PULSE: PHYSIOLOGY-ANCHORED SET ENCODING FOR VARIABLE-LEAD ECG REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most ECG representation models assume fixed lead configurations, limiting robustness to variations in lead availability, ordering, and signal degradation. In real-world deployment, ECG lead configurations vary widely across acquisition systems, from standard 12-lead recordings to single-lead wearables. We introduce PULSE, a physiology-anchored set encoding framework for variable-lead ECG representation learning, designed to learn from arbitrary subsets and orderings of the 12 standard ECG leads. The framework encodes each lead independently without explicit lead-identity embeddings and then combines the leads through two parallel branches: one models interactions between leads with a single pairwise cross-lead attention block and pools them by permutation-invariant set aggregation, while the other pools per-lead representations that stay unchanged by which other leads are present. It is pretrained with a self-supervised objective: given randomly selected lead subsets with temporal masking, it predicts fixed morphology and rhythm descriptors computed from the corresponding clean signals. A single fine-tuned model then supports varying lead configurations without configuration-specific retraining. Comprehensive experiments across multiple ECG datasets and lead configurations demonstrate that PULSE achieves competitive or superior performance to existing self-supervised ECG representation-learning methods, while remaining robust across variable-lead settings. With 0.43M parameters, PULSE runs on a Raspberry Pi 4B, processing a 10-second ECG in about 90 ms, more than $100\times$ faster than real time, with the same accuracy as on a workstation. These results show that variable-lead representation learning, compact modeling, and edge deployment can be achieved within a single ECG encoder.

1 INTRODUCTION

Electrocardiogram (ECG) is the most widely used non-invasive measurement of cardiac electrical activity which supports the diagnosis of conditions from arrhythmias to myocardial infarction (Ameta et al., 2026). Deep learning performs well on this task but depends on large collections of expert-labeled recordings, which are costly to obtain. Self-supervised pretraining reduces this dependence by learning general representations from large collections of unlabeled recordings, which can then be adapted to downstream diagnostic tasks with substantially fewer labeled examples. This paradigm has been successful across domains, including masked representation learning (He et al., 2022).

Self-supervised ECG pretraining has advanced quickly in scale. ECG-FM pretrains on 1.5 million recordings with contrastive and generative objectives (McKeen et al., 2024). Joint-embedding predictive pretraining on more than one million recordings improves classification without hand-crafted augmentations (Weimann & Conrad, 2024). ECG encoders also use contrastive learning (Kiyasseh et al., 2021; Gopal et al., 2021) and masked reconstruction (Na et al., 2024b). A recent benchmark shows that compact models can match or outperform much larger ECG foundation models on clinical tasks (Al-Masud et al., 2026), highlighting the importance of the learned representation and pretraining objective beyond model scale.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A standard 12-lead ECG contains the limb leads I, II, III, aVR, aVL, and aVF and the precordial leads V1–V6. These leads capture different spatial projections of cardiac activity, so missing leads can remove information that the remaining leads cannot fully recover. Existing work addresses variable-lead settings through lead masking (Oh et al., 2022), attention-based aggregation (Ruiperez-Campillo et al., 2026), contrastive distillation (Nguyen et al., 2025), and multimodal supervision (Liu et al., 2025). ST-MEM, for example, adapts to reduced-lead settings by fine-tuning a separate model for each configuration (Na et al., 2024b).

A natural way to handle variable lead availability is to treat the available leads as an unordered set. Shared per-lead processing is permutation equivariant, while symmetric aggregation, such as mean pooling or attention with learned queries, produces a permutation-invariant representation (Zaheer et al., 2017; Lee et al., 2019). This formulation removes dependency on a fixed lead ordering and allows the encoder to operate on arbitrary subsets of available leads.

Existing methods assumes either a fixed lead set or need extra training per configuration. We therefore ask whether ECG representation learning can be formulated over the set of available leads and remain effective across lead configurations. To address this challenge, we propose PULSE: Physiology-Anchored Set Encoding for Variable-Lead ECG Representation Learning, a compact set-based framework that combines permutation-invariant lead aggregation with physiology-anchored self-supervised learning. Our contributions are as follows:

1. A compact 0.43M-parameter encoder for variable-lead ECGs that produces a fixed-dimensional, lead-order-invariant representation for any subset of the available leads. The encoder uses shared-weight per-lead temporal processing followed by a Cross-Lead Branch, which enables lead interaction through a single cross-lead attention block and symmetric aggregation, and an Anonymous Branch, which independently encodes each lead and produces a subset-stable representation unaffected by other leads are present.
2. A physiology-anchored self-supervised pretraining framework that jointly predicts shared morphology, lead-specific morphology, rhythm, and global descriptors from corrupted recordings. Pretraining uses random lead subsets, lead-specific and shared temporal masking, and fixed targets built from morphology and rhythm statistics of the clean signal. The targets are predicted with contrastive losses augmented by a small regression term.
3. A comprehensive evaluation on CPSC2018, PTB-XL, and PhysioNet2017 under variable-lead and lead-order conditions, including linear probing and fine-tuning across multiple lead configurations. We evaluate classification performance using accuracy, macro F1, and AUROC, and further analyze robustness to missing and reordered leads. We also evaluate the computational efficiency and edge deployment of PULSE on a Raspberry Pi 4B.

The remainder of the paper is organized as follows. Section 2 presents methods, Section 3 describes the experimental setup, and Section 4 presents the results. Section 5 concludes the paper, with additional details, analyses, and edge-deployment results provided in the appendices.

2 METHOD

We consider an ECG dataset $\mathbb{X} = \{x^{(i)}\}_{i=1}^{N_{\text{rec}}}$ of N_{rec} recordings. Each recording is a set of leads, $x^{(i)} = \{x_c^{(i)}\}_{c \in \mathcal{C}^{(i)}}$, where $\mathcal{C}^{(i)} \subseteq \{1, \ldots, 12\}$ is the set of available leads and each lead $x_c^{(i)} \in \mathbb{R}^T$ is a waveform of T samples. All recordings are 10 s long and sampled at 250 Hz, so $T = 2500$, and a recording has $C = |\mathcal{C}^{(i)}| \in \{1, \ldots, 12\}$ leads.

Fig. 1 shows the PULSE architecture, which has three stages. In the *input and augmentation* stage, a random subset of the leads is kept, in random order, and receives amplitude scaling and noise. In the *lead-order-invariant set encoder* stage, an Intra-Lead Encoder first converts each lead into a sequence of tokens, replaces the temporally masked tokens with a learned mask token, and adds temporal context with a local-global temporal block, using the same weights for every lead. Its output feeds two parallel branches: a Cross-Lead Branch, in which the leads exchange information, giving Z, and an Anonymous Branch, in which each lead is summarized without knowing which other leads are present, giving E. In the *physiology descriptor* stage, four fixed descriptors computed from the clean signal are predicted from the encoder's representations. The encoder f takes a single

recording $x \in \mathbb{R}^{C\times T}$ with any number of leads C and returns the dual-invariant representation z, the concatenation of Z and E:

$$z = f(x) = Z \oplus E \in \mathbb{R}^{256}, \tag{1}$$

where $\oplus$ denotes concatenation. Throughout, lead-order invariance means that z is unchanged under any reordering of the available leads, and the encoder is defined for any $C \geq 1$ without lead-index embeddings. During pretraining, prediction heads read from z and from three intermediate representations: the pooled lead sequence s of the Cross-Lead Branch, its rhythm tokens ρ, and the per-lead tokens e_c of the Anonymous Branch. These are not encoder outputs, and the heads are discarded after pretraining. For downstream tasks, the classifier uses z together with a pooled feature z^{pool} from the Cross-Lead Branch and lead statistics v^{ft} from the Anonymous Branch. Sections 2.1, 2.2, and 2.3 describe the three stages, and Section 2.4 describes the downstream evaluation.

2.1 Input and Augmentation

Each batch uses a random lead count C, and each recording keeps a random subset of C of its leads, in random order. The kept waveforms receive random amplitude scaling and Gaussian noise as shown in Fig. 1 stage-1. Lead sampling is detailed in Appendix A.3.

2.2 Lead-Order-Invariant Set Encoder

The encoder, shown in Fig. 1 stage-2, first processes each lead independently, performs a single cross-lead interaction, and then combines the leads using symmetric operations. The *Intra-Lead Encoder*, Fig. 1 (a), extracts temporal features from each lead independently. The *Cross-Lead Branch*, Fig. 1 (b), uses Cross-Lead Attention for information exchange, Set Aggregation to pool the leads, and Global-Rhythm Fusion to integrate full- and coarse-resolution temporal context into Z. The *Anonymous Branch*, Fig. 1 (c), applies Anonymous Projection to each lead and uses Anonymous Readout to pool lead summaries into E. The Intra-Lead Encoder, Cross-Lead Attention, and Anonymous Projection are permutation-equivariant, while Set Aggregation and Anonymous Readout provide lead-order invariance through symmetric pooling.

Intra-Lead Encoder. Each available lead is independently processed using the same Intra-Lead Encoder parameters. A convolutional tokenizer (f_{tok}) maps the waveform x_c to $N = 100$ tokens, each with width $D = 64$. The tokenizer uses three parallel 1D convolutions with kernel sizes 5, 11 and 21, followed by a residual depthwise convolutional block and gated strided downsampling. Each output token represents 25 input samples, corresponding to 0.1 s. After tokenization, a fixed sinusoidal temporal encoding $p \in \mathbb{R}^{N\times D}$ is added to the tokens of every lead. Since the same encoding is used for all leads, it provides temporal position information without encoding lead identity. The resulting tokens are then processed by a local-global temporal block (f_{lg}), which captures local temporal patterns using depthwise convolutions and global temporal patterns using multi-head self-attention across the tokens of each lead:

$$h_c = f_{\text{lg}}(f_{\text{tok}}(x_c) + p) \in \mathbb{R}^{N\times D}, \tag{2}$$

where $h_{c,n} \in \mathbb{R}^D$ is the representation of temporal position $n = 1, \ldots, N$ for lead c. Thus, h_c depends only on x_c and does not use information from other leads.

The token sequences (h_c) of all C leads feed two parallel branches. In the **Cross-Lead Branch**, Cross-Lead Attention lets the leads exchange information, and Set Aggregation pools them into one sequence. A Global Temporal Encoder and a Rhythm Encoder then process this sequence in parallel, and a Global-Rhythm Fusion combines them into Z. In the **Anonymous Branch**, Anonymous Projection transforms each lead separately, and Anonymous Readout pools the leads with symmetric operations into E.

Cross-Lead Attention. Cross-Lead Attention,Fig. 1 (b), is the only component in PULSE that explicitly models pairwise interactions between leads; the other components either process each lead separately or pool the leads without comparing them with each other. At each temporal position n, the tokens $h_{c,n}$ of the C available leads are stacked into $H_n \in \mathbb{R}^{C\times D}$, where row c is $h_{c,n}$, and a

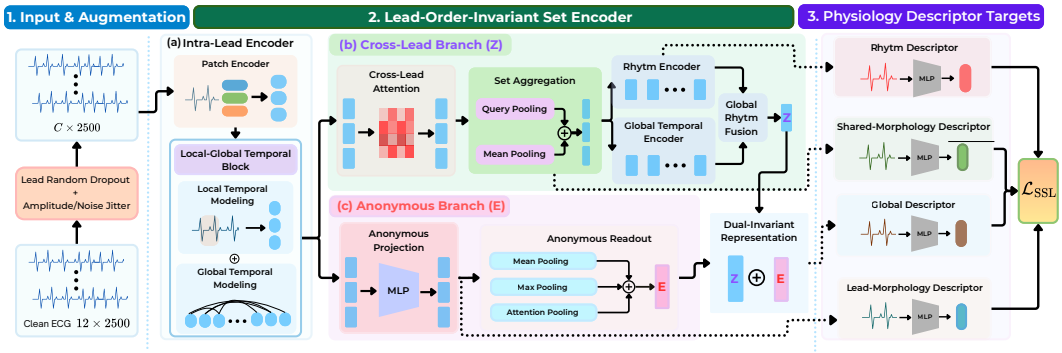


Figure 1: **Pretraining architecture.** The pipeline comprises three stages: (1) input and augmentation, (2) lead-order-invariant set encoding, and (3) physiology descriptor target prediction. In stage (1), the clean ECG undergoes random lead dropout and amplitude/noise jitter. In stage (2), each retained lead is patch-encoded by (a) the Intra-Lead Encoder, and then processed in parallel by (b) the Cross-Lead Branch (Z) and (c) the Anonymous Branch (E). The two branches form the dual-invariant representation $Z \oplus E$, which is trained in stage (3) against four fixed physiology descriptors computed from the clean ECG.

single transformer block performs self-attention along the lead axis:

$$a_{c,n} = \text{softmax}\left(\frac{h_{c,n}W_q\,(H_nW_k)^\top}{\sqrt{d}}\right)H_nW_v, \tag{3a}$$

$$\tilde{u}_{c,n} = \text{RMSNorm}(h_{c,n} + a_{c,n})\,, \tag{3b}$$

$$u_{c,n} = \tilde{u}_{c,n} + \text{FFN}(\text{RMSNorm}(\tilde{u}_{c,n}))\,, \qquad u_{c,n} \in \mathbb{R}^D. \tag{3c}$$

Here, W_q, W_k, and W_v are the query, key, and value projections, shared by all leads, so no lead has its own weights, and the softmax is computed over the C available leads. Equation 3a shows one attention head for clarity; the model uses four heads, each with dimension $d = D/4 = 16$, whose outputs are concatenated and linearly projected. Stacking the outputs $u_{c,n}$ of all leads gives $U_n \in \mathbb{R}^{C\times D}$, and we write $U_n = F(H_n)$ for the whole block. Because attention uses no lead-index embedding and the remaining operations act on each lead separately with shared parameters, F is permutation equivariant: for any permutation matrix P,

$$F(PH_n) = P\,F(H_n). \tag{4}$$

Set Aggregation. Set aggregation denoted by A,,Fig. 1 (b), pools the tokens of the C available leads in U_n at each position n into a single vector $s_n = A(U_n) \in \mathbb{R}^{D_g}$, where $D_g = 96$. Thus, the output size is independent of the number of available leads. It uses two pooling operations: *Query Pooling*, which assigns content-dependent weights to the leads, and *Mean Pooling*, which gives equal weight to all leads. In Query Pooling, two learned query vectors $q_1, q_2 \in \mathbb{R}^D$ weight the lead tokens by how well each token matches the query, using their own learned projections W'_q, W'_k, W'_v, and take the weighted average. Their outputs are concatenated and projected:

$$b_{k,n} = \text{softmax}\left(\frac{q_kW'_q\,(U_nW'_k)^\top}{\sqrt{d}}\right)U_nW'_v, \qquad k \in \{1,2\}, \tag{5a}$$

$$s_n^{\text{query}} = \text{Linear}\big(b_{1,n} \oplus b_{2,n}\big) \in \mathbb{R}^{D_g}. \tag{5b}$$

The queries are matched only with the lead tokens, not the leads with each other. Therefore, the cost of Query Pooling grows linearly with C. Mean Pooling averages the lead tokens and applies a linear

projection; its output is then added to the Query Pooling output and normalized:

$$\mu_n = \frac{1}{C}\sum_{c=1}^{C} u_{c,n}, \qquad s_n^{\text{mean}} = \text{Linear}\big(\mu_n\big), \tag{6a}$$

$$s_n = \text{RMSNorm}\big(s_n^{\text{query}} + s_n^{\text{mean}}\big) \in \mathbb{R}^{D_g}. \tag{6b}$$

Applying A at every temporal position n gives the sequence $s = (s_1, \ldots, s_N) \in \mathbb{R}^{N\times D_g}$, which is passed to the rest of the Cross-Lead Branch. Reordering the leads does not change any s_n, because both pooling operations are symmetric over the leads: $A(PU_n) = A(U_n)$. Together with the permutation equivariance of Cross-Lead Attention, this gives, at every temporal position n,

$$A\big(F(PH_n)\big) = A\big(P\,F(H_n)\big) = A\big(F(H_n)\big) = s_n. \tag{7}$$

Hence s does not depend on the order of the leads. The rest of the Cross-Lead Branch uses only s and works along time, so Z does not depend on lead order either.

Global-Rhythm Fusion. The aggregated sequence s is fed into two parallel branches, (Fig. 1b),: a Global Temporal Encoder and a Rhythm Encoder. Their outputs are then mixed with s through learned gates to give a fused sequence, which is reduced to the branch representation Z.

The Global Temporal Encoder f_{glob} applies multi-head self-attention across all N temporal positions. Its output $s^{\text{glob}} = f_{\text{glob}}(s) \in \mathbb{R}^{N\times D_g}$ captures long-range temporal dependencies.

The Rhythm Encoder f_{rhy} first shortens the sequence by merging every five neighboring positions into one, using a learned convolution blended with average pooling through a learned gate. It then applies multi-head self-attention, with its own parameters, to this shorter sequence. Its output is a sequence of rhythm tokens $\rho = (\rho_1, \ldots, \rho_{N_r}) \in \mathbb{R}^{N_r\times D_g}$, with $N_r = N/5 = 20$. Each token ρ_j covers five positions (0.5 s) and captures the signal at a coarser time scale. To mix these tokens with s, we repeat each one over the five positions it covers, which gives $s^{\text{rhy}} \in \mathbb{R}^{N\times D_g}$ with the same length as s.

We now have three sequences of the same length: s, s^{glob}, and s^{rhy}. They are mixed with two learned scalar gates, γ_{glob} and γ_{rhy}, which enter through the sigmoid σ. The gates are initialized so that $\sigma(\gamma_{\text{glob}}) = \sigma(\gamma_{\text{rhy}}) = 0.1$, so the fused sequence starts mostly from s and takes in the two branches gradually. To obtain Z, we concatenate the mean and the maximum of the fused sequence over its N temporal positions and pass the result through a linear layer:

$$s_n^{\text{fus}} = \text{RMSNorm}\big(s_n + \sigma(\gamma_{\text{glob}})(s_n^{\text{glob}} - s_n) + \sigma(\gamma_{\text{rhy}})(s_n^{\text{rhy}} - s_n)\big), \tag{8a}$$

$$z^{\text{pool}} = \Big(\frac{1}{N}\sum_{n=1}^{N} s_n^{\text{fus}}\Big) \oplus \Big(\max_n s_n^{\text{fus}}\Big) \in \mathbb{R}^{2D_g}, \qquad Z = \text{Linear}\big(z^{\text{pool}}\big). \tag{8b}$$

Anonymous Projection. It gives each lead a representation computed from that lead alone (Fig. 1c). The same function f_{anon}, a two-layer perceptron with RMS normalization at its input and output, is applied to every token of every lead:

$$e_c = f_{\text{anon}}(h_c) \in \mathbb{R}^{N\times D_a}, \qquad D_a = 64, \tag{9}$$

where $e_{c,n} \in \mathbb{R}^{D_a}$ is the token of lead c at temporal position n. Because h_c depends only on x_c, so does e_c: it does not change when other leads are added, removed, or reordered. We call this property *subset stability*, and we confirm numerically that e_c is the same across lead subsets up to floating-point precision. It also gives a per-lead summary $e_c^{\text{pool}} \in \mathbb{R}^{D_a}$, a linear map of the temporal mean and elementwise maximum of e_c, which is used only by the pretraining regularizer.

Anonymous Readout. This step takes the per-lead tokens e_c and reduces them to the branch output E, Fig. 1 (c). First, each lead is reduced to a single vector v_c by taking the mean and the maximum of its tokens over time and passing them through a small network. The C lead vectors are then combined using three symmetric operations: their mean, their elementwise maximum, and an

attention-weighted average:

$$v_c = \mathrm{RMSNorm}\Big(\mathrm{GELU}\Big(\mathrm{Linear}\Big(\Big(\frac{1}{N}\sum_{n=1}^{N} e_{c,n}\Big) \oplus \Big(\max_n e_{c,n}\Big)\Big)\Big)\Big) \in \mathbb{R}^{128}, \tag{10a}$$

$$\bar{v} = \frac{1}{C}\sum_{c=1}^{C} v_c, \qquad v^{\max} = \max_c v_c, \qquad v^{\mathrm{attn}} = \sum_{c=1}^{C} \alpha_c v_c, \tag{10b}$$

$$E = \mathrm{RMSNorm}\big(\bar{v} + \sigma(\gamma_{\max})\,(v^{\max} - \bar{v}) + \sigma(\gamma_{\mathrm{attn}})\,(v^{\mathrm{attn}} - \bar{v})\big) \in \mathbb{R}^{128}, \tag{10c}$$

where the maxima are elementwise, and the attention weights $\alpha_c = \mathrm{softmax}_c\big(\mathrm{RMSNorm}(v_c)\, w^\top\big)$ use a learned vector $w \in \mathbb{R}^{128}$. As in Global-Rhythm Fusion, the gates are initialized so that $\sigma(\gamma_{\max}) = \sigma(\gamma_{\mathrm{attn}}) = 0.1$, so E starts mostly from the mean $\bar{v}$. All three operations are symmetric in the leads, so E is invariant to lead order.

Dual-Invariant Representation. The final encoder representation is obtained by concatenating the outputs of the two branches:

$$z = Z \oplus E \in \mathbb{R}^{256}, \tag{11}$$

and is invariant to lead order because both Z and E are invariant. The encoder has 428,692 parameters.

2.3 Physiology-Anchored Self-Supervised Pretraining

PULSE is pretrained on unlabeled 12-lead recordings, none of these recordings are used in the downstream datasets. During pretraining, the encoder receives a corrupted version of each recording and predicts descriptors computed from the corresponding clean recording. Because these targets are produced by a parameter-free function, they remain fixed during training shown in the Fig. 1 stage-3.

Channel-Temporal Corruption. A temporal mask hides 50–70% of the positions of each lead. It has a *shared* part $\mathcal{M}^{\mathrm{sh}}$, common to all kept leads, and a *lead-specific* part $\mathcal{M}_c^{\mathrm{lead}}$ for each lead. The rhythm mask $\mathcal{M}^{\mathrm{rhy}}$ contains token j whenever any of the five positions of ρ_j lies in $\mathcal{M}^{\mathrm{sh}}$. Masked positions of $f_{\mathrm{tok}}(x_c)$ are replaced by a learned mask token before p is added in equation 2; the mask is slightly dilated for this replacement, and the losses use the undilated masks. Lead sampling and masking are detailed in Appendix A.3.

Physiology Prediction Targets. The targets are fixed descriptors computed from the clean recording before corruption. They capture two aspects of the ECG signal: *morphology*, which describes the local waveform shape, and *rhythm*, which describes slower temporal patterns. A parameter-free function first standardizes each lead using its own mean and standard deviation and applies $\tanh$ to each statistic. It then computes 16 *morphology statistics* for every 25-sample patch (one token, 0.1 s) and 12 *rhythm statistics* for every 125-sample window (one rhythm token, 0.5 s), including amplitude, slope, smoothness, and autocorrelation features. These statistics form four targets: the *shared morphology* m_n^{sh} at position n, the average of the morphology statistics over the kept leads; the *lead morphology* $m_{c,n}^{\mathrm{lead}}$, the morphology statistics of each kept lead c at position n; the *rhythm descriptor* r_j of rhythm token j, obtained by averaging the rhythm statistics across all 12 leads and therefore unchanged by which leads are kept; and the *global descriptor* g, which stacks the mean, standard deviation, maximum, and minimum of m_n^{sh} over n with the mean and standard deviation of r_j over j. These descriptors are not clinical measurements. They are used only during pretraining, and the target extractor is discarded afterwards.

Prediction Heads. A separate linear layer, called a prediction head, predicts each target from the encoder representation that carries the matching information: the aggregated sequence s, the per-lead tokens e_c, the rhythm tokens ρ, and the encoder output z. A hat marks the prediction of the corresponding target, indexed by position n, lead c, or rhythm token j:

$$\hat{m}_n^{\mathrm{sh}} = \mathrm{Linear}(s_n), \qquad \hat{m}_{c,n}^{\mathrm{lead}} = \mathrm{Linear}(e_{c,n}), \qquad \hat{r}_j = \mathrm{Linear}(\rho_j), \qquad \hat{g} = \mathrm{Linear}(z). \tag{12}$$

The heads are discarded after pretraining.

Physiology-Anchored Objective. Each of the four targets defines a separate prediction task. For m^{sh}, m^{lead}, and r, we compute the loss independently at each masked position. At position n, we collect the samples masked at that position–recordings for m^{sh} and r, and kept recording–lead pairs for m^{lead} and require each prediction to match the clean target from the same sample while distinguishing it from the targets of other samples at the same position. We use an InfoNCE loss with cosine similarity and temperature $\tau = 0.2$, averaged over masked positions containing at least two samples, together with a small Smooth-L1 regression term:

$$\mathcal{L}_{\mathrm{pos}} = \mathcal{L}_{\mathrm{NCE}}(\hat{y}, y) + \beta\,\mathrm{SmoothL1}(\hat{y}, y), \qquad \beta = 0.1. \tag{13}$$

Applying this objective with the masks $\mathcal{M}^{\mathrm{sh}}$, $\mathcal{M}_c^{\mathrm{lead}}$, and $\mathcal{M}^{\mathrm{rhy}}$ yields the corresponding losses $\mathcal{L}_{\mathrm{sh}}$, $\mathcal{L}_{\mathrm{lead}}$, and $\mathcal{L}_{\mathrm{rhy}}$.For the global descriptor g, we use a symmetric InfoNCE loss across the samples in the batch, together with the same regression term, denoted by $\mathcal{L}_{\mathrm{gl}}$. The complete self-supervised objective is

$$\mathcal{L}_{\mathrm{SSL}} = \mathcal{L}_{\mathrm{sh}} + w_{\mathrm{lead}}(t)\mathcal{L}_{\mathrm{lead}} + \mathcal{L}_{\mathrm{rhy}} + \mathcal{L}_{\mathrm{gl}}, \tag{14}$$

where $w_{\mathrm{lead}}(t)$ increases linearly with training step t.

2.4 Variable-Lead Downstream

The downstream classifier uses three representations that are already produced by the encoder: the encoder output z from equation 11, the pooled fused sequence z^{pool} from equation 8b, and the lead-vector statistics $v^{\mathrm{ft}} = \bar{v} \oplus v^{\max} \in \mathbb{R}^{256}$, computed from the mean and elementwise maximum of the lead vectors v_c in equation 10b. These representations are concatenated as

$$\phi(x) = z \oplus z^{\mathrm{pool}} \oplus v^{\mathrm{ft}} \in \mathbb{R}^{704}, \tag{15}$$

which is lead-order invariant because each component is lead-order invariant.

For *fine-tuning*, the encoder is trained jointly with a LayerNorm and a linear classifier applied to $\phi(x)$. We use AdamW, a cosine learning-rate schedule, label smoothing, mixup, and amplitude, noise, and time-shift augmentation.

For *linear probing*, the encoder is frozen and only a linear classifier is trained using standardized $\phi(x)$. All experiments use stratified five-fold cross-validation. In each fold, 70% of the recordings are used for training, 10% for validation, and 20% for testing. We report the mean and standard deviation across the five folds.

3 Experimental Setup

This section describes the datasets and baseline methods used to evaluate PULSE. Additional dataset and implementation details are provided in Appendix A, including preprocessing, data splits, training configurations, and hardware specifications.

3.1 Datasets

We pretrain on two 12-lead sources and evaluate on three labeled downstream datasets, none of which is used in pretraining. All recordings are 10 s long and sampled at 250 Hz.

Pre-training datasets. We use 132,930 recordings from CODE-15% (Ribeiro et al., 2021) and 42,790 from the arrhythmia database of Zheng et al. (2022), which combines the Chapman-Shaoxing and Ningbo datasets. In both cases we keep only the 10 s recordings. The resulting 175,720 recordings are used without labels.

Downstream datasets. CPSC2018 (Liu et al., 2018) has 8,686 recordings in 9 arrhythmia classes. PTB-XL (Wagner et al., 2020) has 16,244 recordings in 5 diagnostic superclasses. Both are 12-lead. PhysioNet2017 (Clifford et al., 2017) has 8,528 single-lead (lead I) recordings in 4 classes.

Table 1: Comparison of self-supervised learning methods on PTB-XL and CPSC2018. Results are reported for linear evaluation and fine-tuning. Mean ± standard deviation are reported over five runs, with the best result in each metric shown in bold.

Method	PTB-XL			CPSC2018		
	Accuracy	**F1**	**AUROC**	**Accuracy**	**F1**	**AUROC**
Linear Evaluation						
3KG	0.6236 ± 0.0103	0.4067 ± 0.0074	0.7490 ± 0.0031	0.4303 ± 0.0195	0.3908 ± 0.0186	0.7929 ± 0.0089
SimCLR	0.6123 ± 0.0109	0.4196 ± 0.0065	0.7579 ± 0.0064	0.4429 ± 0.0087	0.4122 ± 0.0058	0.8057 ± 0.0081
CMSC	0.7121 ± 0.0072	0.5800 ± 0.0107	0.8726 ± 0.0066	0.6504 ± 0.0134	0.6116 ± 0.0137	0.9131 ± 0.0033
MoCo v3	0.6921 ± 0.0092	0.5458 ± 0.0057	0.8573 ± 0.0063	0.6622 ± 0.0179	0.6378 ± 0.0193	0.9172 ± 0.0062
PULSE (Ours)	**0.7643 ± 0.0073**	**0.6319 ± 0.0088**	**0.9055 ± 0.0050**	**0.7252 ± 0.0122**	**0.6997 ± 0.0156**	**0.9459 ± 0.0039**
Fine-tuning						
3KG	0.8062 ± 0.0043	0.6706 ± 0.0127	0.9276 ± 0.0088	0.8538 ± 0.0116	0.8454 ± 0.0159	0.9684 ± 0.0044
SimCLR	0.8103 ± 0.0039	0.6764 ± 0.0113	0.9355 ± 0.0073	0.8529 ± 0.0126	0.8459 ± 0.0161	0.9701 ± 0.0034
CMSC	0.8109 ± 0.0030	0.6663 ± 0.0060	0.9200 ± 0.0076	0.8425 ± 0.0056	0.8230 ± 0.0081	0.9655 ± 0.0033
MoCo v3	0.8123 ± 0.0049	0.6658 ± 0.0111	0.9311 ± 0.0016	0.8181 ± 0.0225	0.7977 ± 0.0241	0.9611 ± 0.0049
PULSE (Ours)	**0.8175 ± 0.0050**	**0.6786 ± 0.0157**	**0.9374 ± 0.0060**	**0.8775 ± 0.0087**	**0.8602 ± 0.0133**	**0.9758 ± 0.0056**

3.2 BASELINE

We compare PULSE with four representative self-supervised learning baselines: MoCo v3 (Chen et al., 2021), CMSC (Kiyasseh et al., 2021), SimCLR (Chen et al., 2020), and 3KG (Gopal et al., 2021). To ensure a controlled comparison, all baselines use the same backbone and lead encoder as PULSE, with the pretraining strategy being the only methodological difference. This design isolates the contribution of the pretraining approach from architectural variations. All methods are pretrained on the same dataset and evaluated using an identical 5-fold evaluation protocol on CPSC2018 and PTB-XL.

4 EXPERMENTS AND RESULTS

This section evaluates PULSE through quantitative and qualitative experiments to assess its effectiveness. With 0.43M parameters, PULSE is designed as a lightweight model for practical deployment.

4.1 EXPERIMENTAL RESULTS

PULSE demonstrates strong transfer performance on the unseen PTB-XL and CPSC2018 datasets under both linear evaluation and fine-tuning, as reported in Table 1. Under linear evaluation, PULSE achieves the highest accuracy, F1, and AUROC on both datasets under linear evaluation. Its F1 score is higher than the competing methods by 0.052–0.225 on PTB-XL and by 0.062–0.309 on CPSC2018. These results indicate effective transfer without updating the encoder. Fine-tuning improves performance across methods and narrows the performance gaps. PULSE retains the highest mean F1 on both datasets, achieving 0.8602 on CPSC2018 compared with 0.8459 for the next-best method. On PTB-XL, PULSE achieves an F1 of 0.6786 compared with 0.6764 for the next-best method, with the difference falling within the reported variability. A comparison with the larger ST-MEM model is provided in Appendix B.5.

4.2 LEAD-AGNOSTIC ECG REPRESENTATION LEARNING

To evaluate robustness to missing leads, each method is tested with all 12 leads, the six limb leads (I, II, III, aVR, aVL, and aVF), and a single lead (I); PhysioNet2017 provides an additional single-lead evaluation. Table 2 reports AUROC across these configurations. Under linear evaluation, PULSE achieves the highest AUROC across datasets and lead settings, with CPSC2018 decreasing from 0.946 with 12 leads to 0.901 with a single lead, while remaining above the 12-lead results of SimCLR and 3KG. After fine-tuning, PULSE remains highest across configurations except CPSC2018 with one lead, where 3KG is marginally higher. On single-lead PhysioNet2017, PULSE achieves an AUROC of 0.898 versus 0.808 for the next-best method. These results demonstrate robust transfer across varying lead configurations, including single-lead settings.

Table 2: Robustness to reduced-lead configurations on PTB-XL, CPSC2018, and PhysioNet2017. Results report AUROC for linear evaluation and fine-tuning; PTB-XL and CPSC2018 use 12, 6, and 1-lead inputs, while PhysioNet2017 is inherently single-lead.

Methods	*PTB-XL*			*CPSC2018*			*PhysioNet2017*
	12-lead	**6-lead**	**1-lead**	**12-lead**	**6-lead**	**1-lead**	**1-lead**
				Linear Evaluation			
3KG	0.7490 ± 0.0031	0.6922 ± 0.0140	0.6772 ± 0.0134	0.7929 ± 0.0089	0.7742 ± 0.0087	0.7524 ± 0.0074	0.6347 ± 0.0196
SimCLR	0.7579 ± 0.0064	0.7356 ± 0.0064	0.6496 ± 0.0081	0.8057 ± 0.0081	0.7832 ± 0.0096	0.7777 ± 0.0030	0.6248 ± 0.0209
CMSC	0.8726 ± 0.0066	0.8217 ± 0.0139	0.7349 ± 0.0131	0.9131 ± 0.0033	0.8959 ± 0.0045	0.8550 ± 0.0041	0.6646 ± 0.0249
MoCo v3	0.8573 ± 0.0063	0.8269 ± 0.0097	0.7410 ± 0.0091	0.9172 ± 0.0062	0.8970 ± 0.0081	0.8677 ± 0.0069	0.6990 ± 0.0126
PULSE (Ours)	**0.9055 ± 0.0050**	**0.8697 ± 0.0045**	**0.7869 ± 0.0032**	**0.9459 ± 0.0039**	**0.9291 ± 0.0046**	**0.9010 ± 0.0053**	**0.7186 ± 0.0109**
				Fine-tuning			
3KG	0.9276 ± 0.0088	0.9015 ± 0.0053	0.8161 ± 0.0036	0.9684 ± 0.0044	0.9543 ± 0.0062	**0.9546 ± 0.0020**	0.6624 ± 0.1664
SimCLR	0.9355 ± 0.0073	0.9008 ± 0.0097	0.8045 ± 0.0067	0.9701 ± 0.0034	0.9561 ± 0.0049	0.9528 ± 0.0048	0.7532 ± 0.1760
CMSC	0.9200 ± 0.0076	0.9018 ± 0.0051	0.8115 ± 0.0125	0.9655 ± 0.0033	0.9604 ± 0.0043	0.9513 ± 0.0069	0.8078 ± 0.1603
MoCo v3	0.9311 ± 0.0016	0.8978 ± 0.0028	0.8184 ± 0.0116	0.9611 ± 0.0049	0.9618 ± 0.0038	0.9398 ± 0.0064	0.7209 ± 0.2071
PULSE (Ours)	**0.9374 ± 0.0060**	**0.9019 ± 0.0027**	**0.8230 ± 0.0119**	**0.9758 ± 0.0056**	**0.9643 ± 0.0073**	0.9529 ± 0.0064	**0.8981 ± 0.0093**

Table 3: Ablation of the Cross-Lead Branch (Z), Anonymous Branch (E), and pretraining objectives under linear evaluation. "w/o Z" and "w/o E" remove the corresponding branches, while loss variants use only the indicated objectives: $\mathcal{L}_{\mathrm{sh}}$ (shared morphology), $\mathcal{L}_{\mathrm{lead}}$ (lead-specific morphology), and $\mathcal{L}_{\mathrm{rhy}}$ (rhythm). Results are mean ± standard deviation over five folds.

	PTB-XL			CPSC2018		
Variant	Acc.	Macro F1	AUROC	Acc.	Macro F1	AUROC
$\mathcal{L}_{\mathrm{sh}} + \mathcal{L}_{\mathrm{lead}}$	0.714±0.007	0.549±0.010	0.863±0.003	0.604±0.014	0.568±0.011	0.896±0.004
$\mathcal{L}_{\mathrm{rhy}}$	0.714±0.010	0.557±0.005	0.866±0.005	0.651±0.010	0.617±0.009	0.912±0.000
w/o Z	0.736±0.004	0.597±0.012	0.892±0.002	0.669±0.014	0.639±0.017	0.925±0.004
w/o E	0.747±0.005	0.600±0.006	0.888±0.005	0.687±0.008	0.653±0.008	0.929±0.003
PULSE (full)	**0.764±0.007**	**0.632±0.009**	**0.906±0.005**	**0.725±0.012**	**0.700±0.016**	**0.946±0.004**

4.3 Ablation Studies

Removing either the Cross-Lead Branch (Z) or Anonymous Branch (E) reduces performance across all metrics on CPSC2018 and PTB-XL Table 3, confirming their complementary contributions. Using morphology losses alone also causes a substantial drop, particularly on CPSC2018. The full PULSE model achieves the best results across both datasets.

5 Conclusion

We presented PULSE, a compact physiology-anchored framework that treats the leads of an ECG as an unordered set. A Cross-Lead Branch models inter-lead interactions and an Anonymous Branch preserves subset-stable lead information, giving a fixed-dimensional representation that is independent of lead order and count, while the self-supervised objective anchors learning to fixed morphology and rhythm descriptors of the clean signal. Across PTB-XL, CPSC2018, and PhysioNet2017, PULSE outperforms four self-supervised baselines in linear evaluation, stays competitive after fine-tuning, and remains robust with six leads or a single lead; ablations show that both branches and the anchored objectives contribute. With only 0.43M parameters and real-time inference on a Raspberry Pi 4B, PULSE offers a lightweight approach to robust ECG representation learning across heterogeneous acquisition settings.

6 Reproducibility Statement

We provide the implementation of PULSE along with the training and evaluation code required to reproduce our experiments. We release the pre-trained encoder, as well as the fine-tuned and linear-probed model parameters, enabling direct reproduction of the reported downstream results. The paper provides detailed descriptions of the datasets, preprocessing procedures, training configurations, hyperparameters, and evaluation protocols. All datasets used in our experiments are publicly

available through their respective sources. These resources are intended to support reproduction of the representation learning and downstream evaluation results reported in this work.

6.1 HYPERPARAMETERS

All training configurations, including the epochs, batch sizes, optimizers, learning rates, and hardware specifications for pre-training, fine-tuning, and linear evaluation are provided in Appendix A.2.

6.2 CODE AVAILABILITY

The code and pre-trained model parameters will be available at `https://anonymous.4open.science/r/PULSE-BE65`. Please refer to the `README.md` file and follow the provided instructions to reproduce the experiments.

REFERENCES

M. A. Al-Masud, Juan Miguel Lopez Alcaraz, and Nils Strodthoff. Benchmarking ECG foundation models: A reality check across clinical tasks. In *International Conference on Learning Representations*, 2026.

Durgesh Ameta, Sanjay Somasundaram, S. Srivarshini, and Amit Shukla. PERFECG-Net: A lightweight periodicity-focused fetal ECG r-peak detection network. *IEEE Transactions on Instrumentation and Measurement*, 75:1–11, 2026. doi: 10.1109/TIM.2026.3682832.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Gari D. Clifford, Chengyu Liu, Benjamin Moody, Li-wei H. Lehman, Ikaro Silva, Qiao Li, A. E. Johnson, and Roger G. Mark. AF classification from a short single lead ECG recording: the PhysioNet/Computing in Cardiology Challenge 2017. In *Computing in Cardiology*, volume 44, 2017.

Bryan Gopal, Ryan Han, Gautham Raghupathi, Andrew Ng, Geoff Tison, and Pranav Rajpurkar. 3KG: Contrastive learning of 12-lead electrocardiograms using physiologically-inspired augmentations. In *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pp. 156–167, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Dani Kiyasseh, Tingting Zhu, and David A. Clifton. CLOCS: Contrastive learning of cardiac signals across space, time, and patients. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019.

Che Liu, Cheng Ouyang, Zhongwei Wan, Haozhe Wang, Wenjia Bai, and Rossella Arcucci. Knowledge-enhanced multimodal ECG representation learning with arbitrary-lead inputs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, Jianqing Li, and Eddie Ng Yin Kwee. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7):1368–1373, 2018.

Kaden McKeen, Sameer Masood, Augustin Toma, Barry Rubin, and Bo Wang. ECG-FM: An open electrocardiogram foundation model. *arXiv preprint arXiv:2408.05178*, 2024.

Yeongyeon Na, Minje Park, Yunwon Tae, and Sunghoon Joo. Guiding masked representation learning to capture spatio-temporal relationship of electrocardiogram. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yeongyeon Na, Minje Park, Yunwon Tae, and Sunghoon Joo. Guiding masked representation learning to capture spatio-temporal relationship of electrocardiogram. In *International Conference on Learning Representations*, 2024b.

Huynh Dang Nguyen, Trong-Thang Pham, Ngan Le, and Van Nguyen. TolerantECG: A foundation model for imperfect electrocardiogram. *arXiv preprint arXiv:2507.09887*, 2025.

Jungwoo Oh, Hyunseung Chung, Joon-myoung Kwon, Dong-gyun Hong, and Edward Choi. Lead-agnostic self-supervised learning for local and global representations of electrocardiogram. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 338–353, 2022.

Antônio H. Ribeiro, Gabriela M. M. Paixão, Emilly M. Lima, Manoel Horta Ribeiro, Marcelo M. Pinto Filho, Paulo R. Gomes, Derick M. Oliveira, Wagner Meira Jr., Thomas B. Schön, and Antonio Luiz P. Ribeiro. CODE-15%: a large scale annotated dataset of 12-lead ECGs, 2021.

Samuel Ruiperez-Campillo, Moritz Vandenhirtz, Simon Böhi, Sonia Laguna, Irene Cannistraci, Andrea Agostini, Ece Ozkan, Thomas M. Sutter, and Julia E. Vogt. From leads to latents: Attention-driven masked autoencoder for ECG time series. In *ICLR 2026 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2026.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffter. PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, 7(1):154, 2020.

Kuba Weimann and Tim O. F. Conrad. Self-supervised pre-training with joint-embedding predictive architecture boosts ECG classification performance. *arXiv preprint arXiv:2410.13867*, 2024.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

Jianwei Zheng, Huimin Guo, and Huimin Chu. A large scale 12-lead electrocardiogram database for arrhythmia study (version 1.0.0), 2022.

Table 4: Classes and number of recordings per class in the downstream datasets.

Dataset	Class	Recordings
PTB-XL	Normal ECG (NORM)	9,069
	Myocardial infarction (MI)	2,532
	ST/T change (STTC)	2,400
	Conduction disturbance (CD)	1,708
	Hypertrophy (HYP)	535
CPSC2018	Right bundle branch block (RBBB)	1,914
	Atrial fibrillation (AF)	1,266
	Normal	1,201
	Premature ventricular contraction (PVC)	1,104
	ST-segment depression (STD)	971
	Premature atrial contraction (PAC)	892
	First-degree AV block (1AVB)	841
	ST-segment elevation (STE)	272
	Left bundle branch block (LBBB)	225
PhysioNet2017	Normal sinus rhythm	5,076
	Other rhythm	2,415
	Atrial fibrillation (AF)	758
	Noisy	279

A Data and Implementation Details

This section provides additional details on the downstream datasets, training configurations, hardware, and signal preprocessing used in PULSE.

A.1 Dataset Description

Table 4 lists the classes of three downstream datasets and the number of recordings in each class. Every recording has exactly one label. For pretraining, the unlabeled recordings are divided into 80% training and 20% validation subsets.

A.2 Training Configurations and Hardware

PULSE is pretrained using the configuration reported in Table 5. For downstream evaluation, the corresponding fine-tuning and linear-evaluation settings are applied across all methods and datasets using stratified 5-fold cross-validation. In each fold, the data are randomly split into 70% training, 10% validation, and 20% testing subsets. All experiments were conducted using NVIDIA RTX PRO 5000 Blackwell GPUs and AMD EPYC 9554 CPUs with PyTorch 2.12 and CUDA 13.0.

Table 5: Training hyperparameters for PULSE.

	Pretraining	Fine-tuning	Linear evaluation
Batch size	1024	64	64
Learning rate	1.33×10^{-4}	10^{-3}	10^{-3}
Weight decay	0.05	10^{-4}	10^{-4}
Epochs	400	200	200
Patience	–	20	20
Schedule	Cosine annealing	Cosine annealing	Constant
Optimizer	AdamW	AdamW	AdamW

A.3 Patching, Lead Sampling and Masking of the Raw Signal

Patching. Each recording is a 10 s signal sampled at 250 Hz, so each lead has $T = 2{,}500$ samples. Every lead is split independently into non-overlapping patches of 25 samples (100 ms, stride 25),

Table 6: Performance of PULSE across different batch sizes using 12-lead ECGs on PTB-XL and CPSC2018. Results are reported for linear evaluation and fine-tuning. Mean ± standard deviation are reported over five runs.

Batch Size	*PTB-XL*			*CPSC2018*		
	Accuracy	**F1**	**AUROC**	**Accuracy**	**F1**	**AUROC**
			Linear Evaluation			
64	0.764 ± 0.007	$\mathbf{0.632 \pm 0.009}$	0.906 ± 0.005	0.725 ± 0.013	0.700 ± 0.016	0.946 ± 0.004
128	0.761 ± 0.006	0.631 ± 0.006	$\mathbf{0.908 \pm 0.006}$	0.725 ± 0.015	0.696 ± 0.018	0.946 ± 0.004
256	0.764 ± 0.005	0.627 ± 0.003	0.906 ± 0.003	$\mathbf{0.728 \pm 0.009}$	$\mathbf{0.701 \pm 0.011}$	$\mathbf{0.946 \pm 0.005}$
512	$\mathbf{0.766 \pm 0.006}$	0.628 ± 0.007	0.905 ± 0.005	0.726 ± 0.017	0.696 ± 0.018	0.945 ± 0.004
			Fine-tuning			
64	$\mathbf{0.818 \pm 0.005}$	$\mathbf{0.679 \pm 0.016}$	$\mathbf{0.937 \pm 0.006}$	$\mathbf{0.878 \pm 0.009}$	$\mathbf{0.860 \pm 0.013}$	0.976 ± 0.006
128	0.819 ± 0.006	0.678 ± 0.014	0.937 ± 0.003	0.873 ± 0.012	0.856 ± 0.016	0.975 ± 0.004
256	0.816 ± 0.004	0.664 ± 0.023	0.925 ± 0.010	0.867 ± 0.023	0.851 ± 0.027	$\mathbf{0.977 \pm 0.004}$
512	0.810 ± 0.003	0.654 ± 0.020	0.929 ± 0.012	0.857 ± 0.015	0.840 ± 0.020	0.977 ± 0.003

giving $N = 100$ patches per lead. We index the patches by their position $n = 1, \ldots, N$. Positions are the units that are hidden from, or shown to, the encoder.

Lead sampling. In each batch, the encoder sees only a random subset of C of the available leads. The lead count C is drawn from six groups with probabilities as shown in Table 8 The exact value of C is then drawn uniformly within the chosen group, and the leads themselves are chosen at random without replacement. This exposes the encoder to every lead count from one to twelve during pretraining. Figure 3 shows one draw with $C = 6$ leads (III, aVR, aVL, aVF, V2, V4).

Masking. Each kept lead c receives its own mask ratio $\eta_c \in [0.5, 0.7]$, so 50–70% of its $N = 100$ positions are hidden. The mask combines two parts:

1. **Shared mask** $\mathcal{M}^{\mathrm{sh}}$. One contiguous block of positions is hidden in every kept lead. Its length is 30–50% of the per-lead masked count, and at least one position is masked.
2. **Lead-specific mask** $\mathcal{M}_c^{\mathrm{lead}}$. Each lead hides additional positions independently, as random contiguous runs placed outside $\mathcal{M}^{\mathrm{sh}}$. These masks therefore differ from lead to lead.

The tokenizer has an overlapping receptive field that ca leak information into neighbouring positions. To prevent this, the mask is dilated by three positions on each side when computing what the encoder can actually see, and masks are resampled until at least 10% of the positions remain visible after this dilation. The dilation is not drawn in Figure 3.

B Additional Results and Comparisons

This section presents additional experiments on batch-size sensitivity and lead-order invariance, complementing the main evaluation of PULSE.

B.1 PULSE Performance on Different Batch Sizes

Table 6 reports PULSE with batch sizes of 64, 128, 256, and 512, using 12-lead ECGs on PTB-XL and CPSC2018. In linear evaluation, the results are almost the same for all batch sizes: AUROC changes by at most 0.003 on PTB-XL and 0.001 on CPSC2018, and F1 by at most 0.006. Fine-tuning is slightly more sensitive. F1 decreases gradually as the batch size grows, from 0.679 to 0.654 on PTB-XL and from 0.860 to 0.840 on CPSC2018, while AUROC stays close 0.925-0.937 on PTB-XL and 0.975-0.977 on CPSC2018. These differences are similar in size to the fold-to-fold standard deviation. We use a batch size of 64 in our main experiments, and the results show that PULSE does not depend strongly on this choice.

B.2 Lead-Order Invariance

PULSE treats the leads of a recording as an unordered set, so its representation should not depend on the order in which the leads are supplied. To test this, we repeat the linear probe of the main paper on all 8,686 CPSC and all 16,244 PTB-XL records with the final pretrained encoder kept frozen. In the reordered setting, every record keeps all 12 leads, but its leads are shuffled by a random permutation that is fixed per record and applied identically to training, validation and test data. The original-order run uses the same checkpoint, folds and hyperparameters. The results are identical to four decimals Table 7.

Table 7: Linear probe under random lead order. Mean $\pm$ std over 5 folds on all records. All 12 leads are kept; only their order changes.

Dataset	Lead order	Acc.	Macro F1	AUROC
CPSC	Original	0.725 ± 0.012	0.700 ± 0.016	0.946 ± 0.004
	Random	0.725 ± 0.012	0.700 ± 0.016	0.946 ± 0.004
PTB-XL	Original	0.764 ± 0.007	0.632 ± 0.009	0.906 ± 0.005
	Random	0.764 ± 0.007	0.632 ± 0.009	0.906 ± 0.005

B.3 Robustness to Random Lead Availability

To assess robustness to unknown lead availability, the final pretrained encoder is evaluated with and without random lead dropout under both linear probing and full fine-tuning, as shown in Table 9.

Random lead dropout. For each record, a random number of available leads is sampled according to the distribution in Table 8. The lead count is sampled uniformly within each range, after which the corresponding leads are selected without replacement. A fixed seed is assigned to each record to ensure reproducibility, and the same dropout pattern is used consistently across the training, validation, and test sets.

Results. Table 9 summarizes the effect of random lead dropout across both datasets and evaluation protocols. Random lead dropout reduces performance, with the largest Macro F1 reductions reaching 0.079–0.091. Fine-tuning is more robust to lead dropout in AUROC, with decreases of 0.015 and 0.043 on CPSC2018 and PTB-XL, respectively, compared with 0.031 and 0.059 under linear probing. Despite the reduced lead availability, the model retains substantial predictive performance, demonstrating robustness to randomly missing leads.

B.4 Downstream Fine-Tuning Stability on PhysioNet2017

PhysioNet2017 is a single-lead and strongly class-imbalanced dataset, making it a useful setting for assessing downstream stability. Figure 2 shows the per-fold AUROC, highlighting the stability of each method across folds. All methods use the same fine-tuning protocol. Under linear evaluation, all methods remain stable, with PULSE achieving the highest AUROC. In contrast, full fine-tuning produces substantial fold-to-fold variation among the baselines, contributing to the large standard deviations reported in Table 2. PULSE achieves 0.898 ± 0.009 and maintains a narrow AUROC range across all five folds.

B.5 Comparison with ST-MEM

PULSE uses 0.43M parameters, roughly $200\times$ fewer than ST-MEM Na et al. (2024a). As shown in Table 11, PULSE achieves competitive performance across accuracy, Macro F1, and AUROC under both linear evaluation and fine-tuning, despite its substantially smaller size. These results demonstrate that PULSE provides competitive downstream performance with substantially fewer parameters.

Table 8: Lead availability distribution used for random lead dropout during pre-training and downstreaming. The number of retained leads is sampled from the specified groups, with counts sampled uniformly within each group.

Number of retained leads	Probability
1	0.10
2	0.10
3–4	0.20
5–8	0.25
9–11	0.15
12	0.20

Table 9: Robustness to random lead availability using the pretrained encoder. For each record, a random subset of the 12 leads is retained according to the lead-dropout distribution used during pretraining. Results are mean ± std over five folds. "All" denotes the complete 12-lead input, while "Random" uses the sampled subset.

		CPSC2018			PTB-XL		
Protocol	Input	Acc.	Macro F1	AUROC	Acc.	Macro F1	AUROC
Linear	All	0.725±0.012	0.700±0.016	0.946±0.004	0.764±0.007	0.632±0.009	0.906±0.005
	Random	0.648±0.012	0.621±0.011	0.915±0.002	0.703±0.006	0.541±0.008	0.846±0.006
Fine-tune	All	0.878±0.009	0.860±0.013	0.976±0.006	0.818±0.005	0.679±0.016	0.937±0.006
	Random	0.800±0.010	0.778±0.010	0.961±0.004	0.758±0.004	0.589±0.009	0.894±0.007

C Edge Deployment

This section evaluates the computational efficiency and practical edge deployment of PULSE on a resource-constrained Raspberry Pi 4B.

C.1 Raspberry Pi 4B Deployment

To evaluate the practical deployment efficiency of PULSE, we exported the best-performing fold models after fine-tuning to ONNX and evaluated them on a Raspberry Pi 4B. Inference was performed using ONNX Runtime with the `CPUExecutionProvider` on a 64-bit ARM system. For each of CPSC2018 and PTB-XL, we evaluated one 12-lead model and one variable-lead model, both selected from the best-performing fold of the corresponding cross-validation setup. The variable-lead models accept the ECG and a lead-validity mask as inputs. Each exported model is approximately 1.9 MB and contains 428,692 encoder parameters, in addition to a lightweight dataset-specific classification head. The PULSE encoder requires 0.417 GFLOPs for a 12-lead, 10-second ECG recording.

Evaluation Protocol. Inference latency was measured with a batch size of one after 10 warm-up inferences, followed by 100 timed inferences per recording. The reported latency corresponds to the processing time for a single 10-second ECG recording. Memory usage was measured as the increase in resident set size relative to the runtime baseline after model loading and one warm-up inference. To verify numerical consistency between the workstation and edge deployment, we additionally recomputed the evaluation metrics on the Raspberry Pi and compared them with the corresponding workstation values.

Results. Table 12 summarizes the inference latency and memory overhead. Across the four configurations, PULSE processes a 10-second ECG in 84.4–91.9 ms on average, corresponding to approximately 11 recordings per second. Thus, inference is substantially faster than the duration of the input signal, indicating that the encoder can support real-time processing on a resource-constrained CPU platform. The additional memory overhead ranges from 6.1 to 29.3 MB. The maximum observed latency ranges from 87.4 to 131.5 ms, reflecting the variability of CPU-based inference on the Raspberry Pi.

Table 10: Summary of downstream datasets and evaluation splits. All datasets use stratified five-fold cross-validation.

Dataset	Recordings	Leads	Classes	Train	Val.	Test
PTB-XL	16,244	12	5	70%	10%	20%
CPSC2018	8,686	12	9	70%	10%	20%
PhysioNet2017	8,528	1	4	70%	10%	20%

Table 11: Comparison of PULSE with ST-MEM on PTB-XL and CPSC2018. ST-MEM results are taken from its original paper.

Method	PTB-XL			CPSC2018		
	Accuracy	F1	AUROC	Accuracy	F1	AUROC
Linear Evaluation						
ST-MEM	0.726 ± 0.005	0.508 ± 0.008	0.838 ± 0.011	0.723 ± 0.008	0.641 ± 0.010	0.938 ± 0.002
PULSE (Ours)	$\mathbf{0.764 \pm 0.007}$	$\mathbf{0.632 \pm 0.009}$	$\mathbf{0.906 \pm 0.005}$	$\mathbf{0.725 \pm 0.012}$	$\mathbf{0.700 \pm 0.016}$	$\mathbf{0.946 \pm 0.004}$
Fine-tuning						
ST-MEM	$\mathbf{0.825 \pm 0.002}$	0.655 ± 0.003	0.933 ± 0.003	0.872 ± 0.009	0.857 ± 0.012	$\mathbf{0.980 \pm 0.001}$
PULSE (Ours)	0.818 ± 0.005	$\mathbf{0.679 \pm 0.016}$	$\mathbf{0.937 \pm 0.006}$	$\mathbf{0.878 \pm 0.009}$	$\mathbf{0.860 \pm 0.013}$	0.976 ± 0.006

Table 13 reports the evaluation metrics obtained directly on the Raspberry Pi together with the maximum absolute deviation from the corresponding workstation values. Across all four configurations, the maximum deviation is 1.1×10^{-4}, indicating close numerical agreement between the ONNX/ARM deployment and the original workstation evaluation. These results demonstrate that the compact PULSE encoder can be deployed on a low-resource edge device while maintaining both low inference latency and numerical consistency.

Table 12: Inference latency and memory overhead of PULSE on a Raspberry Pi 4B using ONNX Runtime with CPU execution and batch size one. Latency is measured per 10-second ECG recording. Memory denotes the increase in resident memory over the runtime baseline.

Dataset	Model	Mean (ms)	Min (ms)	Max (ms)	Memory (MB)
CPSC2018	12-lead	91.9	89.3	131.5	29.3
CPSC2018	Variable-lead	90.0	87.2	125.1	6.1
PTB-XL	12-lead	88.5	86.2	116.2	11.7
PTB-XL	Variable-lead	84.4	83.7	87.4	11.7

Table 13: Downstream metrics computed on the Raspberry Pi 4B on best fold and the maximum absolute deviation from the corresponding workstation values.

Dataset	Model	Acc.	Macro F1	AUROC	AUPRC	Max. Dev.
CPSC2018	12-lead	.8912	.8874	.9784	.9246	3.6×10^{-5}
CPSC2018	Variable-lead	.8169	.8067	.9619	.8597	1.1×10^{-4}
PTB-XL	12-lead	.8224	.6865	.9351	.7678	4.3×10^{-5}
PTB-XL	Variable-lead	.7596	.5944	.8860	.6513	3.7×10^{-5}

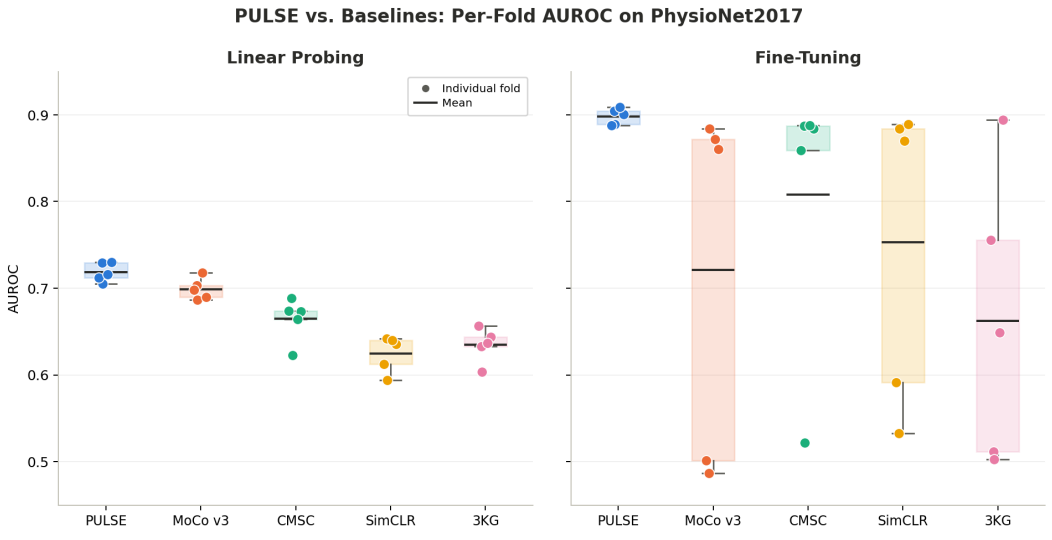


Figure 2: Per-fold AUROC on PhysioNet2017 for linear probing (left) and fine-tuning (right). Dots are the five folds and the line is the mean. Several baseline folds fall to chance level (AUROC $\approx$ 0.5) during fine-tuning, while PULSE stays stable.

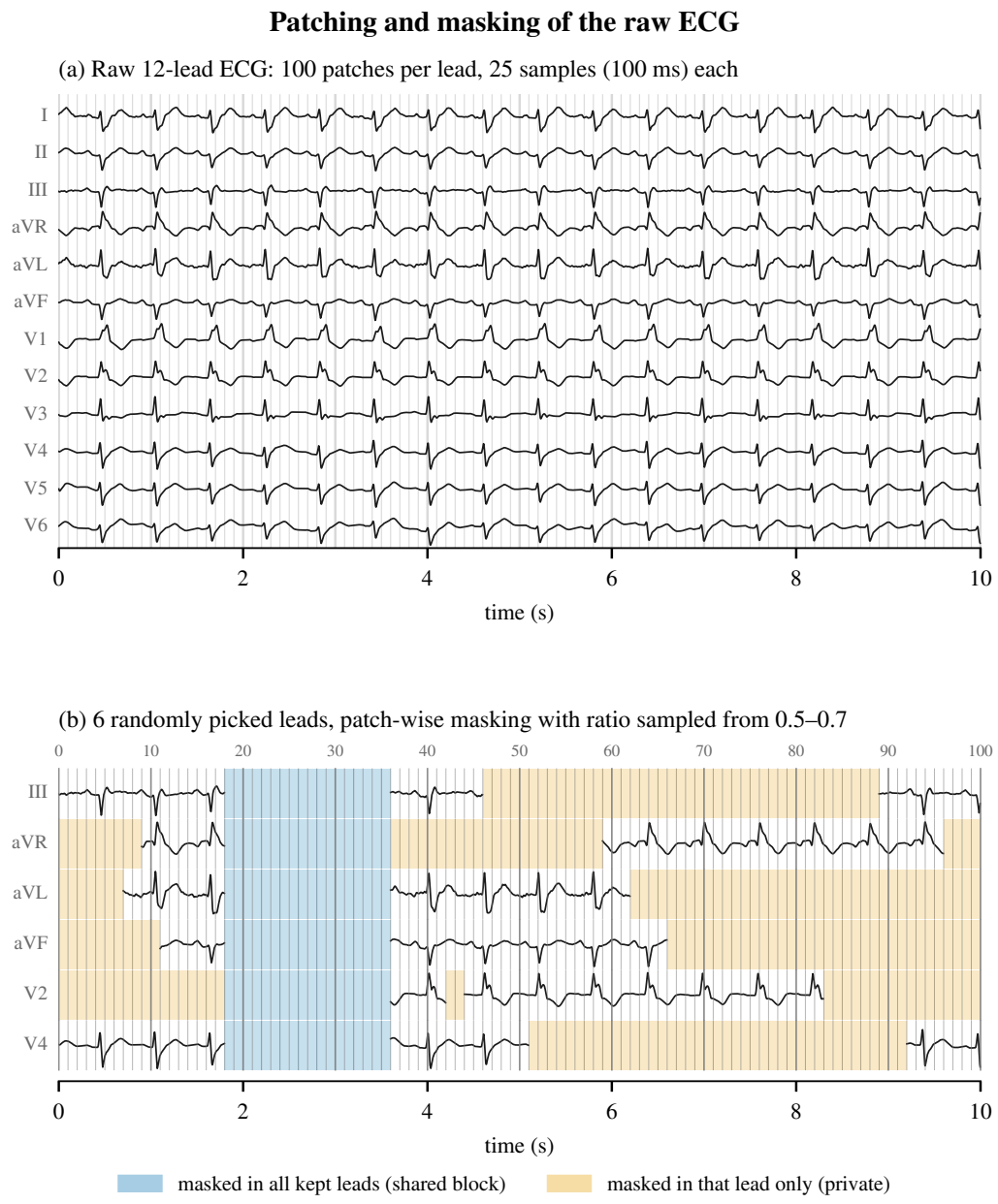


Figure 3: **Patching and masking of the raw ECG.** (a) A 10 s, 12-lead recording (250 Hz) is split into 100 non-overlapping patches of 25 samples (100 ms) per lead. (b) A random subset of leads is kept (six shown), and each kept lead is masked patch-wise with a ratio sampled from 0.5–0.7: a block shared by all kept leads (blue) plus lead-specific patches (orange). Vertical lines mark patch boundaries, and every tenth patch is numbered. Traces are rescaled per lead for display.