VarDrop: Enhancing Training Efficiency by Reducing Variate Redundancy in Periodic Time Series Forecasting

Junhyeok Kang[1], **Yooju Shin**[2], **Jae-Gil Lee**[2*]
[1]LG AI Research
[2]KAIST
junhyeok.kang@lgresearch.ai, {yooju.shin, jaegil}@kaist.ac.kr

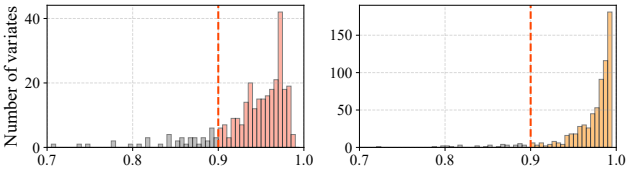
Abstract

Variate tokenization, which independently embeds each variate as separate tokens, has achieved remarkable improvements in multivariate time series forecasting. However, employing self-attention with variate tokens incurs a quadratic computational cost with respect to the number of variates, thus limiting its training efficiency for large-scale applications. To address this issue, we propose *VarDrop*, a simple yet efficient strategy that reduces the token usage by omitting redundant variate tokens during training. *VarDrop* adaptively excludes redundant tokens within a given *batch*, thereby reducing the number of tokens used for dot-product attention while preserving essential information. Specifically, we introduce k-dominant frequency hashing (k-DFH), which utilizes the ranked dominant frequencies in the frequency domain as a hash value to efficiently group variate tokens exhibiting similar periodic behaviors. Then, only representative tokens in each group are sampled through stratified sampling. By performing sparse attention with these selected tokens, the computational cost of scaled dot-product attention is significantly alleviated. Experiments conducted on public benchmark datasets demonstrate that *VarDrop* outperforms existing efficient baselines.

1 Introduction

Transformers have demonstrated impressive performance in time series forecasting, primarily due to their attention mechanisms (Trirat et al. 2024; Shin et al. 2024; Nie et al. 2023; Zhang and Yan 2023; Wu et al. 2021; Zhou et al. 2021). Traditionally, most methods have employed temporal tokenization, treating all variates at a given timestamp as a single token. However, recent studies reveal that *variate tokenization*—where each variate is embedded as a separate token—outperforms temporal tokenization in capturing inter-variate dependencies thereby increasing forecasting accuracy (Liu et al. 2024b). Due to its broad applicability to Transformers, variate tokenization has been adopted in recent advancements in multivariate time series forecasting (Liu et al. 2024c; Wang et al. 2024; Han et al. 2024).

Despite its advantages, the feasibility of variate tokenization in real-world applications is hindered by the increasing number of variates (N). Many public benchmark datasets demonstrate the high dimensionality often encountered in real-world applications. For instance, the Electricity dataset includes 321 variates, each corresponding to a customer, and the Traffic dataset comprises 862 variates, each representing a sensor (Wu et al. 2021). The large N introduces inefficiency, as the computational cost of attention mechanisms increases quadratically with N. This inefficiency results in a superfluous carbon footprint during the model training process. In addition, the large N can lead to overfitting and reduced attention performance by diluting important information (Peysakhovich and Lerer 2023; Liu et al. 2024a). Therefore, reducing N while retaining the essential information is critical for the success of variate tokenization.



(a) Electricity dataset. (b) Traffic dataset.

Figure 1: *Variate redundancy* in public periodic time series datasets. Most variates are highly correlated with others. For more detailed analysis, please refer to Appendix A.

To address this issue, we propose removing *variate redundancy* in multivariate time series. As different variates often share the same characteristics such as trends and seasonality at some timestamps, excessive correlations between them are frequently observed in periodic time series. Figure 1 shows the variate redundancy in the term of maximum Pearson correlation coefficient values (refer to Eq. (4)) between the variates in two datasets. For the Electricity and Traffic datasets, 79.4% and 94.7% of all variates exhibit strong correlations with other variates exceeding 0.9, respectively. By selecting a few informative and distinguishing variates, we can significantly reduce computational cost while preserving the original periodic information of the data.

However, efficiently picking out the representative variates is challenging for two reasons. First, the correlation between variates fluctuates due to inherent distribution shifts in time series data. This correlation shift in sliding windows generates a unique set of correlations for each batch as shown in

*Jae-Gil Lee is the corresponding author.

Appendix B. We should filter out redundant variates in an online manner due to this correlation shift. Second, computing similarity between variates requires extensive computational cost. Popular similarity metrics, such as the Pearson correlation coefficient and dynamic time warping (DTW), have computational complexity in the order of $O(N^2T^2)$ where T denotes the window length. In summary, we need a simple selection process with a fast similarity metric to leverage variate redundancy effectively.

To this end, we propose a simple yet efficient training strategy for variate tokens called *VarDrop*. Given a multivariate time series, *VarDrop* performs fast Fourier transform for each variate and identifies k-dominant frequencies whose amplitude values are top-k in frequency domain. By ranking k frequencies in the descending order of amplitude, k-dominant frequency hashing (k-DFH) generates a meaningful hash for each variate. *VarDrop* groups the variates with the same hash and select representative variate tokens for each group through stratified random sampling. Scaled dot-product attention is then computed with selected tokens where each represents unique temporal patterns in each batch. As a result, *VarDrop* resolves variate redundancy and successfully reduces the number of tokens for each batch without incurring significant computational overhead.

Here we summarized our key contributions as follows:

- We propose a simple yet efficient training strategy, *VarDrop*, which significantly improves computational efficiency by disregarding redundant variates in variate-tokenized Transformers.
- *VarDrop* leverages k-dominant frequency hashing that efficiently identifies similar tokens in each batch by applying the fast Fourier transform.
- We demonstrate the effectiveness of *VarDrop* through extensive experiments conducted on four benchmark datasets, comparing to state-of-the-art methods.

2 Related Work

2.1 Tokenization Strategies for Time Series Data

Inspired by the success of Transformers in natural language processing, numerous Transformers have been proposed for multivariate time series forecasting. Previous studies, such as Autoformer (Wu et al. 2021), Fedformer (Zhou et al. 2022), Crossformer (Zhang and Yan 2023), and PatchTST (Nie et al. 2023), adopted a temporal tokenization method similar to language models, treating the values of all variates at a specific timestamp as a single token. The multiple variates are not considered individually, but processed as a whole when generating representations for each temporal token. As a result, these embedded temporal tokens fail to properly capture the correlations between different variates in multivariate time series. Moreover, temporal tokens could not consider relevant contexts due to the narrow receptive field (Shin et al. 2023).

To address the limitations in temporal tokenization, iTransformer introduced variate tokenization (Liu et al. 2024b). It is a special type of tokenization as the various tokenization methods used in Flowformer (Wu et al. 2022) and treats each variate as one token, aiming to better model the variate dependencies. After the success of iTransformer, variate tokenization became a prevalent technique in forecasting models. Timer merges multiple variates from different domain into a single time series and treats the time series as a single token (Liu et al. 2024c). MCformer tokenizes each variate and mixes the variates to capture inter-variate correlations (Han et al. 2024). TimeXer also leverages variate tokenization in the introduction of exogenous variates (Wang et al. 2024).

2.2 Efficient Transformers for Time Series Data

Most previous efficient methods are designed for temporal tokens with sequential nature. Due to a number of tokens in the temporal axis, the sparse attention is prevalent to reduce the number of tokens in attention value computation. Here, a portion of query-key pairs is only considered instead of computing every query-key pair. LogSparse is one of the early methods for sparse attention, matching a query to the previous keys with an exponential step size and itself, showing the similar behavior in causal convolution (Li et al. 2019). Big-Bird suggests a compound sparse attention method, containing global, local, and random query-key matching strategies (Zaheer et al. 2020). Reformer applies locality-sensitive hashing, which makes a chuck of similar tokens in an input sequence (Kitaev, Kaiser, and Levskaya 2020). These methods rely on the temporal locality of input sequences, making them unsuitable for variate tokens lacking a sequential nature.

There are also efficient Transformers not based on temporal locality between timestamps (Shin et al. 2022). Informer selects the dominant query-key pairs that has more influence in attention value computation (Zhou et al. 2021). Autoformer adopts Fourier-based auto-correlation computation in the attention to reduce the computational complexity from $O(T^2)$ to $O(T\log T)$ (Wu et al. 2021). Pyraformer constructs a pyramidal graph in matching query-key pairs to scale the attention module into longer sequences (Liu et al. 2022). FEDformer randomly selects a fixed number of Fourier components in time series to have linear computational complexity in the forward pass (Zhou et al. 2022). Crossformer proposes two-stage attention for temporal dimension and variate dimension to reduce the computation complexity from $O(N^2T^2)$ to $O(N^2T)$ (Zhang and Yan 2023). However, these methods do not consider variate redundancy in multivariate time series for boosting efficiency in Transformers.

3 Proposed Method: *VarDrop*

Problem Definition. Given a multivariate periodic time series $\mathcal{X} \in \mathbb{R}^{N\times T}$, where N is the number of variates and T is the length of the time series, *multivariate time series forecasting* aims to predict the forecast horizon $\mathcal{X}_{t+1:t+H} \in \mathbb{R}^{N\times H}$ based on a lookback window $\mathcal{X}_{t-T+1:t} \in \mathbb{R}^{N\times T}$. Variate tokenization converts a lookback window into the variate tokens $\mathcal{V} \in \mathbb{R}^{N\times d}$ through a embedding layer composed of multi-layer perceptron (MLP) shared by each variate. Note that d is the dimension of the embedding. A variate-tokenized Transformer model then predicts the future values $\hat{\mathcal{X}}_{t+1:t+H} \in \mathbb{R}^{N\times H}$ for multivariate time series forecasting.

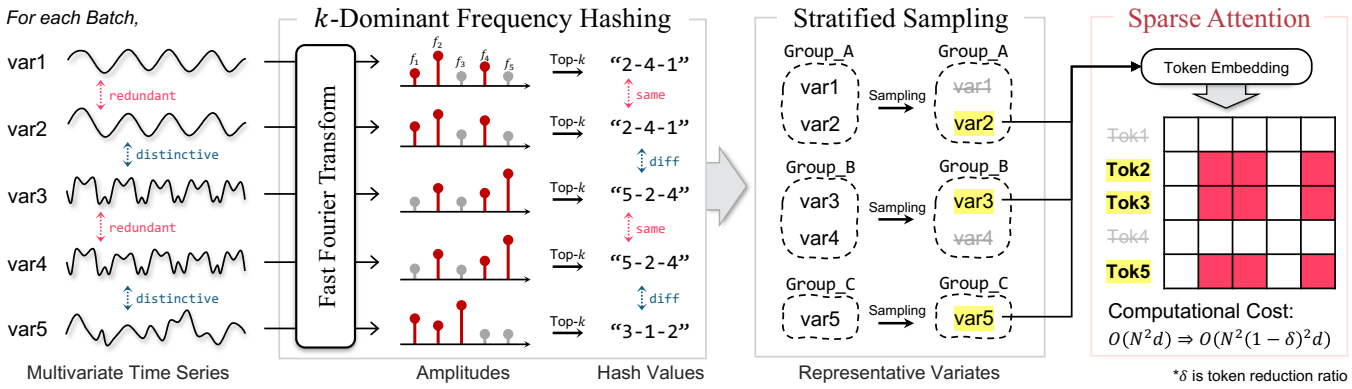


Figure 2: Overall procedure of *VarDrop*. Given a batch of multivariate time series, the hash values representing the top-k amplitudes for each variate are generated through k-Dominant Frequency Hashing (k-DFH). Then, stratified sampling is conducted based on groups of variates that share the same hash value to omit redundant variates. Finally, sparse attention is performed on the reduced set of variate tokens, enabling efficient training.

3.1 Overview of *VarDrop*

Figure 2 illustrates the overall procedure of *VarDrop*. When a batch of multivariate periodic time series containing N variates is given, k-dominant frequency hashing generates a hash value for each variate in the frequency domain, resulting in a total of N hash values denoted as $\mathcal{H} \in \mathbb{R}^N$. These values represent the inherent periodic behaviors of each variate, enabling efficient grouping of variates with similar patterns. Some of the redundant variates that share the same patterns with others are then disregarded by stratified sampling. Using the selected $N(1-\delta)$ variate tokens, the computational complexity of self-attention is reduced from $O(N^2d)$ to $O(N^2(1-\delta)^2d)$, where d is the embedding size and δ is the token reduction ratio. Similar to Dropout (Srivastava et al. 2014), *VarDrop* is utilized only during the training stage, as multivariate time series forecasting requires predictions for all variables during the inference stage.

3.2 Efficient Adaptive Variate Token Grouping

To achieve efficient sparse attention by disregarding redundant tokens, it is crucial to first identify groups of highly correlated variates. By leveraging the variate redundancy, we can selectively focus attention on the most significant tokens, thereby reducing computational complexity.

k-Dominant Frequency Hashing. We introduce k-dominant frequency hashing (k-DFH), a simple solution that enables efficient grouping of variates in the frequency domain using the fast Fourier transform (FFT). To explain k-DFH, we first define the k-dominant frequency in Definition 3.1.

Definition 3.1 (k-DOMINANT FREQUENCY)**.** Given a time series x, after performing the Fourier transform, a frequency f is *dominant* if its corresponding amplitude spectrum $\mathcal{A}_f$ is among the top-k amplitude values.

According to Fourier theorem, a time series can be modeled with few Fourier spectra in the frequency domain (Brigham 1988). Thus, the overall periodic behaviors of variates can be successfully condensed as the proper k dominant frequencies while ignoring unessential properties.

Algorithm 1: k-Dominant Frequency Hashing

INPUT: A batch of multivariate time series data $\mathcal{B}$, the number of variates N, batch size B, the number of dominant frequencies k, a cutoff frequency ε.
OUTPUT: Hash values $\mathcal{H} \in \mathbb{R}^N$

1: $\mathcal{A} \leftarrow$ `Fast_Fourier_Transform`$(\mathcal{B})$;
2: $\hat{\mathcal{A}} \leftarrow$ `Low_Pass_Filter`$(\mathcal{A}, \varepsilon)$;
3: $\bar{\mathcal{A}} \leftarrow \frac{1}{|B|}\sum_{j=1}^{B} \hat{\mathcal{A}}_j$;
4: $\mathcal{F}^* \leftarrow$ `Top-k_Frequency`$(\bar{\mathcal{A}}, k)$;
5: $\mathcal{H} \leftarrow$ `Generate_Hash_Value`$(\mathcal{F}^*)$;
6: **return** $\mathcal{H} \in \mathbb{R}^N$

k-DFH uses the order of these frequencies as a hash value, enabling efficient clustering of correlated variates.

Procedure of k-DFH. Algorithm 1 outlines the k-DFH procedure. Consider a batch of time series $\mathcal{B} = \{\mathcal{X}_1, \ldots, \mathcal{X}_B\} \in \mathbb{R}^{B \times N \times T}$, where B represents the batch size. First, each variate $\mathcal{X}^{(i)} \in \mathbb{R}^T$ in the time domain is converted to the frequency domain through Fast Fourier Transform and real amplitudes $\mathcal{A}$ are only retained (Line 1). To support efficient operations, the frequency components above a specified cutoff frequency ε are removed by incorporating a low-pass filter (LPF) while preserving low-frequency information (Line 2). Then, the amplitude values are averaged across instances within the batch to support batch-wise grouping, thereby alleviating instance-level noise (Line 3). After that, the k dominant frequencies $\mathcal{F}^* \in \mathbb{R}^k$ are identified based on these amplitude values $\bar{\mathcal{A}}$ (Line 4). Finally, k dominant frequencies $\mathcal{F}^*$ are utilized as the input of hash function and generates hash value $\mathcal{H} \in \mathbb{R}^N$ for variate tokens (Line 5). Based on the hash values, the variates with the same hash value form a group, leading to multiple distinct groups that represent unique periodic behaviors, as shown in Figure 2.

Effect of Hyperparameters. The k-DFH algorithm has a hyperparameter k that determines the granularity of groups. Increasing k results in a more fine-grained grouping, which reduces diversity between variates within groups and en-

hances the their similarity. Conversely, decreasing the value of k leads to a coarse-grained grouping approach, resulting in larger differences in periodicity between groups and diminished intra-group similarity. The choice of k is thus crucial and depends on the specific characteristics of the time series data and the desired balance between granularity and variability. We provide the experimental results concerning the impact of the k value on the forecasting results and token reduction ratios in Section 4.5. Because there is a trade-off between performance degradation and efficiency, the optimal selection of two hyperparameters depends on the application requirement. We also report its empirical evidence in Figure 4 of Section 4. For another minor hyperparameter ε of LPF, by selecting an appropriate cut-off frequency, *VarDrop* enables more efficient and robust grouping by enhancing the representativeness of hash values while keeping essential information (Xu, Zeng, and Xu 2024).

Time Complexity Analysis. The k-DFH algorithm comprises steps including fast Fourier transform, low-pass filtering, amplitude averaging, identification of k dominant frequencies, and hash value generation. The overall time complexity is dominated by the fast Fourier transform step, which has a complexity of $O(B \cdot N \cdot T \log T)$. Subsequent steps, involving low-pass filtering and amplitude averaging, have lower complexities of $O(B \cdot N \cdot \varepsilon)$ each. Finding the k dominant frequencies among the averaged amplitude values $\bar{A}$ requires sorting algorithm. Using an efficient sorting algorithm, the complexity is $O(N \cdot \varepsilon \log \varepsilon)$ for this step as there are ε frequencies in $\bar{A}$. Therefore, the overall time complexity of the k-DFH algorithm becomes $O(B \cdot N \cdot T \log T)$, making the algorithm efficient for datasets with large N. In Section 4.3, we empirically verify that the runtime of the proposed method is significantly faster than existing efficient methods, demonstrating the practicality of k-DFH.

Theoretical Analysis. The rationale behind k-DFH is that variates exhibiting similar periodic behaviors have the same dominant frequencies in the frequency domain. To provide its theoretical justification, we formalize this in Theorem 3.2 where the *proof* is provided in Appendix C.

Theorem 3.2 (ERROR OF k-DFH APPROXIMATION)**.** *The error between a time series and its reconstructed signal from its hash value through k-DFH is given by the cumulative contribution of the non-dominant frequencies.*

According to Theorem 3.2, the k dominant frequencies capture the majority of the signal's energy, and the reconstruction error using only these frequency components remains relatively small. This ensures that the errors between variates sharing the same hash value do not significantly deviate from one another. To support empirical evidence, we include visualizations of the variate groups generated by k-DFH in Section 4.4. These visualizations illustrate that the variates within the same group exhibit similar periodic behavior, consistent with the theoretical expectations.

Noise Robustness. One of the key advantages of k-DFH is its robustness to noise. Let $\mathcal{X}(t)$ be a time series composed of trends $\mathrm{T}(t)$, seasonality $\mathrm{S}(t)$ and noise $\mathrm{E}(t)$, such that $\mathcal{X} = \mathrm{T}(t) + \mathrm{S}(t) + \mathrm{E}(t)$. The frequency spectrum $\Phi(\mathcal{X})$ can be expressed as the sum of the frequency spectrums of the signal and noise: $\Phi(\mathcal{X}) = \Phi(\mathrm{T}) + \Phi(\mathrm{S}) + \Phi(\mathrm{E})$. Noise $\mathrm{E}(t)$ typically manifests as lower-amplitude components spread across the frequency spectrum, while the trends $\mathrm{T}(t)$ and seasonality $\mathrm{S}(t)$ contributes higher-amplitude components at specific frequencies. Since $\Phi(\mathrm{E})$ contributes relatively low amplitudes, the dominant frequencies below cutoff frequency ε are predominantly those of $\mathrm{T}(t)$ and $\mathrm{S}(t)$, making k-DFH invariant to undesirable high-frequency noise.

3.3 Sparse Attention via Stratified Sampling

Variate Reduction using Stratified Sampling. By leveraging the k-DFH, similar variates can be grouped based on their dominant frequencies in the frequency domain. Once these groups are formed, stratified sampling is applied to selectively retain a subset of variates within each group, thereby eliminating redundant variate tokens. Formally, let $\mathcal{G}_i$ denote the set of variates in the i-th group, and let gs, a hyperparameter representing the group size, determine the maximum number of variates to retain per group. The retained subset of variates from group $\mathcal{G}_i$, denoted as $\mathcal{S}_i$, is defined as

$$\mathcal{S}_i \subseteq \mathcal{G}_i \quad \text{and} \quad |\mathcal{S}_i| = \min(|\mathcal{G}_i|, gs). \tag{1}$$

The set of all variates retained across all groups, denoted as $\mathcal{S}$, is the union of the retained subsets from each group:

$$\mathcal{S} = \cup_{i=1}^{G} \mathcal{S}_i, \tag{2}$$

where G is the total number of generated groups. Our method is not limited to any specific sampling method during the stratified sampling process. Any sampling technique can be freely chosen based on the requirements of the application. In this study, we adopted random sampling as the sampling method due to its ease of implementation.

Efficient Self-Attention Disregardng Redundant Variates Tokens. In Transformers employing variate tokens, the role of self-attention mechanisms is capturing dependencies between variates (Liu et al. 2024b; Wang et al. 2024). The self-attention mechanism computes attention scores between each pair of variate tokens, resulting in a computational complexity of $O(N^2 d)$, where N is the number of tokens and d is the embedding dimension. The attention scores are computed using the scaled dot-product attention, defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where $Q \in \mathbb{R}^{N \times d_k}$, $K \in \mathbb{R}^{N \times d_k}$, and $V \in \mathbb{R}^{N \times d_k}$ are the query, key, and value matrices, respectively, and d_k is the dimension of the query, key, and value vectors. The bottleneck in the variate-tokenized Transformer lies in the quadratic computational complexity with respect to the number of tokens. As the number of tokens can be reduced by a token reduction ratio $\delta = 1 - \frac{1}{N}\sum_i^G \min(|\mathcal{G}_i|, gs)$, the computational cost of self-attention decreases significantly, resulting in a complexity of $O(N^2(1-\delta)^2 d)$. Moreover, the proposed method performs token reduction directly from raw variates, enhancing efficiency by eliminating the need for token embedding of redundant variates. Additionally, *VarDrop* is an architecture-free method that can be applied to any type of Transformer using various tokens.

Type		Efficient Variate-Tokenized Transformers										Ground	
Model		*VarDrop*		FlashAttention		Flowformer		Reformer		Informer		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	0.153	0.245	0.178	0.265	0.183	0.267	0.182	0.275	0.190	0.286	0.150	0.242
	192	0.167	0.257	0.189	0.276	0.192	0.277	0.192	0.286	0.201	0.297	0.166	0.256
	336	0.183	0.275	0.207	0.294	0.210	0.295	0.210	0.304	0.218	0.315	0.184	0.276
	720	0.220	0.305	0.251	0.329	0.255	0.332	0.249	0.339	0.255	0.347	0.214	0.302
	Avg	**0.181**	**0.271**	0.206	0.291	0.210	0.293	0.208	0.301	0.216	0.311	0.178	0.269
Traffic	96	0.396	0.274	0.464	0.320	0.493	0.339	0.617	0.356	0.632	0.367	0.398	0.272
	192	0.417	0.281	0.479	0.326	0.506	0.345	0.629	0.361	0.641	0.370	0.418	0.279
	336	0.435	0.289	0.501	0.337	0.526	0.355	0.648	0.370	0.663	0.379	0.431	0.286
	720	0.472	0.308	0.524	0.350	0.572	0.381	0.694	0.394	0.713	0.405	0.465	0.304
	Avg	**0.430**	**0.288**	0.492	0.333	0.524	0.355	0.647	0.370	0.662	0.380	0.428	0.285
Weather	96	0.178	0.218	0.177	0.218	0.183	0.223	0.169	0.225	0.180	0.251	0.176	0.216
	192	0.227	0.258	0.229	0.261	0.231	0.262	0.213	0.265	0.244	0.318	0.225	0.257
	336	0.281	0.297	0.283	0.300	0.286	0.301	0.268	0.317	0.282	0.343	0.281	0.299
	720	0.357	0.347	0.360	0.352	0.363	0.352	0.340	0.361	0.377	0.409	0.358	0.350
	Avg	0.261	**0.280**	0.262	0.283	0.266	0.285	**0.248**	0.292	0.271	0.330	0.260	0.281
Solar-Energy	96	0.202	0.238	0.205	0.237	0.208	0.239	0.203	0.241	0.198	0.237	0.205	0.236
	192	0.237	0.262	0.239	0.263	0.244	0.266	0.236	0.266	0.232	0.263	0.239	0.263
	336	0.254	0.275	0.250	0.275	0.258	0.277	0.249	0.276	0.248	0.276	0.249	0.273
	720	0.252	0.274	0.252	0.277	0.259	0.279	0.253	0.279	0.251	0.278	0.250	0.275
	Avg	0.236	**0.262**	0.236	0.263	0.242	0.265	0.235	0.265	**0.232**	0.264	0.236	0.262
Overall		**0.277**	**0.275**	0.299	0.292	0.311	0.299	0.334	0.307	0.345	0.321	0.275	0.274
Relative error (%)		**0.6**	**0.4**	8.6	6.6	12.7	9.2	21.4	12.0	25.4	17.2	-	-

Table 1: Performance comparison of multivariate time series forecasting with four efficient baselines on the four benchmark datasets. GROUND refers to the ground-truth performance using all variate tokens. Variate tokenization is used for all compared methods. The forecasting horizon T $\in \{96, 192, 336, 720\}$ for all methods. The results are mostly taken from Liu et al. (2024b).

4 Experiments

In this section, we compare the proposed *VarDrop* with efficient Transformer baselines using public benchmark datasets for multivariate time series forecasting, evaluating both (i) forecasting performance and (ii) training efficiency to demonstrate *VarDrop*'s effectiveness.

4.1 Experiment Settings

Datasets. We conducted experiments on four real-world multivariate time series datasets, each containing a large number of variates: Electricity, Traffic, Weather, and Solar-Energy (Wu et al. 2021). Further details on the data description are also provided in Appendix D.

Baselines. We compare the performance of *VarDrop* with four efficient Transformers: Flowformer (Wu et al. 2022), FlashAttention (Dao et al. 2022), Reformer (Kitaev, Kaiser, and Levskaya 2020) and Informer (Zhou et al. 2021). FlashAttention reduces GPU memory bottlenecks by tiling vectors used in attention computation. All efficient baseline methods are modified to use variate tokenization following the previous study (Liu et al. 2024b).

Backbone Model and Ground Truth. To objectively evaluate *VarDrop*, we adopted iTransformer (Liu et al. 2024b), a vanilla Transformer designed for the variate tokenization strategy, as our backbone model. We then used its dense attention results as the ground truth.

Evaluation Metrics. For evaluation metrics, we choose the mean squared error (MSE) and the mean absolute error (MAE), consistent with previous work (Liu et al. 2024b).

Implementation Details. Following previous studies (Liu et al. 2024b), we adopt same configuration for choosing optimization hyperparameters, such as the learning rate. In the experimental setup, the hyperparameters k and gs were selected from $\{3, 4\}$ and $\{5, 10, 20\}$, respectively. The source code of *VarDrop* is publicly available online, and further implementation details can be found in Appendix E.

4.2 Overall Performance Comparison

Forecasting Performance. Table 1 summarizes the forecasting results of *VarDrop*, including ground-truth and efficient baselines across four real-world datasets. *VarDrop* demonstrates superior performance, achieving the lowest relative MSE and MAE, both less than 1% compared to the ground truth. Please note that *VarDrop* accomplishes this high performance while utilizing a significantly reduced number of tokens for the attention process. *VarDrop* particularly excels in high-dimensional datasets such as Electricity and Traffic, increasing the feasibility of variate tokenization in large-scale applications. The high relative errors of efficient baselines stem from the improper handling of variate tokens, as they were designed for temporal tokens. To verify the robustness of *VarDrop*, we also provide the standard deviation of its

Method	Flowformer +*VarDrop*		Reformer +*VarDrop*		Informer +*VarDrop*	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
96	0.176	0.268	0.167	0.258	0.190	0.279
192	0.183	0.273	0.179	0.268	0.192	0.280
336	0.202	0.291	0.196	0.285	0.210	0.298
720	0.245	0.325	0.239	0.320	0.257	0.335
Avg	**0.202**	**0.289**	**0.195**	**0.283**	**0.212**	**0.298**

Table 2: Forecasting results of existing efficient Transformers with *VarDrop* on the Electricity dataset.

Dataset	# Used Tokens	# Variates	Reduction Ratio
Electricity	117.7±3.8	321	**63.33%**
Traffic	188.4±20.6	862	**78.14%**
Weather	7.1±1.3	21	**66.19%**
Solar-Energy	20.0±0.3	137	**85.38%**

Table 3: Token reduction results with standard deviations.

performance across five independent runs initialized with different random seeds in Appendix F.

Integration with Baselines. Due to its modular design, *VarDrop* can be easily attached to existing efficient methods. Specifically, by first applying *VarDrop* to reduce the number of tokens, the resulting reduced variates can be effectively used as input for existing Transformers. Table 2 presents the forecasting results of three efficient baselines and the corresponding results when integrated with *VarDrop*. This demonstrates that *VarDrop* further enhances the performance of efficient baselines by eliminating redundant variables.

4.3 Efficiency of *VarDrop*

Token Reduction Results. Table 3 presents the token reduction ratios achieved by *VarDrop* across four benchmark datasets: Electricity, Traffic, Weather, and Solar-Energy. Our method adaptively identifies and drops redundant variate tokens during the training stage for each batch. Therefore, we report the average number of tokens over all iterations within an epoch, along with the corresponding standard deviation. Table 3 verifies that *VarDrop* significantly reduces the number of tokens required for training. For example, in the Traffic dataset, *VarDrop* uses an average of 188.4±20.6 tokens out of 862, achieving a token reduction ratio of 78.14%. Similarly, the Solar-Energy dataset exhibits an impressive reduction ratio of 85.38%, with only 20.0±0.3 tokens used out of 137. These results demonstrate that *VarDrop* can be adopted for large-scale applications due to its efficiency and scalability.

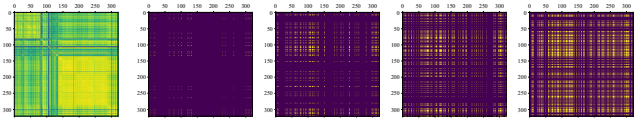
Comparison of Training Times. To validate the efficiency of *VarDrop*, we compared the average running time per iteration with efficient baselines during the training stage. Table 4 illustrates that *VarDrop* achieves the lowest average training time compared to the baselines. The ranking of training speeds for efficient baselines varied depending on the dataset. Notably, *VarDrop* increased the training speed of iTransformer from 68ms/iter to 33ms/iter, improving it by 2.06 times.

Method	Electricity	Traffic	Weather	Solar	*Avg*
iTransformer	40.9	198.5	19.4	**13.1**	68.0
Informer	40.2	<u>118.9</u>	<u>17.0</u>	16.6	<u>48.2</u>
Flowformer	<u>38.4</u>	123.1	24.9	17.1	50.9
Reformer	84.7	308.2	21.0	27.0	110.2
VarDrop	**30.8**	**72.7**	**12.6**	<u>15.9</u>	**33.0**

Table 4: Comparison of training times on benchmark datasets with an input-96-predict-96 setting. The unit is *ms/iteration*.

Method	Memory Footprint
iTransformer	3.41 GB
Informer	2.87 GB
Flowformer	3.33 GB
Reformer	4.12 GB
Flashformer	<u>2.87 GB</u>
VarDrop	**2.22 GB**

Table 5: Comparison of GPU memory footprints of efficient baselines on Electricity with an input-96-predict-96 setting.



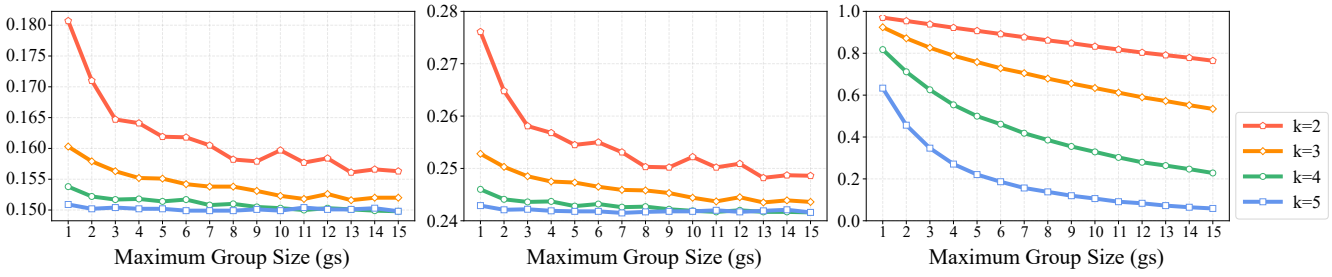
(a) Correlation. (b) gs=1. (c) gs=5. (d) gs=10. (e) gs=20.

Figure 3: Correlation matrix and corresponding sparse matrices with varying gs values on the Electricity dataset.

Comparison of Memory Footprints. We also report the results of comparing the GPU memory footprints of the baselines in Table 5. We observed that our proposed methodology uses less memory than all other efficient baselines. Remarkably, *VarDrop* utilizes 2.22 GB of memory, which is 65.1% of the memory foorprint of iTransformer. This result is consistent with the token reduction ratio presented in Table 3. As discussed in Section 3.3, this improved efficiency is attributed to the high reduction ratios, achieving $O(N^2(1-\delta)^2 d)$. This significant reduction in computational overhead allows the Transformers exploiting variate tokens to reduce training time and memory usage. Overall, these results suggest that *VarDrop* is a promising approach for efficient training using variate tokenization.

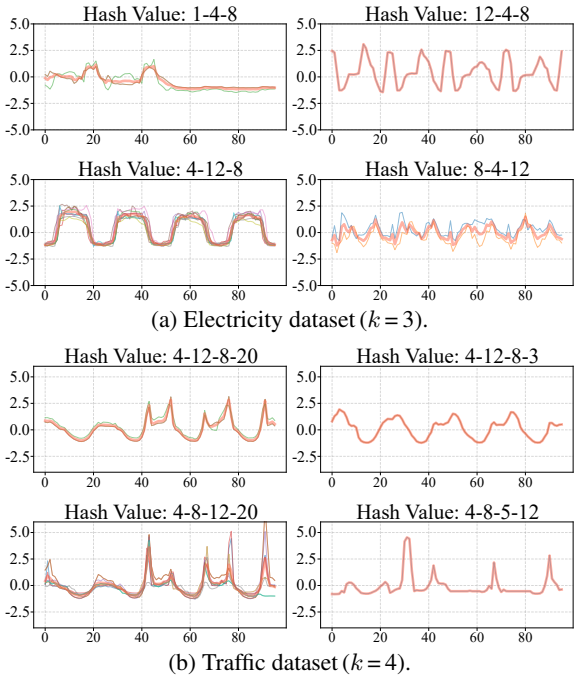
4.4 Qualitative Analysis through Visualization.

Effects of Maximum Group Size. Increasing the group size gs results in a higher density of the sparse matrix, leading to the lower level of token reduction ratio δ. Figure 3 demonstrates how changes in the group size gs affect sparse matrices on the Electricity dataset. The token reduction ratios δ for each matrix are as follows: (b) 93.1%, (c) 76.1%, (d) 63.5%, and (e) 47.3%. The redundant variates indicated in yellow in Figure 3(a) are dropped less frequently even with large gs values via stratified sampling . Please refer to Figure 10 of Appendix G for more visualization results on other benchmark datasets.



(a) MSE (↓). (b) MAE (↓). (c) Token Reduction Ratio δ (↑).

Figure 4: Forecasting results with input-96-predict-96 setting with their token reduction ratios for varying two hyperparameters $k \in \{2, 3, 4, 5\}$ and $gs \in \{1, \ldots, 15\}$ on the Electricity dataset.



(a) Electricity dataset ($k = 3$). (b) Traffic dataset ($k = 4$).

Figure 5: Visual examples of variates grouped by k-DFH. The red lines represent the mean values of the variates within each group, illustrating the distinctive periodic patterns. The hash values represent the ordered set of k dominant frequencies.

Visualization of k-DFH's Grouping Results. Figure 5 illustrates various grouping results obtained from k-DFH with each hash value. Each subfigure demonstrates how the variates are clustered based on their hash values, representing overall periodic behaviors. In this visualization, the order of dominant frequencies distinguishes different patterns. In Figure 5(a), two variates with hash values 4-12-8 and 12-4-8 display distinct periodic behaviors despite having the same set of dominant frequencies. This verifies the effectiveness of k-DFH in grouping only similar variates together, as proved in Theorem 3.2. In Figure 5(b), the variate with the hash value 4-8-5-12 exhibits unique spikes that distinguish it from other samples, resulting in a different hash value. This implies that the k-DFH method can effectively isolate variates that do not share common periodic characteristics with others. Overall, k-DFH efficiently assigns highly correlated variates to the same clusters. Please refer to Figure 9 for more results.

4.5 Hyperparameter Sensitivity Analysis

To examine the effect of *VarDrop*'s two crucial hyperparameters, k and gs, we conducted a sensitivity analysis by varying these values. Figure 4 illustrates the two forecasting errors (MSE and MAE) and the token reduction ratio δ for different values of the length of hash value $k \in \{2, 3, 4, 5\}$ and the maximum group size $gs \in \{1, \ldots, 15\}$ on the Electricity dataset. As shown in these results, as the value of k increases, the MSE and MAE decrease, while the token reduction ratio δ increases. Interestingly, there are sweet spots in determining the proper levels of k and gs. For instance, the forecasting errors and the reduction ratio when $k = 4$ and $gs = 1$ outperform those when $k = 3$ and $gs = 5$. This observation indicates that a higher reduction ratio does not necessarily result in lower forecasting performance and shows that *VarDrop*'s originality differs from random sampling. This is because, as k increases, the groups generated by k-DFH can represent the underlying behaviors in the data with greater precision, as evidenced in Theorem 3.2. Consequently, selecting the appropriate levels of these two hyperparameters k and gs can significantly enhance both accuracy and computational efficiency, tailored to the specific needs of the application.

5 Conclusion

This paper introduces a simple yet efficient training strategy for Transformers using variate tokenization, named *VarDrop*, for periodic time series forecasting. *VarDrop* adaptively identifies groups of variates that exhibit similar behaviors through k-DFH. The number of variate tokens is then reduced by disregarding redundant variates within each group via stratified sampling. By dropping these redundant variate tokens for each batch, the training efficiency of the attention mechanism is significantly enhanced. Experimental results on benchmark datasets demonstrate that the proposed method outperforms state-of-the-art efficient methods. Furthermore, due to its modularity, our approach can be easily applied to existing Transformers utilizing variate tokens. We hope that our work enhances the potential of variate tokenization in large-scale applications with numerous variables.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II200862, DB4DL: High-Usability and Performance In-Memory Distributed DBMS for Deep Learning, 50% and No. RS-2022-II220157, Robust, Fair, Extensible Data-Centric Continual Learning, 50%).

References

Brigham, E. O. 1988. *The Fast Fourier Transform and Its Applications*. Prentice-Hall, Inc.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and Memory-Efficient Exact Attention with OI-Awareness. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 16344–16359.

Han, W.; Member, T. Z.; Chen, L.; Ning, H.; Luo, Y.; and Wan, Y. 2024. MCformer: Multivariate Time Series Forecasting with Mixed-Channels Transformer. *IEEE Internet of Things Journal*, 11(17): 28320–28329.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The Efficient Transformer. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024a. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024b. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024c. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *Proceedings of International Conference on Machine Learning (ICML)*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Peysakhovich, A.; and Lerer, A. 2023. Attention Sorting Combats Recency Bias In Long Context Language Models. arXiv:2310.01427.

Shin, Y.; Park, J.; Yoon, S.; Song, H.; Lee, B. S.; and Lee, J.-G. 2024. Exploiting Representation Curvature for Boundary Detection in Time Series. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Shin, Y.; Yoon, S.; Kim, S.; Song, H.; Lee, J.-G.; and Lee, B. S. 2022. Coherence-based Label Propagation over Time Series for Accelerated Active Learning. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Shin, Y.; Yoon, S.; Song, H.; Park, D.; Kim, B.; Lee, J.-G.; and Lee, B. S. 2023. Context Consistency Regularization for Label Sparsity in Time Series. In *Proceedings of International Conference on Machine Learning (ICML)*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1): 1929–1958.

Trirat, P.; Shin, Y.; Kang, J.; Nam, Y.; Na, J.; Bae, M.; Kim, J.; Kim, B.; and Lee, J.-G. 2024. Universal time-series representation learning: A survey. *arXiv preprint arXiv:2401.03717*.

Wang, Y.; Wu, H.; Dong, J.; Liu, Y.; Qiu, Y.; Zhang, H.; Wang, J.; and Long, M. 2024. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *Proceedings of International Conference on Machine Learning (ICML)*.

Wu, H.; Wu, J.; Xu, J.; Wang, J.; and Long, M. 2022. Flowformer: Linearizing Transformers with Conservation Flows. In *Proceedings of International Conference on Machine Learning (ICML)*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 22419–22430.

Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling Time Series with 10k Parameters. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big Bird: Transformers for Longer Sequences. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*, 17283–17297.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of International Conference on Machine Learning (ICML)*, 27268–27286.