CASA: CNN Autoencoder-based Score Attention for Efficient Multivariate Long-term Time-series Forecasting

Minhyuk Lee[1] , **HyeKyung Yoon**[3] and **MyungJoo Kang**[1,2,3 †]
[1]Department of Mathematical Sciences of Seoul National University
[2]Research Institute of Mathematics of Seoul National University
[3]Interdisciplinary Program in Artificial Intelligence of Seoul National University
{356min, yhk04150, mkang}@snu.ac.kr,

Abstract

Multivariate long-term time series forecasting is critical for applications such as weather prediction, and traffic analysis. In addition, the implementation of Transformer variants has improved prediction accuracy. Following these variants, different input data process approaches also enhanced the field, such as tokenization techniques including point-wise, channel-wise, and patch-wise tokenization. However, previous studies still have limitations in time complexity, computational resources, and cross-dimensional interactions. To address these limitations, we introduce a novel CNN Autoencoder-based Score Attention mechanism (CASA), which can be introduced in diverse Transformers model-agnosticically by reducing memory and leading to improvement in model performance. Experiments on eight real-world datasets validate that CASA decreases computational resources by up to 77.7%, accelerates inference by 44.0%, and achieves state-of-the-art performance, ranking first in 87.5% of evaluated metrics. Our code is available at https://github.com/lmh9507/CASA.

1 Introduction

Multivariate Long-Term Time Series Forecasting (LTSF) plays a pivotal role in real-world applications, including weather prediction, traffic flow analysis [Ji *et al.*, 2023], and solar energy forecasting [Lai *et al.*, 2018]. LTSF has seen rapid advancements driven by the emergence of Transformer model [Vaswani, 2017]. Subsequent Transformer variants have further demonstrated the effectiveness of the multi-head self-attention mechanism in capturing temporal dependencies and cross-dimensional correlations. [Zhou *et al.*, 2021; Wu *et al.*, 2021; Liu *et al.*, 2022b; Zhou *et al.*, 2022; Liu *et al.*, 2022c; Zhang and Yan, 2023; Nie *et al.*, 2022].

Despite numerous efforts, Transformer-based models have not consistently outperformed CNN- or MLP-based architectures in the LTSF domain [Wu *et al.*, 2023; Ekambaram *et al.*, 2023; Das *et al.*, 2023; Zeng *et al.*, 2023]. Notably, DLinear [Zeng *et al.*, 2023], a model constructed with simple linear layers, raises critical questions about the effectiveness and necessity of Transformer-based architectures in this field, especially considering their demanding computational and time resources. To address the aforementioned challenges, diverse tokenization techniques have been introduced into the core architecture of Transformer-family models [Liu *et al.*, 2023; Nie *et al.*, 2022]. We evaluate the effectiveness of three models, each employing a distinct tokenization technique, as illustrated in Figure 2. Figure 1 demonstrates that channel-wise tokenization achieves the best performance among the three, as highlighted in [Yu *et al.*, 2024]. However, it still leads to rapid saturation during training. Moreover, computational cost and memory usage remain significant challenges. Our objective is to develop a more efficient model that delivers superior predictive performance while mitigating these drawbacks. **We tackle these issues by refining the self-attention mechanism, the cornerstone of Transformer-based models.**

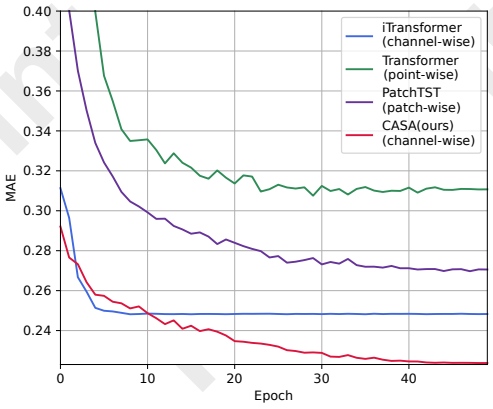

Figure 1: The validation loss of iTransformer, Transformer, PatchTST, and our model on the Traffic dataset is evaluated. Point-wise and patch-wise implemented models exhibit lower performance compared to channel-wise models. However, while the iTransformer model rapidly saturates, CASA demonstrates consistent learning and achieves the lowest loss value.

In this paper, we propose **CNN Autoencoder-based Score**

† Corresponding author.

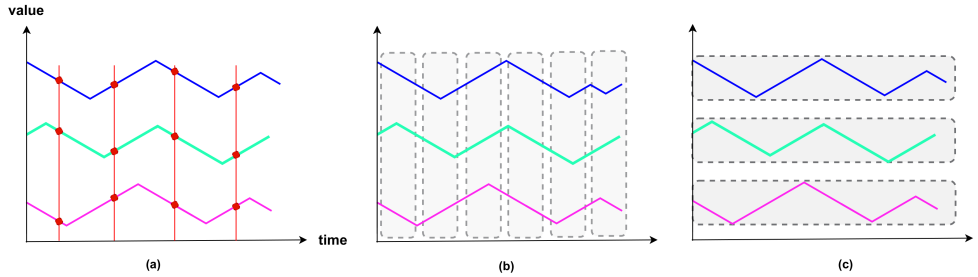

Figure 2: (a) point-wise token (b) patch-wise token (c) channel-wise token

Attention (CASA), a simple yet novel module that addresses the aforementioned gap by effectively capturing correlations and avoiding saturation, thereby facilitating consistent learning, designed to serve as an alternative to the conventional self-attention mechanism. We retain the vanilla Transformer encoder while substituting the attention mechanism with a CNN-based module. CASA approximates $\frac{QK^T}{\sqrt{d_k}}$ rather than directly calculating it, as done in traditional multi-head self-attention. This design addresses a critical limitation of conventional methods by sufficiently accounting for significant correlation between variates in the calculation of attention scores (see Section 3.3 for details).

Our key contributions are as follows:

- **We present a simple yet effective CNN Autoencoder-based Score Attention (CASA) module** as an alternative to self-attention. It scales linearly with the number of variates, input length, and prediction length. Compared to Transformer-based variants, CASA reduces memory usage by up to **77.7%** and improves computational speed by **44.0%**, depending on the dataset.
- **CASA can be agnostically integrated into Transformer models, regardless of the tokenization technique used** (e.g., point-wise, patch-wise, or channel-wise). To the best of our knowledge, this is the first module validated across individual tokenization techniques, successfully enhancing cross-dimensional information capture and improving prediction performance.
- CASA achieved **first place in 54 out of 64 metrics** across 8 real-world datasets and ranked highest in **14 out of 16 average metrics**, establishing itself as a highly competitive solution for multivariate LTSF.

2 Related Works

Transformer variants The vanilla Transformer model [Vaswani, 2017], widely recognized for its success in natural language processing, has also achieved notable advancements in time-series forecasting. Diverse Transformer variants have been introduced to enhance forecasting performance, which can be broadly grouped into three approaches. The first approach modifies the traditional self-attention mechanism with alternatives by incorporating specialized modules, or pyramidal attention [Liu *et al.*, 2022b], to reduce memory requirements while capturing multi-resolution representations. Additional modifications, including the trapezoidal architecture [Zhou *et al.*, 2021] and de-stationary attention [Liu *et al.*, 2022c], aim to improve robustness and address issues like over-stationarization. The second approach leverages frequency-domain techniques, such as Fast Fourier Transform (FFT) [Zhou *et al.*, 2022] and auto-correlation mechanisms [Wu *et al.*, 2021], to better extract temporal patterns. The third approach introduces hierarchical encoder-decoder frameworks [Zhang and Yan, 2023] with routing mechanisms to capture cross-dimensional information, although these methods sometimes encounter challenges such as slower learning and higher computational demands.

Alternatives of Transformers While Transformer variants have significantly advanced the time-series forecasting domain, CNN-based models present promising alternatives. These approaches include methods that model segmented signal interactions [Liu *et al.*, 2022a] and those that reshape 1D time-series data into 2D tensors [Wu *et al.*, 2023], enabling the capture of both inter-period and intra-period dynamics. Similarly, linear models [Zeng *et al.*, 2023] have demonstrated simplicity while achieving high prediction performance. However, these methods generally fall short of explicitly addressing cross-dimensional interactions, which are crucial for improving multivariate time-series forecasting. Other methods have been developed to modify aspects of the Transformer architecture, particularly focusing on tokenization techniques. For instance, PatchTST [Nie *et al.*, 2022] segments input data into patches to extract local information within individual variates, while iTransformer [Liu *et al.*, 2023] treats each variate as a token, enabling the self-attention mechanism to capture multivariate correlations. However, a common drawback of these methods is their reliance on self-attention, which demands substantial computational resources. Furthermore, their prediction performance remains suboptimal, with both models struggling to effectively capture multivariate dependencies, which critically impacts their predictive accuracy. In contrast, our proposed model, CASA, addresses these challenges by significantly reducing resource consumption while achieving superior prediction performance, offering an efficient and effective alternative to traditional self-attention-based approaches.

3 Method

This section provides an overview of CASA, including the motivation behind its architecture, a detailed explanation of

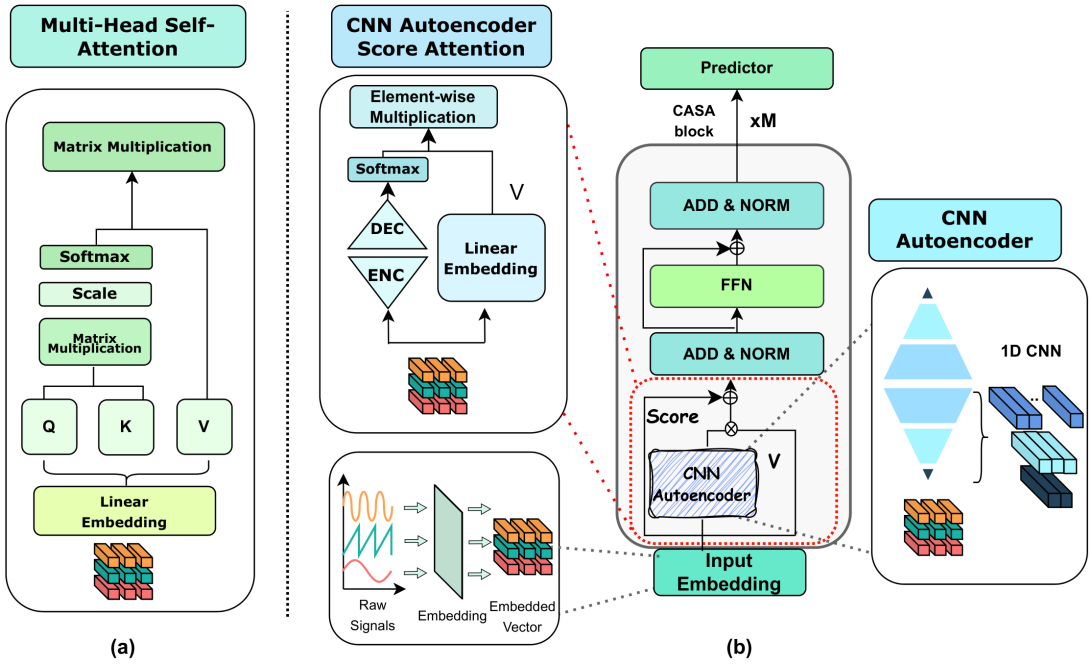

Figure 3: (a) Conventional Self-Attention. (b) Overall architecture of our CASA block. The time-series data is embedded using channel-wise tokenization. The 1D CNN Autoencoder is then used to compute cross-dimensional information. The softmax output and the value are multiplied element-wise. Our CASA places a strong emphasis on capturing essential cross-dimensional information by calculating high-dimensional spatial relationships before compressing the channel information.

its structure, and a complexity analysis. Section 3.1, we formulate the problem we aim to solve and define the necessary notations. Section 3.2 provides an explanation of the overall architecture. Section 3.3 discusses the limitations of conventional Transformers in capturing cross-dimensional interactions. Section 3.4 introduces CNN Autoencoder-based Score Attention (CASA), an improved self-attention module designed to address these issues. We confirm theoretical efficiency of CASA through complexity analysis.

3.1 Problem Formulation

In multivariate LTSF, given an input length L, the number of variates N, the number of layers M, and a prediction length H, denote the input and the output $X \in \mathbb{R}^{N \times L}$ and $Y \in \mathbb{R}^{N \times H}$ respectively. The hidden dimension is denoted as D, and the intermediate feature after embedding is represented as $Z_i \in \mathbb{R}^{N \times D}$ $(i \in \{0, 1, \cdots M\})$. Since this is not a univariate time-series forecasting problem, the input is consistently represented as a matrix.

3.2 Overall Architecture

The overall framework is depicted in Figure 3. Following prior works [Liu *et al.*, 2023; Yu *et al.*, 2024], we adopt a channel-wise tokenization approach, utilizing the vanilla Transformer encoder as the backbone. To address the challenges outlined in Section 3.3, we enhance the Transformer by replacing only the attention mechanism with a 1D **CNN Autoencoder Score Attention (CASA)**. Although this modification alters only a very small portion of the overall model, it effectively reduces computational cost and memory usage while improving prediction performance.

The input X is linearly embedded to produce the intermediate feature Z_0. The final feature Z_M, obtained by passing Z_0 through M CASA blocks, is then fed into the predictor where the output becomes Y. The pipeline is summarized by the following equation.

$$Z_0 = \text{Embedding}(X) \tag{1}$$
$$Z_{i+1} = \text{CASA block}(Z_i) \tag{2}$$
$$Y = \text{Predictor}(Z_M) \tag{3}$$

3.3 Limitation of Self-Attention

We demonstrate that the existing self-attention mechanism does not sufficiently consider cross-dimension information when embedding queries and keys. the structure of the self-attention mechanism in the conventional Transformer is as follows (f_j: affine map):

$$\text{Attention}(Z_{i+1}) = \text{softmax}\left(\frac{Q_{i+1}K_{i+1}^T}{\sqrt{d_k}}\right) * V_{i+1} \tag{4}$$

$$Q_{i+1} = f_1(Z_i), \;\; K_{i+1} = f_2(Z_i), \;\; V_{i+1} = f_3(Z_i) \tag{5}$$

At this point, since f_j is an affine map, queries and keys are computed through the following operations:

$$Q_{i+1} = Z_{i,1}W_{i,1} + b_{i,1}, \quad K_{i+1} = Z_{i,2}W_{i,2} + b_{i,2} \tag{6}$$

$$\text{where } W_{i,j} \in \mathbb{R}^{D \times D} \text{ and } b_{i,j} \in \mathbb{R}^{N \times D} \tag{7}$$

	CASA	SOFTS	iTransformer	PatchTST	Transformer
Complexity	$O(NL+NH)$	$O(NL+NH)$	$O(N^2+NL+NH)$	$O(NL^2+NH)$	$O(NL+L^2+HL+NH)$
Memory(MB)	1684	1720	10360	21346	4772
Inference(s/iter)	0.05	0.042	0.162	0.441	0.087

Table 1: A complexity comparison conducted on the Traffic dataset among baseline models, including the vanilla Transformer, PatchTST, iTransformer, and SOFTS, with respect to window length L, number of channels N, and forecasting horizon H. Notably, the complexity of CASA scales linearly with N, L, and H. Detailed implementation information is provided in Appendix D.

Proposition 1. *Query and key embeddings are variate-independent operations in the conventional Transformer using channel-wise tokenization.*

Proof. See Appendix E.1. □

Proposition 2. *Query and key embeddings are time-independent operations in the conventional Transformer using point-wise and patch-wise tokenization.*

Proof. See Appendix E.2. □

Proposition 1 and Proposition 2 imply that each tokenization method does not consider the correlation between variates or time points when embedding the key and query. Since multivariate time series exhibit correlations both between variates and across time points, this reduces the potential of the Transformer architecture. Especially, based on the Proposition 1, the Transformer using channel-wise tokenization does not directly incorporate cross-dimensional information when embedding the r-th variate into queries and keys. In other words, the self-attention mechanism embeds queries and keys through variate-independent feature refinement operations and then computes the attention map using $\frac{Q_i K_i^T}{\sqrt{d_k}}$. **In the LTSF domain, tokens (i.e., variates) exhibit inherent correlations** (see Section 4.4), **which reduce the effectiveness of variate-independent operations during feature refinement.** This limitation can hinder performance in the multivariate LTSF domain, where capturing correlations between variates is important.

3.4 CNN Autoencoder-based Score Attention

To address the issue posed by the self-attention mechanism's variate-independent operation, we treat each variate as a channel and apply a convolution instead of using the affine map from the conventional self-attention mechanism. This approach ensures that the operation becomes variate-dependent. In more detail, instead of directly computing $\frac{Q_i K_i^T}{\sqrt{d_k}}$, we designed a score network `Score` to approximate this operation using 1D CNN Autoencoder. The modified structure of self-attention is as follows (f: affine map, $\circledast$: element-wise product):

$$\text{Attention}(Z_{i+1}) = \text{softmax}(\texttt{Score}(Z_i)) \circledast V_{i+1} \quad (8)$$

$$V_{i+1} = f(Z_i) \quad (9)$$

By approximating $\frac{Q_i K_i^T}{\sqrt{d_k}}$ via a CNN architecture instead of direct computation, we reduce complexity compared to the affine map (see the paragraph below), addressing the limitations of self-attention. This facilitates the development of a linear complexity model with enhanced performance (Section 4.1). To explain the score network in more detail, we adopted an inverted bottleneck autoencoder structure, inspired by previous research [Wilson *et al.*, 2016; Bengio *et al.*, 2013], which demonstrated that embedding low-dimensional features into a high-dimensional latent space can improve expressiveness. In summary, we leverage CNN operations to incorporate information across all variates, embedding them into a high-dimensional feature space before compressing the channels to retain only essential cross-variable information. Consequently, despite its simple architecture, the proposed module outperforms existing self-attention mechanisms while maintaining efficiency, constituting a significant contribution.

Complexity Analysis CASA is an efficient algorithm that exhibits **linear complexity** not only with respect to the number of tokens, i.e., the number of variates N, but also with respect to the input length L and prediction length H. The detailed complexity calculation is as follows. Let the kernel size of the score network be denoted as k. The complexities of the Reversible Instance Normalization (RevIN), series embedding, and MLP are $O(NL)$, $O(NLD)$, and $O(ND^2)$, respectively. Additionally, the complexity of the score network, composed of CNN autoencoder blocks, is $O(NkD^2)$. The predictor has a complexity of $O(NDH)$. Thus, the overall complexity of our method is $O(NL + NLD + ND^2 + NkD^2 + NDH)$, which scales linearly with respect to N, L, and H. Since the hidden dimension and kernel size are constants in the algorithm, they can be ignored. Consequently, N is dominated by NL and NH(Since L and H typically take on large values in LTSF), leading to the final complexity summarized in Table 1. In addition, the results for memory usage and inference time are included in the table, empirically demonstrating the efficiency of CASA. For details of the implementation, refer to the Appendix D.

4 Experiments

Dataset We conduct our comprehensive experiment on 8 benchmark datasets [Zhou *et al.*, 2021], such as Traffic, ETT series including 4 subsets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Solar, Electricity. More detailed information on Dataset is described in Appendix A.

Baselines We chose totally 8 comtemporarlly baseline models, including SOFTS [Han *et al.*, 2024], iTransformer [Liu *et al.*, 2023], PatchTST [Nie *et al.*, 2022] , TSMixer

Dataset	**CASA**(ours)	**SOFTS** Han *et al.*, 2024	**iTransformer** Liu *et al.*, 2023	**PatchTST** Nie *et al.*, 2022	**TSMixer** Ekambaram *et al.*, 2023	**Crossformer** Zhang and Yan, 2023	**TiDE** Das *et al.*, 2023	**DLinear** Zeng *et al.*, 2023	**FEDformer** Zhou *et al.*, 2022
	MSE(↓) / MAE(↓)	MSE / MAE	MSE / MAE	MSE / MAE	MSE / MAE	MSE / MAE	MSE / MAE	MSE / MAE	MSE / MAE
ETTm1	**0.386** / **0.393**	0.393 / 0.403	0.407 / 0.410	0.396 / 0.406	0.398 / 0.407	0.513 / 0.496	0.419 / 0.419	0.474 / 0.453	0.543 / 0.490
ETTm2	**0.276** / **0.319**	0.287 / 0.330	0.288 / 0.332	0.287 / 0.330	0.289 / 0.333	0.757 / 0.610	0.358 / 0.404	0.350 / 0.401	0.305 / 0.349
ETTh1	**0.438** / **0.434**	0.449 / 0.442	0.454 / 0.447	0.453 / 0.446	0.463 / 0.452	0.529 / 0.522	0.541 / 0.507	0.456 / 0.452	0.440 / 0.460
ETTh2	0.374 / **0.397**	**0.373** / 0.400	0.383 / 0.407	0.385 / 0.410	0.401 / 0.417	0.942 / 0.684	0.611 / 0.550	0.559 / 0.515	0.437 / 0.449
ECL	**0.168** / **0.259**	0.174 / 0.264	0.178 / 0.270	0.189 / 0.276	0.186 / 0.287	0.244 / 0.334	0.251 / 0.344	0.212 / 0.300	0.214 / 0.327
Traffic	0.421 / **0.261**	**0.409** / 0.267	0.428 / 0.282	0.454 / 0.286	0.522 / 0.357	0.550 / 0.304	0.760 / 0.473	0.625 / 0.383	0.610 / 0.376
Weather	**0.243** / **0.267**	0.255 / 0.278	0.258 / 0.278	0.256 / 0.279	0.256 / 0.279	0.259 / 0.315	0.271 / 0.320	0.265 / 0.317	0.309 / 0.360
Solar	**0.221** / **0.244**	0.229 / 0.256	0.233 / 0.262	0.236 / 0.266	0.260 / 0.297	0.641 / 0.639	0.347 / 0.417	0.330 / 0.401	0.291 / 0.381
$1^{st}/2^{nd}$ count	14 / 2	2 / 13	0 / 1	0 / 2	0 / 0	0 / 0	0 / 0	0 / 0	0 / 1

Table 2: Multivariate forecasting results with horizon $H \in \{96, 192, 336, 720\}$ and fixed lookback window length $L = 96$. Red values represent the best performance, while underlined values represent the second-best performance. Results are averaged from all prediction horizons. Full results are listed in Table 6. (Appendix B)

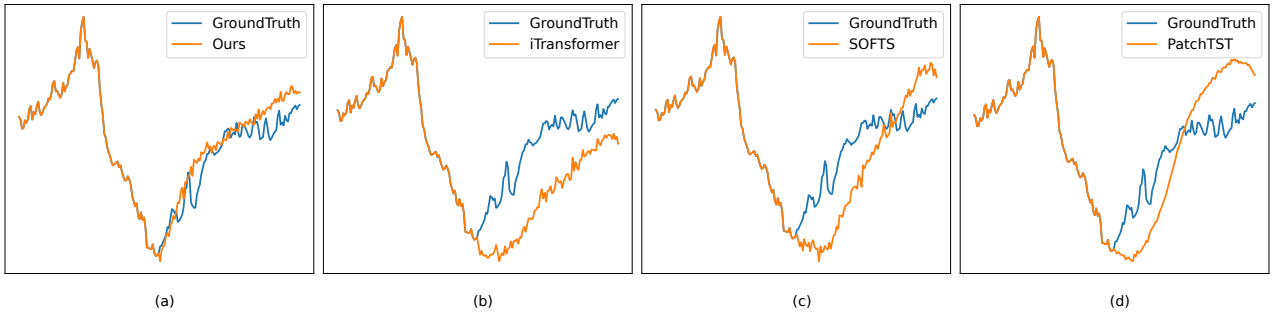

Figure 4: Prediction results for our model and baseline models on the Weather dataset, with sequence lengths L and H set to 96. (a) CASA, (b) iTransformer, (c) SOFTS, (d) PatchTST.

[Ekambaram *et al.*, 2023], Crossformer [Zhang and Yan, 2023], TiDE [Das *et al.*, 2023] , DLinear [Zeng *et al.*, 2023], FEDformer [Zhou *et al.*, 2022].

Setup Our comprehensive experiments results are based on MSE (Mean Squared Error) and MAE (Mean Absolute Error) metrics. Our main experiments are conducted on the conditions with $L = 96$ and the $H \in \{96, 192, 336, 720\}$.

4.1 Multivariate Forecasting Results

The main results are presented in Table 2, where red-bold text indicates the best score and blue-underlined text represents the second-best score. CASA demonstrates the lowest MSE and MAE losses across 8 benchmark datasets, surpassing the previous state-of-the-art model, SOFTS, by a substantial margin. Additionally, the second-lowest performance scores exhibit a smaller gap from the best score compared to the others. Notably, the proposed model showcases its robustness on relatively large datasets, such as Traffic, Weather, and Solar, highlighting its ability to capture complex correlations, which significantly enhances the model's predictive performance.

Figure 4 visualizes the prediction performance of weather dataset from CASA, SOFTS, iTransformer, and PatchTST models against the ground truth. CASA shows the closest alignment with the label, while iTransformer shows similar tendency along the ground truth with large deviation. SOFTS and PatchTST prediction slightly detours the label. The full results of different prediction lengths and the visualization results on the rest of the datasets are demonstrated in Appendix B and Appendix C, respectively.

4.2 Superiority Analysis of CASA

Replacing Self-Attention with CASA To ensure the proposed model's adaptability to diverse tokenization techniques, we integrate CASA into various Transformer variants including the vanilla Transformer, PatchTST, and iTransformer. Especially for the vanilla Transformer, we only use the encoder architecture to appropriately compare the effectiveness of CASA.

The results on seven benchmark datasets are presented in Table 3. With the exception of two specific cases—ETTm2 compared to PatchTST and ETTh2 compared to the vanilla Transformer—CASA consistently enhances the performance of the original and variants models, achieving improvements in 40 out of 42 results across the benchmarks. These findings not only demonstrate that replacing self-attention with CASA significantly boosts forecasting accuracy, but also highlight its flexibility and adaptability, as it can seamlessly integrate with diverse tokenization techniques, making it a versatile enhancement for Transformer-based architectures.

Model	Comp	ECL		Traffic		Weather		ETTm1		ETTm2		ETTh1		ETTh2	
		MSE(↓)	MAE(↓)	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Transformer	Attention	0.203	0.292	0.655	0.359	0.245	0.296	0.407	0.417	0.369	0.398	0.482	0.465	0.522	**0.481**
	CASA	**0.201**	**0.287**	**0.645**	**0.354**	**0.242**	**0.293**	**0.390**	**0.411**	**0.368**	**0.388**	**0.465**	**0.461**	**0.512**	0.490
PatchTST	Attention	0.189	0.276	0.454	0.286	0.256	0.279	0.396	0.406	0.287	**0.330**	0.453	0.446	0.385	0.410
	CASA	**0.186**	**0.273**	**0.440**	**0.280**	**0.253**	**0.277**	**0.386**	**0.402**	**0.285**	0.332	**0.452**	**0.443**	**0.365**	**0.399**
iTransformer	Attention	0.178	0.270	0.428	0.282	0.258	0.278	0.407	0.410	0.288	0.332	0.454	0.447	0.383	0.407
	CASA	**0.168**	**0.259**	**0.421**	**0.261**	**0.244**	**0.267**	**0.386**	**0.393**	**0.276**	**0.319**	**0.438**	**0.434**	**0.374**	**0.397**

Table 3: The performance of CASA across three distinct Transformer-based models, each employing different tokenization techniques. The standard self-attention module is replaced with our CASA. Among the 42 metrics assessed, CASA demonstrated improvements in 40 of them.

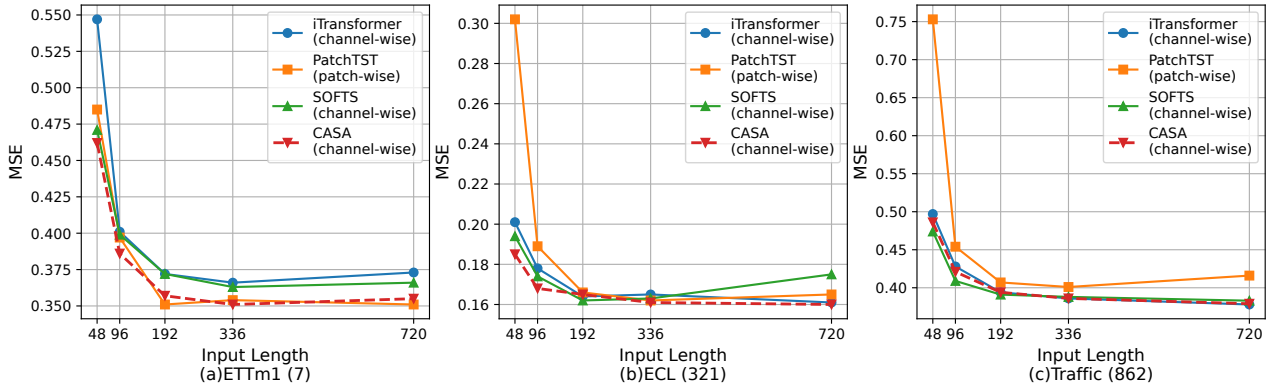

Figure 5: Experimental results on the ETTm1, Electricity, and Traffic datasets (with 7, 321, and 862 variates, respectively). Our CASA remains robust across varying input and prediction lengths (48 to 720). Unlike PatchTST, which struggles as the number of variates increases, models like iTransformer and SOFTS, which tokenize variates, exhibit stronger performance.

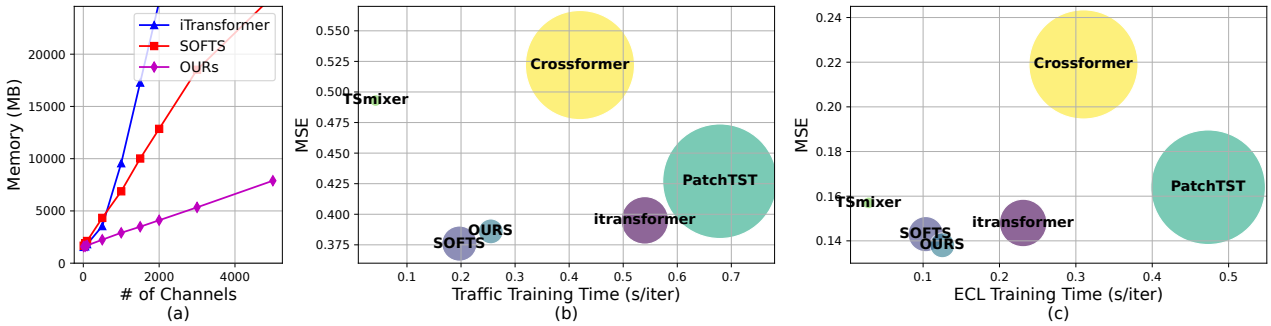

Figure 6: (a) Memory usage scaling with the number of tokens, demonstrating CASA's linear growth and reduced memory consumption compared to SOFTS. (b, c) Experimental results on Traffic and Electricity datasets (batch size: 16, input/prediction length: 96), highlighting CASA's low memory usage and balanced trade-off between speed and performance.

Robustness of CASA under varying conditions To validate the robustness of CASA with respect to input length and the number of variates, we conducted experiments on the ETTm1, Electricity, and Traffic datasets, which contain 7, 321, and 862 variates, respectively, with prediction lengths ranging from 48 to 720. As shown in Figure 5, the performance of models utilizing non-channel-wise tokenization declines as the number of variates increases. Specifically, PatchTST experiences a significant performance drop on the Traffic dataset, recording the highest MSE losses. In contrast, models such as iTransformer and SOFTS, which employ channel-wise tokenization, demonstrate greater resilience to increases in the number of variates. However, both models exhibit elevated MSE losses on the ETTm1 dataset, while their performance improves on the Electricity and Traffic datasets. In comparison, our proposed model maintains stable MSE losses across all three datasets, achieving notably lower MSE losses on the ETTm1 dataset. This under-

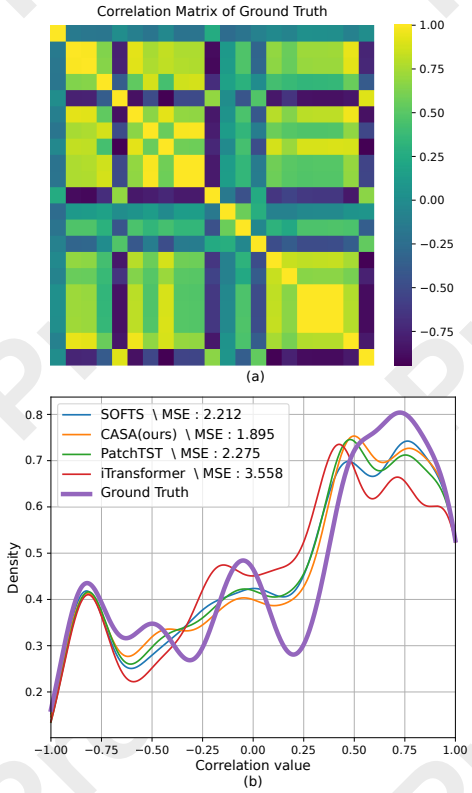

Figure 7: (a) Correlation matrix among the Ground Truth variates, computed on the Weather Dataset. (b) PDFs of the Ground Truth and each model, computed using KDE.

Model	MSE(↓)	Cosine Similarity(↑)	SSIM(↑)
CASA(ours)	**1.1204**	**0.9965**	**0.9912**
SOFTS	<u>1.2520</u>	0.9957	0.9889
iTransformer	1.8675	0.9910	0.9791
PatchTST	1.2393	<u>0.9960</u>	<u>0.9896</u>

Table 4: Metrics for the correlation matrices of the Ground Truth and each model. Boldface indicates the best performance, and underlining indicates the second-best performance.

scores CASA's ability to deliver high predictive accuracy under varying conditions, ensuring consistent performance even as the input data length increases.

4.3 Model Efficiency Analysis

In this section, we empirically validate the efficiency of CASA, as theoretically outlined in Section 3.4. For comparison, we use iTransformer and SOFTS as baselines. Figure 6 (a) depicts memory usage, revealing that CASA exhibits linear complexity and effectively leverages practical computational resources. This performance is comparable to SOFTS, which also demonstrates similar complexity but increases memory usage significantly. Notably, CASA significantly outperforms iTransformer, which suffers from quadratic memory growth. Figures 6 (b) and (c) present memory footprints, inference time, and MSE for the Traffic and Electricity datasets, using a batch size of 16 and input/inference sequence lengths of 96. CASA consumes fewer computational resources than Transformer variants such as iTransformer, Crossformer, and PatchTST. Regarding MSE, CASA achieves the second-lowest value on Traffic and the lowest on Electricity, all while maintaining fast inference and efficient resource usage. Although CASA slightly exceeds TSMixer in memory usage, it delivers stronger overall performance, striking an effective balance between accuracy and resource efficiency.

4.4 Investigating Cross-Dimensional Interactions: A Correlation Matrix Analysis of CASA

We evaluate CASA's capacity to capture cross-dimensional interactions by examining the correlation matrices derived from each model's predictions (i.e., correlations among all variates) on the Weather Dataset, comparing outcomes from CASA, SOFTS, iTransformer, and PatchTST against the Ground Truth. The Ground Truth correlation matrix shows distinct positive and negative correlation blocks, indicating a clear spatial structure with a grid-like arrangement (Figure 7 (a)). Furthermore, kernel density estimation (KDE) using a Gaussian kernel reveals that correlation values are mostly concentrated in the positive domain (Figure 7 (b)), suggesting many variates rise or fall in sync. Notably, CASA's correlation matrix most closely approximates the Ground Truth, as evidenced by the lowest MSE loss between their probability density functions (PDFs) (Figure 7 (b)). This underscores CASA's ability to preserve intricate relationships essential for capturing temporal and spatial dependencies in weather data. For a more comprehensive evaluation, we compare each model's correlation matrix to the Ground Truth using Mean Squared Error (MSE), cosine similarity, and the Structural Similarity Index Measure (SSIM). CASA outperforms all other models across these metrics, validating its effectiveness in capturing cross-dimensional interactions (Table 4).

5 Conclusion

In this study, we introduce the CASA model, which demonstrates remarkable effectiveness and solid predictive performance, as confirmed by a broad range of experiments. By delivering state-of-the-art results while requiring notably fewer computational resources and less processing time than existing approaches. Moreover, its CNN-based autoencoder module successfully captures cross-dimensional interactions throughout the compress-and-decompress procedure, which contributes to its outstanding performance. Additionally, CASA shows strong potential as a versatile alternative to conventional attention mechanisms in various Transformer configurations, remaining unaffected by different tokenization methods. This further highlights its adaptability, practicality, and efficiency across diverse use cases.

Contribution Statement

Minhyuk Lee and HyeKyung Yoon contributed equally to this manuscript and are recognized as joint first authors.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)] and the National Research Foundation of Korea (NRF) [RS-2024-00421203, RS-2024-00406127, RS-2021-NR059802]

References

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[Das *et al.*, 2023] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

[Ekambaram *et al.*, 2023] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant 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*, pages 459–469, 2023.

[Han *et al.*, 2024] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *In the twelfth international conference on learning representations*, 2024.

[Ji *et al.*, 2023] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4356–4364, 2023.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[Liu *et al.*, 2022a] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 2022.

[Liu *et al.*, 2022b] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. 2022.

[Liu *et al.*, 2022c] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[Liu *et al.*, 2023] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *the eleventh international conference on learning representations*, 2023.

[Nie *et al.*, 2022] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *In the tenth international conference on learning representations*, 2022.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wilson *et al.*, 2016] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

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

[Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *In the eleventh international conference on learning representations*, 2023.

[Yu *et al.*, 2024] Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. *International conference on machine learning*, 2024.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[Zhang and Yan, 2023] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *the eleventh international conference on learning representations*, 2023.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.