NIMO: Module-Level Interpretability for Time Series X-formers via Mask Joint Optimization

Shaoqi Tan[1], Yong Wang[1†], Hongwei Zhu[1], Zhicheng Zhang[2], Wen Yin[1], Ruizheng Huang[1]
[1]University of Electronic Science and Technology of China
[2]National University of Defense Technology

***Abstract*—Transformer-based models, denoted as X-formers, employ complex architectures to capture intricate temporal dependencies. However, their internal decision-making processes lack transparency, hindering the understanding of how specific modules interact with distinct time-series patterns. To address this challenge, we introduce Neural Network Input and Module Mask Optimization (NIMO), a unified and fully differentiable framework designed to provide fine-grained module-level interpretability. Unlike traditional methods that rely on unstable discrete searches or input-only attribution, NIMO leverages a Temporal-Module Fusion Network (TMFN) and a joint optimization strategy to establish a direct mapping between dynamic data features and static model modules. This methodology explicitly unveils the functional roles within X-formers, revealing that Position Embeddings and Attention are critical for capturing seasonality, while Temporal Embeddings and Feed-Forward Networks primarily model trends. Extensive experiments on 12 representative X-formers validate the fidelity of our explanations. Furthermore, we demonstrate that this interpretability is actionable: by pruning identified non-salient modules, NIMO achieves a 46.40% improvement in computational efficiency and a 42.35% reduction in model parameters without compromising forecasting performance.**

***Index Terms*—time series forecasting, module-level interpretability, x-formers, mask joint optimization**

I. Introduction

In recent years, Transformer-based models, denoted as X-formers, have achieved significant performance improvements in time series forecasting. However, as these architectures grow in depth and complexity, they have increasingly evolved into black boxes. This lack of transparency severely hinders the deployment of X-formers in high-stakes domains such as healthcare and power grid management, where understanding the rationale behind a prediction is as critical as the prediction itself. Consequently, opening this black box to reveal the underlying decision-making mechanisms has become a focal point of current research.

To address this challenge, existing interpretability research generally falls into two paradigms: Ante-hoc and Post-hoc methods. Ante-hoc methods aim to achieve interpretability through intrinsic architectural designs. Evolving from early attention-based RNNs [1] and TCNs [2], the Temporal Fusion Transformer (TFT) [3] established a benchmark by explicitly identifying critical features via variable selection networks. Recent advancements have further expanded this direction.

†This work was supported by the Sichuan Science and Technology Program under Grant Number 2024ZDZX0011.

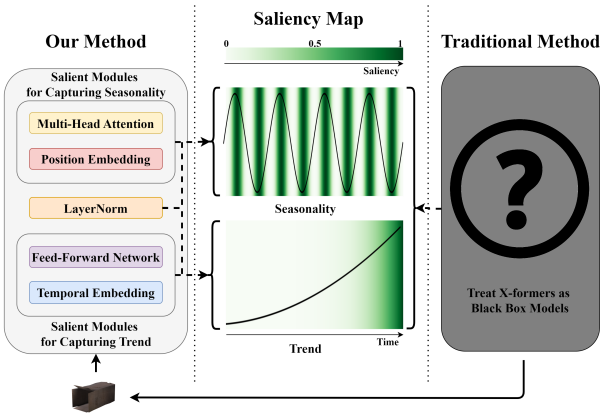

Fig. 1. Comparison between traditional input-only attribution and NIMO's structure-aware interpretation. Unlike traditional methods that treat the model as a black box (right), NIMO opens the box to reveal the fine-grained mapping between dynamic temporal features and static functional modules (left).

For instance, BasisFormer [4] interprets forecasting as a linear combination of learnable basis vectors, while SHAPformer [5] and ShapTST [6] attempt to integrate Shapley value estimation directly into the forward pass of the network. However, these methods often face a dilemma where achieving interpretability necessitates imposing specific constraints on the architecture, which can limit model expressiveness or compromise forecasting performance. Furthermore, **they are inherently inapplicable to the vast array of existing effective yet opaque X-former architectures**.

Conversely, Post-hoc methods interpret trained models by attributing predictions to input features without altering the model structure. While early Gradient-based methods such as Integrated Gradients [7] and DeepSHAP [8] offer computational efficiency, they often suffer from gradient saturation and noise in deep nonlinear models. Consequently, state-of-the-art research has shifted towards Perturbation-based methods including Dynamask [9], Extrmask [10], and Gatemask [11]. These approaches utilize learnable optimization objectives to dynamically identify optimal input masks and accurately pinpoint salient time steps. While these methods effectively identify input saliency, **they fail to identify the specific architectural components responsible for processing these features**, as illustrated in Fig 1. Although GeMIMO [12] recently attempted module-level interpretation using genetic

algorithms, **its discrete search strategy suffers from convergence instability in high-dimensional spaces**.

To overcome these limitations, we propose Neural Network Input and Module Mask Optimization (**NIMO**), **a unified and fully differentiable framework designed to provide stable, fine-grained interpretability**. Unlike traditional methods that treat the model as a black box, NIMO leverages a Temporal-Module Fusion Network (TMFN) and a joint optimization strategy to establish a direct mapping between dynamic temporal features and static architectural modules. NIMO facilitates this through three coordinated stages: Input Perturbation, Module Perturbation, and Joint Optimization. Input Perturbation employs the TMFN to fuse temporal dynamics with structural embeddings, generating context-aware input masks. Simultaneously, Module Perturbation systematically evaluates the contribution of individual modules, such as Attention or Feed-Forward Networks, by optimizing learnable binary masks. Finally, Joint Optimization synchronizes these masks to ensure that the revealed correspondence between data patterns and model modules is both accurate and consistent. By systematically coordinating these three stages, NIMO transforms the interpretability analysis into a holistic evaluation of the alignment between data and modules, explicitly revealing the correspondence between temporal patterns and functional components.

Our main contributions are summarized as follows:

- We introduce NIMO, a unified and fully differentiable interpretability framework that transcends the limitations of traditional methods. By leveraging the TMFN and a joint optimization strategy, NIMO stably establishes a fine-grained mapping between dynamic time-series features and static architectural modules, transforming the analysis of opaque X-formers into a transparent, module-level evaluation.
- We reveal the underlying decision-making mechanisms within complex X-formers. Unlike previous methods that only identify salient time steps, NIMO explicitly exposes distinct interaction patterns, such as the critical role of Position Embeddings in capturing seasonality versus the dominance of Temporal Embeddings in modelling long-term trends.
- We validate the effectiveness of NIMO through comprehensive Mask Fidelity experiments. Results demonstrate that NIMO outperforms state-of-the-art baselines in identifying salient data features. Furthermore, by pruning non-salient modules, we achieve a 46.40% improvement in computational efficiency and a 42.35% reduction in model parameters without compromising forecasting performance.

II. Related Works

A. Ante-hoc Methods

Ante-hoc methods achieve interpretability through intrinsic architectural designs. Early research utilized attention-based RNNs such as RETAIN [1] and CNNs including TCN-Attention [2] and DSAN [13]. Additionally, prototype-learning frameworks like TapNet [14] were proposed. However, the primary research focus has since shifted towards Transformers. In this domain, the TFT [3] established a benchmark by integrating variable selection networks with temporal self-attention to explicitly identify critical features. Recent advancements have further expanded Transformer interpretability beyond simple attention weights. For instance, BasisFormer [4] interprets forecasting as a linear combination of learnable basis vectors. Concept Bottleneck Models [15] constrain latent representations to align with human-understandable concepts. Notably, SHAPformer [5] and ShapTST [6] incorporate Shapley value estimation directly into the forward pass of the network in an attempt to unify efficient inference with theoretically grounded explanations.

B. Post-hoc Methods

Post-hoc methods interpret trained models by attributing predictions to input features without altering internal architectures. Initially transplanted from CV and NLP, Gradient-based methods including GRAD [16], Integrated Gradients [7], DeepLIFT [17], GradSHAP [18], and DeepSHAP [8] estimate feature saliency via local sensitivity or partial derivatives. However, they often suffer from gradient saturation and noise issues in deep non-linear forecasting models. Consequently, research focus has shifted to perturbation-based methods, which infer feature saliency by systematically perturbing inputs and observing the resulting changes in model outputs. While early approaches such as RISE [19] and Feature Occlusion [20] applied random or fixed masking, they risk disrupting temporal continuity. Addressing this issue, state-of-the-art techniques such as Dynamask [9], Extrmask [10], CGS-mask [21], and ContraLSP [11] introduce learnable optimization objectives to dynamically identify optimal masks. These methods ensure explanations are robust and respect the sequential dependencies inherent in time series data. Recently, GeMIMO [12] proposed identifying salient modules via genetic algorithms. However, its discrete search strategy often suffers from convergence instability, particularly when applied to deep models and high-dimensional time series data.

III. Methodology

We now introduce the NIMO framework, whose overall architecture is shown in Fig 2. NIMO is a unified and fully differentiable framework designed to enhance the interpretability of X-formers by explicitly revealing the intrinsic correspondence between dynamic input features and static model modules. NIMO comprises three main stages: input perturbation, module perturbation, and joint optimization. The input perturbation and module perturbation independently perturb the input data $\mathbf{X} \in \mathbb{R}^{T \times D}$ (where D is feature dimensionality and T is sequence length) and the modules of X-formers, evaluating the saliency of each element based on the resulting impact on the model's inference. The joint optimization simultaneously refines both input and module masks to maximize model interpretability.

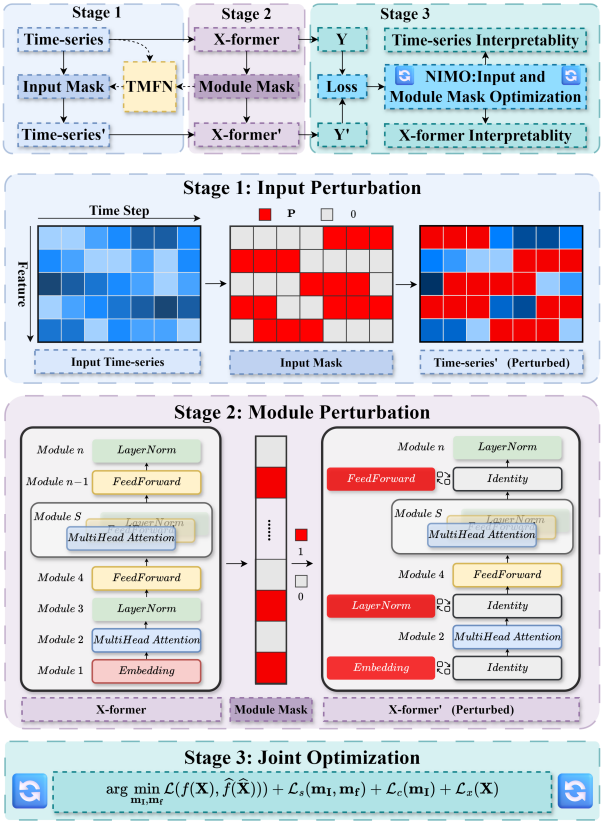

Fig. 2. The overall architecture of the proposed NIMO framework

A. Input Perturbation

We utilize a learnable input mask $\mathbf{m_I} \in [0,1]^{T\times D}$ to identify salient segments in the input data $\mathbf{X}$. Instead of analyzing input saliency in isolation, we propose the Temporal-Module Fusion Network (TMFN), illustrated in Fig 3, to generate $\mathbf{m_I}$ from the input data $\mathbf{X}$ conditioned on the specific architectural state represented by the module mask $\mathbf{m_f} \in \{0,1\}^{1\times N}$, where N denotes the number of modules in the X-formers. This mechanism ensures that $\mathbf{m_I}$ captures context-aware dependencies between data patterns and the model's structural decisions.

- **Feature Fusion:** First, we project the input $\mathbf{X}$ to align with the dimensions of $\mathbf{m_f}$. This step enables the integration of structural information via element-wise operations, preparing the fused features for subsequent context Aggregation.
- **Context Aggregation:** An attention mechanism first aggregates temporal dependencies guided by the module mask. Subsequently, a Feed-Forward Network fuses information across feature dimensions to capture complex inter-variable interactions. This two-step process yields a refined latent representation $\mathbf{W_I} \in \mathbb{R}^{T\times D}$ that encodes both temporal and structural contexts.
- **Mask Generation:** To allow gradient-based optimization, we generate $\mathbf{m_I}$ using a reparameterization approach. Let $\mu_I \in \mathbb{R}^{T\times D}$ be a learnable parameter matrix. We first compute the intermediate soft-mask $\mu'_I \in \mathbb{R}^{T\times D}$ via a structural-conditioned gating mechanism:

$$\mu'_I = \mu_I \odot \sigma(\mathbf{W_I} \odot \mu_I). \tag{1}$$

Subsequently, we introduce Gaussian noise $\epsilon_I \sim \mathcal{N}(0, \delta^2)$ for robustness and obtain the final mask via clipping:

$$\mathbf{m_I} = \min\left(1, \max(0, \mu'_I + \epsilon_I)\right). \tag{2}$$

Finally, the perturbed input $\widehat{\mathbf{X}}$ is generated by mixing the original input with a perturbation baseline $\mathbf{P}$:

$$\widehat{\mathbf{X}} = \Phi_I(\mathbf{X}, \mathbf{m_I}) = \mathbf{m_I} \odot \mathbf{X} + (\mathbf{1} - \mathbf{m_I}) \odot \mathbf{P}, \tag{3}$$

where $\mathbf{P} = \mathrm{NN}(\mathbf{X})$ is generated by a bidirectional GRU [10] to serve as an informative baseline.

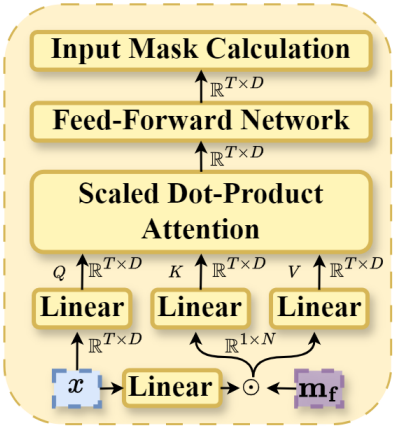

Fig. 3. The detailed architecture of the TMFN

B. Module Perturbation

The module mask $\mathbf{m_f}$ identifies the modules of the X-formers that contribute significantly to the model's performance. The process consists of three key steps:

- **Nonlinear Soft-Masking:** To enhance gradient flow and mitigate saturation issues prevalent in deep architectures, we strictly map the structural parameters $\mu_f \in \mathbb{R}^{1\times N}$ through a self-gated activation function to obtain the intermediate soft-mask μ'_f:

$$\mu'_f = \mu_f \odot \sigma(\mu_f) = \frac{\mu_f}{1 + e^{-\mu_f}} \tag{4}$$

- **Discrete Mask Generation:** During the forward pass, the discrete module mask $\mathbf{m_f}$ is derived via a quantization operation on the self-gated values:

$$\mathbf{m_f} = \lfloor \min(1, \max(0, \mu'_f)) + 0.5 \rfloor \tag{5}$$

When the i-th module is identified as non-salient (where $\mathbf{m_f}[i] = 0$), it is functionally replaced by an Identity Block. We implement this structurally using a residual-weighted formulation to preserve the data flow topology:

$$\widehat{\mathbf{Y}}_i = \mathbf{m_f}_{[i]} \cdot f_i(\widehat{\mathbf{X}}_i) + (1 - \mathbf{m_f}_{[i]}) \cdot \widehat{\mathbf{X}}_i \tag{6}$$

where $\widehat{\mathbf{Y}}_i$ denotes the perturbed output of the i-th module, $f_i(\widehat{\mathbf{X}}_i)$ represents the original output of the i-th layer, $\widehat{\mathbf{X}}_i$ is the input to the i-th layer, and $\mathbf{m_f}_{[i]}$ serves as the binary decision gate. The perturbation process for the entire model is defined as Φ_f, yielding $\widehat{f} = \Phi_f(f, \mathbf{m_f})$. The final joint output $\widehat{\mathbf{Y}}$ combines both input and module perturbations:

$$\widehat{\mathbf{Y}} = \widehat{f}(\widehat{\mathbf{X}}) = \Phi_f(f, \mathbf{m_f}) \circ \Phi_I(\mathbf{X}, \mathbf{m_I}) \tag{7}$$

- **Approximate Gradient Flow:** A critical challenge in optimizing binary masks is the vanishing gradient problem caused by the floor function. To address this, we apply the STE strategy during backpropagation. The quantization operator is bypassed, allowing the gradient $\nabla_{\mathbf{m_f}}\mathcal{L}$ to flow directly to the intermediate mask μ'_f and subsequently to the structural parameters μ_f:

$$\frac{\partial \mathcal{L}}{\partial \mu_f} \approx \frac{\partial \mathcal{L}}{\partial \mu'_f} \cdot \frac{\partial \mu'_f}{\partial \mu_f} \tag{8}$$

This mechanism ensures stable convergence and enables NIMO to optimize discrete structural choices within a fully differentiable framework.

C. Joint Optimization

The joint optimization learning objective of NIMO is to optimize the input mask $\mathbf{m_I}$ and the module mask $\mathbf{m_f}$ to achieve a balance between predictive consistency and interpretability. The optimization problem can be formulated as:

$$\arg\min_{\mathbf{m_I}, \mathbf{m_f}} \mathcal{L}(f(\mathbf{X}), \widehat{f}(\widehat{\mathbf{X}})) + \mathcal{L}_s(\mathbf{m_I}, \mathbf{m_f}) + \mathcal{L}_c(\mathbf{m_I}) + \mathcal{L}_x(\mathbf{X}) \tag{9}$$

The detailed components of this objective function are defined as follows:

- **Fidelity Loss:** Defined as $\mathcal{L}(f(\mathbf{X}), \widehat{f}(\widehat{\mathbf{X}}))$, it measures the discrepancy between the original and perturbed model outputs, ensuring that the selected features faithfully represent the model's decision logic.
- **Perturbation Constraint ($\mathcal{L}_x$):** Defined as $\lambda_x \|\mathrm{NN}(\mathbf{X})\|_1$, it constrains the magnitude of the generated perturbation baseline $\mathbf{P}$ to ensure it remains within a valid range, preventing out-of-distribution shifts.
- **Sparsity Constraint ($\mathcal{L}_s$):** Defined as $\lambda_s(\|\mathbf{m_I}\|_1 + \|\mathbf{m_f}\|_1)$, it encourages the identification of the most critical data segments and modules by filtering out redundant information.
- **Continuity Constraint ($\mathcal{L}_c$):** Defined as $\lambda_c \sum_{t=1}^{T-1} \|\mathbf{m_I}_{[t+1]} - \mathbf{m_I}_{[t]}\|_1$, it penalizes abrupt transitions in $\mathbf{m_I}$ to ensure temporal saliency respects the sequential continuity of time series data.

IV. Evaluation Results

A. Saliency Analysis of X-formers

Setups: To validate NIMO as a unified interpretability framework, we conducted a systematic analysis on 12 representative X-formers [22]–[33]. To allow the models sufficient capacity to disentangle complex temporal patterns and distribute functional roles, we configured the networks with adequate depth. This setup provides a rigorous testbed for NIMO to pinpoint truly salient components in a complex architecture. Specifically, encoder-only models were configured with 6 encoder layers, while encoder-decoder models employed 3 layers for both the encoder and the decoder. For the long-term forecasting task, we fixed both the input look-back window and the prediction horizon at 96 time steps. Robustness is ensured by reporting the mean and standard deviation across 5 random seeds. All models were trained on an A800 GPU with 80 GB of memory. We empirically selected $\lambda_s = \lambda_c = \lambda_x = 1$ via grid search over $\{0.01, 0.1, 1, 10, 100\}$. The remaining protocols followed the TimesNet [34] setting. For statistical analysis, we categorized the functional modules into Position Embedding (PE), Temporal Embedding (TE), Attention (Attn), Feed-Forward Network (FFN), and LayerNorm (LN).

Temporal Saliency Patterns: A preliminary examination of the input masks generated by NIMO (Fig 4) reveals two distinct temporal patterns. The seasonality-dominant pattern (e.g., Traffic, Electricity) is characterized by recurrent intervals of high saliency, indicating a strong dependence on periodicity. Conversely, the trend-dominant pattern (e.g., ETTm2, exchange) is characterized by contiguous saliency segments or a primary focus on recent history, reflecting a reliance on local continuity and evolving trends.

Architectural Saliency Patterns: Table I presents the module saliency scores S, calculated as the average retention rate across 5 random seeds: $S = \frac{1}{5\times N}\sum_{k=1}^{5}\sum_{i=1}^{N} \mathbf{m_f}_{[i]}^{(k)} \times 100\%$, where $\mathbf{m_f}_{[i]}^{(k)} \in \{0, 1\}$ denotes the binary state of the i-th module in the k-th run. The results demonstrate a clear architectural specialization:

- **Role of PE and Attn in Seasonality:** In seasonality-dominant datasets such as Electricity, PE and Attn exhibit high saliency scores, with PE surpassing 64 and Attn approaching 60. This finding underscores the critical mechanism for capturing periodicity: PE encodes the sequential order essential for identifying fixed temporal intervals, while Attn utilizes these cues to retrieve correlated patterns from historical cycles.
- **Role of TE and FFN in Trend:** In contrast, trend-dominant datasets such as Exchange exhibit a marked reliance on TE and FFN. Specifically, TE achieves a saliency score of 71.4 while FFN maintains a substantial score of 57.5. This importance alignment is theoretically consistent with the functional characteristics of these components: TE provides continuous temporal semantics essential for tracking evolving dynamics, while FFN acts as the primary non-linear approximator required to fit complex trend curves.
- **Stabilization Role of LN:** Distinct from other modules that adapt to specific data patterns, LN maintains a stable importance score of approximately 55 across most datasets. This consistency indicates that LN functions as a fundamental stabilizer for X-formers, ensuring training

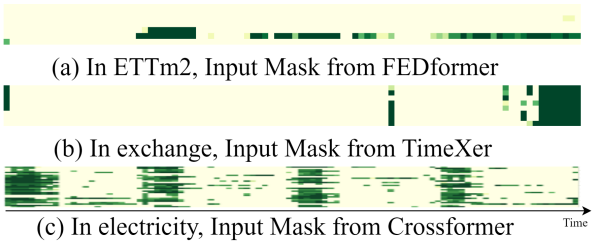

Fig. 4. Long-term Forecasting Data Saliency Analysis. Fig.(a) and (b) represent trend-dominated data, while Fig.(c) represents seasonality-dominated data. The intensity of green color corresponds to the saliency level of data features.

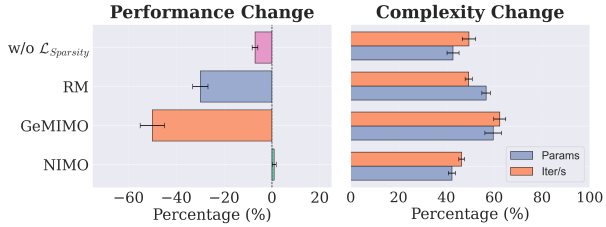

Fig. 5. Quantitative Evaluation of Module Intervention.

stability independent of specific data characteristics.

Stability and Variance Analysis: The standard deviation results in Table I reveal a distinct stability contrast. Widespread components such as LN and FFN exhibit minimal variance due to their distributed presence across deep networks. This suggests that their contribution is structural and consistent. Conversely, PE and TE display higher fluctuations, such as PE reaching a standard deviation of 8.1 on the Traffic dataset. Since these modules exist only at the input stage, the binary mask acts as a strict gate, resulting in higher sensitivity. Notably, despite this sensitivity, NIMO maintains significantly lower variance compared to search-based methods, validating the stability of our differentiable optimization strategy.

B. Mask Fidelity Analysis

Setups: To verify the fidelity of NIMO's interpretations, we conducted Module Intervention and Salient Feature Retention experiments. We compared NIMO with state-of-the-art saliency methods, including GeMIMO, CGS-mask, Dynamask, Extrmask, and Gatemask. All experiments on X-formers followed the protocol, metrics, and datasets used in TimesNet.

Module Intervention: We assessed the fidelity of the generated module masks by observing performance shifts under Module intervention. In addition to GeMIMO, we also introduced Random Mask (RM) as a baseline. The results are shown in Fig 5. NIMO identifies that approximately 42.4% of the parameters (Params) in standard X-formers are functionally redundant. By pruning these non-salient modules, we achieved a 46.4% improvement in computational efficiency (Iter/s) without compromising forecasting accuracy. This result transcends a simple engineering improvement and scientifically validates NIMO's ability to distinguish salient modules from non-salient ones. If NIMO had incorrectly identified a salient module as non-salient, the intervention would have led to a performance collapse.

Salient Feature Retention: To assess the fidelity of the identified salient features, we conducted experiments on salient feature retention across long-term forecasting, short-term forecasting, and classification tasks. The results are shown in Table II. NIMO yielded the largest drop in MSE_n and SMAPE_n, while achieving a substantial accuracy gain. This superiority is primarily attributed to NIMO's TMFN, which effectively captures critical information across both temporal and feature dimensions. Additionally, by excluding non-salient modules during the attribution process, NIMO reduces the interference of irrelevant model components, leading to cleaner and more accurate saliency maps.

Ablation Study: We validated the effectiveness of our optimization objectives through ablation studies. The ablation experiment for the module mask is illustrated in Fig 5. When the sparsity constraint of the module mask is removed, NIMO struggles to focus on salient modules, leading to a decline in mask fidelity. The ablation experiment for the input mask is presented in Table II. Excluding the sparsity and smoothness constraints results in the input mask exhibiting noise and discontinuities, thereby preventing the accurate localization of key temporal patterns. Furthermore, removing the perturbation constraint leads to the generation of uninformative perturbations, which impairs NIMO's capacity to recognize salient features.

TABLE I
MODULE SALIENCY SCORES OF X-FORMERS ACROSS DIFFERENT DATASETS.

Datasets	PE	TE	FFN	Attn	LN
ETTh1	35.7 ± 6.5	71.4 ± 8.1	53.6 ± 1.7	51.2 ± 1.5	47.5 ± 0.2
ETTh2	42.9 ± 3.2	71.4 ± 6.5	47.3 ± 0.6	37.5 ± 1.5	52.1 ± 2.7
ETTm1	35.7 ± 6.5	50.0 ± 3.2	49.2 ± 2.8	49.6 ± 2.5	57.8 ± 2.7
ETTm2	35.7 ± 4.8	50.0 ± 4.8	63.3 ± 0.6	48.9 ± 2.0	56.8 ± 1.5
illness	50.0 ± 4.8	50.0 ± 6.5	54.6 ± 2.8	53.2 ± 2.0	49.7 ± 0.2
weather	64.3 ± 3.2	21.4 ± 6.5	38.9 ± 0.6	54.3 ± 2.5	57.2 ± 0.2
exchange	50.0 ± 3.2	71.4 ± 3.2	57.5 ± 1.0	51.8 ± 0.9	56.1 ± 2.7
electricity	64.3 ± 4.8	35.7 ± 3.2	52.2 ± 2.4	58.9 ± 1.5	59.9 ± 2.7
traffic	71.4 ± 8.1	50.0 ± 6.5	50.2 ± 0.3	59.8 ± 1.5	54.8 ± 0.2

TABLE II
WHEN THE RETENTION RATIO OF SALIENT FEATURES IN DIFFERENT SALIENCY METHODS CHANGES FROM 0.1 TO 0.6, THE PERFORMANCE CHANGES OF THE X-FORMERS. $\textbf{MSE}_n$ REPRESENTS NORMALIZED MSE, $\textbf{SMAPE}_n$ REPRESENTS NORMALIZED SMAPE.

Methods	$\text{MSE}_n\downarrow$	$\text{SMAPE}_n\downarrow$	Accuracy↑
Dynamask	-0.0791 ± 0.0052	-0.2830 ± 0.0081	**0.3619 ± 0.0045**
Extrmask	-0.1046 ± 0.0048	-0.3218 ± 0.0075	0.2838 ± 0.0051
Gatemask	-0.1219 ± 0.0035	-0.3132 ± 0.0090	0.2364 ± 0.0062
CGS-mask	-0.1178 ± 0.0041	-0.3419 ± 0.0068	0.2864 ± 0.0044
w/o $\mathcal{L}_{Sparsity}$	-0.1052 ± 0.0038	-0.3121 ± 0.0065	0.2915 ± 0.0040
w/o $\mathcal{L}_{Smooth}$	-0.1185 ± 0.0032	-0.3544 ± 0.0055	0.3012 ± 0.0038
w/o $\mathcal{L}_x$	-0.1005 ± 0.0028	-0.3012 ± 0.0048	0.2885 ± 0.0035
NIMO	**-0.1239 ± 0.0022**	**-0.3877 ± 0.0053**	0.3177 ± 0.0049

V. Conclusion

In this paper, we propose NIMO, a unified and fully differentiable framework designed to enhance the interpretability of X-formers by establishing a fine-grained mapping between dynamic temporal features and static architectural modules. By leveraging the TMFN and a stable joint optimization strategy, NIMO effectively reveals how specific model components interact with distinct data patterns. Our empirical analysis **unveils** distinct functional roles within X-formers: PE and Attn are crucial for capturing seasonality, while TE and FFN primarily focus on modelling trends. Furthermore, interventional experiments validate the fidelity and actionability of our explanations. By pruning non-salient modules identified by NIMO, we achieve a 46.40% improvement in computational efficiency and a 42.35% reduction in model parameters without compromising performance. These results confirm that NIMO provides accurate and stable guidance for analyzing and understanding complex time series X-formers.

References

[1] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," *Advances in neural information processing systems*, vol. 29, 2016.

[2] L. Pantiskas, K. Verstoep, and H. Bal, "Interpretable multivariate time series forecasting with temporal attention convolutional neural networks," in *IEEE Symposium Series on Computational Intelligence*. IEEE, 2020, pp. 1687–1694.

[3] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International journal of forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[4] Z. Ni, H. Yu, S. Liu, J. Li, and W. Lin, "Basisformer: Attention-based time series forecasting with learnable and interpretable basis," *Advances in Neural Information Processing Systems*, vol. 36, pp. 71 222–71 241, 2023.

[5] M. Hertel, S. Pütz, R. Mikut, V. Hagenmeyer, and B. Schäfer, "Explainable time-series forecasting with sampling-free SHAP for Transformers," *arXiv preprint arXiv:2512.20514*, 2025.

[6] Q. Cheng, J. Xing, C. Xue, and X. Yang, "Unifying prediction and explanation in time-series transformers via shapley-based pretraining," in *2025 21st IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 2025, pp. 325–330.

[7] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[8] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[9] J. Crabbé and M. Van Der Schaar, "Explaining time series predictions with dynamic masks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2166–2177.

[10] J. Enguehard, "Learning perturbations to explain time series predictions," in *International Conference on Machine Learning*. PMLR, 2023, pp. 9329–9342.

[11] Z. Liu, Y. Zhang, T. Wang, Z. Wang, D. Luo, M. Du *et al.*, "Explaining time series via contrastive and locally sparse perturbations," in *International Conference on Learning Representations*, 2024.

[12] Z. Zhang, Y. Wang, S. Tan, and Y. Luo, "GeMIMO: Searching the cores of X-formers for time series forecasting," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2025, pp. 1–5.

[13] H. Lin, R. Bai, W. Jia, X. Yang, and Y. You, "Preserving dynamic attention for long-term spatial-temporal prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 36–46.

[14] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, "TapNet: Multivariate time series classification with attentional prototypical network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6845–6852.

[15] A. van Sprang, E. Acar, and W. Zuidema, "Interpretability for time series Transformers using a concept bottleneck framework," in *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

[16] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.

[17] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3145–3153.

[18] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, 2018.

[19] V. Petsiuk, A. Das, and K. Saenko, "RISE: randomized input sampling for explanation of black-box models," in *British Machine Vision Conference*. BMVA Press, 2018, p. 151.

[20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[21] F. Lu, W. Li, Y. Sun, C. Song, Y. Ren, and A. Y. Zomaya, "CGS-Mask: Making time series predictions intuitive for all," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, 2024, pp. 14 149–14 157.

[22] Z. Zhang, Y. Wang, S. Tan, B. Xia, and Y. Luo, "Enhancing Transformer-based models for long sequence time series forecasting via structured matrix," *Neurocomputing*, p. 129429, 2025.

[23] Y. Wang, J. Peng, X. Wang, Z. Zhang, and J. Duan, "Replacing self-attentions with convolutional layers in multivariate long sequence time-series forecasting," *Applied Intelligence*, pp. 522–543, 2024.

[24] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2023.

[25] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed Transformer for long-term series forecasting," in *International Conference on Machine Learning*, 2022.

[26] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary Transformers: Exploring the stationarity in time series forecasting," in *Advances in Neural Information Processing Systems*, 2022, pp. 9881–9893.

[27] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong *et al.*, "Informer: Beyond efficient Transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[28] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, 2021, pp. 22 419–22 430.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu *et al.*, "TimeXer: Empowering Transformers for time series forecasting with exogenous variables," in *Advances in Neural Information Processing Systems*, 2024, pp. 469–498.

[31] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma *et al.*, "iTransformer: Inverted Transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

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

[33] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu *et al.*, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International Conference on Learning Representations*, 2021.

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