently bounded, which cannot cover all tokens simultaneously and therefore cannot fully recover the missing structure. Although these methods demonstrate the existence of the problem, they cannot fundamentally solve it.

In contrast, our TimeFT tokenization solves this problem. The token can almost losslessly reconstruct the original signal. At first glance, this may appear to introduce redundant information. However, this can be approximately viewed as a phase decomposition in the frequency domain. Extensive experiments on forecasting, classification, and anomaly detection confirm that this representation consistently improves model performance rather than introducing harmful redundancy. Furthermore, based on Appendix B.2, Phase Tokens also contain only the low-frequency information of the original sequence, resulting in similar information redundancy and even worse because of frequency aliasing. However, phase tokens have also been shown to be an effective tokenization method, such as PhaseFormer [4]. Therefore, this phenomenon does not pose a risk.

B.5 Difference between Quantization Methods and Our Main Topic

Time series quantization methods originate from attempts to process time series in a text-like fashion inspired by large language models [28]. They discretize individual time points from continuous values into a finite set of bins, leaving the input shape unchanged. This paradigm is fundamentally different from the tokenization framework that dominates current time series modeling. A large number of current TSFMs [45, 46, 49] and deep learning time series models [2, 4, 42] adopt patch or phase tokenizations, whereas quantization remains a marginal approach.

For instance, Apollo [29] views quantization as a sampling procedure and proposes to first remove high-frequency content before discretizing the continuous data points, a scheme analogous to the "bins" used in Chronos [28]. Because such methods do not involve partitioning the time series into subsequences, they fall outside the scope of our tokenization focused analysis. We therefore do not discuss them further.

Our work is the first to provide a rigorous frequency domain theoretical foundation for the distortions introduced by the prevailing tokenization paradigm, and to derive precise boundary conditions for distortion-free tokenization. Building on this novel perspective, we propose TimeFT, a general, parameter-free, and highly effective tokenizer that consistently improves performance in diverse tasks and models.

C Reconstruction Methods

C.1 The further explanation of Figure 2

In Figure 2, the further explanation of spectrum is as follows:

(a) Patch Token Spectrum: The rectangular window truncation causes the original spectrum to be convolved with a Dirichlet kernel, producing a broadened main lobe and oscillatory sidelobes. The true frequency components are indistinguishable due to leakage. The reconstruction (c) shows severe deviation from the original.

(b) Phase Token Spectrum: Downsampling by factor S creates S-1 aliased replicas that fold into the baseband. The Nyquist frequency is marked with a black vertical line; components above this threshold alias into lower frequencies, creating false peaks.

(c) TimeFT: The spectrum is partitioned into alias-free bands, shifted to baseband, and Nyquist-sampled. The reconstruction (i) is nearly identical to the original, as shown in Table 17.

C.2 Reconstruction from Patch Tokens

A patch token is obtained by multiplying the original sequence $x[n]$ of length N by a rectangular window of length L starting at n_0, and then discarding the zero entries. Reconstruction therefore amounts to padding the token with zeros to the original length and shifting it back to its original temporal position.

$$\hat{x}[n] = \begin{cases} y[n - n_0], & n_0 \le n \le n_0 + L - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{55}$$

Equivalently, in the frequency domain,

$$\hat{X}(e^{j\omega}) = Y(e^{j\omega})\, e^{-j\omega n_0} = X_w(e^{j\omega}) \tag{56}$$

where X_w is the DTFT of the windowed sequence. As shown in Appendix B.1, it inherits these distortions without any possibility of recovery, the discarded information is permanently lost.

C.3 Reconstruction from Phase Tokens

A phase token is obtained by downsampling the original sequence $x[n]$ of length N by an integer factor S, which yields $y[m] = x[mS]$ for $m = 0, \ldots, M_f - 1$, where $M_f = \lfloor N/S \rfloor$. Reconstruction consists in upsampling $y[m]$ back to the original length N. The practical method used in our experiments is Fourier-based sinc interpolation. In terms of RFFT, the algorithm computes the M_f-point RFFT of the real sequence y,

$$Y_r[k] = \sum_{m=0}^{M_f - 1} y[m]\, e^{-j2\pi km/M_f}, \qquad k = 0, \ldots, \lfloor M_f/2 \rfloor \tag{57}$$

and then zero-pads it to the length required by the original signal. The RFFT of the reconstructed sequence $\hat{x}[n]$ thus becomes

$$\hat{X}_r[k] = \begin{cases} Y_r[k], & 0 \le k \le \lfloor M_f/2 \rfloor, \\ 0, & \lfloor M_f/2 \rfloor < k \le \lfloor N_f/2 \rfloor. \end{cases} \tag{58}$$

The time domain reconstruction is finally obtained by IRFFT, $\hat{x} = \mathcal{F}_{\mathbb{R}}^{-1}(\hat{X}_r)$. We denote the upsampling operator here as $\mathcal{R}(\cdot)$. Since $\lfloor M_f/2 \rfloor = \lfloor N/(2S) \rfloor$, zero-padding in Equation 58 is equivalent to applying an ideal low-pass filter with cutoff frequency π/S to the upsampled sequence. The quality of reconstruction critically depends on whether the original signal is bandlimited to $|\omega| \le \pi/S$. If $x[n]$ contains energy at frequencies $|\omega| > \pi/S$, the downsampling step produces spectral aliasing that maps high-frequency components to the low-frequency region (§B.2). The upsampling process cannot distinguish between a true low-frequency component and an aliased high-frequency one, so the reconstructed signal will still contain wrong patterns.

C.4 Reconstruction from TimeFT Tokens

The TimeFT tokenizer produces S tokens per variate. Each token is a vector of length $K \cdot L$, where $K = \lceil N_f/B \rceil$ is the number of frequency blocks, $B = \lfloor N/(2S) \rfloor$ is the bandwidth, and $L = \lceil N/S \rceil$ is an intermediate variable. Reconstruction proceeds by inverting the two stages of the tokenizer: (i) polyphase recomposition and (ii) inverse spectral shifting.

Polyphase Recomposition. For a fixed variate and frequency block index k, the S tokens with offsets $\lambda = 0, \ldots, S - 1$ represent the S polyphase components of the bandlimited time domain sequence $\mathbf{x}_{\text{time}}^{(k)}$ obtained after spectral shifting and IRFFT 11. The reconstruction first upsamples every sequence back to length N using the same Fourier-based sinc interpolation described in the previous subsection:

$$\hat{\mathbf{x}}_{\text{time}}^{(k)} = \mathcal{R}(\mathbf{x}_{\text{time}}^{(k)}) \tag{59}$$

Equivalently, this can be expressed as the output of a S-channel synthesis filter bank with ideal interpolation filters.

Inverse Spectral Shifting. The reconstructed bandlimited signal $\hat{\mathbf{x}}^{(k)}_{\text{time}}$ occupies the low-frequency region of the k-th spectral block. To recover the contribution of this block to the full spectrum, we apply N-point RFFT, obtaining $\hat{\mathbf{X}}^{(k)}_{\text{block}} \in \mathbb{C}^{N_f}$. Only the first B frequency bins are retained and the remainder are set to zero.

$$\big[\hat{\mathbf{X}}^{(k)}_{\text{block}}\big]_f = \begin{cases} [\text{RFFT}(\hat{\mathbf{x}}^{(k)}_{\text{time}})]_f, & 0 \le f < B, \\ 0, & \text{otherwise.} \end{cases} \tag{60}$$

The original spectrum was partitioned into contiguous blocks of size B 7). The inverse operation simply places the retained frequency bins of the k-th block back into their original positions:

$$\hat{\mathbf{X}}_{\text{freq}}[kB + b] = \big[\hat{\mathbf{X}}^{(k)}_{\text{block}}\big]_b, \qquad k = 0, \dots, K-1, \;\; b = 0, \dots, B-1 \tag{61}$$

All bins of $\hat{\mathbf{X}}_{\text{freq}}$ that lie beyond the last block (if any) are obtained from the zero-padding that was applied during the forward decomposition. Finally, an IRFFT produces the reconstructed time domain sequence:

$$\hat{\mathbf{x}} = \mathcal{F}^{-1}_{\mathbb{R}}\big(\hat{\mathbf{X}}_{\text{freq}}\big) \in \mathbb{R}^N \tag{62}$$

Fidelity of the Reconstruction. Spectral shifting does not distort the raw time series. For the frequency, spectral shifting is merely a change of basis. Moving a frequency band from to can be viewed as a signal modulation rather than a distortion. Moreover, since our method satisfies the boundary condition in Eq. 4, no aliasing issues arise. For the magnitude, the phase factor has a modulus of 1, which preserves all amplitude.

The only deviation from perfect reconstruction comes from the simplified spectral shifting of Equation 10, causing the DC coefficient of each block to lose its imaginary part. Apart from this single complex value per block, the entire process is algebraically equivalent to passing the signal through a bank of ideal bandpass filters followed by perfect reconstruction synthesis. Because the block bandwidth B is chosen to respect the alias-free condition $B \le N/(2S)$, no aliasing occurs during downsampling, and polyphase decomposition captures all samples without loss. Consequently, the reconstructed signal $\hat{\mathbf{x}}$ differs from the original $\mathbf{x}$ only in the imaginary parts of the zero-frequency bins of each spectral block. This distortion is negligible in magnitude and that leaves the amplitude spectrum essentially identical to that of the original. This explains why TimeFT tokens are almost distortion-free, instead of information loss of patch tokens and aliasing artifacts from phase tokens.

D Full Experiment Results

In this section, we present detailed experiments of TimeFT, including three major tasks of long-term forecasting D.1, classification D.2 and anomaly detection D.3. Moreover, we provide settings for other experiments D.8, variation in model parameters D.9 and reconstruction performance of different tokenizations D.10. The computational efficiency (§4.3), the plug-in experiment (§4.4), the ablation study (§4.5) and the hyperparameter sensitivity (§4.7) please see Section 4. All experiments were performed using Intel Core i7-13700H CPU and NVIDIA GeForce RTX 4060 GPU. The training and testing procedures in the experiments follow the Time Series Library [52].

In the experiments, we select datasets following the standard protocol established by prior works. **Long-term forecasting** is performed on seven real-world datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity and Traffic [11–13]. **Classification** uses all UEA datasets [14–16]. **Anomaly detection** is conducted on five widely used benchmarks [14, 17, 18]: SMD, MSL, SMAP, SWaT, and PSM.

D.1 Long-Term Forecasting Task

We evaluate the long-term forecasting performance of TimeFT against state-of-the-art tokenization baselines across diverse benchmarks. The experiments adopt the same settings as TimeMosaic [19]. We evaluate our model on widely used real-world time series forecasting datasets, including Electricity

Transformer Temperature (ETT), Weather, Electricity, and Traffic. The details of the datasets for this task are as follows:

- **ETT** [1] comprises four subsets: ETTh1 and ETTh2 (hourly), and ETTm1 and ETTm2 (every 15 minutes), collected from two electricity transformers.
- **Weather** [2] contains 21 meteorological variables recorded every 10 minutes in 2020 at the Max Planck Institute for Biogeochemistry.
- **Electricity** [3] provides hourly power consumption data of 321 clients from 2012 to 2014.
- **Traffic**[4] consists of hourly road occupancy rates from 862 freeway sensors in the San Francisco Bay Area between January 2015 and December 2016.

Table 7 summarizes the key characteristics of the datasets used in our evaluation. The collection encompasses various sampling rates, ranging from 10-minute and 15-minute intervals to hourly and daily recordings. And they exhibits considerable diversity in dimensionality, spanning from the 7-dimensional ETT subsets to the 321-dimensional ECL dataset. For each dataset, we report the partition sizes in the format (Training, Validation, Test) to clearly specify the data splitting strategy adopted across all experiments.

Table 7: Summary of datasets for long-term forecasting task. Dim denotes dimension, which is the variate number of each dataset. Dataset sizes represent the number of temporal observations in each partition (Training, Validation, Test). Frequency denotes the sampling interval of time points.

Dataset	Dim	Dataset Size (Train, Val, Test)	Frequency	Domain Information
ETTm1	7	(34465, 11521, 11521)	15min	Energy Infrastructure
ETTm2	7	(34465, 11521, 11521)	15min	Energy Infrastructure
ETTh1	7	(8545, 2881, 2881)	Hourly	Energy Infrastructure
ETTh2	7	(8545, 2881, 2881)	Hourly	Energy Infrastructure
Weather	21	(36792, 5271, 10540)	10min	Meteorological Monitoring
Electricity	321	(18317, 2633, 5261)	Hourly	Electricity Consumption
Traffic	862	(12185,1757,3509)	Hourly	Traffic Flow

As demonstrated in Table 8, TimeFT achieves the best performance in the majority of datasets and metrics. By outperforming both classic and recent baselines, TimeFT establishes an excellent paradigm for obtaining tokens of time series.

D.2 Classification

For the classification task, we employ publicly available multivariate time series datasets from the University of East Anglia (UEA [5]) Time Series Classification repository, which are widely recognized benchmarks in the field. These datasets capture a diverse range of real-world scenarios,including human activity recognition, sensor data interpretation, and complex system monitoring, and thus provide a multidimensional foundation for evaluating time series models.

We evaluate the classification performance of TimeFT against state-of-the-art tokenization methods on all UEA datasets. We run and evaluate several recent tokenization methods for comparison. As shown in Table 9, TimeFT achieves the highest average accuracy and achieves the best or second-best result in a majority of datasets. This strong performance improvement is due to TimeFT's almost distortion-free tokenization ability, which faithfully preserves discriminative temporal patterns that are often corrupted by tokenizers.

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

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

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

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

[5] https://www.timeseriesclassification.com/

Table 8: For long-term forecasting, we evaluate prediction horizons $H \in \{96, 192, 336, 720\}$. All baselines use a fixed input sequence length of 320. The term "Avg." refers to the average metric over the four horizons. The best and second-best results are marked as **best** and second, respectively. The column "1st Count" reports the number of times each method ranks first among all competitors.

Models		TimeFT (**Ours**)		EntroPE (2026)		TimeMosaic (2026)		PhaseFormer (2026)		TimeFilter (2025)		xPatch (2025)		PatchMLP (2025)		LightTS (2024)		iTransformer (2024)		TimeMixer (2024)		PatchTST (2023)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.290**	**0.339**	0.316	0.349	0.304	0.344	0.312	0.352	0.355	0.373	0.293	0.347	0.305	0.357	0.313	0.363	0.327	0.371	0.307	0.352	0.321	0.361	0.299	0.344
	192	**0.325**	**0.357**	0.344	0.371	0.339	0.364	0.339	0.381	0.376	0.396	0.332	0.371	0.351	0.384	0.350	0.387	0.357	0.388	0.332	0.373	0.342	0.385	0.337	0.370
	336	**0.360**	**0.381**	0.386	0.394	0.375	0.385	0.375	0.398	0.401	0.413	0.364	0.395	0.387	0.403	0.399	0.418	0.393	0.412	0.362	0.390	0.386	0.410	0.369	0.385
	720	**0.418**	0.415	0.431	0.419	0.422	**0.413**	0.428	0.437	0.465	0.449	0.428	0.432	0.454	0.441	0.489	0.479	0.467	0.452	0.422	0.426	0.439	0.441	0.424	0.419
	Avg.	**0.348**	**0.373**	0.369	0.383	0.360	0.377	0.364	0.392	0.399	0.408	0.354	0.386	0.374	0.396	0.388	0.412	0.386	0.406	0.356	0.385	0.372	0.399	0.357	0.380
ETTm2	96	**0.162**	**0.244**	0.173	0.256	0.164	0.247	0.174	0.261	0.174	0.261	0.168	0.259	0.179	0.265	0.187	0.286	0.187	0.271	0.169	0.258	0.181	0.270	0.172	0.270
	192	**0.223**	**0.287**	0.231	0.301	0.226	0.291	0.231	0.306	0.229	0.298	0.226	0.299	0.232	0.301	0.311	0.384	0.251	0.314	0.240	0.308	0.241	0.311	0.239	0.322
	336	**0.275**	**0.321**	0.288	0.334	0.279	0.324	0.291	0.342	0.312	0.351	0.281	0.337	0.285	0.334	0.399	0.418	0.319	0.355	0.282	0.334	0.303	0.348	0.305	0.367
	720	0.361	**0.375**	0.364	0.388	**0.353**	0.376	0.374	0.388	0.396	0.408	0.372	0.391	0.380	0.394	0.489	0.479	0.413	0.412	0.369	0.391	0.385	0.399	0.412	0.432
	Avg.	**0.255**	**0.307**	0.264	0.320	0.256	0.310	0.268	0.324	0.278	0.330	0.262	0.322	0.269	0.324	0.347	0.392	0.293	0.338	0.265	0.323	0.278	0.332	0.282	0.348
ETTh1	96	0.376	0.392	0.378	0.396	**0.362**	**0.387**	0.389	0.411	0.460	0.457	0.380	0.403	0.411	0.426	0.432	0.443	0.403	0.417	0.376	0.398	0.403	0.423	0.377	0.400
	192	0.417	0.416	0.409	0.418	**0.400**	**0.411**	0.461	0.472	0.439	0.443	0.425	0.430	0.443	0.445	0.474	0.472	0.443	0.444	0.412	0.420	0.479	0.478	0.410	0.420
	336	**0.434**	**0.429**	0.447	0.446	0.436	0.435	0.485	0.476	0.474	0.463	0.444	0.439	0.468	0.461	0.515	0.501	0.470	0.464	0.445	0.446	0.490	0.481	0.452	0.458
	720	**0.446**	**0.458**	0.462	0.474	0.451	0.460	0.619	0.542	0.670	0.564	0.534	0.509	0.525	0.508	0.592	0.567	0.632	0.573	0.488	0.489	0.641	0.559	0.477	0.497
	Avg.	0.417	**0.423**	0.424	0.433	**0.412**	**0.423**	0.489	0.475	0.511	0.482	0.446	0.445	0.462	0.460	0.503	0.496	0.487	0.475	0.430	0.438	0.503	0.485	0.429	0.444
ETTh2	96	0.297	**0.339**	0.308	0.354	0.300	0.342	0.309	0.361	0.367	0.389	**0.280**	0.343	0.311	0.371	0.324	0.383	0.323	0.369	0.285	0.346	0.324	0.374	0.305	0.368
	192	0.349	**0.383**	0.366	0.392	0.354	0.387	0.364	0.403	0.372	0.404	**0.345**	0.386	0.381	0.413	0.438	0.455	0.421	0.422	0.358	0.391	0.378	0.416	0.407	0.435
	336	**0.375**	**0.401**	0.387	0.411	0.378	0.403	0.392	0.425	0.402	0.426	0.382	0.415	0.401	0.427	0.496	0.490	0.440	0.441	0.380	0.409	0.405	0.432	0.493	0.490
	720	0.410	**0.437**	0.434	0.447	0.421	0.439	0.469	0.477	0.465	0.464	**0.408**	0.443	0.426	0.451	1.151	0.749	0.459	0.466	0.410	0.439	0.484	0.484	0.762	0.620
	Avg.	0.358	**0.390**	0.374	0.401	0.363	0.393	0.384	0.417	0.402	0.421	**0.354**	0.397	0.380	0.416	0.602	0.520	0.411	0.425	0.358	0.396	0.398	0.427	0.492	0.478
Weather	96	0.147	0.194	0.153	0.197	**0.144**	**0.189**	0.152	0.201	0.156	0.209	0.148	0.196	0.153	0.206	0.152	0.212	0.160	0.209	0.150	0.199	0.155	0.204	0.177	0.242
	192	0.199	0.236	0.201	0.239	0.197	**0.232**	0.201	0.249	0.195	0.246	**0.190**	0.238	0.197	0.245	0.197	0.255	0.207	0.251	0.193	0.244	0.201	0.249	0.217	0.276
	336	0.248	0.279	0.254	0.283	0.245	**0.276**	0.249	0.282	0.251	0.286	**0.241**	0.280	0.247	0.282	0.249	0.299	0.256	0.289	0.244	0.283	0.255	0.289	0.262	0.312
	720	0.332	0.335	0.339	0.336	0.327	**0.327**	0.331	0.340	0.322	0.339	**0.318**	0.333	0.324	0.334	0.323	0.357	0.325	0.338	0.319	0.336	0.333	0.340	0.326	0.367
	Avg.	0.232	0.261	0.237	0.264	0.228	**0.256**	0.233	0.268	0.231	0.270	**0.224**	0.262	0.230	0.267	0.230	0.281	0.237	0.272	0.227	0.266	0.236	0.271	0.246	0.299
Electricity	96	0.139	0.234	0.148	0.241	0.140	0.236	0.145	0.248	**0.137**	**0.233**	0.143	0.241	0.165	0.271	0.171	0.285	0.145	0.244	0.141	0.238	0.142	0.245	0.143	0.242
	192	**0.151**	**0.245**	0.160	0.258	0.156	0.250	0.160	0.261	0.153	0.246	0.157	0.252	0.183	0.288	0.192	0.305	0.165	0.263	0.158	0.254	0.159	0.259	0.157	0.255
	336	**0.166**	**0.259**	0.181	0.275	0.173	0.267	0.178	0.275	0.169	0.263	0.172	0.266	0.201	0.304	0.214	0.324	0.184	0.282	0.175	0.270	0.176	0.276	0.172	0.272
	720	**0.206**	**0.293**	0.219	0.309	0.212	0.298	0.224	0.309	0.208	0.296	0.209	0.297	0.250	0.343	0.251	0.349	0.227	0.317	0.217	0.307	0.215	0.308	0.207	0.305
	Avg.	**0.165**	**0.258**	0.177	0.271	0.170	0.263	0.177	0.273	0.167	0.260	0.170	0.264	0.200	0.302	0.207	0.316	0.180	0.277	0.173	0.267	0.173	0.272	0.170	0.269
Traffic	96	**0.395**	0.279	0.421	0.304	0.413	0.297	0.418	0.289	0.397	**0.274**	0.413	0.297	0.486	0.361	0.493	0.367	0.422	0.310	0.421	0.297	0.403	0.285	0.422	0.296
	192	0.412	0.281	0.433	0.288	0.424	0.281	0.426	0.301	**0.411**	**0.278**	0.425	0.298	0.515	0.378	0.511	0.373	0.449	0.327	0.441	0.311	0.419	0.292	0.435	0.302
	336	**0.421**	0.291	0.446	0.295	0.435	0.287	0.443	0.304	0.423	**0.284**	0.430	0.297	0.533	0.386	0.521	0.378	0.469	0.339	0.425	0.317	0.432	0.298	0.447	0.308
	720	0.452	0.309	0.477	0.314	0.464	0.302	0.478	0.324	**0.449**	**0.300**	0.450	0.306	0.564	0.401	0.504	0.360	0.512	0.364	0.489	0.340	0.459	0.313	0.476	0.327
	Avg.	**0.420**	0.290	0.444	0.300	0.434	0.287	0.441	0.305	**0.420**	**0.284**	0.430	0.300	0.525	0.382	0.507	0.370	0.463	0.335	0.444	0.316	0.428	0.297	0.445	0.308
1st Count		**40**		0		14		0		10		8		0		0		0		0		0		0	

D.3 Anomaly Detection

For anomaly detection, we adopt five widely recognized datasets that span diverse application domains, allowing a thorough evaluation of detection capabilities under varied conditions. The detailed characteristics of these datasets are presented in Table 10.

- **SMD (Server Machine Dataset)**: Comprising multivariate server telemetry, SMD targets the identification of anomalous server behavior indicative of failures or security breaches.
- **MSL (Mars Science Laboratory)**: Originating from the Mars rover, this telemetry dataset focuses on space exploration, where detecting anomalies is critical for diagnosing spacecraft subsystem faults.
- **SMAP (Soil Moisture Active Passive)**: Containing satellite soil moisture measurements, SMAP supports environmental monitoring; detected irregularities can reveal critical changes in environmental and agricultural conditions.

Table 9: We evaluate classification tasks on all UEA datasets. Best results are in **bold** and second best are underlined.

Datasets	**TimeFT (Ours)**	EntroPE (2026)	TimeMosaic (2026)	PhaseFormer (2026)	PatchMLP (2025)	GPT4TS (2024)	PatchTST (2023)	TimesNet (2023)	MLP -
ArticularyWordRecognition	97.50	97.00	96.50	95.00	97.20	93.33	**97.67**	96.18	97.33
AtrialFibrillation	53.00	52.00	50.50	48.00	53.00	33.33	**53.33**	33.33	46.66
BasicMotions	98.00	98.00	97.00	95.50	98.00	92.50	92.50	**100.00**	85.00
Cricket	91.00	90.00	89.00	87.00	90.50	8.33	84.72	87.50	**91.67**
Epilepsy	85.00	84.00	82.50	80.00	84.50	**85.51**	65.94	78.13	60.14
EthanolConcentration	**31.94**	29.27	28.90	26.13	29.65	25.48	28.90	27.73	28.09
FaceDetection	**69.48**	69.03	67.95	64.65	67.14	65.58	68.96	67.47	67.42
FingerMovements	63.00	62.50	61.00	59.50	63.00	57.00	62.00	59.38	**64.00**
HandMovementDirection	58.00	56.00	54.50	52.00	57.00	18.92	**58.11**	50.00	**58.11**
Handwriting	24.70	23.31	24.80	23.16	23.87	3.76	26.00	**26.18**	22.47
Heartbeat	**77.06**	76.64	73.34	76.10	73.54	36.59	76.59	74.48	73.17
InsectWingbeat	**11.00**	10.50	10.00	9.50	10.20	10.00	10.00	10.00	10.00
JapaneseVowels	**98.87**	98.76	97.23	95.57	97.49	98.11	98.65	97.83	97.84
Libras	**82.00**	80.00	78.50	76.00	80.50	79.44	81.11	77.84	73.33
LSST	**68.50**	66.00	64.00	61.50	66.50	46.39	67.80	59.21	35.77
MotorImagery	**62.00**	60.00	58.50	56.00	60.50	50.00	61.00	51.04	61.00
NATOPS	**97.50**	95.50	94.00	92.00	96.00	91.67	96.67	81.82	93.89
PEMS-SF	**88.86**	87.12	87.15	86.80	78.85	87.28	88.44	88.13	82.08
PenDigits	**99.80**	98.50	98.00	97.00	99.00	97.74	99.23	98.19	92.94
PhonemeSpectra	**19.00**	17.00	15.50	14.00	17.50	3.01	11.69	18.24	7.10
RacketSports	**85.00**	83.00	81.50	79.50	83.50	76.97	84.21	82.64	78.95
SelfRegulationSCP1	89.05	88.58	89.35	86.03	89.36	**91.47**	89.76	77.43	88.40
SelfRegulationSCP2	**59.44**	57.42	52.07	56.46	53.62	51.67	54.44	52.84	51.67
SpokenArabicDigits	**99.71**	98.75	98.73	97.60	96.99	99.36	99.68	98.36	96.68
StandWalkJump	**61.00**	58.50	57.00	55.00	59.00	33.33	60.00	53.33	60.00
UWaveGestureLibrary	83.75	**84.38**	81.88	76.81	78.63	**84.38**	80.00	83.13	81.88
Average	**71.31**	70.07	68.82	67.19	69.42	58.51	69.13	66.55	65.60
1st	**16**	1	0	0	0	3	3	2	3

Table 10: Summary of datasets used for anomaly detection evaluation. Dim denotes the variate number of each dataset. Dataset sizes represent the number of temporal observations in each partition (Training, Validation, Test). Frequency denotes the sampling interval of time points.

Dataset	Dim	Dataset Size (Train, Val, Test)	Length	Domain Information
SMD	38	(566724, 141681, 708420)	100	Server Machine
MSL	55	(44653, 11664, 73729)	100	Spacecraft
SMAP	25	(108146, 27037, 427617)	100	Spacecraft
SWaT	51	(396000, 99000, 449919)	100	Infrastructure
PSM	25	(105984, 26497, 87841)	100	Server Machine

Table 11: For anomaly detection task, we evaluate F1-score (%) on five benchmark datasets. Best results are in **bold** and second best are underlined.

Models	**TimeFT (Ours)**	EntroPE (2026)	TimeMosaic (2026)	Phaseformer (2026)	LightTS (2024)	TimesNet (2023)	PatchTST (2023)	DLinear (2023)	ETSformer (2023)	FEDformer (2022)
SMD	**85.52**	84.75	84.25	84.85	82.53	84.61	84.62	77.10	83.13	85.08
MSL	82.95	81.95	78.42	79.15	78.95	81.84	78.70	84.88	**85.03**	78.57
SMAP	**71.80**	69.80	68.10	68.75	69.21	69.39	68.82	69.26	69.50	70.76
SWaT	**93.65**	93.10	85.90	86.12	93.33	93.02	85.72	87.52	84.91	93.19
PSM	96.68	97.28	95.85	96.20	97.15	**97.34**	96.08	93.55	91.76	97.23
Average	**86.12**	85.38	82.50	83.01	84.23	85.24	82.79	82.46	82.87	84.97

- **SWaT (Secure Water Treatment)**: Derived from a water treatment testbed, SWaT simulates plant operations and is essential for ensuring the safety and integrity of water treatment processes through anomaly detection.
- **PSM (Pump Sensor Monitoring)**: Collected from industrial pump sensors, PSM enables anomaly detection to flag equipment malfunctions and maintenance needs, thereby helping to prevent industrial incidents.

We evaluate the anomaly detection capability of TimeFT against state-of-the-art tokenization baselines and classic time series models on five standard benchmarks. We run and evaluate several recent tokenization methods for comparison. As shown in Table 11, TimeFT achieves the highest average F1-score of 86.12%. TimeFT preserves the integrity of anomalous patterns and shows excellent detection performance. Experiments on diverse tasks and extensive benchmarks demonstrate that TimeFT exhibits superior performance compared to existing tokenization methods.

D.4 Robustness Analysis under Noisy Conditions

Bandwidth Estimation. Since deep learning models typically require fixed-dimensional input, time-series models often employ an input window to perform tasks such as forecasting, specifically a hyperparameter of length N, which provides a fixed number of data points for DSP analysis. Furthermore, the stride S is usually a tokenization hyperparameter used to control the number of tokens. Therefore, the critical bandwidth B_N is independent of noise. It is entirely derived from the distortion conditions of the existing tokenization paradigm.

Experiments under Noisy Conditions. The results of robustness analysis under noisy conditions are as shown in Table 12. TimeFT strictly enforces the alias-free boundary condition. The frequency-domain blocking operation applies ideal bandpass filters, which mathematically zero out all out-of-band noise components before any downsampling. Each sub-band is then independently shifted to baseband and satisfies the Nyquist criterion. Consequently, the subsequent sampling step will not incur aliasing. High-frequency noise is strictly confined to the corresponding high frequency. This gives TimeFT excellent noise robustness.

Table 12: Long-term forecasting results on ETTm1 under Gaussian noise. σ is the standard deviation of the noise.

Noise (σ)	0		0.2		0.4		0.6		0.8		1.0	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST	0.321	0.361	0.343	0.377	0.368	0.394	0.397	0.419	0.440	0.447	0.483	0.485
PhaseFormer	0.312	0.352	0.330	0.364	0.350	0.381	0.382	0.395	0.414	0.426	0.458	0.449
TimeFT (ours)	**0.290**	**0.339**	**0.302**	**0.347**	**0.318**	**0.355**	**0.335**	**0.374**	**0.364**	**0.387**	**0.391**	**0.412**

Table 13: Degradation Rate of the long-term forecasting task under Gaussian noise. Metric: MSE.

Noise (σ)	0	0.2	0.4	0.6	0.8	1.0
PatchTST	0	6.85%	14.64%	23.68%	37.07%	50.47%
PhaseFormer	0	5.77%	12.18%	22.44%	32.69%	46.79%
TimeFT (ours)	**0**	**4.14%**	**9.66%**	**15.52%**	**25.52%**	**34.83%**

Qualitative Analysis. It can be observed that there is a positive correlation between the level of distortion and the degradation in downstreaming tasks performance, which validates the argument presented in our paper. For TimeFT, the frequency-domain blocking operation applies ideal bandpass filters, which mathematically zero out all out-of-band noise components before any downsampling. Each sub-band is then independently shifted to baseband and satisfies the Nyquist criterion.

Consequently, the subsequent sampling step incurs zero aliasing. High-frequency noise is strictly confined to the corresponding high frequency, while the low frequency remains completely immune to contamination. This band-decoupled robustness is absent in both patch and phase tokenization.

Quantitative Analysis. To measure the sensitivity of each tokenization method to spectral distortion, we define the Degradation Rate $D(\sigma)$ as the relative increase in MSE from the noise-free baseline $\sigma = 0$.

$$D(\sigma) = (MSE(\sigma) - MSE(0))/MSE(0) \times 100\%$$

As shown in Table 13, the degradation rates reveal a clear hierarchy. TimeFT exhibits the lowest performance deterioration across all noise levels, with a substantially smaller average slope compared to PatchTST and PhaseFormer. This quantitatively demonstrates that TimeFT's spectral-distortion-free design confers greater robustness to input corruption. Moreover, the performance gap widens slowly with increasing noise, suggesting that TimeFT exhibits superior anti-distortion performance compared to prior tokenization methods.

D.5 Experiments on More Architectures

As shown in Table 14, the experimental results of more diverse and state-of-the-art architectures demonstrate that TimeFT's near-lossless tokenization is readily transferable to different architectures, delivering immediate gains without modifying the base model design.

Table 14: Plug-in experiments of the long-term forecasting task to show the general capability. All settings are the same as Table 5.

Dataset	ETTm1		ETTm2		ETTh1		ETTh2		Weather		Electricity		Traffic	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST	0.321	0.361	0.181	0.270	0.403	0.423	0.324	0.374	0.155	0.204	0.142	0.245	0.403	0.285
+TimeFT	**0.290**	**0.339**	**0.162**	**0.244**	**0.376**	**0.392**	**0.297**	**0.339**	**0.147**	**0.194**	**0.139**	**0.234**	**0.395**	**0.279**
Δ% Improvement	9.7%	6.1%	10.5%	9.6%	6.7%	7.3%	8.3%	9.4%	5.2%	4.9%	2.1%	4.5%	2.0%	2.1%
PhaseFormer	0.312	0.352	0.174	0.261	0.389	0.411	0.309	0.361	0.152	0.201	0.145	0.248	0.418	0.289
+TimeFT	**0.293**	**0.341**	**0.161**	**0.249**	**0.379**	**0.397**	**0.299**	**0.343**	**0.149**	**0.193**	**0.143**	**0.242**	**0.406**	**0.278**
Δ% Improvement	6.1%	3.1%	7.5%	4.6%	2.6%	3.4%	3.2%	5.0%	1.9%	3.9%	1.4%	2.4%	2.9%	3.8%
PatchMLP	0.305	0.357	0.179	0.265	0.411	0.426	0.311	0.371	0.153	0.206	0.165	0.271	0.486	0.361
+TimeFT	**0.286**	**0.339**	**0.165**	**0.250**	**0.390**	**0.405**	**0.294**	**0.348**	**0.147**	**0.197**	**0.160**	**0.264**	**0.472**	**0.349**
Δ% Improvement	6.2%	5.0%	7.8%	5.7%	5.1%	4.9%	5.5%	6.2%	3.9%	4.4%	3.0%	2.6%	2.9%	3.3%
iTransformer	0.327	0.371	0.187	0.271	0.403	0.417	0.323	0.369	0.160	0.209	0.145	0.244	0.422	0.310
+TimeFT	**0.309**	**0.355**	**0.174**	**0.257**	**0.390**	**0.403**	**0.312**	**0.354**	**0.156**	**0.202**	**0.142**	**0.239**	**0.412**	**0.302**
Δ% Improvement	5.5%	4.3%	7.0%	5.2%	3.2%	3.4%	3.4%	4.1%	2.5%	3.3%	2.1%	2.0%	2.4%	2.6%

D.6 Experiments on Different Random Seeds

As shown in Table 15, it can be observed that the robustness of different methods under multiple random seeds is basically the same.

Table 15: Long-term forecasting results under different random seeds to show the effect of random seeds.

Models	TimeFT (ours)		EntroPE		TimeMosaic		PhaseFormer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.348±0.005**	**0.373±0.004**	0.369±0.007	0.383±0.006	0.360±0.004	0.377±0.008	0.364±0.006	0.392±0.005
ETTm2	0.255±0.003	**0.307±0.004**	0.264±0.006	0.320±0.005	**0.256±0.003**	0.310±0.007	0.268±0.009	0.324±0.006
ETTh1	0.417±0.006	**0.423±0.005**	0.424±0.008	0.433±0.004	**0.412±0.005**	**0.423±0.007**	0.489±0.009	0.475±0.004
ETTh2	**0.358±0.004**	**0.390±0.007**	0.374±0.008	0.401±0.005	0.363±0.003	0.393±0.006	0.384±0.006	0.417±0.005
Weather	0.232±0.003	0.261±0.005	0.237±0.005	0.264±0.007	**0.228±0.004**	**0.256±0.008**	0.233±0.005	0.268±0.009
Electricity	**0.165±0.004**	**0.258±0.006**	0.177±0.007	0.271±0.005	0.170±0.003	0.263±0.008	0.177±0.007	0.273±0.008
Traffic	**0.420±0.008**	0.290±0.006	0.444±0.009	0.300±0.005	0.434±0.006	**0.287±0.004**	0.441±0.003	0.305±0.006

D.7 Non-stationary Data Analysis

TimeFT is orthogonal to non-stationary property, just as RevIN normalization is orthogonal to PatchTST.

RFFT still holds for non-stationary data. As one of the foundations of TimeFT, RFFT and the Fourier transform family are indeed affected by abrupt shifts. This is due to the fact that the Fourier transform operates as a convolution across the entire time domain on a time series of length N. But as basis transformations, the FFT family does not need the stationarity assumption. For non-stationary signals or abrupt shifts, the Fourier transform will just simply produce a broadband spectrum containing high-frequency components associated with the sharp transitions.

However, our TimeFT addresses the distortion problem in tokenization, which is an issue independent of the abrupt drifts in the raw data. This orthogonality arises from the fact that the characteristics of the raw data and the tokenization process are two distinct types of problems. This is similar to how PatchTST's patch tokenization also cannot resolve abrupt shifts in the original data. For data types that suffer from abrupt shift issues, researchers typically adopt additional methods, such as RevIN normalization [53], to suppress such abrupt changes.

Therefore, TimeFT is not designed exclusively for periodic or stationary data. Rather, it addresses the distortion problem inherent in the general tokenization paradigm, which is orthogonal to the non-stationarity of the raw data. If handling highly non-stationary data types is required, we can introduce dedicated modules, such as RevIN normalization, as many prior works have done.

D.8 Settings for Other Experiments

Time Complexity Analysis. In the time complexity analysis experiment, we ensure that all experimental parameters are identical. The test is conducted on the ETTm1 dataset, with an input length of 320 and a prediction length of 336.

Plu-in Experiments. In the plug-in experiments, we replace the tokenization modules of different methods with TimeFT, leaving the other components unchanged. For each method, we carefully tune the hyperparameters both before and after the substitution to ensure strong performance.

Ablation Study. In the ablation study, removing a Stage means removing that module from the pipeline while keeping the rest unchanged. The evaluation is conducted on the long-term forecasting task, with an input length of 320 and a prediction length of 96.

Hyperparameter Sensitive Analysis. Generally speaking, the only hyperparameter of TimeFT is the stride S, and all other parameters are mathematically derived. Moreover, as a tokenizer, TimeFT is closely related to input length N. Therefore, in the hyperparameter sensitivity experiment, we compare the effects of different strides S and input lengths N on model performance. For each setting, we tune other hyperparameters because the optimal configuration changes accordingly.

D.9 Variation in Model Parameters

When analyzing the impact of TimeFT on the model parameters, we observed that hyperparameters significantly affect this metric. Therefore, we keep all hyperparameters identical across experiments and compare only the variation in the embedding module parameters and the total parameter size, since the tokenizer mainly influences the number of embedding parameters. "Total of Model" denotes the total parameter count of the model under identical hyperparameter settings. As shown in Table 16, although TimeFT increases the number of embedding parameters due to the change in input dimensionality, this change is negligible compared to the total number of parameters.

Table 16: Variation in Model Parameter Count.

Layers	TimeFT Embedding Layer	PatchTST Embedding Layer	Total of Model
Params (MB)	0.699	0.031	50.337

D.10 Reconstruction Performance of Different Token Types

To intuitively illustrate the degree of distortion for different types of tokens, we compare the mean absolute error between the reconstructed sequences and the original sequences across different datasets. This experiment can be viewed as a statistical summary of Figure 2 and the visualization

section F. We ensure that all hyperparameters remain identical in the experiments. As shown in Table 17, TimeFT tokens are almost distortion-free and reduce more than 80% reconstruction error in many datasets. The reconstruction methods are provided in Section C.

Table 17: Reconstruction performance of different token types across different datasets. The percentage represents the MAE reduction of TimeFT tokens relative to the best one of others. Metric: MAE.

Datasets	ETTm1	ETTm2	ETTh1	ETTh2	Weather	Electricity	Traffic
Patch	0.768	0.747	0.755	0.722	0.726	0.821	0.719
Phase	0.449	0.525	0.823	0.714	0.280	0.998	0.986
TimeFT	**0.102**	**0.115**	**0.144**	**0.133**	**0.079**	**0.122**	**0.138**
Reduction	77.3%	78.1%	80.9%	81.4%	71.8%	85.1%	80.8%

E Limitations and Social Impacts

Limitations. Our work carries two limitations. Owing to the substantial computational expense involved in pre-training large-scale time series foundation models, our experimental evaluation does not encompass pre-trained TSFMs. Consequently, the behavior and potential benefits of TimeFT in a pre-training and fine-tuning pipeline remain unverified, and investigating its effectiveness under TSFMs regimes is left to future work. Furthermore, although tokenization has become a prevailing paradigm in modern time series analysis, our method intrinsically assumes a tokenization-based processing flow and therefore yields no direct improvement for models that operate directly on raw sequences without any tokenization step.

Social Impacts. As a fundamental component in time series modeling, our work can enhance the performance of a wide range of time series methods on various real-world tasks, such as forecasting, anomaly detection, and classification. It can serve as a foundational module in improving the application of time series models in fields such as healthcare, transportation, and economics.

F Visualization of TimeFT

Since TimeFT is a tokenization method, we present visualizations comparing different classes of tokens in the time domain, frequency domain, and after equivalent reconstruction. As shown in Figures 9 to 15, patch tokens lead to (b) a completely different frequency spectrum and (c) cause the loss of all information outside the rectangular window. Phase tokens cause (e) frequency components above the Nyquist frequency to completely alias into the low frequency, thereby (f) exhibiting erroneous patterns. Our TimeFT (h) effectively solves the above issues and (i) achieves almost distortion-free tokenization.

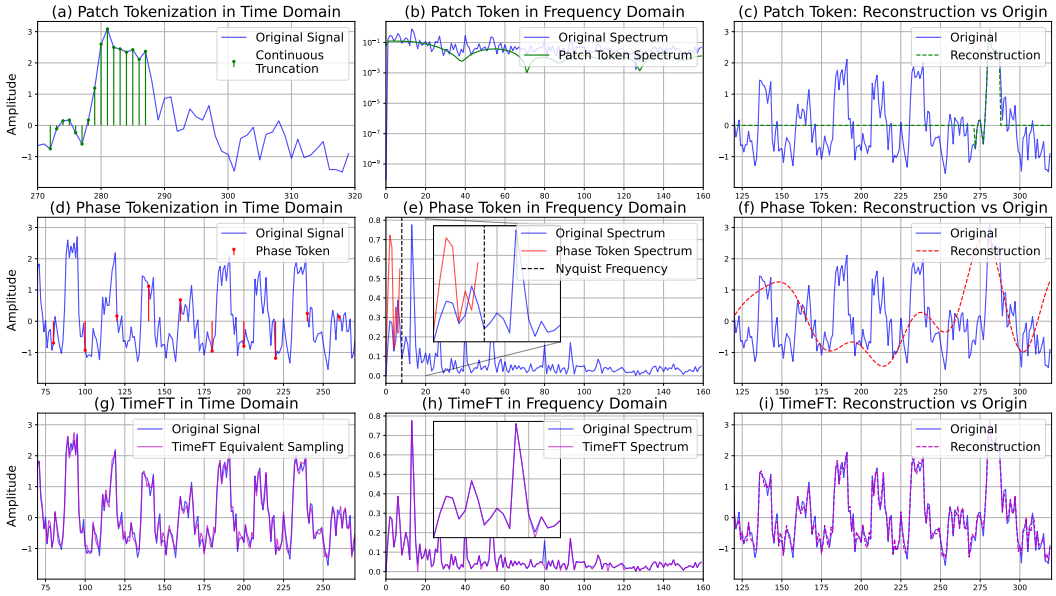


Figure 9: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in ETTh1 dataset.

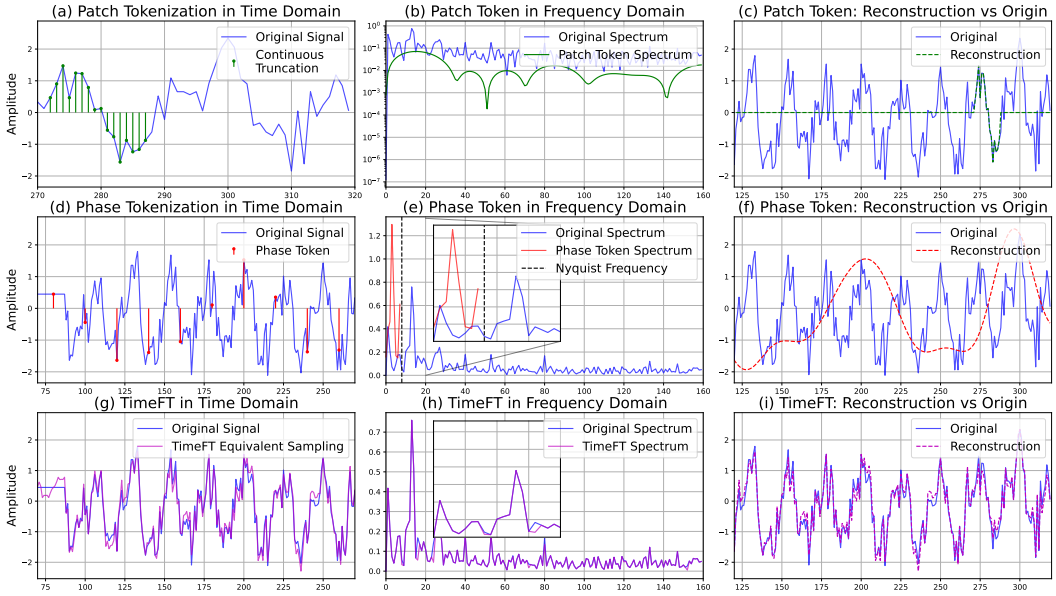


Figure 10: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in ETTh2 dataset.

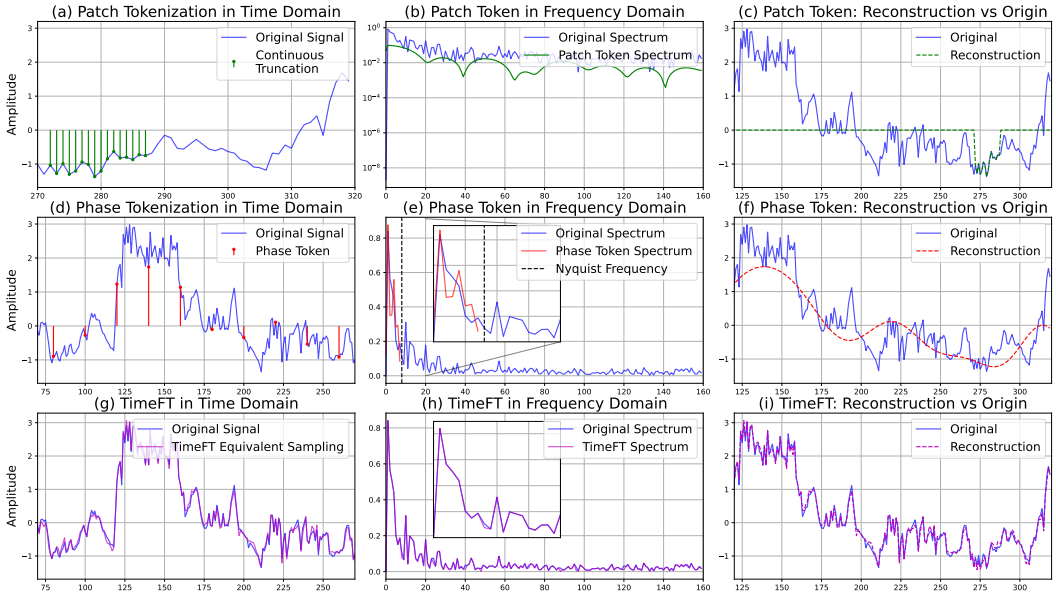


Figure 11: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in ETTm1 dataset.

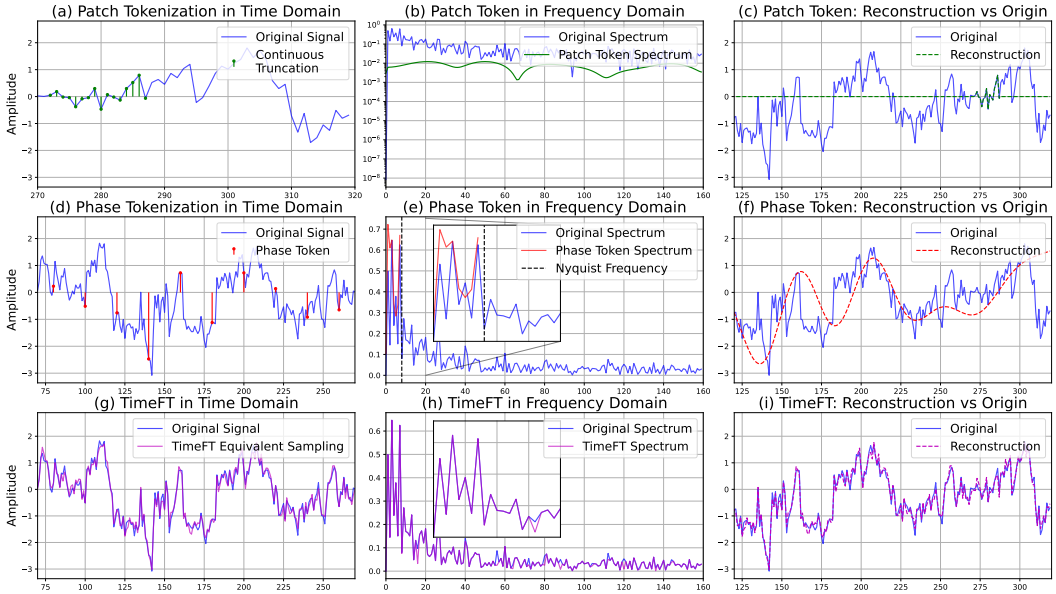


Figure 12: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in ETTm2 dataset.

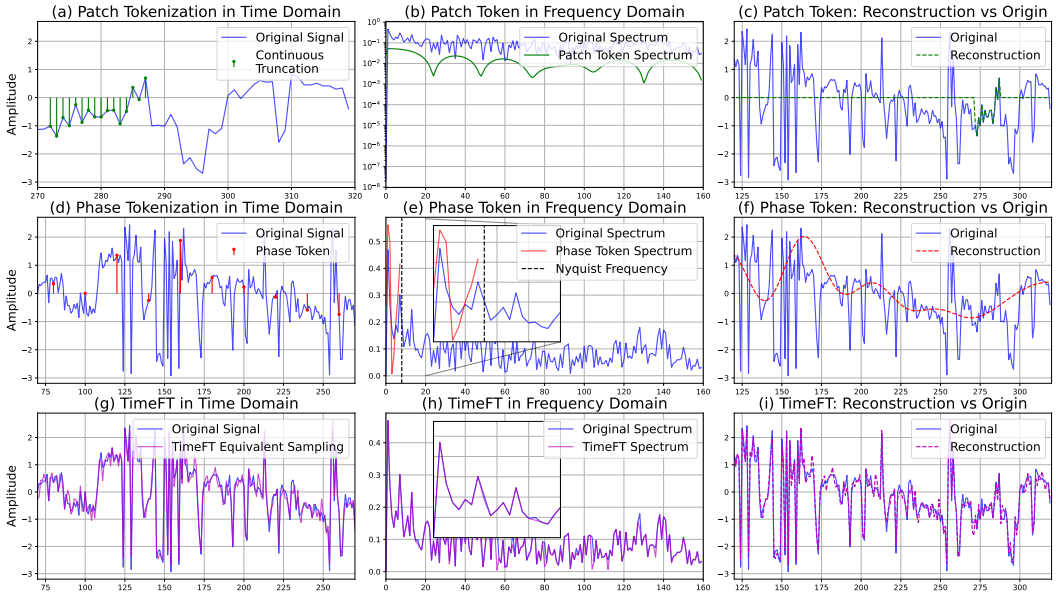


Figure 13: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in Weather dataset.

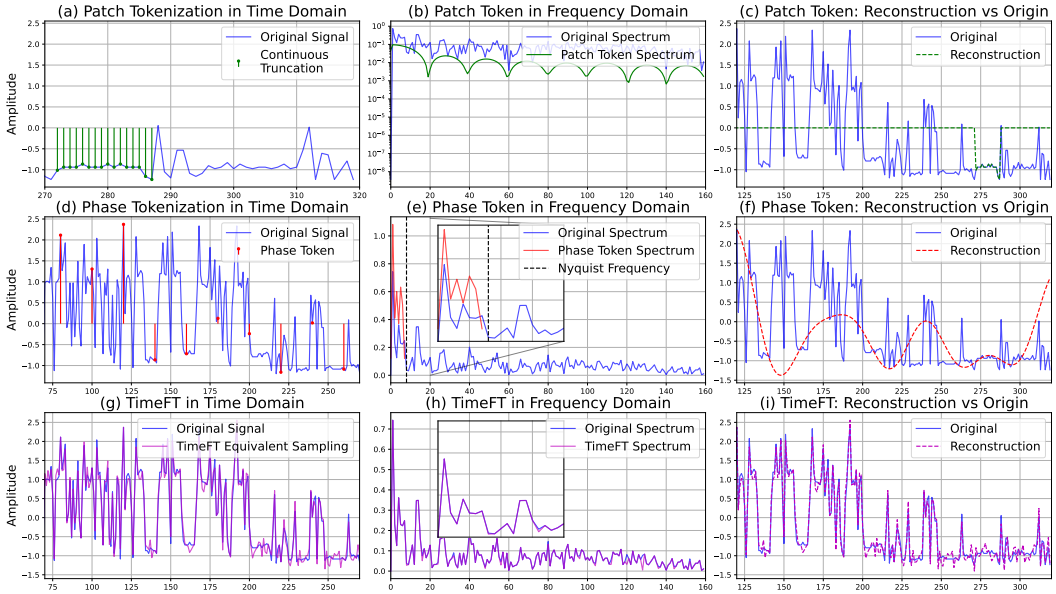


Figure 14: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in Electricity dataset.

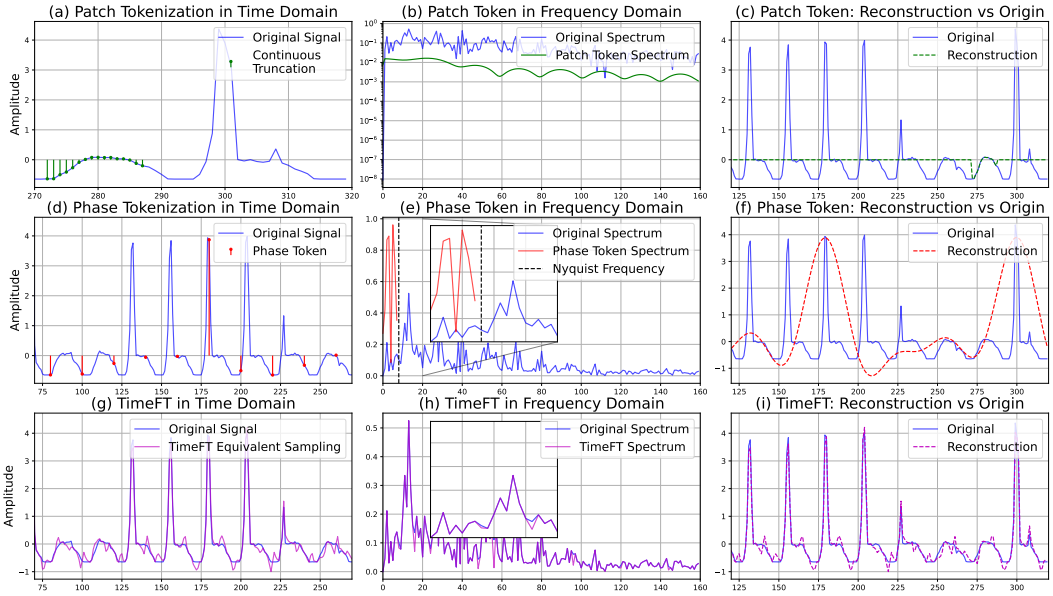


Figure 15: The visualization of tokenization methods across the time domain, frequency domain and reconstruction vs origin in Traffic dataset.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,

- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are well-explained throughout the abstract and introduction, and the entire paper is written to provide a consistent and comprehensive conclusion.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our paper discusses the limitations in the Limitation section E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our paper provides the full set of assumptions and a complete and correct proof in Section B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our paper provides detailed information to reproduce all experiments and the complete experiment code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please see the code link in Abstract.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Our paper provides detailed information about the experiments, including data, hyperparameters, how they were chosen, and the type of optimizer, in the main text and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Experiments are repeated three times and the results are reported in Tables. We also evaluate the sensitivity of hyperparameter in §4.7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reported the computer resources used in the experiments in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper fully conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our paper discusses the potential broader impacts of our research in the Appendix E.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our model does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly credited the creators or original owners of assets used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not include new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is not an important, original, or non-standard component of the core methods in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.