



# SSA: A Content-Based Sparse Attention Mechanism

Yang Sun<sup>1,3</sup> , Wei Hu<sup>1,3</sup> , Fang Liu<sup>2,4</sup> , Feihu Huang<sup>1,3</sup>,  
and Yonghao Wang<sup>5</sup>

<sup>1</sup> Wuhan University of Science and Technology, Wuhan, China  
{202013704135, huwei, huangfeihu}@wust.edu.cn

<sup>2</sup> Wuhan University, Wuhan, China  
liufangfang@whu.edu.cn

<sup>3</sup> Hubei Province Key Laboratory of Intelligent Information Processing  
and Real-Time Industrial System, Wuhan, China

<sup>4</sup> Department of Information Engineering, Wuhan Institute of City, Wuhan, China

<sup>5</sup> DMT Lab, Birmingham City University, Birmingham, UK  
yonghao.wang@bcu.ac.uk

**Abstract.** Recently, many scholars have used attention mechanisms to achieve excellent performance results on various neural network applications. However, the attention mechanism also has shortcomings. Firstly, the high computational and storage consumption makes the attention mechanism difficult to apply on long sequences. Second, all tokens are involved in the computation of the attention map, which may increase the influence of noisy tokens on the results and lead to poor training results. Due to these two shortcomings, attention models are usually strictly limited to sequence length. Further, attention models have difficulty in exploiting their excellent properties for modelling long sequences. To solve the above problems, an efficient sparse attention mechanism (SSA) is proposed in this paper. SSA is based on two separate layers: the local layer and the global layer. These two layers jointly encode local sequence information and global context. This new sparse-attention patterns is powerful in accelerating reasoning. The experiments in this paper validate the effectiveness of the SSA mechanism by replacing the self-attentive structure with an SSA structure in a variety of transformer models. The SSA attention mechanism has achieved state-of-the-art performance on several major benchmarks. SSA was validated on a variety of datasets and models encompassing language translation, language modelling and image recognition. With a small improvement in accuracy, the inference calculation speed was increased by 24%.

**Keywords:** Transformers · Sequence · Local attention · Global attention · Sparsity

## 1 Introduction

The attention mechanism is widely used in sequence modelling [13]. Initially validated only on machine translation, attention mechanisms have now been

widely used in natural language processing and computer vision [12]. In recent years, state-of-the-art neural networks have also been implemented by attention mechanisms, such as Transformer-XL [3], bert [5].

Self-attention is one of the classical attention mechanisms. The self-attention processes the input sequence sequentially. At each time step, attention is assigned weights to the preceding elements, and these weights are summed as the attention weights of the current element. The process of assigning weights is called connection building. The excellent performance of the attention mechanism is due to the fact that it maintains more connections than CNN and RNN, and these connections are able to capture more feature information in the data. However, too many connections also make the complexity higher than CNNs and RNNs. Specifically, on a sequence of length  $n$ , weights need to be assigned to the sequence data of length  $i$  for each position  $i < n$ . The complexity of attention is  $\frac{n(n-1)}{2}$ . The square level of complexity limits the performance of the attention model to the length of the sequence. As computing devices such as GPUs have been updated, the sequence length limit for attention models has been increased to 512 tokens. Nonetheless, the overly complex models lead to an attention mechanism that is particularly difficult to handle for large sequence modelling. This clearly does not satisfy most application scenarios. Long sequences are the trend in sequence modelling, including document-level machine translation, high-resolution image recognition, speech, video generation, etc. At the same time the attention mechanism has a second drawback, it has the potential to reduce the noise resistance of the model [2]. If the input sequence contains noisy tokens, the noisy tokens will be involved in too much of the computation process, which will lead to impaired model performance.

In the self-attention model, each element pays attention to the other elements. However, the training results show that most of the attention matrix elements are small, which indicates that these tokens do not have a significant impact on the model results, but they are still heavily involved in the attention calculation process, which leads to wasted computational and storage resources. These non-essential attention calculations can be removed to optimise model complexity and reduce the impact of noise on model accuracy. This optimised model is known as a sparse attention model.

Many optimisation schemes of the sparse attention mechanism have been proposed. However, each element of the local attention model will only focus on other elements at a fixed location and cannot flexibly encode remote dependencies. An alternative to local attention is context-based sparse attention, which enables more flexible encoding of distant dependencies. Scholars limit the number of connections per element by analysing the context, an approach known as content-based sparse attention [14]. The process of assigning a connection to each element is called constructing a sparsity pattern. Developers can define their own suitable sparsity patterns depending on the deep learning task and dataset. As a result, content-based sparse attention is more flexible than local attention. Previous work has demonstrated that sparsity patterns can have a significant impact on model performance.

This paper proposes a sparsity pattern that can flexibly encode local information and global dependencies. And Sparse Spectral Attention (SSA) is implemented based on this sparsity pattern. The SSA mechanism has the following features: (1) the SSA mechanism maintains the ability to aggregate local information and long-distance dependent information, (2) the SSA model is less complex than the attention model, and (3) the SSA model reduces the impact of noisy data on the model and improves the model’s resistance to interference. SSA combines the advantages of both local attention and content-based sparse attention and achieves good performance in several sequence modelling tasks.

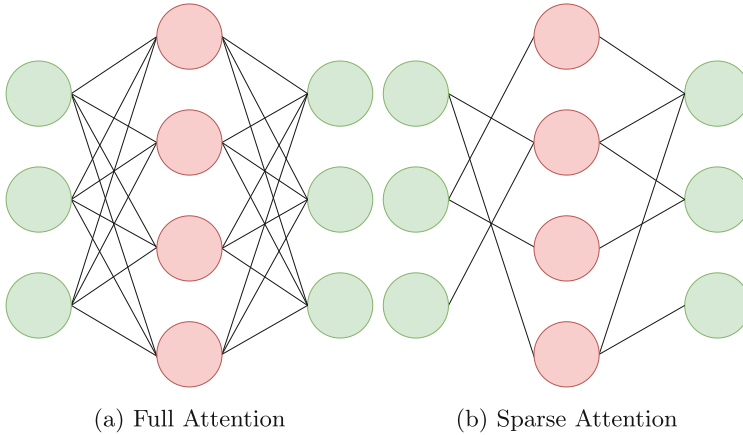
The contributions of this article are:

- We propose a sparse attention mechanism to replace the original self-attention mechanism. SSA can encode global features and local information, and reduce model complexity.
- We have analysed some recent works on sparse attention and compared them with SSA.
- We have replaced self-attention with SSA in the official codes of several state-of-the-art transformer models, involving machine translation, image recognition (Ciar10, Cifar100, ImageNet-64) and language modelling (enwik8). Experimental results and analysis are proposed.

## 2 Related Works

With the rapid advances in computer hardware [11, 26, 27] and network infrastructure [13, 15, 25], big data [23, 24, 34] and machine learning [9, 17, 19] have been successfully applied in various areas, such as finance [4, 22], tele-health [18, 21, 31], and transportation [16, 20]. One of the most successful areas is nature processing language. In recent years, many optimisations [10] have been proposed to improve the computational efficiency of attention models. Local attention and content-based sparse attention are the dominant research directions. The core idea is to limit the number of connections. Figure 1a shows the connections constructed by the attention model in the language sequence. Each edge represents a connection. It is clear to see that the attention model needs to maintain a square level number of connections for sequences of length  $n$ . Not all of these connections are necessary.

In contrast, sparse attention, as shown in Fig. 1b, removes most of the connections and the necessary ones are retained. Recent achievements on local attention include [1, 35, 37], etc. The above achievements are all local attention models based on fixed positions. At each time step, local attention sequentially divides the sequence into multiple shorter sequences and then creates connections in each of the multiple shorter sequences. This strategy allows the model to extract features based on the local neighborhoods of the current time step. The non-zero elements of the attention matrix are concentrated on the diagonal, so that only the non-zero elements need to be stored structurally to achieve significant savings in computational and memory overhead. Despite the good results achieved with local attention, local attention cannot encode remote dependence.



**Fig. 1.** The connections in attention mechanisms

Content-based sparse attention is a more flexible attention mechanism. While local attention and strided attention are fixed sparsity patterns, the sparsity patterns of content-based sparse attention are learned in neural networks. Reformer [7] proposed *location-sensitive hashing* (LSH) to infer attentional sparsity patterns. Reformer linearly projects tokens onto a random hyperplane and assigns them to different hash buckets. Tokens that fall into the same hash bucket can get to attend to each other. Similar work includes Cluster-Former [32], Fastformer [33] and Sparse sinkhorn attention [29]. Each of these results defines a different sparsity pattern to limit the number of connections for attention. However, it is often necessary to instantiate the full attention matrix for sparsification before a content-based sparse model can be built. These sparse attentions also lead to high storage consumption. The Routing transformer [28] explores sparse attention based on K-means clustering. Compared to other models, Routing transformer does not need to maintain an attention matrix larger than the batch size at all times to complete the clustering assignment. This reduces storage consumption while reducing computational consumption.

Our work combines the advantages of both local attention and content-based sparse attention as described above. Our work adds two separate sub-layers to the attention model, which encode local information and global context respectively, and subsumes the dependency information from the two sub-layers for attention.

### 3 Sparse Spectral Attention

The proposed framework relies on two transformer layers: (1) the local layer and (2) the global layer. The overall structure of the model is shown in Fig. 2, and our work is focused before Dot Product Attention. The former uses an dilated sliding window to encode local sequence information, while the latter encodes the global context through attention map pruning.

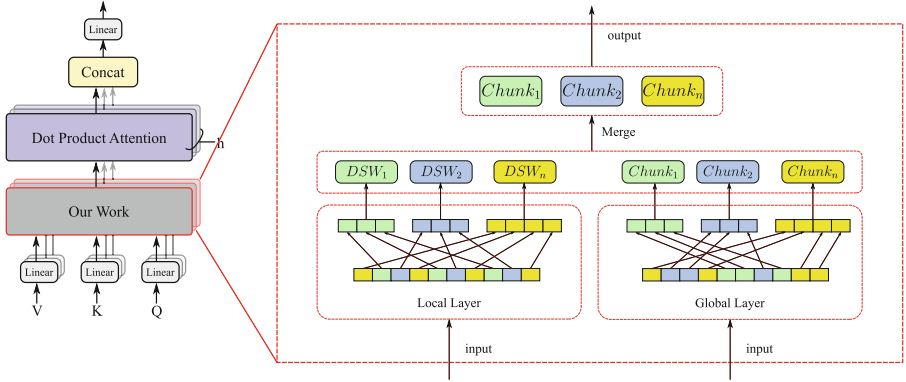


Fig. 2. The overall structure of SSA.

### 3.1 Local Layer

The core of the local layer is an dilated sliding window that focuses on encoding local sequence information. Although our model intends to capture global contextual dependencies, local sequence information also plays an important role [6]. As shown in Fig. 2, the Local layer consists of the dilated sliding window layer, which has a larger perceptual space than the standard sliding window. Dilated sliding window divides a long sequence  $X$  of length  $n$  into overlapping windows of size  $w$  and step size  $m$ . The sliding sequence  $DW_k^i$  for each time step can be expressed as

$$DW_k^i = x_i [m \times k : (m \times k + w) \times d : d] \tag{1}$$

while  $[index_1 : index_2 : step]$  indicates the selection row from the order of the input matrix between rows  $index_1$  and  $index_2$ . Unlike standard sliding windows, dilated sliding windows have gaps of size dilation  $d$ . This gap allows the Dilated Sliding Window layer to increase the receptive field without increasing complexity. In two models with the same number of layers, the receptive field based on the dilated sliding windows is expanded by  $d$  times.

### 3.2 Global Layer

The global layer implements a sparsity pattern based on structured pruning, which focuses on encoding global contextual information. The structure is shown in Fig. 2. We first chunk the sequence and construct a sparse attention matrix. The core process has two steps: 1) partition the attentional similarity graph into multiple subgraphs based on the undirected graph cut algorithm. 2) for each query, the set of keys found in the same subgraph is defined as  $S_i$ .

The Global Layer input consists of a matrix  $Q$  consisting of  $query_i$  vectors and a matrix  $K$  consisting of  $key_i$  vectors. The adjacency matrix of the attention map is denoted as  $A = QK^T$ , where  $A$  is an  $N \times N$  matrix and  $N$  is the length

of the sequence. The element  $A_{ij}$  in the attention map represents the relevance measure between  $token_i$  and  $token_j$ . In order to reduce the computational effort of the model, we tend to prune the smallest part of the elements of the attention map.

In the pruning process, we traverse the attention map by row and retain the largest elements. The pruning scheme is defined as:

$$[\tau_k(A)]_{ij} = \begin{cases} A_{ij} & A_{ij} \in Tok_k\{row_j\} \\ c & A_{ij} \notin Tok_k\{row_j\} \end{cases} \tag{2}$$

$\tau_k(A)$  is the sparse attention map after pruning, and  $c$  is a small constant.

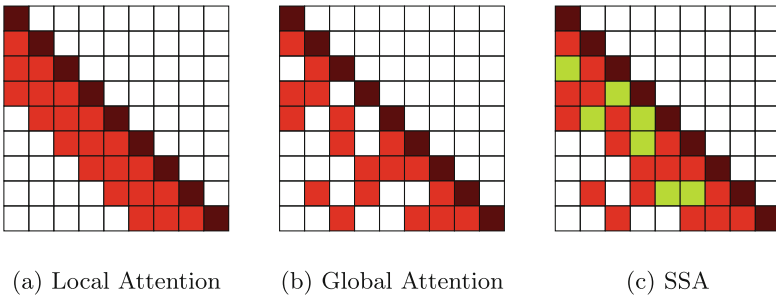
After the two representation layers have calculated the two buckets  $DSW$  and  $\tau_k(A)$ , we merge the two buckets together.

$$\chi(A) = DSW \cup \tau_k(A) \tag{3}$$

Ultimately, the procedure for the local and global layers can be formulated as:

$$\hat{V} = softmax(\chi(\frac{QK^T}{\sqrt{d}})) \cdot V \tag{4}$$

The different approach of our proposed model to other models dealing with long sequences is shown in Fig. 3.



**Fig. 3.** Each square represents a hidden state. Local attention is shown in (a). Local attention is built by a sliding window (red square) with cross-sequential attention. Content-based sparse attention is shown in (b), where attention is built through global contextual information. (c) is our proposed method to process both local and global information by subsuming the hidden states of (a) and (c). The yellow boxes in c are from the local layer and the red boxes are from the global layer. (Color figure online)

### 3.3 Analysis of Sparsity Patterns

Table 1 analyses the modelling complexity of the different sparsity patterns. Papers [1, 14] have the lowest sparsity pattern construction complexity, and their complexity in maintaining a sliding window to select neighbouring data on the sequence is linear. Similarly Reformer [7], Star transformer [37] have lower complexity. Although they are content-based sparse attention and the sparsity pattern is different from local attention, their computational overhead is not significant. Strided attention [38] is a square level of complexity, as they need to traverse the attention map to determine whether dependencies are to be retained. This scheme is commonly used in sparse attention models based on clustering, which increases the complexity but allows for more accurate accuracy. As a comparison, our proposed model incorporates two representation layers that can operate independently. These two representation layers encode local and global information separately, and the complexity of our Local layer is linear, and the complexity of our Global layer is  $O(l^{1.5})$ . This indicates that our Local layer has similar time complexity to other content-independent sparsity patterns, and our Global layer can be constructed more quickly than other content-based sparsity patterns.

**Table 1.** Complexity of sparse attention construction

Schemes	Sparsity patterns	Complexity
Local attention	Sliding window	$O(n)$
Strided attention	SortCut	$O(n^2)$
Routing transformer	K-means	$O(n^{1.5})$
Reformer	LSH	$O(n)$
Star transformer	Star-shaped topology	$O(n)$
Longformer	Sliding window	$O(n)$
Sparse sinkhorn attention	SortCut	$O(n)$
Ours-local layer	Sliding window	$O(n)$
Ours-global layer	Purning	$O(l^{1.5})$

## 4 Experiments

In this section, we replace the dense attention of the existing official transformer code with SSA to validate the model effects. The experiments involve machine translation, language modelling and image recognition. Ablation studies are also provided.

### 4.1 Experimental Settings

We experiment on different models and datasets. For natural language processing, two sets of experiments were designed. The first experiment was German-English machine translation with a dataset using Multi30k, which is a smaller dataset containing 145000 training, 5070 development and 5000 test. The second experiment is language modelling with a cropped dataset of enwik8. The complete enwik8 dataset is a dataset of the first 100 million characters dumped from Wikipedia XML. We also conducted experiments for image sequences. We performed image recognition on the cifar10, cifar100 and ImageNet datasets. Cifar10 and cifar100 are labelled subsets of the 80 million micro-image dataset. Cifar10 contains 10 categories of  $128 \times 128$  colour images with 50,000 training images and 10,000 test images. Similarly, cifar100 has more categories and numbers of colour images, with 500 training images and 100 test images per category. All images for this experiment were cropped to  $224 \times 224$ . We explored the effect of different crop ratios on the model in the ablation experiment. Since SSA degenerates to local attention for  $r = 1$ , the experiments were all set to  $r \leq 0.8$ .

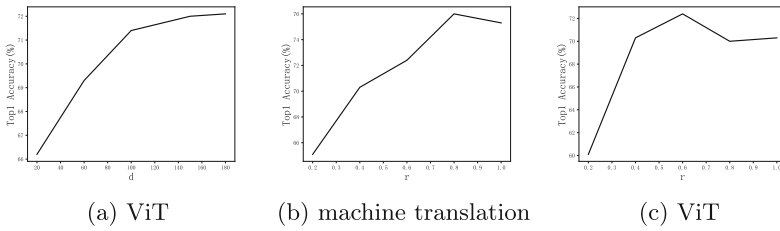


Fig. 4. The impact of hyperparameters

### 4.2 Accuracy Results

We first explored the impact of SSA on the accuracy of the model. These effects were generally positive. We explored the accuracy performance of SSA in different models. Table 2 shows the results of the German-English machine translation. The training set, validator, and test set used for each version of the experiment are the same. As can be seen from Table 2, SSA can achieve better accuracy results than the baseline model. When the parameter  $r$  was set to 0.8, the BLEU of the SSA has improved to 33.08. When  $r$  was set to 0.5, each tokens attended to too little information for training, and eventually the PPL rose to 7.724 and the BLEU fell to 30.97. We suspect that this is due to the size of the Multi30k dataset. Each tokens in a short sequence is full of feature information and setting a small pruning ratio will lose model accuracy. As a comparison, Table 3 shows the results of language modelling on a longer sequence dataset, enwik8.



**Table 2.** Results of German-English machine translation experiments.

Arch	Sparse ratio $r$	PPL	BLEU
VAG-NMT	–	–	31.6
Transformer	–	6.949	30.97
Transformer (SSA)	0.5	7.724	25.13
Transformer (SSA)	0.8	5.671	33.08

**Table 3.** Results of enwik8 language modelling experiments.

Arch	Sparse ratio $r$	BPC
LSTM	–	1.203
Transformer	–	1.18
Transformer (SSA)	0.5	1.08
Transformer (SSA)	0.8	1.03

**Table 4.** Results of image recognition experiments.

Arch	Top-1 Acc
ViT-Cifar10	81.1%
ViT-SSA-Cifar10	81.7%
ViT-Cifar100	83.3%
ViT-SSA-Cifar100	83.7%
T2TViT-ImageNet	82.4%
T2TViT-SSA-ImageNet	82.5%

Our work achieves 1.03 BPC and the results demonstrate that the sparse layer of SSA achieves higher accuracy metrics for models on long sequence datasets.

Table 4 explores the ablation experiments of SSA on image recognition. We replaced the official code of ViT, T2T and replaced the core attention mechanism with SSA. SSA achieved 81.7% accuracy on ViT-Cifar10, 83.3% accuracy on ViT-Cifar100, and 82.5% accuracy on T2T-ImageNet.

### 4.3 The Impact of Hyperparameters

SSA has two important hyperparameters. One is the sliding window size  $d$ . Its effect is shown in Fig. 4a. The sliding window size is tended to be set larger to achieve higher accuracy. But a longer size means more calculations, so the parameters need to balance accuracy with speed of calculation. The other is the pruning ratio  $r = k/n$  of the global layer its effect is shown in Fig. 4b and Fig. 4c. Different models should be assigned different  $r$  values. For a total number of tokens  $n > 512$ , we tend to set a slightly smaller  $r$  value, with  $r$  between 0.4

and 0.5 giving better accuracy results. For  $n < 512$ , the pruning ratio should not be too small, 0.7–0.8 is appropriate. Further, the results demonstrate that  $r$  is influenced by the way tokens are generated. Simple token generation methods, such as vanilla attention [30], ViT [8], embedding sequences directly into tokens, where no information is exchanged between tokens. These models should be set to a smaller pruning ratio. However, complex token generation methods [36] should be set to a larger pruning ratio.

#### 4.4 Speed Results

We evaluate the performance of the acceleration by comparing two versions of the ViT-SSA inference task (with/without sparse matrix multiplication) running on CPU and GPU platforms. The computing platform for the comparison test was a Xeon E5 CPU and a GTX 2080ti. ViT-SSA refers to the Vision transformer which uses the SSA mechanism to replace self-attention. We designed two versions of the experiment, in one version we used dense matrix multiplication for the calculations and the other version used sparse matrix multiplication, and Table 5 presents the average inference time results for each version over multiple experiments.

Compared to the baseline, ViT-SSA-dense does not take full advantage of the sparse matrix of SSA and the inference speed is not significantly improved. In contrast, ViT-SSA-sparse with sparse matrix computation improved inference speed on CPU by 43% and on GPU by 24% over baseline.

**Table 5.** Inference speed results.

Device	Version	Time (ms)
Xeon CPU	ViT-dense	2900
Xeon CPU	ViT-sparse	187
Xeon CPU	ViT-SSA-dense	2832
Xeon CPU	ViT-SSA-sparse	106
GTX 2080ti	ViT-dense	61
GTX 2080ti	ViT-sparse	49
GTX 2080ti	ViT-SSA-dense	58
GTX 2080ti	ViT-SSA-sparse	37

## 5 Conclusion

The experiments are compared in two domains, natural language processing and computer vision, and we give a review of SSA’s models for machine translation, language modelling, and image recognition tasks. We replace the original self-attention mechanism with the SSA attention mechanism on a variety of transformer models. The results show that our model is able to achieve more advanced

performance on several major benchmarks. One of the more significant performance improvements on the accuracy benchmark is V2T-cifar10, with a 0.6% improvement in top1 Accuracy.

## References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv preprint [arXiv:2004.05150](https://arxiv.org/abs/2004.05150) (2020)
2. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. arXiv preprint [arXiv:1911.03584](https://arxiv.org/abs/1911.03584) (2019)
3. Dai, Z., et al.: Transformer-XL: language modeling with longer-term dependency (2018)
4. Gai, K., Du, Z., et al.: Efficiency-aware workload optimizations of heterogeneous cloud computing for capacity planning in financial industry. In: IEEE 2nd CSCloud (2015)
5. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
6. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: a survey. ACM Comput. Surv. (CSUR) (2021)
7. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. arXiv preprint [arXiv:2001.04451](https://arxiv.org/abs/2001.04451) (2020)
8. Kolesnikov, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale (2021)
9. Li, Y., Song, Y., et al.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE TII **17**(4), 2833–2841 (2020)
10. Lin, T., Wang, Y., Liu, X., Qiu, X.: A survey of transformers. arXiv preprint [arXiv:2106.04554](https://arxiv.org/abs/2106.04554) (2021)
11. Liu, M., Zhang, S., et al.:  $H_\infty$  state estimation for discrete-time chaotic systems based on a unified model. IEEE Trans. SMC (B) **42**(4), 1053–1063 (2012)
12. Lu, R., Jin, X., et al.: A study on big knowledge and its engineering issues. IEEE TKDE **31**(9), 1630–1644 (2019)
13. Lu, Z., Wang, N., et al.: IoTDeM: an IoT big data-oriented MapReduce performance prediction extended model in multiple edge clouds. JPDC **118**, 316–327 (2018)
14. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint [arXiv:1508.04025](https://arxiv.org/abs/1508.04025) (2015)
15. Niu, J., Gao, Y., et al.: Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability. JPDC **72**, 1565–1575 (2012)
16. Qiu, H., Qiu, M., Lu, R.: Secure V2X communication network based on intelligent PKI and edge computing. IEEE Netw. **34**(42), 172–178 (2019)
17. Qiu, H., Qiu, M., Lu, Z.: Selective encryption on ECG data in body sensor network based on supervised machine learning. Inf. Fusion **55**, 59–67 (2020)
18. Qiu, H., Qiu, M., et al.: Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. IEEE J. Biomed. Health Inform. **24**, 2499–2505 (2020)
19. Qiu, H., Zheng, Q., et al.: Deep residual learning-based enhanced JPEG compression in the Internet of Things. IEEE TII **17**(3), 2124–2133 (2020)

20. Qiu, H., Zheng, Q., et al.: Topological graph convolutional network-based urban traffic flow and density prediction. *IEEE ITS* **22**(7), 4560–4569 (2020)
21. Qiu, L., Gai, K., Qiu, M.: Optimal big data sharing approach for tele-health in cloud computing. In: *IEEE SmartCloud*, pp. 184–189 (2016)
22. Qiu, M., Cao, D., et al.: Data transfer minimization for financial derivative pricing using Monte Carlo simulation with GPU in 5G. *Int. J. Commun Syst* **29**(16), 2364–2374 (2016)
23. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. *FGCS* **87**, 772–781 (2018)
24. Qiu, M., Guo, M., et al.: Loop scheduling and bank type assignment for heterogeneous multi-bank memory. *JPDC* **69**, 546–558 (2009)
25. Qiu, M., Liu, J., et al.: A novel energy-aware fault tolerance mechanism for wireless sensor networks. In: *IEEE/ACM Conference on GCC* (2011)
26. Qiu, M., Xue, C., et al.: Efficient algorithm of energy minimization for heterogeneous wireless sensor network. In: *IEEE EUC Conference*, pp. 25–34 (2006)
27. Qiu, M., Xue, C., et al.: Energy minimization with soft real-time and DVS for uniprocessor and multiprocessor embedded systems. In: *IEEE DATE Conference*, pp. 1–6 (2007)
28. Roy, A., Saffar, M., Vaswani, A., Grangier, D.: Efficient content-based sparse attention with routing transformers. *Trans. Assoc. Comput. Linguist.* **9**, 53–68 (2021)
29. Tay, Y., Bahri, D., Yang, L., Metzler, D., Juan, D.C.: Sparse sinkhorn attention. In: *International Conference on Machine Learning*, pp. 9438–9447. PMLR (2020)
30. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
31. Wang, J., Qiu, M., Guo, B.: High reliable real-time bandwidth scheduling for virtual machines with hidden Markov predicting in telehealth platform. *FGCS* **49**, 68–76 (2015)
32. Wang, S., Zhou, L., et al.: Cluster-former: clustering-based sparse transformer for question answering. In: *ACL-IJCNLP*, pp. 3958–3968 (2021)
33. Wu, C., Wu, F., Qi, T., Huang, Y., Xie, X.: Fastformer: additive attention can be all you need. *arXiv preprint arXiv:2108.09084* (2021)
34. Wu, G., Zhang, H., et al.: A decentralized approach for mining event correlations in distributed system monitoring. *JPDC* **73**(3), 330–340 (2013)
35. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: *Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1480–1489 (2016)
36. Yuan, L., Chen, Y., et al.: Tokens-to-Token ViT: training vision transformers from scratch on ImageNet. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567 (2021)
37. Zhang, Z., Jiang, Y., et al.: STAR: a structure-aware lightweight transformer for real-time image enhancement. In: *IEEE/CVF CV*, pp. 4106–4115 (2021)
38. Zhou, C., Bai, J., et al.: ATRank: an attention-based user behavior modeling framework for recommendation. In: *32nd AAAI* (2018)