How Many Patches Is an ECG Signal Worth?

Haitao Li
Zhejiang University
Hangzhou, China
lihaitao@zju.edu.cn

Ziyi Liu †
Transtek Medical Electronics Co., Ltd
Zhongshan, China
11313008@zju.edu.cn

Zhengxing Huang
Zhejiang University
Hangzhou, China
zhengxinghuang@zju.edu.cn

***Abstract*—In recent years, ECG self-supervised learning using masked autoencoder (MAE) has been widely used. Typically, MAE employs the Vision Transformer architecture, which involves splitting the ECG signal into several patches. However, the question, 'How many patches is an ECG signal worth?' has not been fully explored in previous studies. Intuitively, the number of patches is closely related to the sampling duration and frequency of the ECG. But is it true that more patches preserve more details and improve performance? Should the ECG be treated as a one-dimensional time series with 12 channels or a two-dimensional signal with one channel? (corresponding to patch split strategy). Moreover, ECG diagnosis relies on many local waveform features, which Transformer is not good at capturing compared to convolution. Can pure attention mechanisms achieve the same effectiveness as convolution? With these questions, we conducted our research. We designed a family of models using different patch split strategies and tested the models' performance with various patch sizes/numbers. The models were evaluated using the PTB-XL dataset across three dimensions: form, rhythm, and diagnosis, providing comprehensive answers to the previously posed questions.**

***Index Terms*—ECG signal, ViT, Transformer, Patch Split strategy**

I. INTRODUCTION

Electrocardiograms (ECGs) are widely used in clinical practice as a non-invasive diagnostic tool for detecting the heart's electrical activity and managing cardiovascular diseases [1], [2]. Traditional supervised learning methods require a large amount of labeled data, demanding significant expertise and time [3]–[8]. Recently, self-supervised pre-training [9], [10] has mitigated this issue.

Among self-supervised learning methods for ECGs, the masked autoencoder(MAE) [9], [11], [12] is the most prevalent. MAE learns a general representation of ECG signals by masking a portion of the ECG signal and then attempting to recover the masked part using the unmasked portion. These pre-trained representations can be applied to various downstream tasks, significantly reducing the dependence on large amounts of labeled data. Typically, MAE adopts a Transformer [13] architecture, which uses MSA (Multi-Head Self-Attention) and requires splitting the input into several tokens/patches. The Transformer architecture was initially used in the natural language processing (NLP) [13], [14] field and was transferred to other domains like image [15], audio [16], [17] and so on. When transferred to the image domain, research found that compared to traditional convolutional neural networks (CNNs) [18], Transformers require much more data to achieve similar performance due to reduced inductive bias [15], [19]. Meanwhile, extensive research [15], [20], [21] has also been conducted on how to split the image into a series of patches.

However, when researchers transferred the Transformer [13] architecture to ECG, such as the widely used ECG masked autoencoder pre-training [11], [12], [22]–[24] mentioned above, they did not give much consideration to the characteristics of ECG but instead directly reused the architecture of ViT. This straightforward reuse is not entirely convincing. For instance, in the computer vision domain, prior to ViT, when researchers attempted to transfer the Transformer architecture to the image domain, there was no consensus on the size and number of patches. Some operated MSA (Multi-Head Self-Attention) at the pixel level [20], others extracted patches of size 2×2 [21], and later ViT proposed that an image is worth 16x16 words. Additionally, [15] pointed out that due to the lack of translational invariance and locality inductive bias compared to CNNs [18], more data is needed to achieve similar performance [25]–[27], which poses a challenge given the relative scarcity of ECG data compared to image data. Furthermore, the diagnosis of ECGs is even more highly dependent on local waveform features than images [1], which Transformer architecture is not good at. All these mentioned above present challenges for transferring the Transformer architecture to the ECG domain.

In this work, to resolve the aforementioned uncertainties, we specifically explored the following questions: (1) How many patches are required for an ECG, and how should the patch size be designed? Smaller patch sizes result in more patches, but does a higher number of patches retain more information, thus improving performance? What's more, we also examined the impact of ECG sampling duration and frequency on the patch size. (2) How to split the ECG signal into patches? This involves deciding whether to treat the ECG signal as a one-dimensional multi-channel signal or a two-dimensional single-channel signal. (3) Is convolution necessary? ECG diagnosis highly depends on local waveform features [1], [28], which convolution operations capture effectively. Can pure attention mechanisms achieve the same effectiveness as convolution? We conducted a series of experiments on the PTB-XL [29] dataset to explore the aforementioned issues, observing the impact of different model designs on the results from three different granularity dimensions: form, rhythm, and diagnosis.

†Corresponding author

II. METHODOLOGY

To investigate the aforementioned questions, we designed a family of models using different patch split strategies and patch size. Section II-A presents an overview of our model framework, while Section II-B discusses the various patch split strategies employed.

A. Overview

The overview of our model architecture is shown in Figure 1. Overall, the process is divided into two steps. First, the ECG signal is split into a sequence of patches, which will be detailed below in Section II-B. Then, the resulting patch sequence is fed into the Transformer [13] model that employs a pure attention mechanism to generate the output, followed by classification using a linear classification head.

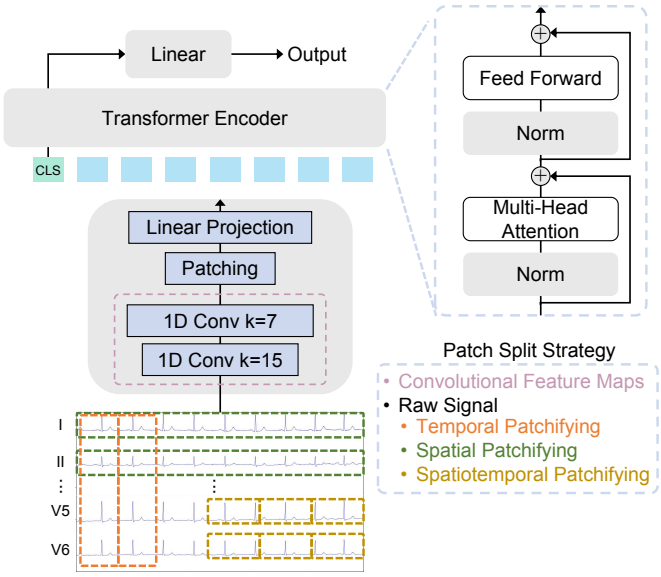

Fig. 1. The overview of our model architecture.

Considering an ECG signal denoted as $X \in \mathbb{R}^{L\times T}$ with L leads and length T = duration $\times$ sampling frequency , we first applied min-max normalization to the signal for each lead to eliminate measurement biases from different instruments and enhance the model's generalization capability. To adapt the Transformer [13] architecture, we employed one of our patch splitting strategies below in Section II-B to generate a sequence of patches, Patch $= \{\text{Patch}_0, ..., \text{Patch}_{n-1}\}$. We also added a [CLS] token at the beginning to capture the global feature. These patches were then added to a learnable positional embedding, Pos $\in \mathbb{R}^D$, to create an embedding sequence $E = \{E_1, ..., E_n\} \in \mathbb{R}^{n\times D}$. This embedding sequence becomes the input for the Transformer encoder.

The Transformer [13] encoder consists of alternating layers of multi-headed self-attention (MSA) and Feed Forward blocks. Layer normalization (LN) [30] is applied before every block, and residual connections are used after every block. The Feed Forward contains two layers with a GELU non-linearity. We used the embedding of [CLS] token from the final layer for linear classification.

$$z_0 = [\mathbf{CLS}; \mathbf{E}_1; \mathbf{E}_2; \cdots; \mathbf{E}_n], \mathbf{E}_i \in \mathbb{R}^D, \quad (1)$$

$$z'_\ell = \text{MSA}(\text{LN}(z_{\ell-1})) + z_{\ell-1}, \quad \ell = 1, \ldots, L \quad (2)$$

$$z_\ell = \text{Feed Froward}(\text{LN}(z'_\ell)) + z'_\ell, \quad \ell = 1, \ldots, L \quad (3)$$

$$\mathbf{y} = \text{Linear}(\text{LN}(z^0_L)) \quad (4)$$

For the specific implementation, the model dimension (model_dim) varies depending on the patch split strategy and patch size. We used 8 attention heads, with the dimension of each head given by dim_head $= \frac{\text{model\_dim}}{8}$. The MLP dimension is calculated as mlp_dim $= 2 \times$ model_dim.

B. Patch Split Strategy

In this section, we provide a detailed explanation of how the ECG signal $X \in \mathbb{R}^{L\times T}$ is split into a sequence of patches, Patch $= \{\text{Patch}_0, ..., \text{Patch}_{n-1}\}$. Our strategies are generally divided into two main categories: Raw Signal Patchifying and Convolutional Feature Maps Patchifying. Raw Signal Patchifying, as the name suggests, directly splits the original ECG signal into patches. Convolutional Feature Maps Patchifying, on the other hand, involves patchifying the ECG feature maps obtained using 1D convolution with the consideration that ECG diagnosis heavily relies on local waveform features. We will discuss these two strategies in detail separately below.

1) Raw Signal Patchifying: Before delving into the specifics, we need to consider a question: should we treat the ECG as a one-dimensional time series with 12 channels, or as a two-dimensional signal with one channel? Different understandings of the ECG signal correspond to different patch split strategies, such as spatial, temporal, and spatiotemporal patchifying, as illustrated in Figure 1.

Spatial Patchifying, as the name suggests, divides the ECG signal into $n = 12$ patches at the spatial (i.e., different leads) level, with each patch containing the complete information of each lead.

$$\text{Patch}_i = X[i], \quad i = 0, 1, \ldots, 11$$

Temporal Patchifying views the ECG as a one-dimensional time series with 12 channels, thus splitting the ECG into n non-overlapping patches along the time dimension, where n is determined by the patch size, given patch size $(12, p)$, n is calculated as T/p. The whole process can be expressed as follows, where 'Flatten' denotes converting into a vector.

$$\text{Patch}_i = \text{Flatten}(X[:, i*p : (i+1)*p]), \quad i = 0, 1, \ldots, n-1$$

Spatiotemporal Patchifying views the ECG as a two-dimensional signal with one channel, using a patchifying method similar to ViT [15]. Specifically, each lead is individually patched, with each patch containing a segment of information from each lead, given patch size $(1, p)$, the number of patches is calculated as $n = 12 * T/p$ with each lead contributing $k = T/p$ patches. The whole process can be

TABLE I
RESULTS OF RAW SIGNAL PATCHIFYING

Patch Split Strategy	Patch Num.	Patch Size	Duration per Patch	Model Dim	Form	Rhythm	Diagnosis	Avg.
Spatial	12	(1, 1000)	10s	1024	58.32	58.33	57.55	58.07
Temporal	10	(12, 100)	1s	1024	59.88	59.56	74.66	64.70
	25	(12, 40)	0.4s	512	61.60	59.60	69.17	63.46
	50	(12, 20)	0.2s	256	63.34	67.53	79.61	70.16
	100	(12, 10)	0.1	128	71.94	73.24	86.80	77.33
	250	(12, 4)	0.04s	64	67.50	83.96	85.56	79.01
	500	(12, 2)	0.02s	32	70.04	81.67	84.21	78.64
	1000	(12, 1)	0.01s	16	64.09	77.63	81.55	74.42
Spatiotemporal	12*1=12	(1, 1000)	10s	1024	58.32	58.33	57.55	58.07
	12*2=24	(1, 500)	5s	512	57.60	59.86	60.24	59.23
	12*5=60	(1, 200)	2s	256	58.73	72.25	67.42	66.13
	12*10=120	(1, 100)	1s	128	66.74	84.26	83.07	78.02
	12*20=240	(1, 50)	0.5s	64	72.33	88.24	86.29	82.29
	12*50=600	(1, 20)	0.2s	32	69.13	82.64	85.54	79.10
	12*100=1200	(1, 10)	0.1s	16	54.64	74.92	79.01	69.52

expressed as follows, where $\lfloor \cdot \rfloor$ denotes the floor operation, and mod denotes the modulo operation.

$$\text{Patch}_i = X[\lfloor \frac{i}{k} \rfloor, (i \bmod k) * p : ((i \bmod k) + 1) * p], \quad i = 0, 1, \ldots, n-1$$

It is worth noting that Spatial Patchifying can be considered a special case of Spatiotemporal Patchifying when the patch size is set to $(1, T)$. Therefore, in the following experiments, Spatial Patchifying will not be discussed separately.

2) Convolutional Feature Maps Patchifying: As discussed above, ECG diagnosis heavily relies on local waveform features [1], which Transformers struggle to capture due to the lack of locality inductive bias compared to CNNs [25]–[27]. Additionally, the relatively small size of the ECG dataset [31] further limits the model to learn this capability through data-driven methods. To address this, we propose splitting the ECG feature maps obtained from 1D convolution into patches, leveraging CNNs' strength in capturing local features and the Transformer's ability to model long temporal sequences, both critical for ECG diagnosis.

Specifically, given the raw ECG signal, we first apply two 1D convolutions with kernel sizes of 15 and 7, respectively, followed by batch normalization and ReLU activation to obtain the feature maps, which are then divided into patches. In Raw Signal Patchifying, ECG patches can be generated in temporal, spatial, or spatiotemporal ways. However, since the 1D convolution treats the ECG signal as a multi-channel 1D signal, to maintain consistency, we used the temporal patch method. Specifically, the feature maps X after convolution are divided into non-overlapping patches defined as $\text{Patch} = \{\text{Patch}_0, \ldots, \text{Patch}_{n-1}\}$, where n is determined by T/p, and p represents the size of each patch.

III. EXPERIMENTS

To address the three questions raised at the end of Section I, we designed a series of models above. Next, we will validate the performance of these models on the PTB-XL [29] dataset to answer these three questions. Unless otherwise specified, all results reported are macro-AUC.

A. Dataset

The PTB-XL dataset [29] contains 21,837 ECG signals from 18,885 patients, with each 12-lead ECG sampled at 100/500 Hz over 10 seconds. The dataset is divided into three multi-label classification subsets: Diagnosis (44 categories), Form (19 categories), and Rhythm (12 categories). We use the official train:validation:test split. These three subsets evaluate the model at different granular levels: the Form subset captures fine-grained features, the Rhythm subset focuses on global features, and the Diagnosis subset requires both. Unless otherwise specified, all experiments use 100Hz ECG signals, denoted as $X \in \mathbb{R}^{12\times1000}$

B. How to Patch?

To address the first two questions in Section I, we experimented with the three Raw Signal Patchifying methods across various patch sizes. The results are shown in Table I.

Question 1: 'How many patches are needed for an ECG, and how should the patch size be designed?'. Regardless of the patching strategy, the model's performance in form, rhythm, and diagnosis initially improves and then declines as the number of patches increases, with optimal performance around 250 patches. Thus, approximately 250 patches are required for an ECG. While ViT suggests that an image requires $16 \times 16 = 256$ patches, ECG signals, due to their periodic nature, seem to require fewer patches than images. Later, we found that using Convolutional Feature Maps Patchifying significantly reduces the required number of patches.

Question 2: 'How to split the ECG signal into patches?', i.e. which patching strategy to choose. Since Spatial Patchifying is a special case of Spatiotemporal Patchifying, we focus on the remaining two methods. Temporal Patchifying requires the patch number to be a multiple of 12, resulting in slight

differences in patch numbers between the two strategies but still comparable(e.g., 10 vs 12). In Table I, red/green highlights the superior method under similar computational complexity.

It can be observed that Temporal Patchifying generally performs better in form and diagnosis, while Spatiotemporal Patchifying excels in rhythm analysis. The advantage of Temporal Patchifying in form reflects its effectiveness at capturing local features. This can be inferred from the shorter duration of each patch; specifically, each patch duration in Temporal Patchifying is one-twelfth that of Spatiotemporal Patchifying under the same patch number, allowing the model to detect more transient waveform features. This also explains its superior performance in diagnosis, as ECG interpretation relies heavily on local waveform features. In contrast, Spatiotemporal Patchifying better captures rhythm by integrating time and spatial (lead) dimensions, crucial for identifying subtle rhythm changes and lead synchronization.

Although each strategy has merits, Spatiotemporal Patchifying consistently outperforms Temporal Patchifying across form, rhythm, and diagnosis when using the optimal patch number, around 250. Thus, for Raw Signal Patchifying, we recommend the spatiotemporal approach.

C. Is Convolution Necessary?

Question 3: 'Is convolution necessary?'. As discussed earlier, to address the Transformer's limitations in capturing local waveform features, we developed the Convolutional Feature Maps Patchifying. It is essentially a variant of Temporal Patchifying. We compared their performance under the same patch number, with the results presented in Table II.

TABLE II
RESULTS OF CONVOLUTIONAL FEATURE MAPS PATCHIFYING.

Patch Num.	Form	Rhythm	Diagnosis	Avg.	**Improve**
10	61.40	77.85	71.92	70.39	5.69
25	60.92	69.66	79.59	70.06	6.60
50	67.15	68.25	83.76	73.05	2.89
100	76.67	89.93	90.82	85.81	8.48
250	74.85	89.60	90.46	84.97	5.96
500	73.45	89.04	89.11	83.87	5.23
1000	67.19	86.17	85.03	79.46	5.04

Improve: Improvement compared to Temporal Patchifying.

It is observed that: (1) Convolutional Feature Maps Patchifying shows a significant improvement over Temporal Patchifying with the same number of patches. This supports our hypothesis that incorporating convolution compensates for the Transformer's lack of locality inductive bias, which is crucial for ECG diagnosis and significantly enhances performance. (2) The optimal patch number dropped from 250 in Raw Signal Patchifying to 100, greatly improving computational efficiency, particularly in large-scale self-supervised pre-training.

D. The Impact of Sampling Duration and Frequency

In all experiments above, we used 10-second, 100Hz ECG signals denoted as $X \in \mathbb{R}^{L \times T}$ with L leads. The length of an ECG sequence, T = duration $\times$ sampling frequency. Therefore, variations in duration and sampling frequency will affect the optimal number of ECG patches.

Firstly, regarding the impact of duration on the ECG patch number: Although ECGs of different lengths will have different optimal patch numbers, they should share the same optimal patch size at the same sampling frequency.

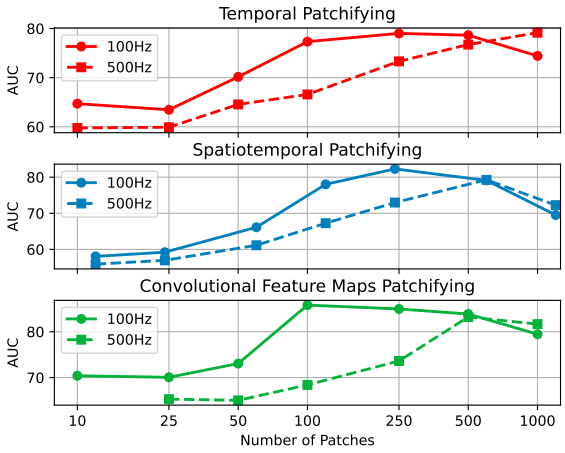

Fig. 2. The Impact of Sampling Frequency.

Next, the impact of sampling frequency: Although the ECG's sampling frequency can be adjusted to any value through post-processing, commonly used frequencies are 100Hz and 500Hz. We repeated all experiments above using ECG signals sampled at 500Hz, and the results are shown in Figure 2. It can be observed that as the sampling frequency increases, the optimal number of patches changes. For Temporal Patchifying, the optimal patch number for the 100Hz ECG is around 250, while for the 500Hz ECG, the model continues to improve as the number of patches increases up to 1000. For Spatiotemporal Patchifying, the optimal patch number rises from 250 to approximately 500. For Convolutional Feature Maps Patchifying, the optimal patch number increases from 100 to 500. Additionally, it is notable that regardless of the patch split strategy used, the 100Hz ECG achieves higher optimal performance with fewer patches. Therefore, although the sampling frequency affects the optimal number of patches, we recommend using 100Hz ECG signals for Transformer-based models in practice to optimize efficiency and performance.

IV. CONCLUSION

In this work, we aim to answer the questions: 'How many patches is an ECG signal worth?' and 'How to split the ECG signal into patches'. To address these, we designed two major patch split strategies: Raw Signal Patchifying and Convolutional Feature Maps Patchifying. We conducted experiments using a range of patch sizes/numbers on the PTB-XL dataset, evaluating our model across three granular dimensions: form, rhythm, and diagnosis. We identified the optimal patch number and patch split strategy, emphasizing the importance of convolutional operations in ECG diagnosis. Finally, we explored the effects of sampling duration and frequency on the patch number.

REFERENCES

[1] R. Vecht, M. A. Gatzoulis, and N. Peters, *ECG diagnosis in clinical practice*. Springer Science & Business Media, 2009.

[2] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu, "A survey on ecg analysis," *Biomedical Signal Processing and Control*, vol. 43, pp. 216–235, 2018.

[3] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ecg classification by 1-d convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2015.

[4] U. R. Acharya, H. Fujita, S. L. Oh, U. Raghavendra, J. H. Tan, M. Adam, A. Gertych, and Y. Hagiwara, "Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network," *Future Generation Computer Systems*, vol. 79, pp. 952–959, 2018.

[5] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. San Tan, "A deep convolutional neural network model to classify heartbeats," *Computers in biology and medicine*, vol. 89, pp. 389–396, 2017.

[6] Ö. Yildirim, "A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification," *Computers in biology and medicine*, vol. 96, pp. 189–202, 2018.

[7] J. H. Tan, Y. Hagiwara, W. Pang, I. Lim, S. L. Oh, M. Adam, R. San Tan, M. Chen, and U. R. Acharya, "Application of stacked convolutional and long short-term memory network for accurate identification of cad ecg signals," *Computers in biology and medicine*, vol. 94, pp. 19–26, 2018.

[8] L. Guo, G. Sim, and B. Matuszewski, "Inter-patient ecg classification with convolutional and recurrent neural networks," *Biocybernetics and Biomedical Engineering*, vol. 39, no. 3, pp. 868–879, 2019.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[11] R. Hu, J. Chen, and L. Zhou, "Spatiotemporal self-supervised representation learning from multi-lead ecg signals," *Biomedical Signal Processing and Control*, vol. 84, p. 104772, 2023.

[12] H. Zhang, W. Liu, J. Shi, S. Chang, H. Wang, J. He, and Q. Huang, "Maefe: Masked autoencoders family of electrocardiogram for self-supervised pretraining and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–15, 2022.

[13] A. Vaswani, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[14] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] A. DOSOVITSKIY, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[17] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in neural information processing systems*, vol. 34, pp. 12 116–12 128, 2021.

[20] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[21] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," *arXiv preprint arXiv:1911.03584*, 2019.

[22] Y. Na, M. Park, Y. Tae, and S. Joo, "Guiding masked representation learning to capture spatio-temporal relationship of electrocardiogram," *arXiv preprint arXiv:2402.09450*, 2024.

[23] J. Jin, H. Wang, J. Li, S. Huang, J. Pan, and S. Hong, "Reading your heart: Learning ecg words and sentences via pre-training ecg language model," in *Artificial Intelligence and Data Science for Healthcare: Bridging Data-Centric AI and People-Centric Healthcare*.

[24] J. Chen, W. Wu, and S. Hong, "Multi-channel masked autoencoder and comprehensive evaluations for reconstructing 12-lead ecg from arbitrary single-lead ecg," in *Artificial Intelligence and Data Science for Healthcare: Bridging Data-Centric AI and People-Centric Healthcare*, 2024.

[25] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in neural information processing systems*, vol. 34, pp. 28 522–28 535, 2021.

[26] Q. Zhang, Y. Xu, J. Zhang, and D. Tao, "Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond," *International Journal of Computer Vision*, vol. 131, no. 5, pp. 1141–1162, 2023.

[27] N. Park and S. Kim, "How do vision transformers work?" *arXiv preprint arXiv:2202.06709*, 2022.

[28] F. G. Yanowitz, "Introduction to ecg interpretation," *LDS Hospital and Intermountain Medical Center*, 2012.

[29] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "Ptb-xl, a large publicly available electrocardiography dataset," *Scientific data*, vol. 7, no. 1, p. 154, 2020.

[30] J. Ba, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[31] E. Merdjanovska and A. Rashkovska, "Comprehensive survey of computational ecg analysis: Databases, methods and applications," *Expert Systems with Applications*, vol. 203, p. 117206, 2022.