The supplement of scFTAT

Binhua Tang and Yiyao Chen
Contact: bh.tang@hhu.edu.cn

1. The main structure of scFTAT:

1.1 The improved Transformer layer:

After preliminary training in the frequency domain space through the FFT encoding layer, the output gene expression matrix is input to the improved Transformer layer. A complete Transformer layer generally consists of a multi-head attention mechanism module, a feedforward function module, and the necessary residual layers. In this study, improvements have been made to these three modules to achieve a more efficient and faster process for cell type identification.

Let *X* be the input gene expression data. To preserve the sequential information of the input gene expression data, absolute positional encoding is performed before entering the multi-head attention layer. Specifically, sine and cosine functions are used to represent the positions of the data. In the Transformer model, the general representation is as follows:

$$PE(p,2i)=\sin(p/1000^{2i/d_m}) \tag{S1}$$

$$PE(p,2i+1)=\cos(p/1000^{2i/d_m}) \tag{S2}$$

where p represents the specific position of the input gene x in the gene expression matrix X, i represents the gene dimension, and d_m represents the dimension of the embedded positional encoding vector. Thus, the original input data X added with the result of absolute positional encoding, $PE(X)$, serves as the input X_{PE} for the subsequent multi-head attention layer.

The multi-head attention mechanism module is essentially composed of multiple parallel self-attention modules. The method for calculating the self-attention module is represented as below,

$$Attention(Q,K,V) = SoftMax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{S3}$$

where $Q = W^q X_{PE}, K = W^k X_{PE}, V = W^v X_{PE}, W^{q.k,v}$, Q, K, V represent the query vector, key vector, and value vector, respectively, and X_{PE} represents the output vector of the input vector after positional encoding.

Equation (S3) calculates attention scores by directly computing the dot product of *Q* and *K*, then normalizing it using SoftMax to obtain the final output. This method directly incorporates the absolute positional encoding information into the context representation. However, using relative positional encoding has better generalization ability and scalability compared to absolute positional encoding, which is more advantageous when dealing with long sequences. For single-cell input data, incorporating the relative information between input genes into model training can improve the performance of the model.

Some Transformer-based models in the computer vision have already explored relative positional encoding [4]. Thus, in this study, the process of calculating attention scores is modified to implement relative positional encoding. In simple terms, after calculating *Q* and *K*, the *Q* and *K* are multiplied with the rotation encoding matrix that corresponds to them. In any even dimension, the rotated positional encoding matrix *R* and the representation of the new *Q* are shown below:

$$R^d_{\theta,m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix} \tag{S4}$$

$$Q = R^d_{\theta,m} \begin{pmatrix} q^0 \\ q^1 \\ q^2 \\ \vdots \\ q^{d-1} \end{pmatrix} \tag{S5}$$

where *d* refers to the spatial dimension, *m* the position of the query vector *Q*, and *q* the specific elements of *Q* in each dimension. Therefore, *θ* can be estimated on *d*, and its vector is denoted as $\Theta = \{\theta_i = 10000^{\frac{-2(i-1)}{d}}, i \in [1, 2, ..., \frac{d}{2}]\}$.

Because Equation (S4) is an orthogonal matrix, it does not change the vector magnitude during operation, thus further ensuring the model stability. In practical implementation, as $R_{\theta,m}^{d}$ is sparse, it can be estimated as follows to reduce the time complexity,

$$Q = R_{\theta,m}^{d} X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ \vdots \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \end{pmatrix} \tag{S6}$$

By this means, the newly updated *Q* and *K* will contain relative positional information, achieved by absolute positional encoding. After obtaining the new *Q* and *K*, attention scores can be further obtained.

Here the SoftMax operation is replaced with the kernel function approximation [5], depicted as,

$$\Phi_1 \Phi_2 \approx SoftMax\left(\frac{QK^T}{\sqrt{d}}\right) \tag{S7}$$

where Φ_1 and Φ_2 represent the updated *Q* and *K*. The specific form of Φ is denoted as,

$$\Phi = \frac{p}{\sqrt{m}} \exp(W^T x - \frac{\|x\|^2}{2}) \tag{S8}$$

where *p* is a positive constant, *W* the product of the original input data matrix and a random orthogonal matrix, *m* the dimension of matrix *W*, and *x* the input *Q* or *K*. The random orthogonal matrix here can reduce the dimensionality of the original input while retaining the corresponding features.

Next, we use the multi-head self-attention mechanism, with each head named as $head_h$, to calculate their respective attention weights in parallel, as shown in the following equation. The specific number of heads is chosen based on the size and number of categories of the dataset.

$$MulitHead(Q, K, V) = Concat(head_1, ..., head_h) \tag{S9}$$

where *Concat* refers to the component-wise sum, $head_h = Attention(Q_h, K_h, V_h)$. After combining into a multi-head attention module, we adopt the Rezero method [6] to rescale the self-attention block. Specifically, the residual connection is represented in the following form,

$$X_i^{'} = X_i + \alpha_i sublayer(X_i) \tag{S10}$$

where X_i is the input of the attention module, $X_i^{'}$ is the output of the module, and α_i is a learnable residual weight that is shared by each multi-head attention module. This parameter is initialized to 0. This ensures that in the early training, the gradients of all parameters in the sublayer function in Equation (S10) will disappear and then reach a suitable value during the training process, which further speeds up the convergence rate of the network.

After completing the processing of the attention module and the corresponding residual layer, a feedforward network is added to obtain nonlinear data features. The work employs a parallel feedforward function module based on global and local information enhancement [6]. In global information, the input global representation is obtained through average pooling, followed by a fully connected operation. In local information, features are directly extracted through a fully connected operation. The approach is relevant to the concept of channel attention [7]. Subsequently, the outputs of the two branches are interacted via concatenation. Finally, the two pieces of information are weighted by a gating unit to obtain the corresponding attention weights, and the output dimension is aligned with the original input dimension. Overall, this module can enhance the model's final expression ability without significantly increasing computational complexity.

2. The experiments on the selected scRNA-seq datasets

2.1 The summary of the ablation experiments:

Table S1. The ablation experiment on the mouse-bladder data

	ACC	F1	Precision	Recall	MCC
Transformer	0.812	0.704	0.718	0.748	0.777
Transformer(K)	0.821	0.730	0.766	0.720	0.783
Transformer(R)	0.827	0.725	0.762	0.714	0.791
Transformer(F)	0.825	0.762	0.762	0.787	0.787
scFTAT	**0.835**	**0.787**	**0.802**	**0.817**	**0.798**

Table S2. The ablation experiment on the mouse-spleen data

	ACC	F1	Precision	Recall	MCC
Transformer	0.790	0.603	0.671	0.614	0.663
Transformer(K)	0.797	0.685	0.711	0.720	0.680
Transformer(R)	0.784	**0.746**	0.761	0.757	0.657
Transformer(F)	0.802	0.740	0.728	0.769	0.685
scFTAT	**0.810**	0.733	**0.753**	**0.769**	**0.692**

Table S3. The ablation experiment on the human-kidney data

	ACC	F1	Precision	Recall	MCC
Transformer	0.920	0.791	0.829	0.773	0.891
Transformer(K)	0.915	0.817	0.906	0.806	0.884
Transformer(R)	0.923	0.766	0.810	0.777	0.895
Transformer(F)	0.926	**0.823**	**0.921**	0.798	0.899
scFTAT	**0.934**	0.822	0.833	**0.829**	**0.910**

Table S4. The ablation experiment on the human-bladder data

	ACC	F1	Precision	Recall	MCC
Transformer	0.880	0.618	0.718	0.576	0.780
Transformer(K)	0.871	0.605	0.717	0.549	0.763
Transformer(R)	0.884	0.721	0.772	0.697	0.789
Transformer(F)	0.882	0.659	0.713	0.638	0.784
scFTAT	**0.893**	**0.839**	**0.872**	**0.813**	**0.807**

2.2 The summary of the comparative experiments:

Table S5. The experiment on the proposed scFTAT

	ACC	F1	Precision	Recall	MCC
Human-bladder	0.89	0.71	0.80	0.66	0.80
Human-kidney	0.92	0.84	0.96	0.78	0.89
Human-fetal-pancreas	0.93	0.81	0.86	0.79	0.91
Mouse-bladder	0.84	0.79	0.80	0.82	0.80
Mouse-kidney	0.90	0.90	0.91	0.91	0.89
Mouse-spleen	0.81	0.73	0.75	0.77	0.70

Table S6. The experiment on scDeepSort

	ACC	F1	Precision	Recall	MCC
Human-bladder	0.649	0.787	0.649	0.649	0.000
Human-kidney	0.460	0.039	0.459	0.459	0.000
Human-fetal-pancreas	0.498	0.130	0.498	0.498	0.360
Mouse-bladder	0.584	0.132	0.584	0.584	0.464
Mouse-kidney	0.374	0.089	0.374	0.374	0.182
Mouse-spleen	0.558	0.072	0.558	0.558	0.000

Table S7. The experiment on Seurat-PCA

	ACC	F1	Precision	Recall	MCC
Human-bladder	0.467	0.449	0.497	0.467	0.476
Human-kidney	0.668	0.645	0.723	0.668	0.646
Human-fetal-pancreas	0.468	0.399	0.474	0.468	0.458
Mouse-bladder	0.560	0.452	0.560	0.560	0.523
Mouse-kidney	0.639	0.471	0.569	0.639	0.614
Mouse-spleen	0.637	0.534	0.566	0.637	0.576

Table S8. The experiment on PCA-Transformer

	ACC	F1	Precision	Recall	MCC
Human-bladder	0.791	0.253	0.239	0.299	0.641
Human-kidney	0.907	0.507	0.494	0.522	0.875
Human-fetal-pancreas	0.855	0.485	0.480	0.496	0.820
Mouse-bladder	0.841	0.702	0.726	0.697	0.811
Mouse-kidney	0.758	0.493	0.489	0.513	0.729
Mouse-spleen	0.881	0.563	0.591	0.558	0.805

References

1. Rao Y, Zhao W, Zhu Z et al. GFNet: Global Filter Networks for Visual Recognition, IEEE Trans Pattern Anal Mach Intell 2023;45:10960-10973.

2. Bachlechner TC, Majumder BP, Mao HH et al. ReZero is All You Need: Fast Convergence at Large Depth. In: Conference on Uncertainty in Artificial Intelligence. 2020.

3. Guo J, Gao J, Hu Y et al. Robust Adaptive Linear Discriminant Analysis with Bidirectional Reconstruction Constraint, ACM Transactions on Knowledge Discovery from Data 2020;14.

4. Dai Z, Yang Z, Yang Y et al. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. 2019, arXiv:1901.02860.

5. Katharopoulos A, Vyas A, Pappas N, Fleuret F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In: Hal D., III, Aarti S. eds). Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research: PMLR, 2020, 5156--

5165.

6. Huang T, Huang L, You S et al. LightViT: Towards Light-Weight Convolution-Free Vision Transformers. 2022, arXiv:2207.05557.

7. Hu J, Shen L, Albanie S et al. Squeeze-and-Excitation Networks, IEEE Trans Pattern Anal Mach Intell 2020;42:2011-2023.