BEAT-SYNCHRONOUS TOKENIZATION FOR ECG TRANSFORMERS

Anonymous

Anonymous

ABSTRACT

Transformer-based electrocardiogram (ECG) models commonly tokenize waveforms into fixed temporal patches. Though convenient, fixed patching can split heartbeat structures across token boundaries. We study beat-synchronous tokenization as a physiologically grounded alternative, comparing fixed patches with three beat-aligned strategies: resampled beats, adaptive pooled beats, and resampled beats augmented with R–R interval information. Experiments span two settings: 10-second 12-lead diagnostic classification on PTB-XL after MIMIC-IV-ECG masked pretraining, and 60-second single-lead rhythm classification on Icentia11k after patient-level contrastive pretraining. On PTB-XL, resampled beat tokens achieve the highest mean macro Area Under the ROC Curve (AUROC; 0.8945) and nearly match the best fixed-patch macro Area Under the Precision-Recall Curve (AUPRC; 0.7414), reducing average sequence length from 100 to 11.2 tokens. On Icentia11k, beat-synchronous tokenizers obtain comparable AUPRC to fixed patching with better stability across runs. These results suggest morphology-preserving beat tokenization is a compact, competitive alternative to fixed temporal patching.

***Index Terms*—** Electrocardiography, ECG Transformers, tokenization, beat-synchronous tokenization, self-supervised learning, representation learning, token efficiency

1. INTRODUCTION

Electrocardiography (ECG) provides a compact, non-invasive view of cardiac electrical activity and remains central to cardiovascular screening, diagnosis, and monitoring. Deep learning has substantially improved automated ECG analysis across clinical prediction tasks [1, 2, 3]. However, fully supervised ECG models often depend on large expert-labeled datasets, which are costly to curate and difficult to scale across institutions, devices, and patient populations. Self-supervised learning (SSL) addresses this bottleneck by pretraining ECG encoders on unlabeled recordings and transferring the learned representations to downstream tasks [4, 5]. This direction has recently expanded toward ECG foundation models trained at large scale for broad transfer across clinical settings [6, 7].

Anonymous.

Despite these advances, many Transformer-based ECG models still represent the waveform as a sequence of fixed temporal patches. Fixed patching is simple and compatible with common SSL objectives such as masked reconstruction [8] or contrastive learning [9, 4], but it treats ECGs as generic time series rather than structured cardiac recordings. A fixed patch may split a cardiac cycle across token boundaries or mix adjacent beats, even though ECG interpretation often depends on beat morphology, inter-beat timing, and rhythm-level organization. Since heart rate varies across patients and recording conditions, the same patch length can correspond to different physiological content across examples.

Recent ECG models have begun to replace fixed temporal patches with beat-aware or physiologically structured representations [10, 11, 12, 13, 14]. These studies support the value of aligning ECG representations with cardiac structure, but tokenization is often introduced together with other modeling changes, such as specialized architectures, discrete vocabularies, or ECG-text training. However, the effect of tokenization alone on performance and efficiency remains unclear.

In this work, we test the hypothesis that beat-synchronous tokens are more efficient alternatives to fixed temporal patches under matched Transformer pretraining and downstream evaluation protocols. We compare fixed temporal patches with three beat-synchronous tokenizers: resampled beat tokens, adaptive pooled beat tokens, and resampled beat tokens augmented with R–R interval information. This setup allows us to evaluate whether cardiac-cycle alignment improves the performance–efficiency tradeoff while keeping the encoder family and evaluation protocol controlled.

We evaluate our hypothesis in two complementary settings. First, we pretrain 12-lead ECG Transformers on MIMIC-IV-ECG [15] using the masked reconstruction SSL objective and evaluate transfer to PTB-XL [16] diagnostic classification on five superclasses. Second, we pretrain single-lead models on the Icentia11k dataset using a contrastive learning objective and evaluate 60-second dominant rhythm classification [17]. These settings test beat-synchronous tokenization on both standard 10-second clinical ECGs and longer-term ambulatory recordings.

Our results show that beat-synchronous tokenization can retain strong downstream performance while using substantially shorter token sequences. On PTB-XL, the resampled

beat tokenizer achieves the highest mean macro AUROC, while the R–R augmented tokenizer remains close to the strongest fine-patch baseline; both use only 11.2 tokens on average compared with 100 tokens for the finest fixed-patch tokenizer. The adaptive-pooling variant performs substantially worse, indicating that beat alignment alone is insufficient without a morphology-preserving beat encoder. On Icentia11k, where atrial fibrillation/atrial flutter (AFib/AFL) windows are rare, beat-synchronous tokenizers obtain comparable AUPRC to fixed patching and show lower run-to-run AUPRC variability, while using approximately 68 beat tokens on average compared with 93 fixed tokens. Overall, these findings suggest that beat-synchronous tokenization can be an effective and token-efficient alternative to fixed temporal patching, but the design of the beat encoder is critical.

2. RELATED WORK

Self-supervised ECG representation learning: SSL has become a common strategy for learning ECG representations from large unlabeled datasets. Contrastive approaches define positive pairs across time, leads, or patients to encourage clinically useful invariances [9, 4], while reconstruction-based methods train encoders to recover masked ECG segments or patches [8]. Benchmarks on 12-lead ECGs show that SSL can approach the performance of fully supervised learning with reduced label dependence [5], and recent ECG foundation models scale pretraining to larger datasets and broader transfer settings [6, 7]. Most of these pipelines, however, represent the waveform using fixed temporal windows or patches, which are convenient for Transformer encoders but not explicitly aligned with cardiac cycles.

ECG tokenization and ECG-language modeling: Recent work has explored more structured ECG tokenization. HeartLang treats heartbeats as words and rhythms as sentences, using QRS-aligned ECG sentences and vocabulary-based pretraining [10]. RhythmBERT further develops this idea by tokenizing P, QRS, and T waveform components into symbolic representations [11]. These approaches support the idea that ECGs contain natural physiological units that may be better suited to sequence modeling than arbitrary time patches. At the same time, their tokenization choices are embedded within larger systems involving discrete vocabularies, clustering, wave segmentation, or specialized pretraining objectives. Our work is complementary: rather than proposing a full ECG-language framework, we directly compare fixed temporal patches against beat-synchronous continuous tokenizers under matched Transformer settings.

Beat-aware and rhythm-focused modeling: Beat-level representations have also been studied in supervised and rhythm-monitoring contexts. Patient-adaptive beat-wise Transformers use beat tokens with patient-specific symbolic morphology for atrial fibrillation detection in long-term monitoring [12]. BEAT-Net uses QRS-aligned tokens in a supervised model designed to model morphology, lead-specific spatial information, and temporal rhythm structure [13]. MELP incorporates beat-level information into ECG-text pretraining through token-, beat-, and rhythm-level alignment [14]. While these studies indicate that beat-level structure is useful, they do not establish whether beat-synchronous token boundaries alone improve the performance–efficiency tradeoff.

Position of this work: Prior work motivates physiologically structured ECG representations, but the effect of tokenization is often entangled with other modeling choices. We therefore focus on a controlled comparison: fixed temporal patches versus beat-synchronous tokens, using comparable Transformer encoders, SSL objectives, and downstream evaluation protocols. By testing resampled beats, adaptive pooled beats, and R–R augmented beats, we not only evaluate whether beat alignment helps, but also which beat-token designs preserve morphology and rhythm information effectively.

3. METHODOLOGY

Fig. 1 summarizes the main difference between fixed temporal patching and beat-synchronous tokenization. We study ECG tokenization as an isolated design choice by keeping the Transformer-style encoder and downstream evaluation protocol comparable across tokenizers. Given an ECG segment X, each tokenizer maps the waveform to a token sequence $\{z_i\}_{i=1}^{N}$, which is processed by a Transformer encoder and pooled into a segment-level representation. We compare fixed temporal patches with three beat-synchronous tokenizers.

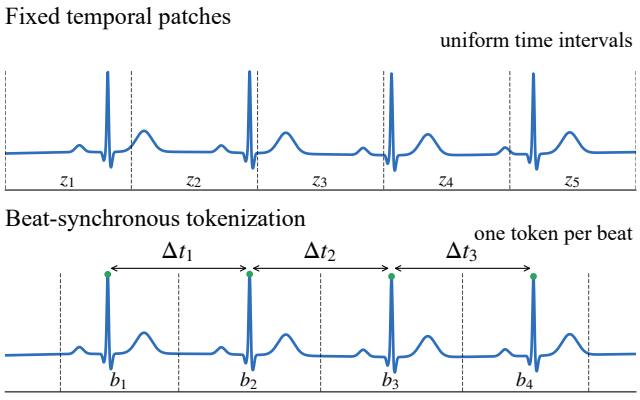

Fig. 1. Fixed temporal patching divides the ECG into uniform time intervals, whereas beat-synchronous tokenization defines tokens from cardiac cycles identified by R-peaks. Δt denotes the R–R interval between consecutive R-peaks, which is added as timing information in Tok3.

Fixed temporal patch tokenizer: The fixed tokenizer partitions the ECG into non-overlapping temporal patches of length p and embeds each patch using a one-dimensional convolution with kernel size and stride p. For 10-second 12-lead

ECGs sampled at 500 Hz, we evaluate $p \in \{50, 100, 250, 500\}$, corresponding to 100, 50, 20, and 10 tokens per recording. For the 60-second Icentia11k setting sampled at 250 Hz, we use $p = 160$, yielding 93 fixed tokens per window.

Beat-synchronous tokenizers: Beat-synchronous tokenization uses consecutive R-peaks to define cardiac-cycle tokens. Let r_i and r_{i+1} be adjacent R-peak locations. The i-th beat is extracted from the interval $[r_i, r_{i+1})$, so the token sequence length is determined by the number of detected beats rather than a fixed time grid. We evaluate three beat-token designs.

Tok1 - resampled beat tokens: Each variable-length beat is resampled to a fixed length and embedded by a one-dimensional convolution. In the 12-lead MIMIC/PTB-XL setting, each beat is represented as a 12×300 segment. In the single-lead Icentia setting, each beat is represented as a 1×160 segment.

Tok2 - adaptive pooled beat tokens: Tok2 preserves beat boundaries but avoids explicit temporal resampling. Variable-length beat segments are passed through a convolutional front end and compressed to one token using adaptive average pooling. This tokenizer tests whether beat alignment alone is sufficient when beat morphology is aggressively compressed.

Tok3 - resampled beat tokens with R–R timing: Tok3 uses the same resampled beat representation as Tok1, but additionally embeds the R–R interval associated with each beat using a small multilayer perceptron. The R–R embedding is added to the beat embedding before the Transformer encoder, providing explicit rhythm timing information.

Encoder architecture: All tokenizers are paired with Transformer encoders using hidden dimension $d = 256$, 8 attention heads, GELU activations, pre-layer normalization, and dropout 0.1. For the MIMIC/PTB-XL experiments, we use 4 Transformer layers. For the Icentia experiments, we use 6 Transformer layers. Fixed-patch encoders use positional encodings over fixed token sequences. Beat-synchronous encoders use padding masks because the number of beats varies across recordings; valid token representations are mean-pooled after the Transformer layers.

Self-supervised objectives: We use the SSL objective that best matches each experimental setting: masked reconstruction for 10-second 12-lead diagnostic transfer, where preserving waveform morphology is central, and patient-level contrastive learning for Icentia11k, where long ambulatory recordings allow positive pairs to be sampled from different windows of the same patient. All SSL models were trained until convergence rather than for a fixed update budget, so that differences between tokenizers were not driven by unequal pretraining progress.

Masked reconstruction pretraining for 12-lead ECG: For the MIMIC-IV-ECG experiments, we pretrain each tokenizer using masked reconstruction. A subset of tokens is replaced by a learned mask token, the Transformer processes the masked sequence, and a prediction head reconstructs the corresponding raw waveform patch or beat segment. Fixed-patch models reconstruct fixed temporal patches of dimension $12p$, while beat-synchronous models reconstruct beat-level targets. We use a mask ratio of 0.5 in all our experiments. Pretraining uses AdamW with learning rate 2×10^{-5}, weight decay 0.05, batch size 256, and 20 epochs.

Patient-level contrastive pretraining for Icentia11k: We pretrain a 60-second single-lead fixed-patch encoder using patient-level contrastive learning. Each positive pair consists of two different 60-second windows sampled from the same patient, while windows from other patients in the minibatch are negatives. The encoder is trained with an InfoNCE-style contrastive objective [18]. In the downstream Icentia comparison, the fixed-patch model uses this pretrained encoder directly. The beat-synchronous and beat-synchronous+HR models initialize shared convolutional and Transformer weights from the same fixed-patch checkpoint by applying the learned 160-sample convolution to resampled beat windows; Tok3 additionally learns the R–R interval encoder during downstream training. This design makes the experiment a direct comparison of downstream tokenization choices under the same pretrained initialization. We evaluate Tok1 and Tok3 in this setting because Tok2 performed substantially worse in the PTB-XL experiment, indicating that adaptive pooling was not a competitive beat-token design.

4. EXPERIMENTAL SETUP

MIMIC-IV-ECG pretraining data: We use MIMIC-IV-ECG as the unlabeled 12-lead pretraining corpus [15]. Records are represented as 10-second, 12-lead ECGs sampled at 500 Hz, giving tensors of shape 12×5000. Invalid values are sanitized by removing highly corrupted records, interpolating minor missing segments, clipping amplitudes to $[-5, 5]$, and applying per-lead z-score normalization. For beat-synchronous pretraining, R-peaks are detected from lead II, and consecutive R-peaks define beat intervals. We use a maximum of 35 beats for padding and positional encoding.

PTB-XL downstream task: We evaluate 12-lead transfer on PTB-XL five-superclass diagnostic classification [16]. The Standard Communication Protocol (SCP) codes are mapped to the standard diagnostic superclasses: NORM (Normal ECG), MI (Myocardial Infarction), STTC (ST/T-Change), CD (Conduction Disturbance), and HYP (Hypertrophy). We follow the official PTB-XL stratified split: folds 1–8 for training, fold 9 for validation, and fold 10 for testing. Records without any superclass label or missing waveform files are removed. Each model is fine-tuned for multi-label classification with a five-output prediction head. We select checkpoints by validation macro AUPRC and report test macro AUROC and macro AUPRC over five runs.

PTB-XL token counts: For fixed patches, the token count is determined by the patch size: $p = 50$ gives 100 tokens, $p = 100$ gives 50 tokens, $p = 250$ gives 20 tokens, and $p = 500$ gives 10 tokens. For beat-synchronous tokenizers, the sequence length is determined by detected cardiac

cycles. On the PTB-XL training set, beat tokenization yields 11.2 beats per record on average, with median 11, interquartile range 10–12, 90th percentile 14, and 95th percentile 16. The maximum padding length of 35 covers all records.

Icentia11k pretraining and downstream splits: We use Icentia11k [17] for long-context single-lead rhythm evaluation. The dataset is sampled at 250 Hz and contains 11,000 patients. We split patients into 8,800 SSL pretraining patients, 1,100 supervised training patients, 550 validation patients, and 550 test patients. All splits are patient-level.

Icentia preprocessing and labeling: ECG records are bandpass filtered between 0.5 and 40 Hz using a zero-phase Butterworth filter and normalized by per-window z-scoring. We use 60-second windows, corresponding to 15,000 samples. Rhythm annotations include normal sinus rhythm (NSR), atrial fibrillation (AFib), and atrial flutter (AFL). Those annotations are converted into non-overlapping intervals. A window is retained only if the annotated rhythm coverage and dominant rhythm occupancy both satisfy a 90% purity threshold. Labels are binarized as N versus AFib/AFL, with AFib and AFL mapped to the positive class. Beat-synchronous models utilize labeled beat annotations as R-peak locations, keep valid beats (including normal and ectopic beats), and exclude unclassified beats.

Icentia evaluation manifests: To ensure fair comparison, validation and test windows are generated once as frozen manifests and reused for all tokenizers. Thus, fixed patching, Tok1, and Tok3 are evaluated on the same patients, records, window start times, and labels. The final validation set contains 1,088 windows, with 83 AFib/AFL positives (7.6%) and 1,005 N windows (92.4%). The test set contains 1,082 windows, with 80 AFib/AFL positives (7.4%) and 1,002 N windows (92.6%). During supervised training, windows are sampled with a balanced class probability (i.e., $p(\mathrm{AFib/AFL}) = 0.5$), whereas validation and test sets use the natural class prevalence.

Icentia token counts: For fixed patching, $p = 160$ gives 93 fixed tokens per 60-second window. For beat-synchronous tokenization, the token count varies with heart rate. Across valid 60-second Icentia windows, the mean number of beats is 68.1, the median is 67, the interquartile range is 59–76, and the 95th percentile is 93. We therefore set the maximum beat sequence length to 93, which covers approximately 95.4% of valid windows.

Icentia downstream training and metrics: The Icentia downstream task is binary classification of N versus AFib/AFL. Models are trained with cross-entropy loss using AdamW, with learning rate 10^{-4} for encoder parameters, 10^{-3} for the classifier, and weight decay 10^{-4}. Each model is trained for 10 epochs with batch size 64, and the best checkpoint is selected by validation AUPRC. We report AUROC and AUPRC on the held-out test set, emphasizing AUPRC because AFib/AFL windows are rare [19].

Data and code availability: The data used in this study are publicly available from MIMIC-IV-ECG [15], PTB-XL [16], and Icentia11k [17]. The code and pretrained models will be released upon acceptance of this manuscript.

Table 1. PTB-XL five-superclass classification after MIMIC-IV-ECG masked pretraining. Results are shown as mean ± standard deviation over five runs. Token counts are provided for a 10-second ECG.

Tokenizer	Tokens	Macro AUROC	Macro AUPRC
Fixed CNN, $p = 50$	100	0.8903 ± 0.0014	$\mathbf{0.7419 \pm 0.0025}$
Fixed CNN, $p = 100$	50	0.8858 ± 0.0007	0.7345 ± 0.0043
Fixed CNN, $p = 250$	20	0.8717 ± 0.0008	0.7033 ± 0.0017
Fixed CNN, $p = 500$	10	0.8479 ± 0.0019	0.6530 ± 0.0055
Tok1, resampled beats	11.2 avg.	$\mathbf{0.8945 \pm 0.0012}$	0.7414 ± 0.0037
Tok2, adaptive pooled beats	11.2 avg.	0.8276 ± 0.0034	0.6328 ± 0.0072
Tok3, resampled beats + R–R	11.2 avg.	0.8928 ± 0.0015	0.7399 ± 0.0039

5. RESULTS AND DISCUSSION

PTB-XL diagnostic classification: Table 1 shows the main 12-lead diagnostic transfer results on PTB-XL. Among fixed temporal patch tokenizers, the finest patch size, $p = 50$, provides the best performance, achieving 0.8903 macro AUROC and 0.7419 macro AUPRC. Performance decreases as the fixed patch size increases, indicating that coarse fixed patches lose diagnostically useful waveform detail. In contrast, Tok1 achieves the highest mean macro AUROC (0.8945) and nearly matches the best fixed-patch macro AUPRC, with a negligible absolute difference from fixed $p = 50$. Tok3 also performs close to fixed $p = 50$, reaching 0.8928 macro AUROC and 0.7399 macro AUPRC.

Token efficiency: The PTB-XL results highlight the main efficiency advantage of beat-synchronous tokenization. Fixed $p = 50$ uses 100 tokens for each 10-second ECG, whereas Tok1 and Tok3 use 11.2 beat tokens on average. Thus, the beat-synchronous models obtain comparable or slightly better AUROC and nearly identical AUPRC while reducing the average sequence length by almost an order of magnitude. This matters for Transformer-based ECG modeling because self-attention has quadratic complexity in the number of tokens [20]. Although fixed $p = 500$ uses a similar number of tokens to the beat-synchronous models, its performance is much lower, suggesting that token count alone does not explain the result. Beat-synchronous tokens preserve a physiologically meaningful unit of analysis while remaining compact.

Beat alignment alone is not sufficient: Tok2 performs substantially worse than Tok1 and Tok3, despite using the same beat boundaries. This suggests that the benefit of beat-synchronous tokenization depends on how each beat is encoded. Adaptive pooling compresses variable-length beats into a single token after a convolutional front end, but this aggressive compression appears to discard morphology needed for downstream diagnosis. In contrast, Tok1 and Tok3 resample each cardiac cycle to a fixed length before convolutional

Table 2. Icentia11k 60-second NSR vs. AFib/AFL classification after patient-level contrastive pretraining. Results are mean ± standard deviation over five runs.

Tokenizer	Tokens	AUROC	AUPRC
Fixed CNN, $p = 160$	93	$\mathbf{0.9888 \pm 0.0031}$	0.8514 ± 0.0676
Tok1, resampled beats	68.1 avg.	0.9715 ± 0.0050	0.8514 ± 0.0076
Tok3, resampled beats + R–R	68.1 avg.	0.9669 ± 0.0065	$\mathbf{0.8515 \pm 0.0202}$

embedding, preserving more within-beat structure. Therefore, the main conclusion is not simply that "beats are better," but that beat-synchronous tokens must preserve morphology and, when relevant, timing information.

Icentia11k long-context rhythm classification: Table 2 reports the 60-second Icentia11k rhythm classification results. Fixed patching obtains the highest AUROC, 0.9888, indicating stronger overall ranking performance in this single-lead rhythm setting. However, AFib/AFL windows are rare in the held-out test set, comprising only 7.4% of examples, making AUPRC especially important [19]. Under this metric, Tok1 and Tok3 achieve essentially the same mean AUPRC as fixed patching while using fewer tokens on average. Tok1 reaches 0.8514 AUPRC, and Tok3 obtains the highest mean AUPRC, 0.8515, although the difference is negligible. These results suggest that beat-synchronous tokenization remains competitive for imbalanced long-context rhythm classification.

Stability under class imbalance: The Icentia11k results also show a notable difference in run-to-run variability. Fixed patching has a large AUPRC standard deviation of 0.0676, while Tok1 and Tok3 have smaller standard deviations of 0.0076 and 0.0202, respectively. Since validation and test windows are frozen and identical across tokenizers, this difference is not due to changing evaluation examples. A plausible explanation is that beat-synchronous tokenization imposes a stronger rhythm-level inductive bias by presenting the model with cardiac-cycle units rather than arbitrary fixed patches. This may be helpful when the positive class is rare and precision-recall performance is sensitive to a small number of difficult examples.

Overall interpretation: Across both settings, beat-synchronous tokenization provides a favorable performance–efficiency tradeoff, but it is not uniformly superior on every metric. On PTB-XL, Tok1 and Tok3 match the strongest fixed-patch baseline with far fewer tokens. On Icentia11k, beat-synchronous tokenizers match fixed-patch AUPRC and are more stable across runs, but fixed patching achieves higher AUROC. These findings support a balanced conclusion: cardiac-cycle alignment is a useful inductive bias for ECG Transformers, especially for compact sequence modeling, but the design of the beat encoder is critical. Morphology-preserving beat representations work well, whereas overly compressed beat tokens lose important diagnostic details.

6. CONCLUSION

We studied two tokenization strategies for Transformer-based ECG representation learning, comparing fixed temporal patches with beat-synchronous tokenizers under matched pretraining and downstream evaluation protocols. Across 12-lead PTB-XL diagnostic classification and 60-second Icentia11k rhythm classification, beat-synchronous tokenization achieved competitive performance with shorter, physiologically meaningful token sequences. The strongest beat-based variants, Tok1 and Tok3, matched fine fixed-patch performance on PTB-XL while using substantially fewer tokens, and achieved comparable AUPRC with lower run-to-run variability on imbalanced Icentia11k rhythm classification. However, the poor performance of the adaptive-pooling variant of beat-synchronous tokenization shows that beat alignment alone is not sufficient: beat-token design must preserve morphology and, when relevant, timing information. These results suggest that beat-synchronous tokenization is a promising token-efficient alternative to fixed temporal patching, but future work should explore stronger beat encoders, larger-scale beat-level pretraining, and broader evaluation across rhythm and morphology-focused ECG tasks.

7. REFERENCES

[1] Zachi I Attia, Suraj Kapa, Francisco Lopez-Jimenez, Paul M McKie, Dorothy J Ladewig, Gaurav Satam, Patricia A Pellikka, Maurice Enriquez-Sarano, Peter A Noseworthy, Thomas M Munger, et al., "Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram," *Nature medicine*, vol. 25, no. 1, pp. 70–74, 2019.

[2] Zachi I Attia, Peter A Noseworthy, Francisco Lopez-Jimenez, Samuel J Asirvatham, Abhishek J Deshmukh, Bernard J Gersh, Rickey E Carter, Xiaoxi Yao, Alejandro A Rabinstein, Brad J Erickson, et al., "An artificial intelligence-enabled ecg algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction," *The Lancet*, vol. 394, no. 10201, pp. 861–867, 2019.

[3] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature medicine*, vol. 25, no. 1, pp. 65–69, 2019.

[4] Nathaniel Diamant, Erik Reinertsen, Steven Song, Aaron D Aguirre, Collin M Stultz, and Puneet Batra, "Patient contrastive learning: A performant, expressive, and practical approach to electrocardiogram modeling," *PLoS computational biology*, vol. 18, no. 2, pp. e1009862, 2022.

[5] Temesgen Mehari and Nils Strodthoff, "Self-supervised representation learning from 12-lead ecg data," *Computers in biology and medicine*, vol. 141, pp. 105114, 2022.

[6] Kaden McKeen, Sameer Masood, Augustin Toma, Barry Rubin, and Bo Wang, "Ecg-fm: An open electrocardiogram foundation model," 2025.

[7] Jun Li, Aaron Aguirre, Junior Moura, Che Liu, Lanhai Zhong, Chenxi Sun, Gari Clifford, M. Brandon Westover, and Shenda Hong, "An electrocardiogram foundation model built on over 10 million recordings with external evaluation across multiple domains," 2024.

[8] Yeongyeon Na, Minje Park, Yunwon Tae, and Sunghoon Joo, "Guiding masked representation learning to capture spatio-temporal relationship of electrocardiogram," *arXiv preprint arXiv:2402.09450*, 2024.

[9] Dani Kiyasseh, Tingting Zhu, and David A Clifton, "Clocs: Contrastive learning of cardiac signals across space, time, and patients," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5606–5615.

[10] Jiarui Jin, Haoyu Wang, Hongyan Li, Jun Li, Jiahui Pan, and Shenda Hong, "Reading your heart: Learning ecg words and sentences via pre-training ecg language model," *arXiv preprint arXiv:2502.10707*, 2025.

[11] Xin Wang, Burcu Ozek, Aruna Mohan, Amirhossein Ravari, Or Zilbershot, and Fatemeh Afghah, "Rhythmbert: A self-supervised language model based on latent representations of ecg waveforms for heart disease detection," in *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026, pp. 6351–6355.

[12] Sangkyu Kim, Jiwoo Lim, and Jaeseong Jang, "Patient-adaptive beat-wise temporal transformer for atrial fibrillation classification in continuous long-term cardiac monitoring," *IEEE Access*, vol. 12, pp. 172358–172367, 2024.

[13] Runze Ma and Caizhi Liao, "Beat-net: Injecting biomimetic spatio-temporal priors for interpretable ecg classification," *arXiv preprint arXiv:2601.07316*, 2026.

[14] Fuying Wang, Jiacheng Xu, and Lequan Yu, "From token to rhythm: A multi-scale approach for ECG-language pretraining," in *Proceedings of the 42nd International Conference on Machine Learning*, Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, Eds. 13–19 Jul 2025, vol. 267 of *Proceedings of Machine Learning Research*, pp. 65059–65074, PMLR.

[15] Brian Gow, Tom Pollard, Larry A Nathanson, Alistair Johnson, Benjamin Moody, Chrystinne Fernandes, Nathaniel Greenbaum, Jonathan W Waks, Parastou Eslami, Tanner Carbonati, Ashish Chaudhari, Elizabeth Herbst, Dana Moukheiber, Seth Berkowitz, Roger Mark, and Steven Horng, "MIMIC-IV-ECG: Diagnostic Electrocardiogram Matched Subset," *PhysioNet*, Sept. 2023, Version 1.0.

[16] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Wojciech Samek, and Tobias Schaeffter, "PTB-XL, a large publicly available electrocardiography dataset," *PhysioNet*, Nov. 2022, Version 1.0.3.

[17] Shawn Tan, Guillaume Androz, Ahmad Chamseddine, Pierre Fecteau, Aaron Courville, Yoshua Bengio, and Joseph Paul Cohen, "Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery," *arXiv preprint arXiv:1910.09570*, 2019.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] Takaya Saito and Marc Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, pp. e0118432, 2015.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.