# STREAMING STROKE CLASSIFICATION OF ONLINE HANDWRITING

*Jing-Yu Liu*[1,2], *Yan-Ming Zhang*[1], *Fei Yin*[1], *Cheng-Lin Liu*[1,2]

[1]Nation Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences

## ABSTRACT

Stroke classification for online handwriting aims at providing each stroke with a semantic label so as to fulfill handwriting segmentation. This task has attracted considerable attention due to its significance in online handwriting analysis. Existing methods are designed for the static situation, where stroke classification is conducted on the completion of handwriting. With the popularity of pad devices and electronic whiteboards, streaming stroke classification becomes increasingly important for instant handwriting processing and feedback. However, streaming classification is much more challenging due to the lack of contextual information and is underexplored in the past. In this paper, we propose Multiple Stroke State Transformer (MSST), a novel framework to enable simultaneous real-time classification and modifiability of previous predictions. Particularly, we set multiple states with duration for each stroke and then divide all states into chunks to perform message passing by Transformer. Experiments on handwritten documents and diagrams demonstrate the superiority of our method.

***Index Terms***— streaming stroke classification, online handwriting analysis, Multiple Stroke State Transformer

## 1. INTRODUCTION

In recent years, the growing use of smart devices largely facilitates the creation of online handwritten documents, which consist of strokes following the writing order. Compared to offline images composed of pixels, online documents are analyzed by considering strokes as basic units. Stroke classification, aiming at providing each stroke a semantic label such as Text/Math/Figure/Table, lays the foundation for online document analysis. Over past decades, many methods have been proposed for the task and made great achievements. Probabilistic graphical models [1, 2, 3] are used to model the temporal stroke interactions. Recurrent neural networks [4, 5, 6] make use of stroke's trajectory to achieve stroke representation learning and contextual relation modeling. Graph neural networks [7, 8, 9] are employed for stroke classification due to their excellent performance on structural data.

However, all these methods are designed for static situations. Specifically, since they make use of global contextual information, they have to wait for the completion of the whole document before classification. For instant and user-friendly processing, the more efficient way is making prediction in real time, namely in the streaming mode. Because streaming classification can only obtain information from strokes written currently and before, the accuracy is limited. As writing continues, written strokes can see their following strokes and receive more context information, so the modification of written strokes' labels should be considered, as shown in Fig. 1.

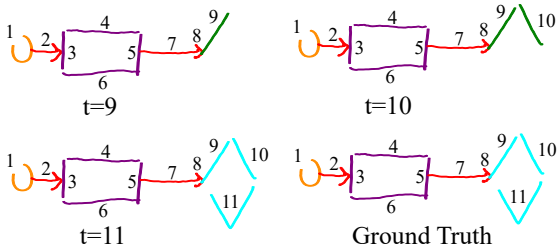


**Fig. 1**. Visualization of streaming stroke classification. Strokes $s_9$ and $s_{10}$ get their prediction at stroke step $t = 9$ and $t = 10$, but correct the prediction at $t = 11$. Different colors denote different classes.

Recently, chunk-based Transformer methods [10, 11, 12, 13, 14, 15, 16] have achieved promising results in streaming speech recognition by using truncated history and limited future information. For example, the idea of Transformer-XL [17] and chunk-wise process are combined to design a streamable model [14]. Some works [10, 12, 13] focus on chunks with variable size, while AM-TRF [15] and Emformer [16] employ memory banks to store the history information of processed chunks.

However, the above methods can not be directly employed in streaming stroke classification because they do not consider real-time inference and modifiability of labels simultaneously. In this paper, based on chunk-based methods, we propose Multiple Stroke States Transformer (MSST) to solve this problem. Each stroke is represented by several states. The first stroke state takes responsibility for real-time classification while others are in charge of label modification. Each state is attached a duration to reduce computational cost and strengthen history information utilization. We con-

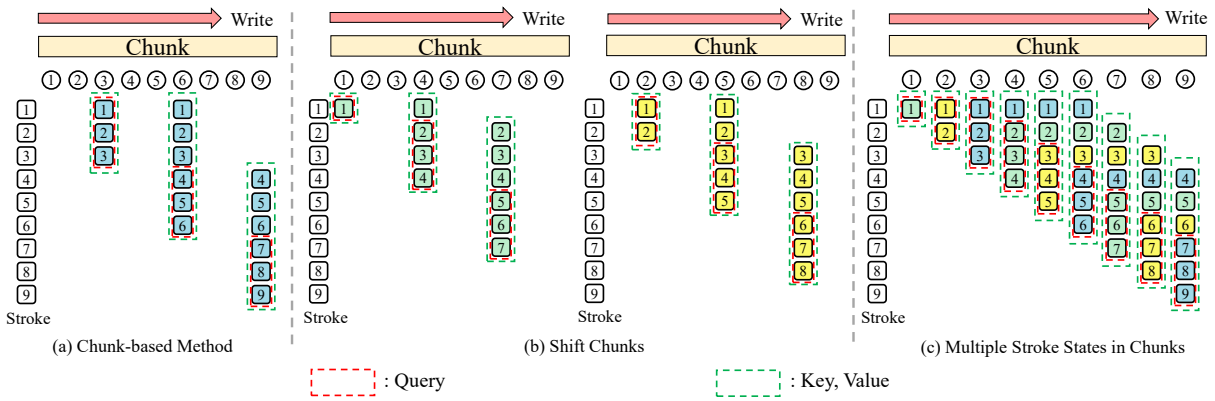


**Fig. 2**. Visualization of chunk-based Transformer in one layer and multiple stroke states. Given a document with $T = 9$ strokes, we set chunk size $n = 3$ and history size $m = 3$. Transformer is conducted in all chunks simultaneously. Strokes in red dashed boxes are Queries and in green dashed boxes are Keys and Values. Each stroke in different colors has different representations.

ducted experiments on online handwritten document dataset IAMonDo [18] and diagram dataset FC [19], and the results demonstrate the superiority of our method.

## 2. METHOD

### 2.1. Problem Definition

We are given an online handwriting document composed of ordered strokes $\boldsymbol{s} = \{s_1, \ldots, s_T\}$, where $T$ is the number of strokes. Each stroke $s_t$ is depicted by a sequence of trajectory points and has a corresponding label $y_t \in \{1, \ldots, C\}$, where $C$ is the number of semantic classes. Our goal is to perform stroke classification with high accuracy in a streaming mode:

$$p(\boldsymbol{y}|\boldsymbol{s}) = \prod_t p(y_t|s_{\leq t}) \tag{1}$$

where $s_{\leq t}$ denotes $\{s_1, \ldots, s_t\}$ and $\boldsymbol{y} = \{y_1, \ldots, y_T\}$. On the other hand, written strokes are allowed to correct their predictions in a predefined step latency $f$, where written strokes can look $f$ following strokes to improve inference accuracy:

$$p^f(\boldsymbol{y}|\boldsymbol{s}) = \prod_t p(y_t|s_{\leq t+f}). \tag{2}$$

In the following, we first introduce the stroke states and chunk construction scheme, and then present the Multiple Stroke State Transformer (MSST) for streaming stroke classification.

### 2.2. Multiple Stroke States

Streaming chunk-based Transformer [14] is an effective method for stroke classification with low latency, where all strokes are divided into chunks with predefined size $n$ in chronological order and conduct Transformer procedure in chunks, as shown in Fig. 2a. To make good use of historical information, each chunk has a history window with predefined size $m$ to memorize strokes written before, which are not updated in the chunk.

However, the division of strokes in chunk-based Transformer leads to few chunks in each document, which causes the inadequate utilization of training data. To overcome the disadvantage, we shift chunks and history windows (Fig. 2b) to build more chunks. Then we put chunks from one document together and fill stroke representation in history windows with the latest updated one, as shown in Fig. 2c, where each stroke step $t$ corresponds to a chunk $Chunk_t$. Each stroke $s_t$ has variable representations in different chunks, and we identify them with $B$ multiple chronological states $\{s_t^1, \ldots, s_t^B\}$. The first state $s_t^1$ only obtains information of strokes written before $s_t$, so it is used for real-time classification. Subsequent states $s_t^b$ $(1 < b \leq B)$ take responsibility for correcting real-time inference by exploring information of limited future strokes. The future strokes provide more context information, so the larger $b$ is, the more strokes $s_t^b$ looks ahead, and the more effective representation $s_t^b$ learns.

### 2.3. Stroke State Duration

The multiple states in Fig. 2c are too dense and the method suffers from high computational cost with large chunk size. This is because $\{s_t^1, \ldots, s_t^{B-1}\}$ only last one stroke step, which leads to frequent updates of stroke representation. To solve this problem, we provide each state a predefined duration $D_b$ $(D_b \geq 1)$ and $D_b$ is the same for all strokes. Stroke state $s_t^b$ with small $b$ learns insufficient representation due to less future information, so it should have short $D_b$, while $s_t^b$ with larger $b$ can last longer. Thus, $D_b$ increases monotonically as $b$ grows $(D_1 \leq D_2 \leq \cdots \leq D_B)$. Our final method is shown in Fig. 3, and the set of stroke state duration can strengthen information utilization (e.g., in Fig. 3, $s_3^3$, $s_5^2$ and $s_6^1$ from $Chunk_6$ make use of $Chunk_5$'s information through not only $s_2^3$ but also $s_4^2$.).

After each stroke state is given a duration time, the composition of chunks should be reconsidered. Let $C_b$ add up

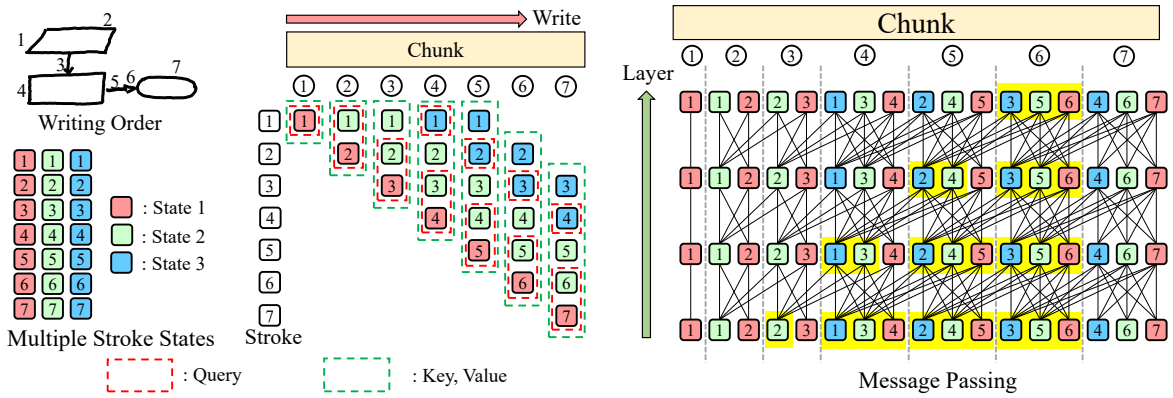


**Fig. 3**. Visualization of MSST and the message passing of stroke states as Transformer layer grows. We set $B = 3$, $D_1 = 1$, $D_2 = 2$, $D_3 = 2$. The example diagram has 7 strokes and leads to 7 chunks. Message passing in $Chunk_6$ is shown in yellow.

**Algorithm 1** Construction of Chunks

**Input**: $S=\{s_1^1, \ldots, s_1^B, s_2^1, \ldots, s_2^B, \ldots, s_T^B\}$, $C=\{C_1, \ldots, C_B\}$, $D=\{D_1, \ldots, D_B\}$

**for** $i = 1, 2, \ldots, T$ **do**
  $Chunk_i^Q = \{\}$, $Chunk_i^{KV} = \{\}$
  **for** $b = 1, 2, \ldots, B$ **do**
    $t = i - C_b$
    Add $s_t^b$ to $Chunk_i^Q$.
    **for** $j = t, t+1, \ldots, t + D_b - 1$ **do**
      Add $s_j^b$ to $Chunk_i^{KV}$.
    **end for**
  **end for**
**end for**

duration of stroke $s_t$ before $s_t^b$:

$$C_b = \begin{cases} 0 & , \quad b = 1 \\ \sum_{k=1}^{b-1} D_k & , \quad 1 < b \leq B \end{cases} \tag{3}$$

Let $Chunk_i^Q$ and $Chunk_i^{KV}$ denote the set of stroke states to be updated (Queries in Transformer) and the ones to provide information (Keys and Values in Transformer) in $Chunk_i$. The chunk construction scheme is illustrated in Algorithm 1.

### 2.4. Multiple Stroke State Transformer

After the construction of chunks, we conduct Transformer procedure to all chunks simultaneously, and we name our method Multiple Stroke State Transformer (MSST). Let $\boldsymbol{X}_i^l \in \mathbb{R}^{d \times n_1}$ and $\boldsymbol{H}_i^l \in \mathbb{R}^{d \times n_2}$ denote features of stroke states in $Chunk_i^Q$ and $Chunk_i^{KV}$, where $l$ is the layer index, $d$ is dim of hidden features, $n_1$ and $n_2$ denote the number of states in $Chunk_i^Q$ and $Chunk_i^{KV}$. Let $\boldsymbol{W}_Q^l, \boldsymbol{W}_K^l, \boldsymbol{W}_V^l \in \mathbb{R}^{d \times d}$ denote three parameter matrices. The standard Transformer procedure can be formulated as:

$$\tilde{\boldsymbol{X}}_i^l = \text{LayerNorm}(\boldsymbol{X}_i^l) \tag{4}$$

$$\boldsymbol{Q}_i^l = \boldsymbol{W}_Q^l \tilde{\boldsymbol{X}}_i^l, \quad \boldsymbol{K}_i^l = \boldsymbol{W}_K^l \boldsymbol{H}_i^l, \quad \boldsymbol{V}_i^l = \boldsymbol{V}_K^l \boldsymbol{H}_i^l \tag{5}$$

$$\hat{\boldsymbol{X}}_i^l = \text{Attention}(\boldsymbol{Q}_i^l, \boldsymbol{K}_i^l, \boldsymbol{V}_i^l, \boldsymbol{R}_i^l) + \boldsymbol{X}_i^l \tag{6}$$

where Attention$(\cdot)$ is the attention operation defined in [20]. Layer normalization and residual connection are applied to facilitate training. $\boldsymbol{R}_i^l \in \mathbb{R}^{n_2 \times n_1 \times d}$ is relative position encodings using methods [21] and [22] to exploit temporal and spatial information between strokes. A feed-forward networks (FFN) made of two fully connected layers and GeLU nonlinearity follows behind to generate features for the next layer:

$$\boldsymbol{X}_i^{l+1} = \text{FFN}(\text{LayerNorm}(\hat{\boldsymbol{X}}_i^l)) + \hat{\boldsymbol{X}}_i^l \tag{7}$$

Finally, a fully connected layer is employed for classification, and the whole model is learned by minimizing the standard cross-entropy loss.

One important principle of our method is to use limited future information. As Transformer layer grows, every stroke state to be updated can obtain information only from current and previous chunks, as illustrated in Fig. 3.

## 3. EXPERIMENTS

### 3.1. Experimental Setup

We conduct experiments on online handwritten document dataset IAMonDo [18] and diagram dataset FC [19] to validate our method. MSST is trained with Adam Optimizer with batchsize = 8, learning rate = 0.0005 and decay rate = 0.0001. MSST has $L = 5$ Transformer layers and 8 heads in multi-head self-attention. We set the embedding dimensionality $d = 256$, and the dropout rate as 0.1. All states of the same stroke are initialized with the same contour-based features following [9]. We set $D_b = 2^{b-1}, B = 7$ for IAMonDo and $D_b = 3^{b-1}, B = 3$ for FC.

**Table 1**. Comparison results on IAMonDo and FC. Subscript of MSST represents the state used for prediction. For example, $\text{MSST}_3$ denotes using strokes' third state for evaluation.

| Dataset | IAMonDo | | | FC | | |
|---|---|---|---|---|---|---|
| Method | $SCA_o$ | $SCA_a$ | Delay | $SCA_o$ | $SCA_a$ | Delay |
| Static | 97.96 | 95.29 | - | 98.80 | 97.79 | - |
| Real-time | 92.52 | 83.74 | 0 | 96.14 | 92.22 | 0 |
| Stream-1 | 92.52 | 83.74 | 0 | 96.14 | 92.22 | 0 |
| Stream-2 | 93.09 | 84.22 | 1 | 97.38 | 94.97 | 1 |
| Stream-3 | 94.21 | 87.35 | 3 | 97.78 | 95.64 | 4 |
| Stream-4 | 95.16 | 89.88 | 7 | 97.95 | 96.10 | 13 |
| Stream-5 | 95.75 | 90.77 | 15 | - | - | - |
| Stream-6 | 96.57 | 91.55 | 31 | - | - | - |
| Stream-7 | 96.71 | 92.05 | 63 | - | - | - |
| $\text{MSST}_1$ | 94.26 | 87.27 | 0 | 96.55 | 93.14 | 0 |
| $\text{MSST}_2$ | 94.56 | 87.88 | 1 | 97.71 | 95.37 | 1 |
| $\text{MSST}_3$ | 95.12 | 88.87 | 3 | 98.10 | 96.24 | 4 |
| $\text{MSST}_4$ | 95.79 | 90.04 | 7 | 98.25 | 96.26 | 13 |
| $\text{MSST}_5$ | 96.51 | 91.36 | 15 | - | - | - |
| $\text{MSST}_6$ | 97.01 | 92.18 | 31 | - | - | - |
| $\text{MSST}_7$ | 97.21 | 92.55 | 63 | - | - | - |

Several methods are chosen for comparison. **Static** is the standard Transformer where documents are complete and all strokes can see each other, and the result acts as the upper bound for streaming methods. **Real-time** is the Transformer with masked attention to make streaming classification. **Stream** is the chunk-based method [14], and we adopt different chunk sizes to obtain different average stroke delay. **MSST** is tested with different stroke states to classify under different delays. We evaluate all methods with the overall stroke classification accuracy ($SCA_o$) and the class-averaged stroke classification accuracy ($SCA_a$) following [9].

### 3.2. Comparison Results

Tab. 1 shows the stroke classification results of MSST and compared methods. There is a big performance gap between streaming methods (including **Real-time**, **Stream** and MSST) and the **Static** method on both IAMonDo and FC. For the same stroke delay, MSST consistently outperforms **Stream**, and more stroke delay makes the gap decrease (from 1.74% to 0.50% on IAMonDo on $SCA_o$). MSST can make real-time prediction and modify the classification by looking ahead. The more strokes MSST looks ahead, the more future information they get and the better performance they achieve, as $SCA_o$ grows from 94.26% to 97.21% on IAMonDo and from 96.55% to 98.25% on FC. Examples are shown in Fig. 4.

### 3.3. Effect of State Number

In this section, we analyze the effect of strokes' state number $B$ in MSST on IAMonDo. We set $B$ from 1 to 7, and state duration $D_b$ is $2^{b-1}$. To make the model of different B have the same size of receptive field, We then increase $D_B$ to 127, 126, 124, 120, 112, 96 for $B$ being 1, 2, 3, 4, 5, 6, respectively. The result is shown in Fig. 5. We can see that more

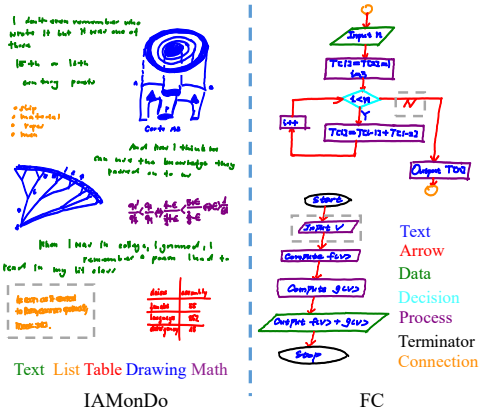


**Fig. 4**. Streaming stroke classification examples using MSST. Different colors denote different classes. The incorrect predictions have been framed by grey dashed boxes.

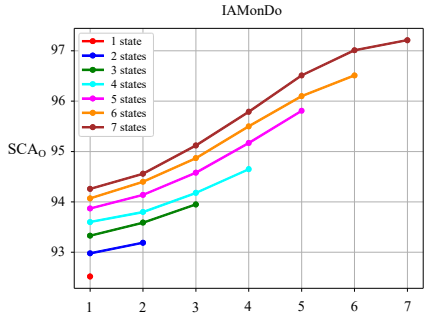


**Fig. 5**. $SCA_o$ of MSST with different state numbers on IAMonDo. Horizontal axis denotes the state used for prediction.

stroke states are beneficial for classification. When using the same state for inference, MSST with more states gets better performance (e.g., $SCA_o$ of $\text{MSST}_1$ with $B = 7$ is higher than the one with $B = 1$). That means high states help message passing to low ones.

## 4. CONCLUSION

In this work, we propose a Transformer-based framework called MSST for real-time stroke classification. We set multiple states with different duration for each stroke and divide them all into chunks to perform message passing by Transformer. Experiments on public handwriting datasets show encouraging results outperforming existing chunk-based method. Our future work will be considering dynamic stroke state which makes inference modification more flexible.

## 5. ACKNOWLEDGMENTS

This work is supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400 and the National Natural Science Foundation of China (NSFC) Grant No. 62276258.

## 6. REFERENCES


[1] Christopher M Bishop, Markus Svensen, and Goeffrey E Hinton, "Distinguishing text from graphics in on-line handwritten ink," in *Proc. ICFHRW*, 2004, pp. 142–147.

[2] Adrien Delaye and Cheng-Lin Liu, "Contextual text/non-text stroke classification in online handwritten notes with conditional random fields," *Pattern Recognition*, vol. 47, no. 3, pp. 959–968, 2014.

[3] Jun-Yu Ye, Yan-Ming Zhang, and Cheng-Lin Liu, "Joint training of conditional random fields and neural networks for stroke classification in online handwritten documents," in *Proc. ICPR*, 2016, pp. 3264–3269.

[4] Emanuel Indermühle, Volkmar Frinken, and Horst Bunke, "Mode detection in online handwritten documents using BLSTM neural networks," in *Proc. ICHFR*, 2012, pp. 302–307.

[5] Illya Degtyarenko, Ivan Deriuga, Andrii Grygoriev, Serhii Polotskyi, Volodymyr Melnyk, Dmytro Zakharchuk, and Olga Radyvonenko, "Hierarchical recurrent neural network for handwritten strokes classification," in *Proc. ICASSP*, 2021, pp. 2865–2869.

[6] Xingyuan Wu, Yonggang Qi, Jun Liu, and Jie Yang, "Sketchsegnet: A RNN model for labeling sketch strokes," in *Proc. MLSPW*, 2018, pp. 1–6.

[7] Jun-Yu Ye, Yan-Ming Zhang, Qing Yang, and Cheng-Lin Liu, "Contextual stroke classification in online handwritten documents with graph attention networks," in *Proc. ICDAR*, 2019, pp. 993–998.

[8] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Xiangzhi Wei, Kun Zhou, and Youyi Zheng, "SketchGNN: Semantic sketch segmentation with graph neural networks," *ACM Transactions on Graphics*, vol. 40, no. 3, pp. 1–13, 2021.

[9] Xiao-Long Yun, Yan-Ming Zhang, Fei Yin, and Cheng-Lin Liu, "InstanceGNN: a learning framework for joint symbol segmentation and recognition in online handwritten diagrams," *IEEE Transactions on Multimedia*, vol. 24, pp. 2580–2594, 2021.

[10] Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Liang Lu, Guoli Ye, and Ming Zhou, "Low latency end-to-end streaming speech recognition with a scout network," *arXiv preprint arXiv:2003.10369*, 2020.

[11] Zhengkun Tian, Jiangyan Yi, Ye Bai, Jianhua Tao, Shuai Zhang, and Zhengqi Wen, "Synchronous transformers for end-to-end speech recognition," in *Proc. ICASSP*, 2020, pp. 7884–7888.

[12] Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei, "Unified streaming and non-streaming two-pass end-to-end model for speech recognition," *arXiv preprint arXiv:2012.05481*, 2020.

[13] Shiliang Zhang, Zhifu Gao, Haoneng Luo, Ming Lei, Jie Gao, Zhijie Yan, and Lei Xie, "Streaming chunk-aware multihead attention for online end-to-end speech recognition," *arXiv preprint arXiv:2006.01712*, 2020.

[14] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*. IEEE, 2021, pp. 5904–5908.

[15] Chunyang Wu, Yongqiang Wang, Yangyang Shi, Ching-Feng Yeh, and Frank Zhang, "Streaming transformer-based acoustic models using self-attention with augmented memory," *arXiv preprint arXiv:2005.08042*, 2020.

[16] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. ICASSP*, 2021, pp. 6783–6787.

[17] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[18] Emanuel Indermühle, Marcus Liwicki, and Horst Bunke, "IAMonDo-database: an online handwritten document database with non-uniform contents," in *Proc. DASW*, 2010, pp. 97–104.

[19] Ahmad-Montaser Awal, Guihuan Feng, Harold Mouchere, and Christian Viard-Gaudin, "First experiments on a new online handwritten flowchart database," in *Document Recognition and Retrieval XVIII*, 2011, vol. 7874, pp. 81–90.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Proc. NIPS*, 2017.

[21] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[22] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proc. ICCV*, 2021, pp. 10033–10041.