Sub-Adjacent Transformer: Improving Time Series Anomaly Detection with Reconstruction Error from Sub-Adjacent Neighborhoods

Wenzhen Yue¹, Xianghua Ying¹*, Ruohao Guo¹, DongDong Chen², Ji Shi¹, Bowei Xing¹, Yuqing Zhu¹ and Taiyan Chen¹
¹National Key Laboratory of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University
²Microsoft Cloud + AI
³Tsinghua University
yuewenzhen@stu.pku.edu.cn, xhying@pku.edu.cn, dochen@microsoft.com

Abstract

In this paper, we present the Sub-Adjacent Transformer with a novel attention mechanism for unsupervised time series anomaly detection. Unlike previous approaches that rely on all the points within some neighborhood for time point reconstruction, our method restricts the attention to regions not immediately adjacent to the target points, termed sub-adjacent neighborhoods. Our key observation is that owing to the rarity of anomalies, they typically exhibit more pronounced differences from their sub-adjacent neighborhoods than from their immediate vicinities. By focusing the attention on the sub-adjacent areas, we make the reconstruction of anomalies more challenging, thereby enhancing their detectability. Technically, our approach concentrates attention on the non-diagonal areas of the attention matrix by enlarging the corresponding elements in the training stage. To facilitate the implementation of the desired attention matrix pattern, we adopt linear attention because of its flexibility and adaptability. Moreover, a learnable mapping function is proposed to improve the performance of linear attention. Empirically, the Sub-Adjacent Transformer achieves state-of-the-art performance across six real-world anomaly detection benchmarks, covering diverse fields such as server monitoring, space exploration, and water treatment.

1 Introduction

In modern industrial systems such as data centers and smart factories, numerous sensors consistently produce significant volumes of measurements [Zhou et al., 2019; Shen et al., 2020]. To effectively monitor the systems’ real-time conditions and avoid potential losses, it is crucial to identify anomalies in the multivariate time series [Pereira and Silva-veira, 2019; Chen, 2018]. This problem is known as time-series anomaly detection [Blázquez-García et al., 2021]. Comprehensive surveys can be found in the literature [Blázquez-García et al., 2021; Schölkopf et al., 2000; Gupta et al., 2013].

*Corresponding Author

Effective and robust time series anomaly detection remains a challenging and open problem [Zhao et al., 2022]. Real-world time series usually exhibit nonlinear dependencies and intricate interactions among time points. Besides, the vast scale of data makes the labeling of these anomalies time-consuming and costly in practice [Lai et al., 2023]. Therefore, time series anomaly detection is usually conducted in an unsupervised manner [Xu et al., 2022], which is also the focus of this paper.

Before the era of deep learning, many classic anomaly detection methods were proposed. These include the density-estimation method proposed in [Breunig et al., 2000], the clustering-based method presented in one-class SVM [Schölkopf et al., 2001], and a series of SVDD works [Tax and Duin, 2004; Liu et al., 2013], along with graph-based methods [Cheng et al., 2008]. Traditional methods, which use handcrafted features, struggle to accurately express the relationship among time points and often suffer from poor generalization. Deep learning-based techniques have witnessed a diversity of methods during the recent decade. Broadly, they can be categorized into three main categories: Reconstruction-based methods reconstruct the time series and compare it with the original data [Park et al., 2018;
prediction-based methods predict future points for comparison with actual data [Deng and Hooi, 2021; Hundman et al., 2018a; Ding et al., 2019; Lai et al., 2023]. Dissimilarity-based methods emphasize the discrepancy representation between normal and abnormal points in various domains [Shen et al., 2020; Sun et al., 2021; Xu et al., 2022]. It is noteworthy that these methods often combine multiple approaches to enhance performance. For example, TranAD [Tuli et al., 2022] integrates an integrated reconstruction error and discriminator loss; while [Lai et al., 2023] uses prediction results as the nominal data and combines them with reconstruction results.

In this paper, we focus on the attention matrix within the Transformer module. Transformers [Vaswani et al., 2017] have achieved great success in natural language processing [Brown et al., 2020], computer vision [Liu et al., 2021] and time series [Xu et al., 2022] in recent years. Previous work in [Xu et al., 2022] primarily concentrated on the dissimilarities in attention distributions. Our paper takes a different approach and introduces a simple yet effective attention learning paradigm. Our fundamental assumption is that anomalies are less related to their non-immediate neighborhoods when compared with normal points. Therefore, focusing exclusively on these non-immediate neighborhoods is likely to result in larger reconstruction errors for anomalies. Based on this assumption, our study introduces two key concepts: sub-adjacent neighborhoods and sub-adjacent attention contributions. As shown in Figure 1, sub-adjacent neighborhoods indicate the areas not immediately adjacent to the target point, and the sub-adjacent attention contribution is defined as the sum of particular non-diagonal elements in the corresponding column of the attention matrix. These concepts are integrated with reconstruction loss, forming the cornerstone of our anomaly detection strategy. Furthermore, we observe that the traditional Softmax operation used in standard self-attention impedes the formation of the desired attention matrix, where the predefined non-diagonal stripes are dominant. In response, we adopt linear attention [Shen et al., 2021; Han et al., 2023] for its greater flexibility in attention matrix configurations. We also tailor the mapping function within this framework, using learnable parameters to enhance performance. The main contributions of this paper are summarized as follows.

- We propose a novel attention learning regime based on the sub-adjacent neighborhoods and attention contributions. Specifically, a new attention matrix pattern is designed to enhance discrimination between anomalies and normal points.
- Furthermore, we leverage the linear attention to achieve the desired attention pattern. To the best of our knowledge, this is the first introduction of linear attention with a learnable mapping function to time series anomaly detection.
- Extensive experiments show that the proposed Sub-Adjacent Transformer delivers state-of-the-art (SOTA) performance across six real-world benchmarks and one synthetic benchmark.

2 Related Works

### Time Series Anomaly detection

In recent years, graph neural networks and self-attention have been explored in time series anomaly detection [Deng and Hooi, 2021; Zhao et al., 2020; Boniol and Palpanas, 2022; Xu et al., 2022; Lai et al., 2023; Wu et al., 2023]. The most related work to ours is Anomaly Transformer [Xu et al., 2022], which also exploits the attention information. The discrepancy between the learned Gaussian distribution and the actual distribution is used to distinguish abnormal points from normal ones. A somewhat complicated min-max training strategy is adopted to train the model. Generally, Anomaly Transformer is complex and somewhat indirect. Our proposed method is more straightforward and exhibits improved performance.

### Linear Attention

Compared with vanilla self-attention, linear attention enjoys more flexibility and lighter computation burden. Existing linear attention methods can be divided into three categories: pattern based methods, kernel based methods and mapping based methods. We mainly focus on mapping based methods in this paper. Efficient attention [Shen et al., 2021] applies Softmax function to Q (in a row-wise manner) and K (in a column-wise manner) to ensure each row of $QK^T$ sums up to 1. Hydra attention [Bolya et al., 2022] studies the special case where attention heads are as many as feature dimension and Q and K are normalized so that $\|Q_i\|$ and $\|K_i\|$ share the same value. EfficientVit [Cai et al., 2023] uses the simple ReLU function as the mapping function. Flatten Transformer [Han et al., 2023] proposes the power function as the mapping function to improve the focus capability. In this paper, we apply Softmax to matrices Q and K both in a row-wise manner to approximate the focus property of the vanilla self-attention. Empirical experiments verify its performance superiority to other mapping methods in time series anomaly detection.

3 Methods

### 3.1 Problem Formulation

Let $X = \{x_1, \ldots, x_T\} \in \mathbb{R}^{T \times D}$ denote a set of time series with $x_t \in \mathbb{R}^D$, where $T$ is the time steps and $D$ is the number of channels. The label vector $y = \{y_1, \ldots, y_T\}$ indicates whether the corresponding time stamp is normal ($y_t = 0$) or abnormal ($y_t = 1$). Our task is to determine anomaly labels for all time points $\hat{y} = \{\hat{y}_1, \ldots, \hat{y}_T\}$, where $\hat{y}_t \in \{0, 1\}$, to match the ground truth $y$ as much as possible. Following the practice of [Xu et al., 2022; Schmidl et al., 2022; Tuli et al., 2022; Lai et al., 2023], we mainly focus on the F1 score with point adjustment.

### 3.2 Sub-Adjacent Neighborhoods

Time points usually have stronger connections with their neighbors and fewer connections with distant points. This characteristic is more pronounced for anomalies [Xu et al., 2022]. As shown in Figure 1, if we rely solely on sub-adjacent neighborhoods to reconstruct time points, the reconstruction errors of anomalies will become more pronounced, thereby enhancing their distinguishability. This is the core idea of the Sub-Adjacent Transformer.
In this paper, we define the sub-adjacent neighborhoods as the region where the distance to the target point is between $K_1$ and $K_2$. $K_1$ and $K_2$ are the pre-defined area bounds and satisfy $K_2 \geq K_1 > 0$. The highlighted red area in Figure 1 (b) show the sub-adjacent neighborhoods of the point marked with the red circle.

We now apply our thought to the attention matrix. The sub-adjacent neighborhoods are represented by the highlighted stripes in the attention matrix, as depicted in Figure 2. To focus attention on these stripes, we introduce the concept of attention contribution. This concept involves viewing the columns of the attention matrix as each point’s contribution to others within the same window. Let $A_{ij}$ denote the element in the $i$-th row and $j$-th column of the attention matrix $A$. The value of $A_{ij}$ reflects the extent of point $i$’s contribution to point $j$; the larger $A_{ij}$ is, the more significant the contribution. We define the sub-adjacent attention contribution of each point as the aggregate of values within the pre-defined sub-adjacent span in the corresponding column, which is

$$SACon(A) = \left[ SACon(A_{i,:}) \right]_{i=0,\ldots,\text{win}_{\text{size}}-1}$$

$$SACon(A_{i,:}) = \sum_{|j-i|+K_1}^{K_2} A_{ij}, \quad 0 \leq j < \text{win}_{\text{size}}$$

where subscript $(::i)$ denotes the $i$th column of the corresponding matrix, and win_size is the window size.

The sub-adjacent attention contribution plays a pivotal role in two aspects. Firstly, it steers the focus of attention towards the sub-adjacent neighborhoods by being integrated into the loss function (Eq. 5). This is achieved by increasing $SACon(A)$ for all points during the training stage. Secondly, it assists in anomaly detection. This is due to its incorporation into the anomaly score calculation, where anomalies typically show a lower attention contribution than normal points.

Moreover, the number of the highlighted cells in Figure 2 is lower for the marginal points (where $i < K_2$ or $i > \text{win}_{\text{size}} - K_2$), leading to imbalance among points. It is mainly because the $j$ in Eq. 1 is bounded by $0 \leq j < \text{win}_{\text{size}}$. To break through this limitation, one plausible way is to use a circular shift function to calculate $A_{ji}$ in Eq. 1:

$$A_{ji} = \langle \text{Roll}(Q,i-j) \rangle_{1,\ldots,K_i}$$

where $Q,K \in \mathbb{R}^{\text{win}_{\text{size}} \times \text{model}}$ are the query and key matrix, respectively, $\langle \cdot \rangle$ represents the inner product of two vectors, Roll$(Q,i-j)$ cyclically shift matrix $Q$ by $i-j$ along the first dimension. The $j$ values in $A_{ji}$ of Eq. 2 can satisfy the conditions $j < 0$ or $j \geq \text{win}_{\text{size}}$ and ensure that the number of $j$s for each $i$ is the same. Actually, the cyclic operation in Eq. 2 is equivalent to setting $SACon(A_{i,:})$ in Eq. 1 as:

$$SACon(A_{i,:}) = \sum_{|j-i|+K_1}^{K_2} A_{ji}, \quad |j| < \text{win}_{\text{size}}$$

Linear Attention. Vanilla self-attention employs the Softmax function on a row-wise basis within the attention matrix, leading to competition among values in the same row, which results in two extra side stripes in the attention matrix. In contrast, linear attention does not face such constraints. As shown in Figure 4(b), the use of Eq. 3 results in two extra side stripes in the attention matrix.

Figure 3: Attention matrices obtained using (a) vanilla self-attention and (b) the linear attention with the proposed mapping function. The SMAP dataset [Hundman et al., 2018b] and the proposed sub-adjacent neighborhoods are used.
\[ \Phi(\cdot) = \text{Softmax}_{\text{row}}(\cdot/\tau) \] (4)

where \( \text{Softmax} \) is applied row-wise to the input matrix, and \( \tau \) is a learnable parameter to adjust dynamically the \( \text{Softmax} \) temperature. Note that in Eq. 4, the \( \text{Softmax} \) function is applied to the matrices \( Q \) and \( K \), rather than directly to the attention matrix, as is the case with vanilla self-attention. This distinction is fundamental to the increased flexibility of the attention matrix. Moreover, we set all negative values in the matrices \( Q \) and \( K \) to a large negative number, such as \(-100\), to ensure that these values are close to 0 after applying the mapping function. Various mapping functions have been explored in prior research, including the power function \([\text{Han et al., 2023}]\), column-wise \( \text{Softmax} \) \([\text{Shen et al., 2021}]\), \( \text{ReLU} \) \([\text{Cai et al., 2023}]\), and elu function \([\text{Katharopoulos et al., 2020}]\). Our empirical experiments demonstrate the effectiveness of our mapping function in time series anomaly detection, as detailed in Table 5.

### 3.3 Loss Function and Anomaly Score

**Loss Function.** Reconstruction loss is fundamental in unsupervised time series anomaly detection. As aforementioned, we also introduce the sub-adjacent attention contribution into the loss function, which guides the model to focus on the sub-adjacent neighborhoods. By integrating these two losses, the loss function for the input series \( X \in \mathbb{R}^{T \times D} \) is formulated as follows:

\[ L_{\text{Total}}(X, \hat{X}, A) = L_{\text{rec}} + \lambda \cdot L_{\text{attn}} \]

\[ = \|X - \hat{X}\|_F^2 - \lambda \cdot \|\text{SACon}(A)\|_1 \] (5)

where \( L_{\text{rec}} \) is the reconstruction loss and \( L_{\text{attn}} \) is the attention loss, \( \lambda > 0 \) is the weight to trade off the two terms. \( \| \cdot \|_F \) and \( \| \cdot \|_1 \) in Eq. 5 are the Frobenius and k-norm, respectively. \( \hat{X} \in \mathbb{R}^{T \times D} \) denotes the reconstructed \( X \).

**Anomaly Score.** To identify anomalies, we combine the sub-adjacent attention contribution with the reconstruction errors. Consistent with with the approach in \([\text{Xu et al., 2022}]\), the \( \text{Softmax} \) function is applied to \(-\text{SACon}(A)\) to highlight the anomalies with less attention contribution. Subsequently, we perform an element-wise multiplication of the attention results and reconstruction errors. Finally, the anomaly score for each point can be expressed as

\[ \text{AnomalyScore}(X) = \text{Softmax}(-\text{SACon}(A)) \]

\[ \odot \left[ \sum_{i=1}^{T} \|X_{i,:} - \hat{X}_{i,:}\|_F^2 \right]_{i=1,\ldots,T} \] (6)

where \( \odot \) is the element-wise multiplication. Typically, the anomalies exhibit less attention contribution and higher reconstruction errors, leading to larger anomaly scores.

**Dynamic Gaussian Scoring.** Following the practice of \([\text{Schmidl et al., 2022}]\), we fit a dynamic Gaussian distribution to anomaly scores obtained by Eq. 6 and design a score based on the fitted distribution. Let \( \mu_i \) and \( \sigma_i \) denote the dynamic mean and standard variance, respectively, which can be computed in the way of sliding windows. Then the final score with dynamic Gaussian fitting can be computed via

\[ \text{DyAnoSco}_i = -\log \left( 1 - \text{cdf} \left( \frac{\text{AnomalyScore}_i - \mu_i}{\sigma_i^2} \right) \right) \] (7)

where \( \text{cdf} \) is the cumulative distribution function of the standard Gaussian distribution \( N(0, 1) \).

### 4 Experiments

#### 4.1 Datasets

We evaluate the Sub-Adjacent Transformer on the following datasets, whose statistics are summarized in Table 1.

- **SWaT (Secure Water Treatment)** \([\text{Mathur and Tippenhauer, 2016}]\) is collected continuously over 11 days from 51 sensors located at a water treatment plant.
- **WADI (Water Distribution)** \([\text{Ahmed et al., 2017}]\) is acquired from 123 sensors of a reduced water distribution system for 16 days.
- **PSM (Pooled Server Metrics)** \([\text{Abdulaal et al., 2021}]\) is collected from multiple servers at eBay with 26 dimensions.
- **MLS (Mars Science Laboratory rover)** and **SMAP (Soil Moisture Active Passive satellite)** are datasets released by NASA with 55 and 25 dimensions respectively.
- **SMD (Server Machine Dataset)** \([\text{Su et al., 2019}]\) is collected from a large compute cluster, consisting of 5 weeks of data from 28 server machines with 38 sensors.
- **NeurIPS-TS (NeurIPS 2021 Time Series Benchmark)** is a synthetic dataset proposed by \([\text{Lai et al., 2021}]\) and includes 5 kinds of anomalies that cover point- and pattern-wise behaviors: global (point), contextual (point), shapelet (pattern), seasonal (pattern), and trend (pattern).

#### 4.2 Implementation Details

Following the common practice, we adopt a non-overlapping sliding window mechanism to obtain a series of sub-series. The sliding window size is set as 100 without particular statements. \( K_1 \) and \( K_2 \) are set as 20 and 30, respectively. The choices of these parameters will be discussed later in ablation studies. The points are judged to be anomalies if their anomaly scores (Eq. 7) are larger than a certain threshold \( \delta \).

In this study, following the practice of paper \([\text{Lai et al., 2023; Xu et al., 2022}]\), the thresholds are chosen to output the best F1 scores. The widely-used point adjustment strategy \([\text{Xu et al., 2022; Tuli et al., 2022; Garg et al., 2022}]\)
Table 2: Quantitative results for various anomaly detection methods in the six real-world datasets. AUC means area under the ROC curve. The largest and second-largest values are highlighted with bold text and underlined text, respectively. The values in this table are as % for ease of display. †: For multi-entity datasets, we use a single model to train and test all entities together, posing additional challenges.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>F1</td>
<td>AUC</td>
<td>F1</td>
<td>AUC</td>
<td>F1</td>
<td>AUC</td>
<td>F1</td>
<td>AUC</td>
</tr>
<tr>
<td>DAGMM [2018]</td>
<td>64.3</td>
<td>70.5</td>
<td>72.2</td>
<td>74.2</td>
<td>85.5</td>
<td>86.2</td>
<td>77.5</td>
<td>65.4</td>
<td>92.4</td>
</tr>
<tr>
<td>LSTM-VAE [2018]</td>
<td>51.8</td>
<td>58.4</td>
<td>60.2</td>
<td>62.5</td>
<td>76.3</td>
<td>77.5</td>
<td>65.4</td>
<td>69.5</td>
<td>86.6</td>
</tr>
</tbody>
</table>

Table 3: Anomaly detection performance in the synthetic dataset NeurIPS-TS. AUC means area under the ROC curve. The largest and second-largest values are highlighted with bold text and underlined text, respectively. The values in this table are presented in percentages.

Wu et al., 2023] is adopted. Note that point adjustment has its practicality: one detected anomaly point will guide system administrators to identify the entire anomalous segment. The Sub-Adjacent Transformer contains 3 layers. Specifically, we set the hidden dimension \( d_{\text{model}} \) as 512, and the head number as 8. The hyper-parameter \( \lambda \) as 10 in Eq. 5 to balance recognition loss and attention loss. The Adam optimizer [Kingma and Ba, 2015] is used with an initial learning rate of \( 10^{-4} \). Following [Xu et al., 2022], the training is early stopped within 10 epochs with the batch size of 128. Experiments are conducted using PyTorch and one NVIDIA RTX A6000 GPU.

4.3 Main Results

**Real-World Datasets.** In Table 2, we report AUC and F1 metrics for various methods on six real-world datasets with 15 competitive baselines. The Sub-Adjacent Transformer consistently achieves state-of-the-art performance across all benchmarks, with an average improvement of 1.7 and 0.4 percentage points in F1 score over the current SOTA for single-entity and multi-entity datasets, respectively. Note that for multi-entity datasets, we combine all entities and train them together using a single model, whereas some methods, such as simple heuristics [Garg et al., 2022] and NPSR [Lai et al., 2023], train each entity separately and average the results. Even though entity-to-entity variations are large (especially MSL and SMD datasets), the Sub-Adjacent Transformer still achieves better anomaly detection results using one model than the multi-model counterpart, showing the advantage of the proposed attention mechanism. We have retained all dimensions of the original datasets, even though some of them are constant and do not play a role. Additionally, we also report F1 results without using point-adjustment in the supplementary material, where the Sub-Adjacent Transformer also exhibits competitive performance.

**Synthetic Dataset.** NeurIPS-TS is a challenging synthetic dataset. We generate the NeurIPS-TS dataset using the source codes released by [Lai et al., 2021], and it includes 5 anomalous types covering both the point-wise (global and contextual) and pattern-wise (shapelet, seasonal and trend) anomalies. As shown in Table 3, the Sub-Adjacent Trans-
Table 4: Performance comparison with different $K_1$ and $K_2$ settings for real-world datasets. The largest value for each dataset is emphasized in bold, while the second largest value is underlined. The values are marked in red if the value is not less than SOTA.

<table>
<thead>
<tr>
<th>$K_1 : K_2$</th>
<th>SWaT</th>
<th>WADI</th>
<th>PSM</th>
<th>MSL</th>
<th>SMAP</th>
<th>SMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 : 0</td>
<td>94.8</td>
<td>93.5</td>
<td>95.3</td>
<td>91.5</td>
<td>92.4</td>
<td>93.8</td>
</tr>
<tr>
<td>0 : 5</td>
<td>96.7</td>
<td>95.9</td>
<td>97.6</td>
<td>93.6</td>
<td>93.7</td>
<td>94.1</td>
</tr>
<tr>
<td>0 : 10</td>
<td>98.5</td>
<td>98.2</td>
<td>98.7</td>
<td>94.4</td>
<td>96.9</td>
<td>95.9</td>
</tr>
<tr>
<td>10 : 20</td>
<td>97.6</td>
<td>98.1</td>
<td>98.1</td>
<td>96.5</td>
<td>97.8</td>
<td>94.4</td>
</tr>
<tr>
<td>20 : 30</td>
<td>99.0</td>
<td>99.3</td>
<td>98.9</td>
<td>96.7</td>
<td>98.2</td>
<td>97.7</td>
</tr>
<tr>
<td>30 : 40</td>
<td>97.6</td>
<td>94.8</td>
<td>97.8</td>
<td>95.6</td>
<td>97.5</td>
<td>94.0</td>
</tr>
<tr>
<td>10 : 30</td>
<td>97.8</td>
<td>97.8</td>
<td>98.8</td>
<td>95.0</td>
<td>97.4</td>
<td>94.5</td>
</tr>
<tr>
<td>20 : 40</td>
<td>97.9</td>
<td>98.0</td>
<td>93.5</td>
<td>95.7</td>
<td>95.6</td>
<td>94.4</td>
</tr>
<tr>
<td>10 : 40</td>
<td>98.0</td>
<td>97.7</td>
<td>98.7</td>
<td>96.1</td>
<td>97.5</td>
<td>97.8</td>
</tr>
</tbody>
</table>

Table 5 shows that larger $K_1$ and $K_2$ values (the second and third rows in Table 4) perform worse than the baseline setting of $K_1 = 0$ and $K_2 = 0$. This highlights the importance of sub-adjacent attention.

4.4 Ablation Studies

Vanilla and Linear Transformer. We compare vanilla self-attention and linear attention with different mapping functions using our method. As shown in Table 5, the proposed Softmax-based mapping function with learnable parameters outperforms other mapping functions and vanilla self-attention, with a maximum F1 score increase of 1.5 percentage points.

Choices of $K_1$ and $K_2$. Choices of $K_1$ and $K_2$ directly affect the performance of the Sub-Adjacent Transformer. We evaluate the impact of $K_1$ and $K_2$ on the window size being 100. The configuration of $K_1 = 20$ and $K_2 = 30$ typically yields the best performance (with the exception of the SMD dataset), while the performance of other settings is also satisfactory, with nearly half of the values exceeding the current SOTA. When the sub-adjacent span is getting close to the diagonal (the second and third rows in Table 4), the performance worsens. We also explore the extreme case where $K_1 = K_2 = 0$. In that case, the performance drops significantly, highlighting the importance of sub-adjacent attention.

Module Ablation. We evaluate the contribution of linear attention module and dynamic scoring module in Table 6. For the Sub-Adjacent Transformer, the baseline means the proposed sub-adjacent attention mechanism accompanied with vanilla self-attention and anomaly score (Eq. 6). Then linear attention and Gaussian dynamic scoring (Eq. 7) are introduced in turn. As shown in Table 6, compared with the baseline, linear attention brings +1.0 improvement on average over six datasets and Gaussian dynamic scoring brings another +0.2 improvement, verifying the effectiveness of the proposed modules.

Furthermore, we also apply the two modules to Anomaly Transformer [Xu et al., 2022]. As we can see from Table 6, the proposed linear attention model provides +1.2 percentage points performance gain averagely, and Gaussian dynamic scoring provides another +0.1 gain. Moreover, as shown in Table 6, without the support of linear attention and dynamic Gaussian score, our method (baseline) still outper-
forms the Anomaly Transformer by an average of +1.8 percentage points, thereby validating the efficacy of our sub-adjacent attention design.

**Choices of Window Size.** Table 7 gives F1 values with different window sizes for six datasets. One can see that our model exhibits good robustness to the window size. The optimal performance is typically achieved with a window size of 100, the only exception being the SWaT dataset.

**Choices of Parameter λ.** Table 8 compares the performance for different values of λ. The best results are achieved when λ is set to 10, while the second-best results occur at different values of λ for different datasets. It is noteworthy that without the inclusion of attention loss (λ = 0), performance drops drastically, affirming the effectiveness of our sub-adjacent attention design.

**5 Conclusion**

In this paper, we present the Sub-Adjacent Transformer, a novel paradigm for utilizing attention in time series anomaly detection. Our method distinctively combines sub-adjacent attention contribution, linear attention and reconstruction error to effectively detect anomalies, thereby enhancing the efficacy of anomaly detection. It offers a novel perspective on the utilization of attention in this domain. Without bells and whistles, our model demonstrates superior performance across common benchmarks. We hope that the Sub-Adjacent Transformer could act as a baseline framework for the future works in time series anomaly detection.

**Acknowledgments**

This work was supported by the National Science Foundation of China (NSFC) (No. 62371009 and No. 61971008).
References


[Gupta et al., 2013] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. IEEE Transactions on Knowledge and data Engineering, 26(9):2250–2267, 2013.

[Han et al., 2023] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer used focused linear attention. In ICCV, pages 5961–5971, 2023.


