Non-stationary Time-aware Kernelized Attention for Temporal Event Prediction

Yu Ma*
Zhining Liu*
haishan.my@antgroup.com
eason.lzn@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

Chenyi Zhuang
chenyi.zcy@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

Yize Tan
yize.tyz@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

Yi Dong
dongyi.dy@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

Wenliang Zhong
yice.zwl@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

Jinjie Gu
jinjie.gujj@antgroup.com
Ant Group
Hangzhou, Zhejiang, China

ABSTRACT

Modeling sequential data is essential to many applications such as natural language processing, recommendation systems, time series predictions, anomaly detection, etc. When processing sequential data, one of the critical issues is how to capture the temporal-correlation among events. Though prevalent and effective in many applications, conventional approaches such as RNNs and Transformers, struggle with handling the non-stationary characteristics (i.e., such temporal-correlation among events would change over time), which is indeed encountered in many real-world scenarios. In this paper, we present a non-stationary time-aware kernelized attention approach for input sequences of neural networks. By constructing the Generalized Spectral Mixture Kernel (GSMK), and integrating it to the attention mechanism, we mathematically reveal its representation capability in terms of the time-dependent temporal-correlation. Following that, a novel neural network structure is proposed, which would enable us to encode both stationary and non-stationary time event series. Finally, we demonstrate the performance of the proposed method on both synthetic data which presents the theoretical insights, and a variety of real-world datasets which shows its competitive performance against related work.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; Kernel methods.

KEYWORDS

temporal event prediction; kernelized attention; non-stationarity

ACM Reference Format:
Yu Ma, Zhining Liu, Chenyi Zhuang, Yize Tan, Yi Dong, Wenliang Zhong, and Jinjie Gu. 2022. Non-stationary Time-aware Kernelized Attention for Temporal Event Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539470

1 INTRODUCTION

Modeling sequential data is essential to many applications such as recommendation systems, time series predictions, anomaly detection, etc. Unlike isolated event datasets, which only depend on features and contexts at one point, sequential data requires catching the relation among events. And in most real-world scenarios, such relations not only depend on the characteristics of the event itself but also the ordering of temporal or spatial information at which the events occur. For example, in continuous-time event sequences, the time span between the event occurrences has significant implications on predicting the next occurring event. A more specific example would be its periodical pattern of a re-occurring event.

Therefore, classical sequence modeling introduces mechanisms like positional encoding and functional time representations to count for order and temporal patterns. RNN models the order and temporal information implicitly by assuming $y_t = M(y_{t-1}) + \epsilon_t$ [2, 31], where $M(\cdot)$ is the mapping function and ϵ_t represents the noise term. To improve the modeling of long-range dependency, the gated mechanism has been introduced into RNN structures which leads to GRU and LSTM [3, 11]. Some recent researches also further improve GRU or LSTM units by adding specifically designed time gates to capture the temporal information of the sequence [1]. The attention mechanism [27] itself contains no recurrence structure to count for the order and temporal information, thus it introduces positional encoding explicitly to model the relative or absolute position of the tokens in the sequence. Since the positional encoding can only incorporate the ordering information, recent researches have extended the positional encoding to several different approaches of time representation learning to further address the temporal patterns [15, 30].

[*] Both authors contributed equally to this research.

KDD '22, August 14–18, 2022, Washington DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539470

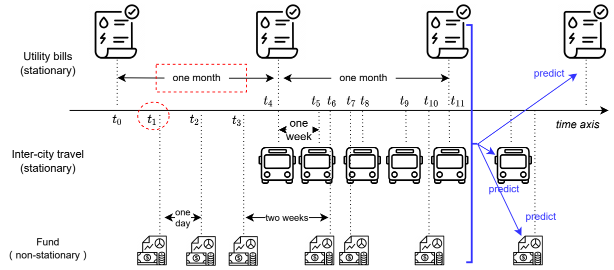

Figure 1: Illustration of a discrete mixed periodic signal, including two stationary cases and one non-stationary case.

Most of the aforementioned methodology is under the implicit assumption that the dataset is stationary, i.e., the underlying distribution does not change over time. However, that is not the case in most real-world scenarios. In recommendation systems, user behaviors and preferences usually change over time [7, 13, 21]. Similar time-varying issues can be found in financial time series [6], medical data analysis [17, 19], etc. Figure 1 illustrates a discrete mixed periodical signal, which records a user's usage of different online services. Among them, *utility bills* and *inter-city travel* services present stationary periodicity, while the purchase of *fund* is non-stationary. Unlike the above mentioned related work that mainly focuses on stationary signals, since the periodicity of a non-stationary single would change over time, a powerful time encoding method should jointly consider both of the time intervals (marked by the red rectangle in Figure 1) and the absolute time (marked by the red circle in Figure 1). Therefore, extending the time representation learning ability in neural network structures to non-stationary signals is new and of significant value in many applications.

In this paper, we propose a non-stationary time-aware kernelized attention (NsTKA) method to address the aforementioned time-varying issues. Specifically, the NsTKA directly models the temporal relations between events utilizing a non-stationary kernel without introducing explicit time mapping functions. Unlike the conventional attention mechanism [27] that makes event input and time input as a whole (e.g., by embedding addition or concatenation), our proposed method handles them separately. Hence, we first present the mathematical relationship between related work and our method. Then, a new attention neural network structure is devised, which strictly corresponds to our proposed NsTKA. Finally, the experimental results on both real-world and synthetic datasets demonstrate the superiority of the proposed method.

To summarize, the main contributions of this paper can be described as follows:

- Mathematically, we present the proposed methodology and its relationship with other related works.
- We design a new non-stationary time-aware kernelized attention structure to address the time-varying properties of sequential data. The structure as a plugin, can also be extended to a variety of neural network structures or other temporal process methodologies.
- Our proposed method obtains competitive performance on the synthetic dataset and extensive real-world datasets.

2 RELATED WORK

2.1 Sequential Encodings

In this paper, we are interested in incorporating non-stationary order and temporal patterns into the attention mechanism. Although the classic attention calculation is order-independent (namely, it is an operation on sets), a wealth of related work has been dedicated to adding explicit positional or temporal encodings.

Positional Encoding. While the original Transformer uses a sinusoidal position signal (see Eq.3 for details) or learnable position embeddings [27], instead of encoding absolute position, it has recently obtained significant improvement by incorporating relative position embeddings [5, 24]. Although extending the attention mechanism to efficiently consider representations of the relative positions between sequence elements allows the model to better recognize ordering information, it suffers from modeling continuous-time event sequences, in which the time span between events often matters.

Temporal Encoding. To overcome the shortcoming above, several temporal encoding methods are proposed. In [30], the authors proposed a method that embeds time span into high-dimensional spaces using a set of stationary kernel basis functions in the form of sine and cosine. In [25], the authors explicitly defined a learnable embedding method that captured both periodic (i.e., the sine term) and non-periodic (i.e., the linear term) patterns. In sequential recommendation, [15] considered both time intervals between two items and the absolute positions of them. Nevertheless, few works have been conducted on non-stationary time series.

Recurrence and Convolution. Through sequential structure, conventional RNNs (e.g., LSTM [11] and GRU [3]) model the order and temporal information implicitly by $h_t = M(h_{t-1}) + x_t$ [2, 31], where M is a nonlinear map, and h_t and x_t are n-vectors representing respectively the hidden state and the external input at time t. To deal with continuous time series, time gate [34] and classic temporal point process [20, 32] are also introduced to couple with RNNs. Although less common, by stacking several layers, CNNs, like [8], inherently capture relative positions within the kernel size of each convolution. We argue that these work are in general orthogonal to ours, and to some extent, our kernelized method can be plugged into any methods aiming to capture the pairwise temporal relations (e.g., time intervals in the time gate).

2.2 Kernelized Attention

With the tremendous success of Transformers, kernelized attention is raised as a new research direction. In [26], the authors presented a new formulation of attention from a view of kernel. To deal with quadratic complexity of attention calculation, [18] proposed a kernelized attention for acceleration. Regarding kernels as a form of approximation of the attention matrix, they can be also viewed as a form of low-rank method [4]. Unlike most of the kernelized methods that focus on reducing the computational complexity, by introducing a non-stationary kernel, our work aims to improve the time representation ability of the attention mechanism. Furthermore, using the mathematical relationship between kernel and attention, in this paper, we organically couple the proposed non-stationary temporal encoding kernel with the original attention mechanism.

3 PRELIMINARIES

We focus on solving the non-stationary time encoding problem in the attention manner. In this section, we first present the main notations used in this problem. Then, the basic attention module, positional and temporal encoding, and kernel functions will be introduced briefly.

3.1 Notations

Let $\mathcal{S} = \{(t_1, e_1), ..., (t_n, e_n)\}$ denotes the sequence that contains a series of event e_i happened at time t_i, and $\phi_t(t), \phi_e(e)$ represents learnable mapping functions that encode t, e into $\mathbb{R}^d$. Accordingly, we use matrices $V_e, V_t \in \mathbb{R}^{n \times d}$ to denote the encoded events and temporal embeddings of the sequence $\mathcal{S}$.

In practice, the temporal embedding V_t is usually added or concatenated with the event embedding V_e. Without loss of generality, we use V to denote the combination of V_t and V_e. On the other hand, regarding the input to attention module, we use K, Q, V to denote the key, query, and value matrices, respectively. Note that, in our self-attention setting, the key and query matrices equal to the value matrix, i.e., $K = Q = V$.

3.2 Attention Module

The attention mechanism in the original Transformer [27] is formed in the scaled dot-product manner:

$$Attention(Q, K, V) = softmax(\frac{(QW^Q)(KW^K)^T}{\sqrt{d_k}})(VW^V), \quad (1)$$

where d_k is the dimension of the hidden representation of the key. W^K, W^Q, W^V are the projection matrices for any specific head of Q, K, V, respectively. To simplify notations in k-head attention module that has k sets of projection matrices, in the following sections, we omit them in equations.

To be more specific, let $\mathbf{q_i}, \mathbf{k_i}, \mathbf{v_i}$ represent the $i-$th component of Q, K, V, which are vectors representing the input embedding at $i-$th position of the query, key, value sequences. $\mathbf{z} = (\mathbf{z_1}, ..., \mathbf{z_i}, ...\mathbf{z_n})$ denotes the output of attention module. Then, the attention formula can be rewritten as:

$$\begin{aligned} \mathbf{z_i} &= \sum_{j=1}^{n} \frac{k_{exp}(\mathbf{q_i}, \mathbf{k_j})}{\sum k_{exp}(\mathbf{q_m}, \mathbf{k_l})} \mathbf{v_j}, \\ k_{exp}(\mathbf{q_i}, \mathbf{k_j}) &= exp(\frac{\mathbf{q_i k_j^T}}{\sqrt{d_k}}). \end{aligned} \quad (2)$$

As we can see from Eq.2, the relative contribution of each key-value pair to each query is determined by the attention weight $\frac{k_{exp}(\mathbf{q_i}, \mathbf{k_j})}{\sum k_{exp}(\mathbf{q_m}, \mathbf{k_l})}$ times the value vector $\mathbf{v_j}$.

3.3 Positional and Temporal Embedding

To count for the order information of a sequence, [27] added a non-learnable positional embedding, i.e., V_t, as the input into the attention module. Each entry in V_t is calculated by the following sin and cos functions, respectively:

$$\begin{aligned} PE(pos, 2i) &= sin(\frac{pos}{10000^{2i/d_{model}}}), \\ PE(pos, 2i+1) &= cos(\frac{pos}{10000^{2i/d_{model}}}), \end{aligned} \quad (3)$$

where pos is the position of the token/event in the sequence, i corresponds to 2 consecutive position embedding dimensions, and d_{model} is the dimension of the token/event embedding. In practice, the dimension of the positional encoding is usually the same as d_{model} that the positional embedding and event embedding can be summed as the attention inputs.

Recently, in [30], the authors further proposed a functional time encoding method which maps the continuous-time values t to high dimensional space by Mercer's feature map:

$$t \mapsto \Phi = [\Phi_{\omega_1}(t), ..., \Phi_{\omega_d}(t)]^T, \quad (4)$$

where

$$\Phi_{\omega_i}(t) = \sum_{j=1}^{N} \sqrt{c_{2j}(\omega_i)} cos\frac{j\pi t}{\omega_i} + \sqrt{c_{2j+1}(\omega_i)} sin\frac{j\pi t}{\omega_i}. \quad (5)$$

$[\sqrt{c_1(\omega)}, ..., \sqrt{c_{2j}(\omega)}cos\frac{j\pi t}{\omega}, \sqrt{c_{2j+1}(\omega)}sin\frac{j\pi t}{\omega}, ...]$ is the truncated Fourier basis under certain frequencies of a class of continuous, positive semi-definite (PSD) and translation-invariant periodical kernels. The advantage of this Mercer's time encoding is that it can take the continuous time-span information instead of the discrete position index. And the frequencies ω_k and corresponding coefficients $\sqrt{c_{kj}}$ are learnable parameters that can be jointly optimized to cover a broad range of bandwidths in order to capture various temporal patterns of the signal.

Limitation discussion. These related methods cannot encode a non-stationary signal well. Specifically, by seeing Eq.3, the amplitude (i.e., 1) and frequencies (i.e., $\frac{1}{10000^{2i/d_{model}}}$) are predefined and then kept constant during the learning process, which limits its representation capability. Furthermore, since it takes the discrete positional index as input, it cannot encode irregular continuous time-interval information into the representation. In Eq.5, although the amplitude (i.e., $\sqrt{c_*(\omega_*)}$) and frequencies (i.e., $\frac{j\pi}{\omega_*}$) are learnable, they are independent with the absolute time t. As mentioned above, the Mercer's time embedding is essentially a dual representation of a temporal translation-invariant kernel. Taking the Mercer's embedding into the context of expressing temporal correlations between events $< \Phi_d^M(t_i), \Phi_d^M(t_j) >$, the correlation would naturally be temporal translation-invariant thus limiting its capability of encoding non-stationary signals.

3.4 Kernel Functions

A real kernel function [28] $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric, PSD function in its arguments for which the following property holds $k(x, x') = \phi(x) \cdot \phi(x')$ for some feature map ϕ. Kernel functions usually serve as a similarity measure between two inputs in the machine learning filed [14].

A stationary kernel is a function of $\tau = x - x'$, i.e., it is invariant to translation of the inputs. Standard kernels such as Gaussian kernel and Matérn kernels are both stationary. In [29], the authors proposed a Spectral Mixture (SM) kernel which supports a broad class of stationary covariances:

$$k(\tau) = \sum_i \alpha_i exp(-2\pi^2 \sigma_i^2 \tau) cos(2\pi \mu_i \tau), \tag{6}$$

where α_i are the weights, σ_i, and μ_i are the variance and mean of mixture Gaussian components.

A non-stationary kernel, on the other hand, can infer the relations in a input-dependent manner, i.e., it is a function of x, x', and $\tau = (x - x')^T$. A classic extension of the stationary SM kernel (i.e., Eq.6) to a non-stationary Spectral Mixture kernel is in the format of:

$$k(x, x', \tau) = \sum_i \alpha_i^2 exp(-2\pi^2 \tau^T \Sigma_i \tau) \Psi_{\mu_i, \mu_i'}(x)^T \Psi_{\mu_i, \mu_i'}(x'),$$

where

$$\Psi_{\mu_i \mu_i'}(x) = \begin{pmatrix} cos(2\pi\mu_i x) + cos(2\pi\mu_i' x) \\ sin(2\pi\mu_i x) + sin(2\pi\mu_i' x) \end{pmatrix}, \tag{7}$$

$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & \rho_i \sigma_i \sigma_i' \\ \rho_i \sigma_i \sigma_i' & \sigma_i'^2 \end{bmatrix}.$$

α_i is the mixture weight for each component. ρ_i, (μ_i, μ_i') , (σ_i, σ_i') are the correlation, means and variances of the bivariate Gaussian components, respectively.

By using a mixture of bi-variate Gaussian components, Eq.7 intrinsically is a closed-form solution to the generalized Fourier inverse transform:

$$k(x, x') = \int_R \int_R e^{2\pi i (x\omega - x'\omega')} \mu_\omega(d\omega, d\omega'), \tag{8}$$

where μ_ω is a positive bounded symmetric measure [9] associated to some PSD spectral density function $S(\omega, \omega')$, which is denoted as the spectral surface. When the spectral measure mass is concentrated only on the diagonal (i.e., $\omega = \omega'$), its closed-form solution would reduce to a stationary one, e.g., Eq.6. For the detailed description, please refer to [23].

Our proposed method is on the basis of the non-stationary Spectral Mixture kernel. In the next section, we first present the mathematical rationale of introducing such a non-stationary kernel into the attention mechanism. Then, in the context of machine learning, by decomposing the kernel function, we explain why it can count for the time-dependent periodical and long-range dependency. Finally, a new attention neural structure is devised. In its implementation, several constraints would be added for robustness.

4 METHODOLOGY

4.1 Attention with Temporal Embedding in Kernel Perspective

Given a sequence $\mathcal{S} = \{(t_1, e_1), ..., (t_n, e_n)\}$ and its mapping functions $\phi_t(t)$ and $\phi_e(e)$ that encodes t, e into $\mathbb{R}^d$, conventional attention related methods usually add or concatenate the event embedding $\phi_e(e)$ and the temporal embedding $\phi_t(t)$. We use the additive form (i.e., $\mathbf{x}_i = \phi_{t,i} + \phi_{e,i}$) as the case to illustrate the attention with temporal embedding in a kernel view [1]. To simplify notations, we omit the variable t_i, e_i in the $\phi_t(\cdot)$ and $\phi_e(\cdot)$.

As stated in Eq.2, the output of the attention module can be viewed as a superposition of relative contributions from each key-value pair to each query. To be more specific, we would consider the attention weights which depend on the similarity measure between the query and key, and the value vector separately. The following mathematical illustration applies to general attention mechanisms, but out of simplicity, we use the self-attention case to demonstrate the derivation where the query, key and value are all the same as $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$.

Following Eq.2 and omitting the constant normalizing factor $\sqrt{d_k}$ for notation simplicity, the attention weights, with explicitly expressing the temporal and event embedding, can be reformatted as :

$$\begin{aligned} k_{exp}(\mathbf{x}_i, \mathbf{x}_j) &= k_{exp}(\phi_{t,i} + \phi_{e,i}, \phi_{t,j} + \phi_{e,j}) \\ &= exp((\phi_{t,i} + \phi_{e,i})(\phi_{t,j} + \phi_{e,j})^T) \\ &= exp(\phi_{t,i}\phi_{t,j}^T + \phi_{e,i}\phi_{e,j}^T + \phi_{t,i}\phi_{e,j}^T + \phi_{e,i}\phi_{t,j}^T) \\ &= k_{exp}(\phi_{t,i}, \phi_{t,j}) \cdot k_{exp}(\phi_{e,i}, \phi_{e,j}) \\ &\quad \cdot k_{exp}(\phi_{t,i}, \phi_{e,j}) \cdot k_{exp}(\phi_{t,j}, \phi_{e,i}), \end{aligned} \tag{9}$$

where the first two terms, $k_{exp}(\phi_{t,i}, \phi_{t,j})$ and $k_{exp}(\phi_{e,i}, \phi_{e,j})$ clearly represent the temporal and event correlation between (t_i, e_i) and (t_j, e_j). With regard to the two cross terms $k_{exp}(\phi_{t,i}, \phi_{e,j})$ and $k_{exp}(\phi_{t,j}, \phi_{e,i})$, they represent the correlation between time i and event j and vice versa.

The first two terms indicate that the event and temporal correlation are calculated separately. From Eq.9, since the exponential of a dot-product operation (i.e., k_{exp}) is noted as an instance of kernel operation [14, 28], the attention weights with temporal embedding can be further considered as the product of temporal kernel and event kernel separately.

The two cross terms are a byproduct of summing up the temporal and event embedding. Since the time and event are projected into two separate vector spaces, intuitively the two cross terms have no valid physical meaning. Even worse, they introduce a kind of noise in the attention weights. Several recent related work [5, 26] also discussed this issue. They claimed that using the product of separate temporal and non-temporal kernels is better than using the direct sum of temporal and non-temporal embeddings.

Since we focus on improving the representation learning ability of the non-stationary temporal correlation, in accordance with Eq.9, in the next subsection, we introduce a non-stationary kernel function to replace the exponential term $k_{exp}(\phi_{t,i}, \phi_{t,j})$.

4.2 Encode Temporal Correlation with Non-stationarity

In this section, we introduce a kernel function to encode temporal correlation with non-stationarity. Note that, the original $k_{exp}(\phi_{t,i}, \phi_{t,j})$ function can be regarded as exponential of a linear kernel. As has been discussed in section 3.3, it suffers from encoding the non-stationary temporal correlation. In order to infer non-stationary long-range and periodic correlations in an input-dependent manner, we utilize the Generalized Spectral Mixture

[1] Note that, since the concatenating form does not introduce the cross-terms $k_{exp}(\phi_{t,i}, \phi_{e,j})$, $k_{exp}(\phi_{t,j}, \phi_{e,i})$, obviously it is consistent with our conclusion that the event and temporal correlations can be calculated separately. We omit its derivation in this paper.

(GSM) kernel proposed in [22]:

$$k_{GSM}(x, x') = \sum_i \alpha_i(x)\alpha_i(x')k_{Gibbs,i}(x, x')cos(2\pi(\mu_i(x)x - \mu_i(x')x')), \quad (10)$$

where

$$k_{Gibbs,i}(x, x') = \sqrt{\frac{2l_i(x)l_i(x')}{l_i(x)^2 + l_i(x')^2}}exp(-\frac{(x - x')^2}{l_i(x)^2 + l_i(x')^2}). \quad (11)$$

Mean $\mu_i(x)$, lengthscale $l_i(x)$ and weight $\alpha_i(x)$ of each component are input-dependent learnable functions. The kernel is essentially a product of three kernels, and its PSD is guaranteed since all of the product kernels are PSD.

Eq.10 is derived from Eq.7, which originally aims to overcome the limit that the non-stationary kernel in Eq.7 vanishes rapidly outside the origin $(x, x') = (0, 0)$. When encoding temporal correlation in our problem, this kernel has the following advantages:

- the cosine term can represent the periodic correlation.
- the Gibbs’ kernel (Eq.11), which is a non-stationary generalization of the Gaussian kernel [10] , can encode both global and local correlations, namely the short- and long- range temporal dependency.
- the value of weight $\alpha_i(x)$ also changes over time, which dynamically adjusts the importance of each component.

Since all the three terms in Eq.10 is input-dependent, it can represent non-stationary temporal correlation well. On the basis of this kernel, we next introduce the implementation of a new attention structure.

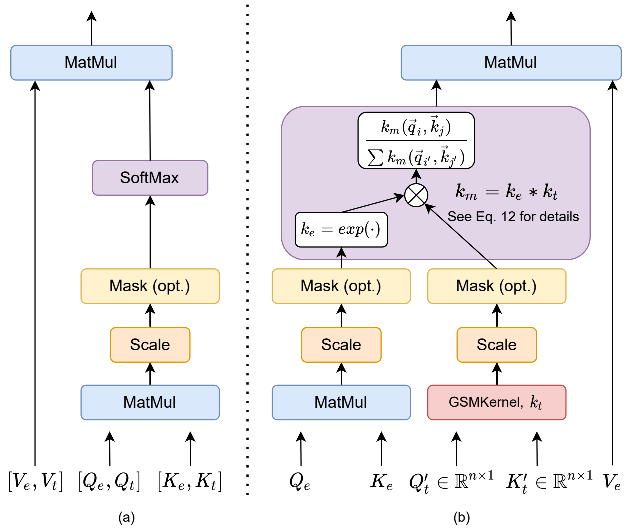

Figure 2: Implementation of NsTKA. (a) The original scale-dot-product attention; (b) Our proposed NsTKA

4.3 Implementation of the NsTKA

Following the above sections, the framework of NsTKA method is illustrated as Figure 2. In the NsTKA, we replace the $k_{exp}(\mathbf{q_i}, \mathbf{k_j})$ in Eq.2 with a new kernel k_m:

$$k_m(\mathbf{q_i}, \mathbf{k_j}) = k_t(t_{q_i}, t_{k_j}) \cdot k_e(\mathbf{e}_{q_i}, \mathbf{e}_{k_j}), \quad (12)$$

where $k_t(t_{q_i}, t_{k_j})$ denotes the temporal kernel which takes the continuous time scalar value of the query and key as input and follows the GSM kernel (i.e., Eq.10) expression. And $k_e(\mathbf{e}_{q_i}, \mathbf{e}_{k_j})$ denotes the event kernel which takes the event embedding of the query and key as input and remains the same expression as $k_{exp}(\cdot)$ in Eq.2.

For the temporal GSM kernel part $k_t(\cdot)$, since $\mu_i(x), l_i(x), \alpha_i(x)$ in Eq.10 are all input-dependent learnable parameters, we use a two-layer fully-connected neural network to jointly optimize them along the training process. Since all the learnable parameters have clear physical meanings, the ReLU activation functions are used to make them non-negative. In addition, a small constant $\epsilon = 1e - 6$ is added to the length scale $l_i(x)$ layer to ensure it is positive.

Differences between our method and related work. Compared with the conventional attention structure shown in Figure 2(a), the first difference is that we use the GSM kernel to calculate the temporal attention weights. The second one is that only the non-temporal input are used as values, i.e., V_e. Although viewed in a kernel way, the implementation of the attention mechanism in [5] is still through the mapping to temporal and event embedding spaces. In [26], a similar method is proposed, which uses the direct product of two separate kernels to represent the temporal and non-temporal correlations. The authors experimented with a few standard kernels such as linear, polynomial and Gaussian kernels without further discussing their properties. In [30], the authors discussed the characteristics of temporal correlation starting in a kernel view but transferred that into a series of basis functions to express stationary temporal embedding. Although the aforementioned work has discussed the kernel view of expressing attention weights in a temporal and non-temporal correlation way, there is limited discussion in diving into the design choice of kernels. In the next section, by comparison, we present the competitive performance of our method.

5 EXPERIMENTS

In this section, we evaluate the proposed method on both a synthetic dataset and 3 real-world datasets. A quantitative comparison with baseline models is first presented, followed by a comprehensive ablation experiment that shows the theoretical insights of the proposed method. Our experiments are intended to answer the following questions:

- **Q1**: Does the proposed method outperform the state-of-the-art temporal event prediction methods?
- **Q2**: By adjusting the mean, length-scale, and weight functions of the GSM kernel, how do variants of NsTKA perform?
- **Q3**: How does NsTKA perform under different levels of non-stationarity?

5.1 Datasets

5.1.1 Synthetic Dataset. A synthetic dataset (SynD) with a mixture of multiple non-stationary signals is introduced to study how the proposed method captures the non-stationary pattern. The dataset is generated by :

$$x(t) = \sum_{i=1}^{N} A_i(t)cos(\omega_i(t)t) + \phi_i) + \epsilon_i, \quad (13)$$

where $A_i(t), \omega_i(t)$ are the amplitude and frequency of each composed sinusoidal signal, respectively, ϕ_i is a random phase between

0 and 2π generated by uniform distribution, and ϵ_i is the noise term following normal distribution. It is well known that when the amplitudes and frequencies A_i, ω_i are both constants with regard to time t, the signal is stationary, otherwise the signal would be non-stationary.

To illustrate the capability of dealing with the non-stationary signal of our proposed method, the amplitudes are designed to be linearly increasing or decreasing with respect to time:

$$A_i(t) = A_{i,0}(1 + \alpha_i t/T), \tag{14}$$

where $A_{i,0}$ is the initial amplitude when $t = 0$, α_i is the increasing or decreasing coefficient corresponding to positive or negative values, and T is the length of the overall time. Similarly, the frequencies are designed to be exponentially increasing or decreasing with time:

$$\omega_i(t) = \omega_{i,0} * exp(\beta_i t/T), \tag{15}$$

where $\omega_{i,0}$ is the initial frequency when $t = 0$ and β_i is the increasing or decreasing coefficient corresponding to positive or negative values. In the following experiments, we use 5 sinusoidal components with different frequencies and amplitudes and varying factors to represent our SynD dataset.

In addition, to observe how different types of non-stationarities affect the performance of different methods, we combine time-varying amplitude function and time-varying frequency function, which leads to three datasets, i.e., SynD-A (only time-varying amplitude function), SynD-F (only time-varying frequency function), SynD-AF (both time-varying amplitude and frequency function are included). Here the task is designed as one-step prediction given a fixed length of the input sequence. Due to that the synthetic dataset is controllable, it is of significant importance for presenting the insights into our method.

5.1.2 Real-world Datasets. To further demonstrate the effectiveness of the proposed method, we compare different methods on three real-world datasets from various domains:

- **ETT** (Electricity Transformer Temperature)[2] is a crucial indicator in the electric power long-term deployment. [33] collected and published 2-year data from two separated counties in China. In this paper, we use the 1-hour level data collection in the first site (noted as $ETTh_1$). The data point consists of the target value "oil temperature" and 6 power load features. The task is to predict the oil temperature in the next few hours.
- **PM2.5** dataset [16] records hourly PM2.5 value and the associated meteorological measurements in Beijing City of China from the year 2010 to 2014. The PM2.5 values are categorized into six levels according to the United States Environmental Protection Agency standard. Our task is to predict the class of the PM2.5 level on the next day at 8 am using the collected data in the previous day, when is the commuter peak period in Beijing [12].
- **An industrial dataset (IndD).**[3] We further provide an industrial dataset that records the behavioral data of millions of users. We collected interaction data with timestamps among 3072058 users and 200 services. Our task is to predict the next service the user would visit on a platform.

Considering consistency and comparability, for the public datasets ETT and PM2.5, we follow the same prepossessing steps mentioned in [33] and [12].

For the ETT dataset, we use the first 12-month data as the training set and use the following 4-month and the next following 4-month data as the corresponding validation and test set. We use the sequence length of 96 data points to predict the next 24 steps. For the PM2.5 dataset, we use 70% of the samples as the training set and 15% of the samples for validation and the remaining 15 % of the samples as the test dataset. We use the previous 24-hour data (i.e., sequence length 24) to predict the class of PM2.5 value at 8 am on the next day. For the IndD, since it is a large dataset, we use 1% of the samples as the validation set and 1% of the samples as the test set. The overall statistics of the datasets are summarized in Table 1.

Name	Label type	Sequence length	Prediction length	Sample numbers
SynD	Continuous value	100	1	2,924
ETT	Continuous value	96	24	8,640
PM2.5	Multi-class	24	1	1,215
IndD	Multi-class	300	1	14,049,979

Table 1: Statistics of datasets.

5.2 Comparison Methods

The baseline methods consist of two parts: (1) a typical RNN related method; and (2) five Transformer and its various variants that are introduced in a progressive manner.

- **TimeLSTM.** A conventional LSTM method equipped with time gates for modeling time intervals. [34]
- **RNN+Attention.** A multi-time attention network that learns embeddings of irregularly sampled time series. [25]
- **TiSASRec.** A state-of-the-art time interval aware self-attention method for sequential recommendation. [15]
- **Transformer w/o PE.** The classic Transformer without positional encoding.
- **Transformer+PE.** The classic Transformer with sinusoidal positional encoding [27].
- **Transformer+ME.** The Transformer with mercer time embedding [30].
- **Transformer+NsTKA.** Our proposed method.

We implemented our proposed method based on Keras[4] with the Adam optimizer. For the proposed NsTKA method, we treat the number of mixture Gaussian components m as a hyper-parameter and select from {3, 4, 5, 6, 7, 8}. For all compared methods including ours, we consider latent dimensions {8, 16, 32, 64, 128} and l_2 regularizer $\{0.0001, 0.001, 0.01, 0.1, 1\}$. All other hyperparameters and initialization strategies (e.g., learning rate, optimizer, dropout, maximum sequence length, maximum time interval, etc.) are similar to the suggested optimal configurations by the methods' authors.

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

[3] Note: The data set does not contain any Personal Identifiable Information (PII); The data set is desensitized and encrypted; Adequate data protection was carried out during the experiment to prevent the risk of data copy leakage, and the data set was destroyed after the experiment; The data set is only used for academic research, it does not represent any real business situation.

[4] https://github.com/alipay/nstka-kdd22

We use the validation set to tune hyper-parameters and terminate training if the validation performance has no improvement after 10 epochs. All performance is reported on the test set.

5.3 Evaluation Metrics

For the PM2.5 and IndD multi-class prediction tasks, we evaluate the model performance by classification accuracy. For the ETT and SynD regression tasks, we adopt the normalized mean squared error (NMSE) to measure the performance, in which the continuous-value data are normalized to zero-mean and unit deviation that they are comparable. Specifically, for the ETT dataset, since it is a multi-step prediction task, we report the averaged NMSE for the multi-step predicted results.

5.4 Performance Comparison (Q1)

Table 2 shows the experimental results. We can see that, compared with RNN-like methods, Transformer variations show a significant improvement. Through introducing PE for explicitly modeling position information, we observed a further improvement. This demonstrates that temporal correlation among events is critical in forecasting the following event. Replacing PE with ME, the continuous time-span information is taken into consideration so that the fine-grained temporal patterns can be captured. The proposed method, Transformer+NsTKA, outperforms competitors by a large margin on both the synthetic datasets and real-world datasets in most cases. Specifically, on different types of non-stationary datasets and a number of temporal datasets on different domains, Transformer+NsTKA consistently shows the best or the second-best performance, which indicates that the proposed method can capture non-stationary patterns under complex settings. Furthermore, on two of the synthetic datasets, Transformer + PE has a better performance than ours. It might due to the reason that the classic PE takes the format of predefined sine and cosine functions and our synthetic dataset is also composed of sinusoidal waves. With proper initialization, the Transformer + PE method can capture the dominating temporal patterns well. Nevertheless, in terms of the representation capability in a much wider range, the above results have well verified the necessity of encoding non-stationary characteristics.

To qualitatively verify the effectiveness of the proposed method, we plot fitted curves in Figure 3. In Figure 3, we compared the prediction on one of the synthetic signal between our proposed method and the one of Transformer + ME, which is a typical stationary time-encoding representation. It is readily observed that the proposed method can precisely fit the curve the best. Since these baseline methods are designed for stationary datasets, the models would try to fit the targeted curve with a stationary basis, resulting in a misalignment both on the frequency and amplitude. Specifically, we can see that curves fitted by these baseline methods cannot align well within valleys and peaks because valleys and peaks do not occur in a fixed period. A similar phenomenon can also be observed on the fitting of the amplitudes. This further demonstrates that the model capacity of these baseline methods is not good enough to fit the non-stationary pattern.

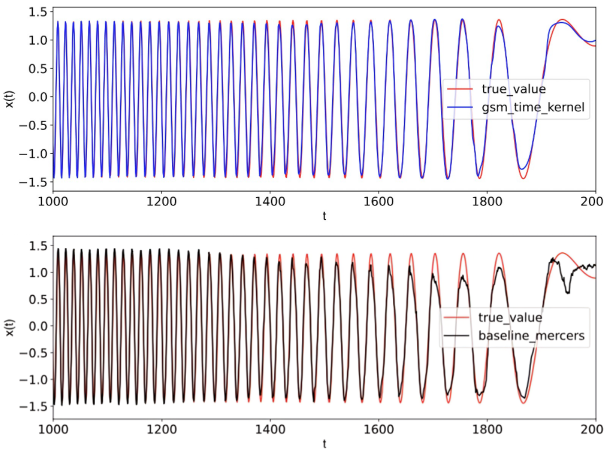

Figure 3: Fitted curves given by different methods. Upper: Transformer+NsTKA. Lower: Transformer+ME.

5.5 Ablation Study (Q2)

As mentioned in section 4.2, when the mean, length-scale and weight functions of the GSM kernel reduce to input-independent constants, the GSM kernel would reduce to a stationary SM kernel. To show the effectiveness of encoding non-stationarity through our proposed method, we firstly modify NsTKA by replacing the non-stationary kernel with a stationary kernel, while other components keep unchanged. Furthermore, the GSM kernel can also be considered as the product of three kernels, i.e., the linear kernel $\alpha_i(x)\alpha_i(x')$, the Gibbs kernel, and the cosine kernel, which are all dependent on the absolute time. By reducing each component into an x-independent term or completely removing the component, we analyze how each component contributes to the improvement.

5.5.1 Effectiveness of Non-stationary Kernel. First, we only replace the non-stationary kernel with a stationary one to verify the necessity of capturing the non-stationary pattern. From the results shown in Table 3, we can see that using a stationary kernel would lead to performance drop on all the datasets. This suggests that the non-stationarity is served as an important factor in the temporal event prediction, and our non-stationary kernel can successfully take advantage of this information to improve the performance.

5.5.2 Effectiveness of Each Kernel. Table 4 shows the experimental results with different combinations of kernels, where μ_i, l_i and α_i denote these kernels are independent of x. We can observe that:

- by singly removing $\alpha_i(x)$ which makes each mixture component equally weighted, a largest performance drop is observed, which suggests that different components serve as different roles in the model and they require careful weighting.
- removing x from the input of the three kernels all leads to the degradation of performance except for the μ_i case. On one hand, it suggests that x is of great importance on both the $l_i(x)$ and $\alpha_i(x)$ parts of the kernel, which represents the short- and long-range temporal dependency and component weights separately as described in section 4.2. And the temporal patterns regarding these components with regard to x are successfully extracted as well. Meanwhile, the slightly

	Datasets					
	NMSE				*Top-1 Accuracy*	
Methods	SynD-A	SynD-F	SynD-AF	ETT	PM2.5	IndD
TimeLSTM	0.0144	0.0482	0.0961	0.1218	0.4827	0.4207
RNN+Attention	0.0092	0.0534	0.0439	0.0521	0.4789	0.4008
TiSASRec	0.0056	0.1089	0.0410	0.0455	0.5057	0.3423
Transformer w/o PE	0.0133	0.0634	0.0619	0.0451	0.5172	0.3765
Transformer+PE	0.0056	**0.0212**	**0.0266**	0.0460	0.5211	0.4993
Transformer+ME	0.0140	0.0629	0.0680	0.0438	0.5326	0.5195
Transformer+NsTKA	**0.0054**	0.0392	0.0377	**0.0410**	**0.5460**	**0.5285**

Table 2: Results on one synthetic dataset and three real-world datasets. Bold values denotes the best results.

Methods	IndD Top-1 Accuracy	SynD NMSE
Transformer + NsTKA	0.5285	0.0377
Transformer + STKA	0.5212	0.0665

Table 3: Ablation study of the non-stationary part of the NsTKA.

contrary results for the μ_i case do not necessarily mean that the frequency components in the cosine term is lack of time-dependency, it could be that well capturing the time-varying pattern of the periodical correlation of the signal is relatively difficult during training. The improvement of representation capability of our proposed NsTKA method might be accompanied with certain decrease in robustness when dealing with certain datasets, which could be improved with more careful hyperparameter selection or more available data samples. To further improve the robustness and performance of the model is also a future topic of our research.

$\mu_i(x)$	$l_i(x)$	$\alpha_i(x)$	NMSE
✓	✓	✓	0.0377
✗	✓	✓	0.0377
✓	✗	✓	0.0418
✓	✓	✗	0.0834
μ_i	✓	✓	0.0355
✓	l_i	✓	0.0532
✓	✓	α_i	0.0680

Table 4: Results on the dataset SynD with different combinations of kernels.

5.6 Case Study (Q3)

To study how the proposed method performs on different scenarios, we create synthetic datasets with different levels of non-stationarity. Specifically, the coefficients α_i, β_i in Eq.14 and Eq.15 are changing with different speed with respect to time. Namely, two speed-coefficients $c_{\alpha_i}, c_{\beta_i}$ are further introduced. As a result, the new amplitude and frequency functions are $A_i(t) = A_{i,0}(1 + c_{\alpha_i}\alpha_i t/T)$ and $\omega_i(t) = \omega_{i,0} * exp(c_{\beta_i}\beta_i t/T)$, respectively. One thing to note is that, when the coefficients change in a negligible magnitude (i.e., $c_{\alpha_i} = 0, c_{\beta_i} = 0$), the synthetic signal is actually stationary. Intuitively, Figure 4 illustrated as an example of signals with different values of $c_{\alpha_i}, c_{\beta_i}$. Compared with the stationary signal, the amplitude and frequency of other two ones change over time to different degrees.

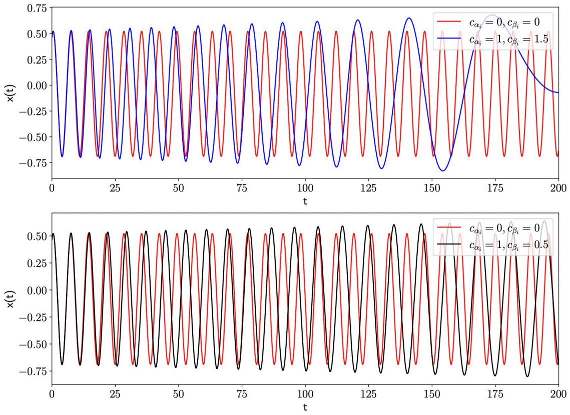

Figure 4: An example of signals with different values of $c_{\alpha_i}, c_{\beta_i}$.

	c_{α_i}	c_{β_i}	Transformer +NsTKA	Transformer +ME	gain
Stationary	0	0	0.032	0.0352	-9.0%
Frequency fluctuates	1	0.1	0.0568	0.1283	-55.7%
	1	0.5	0.0558	0.1053	-47.0%
	1	1	0.0490	0.1046	-53.2%
	1	1.1	0.0431	0.0926	-53.5%
	1	1.5	0.0388	0.1347	-71.2%
Amplitude fluctuates	0.1	1	0.0433	0.1188	-63.6%
	0.5	1	0.0400	0.1257	-68.2%
	1.1	1	0.0412	0.1345	-69.4%
	1.5	1	0.0325	0.1475	-78.0%

Table 5: Mean squared error of methods Transformer+NsTKA and Transformer+ME when fitting the signals with different non-stationary levels.

While keeping all the other parameters in Eq. 13 the same, Table 5 presents the relative gain between NsTKA and ME under the different values of $c_{\alpha_i}, c_{\beta_i}$. We can observe that:

- For a stationary dataset, NsTKA shows a similar performance with the related method, ME, which is derived from a stationary kernel. It suggests that NsTKA can also fit the dataset well on a degenerated setting, even though the kernel is non-stationary.
- For non-stationary datasets, as the increasing level of non-stationarity, NsTKA shows a more and more significant

improvement comparing with the baseline on both decaying scenarios, which demonstrates the powerful ability of NsTKA to delineate the non-stationary temporal patterns.

In summary, NsTKA is proved to be effective on both stationary and non-stationary datasets, which can be widely applied to different domains.

6 CONCLUSION

In this paper, by introducing the Generalized Spectral Mixture Kernel, and integrating it to the attention module, we mathematically reveal its representation capability in terms of the non-stationary temporal-correlation. Then, a new attention structure is devised for input sequences of neural networks. Finally, exhaustive experiments are conducted to present its competitive performance against related work, followed by an ablation experiment which shows its effectiveness on both stationary and non-stationary temporal correlations. We would further improve the robustness and performance of the method in our future work. Furthermore, we plan to extend our method to the reinforcement learning domain, as the ability to predict future off-policy performance in non-stationary environments is common and critical in real-world scenarios.

REFERENCES

[1] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 65–74.

[2] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–66.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Jared Davis, Tamás Sarlós, David Belanger, Lucy J. Colwell, and Adrian Weller. 2020. Masked Language Modeling for Proteins via Linearly Scalable Long-Context Transformers. *CoRR* abs/2006.03555 (2020).

[5] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. 2978–2988.

[6] Daniele Durante and David B Dunson. 2014. Bayesian dynamic financial networks with time-varying predictors. *Statistics & Probability Letters* 93 (2014), 19–26.

[7] Farzad Eskandanian and Bamshad Mobasher. 2019. Modeling the Dynamics of User Preferences for Sequence-Aware Recommendation Using Hidden Markov Models. In *The Thirty-Second International Flairs Conference*. 425–430.

[8] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 1243–1252.

[9] Marc G Genton. 2001. Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research* 2, Dec (2001), 299–312.

[10] Mark N Gibbs. 1997. *Bayesian Gaussian processes for regression and classification.* Ph. D. Dissertation. University of Cambridge.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. 2021. Explainable Multivariate Time Series Classification: A Deep Neural Network Which Learns to Attend to Important Variables As Well As Time Intervals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 607–615.

[13] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 447–455.

[14] Mingi Ji, Weonyoung Joo, Kyungwoo Song, Yoon-Yeong Kim, and Il-Chul Moon. 2020. Sequential recommendation with relation-aware kernelized self-attention. In *Proceedings of the AAAI conference on artificial intelligence*. 4304–4311.

[15] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. 322–330.

[16] Xuan Liang, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. 2015. Assessing Beijing's PM2. 5 pollution: severity, weather impact, APEC and winter heating. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471, 2182 (2015), 20150257.

[17] Manxia Liu, Arjen Hommersom, Maarten van der Heijden, and Peter JF Lucas. 2017. Hybrid time Bayesian networks. *International Journal of Approximate Reasoning* 80 (2017), 460–474.

[18] Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. 2021. Stable, fast and accurate: Kernelized attention with relative positional encoding. In *Advances in Neural Information Processing Systems*. 22795–22807.

[19] Simone Marini, Emanuele Trifoglio, Nicola Barbarini, Francesco Sambo, Barbara Di Camillo, Alberto Malovini, Marco Manfrini, Claudio Cobelli, and Riccardo Bellazzi. 2015. A Dynamic Bayesian Network model for long-term simulation of clinical complications in type 1 diabetes. *Journal of biomedical informatics* 57 (2015), 369–376.

[20] Hongyuan Mei and Jason Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 6754–6764.

[21] Subhabrata Mukherjee, Hemank Lamba, and Gerhard Weikum. 2017. Item recommendation with evolving user preferences and experience. *arXiv preprint arXiv:1705.02519* (2017).

[22] Sami Remes, Markus Heinonen, and Samuel Kaski. 2017. Non-stationary spectral kernels. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 4645–4654.

[23] Yves-Laurent Kom Samo and Stephen Roberts. 2015. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236* (2015).

[24] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*. 464–468.

[25] Satya Narayan Shukla and Benjamin M. Marlin. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[26] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. 4343–4352.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[28] Christopher K Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT press Cambridge, MA.

[29] Andrew Wilson and Ryan Adams. 2013. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*. 1067–1075.

[30] Da Xu, Chuanwei Ruan, Sushant Kumar, Evren Korpeoglu, and Kannan Achan. 2019. Self-attention with functional time representation learning. In *Advances in Neural Information Processing Systems*. 15915–15925.

[31] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. 2020. Do RNN and LSTM have Long Memory?. In *Proceedings of the 37th International Conference on Machine Learning*. 11365–11375.

[32] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. 1513–1522.

[33] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*.

[34] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. 3602–3608.