Integrating Transfer Entropy into Transformer for Time Series Forecasting

YongKyung Oh, Alex A. T. Bui*

Medical & Imaging Informatics (MII) Group, Department of Radiological Sciences,
University of California, Los Angeles (UCLA),
Los Angeles, CA 90024 USA
yongkyungoh@mednet.ucla.edu, buia@mii.ucla.edu

Abstract

Time series forecasting is an important task in a variety of domains. Recently, transformers have shown promise by effectively modeling long-range dependencies through self-attention mechanisms. However, they inherently assume symmetric relationships and lack the ability to explicitly model the directionality of information flow—a critical aspect in time series data where causality plays a significant role. This paper addresses this research gap by proposing a novel transformer architecture that integrates transfer entropy into the attention mechanism to explicitly model directional dependencies and causal relationships in time series forecasting. Empirical validation on the M4 benchmark dataset, a comprehensive collection of time series data for forecasting, shows that our model outperforms state-of-the-art transformer-based models, achieving superior forecasting accuracy. Our method represents a remarkable advancement in time series forecasting by uniquely combining the benefits of transformers with the ability to model causal relationships without requiring additional causal graphs or prior knowledge about the data.

Introduction

Time series forecasting is a critical task across various domains, where accurate predictions are essential for strategic planning and decision-making (Box et al. 2015; Hyndman 2018). The advent of deep learning has led to significant advancements in modeling time series data. Convolutional Neural Network (CNN) (LeCun, Bengio, and Hinton 2015), Recurrent Neural Network (RNN) (Rumelhart, Hinton, and Williams 1986; Medsker and Jain 1999), Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber 1997), and Gated Recurrent Unit (GRU) (Chung et al. 2014) have shown promise in capturing temporal dependencies and nonlinear patterns in sequential data. These models, however, face challenges with long-range dependencies due to issues like limited receptive fields in CNNs and vanishing gradients in RNNs, which can hinder their ability to model complex temporal dynamics (Bengio, Simard, and Frasconi 1994; Yu and Koltun 2016; Yu et al. 2019).

Transformer, first proposed by Vaswani (2017), marked a breakthrough in sequence modeling through its innovative self-attention mechanism, which can model relationships between distant elements directly, eliminating the need for recurrent or convolutional operations. Transformers have been successfully adapted for time series forecasting, demonstrating superior performance due to their ability to model complex temporal patterns and handle long sequences efficiently (Lim et al. 2021; Zhou et al. 2021). Despite their success, standard transformer architectures inherently assume symmetric relationships and do not explicitly model the directionality of information flow, which is crucial in time series data where past events causally influence future outcomes (Oord 2016; Lai et al. 2018).

Existing transformer-based models for time series forecasting focus primarily on capturing dependencies between time steps but overlook the causal and directional relationships inherent in temporal data (Li et al. 2019; Wu et al. 2023; Zeng et al. 2023). The self-attention mechanism in transformers assigns attention weights based on similarity measures without considering the causal influence of past states on current or future states. This limitation restricts the model's ability to fully capture underlying causal dynamics, potentially leading to suboptimal forecasting performance, especially in complex multivariate time series where interactions between variables are directional and time-lagged.

To bridge this gap, we propose a novel transformer architecture that integrates transfer entropy into the attention mechanism, enabling explicit modeling of directional dependencies and causal relationships in time series forecasting. Transfer entropy is an information-theoretic measure that quantifies the directed transfer of information between stochastic processes, effectively capturing the causal influence of one variable on another over time (Schreiber 2000; Vicente et al. 2011; Oh, Kwak, and Kim 2023). By incorporating a neural estimator of transfer entropy into the attention logits, our model biases the attention mechanism towards causally relevant time steps, effectively capturing the directionality of information flow.

Our key contributions are as follows:

- We introduce a neural transfer entropy estimator within the transformer framework, modifying the attention mechanism to account for directed information flow between time steps. This integration allows the model to capture causal influences inherently, without requiring additional causal graphs or prior knowledge.

*Corresponding author

- We demonstrate that integrating transfer entropy into the attention mechanism enhances the model's capacity to represent causal relationships and temporal dependencies, leading to improved forecasting performance.

This advancement opens new avenues for incorporating causal inference principles into deep learning models for time series analysis. It addresses a significant gap in the literature by providing a framework that combines the strengths of transformers with the ability to model causal relationships, which is essential for understanding and predicting complex temporal dynamics.

Related works

Conventional time series forecasting methods

Time series forecasting has traditionally relied on statistical models such as Autoregressive Integrated Moving Average (ARIMA) (Box et al. 2015), Exponential Smoothing Methods (Gardner Jr 1985), and Seasonal Decomposition (Cleveland et al. 1990). These models are well-understood and interpretable but often struggle with complex patterns, nonlinear relationships, and high-dimensional data inherent in modern time series applications (Hyndman 2018; Oh, Lim, and Kim 2024). Their limitations become pronounced when dealing with large-scale datasets or when capturing intricate temporal dependencies is crucial for forecasting accuracy.

The emergence of deep learning methods has marked a breakthrough in time series forecasting, offering unprecedented ability to model complex relationships and long-range interactions in temporal data. Recurrent neural networks have been widely used due to their ability to model temporal sequences and handle vanishing gradient problems to some extent. They have been applied successfully in various forecasting tasks, including speech recognition and language modeling (Graves 2013). However, such methods face challenges with long-range dependencies and parallelization limitations. Transformers (Vaswani 2017), initially designed for natural language processing tasks, have gained attention in time series forecasting due to their ability to model long-range dependencies without relying on recurrence. The self-attention mechanism allows transformers to weigh the relevance of different time steps, capturing both short-term and long-term patterns effectively. While several models have advanced the field, they primarily focus on improving efficiency and handling long sequences. They do not explicitly model the directionality of information flow or causal relationships inherent in time series data.

Transfer entropy and causality in time series

Transfer entropy, introduced by Schreiber (2000), is an information-theoretic measure that quantifies the directed flow of information between two stochastic processes. It is particularly useful for detecting causal relationships and directional dependencies in time series data without assuming linearity or specific model structures. Transfer entropy has been applied in various domains such as finance (Marschinski and Kantz 2002), neuroscience (Vicente et al. 2011), climate science (Runge et al. 2012), and traffic network (Oh, Kwak, and Kim 2023).

Integrating transfer entropy into deep learning models for time series analysis is a relatively unexplored area. The primary challenge lies in estimating transfer entropy in a manner compatible with neural network training and scalable to large datasets. Some efforts have been made to incorporate causality and information-theoretic measures into neural networks: Nauta, Bucur, and Seifert (2019) introduced a framework for causal discovery using recurrent neural networks, aiming to learn causal structures from time series data. Tank et al. (2021) proposed a method to detect nonlinear Granger causality using neural networks, extending traditional Granger causality concepts to capture more complex relationships. However, these approaches do not integrate transfer entropy directly into the model architecture, particularly within the attention mechanism of transformers. They focus on detecting causal relationships rather than leveraging them to improve forecasting performance.

To the best of our knowledge, no prior work has directly embedded transfer entropy estimation within the attention mechanism of transformer models for time series forecasting. This gap presents an opportunity to enhance model performance by explicitly capturing directional dependencies and causal influences.

Methodology

Transfer Entropy

Transfer entropy quantifies the amount of directed information transfer from a source process X to a target process Y. Formally, the transfer entropy $T_{X\to Y}$ is defined as:

$$T_{X\to Y} = \sum_t p(y_t, y_{t-1}^{(k)}, x_{t-1}^{(l)}) \log \frac{p(y_t \mid y_{t-1}^{(k)}, x_{t-1}^{(l)})}{p(y_t \mid y_{t-1}^{(k)})}, \tag{1}$$

where $y_{t-1}^{(k)}$ and $x_{t-1}^{(l)}$ denote the histories of Y and X up to lags k and l, respectively. The numerator represents the conditional probability of y_t given its own past and the past of X, while the denominator considers only the past of Y. This measure captures the reduction in uncertainty of y_t due to knowledge of $x_{t-1}^{(l)}$, beyond what is explained by $y_{t-1}^{(k)}$.

Neural Transfer Entropy Estimator

To estimate the transfer entropy within our model, we introduce a neural network estimator T_θ that approximates the conditional probabilities required for the computation of transfer entropy. The estimator takes as input the current state of Y, y_t, the past k states of Y, $y_t^{(k)}$, and the past l states of X, $x_t^{(l)}$:

$$\mathbf{z}_t = [y_t, y_t^{(k)}, x_t^{(l)}] \in \mathbb{R}^{d_z}, \tag{2}$$

where $d_z = d(1 + 2w)$ with $w = \max(k, l)$ and d is the dimensionality of the input features. The neural estimator T_θ is defined as:

$$T_\theta(\mathbf{z}_t) = f_\theta(\mathbf{z}_t), \tag{3}$$

where f_θ is a feedforward neural network parameterized by θ. The output $T_\theta(\mathbf{z}_t)$ represents the estimated log-probability $\log p(y_t \mid y_t^{(k)}, x_t^{(l)})$.

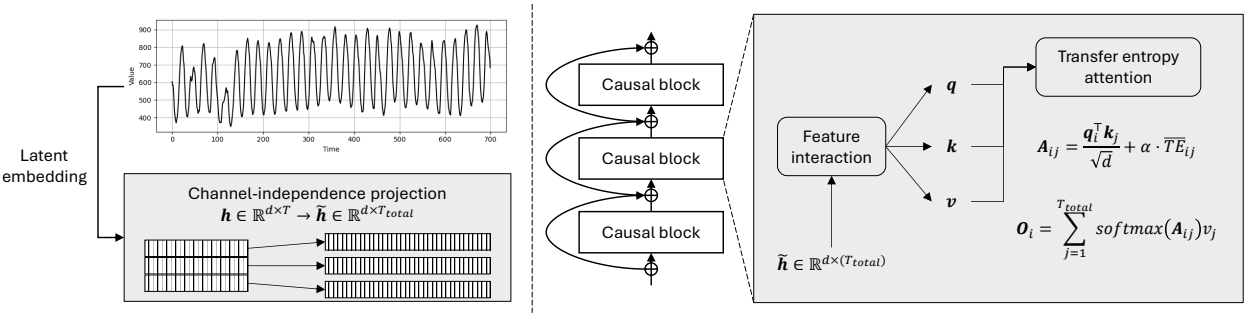


Figure 1: Overview of the proposed method. The channel-independent projection layer maps the input to a prediction horizon after latent embedding. The causal block includes a convolution layer for feature interaction and derives the query (q), key (k), and value (v) vectors. Following this, the transfer entropy attention score is computed and combined with residual connections.

To compute the transfer entropy, we calculate joint term and marginal term. First, the joint term is defined as:

$$T_{\text{joint}} = T_\theta(\mathbf{z}_t) = \log p(y_t \mid y_t^{(k)}, x_t^{(l)}). \tag{4}$$

We shuffle the source history $x_t^{(l)}$ to break the dependency, creating $x_{\text{shuffled}}^{(l)}$. The marginal term is then computed as:

$$\begin{aligned} T_{\text{marginal}} &= T_\theta([y_t, y_t^{(k)}, x_{\text{shuffled}}^{(l)}]) \\ &= \log p(y_t \mid y_t^{(k)}, x_{\text{shuffled}}^{(l)}). \end{aligned} \tag{5}$$

By shuffling $x_t^{(l)}$, we approximate the marginal distribution $p(y_t \mid y_t^{(k)})$. Using these terms, the estimated transfer entropy score at time t is defined as:

$$\text{TE}_t = T_{\text{joint}} - \log \mathbb{E}_{x_{\text{shuffled}}} \left[\exp(T_{\text{marginal}})\right]. \tag{6}$$

This score quantifies the directed information transfer from X to Y at time t, as estimated by the neural network.

Integration into the Attention Mechanism

To capture directional dependencies and causal relationships in time series data, we integrate the transfer entropy estimation into the transformer's attention mechanism. This integration allows the model to adjust attention weights based on the estimated causal influence from one time step to another, enhancing its ability to model temporal causality.

To incorporate transfer entropy into the attention mechanism, we adjust the attention logits as follows:

$$\mathbf{A}_{ij} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} + \alpha \cdot \overline{\text{TE}}_{ij}, \tag{7}$$

where $\alpha = \frac{1}{\sqrt{d}}$ is a scaling factor for consistent calculation, and $\overline{\text{TE}}_{ij}$ is the transfer entropy score between positions i and j with causal mask. The transfer entropy score TE_{ij} is computed using a neural transfer entropy estimator, which takes as input the query, key, and value vectors at positions i and j. Then, the input to the estimator is constructed as:

$$\mathbf{z}_{ij} = [\mathbf{q}_i, \mathbf{k}_j^{(k)}, \mathbf{v}_j^{(l)}]. \tag{8}$$

The neural network estimates the transfer entropy by modeling the conditional dependency of $\mathbf{q}_i$ on $\mathbf{k}_j^{(k)}$ and $\mathbf{v}_j^{(l)}$. For lagged information, we applied zero padding for k_j and v_j with k and l steps, respectively. To enforce temporal causality and prevent the model from attending to future positions, we apply a causal mask to the attention logits. The causal mask ensures that $\text{TE}_{ij} = -\infty$ for all $j > i$, effectively setting the attention weights to zero for future positions. The attention weights are then computed using the softmax function. Finally, the output of the attention mechanism at position i is calculated as:

$$\mathbf{O}_i = \sum_{j=1}^{T} \texttt{softmax}(\mathbf{A}_{ij}) \mathbf{v}_j. \tag{9}$$

By integrating the transfer entropy scores into the attention mechanism and applying causal masking, our model effectively captures the causal relationships and directional dependencies in time series data. This enhances the model's ability to focus on relevant past information when making predictions, leading to improved forecasting performance. Please refer the Appendix for more detailed explanation for implementation of transfer entropy attention.

Model Components

Our model consists of several key components designed to capture temporal dependencies, feature interactions, and causal relationships, as represented in Figure 1.

Channel-independence projection. Motivated from Nie et al. (2022), we apply channel-independence projection to stabilize training. Linear mappings are applied with hidden state $\mathbf{h} \in \mathbb{R}^{d \times T}$ to $\widetilde{\mathbf{h}} \in \mathbb{R}^{d \times T_{\text{total}}}$, where $T_{\text{total}} = T + T_{\text{pred}}$.

Convolution layer for feature interaction. To capture local temporal dependencies and feature interactions, we incorporate temporal convolutional layer, similar to Lea et al. (2017) and Franceschi, Dieuleveut, and Jaggi (2019).

Transfer entropy attention. The proposed model architecture integrates the neural transfer entropy estimator into the attention mechanism. $k = l = 1$ are used for TE.

Experiment

Makridakis, Spiliotis, and Assimakopoulos (2018) introduced the M4 benchmark, a collection of 100,000 time series

Table 1: Performance comparison of transformer-based time series forecasting methods.
(∗ denotes –former; 'Stationary' denotes Non-stationary Transformer.)

Frequency	Metric	LogTrans*	Re*	In*	Pyra*	Auto*	Stationary	FED*	ETS*	PatchTST	Proposed
Yearly	SMAPE	17.107	16.169	14.727	15.530	13.974	13.717	13.728	18.009	13.477	13.400
	MASE	4.177	3.800	3.418	3.711	3.134	3.078	3.048	4.487	3.019	3.024
	OWA	1.049	0.973	0.881	0.942	0.822	0.807	0.803	1.115	0.792	0.790
Quarterly	SMAPE	13.207	13.313	11.360	15.449	11.338	10.958	10.792	13.376	10.380	10.125
	MASE	1.827	1.775	1.401	2.350	1.365	1.325	1.283	1.906	1.233	1.185
	OWA	1.266	1.252	1.027	1.558	1.012	0.981	0.958	1.302	0.921	0.892
Monthly	SMAPE	16.149	20.128	14.062	17.642	13.958	13.917	14.260	14.588	12.959	12.577
	MASE	1.660	2.614	1.141	1.913	1.103	1.097	1.102	1.368	0.970	0.917
	OWA	1.340	1.927	1.024	1.511	1.002	0.998	1.012	1.149	0.905	0.867
Others	SMAPE	23.236	32.491	24.460	24.786	5.485	6.302	4.954	7.267	4.952	4.677
	MASE	16.288	33.355	20.960	18.581	3.865	4.064	3.264	5.240	3.347	3.183
	OWA	5.013	8.679	5.879	5.538	1.187	1.304	1.036	1.591	1.049	0.994
Weighted Average	SMAPE	16.018	18.200	14.086	16.987	12.909	12.780	12.840	14.718	12.059	11.783
	MASE	3.010	4.223	2.718	3.265	1.771	1.756	1.701	2.408	1.623	1.579
	OWA	1.378	1.775	1.230	1.480	0.939	0.930	0.918	1.172	0.869	0.847

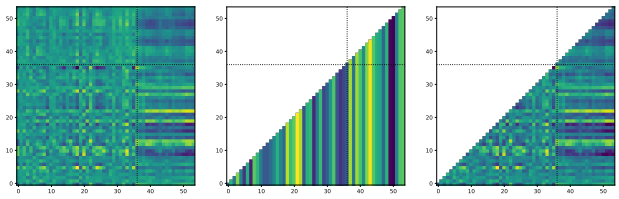
Table 2: Performance comparison of non-transformer-based time series forecasting methods.

Frequency	Metric	LSTM	TCN	N-BEATS	LSSL	LightTS	DLinear	N-HiTS	TimesNet	GPT4TS	Proposed
Yearly	SMAPE	176.040	14.920	13.487	61.675	14.247	16.965	13.422	15.378	15.110	13.400
	MASE	31.033	3.364	3.036	19.953	3.109	4.283	3.056	3.554	3.565	3.024
	OWA	9.290	0.880	0.795	4.397	0.827	1.058	0.795	0.918	0.911	0.790
Quarterly	SMAPE	172.808	11.122	10.564	65.999	11.364	12.145	10.185	10.465	10.597	10.125
	MASE	19.753	1.360	1.252	17.662	1.328	1.520	1.180	1.227	1.253	1.185
	OWA	15.049	1.001	0.936	9.436	1.000	1.106	0.893	0.923	0.938	0.892
Monthly	SMAPE	143.237	15.626	13.089	64.664	14.014	13.514	13.059	13.513	13.258	12.577
	MASE	16.551	1.274	0.996	16.245	1.053	1.037	1.013	1.039	1.003	0.917
	OWA	12.747	1.141	0.922	9.879	0.981	0.956	0.929	0.957	0.931	0.867
Others	SMAPE	186.282	7.186	6.599	121.844	15.880	6.709	4.711	6.913	6.124	4.677
	MASE	119.294	4.677	4.430	91.650	11.434	4.953	3.054	4.507	4.116	3.183
	OWA	38.411	1.494	1.393	27.273	3.474	1.487	0.977	1.438	1.259	0.994
Weighted Average	SMAPE	160.031	13.961	12.250	67.156	13.525	13.639	12.035	12.880	12.690	11.783
	MASE	25.788	1.945	1.698	21.208	2.111	2.095	1.625	1.836	1.808	1.579
	OWA	12.642	1.023	0.896	8.021	1.051	1.051	0.869	0.955	0.940	0.847

spanning business, financial, and economic domains. The benchmark features six distinct collections of data sampled at different frequencies, from hourly observations to yearly records. Detailed specifications of the dataset and benchmark methods are provided in the Appendix.

Table 1 and 2 present performance comparisons between our approach and benchmark methods. We report the mean performance across three runs and incorporate benchmark results as reported in Wu et al. (2023) and Zhou et al. (2023) for fair comparison. Our proposed method achieves superior accuracy across multiple frequency ranges in the M4 forecasting benchmark, demonstrating the advantages of incorporating causal metrics into the transformer architecture. We provide comprehensive ablation studies and sensitivity analyses in the Appendix to further validate our approach.

Figure 2 illustrates the attention patterns in time series forecasting using monthly frequency data. The standard attention map shown in (a) displays the conventional symmetric self-attention pattern. In contrast, our proposed attention mechanism, shown in (c), incorporating with TE in (b), re-



(a) Standard attention (b) Attention map using TE (c) Proposed transfer entropy attention

Figure 2: Example of attention map with proposed transfer entropy attention using 'Monthly' frequency data

sulting in an asymmetric attention pattern that leverages both positional relationships and information-theoretic causal dependencies. This approach enhances the model's ability to capture temporal dynamics, particularly near the prediction horizon, providing a more theoretically grounded approach to temporal dependency modeling.

Conclusion

We presented a transformer architecture that integrates transfer entropy into the attention mechanism, addressing the limitation of traditional transformers in capturing directional dependencies. Theoretical propositions and empirical results support the effectiveness of our approach. Our model achieves improved forecasting accuracy on the M4 dataset, highlighting its potential for time series forecasting tasks.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2024-00407852), and Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health and Welfare, Republic of Korea (HI19C1095).

References

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2): 157–166.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramirez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 6989–6997.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cleveland, R. B.; Cleveland, W. S.; McRae, J. E.; Terpenning, I.; et al. 1990. STL: A seasonal-trend decomposition. *J. off. Stat*, 6(1): 3–73.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32.

Gardner Jr, E. S. 1985. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1): 1–28.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hyndman, R. 2018. *Forecasting: principles and practice*. OTexts.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 156–165.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

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. *Advances in neural information processing systems*, 32.

Lim, B.; Arık, S. Ö.; Loeff, N.; and Pfister, T. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

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

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Makridakis, S.; Spiliotis, E.; and Assimakopoulos, V. 2018. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4): 802–808.

Marschinski, R.; and Kantz, H. 2002. Analysing the information flow between financial time series: An improved estimator for transfer entropy. *The European Physical Journal B-Condensed Matter and Complex Systems*, 30: 275–281.

Medsker, L.; and Jain, L. C. 1999. *Recurrent neural networks: design and applications*. CRC press.

Nauta, M.; Bucur, D.; and Seifert, C. 2019. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1): 19.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Oh, Y.; Kwak, J.; and Kim, S. 2023. Time delay estimation of traffic congestion propagation due to accidents based on statistical causality. *Electronic Research Archive*, 31(2): 691–707.

Oh, Y.; Lim, D.; and Kim, S. 2024. Stable Neural Stochastic Differential Equations in Analyzing Irregular Time Series Data. *arXiv preprint arXiv:2402.14989*.

Oord, A. v. d. 2016. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *nature*, 323(6088): 533–536.

Runge, J.; Heitzig, J.; Petoukhov, V.; and Kurths, J. 2012. Escaping the curse of dimensionality in estimating multivariate transfer entropy. *Physical review letters*, 108(25): 258701.

Schreiber, T. 2000. Measuring information transfer. *Physical review letters*, 85(2): 461.

Tank, A.; Covert, I.; Foti, N.; Shojaie, A.; and Fox, E. B. 2021. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8): 4267–4279.

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

Vicente, R.; Wibral, M.; Lindner, M.; and Pipa, G. 2011. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1): 45–67.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

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

Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*.

Yu, Y.; Si, X.; Hu, C.; and Zhang, J. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7): 1235–1270.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; and Li, J. 2022. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*.

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 the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

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 *International conference on machine learning*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.

Implementation of transfer entropy attention

Standard attention mechanism

In the standard self-attention mechanism (Vaswani 2017), the attention logits are computed as:

$$\mathbf{A}_{ij} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}}, \tag{10}$$

where $\mathbf{q}_i \in \mathbb{R}^d$ is the query vector at position i, $\mathbf{k}_j \in \mathbb{R}^d$ is the key vector at position j, and d is the dimensionality of the key and query vectors. The attention weights are then obtained using the softmax function:

$$\mathbf{W}_{ij} = \frac{\exp(\mathbf{A}_{ij})}{\sum_{m=1}^{T} \exp(\mathbf{A}_{im})}, \tag{11}$$

where T is the sequence length. The output of the attention mechanism at position i is computed as a weighted sum of the value vectors $\mathbf{v}_j$:

$$\mathbf{O}_i = \sum_{j=1}^{T} \mathbf{W}_{ij} \mathbf{v}_j, \tag{12}$$

where $\mathbf{v}_j \in \mathbb{R}^{d_v}$ is the value vector at position j, and d_v is the dimensionality of the value vectors.

Neural transfer entropy estimator

To estimate the transfer entropy from a source process X to a target process Y, we employ a neural network estimator T_θ. The estimator takes as input the current state y_t of Y, the past k states $y_t^{(k)} = [y_{t-k}, \ldots, y_{t-1}]$ of Y, and the past l states $x_t^{(l)} = [x_{t-l}, \ldots, x_{t-1}]$ of X. The input vector is constructed as:

$$\mathbf{z}_t = [y_t, y_t^{(k)}, x_t^{(l)}] \in \mathbb{R}^{d_z}, \tag{13}$$

where $d_z = d_y + k \cdot d_y + l \cdot d_x$, with d_y and d_x being the dimensionalities of Y and X, respectively. The neural estimator computes:

$$T_\theta(\mathbf{z}_t) = f_\theta(\mathbf{z}_t), \tag{14}$$

where f_θ is a feedforward neural network parameterized by θ, outputting a scalar value representing $\log p(y_t \mid y_t^{(k)}, x_t^{(l)})$. To compute the transfer entropy, we calculate two terms:

Joint term:

$$T_{\text{joint},ij} = T_\theta\left([\mathbf{q}_i, \mathbf{k}_j^{(k)}, \mathbf{v}_j^{(l)}]\right), \tag{15}$$

where $\mathbf{q}_i$, $\mathbf{k}_j$, and $\mathbf{v}_j$ correspond to y_i, $y_j^{(k)}$, and $x_j^{(l)}$, respectively. We applied zero padding for the lagged k_j and v_j with k and l steps, respectively.

Marginal term: We create a shuffled version of the source history $x_j^{(l)}$, denoted as $x_{\text{shuffled}}^{(l)}$, to break the dependency between Y and X. The marginal term is then computed as:

$$T_{\text{marginal},ij} = T_\theta\left([\mathbf{q}_i, \mathbf{k}_j^{(k)}, \mathbf{v}_{j,\text{shuffled}}^{(l)}]\right). \tag{16}$$

Transfer entropy score: The transfer entropy score between positions i and j is estimated as:

$$\text{TE}_{ij} = T_{\text{joint},ij} - \log \mathbb{E}_{x_{\text{shuffled}}^{(l)}}\left[\exp\left(T_{\text{marginal},ij}\right)\right]. \tag{17}$$

This score quantifies the information gained about y_i from knowing $x_j^{(l)}$, beyond what is provided by $y_j^{(k)}$.

Adjusted attention mechanism with transfer entropy score with causal masking

To preserve temporal causality and prevent the model from accessing future information, we apply a causal mask to the attention logits and transfer entropy scores. The causal mask is defined as:

$$\text{Mask}_{ij} = \begin{cases} 0, & \text{if } j \le i, \\ -\infty, & \text{if } j > i. \end{cases} \tag{18}$$

Applying the mask to the adjusted attention logits, we have:

$$\overline{\text{TE}}_{ij} = \text{TE}_{ij} + \text{Mask}_{ij}. \tag{19}$$

This ensures that each position i can only attend to its current and previous positions $j \le i$, enforcing the unidirectional flow of information from past to present.

We integrate the transfer entropy scores into the attention mechanism by adjusting the attention logits:

$$\mathbf{A}_{ij} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} + \alpha \cdot \overline{\text{TE}}_{ij}, \tag{20}$$

where $\alpha = \frac{1}{\sqrt{d}}$ is a scaling factor ensuring consistency with the scaled dot-product attention. The inclusion of TE_{ij} biases the attention mechanism towards positions with higher estimated causal influence from X to Y. After adjusting the attention logits, we compute the attention weights using the softmax function:

$$\mathbf{W}_{ij} = \frac{\exp(\mathbf{A}_{ij})}{\sum_m \exp(\mathbf{A}_{im})}, \tag{21}$$

where the summation in the denominator is over all positions m that the model attends to. Finally, the output of the attention mechanism at position i is calculated as:

$$\mathbf{O}_i = \sum_{j}^{T} \mathbf{W}_{ij} \mathbf{v}_j, \tag{22}$$

where the sum is over positions j up to and including i, and $\mathbf{v}_j$ are the value vectors corresponding to the keys.

By integrating the transfer entropy estimator into the attention mechanism, our model effectively captures causal relationships in time series data. The adjusted attention logits incorporate both the similarity between the query and key vectors and the estimated causal influence from the transfer entropy scores. The application of causal masking maintains temporal causality, allowing the model to focus on relevant past information when making predictions.

Experimental details

Dataset

The M4 competition dataset (Makridakis, Spiliotis, and Assimakopoulos 2018) is a comprehensive benchmark for time series forecasting, consisting of 100,000 univariate time series from various domains. The dataset comprises different sampling frequencies as shown Table 3.

Table 3: Data statistics

Dataset	Dimension	Series Length	Dataset Size	Domain
Yearly	1	6	(23000, 23000)	Demographic
Quarterly	1	8	(24000, 24000)	Finance
Monthly	1	18	(48000, 48000)	Industry
Weakly	1	13	(359, 359)	Macro
Daily	1	14	(4227, 4227)	Micro
Hourly	1	48	(414, 414)	Other

Evaluation metrics

For evaluation of forecasting on the M4 benchmark, we adopt the Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Scaled Error (MASE), and Overall Weighted Average (OWA), following the evaluation protocol used in N-BEATS (Oreshkin et al. 2019). Note that OWA is a specific metric used in the M4 competition. The calculations for these metrics are as follows:

$$\text{SMAPE} = \frac{200}{H} \sum_{h=1}^{H} \frac{|\mathbf{Y}_h - \hat{\mathbf{Y}}_h|}{|\mathbf{Y}_h| + |\hat{\mathbf{Y}}_h|},$$

$$\text{MASE} = \frac{1}{H} \sum_{h=1}^{H} \frac{|\mathbf{Y}_h - \hat{\mathbf{Y}}_h|}{\frac{1}{H-s}\sum_{j=s+1}^{H} |\mathbf{Y}_j - \mathbf{Y}_{j-s}|},$$

$$\text{OWA} = \frac{1}{2}\left[\frac{\text{SMAPE}}{\text{SMAPE}_{\text{Naïve2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naïve2}}}\right],$$

where s is the seasonal periodicity of the time series data, H denotes the forecasting horizon, and $\mathbf{Y}_h$ and $\hat{\mathbf{Y}}_h$ are the actual and predicted values at time step h, respectively.

Experimental setup

We implemented our model using the Time Series Library[1] (Wu et al. 2023). The proposed model architecture comprises two causal blocks, each with 32-dimensional embeddings. Both the convolutional layers and attention mechanism operate with a hidden dimension of 32. The neural transfer entropy estimator was implemented as a feedforward network using `ReLU` activation. For optimization, we used a learning rate of 0.001 and a batch size of 32. We applied dimension-wise normalization and denormalization for each batch during training.

Benchmark methods

In this study, we used two groups of benchmark methods: **Transformer-based methods.** (LogTransformer (Li et al. 2019), Reformer (Kitaev, Kaiser, and Levskaya 2020), Informer (Zhou et al. 2021), Pyraformer (Liu et al. 2022a), Autoformer (Wu et al. 2021), Non-stationary Transformer (Liu et al. 2022b), FEDformer (Zhou et al. 2022), ETSformer (Woo et al. 2022), and PatchTST (Nie et al. 2022).) **Non-transformer-based methods** (LSTM (Hochreiter and Schmidhuber 1997), TCN (Franceschi, Dieuleveut, and Jaggi 2019), N-BEATS (Oreshkin et al. 2019), LSSL (Gu, Goel, and Ré 2021), LightTS (Zhang et al. 2022), DLinear (Zeng et al. 2023), N-HiTS (Challu et al. 2023), TimesNet (Wu et al. 2023), and GPT4TS (Zhou et al. 2023).)

[1]https://github.com/thuml/Time-Series-Library

Ablation study

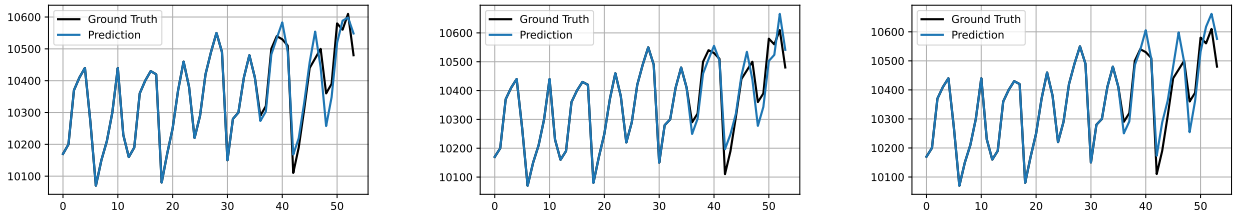
We systematically modify or remove components to assess their contributions to the overall forecasting performance. The key components under investigation are the channel-independence projection, the temporal-convolution layer for feature interaction, and the transfer entropy attention mechanism. In the case of removing channel-independence projection, we used multi-dimensional linear layer instead.

Table 4: Effect of different configurations on model performance using transfer entropy attention

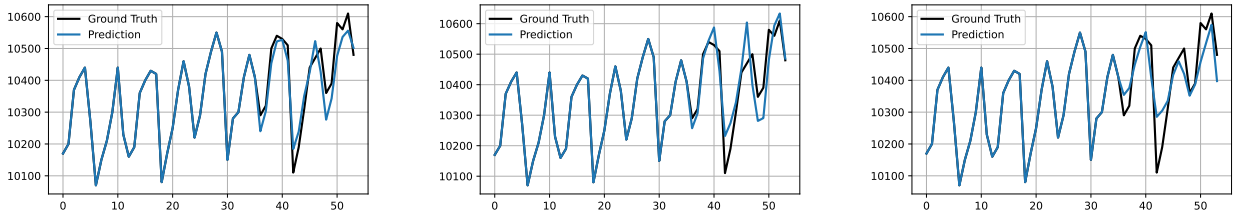
Channel-independence		O	O	X	X	Baseline
Temporal-convolution		O	X	O	X	
Yearly	SMAPE	13.400	13.458	13.448	13.512	13.710
	MASE	3.024	3.059	3.042	3.063	3.087
	OWA	0.790	0.797	0.794	0.799	0.808
Quarterly	SMAPE	10.125	10.082	10.177	10.179	10.487
	MASE	1.185	1.175	1.193	1.190	1.230
	OWA	0.892	0.886	0.897	0.896	0.925
Monthly	SMAPE	12.577	12.729	12.659	12.654	13.217
	MASE	0.917	0.933	0.932	0.931	0.990
	OWA	0.867	0.880	0.877	0.876	0.923
Others	SMAPE	4.677	4.726	4.790	4.880	5.291
	MASE	3.183	3.162	3.224	3.271	3.585
	OWA	0.994	0.996	1.012	1.029	1.122
Weighted Average	SMAPE	11.783	11.862	11.851	11.868	12.279
	MASE	1.579	1.592	1.595	1.600	1.660
	OWA	0.847	0.854	0.854	0.856	0.886

Table 5: Effect of different configurations on model performance using standard attention

Channel-independence		O	O	X	X	Baseline
Temporal-convolution		O	X	O	X	
Yearly	SMAPE	13.504	13.759	13.487	13.716	13.710
	MASE	3.082	3.123	3.050	3.141	3.087
	OWA	0.801	0.814	0.796	0.815	0.808
Quarterly	SMAPE	10.089	10.100	10.120	10.106	10.487
	MASE	1.180	1.179	1.187	1.185	1.230
	OWA	0.889	0.888	0.892	0.891	0.925
Monthly	SMAPE	12.673	12.706	12.618	12.803	13.217
	MASE	0.928	0.928	0.928	0.945	0.990
	OWA	0.875	0.877	0.873	0.888	0.923
Others	SMAPE	4.626	4.639	4.876	5.315	5.291
	MASE	3.150	3.160	3.332	3.456	3.585
	OWA	0.983	0.986	1.038	1.104	1.122
Weighted Average	SMAPE	11.841	11.919	11.831	11.992	12.279
	MASE	1.595	1.605	1.598	1.634	1.660
	OWA	0.853	0.859	0.854	0.869	0.886

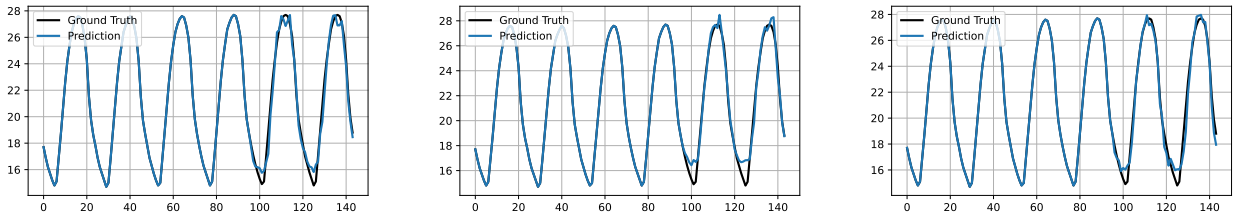


(a) Proposed (b) Without channel-independence (c) Without temporal-convolution

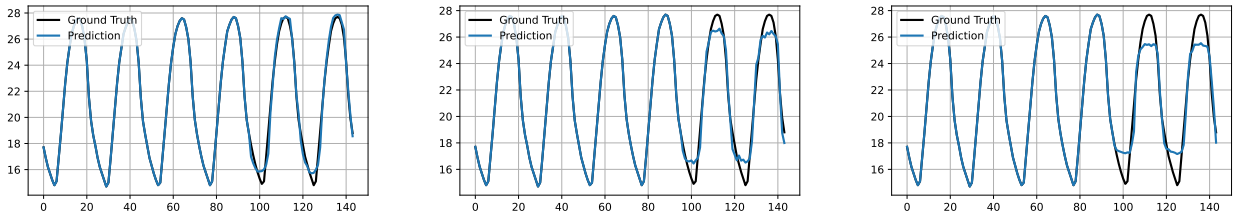


(d) Without both modules (e) Standard attention (f) Linear layer only

Figure 3: Comparison of predicted values using 'Monthly' frequency



(a) Proposed (b) Without channel-independence (c) Without temporal-convolution



(d) Without both modules (e) Standard attention (f) Linear layer only

Figure 4: Comparison of predicted values using 'Hourly' frequency

Tables 4 presents the quantitative results of the ablation study on benchmark datasets. Removing channel-independence projection results in a noticeable decrease in performance, demonstrating its role in stabilizing training and enhancing feature representation. Also, temporal convolution shows its capability of capturing local temporal dependencies and feature interactions. Removing both the channel-independence projection and convolution layer further degrades performance, suggesting that these components complement each other in the model.

In Table 5, replacing the transfer entropy attention with standard attention reduces the model's effectiveness. The proposed transfer entropy attention consistently outperforms the standard attention, confirming its advantage in modeling causal relationships. The full model significantly outperforms the baseline linear layer model, emphasizing the collective contributions of the proposed components.

Qualitative Analysis Figures 3 and 4 illustrate the forecasting results under different ablation settings, using monthly frequency and hourly frequency, respectively. The following configurations are considered: (a) Proposed method with all components: The full model incorporating channel-independence projection, temporal convolution layer, and transfer entropy attention; (b) Without channel-independence projection: The model without the channel-independence projection module; (c) Without temporal-convolution layer: The model

without the convolution layer for feature interaction. (d) Without both modules: The model without both the channel-independence projection and temporal-convolution layer; (e) Standard attention: The model using standard attention instead; (f) Linear layer only: The baseline model using only a linear layer.

The proposed method with all components closely follows the ground truth, capturing both trends and fluctuations effectively. In contrast, different ablation settings from (b) to (f) show the degraded performance in forecasting.

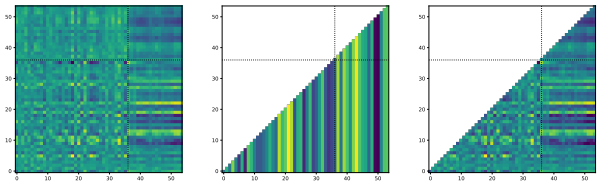
Sensitivity analysis

In the proposed transfer entropy attention, the lag parameters k and l play a crucial role in capturing temporal dependencies by determining the number of past time steps considered for the target and source variables, respectively. By default, we set $k = l = 1$, implying that the model looks one step back in time for both variables when estimating transfer entropy. To understand the impact of these parameters on forecasting performance, we conduct a sensitivity analysis by varying $k = l$ from 1 to 5.

Table 6: Sensitivity analysis of time lag (k, l)

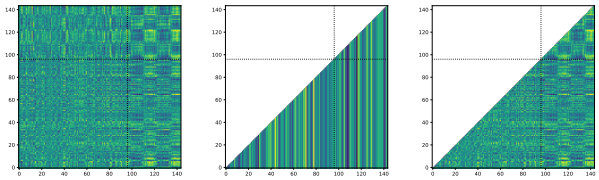
Lag ($k = l$)		1	2	3	4	5
Yearly	**SMAPE**	13.400	13.403	13.491	13.641	13.529
	MASE	3.024	3.026	3.045	3.103	3.061
	OWA	0.790	0.791	0.796	0.808	0.799
Quarterly	**SMAPE**	10.125	10.201	10.127	10.132	10.093
	MASE	1.185	1.197	1.182	1.180	1.175
	OWA	0.892	0.899	0.891	0.890	0.887
Monthly	**SMAPE**	12.577	12.653	12.604	12.546	12.657
	MASE	0.917	0.923	0.920	0.916	0.928
	OWA	0.867	0.872	0.870	0.866	0.875
Others	**SMAPE**	4.677	4.719	4.676	4.626	4.617
	MASE	3.183	3.154	3.180	3.140	3.137
	OWA	0.994	0.994	0.993	0.982	0.980
Weighted Average	**SMAPE**	11.783	11.840	11.817	11.822	11.840
	MASE	1.579	1.584	1.585	1.594	1.588
	OWA	0.847	0.851	0.850	0.853	0.852

Table 6 presents the model's performance with varying lag parameters. From our sensitivity analysis of the lag parameters k and l, we observe that the model's performance remains robust across different lag values in general. Specifically, for datasets with longer sampling frequencies—such as yearly, quarterly, or monthly data—the effect of varying the lag parameters is negligible. This indicates that considering additional past time steps beyond the immediate previous state does not significantly enhance forecasting accuracy in these cases. However, for datasets with shorter sampling frequencies, like weekly, daily, or hourly data, adjusting the lag parameters leads to noticeable improvements in performance. The enhancement in these higher-frequency datasets can be attributed to their longer sequence lengths, which allow the model to benefit from incorporating information from multiple past time steps. BThus, while the model is generally robust to changes in lag parameters, carefully selecting k and l becomes more impactful for datasets with higher temporal resolutions.



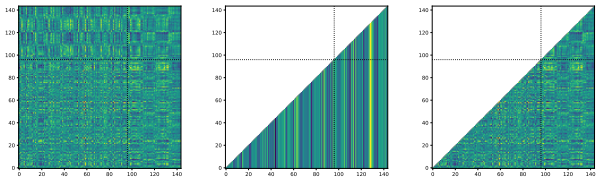
(a) Standard attention (b) Attention map using TE (c) Proposed transfer entropy attention

Figure 5: Attention map with proposed transfer entropy attention ($k = l = 1$) using 'Monthly' frequency data



(a) Standard attention (b) Attention map using TE (c) Proposed transfer entropy attention

Figure 6: Attention map with proposed transfer entropy attention ($k = l = 1$) using 'Hourly' frequency data



(a) Standard attention (b) Attention map using TE (c) Proposed transfer entropy attention

Figure 7: Attention map with proposed transfer entropy attention ($k = l = 5$) using 'Hourly' frequency data

Figure 5 shows the attention mechanisms for monthly data with $k = l = 1$, where the standard attention exhibits symmetric patterns while the TE score map reveals clear directional dependencies. The dashed lines indicate prediction horizons, and the proposed method consistently demonstrates how transfer entropy guides the attention mechanism to focus on informationally relevant temporal relationships.

For hourly data in Figure 6, similar patterns emerge with $k = l = 1$, though the attention maps show finer granularity due to the higher sampling frequency. When increasing the lag parameters to $k = l = 5$ for hourly data in Figure 7, the attention patterns become more smooth, capturing longer-range dependencies while preserving the essential asymmetric structure of transfer entropy.